ADuCRF101 User Guide

UG-231

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

How to Set Up and Use the ADuCRF101

SCOPE

This user guide provides a detailed description of the ADuCRF101 functionality and features.

FUNCTIONAL BLOCK DIAGRAM

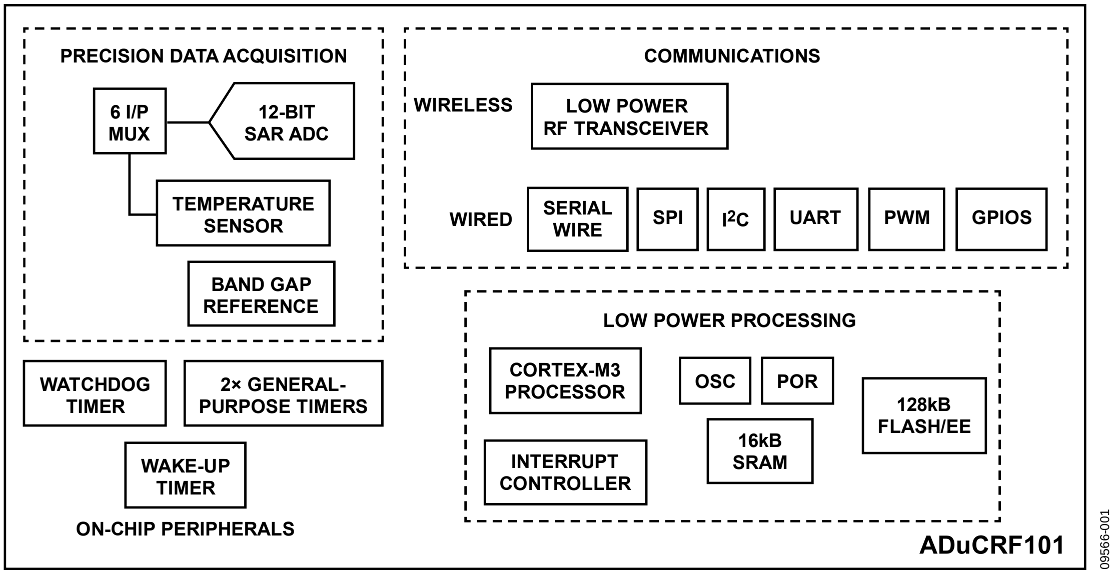


Figure 1.

PLEASE SEE THE LAST PAGE FOR AN IMPORTANT WARNING AND LEGAL TERMS AND CONDITIONS.

TABLE OF CONTENTS

REVISION HISTORY

11/14—Rev. 0 to Rev. A

4/13—Revision 0: Initial Version

USING THE ADuCRF101 USER GUIDE

NUMBER NOTATIONS

Table 1. Number Notations

Notation	Description
Bit N	Bits are numbered in little endian format, that is, the least significant bit of a number is referred to as Bit 0.
V[X:Y]	Bit field representation covering Bit X to Bit Y of a value or a field (V).
0xNN	Hexadecimal (Base 16) numbers are preceded by the prefix 0x.
0bNN	Binary (Base 2) numbers are preceded by the prefix 0b.
NN	Decimal (Base 10) are represented using no additional prefixes or suffixes.

REGISTER ACCESS CONVENTIONS

Table 2. Register Access Conventions

Mode	Description
RW	Memory location has read and write access.
R	Memory location is read access only. A read always returns 0, unless otherwise specified.
W	Memory location is write access only.

Note that register bit combinations not documented are reserved and should not be used.

ACRONYMS AND ABBREVIATIONS

Table 3.

Acronym/Abbreviation	Description
ADC	Analog-to-digital converter
AFC	Automatic frequency control
ARM	Advanced RISC machine
CRC	Cyclic redundancy check
CTS	Clear to send
DMA	Direct memory access
DMIPS	Dhrystone million instructions per second (MIPS)
DWT	Data watch point and trigger
FPB	Flash patch and breakpoint
GPIO	General-purpose input and output
I^2C	Inter IC
kB	Kilobyte
LFSR	Linear feedback shift register
LSB	Least significant byte/bit
MMR	Memory mapped register
MSB	Most significant byte/bit
NMI	Nonmaskable interrupt
NVIC	Nested vectored interrupt controller
PA	Power amplifier
PWM	Pulse-width modulation
RISC	Reduced instruction set computer
Rx	Receive
SAR	Successive approximation register
SPI	Serial peripheral interface
SWD	Sync word detect/serial wire debug
Tx	Transmit
UART	Universal asynchronous transmitter
WDT	Watch dog timer
WUT	Wake-up timer

INTRODUCTION TO THE ADuCRF101

The ADuCRF101 is a fully integrated data acquisition solution designed for low power wireless applications. It features a 12-bit ADC, a low power Cortex™-M3 processor from ARM®, a 431 MHz to 464 MHz and 862 MHz to 928 MHz RF transceiver, and Flash/EE memory packaged in a 9 mm × 9 mm LFCSP.

The acquisition section consists of a 12-bit ADC. The six inputs can be configured as single ended or differential. When configured in single-ended mode, they can be used for ratiometric measurements on sensors, powered from the internal LDO. An internal battery monitor channel and an on-chip temperature sensor are also available.

This wireless data acquisition system is designed to operate in battery-powered applications where low power is critical. The device can be configured in normal operating mode or different low power modes under direct program control. In flexi mode, any peripheral can operate and wake-up the device. In hibernate mode, the internal wake-up timer remains active. In shutdown mode, only an external interrupt can wake up the device.

The ADuCRF101 integrates a low power Cortex-M3 processor from ARM. It is a 32-bit RISC machine, offering up to 1.25 DMIPS peak performance. The Cortex-M3 processor also has a flexible 14-channel DMA controller supporting communication peripherals SPI, UART, and I^{2}C. In addition, 128 kB of nonvolatile Flash/EE memory and 16 kB of SRAM are provided on-chip.

A 16 MHz on-chip oscillator generates the system clock. This clock can be internally divided for the processor to operate at a lower frequency thus saving power. A low power internal 32 kHz oscillator is available and can be used to clock the timers. There are two general-purpose timers, a wake-up timer and a system watchdog timer.

A range of communication peripherals can be configured as required in a specific application. These peripherals include UART, I^{2}C, and SPI, GPIO ports, PWM, and RF transceiver.

The RF transceiver communicates in the 431 MHz to 464 MHz and 862 MHz to 928 MHz frequency bands using multiple configurations.

On-chip factory firmware supports in-circuit serial download via the UART while nonintrusive emulation and program download is also supported via the serial wire interface. These features are supported by the low cost development system.

The part operates from 2.2 V to 3.6 V and is specified over an industrial temperature range of −40 °C to +85 °C.

MAIN FEATURES OF THE ADuCRF101

Precision Data Acquisition

- Multichannel, 12-bit ADC.
- Ratiometric measurements supported.
- On-chip voltage reference and temperature sensor.

Low Power

- Operates directly from a battery.
- Supply range: 2.2 V to 3.6 V.
- Power consumption:
  - 280 nA, in power-down mode, nonretained state.
  - 1.9 µA, in power-down mode, processor memory, and RF transceiver memory retained.
  - 210 µA/MHz, Cortex-M3 processor in active mode.
  - 12.8 mA RF transceiver in receive mode; Cortex-M3 processor in power-down mode.
  - 9 mA to 32 mA RF transceiver in transmit mode; Cortex-M3 processor in power-down mode.

Communication

- High performance RF transceiver:
  - Frequency bands: 431 MHz to 464 MHz and 862 MHz to 928 MHz .
  - Multiple data rates supported.
  - Single-ended and differential PA with programmable output power.
- UART
  - Industry standard, 16450 UART peripheral.
  - Support for DMA.

- I^{2}C
  - 2-byte transmit and receive FIFOs for the master and slave.
  - Support for DMA.
- SPI
  - Master or slave mode with separate 4-byte Rx and Tx FIFOs.
  - Rx and Tx DMA channels.
- 8-channel PWM.
- 28-pin GPIO port.

Processing

- ARM Cortex-M3 32-bit processor operating from an internal 16 MHz oscillator.
- 128 kB Flash/EE memory, 16 kB SRAM.
- In-circuit download and debug via serial wire.
- On-chip UART download capability.

On-Chip Peripherals

- Two general-purpose timers.
- Wake-up timer.
- Watchdog timer.

Packages and Temperature Range

- 64-lead LFCSP (9 mm × 9 mm) package −40 °C to 85 °C.

Tools

- Low cost development system.
- Third-party compiler and emulator tool support.
- Various protocol stacks available.

Applications

- Battery-powered wireless sensor.
- Medical telemetry systems.
- Industrial and home automation.
- Asset tracking.
- Security systems (access systems).
- Health and fitness applications.

MEMORY ORGANIZATION

Features

- Cortex-M3 memory system features.
  - Predefined memory map.
  - Support for bit-band operation for atomic operations.
  - Unaligned data access.
- ADuCRF101 on-chip peripherals are accessed via memory mapped registers, situated in the bit band region.
- User memory.
  - 16 kB SRAM.
  - 128 kB Flash.
- On-chip kernel for manufacturer data and in-circuit download.

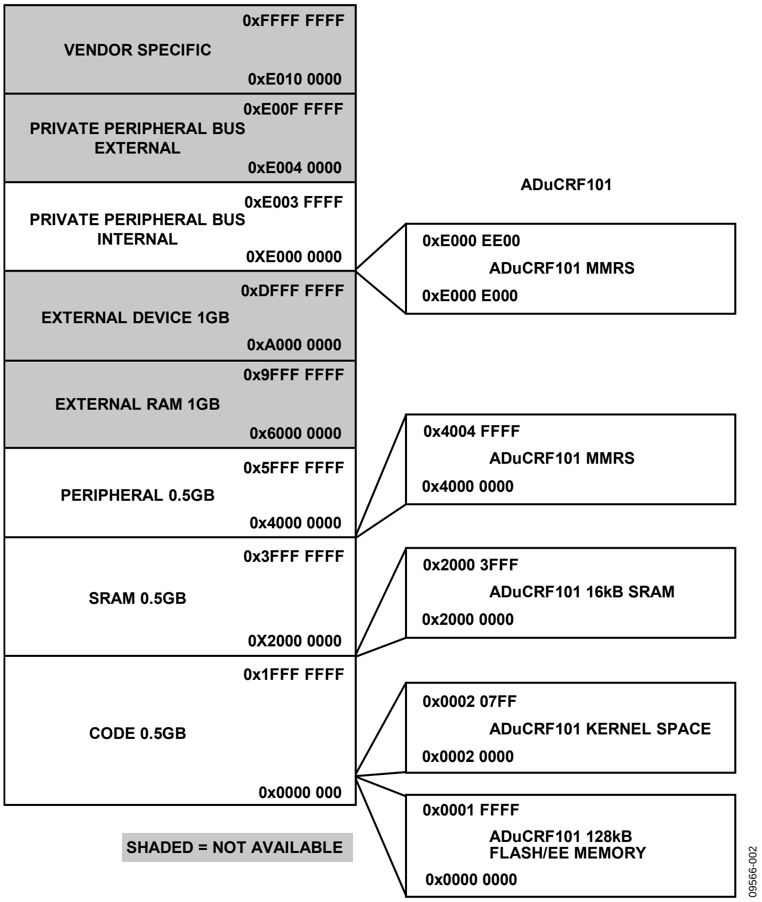


Figure 2. Cortex-M3 Memory Map Diagram

CORTEX-M3 PROCESSOR

CORTEX-M3 PROCESSOR FEATURES

High Performance

- 1.25 DMIPS/MHz.
- Many instructions, including multiply, are single cycle.
- Separate data and instruction buses allow simultaneous data and instruction accesses to be performed.
- Optimized for single-cycle flash usage.

Low Power

- Low standby current.
- Core implemented using advanced clock gating so that only the actively used logic consumes dynamic power.
- Power-saving mode support (sleep and deep sleep modes). The design has separate clocks to allow unused parts of the processor to be stopped.

Advanced Interrupt-Handling

- The nested vectored interrupt controller (NVIC) supports up to 240 interrupts. The ADuCRF101 implements some of these. The vectored interrupt feature greatly reduces interrupt latency because there is no need for software to determine which interrupt handler to serve. In addition, there is no need to have software to set up nested interrupt support.
- The Cortex-M3 processor automatically pushes registers onto the stack at the entry interrupt and pops them back at the exit interrupt. This reduces interrupt handling latency and allows interrupt handlers to be normal C functions.
- Dynamic priority controls for each interrupt.
- Latency reduction using late arrival interrupt acceptance and tail-chain interrupt entry.
- Immediate execution of a nonmaskable interrupt request for safety-critical applications.

System Features

- Support for bit-band operation and unaligned data access.
- Advanced fault handling features include various exception types and fault status registers.

Debug Support

- Serial wire debug interfaces (SW-DP).
- Flash patch and breakpoint (FPB) unit for implementing breakpoints. Six active breakpoints are available.
- Data watchpoint and trigger (DWT) unit for implementing watchpoints. Two active watchpoints are available.

CORTEX-M3 PROCESSOR OVERVIEW

The ADuCRF101 contains an embedded ARM Cortex-M3 processor, Revision r2p0. The ARM Cortex-M3 provides a high performance, low-cost platform that meets the system requirements of minimal memory implementation, reduced pin count and low power consumption while delivering outstanding computational performance and exceptional system response to interrupts.

CORTEX-M3 PROCESSOR OPERATION

Several Cortex-M3 components are flexible in their implementation. This section details the actual implementation of these components in the ADuCRF101.

Serial Wire Debug (SW/JTAG-DP)

The ADuCRF101 only supports the serial wire interface via the SWCLK and SWDIO pins. It does not support the 5-wire JTAG interface.

ROM Table

The ADuCRF101 implements the default ROM table.

Nested Vectored Interrupt Controller Interrupts (NVIC)

The Cortex-M3 processor includes a nested vectored interrupt controller (NVIC) which offers several features:

- Nested interrupt support
- Vectored interrupt support

- Dynamic priority changes support
- Interrupt masking

In addition, the NVIC has a nonmaskable interrupt (NMI) input.

The NVIC is implemented on the ADuCRF101, and more details are available in the System Exceptions and Peripheral Interrupts section.

Wake-Up Interrupt Controller (WIC)

The ADuCRF101 has a modified WIC, which provides the lowest possible power-down current. This feature is transparent to the user and more details are available in the Power Management Unit section. It is not recommended to enter a power saving mode while servicing an interrupt. However, if the part does enter a power saving mode while servicing an interrupt it can only be woken up by a higher priority interrupt source.

µDMA

The ADuCRF101 implements the ARM µDMA. More details are available in the DMA Controller section.

RELATED DOCUMENTS

- Cortex-M3 Revision r2p0 Technical Reference Manual (DDI0337)
- Cortex-M3 Errata Notice: Cortex-M3/Cortex-M3 with ETM (AT420/AT425)
- ARMv7-M Architecture Reference Manual (DDI0403)
- ARMv7-M Architecture Reference Manual Errata Markup
- ARM Debug Interface v5 (IHI0031)
- PrimeCell µDMA Controller (PL230) Technical Reference Manual Revision r0p0 (DDI0417)

SYSTEM EXCEPTIONS AND PERIPHERAL INTERRUPTS

The ADuCRF101 integrates a Cortex-M3 processor, which supports a number of system exceptions and interrupts generated by peripherals. Table 4 lists the Cortex-M3 system exceptions.

Table 4. System Exceptions

Exception Number	Type	Priority	Description
1	Reset	−3 (highest)	Any reset
2	NMI	−2	Nonmaskable interrupt connected to power supply monitor of ADuCRF101.
3	Hard fault	−1	All fault conditions, if the corresponding fault handler is not enabled.
4	Memory management fault	Programmable	Memory management fault; access to illegal locations.
5	Bus fault	Programmable	Prefetch fault, memory access fault, data abort, and other address/memory related faults.
6	Usage fault	Programmable	Faults such as undefined instruction executed or illegal state transition attempt.
7 to 10	Reserved	N/A	
11	SVCall	Programmable	System service call with SVC instruction. Used for system function calls.
12	Debug monitor	Programmable	Debug monitor (breakpoint, watchpoint, or external debug requests).
13	Reserved	N/A	
14	PendSV	Programmable	Pendable request for system service. Used for queuing system calls until other tasks and interrupts are serviced.
15	SYSTICK	Programmable	System tick timer.

Reset is functional in any of the low power modes.

The nonmaskable interrupt (NMI) is connected to the power supply monitor of the ADuCRF101. The NMI wakes up the ADuCRF101 from Hibernate and Flexi modes. It is not available in shutdown mode.

The peripheral interrupts are controlled by the nested vectored interrupt controller and are listed in Table 5. Only a limited number of interrupts are capable of waking the processor up from hibernate or shutdown mode as shown in Table 5. All peripheral interrupts must also be configured in the corresponding peripheral. The RF transceiver and pin interrupts are configured in the external interrupt detection unit. See the Pin and RF Transceiver Interrupts Configuration section.

Two steps are normally required to configure an interrupt

1. Configure a peripheral to generate an interrupt request to the NVIC.
2. Configure the NVIC for that peripheral request.

Table 5. Peripherals Interrupts

			Wake-Up Processor from		
Number	Vector	Peripheral	Flexi Mode	Hibernate Mode	Shutdown Mode
0	Wake-up timer	Wake-up timer	Yes	Yes	No
1	External Interrupt 0	RF transceiver	Yes	Yes	Yes
2	External Interrupt 1	Pin	Yes	Yes	Yes
3	External Interrupt 2	Pin	Yes	Yes	No
4	External Interrupt 3	Pin	Yes	Yes	No
5	External Interrupt 4	Pin	Yes	Yes	No
6	External Interrupt 5	Pin	Yes	Yes	No
7	External Interrupt 6	Pin	Yes	Yes	No
8	External Interrupt 7	Pin	Yes	Yes	No
9	External Interrupt 8	RF transceiver	Yes	Yes	Yes
10	Watchdog timer	Watchdog timer	Yes	Yes	No
11	Reserved	Reserved	N/A	N/A	N/A
12	Timer0	Timer0	Yes	No	No
13	Timer1	Timer1	Yes	No	No
14	ADC	ADC	Yes	No	No
15	Flash controller	Flash controller	Yes	No	No
16	UART	UART	Yes	No	No
17	SPI0	SPI0, interface to RF transceiver	Yes	No	No
18	SPI1	SPI1, user SPI	Yes	No	No
19	I^{2}C slave	I^{2}C	Yes	No	No
20	I^{2}C master	I^{2}C	Yes	No	No
21 to 22	Reserved	Reserved	N/A	N/A	N/A
23	DMA error	DMA controller	Yes	No	No
24	DMA SPI1 TX IRQ	DMA controller	Yes	No	No
25	DMA SPI1 RX IRQ	DMA controller	Yes	No	No
26	DMA UART TX IRQ	DMA controller	Yes	No	No
27	DMA UART RX IRQ	DMA controller	Yes	No	No
28	DMA I^{2}C slave TX	DMA controller	Yes	No	No
29	DMA I^{2}C slave RX	DMA controller	Yes	No	No
30	DMA I^{2}C master TX	DMA controller	Yes	No	No
31	DMA I^{2}C master RX	DMA controller	Yes	No	No
32 to 34	Reserved	Reserved	N/A	N/A	N/A
35	ADC DMA	DMA controller	Yes	No	No
36	DMA SPI0 TX IRQ	DMA controller	Yes	No	No
37	DMA SPI0 RX IRQ	DMA controller	Yes	No	No
38	PWM_TRIP	PWM	Yes	No	No
39	PWM0	PWM	Yes	No	No
40	PWM1	PWM	Yes	No	No
41	PWM2	PWM	Yes	No	No
42	PWM3	PWM	Yes	No	No

The priority of the peripheral interrupts is programmable. Eight levels of priority are available.

Internally to the Cortex-M3 processor, the highest user-programmable priority (0) is treated as fourth priority, after a reset, NMI, and a hard fault.

Note that 0 is the default priority for all the programmable priorities. If the same priority level is assigned to two or more interrupts, their hardware priority (the lower the position number) determines the order in which the processor activates them. For example, if both SPI0 and SPI1 are Priority Level 1, then SPI0 has higher priority.

To enable an interrupt for any peripheral listed from 0 to 31 in Table 5, set the appropriate bit in the ISER0 register—ISER0 is a 32-bit register and each bit corresponds to the first 32 entries of Table 5.

For example, to enable ADC interrupt source in the NVIC, set ISER0[14] = 1. Similarly, to disable ADC interrupt, set ICER0[14] = 1.

To enable an interrupt for any peripheral numbered from 32 to 42 in Table 5, set the appropriate bit in the ISER1 register—ISER1 is a 32-bit register and ISER1 Bit 0 to Bit 10 correspond to the entries 32 to 42 of Table 5.

For example, to enable the PWM0 interrupt source in the NVIC, set ISER1[7] = 1. Similarly, to disable the PWM0 interrupt, set ICER1[7] = 1.

Table 6 lists the NVIC registers to enable and disable relevant interrupts and set the priority levels.

Table 6. NVIC Registers

Address	Analog Devices Header File Name	Description	Access
0xE000E100	ISER0	Set IRQ 0 to IRQ 31 enable	RW
0xE000E104	ISER1	Set IRQ 32 to IRQ 42 enable	RW
0xE000E180	ICER0	Clear IRQ 0 to IRQ 31 enable	RW
0xE000E184	ICER1	Clear IRQ 32 to IRQ 42 enable	RW
0xE000E200	ISPR0	Set IRQ 0 to IRQ 31 pending	RW
0xE000E204	ISPR1	Set IRQ 32 to IRQ 42 pending	RW
0xE000E280	ICPR0	Clear IRQ 0 to IRQ 31 pending	RW
0xE000E284	ICPR1	Clear IRQ 32 to IRQ 42 pending	RW
0xE000E400	IPR0	IRQ 0 to IRQ 3 priority	RW
0xE000E404	IPR1	IRQ 4 to IRQ 7 priority	RW
0xE000E408	IPR2	IRQ 8 to IRQ 11 priority	RW
0xE000E40C	IPR3	IRQ 12 to IRQ 15 priority	RW
0xE000E410	IPR4	IRQ 16 to IRQ 19 priority	RW
0xE000E414	IPR5	IRQ 20 to IRQ 23 priority	RW
0xE000E418	IPR6	IRQ 24 to IRQ 27 priority	RW
0xE000E41C	IPR7	IRQ 28 to IRQ 31 priority	RW
0xE000E420	IPR8	IRQ 32 to IRQ 35 priority	RW
0xE000E424	IPR9	IRQ 36 to IRQ 39 priority	RW
0xE000E428	IPR10	IRQ 40 to IRQ 42 priority	RW

PIN AND RF TRANSCEIVER INTERRUPTS CONFIGURATION

Nine external interrupts are implemented. The two RF transceiver interrupts and seven pin interrupts can be separately configured to detect any combination of the following type of events:

- Edge: rising edge, falling edge, or both rising and falling edges. An interrupt signal (pulse) is sent to the NVIC upon detecting a transition from low to high, high to low, or on either high to low or low to high.
- Level: high or low. An interrupt signal is generated and remains asserted in the NVIC until the conditions generating the interrupt deassert. The level needs to be maintained for a minimum of one core clock cycle to be detected.
- These nine interrupt sources can wake-up the device when in hibernate mode. Only external interrupt 0, external interrupt 1, and external interrupt 8 (RF transceiver) can wake-up the device from shutdown mode.

External interrupt 0 and external interrupt 7 are used internally and are not available on a pin.

External interrupt 8 is also used internally and, while visible on an external pin, should only be used for monitoring purposes.

After an external interrupt has been asserted by the interrupt detection unit (IDU) an interrupt pulse to the Cortex-M3 processor is generated. Unless this external interrupt is cleared, subsequent interrupts on the same port are ignored by the IDU, and no further pulses are generated.

REGISTER SUMMARY (INTERRUPTS)

Table 7. Interrupts Register Summary

Address	Name	Description	Reset	RW
0x40002420	EI0CFG	External Interrupt Configuration Register 0	0x0000	RW
0x40002424	EI1CFG	External Interrupt Configuration Register 1	0x0000	RW
0x40002428	EI2CFG	External Interrupt Configuration Register 2	0x0000	RW
0x40002430	EICLR	External Interrupts Clear Register	0x0000	RW
0x40002434	NMICLR	NMI Clear Register	0x00	RW

REGISTER DETAILS (INTERRUPTS)

External Interrupt Configuration Register 0

Address: 0x40002420, Reset: 0x0000, Name: EI0CFG

Table 8. Bit Descriptions for EI0CFG

Bits	Bit Name	Description	Reset	Access
15	IRQ3EN	External Interrupt 3 enable bit. 0: DIS. External Interrupt 3 disabled. 1: EN. External Interrupt 3 enabled.	0x0	RW
[14:12]	IRQ3MDE	External Interrupt 3 detection mode. 000: RISE. Rising edge. 001: FALL. Falling edge. 010: RISEORFALL. Rising or falling edge. 011: HIGHLEVEL. High level. 100: LOWLEVEL. Low level.	0x0	RW
11	IRQ2EN	External Interrupt 2 enable bit. 0: DIS. External Interrupt 2 disabled. 1: EN. External Interrupt 2 enabled.	0x0	RW
[10:8]	IRQ2MDE	External Interrupt 2 detection mode. 000: RISE. Rising edge. 001: FALL. Falling edge. 010: RISEORFALL. Rising or falling edge. 011: HIGHLEVEL. High level. 100: LOWLEVEL. Low level.	0x0	RW
7	IRQ1EN	External Interrupt 1 enable bit. 0: DIS. External Interrupt 1 disabled. 1: EN. External Interrupt 1 enabled.	0x0	RW
[6:4]	IRQ1MDE	External Interrupt 1 detection mode. 000: RISE. Rising edge. 001: FALL. Falling edge. 010: RISEORFALL. Rising or falling edge. 011: HIGHLEVEL. High level. 100: LOWLEVEL. Low level.	0x0	RW
3	IRQ0EN	External Interrupt 0 enable bit. 0: DIS. RF transceiver clock IRQ disabled. 1: EN. RF transceiver clock IRQ enabled.	0x0	RW
[2:0]	IRQ0MDE	External Interrupt 0 detection mode. 000: RISE. Rising edge. 001: FALL. Falling edge. 010: RISEORFALL. Rising or falling edge. 011: HIGHLEVEL. High level. 100: LOWLEVEL. Low level.	0x0	RW

External Interrupt Configuration Register 1

Address: 0x40002424, Reset: 0x0000, Name: EI1CFG

Table 9. Bit Descriptions for EI1CFG

Bits	Bit Name	Description	Reset	Access
15	IRQ7EN	External Interrupt 7 enable bit. 0: DIS. External Interrupt 7 disabled. 1: EN. External Interrupt 7 enabled.	0x0	RW
[14:12]	IRQ7MDE	External Interrupt 7 detection mode. 000: RISE. Rising edge. 001: FALL. Falling edge. 010: RISEORFALL. Rising or falling edge. 011: HIGHLEVEL. High level. 100: LOWLEVEL. Low level.	0x0	RW
11	IRQ6EN	External Interrupt 6 enable bit. 0: DIS. External Interrupt 6 disabled. 1: EN. External Interrupt 6 enabled.	0x0	RW
[10:8]	IRQ6MDE	External Interrupt 6 detection mode. 000: RISE. Rising edge. 001: FALL. Falling edge. 010: RISEORFALL. Rising or falling edge. 011: HIGHLEVEL. High level. 100: LOWLEVEL. Low Level.	0x0	RW
7	IRQ5EN	External Interrupt 5 enable bit. 0: DIS. External Interrupt 5 disabled. 1: EN. External Interrupt 5 enabled.	0x0	RW
[6:4]	IRQ5MDE	External Interrupt 5 detection mode. 000: RISE. Rising edge. 001: FALL. Falling edge. 010: RISEORFALL. Rising or falling edge. 011: HIGHLEVEL. High level. 100: LOWLEVEL. Low Level.	0x0	RW
3	IRQ4EN	External Interrupt 4 enable bit. 0: DIS. External Interrupt 4 disabled. 1: EN. External Interrupt 4 enabled.	0x0	RW
[2:0]	IRQ4MDE	External Interrupt 4 detection mode. 000: RISE. Rising edge. 001: FALL. Falling edge. 010: RISEORFALL. Rising or falling edge. 011: HIGHLEVEL. High level. 100: LOWLEVEL. Low Level.	0x0	RW

External Interrupt Configuration Register 2

Address: 0x40002428, Reset: 0x0000, Name: EI2CFG

Table 10. Bit Descriptions for EI2CFG

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x000	R
3	IRQ8EN	External Interrupt 8 (RF transceiver) enable bit. 0: DIS. RF transceiver IRQ disabled. 1: EN. RF transceiver IRQ enabled.	0x0	RW
[2:0]	IRQ8MDE	External Interrupt 8 (RF transceiver) detection mode. 000: RISE. Rising edge. 001: FALL. Falling edge. 010: RISEORFALL. Rising or falling edge. 011: HIGHLEVEL. High level. 100: LOWLEVEL. Low level.	0x0	RW

External Interrupts Clear Register

Address: 0x40002430, Reset: 0x0000, Name: EICLR

Table 11. Bit Descriptions for EICLR

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x00	R
8	IRQ8	External Interrupt 8 (RF transceiver) clear bit. Notes: Cleared automatically by hardware. 1: CLR. Clear an internal interrupt flag.	0x0	RW
7	IRQ7	External Interrupt 7 clear bit. Notes: Cleared automatically by hardware. 1: CLR. Clear an internal interrupt flag.	0x0	RW
6	IRQ6	External Interrupt 6 clear bit. Notes: Cleared automatically by hardware. 1: CLR. Clear an internal interrupt flag.	0x0	RW
5	IRQ5	External Interrupt 5 clear bit. Notes: Cleared automatically by hardware. 1: CLR. Clear an internal interrupt flag.	0x0	RW
4	IRQ4	External Interrupt 4 clear bit. Notes: Cleared automatically by hardware. 1: CLR. Clear an internal interrupt flag.	0x0	RW
3	IRQ3	External Interrupt 3 clear bit. Notes: Cleared automatically by hardware. 1: CLR. Clear an internal interrupt flag.	0x0	RW
2	IRQ2	External Interrupt 2 clear bit. Notes: Cleared automatically by hardware. 1: CLR. Clear an internal interrupt flag.	0x0	RW
1	IRQ1	External Interrupt 1 clear bit. Notes: Cleared automatically by hardware. 1: CLR. Clear an internal interrupt flag.	0x0	RW
0	IRQ0	External Interrupt 0 clear bit. Notes: Cleared automatically by hardware. 1: CLR. Clear an internal interrupt flag.	0x0	RW

NMI Clear Register

Address: 0x40002434, Reset: 0x00, Name: NMICLR

Table 12. Bit Descriptions for NMICLR

Bits	Bit Name	Description	Reset	Access
[7:1]	RESERVED	Reserved.	0x00	R
0	CLEAR	NMI clear bit. Notes: Cleared automatically by hardware. 1: EN. Clear an internal interrupt flag when the NMI interrupt is set.	0x0	RW

CLOCKING ARCHITECTURE

FEATURES

- The ADuCRF101 integrates two on-chip oscillators and circuitry for two external crystals:
  - HFOSC is a 16 MHz internal oscillator, used in active and flexi mode.
  - LFOSC is a 32 kHz, low power internal oscillator, on at all times except at reset and in shutdown mode.
  - LFXTAL is an optional 32 kHz external crystal, which can be used for more accurate timer clocking.
  - HFXTAL is a 26 MHz external crystal, used to set the RF transceiver communication frequency. (connect to XOSC26P and XOSC26N pin).
- Power saving clock mechanism: clocks to individual peripherals can be disabled when that peripheral is not being used.

CLOCKING ARCHITECTURE BLOCK DIAGRAM

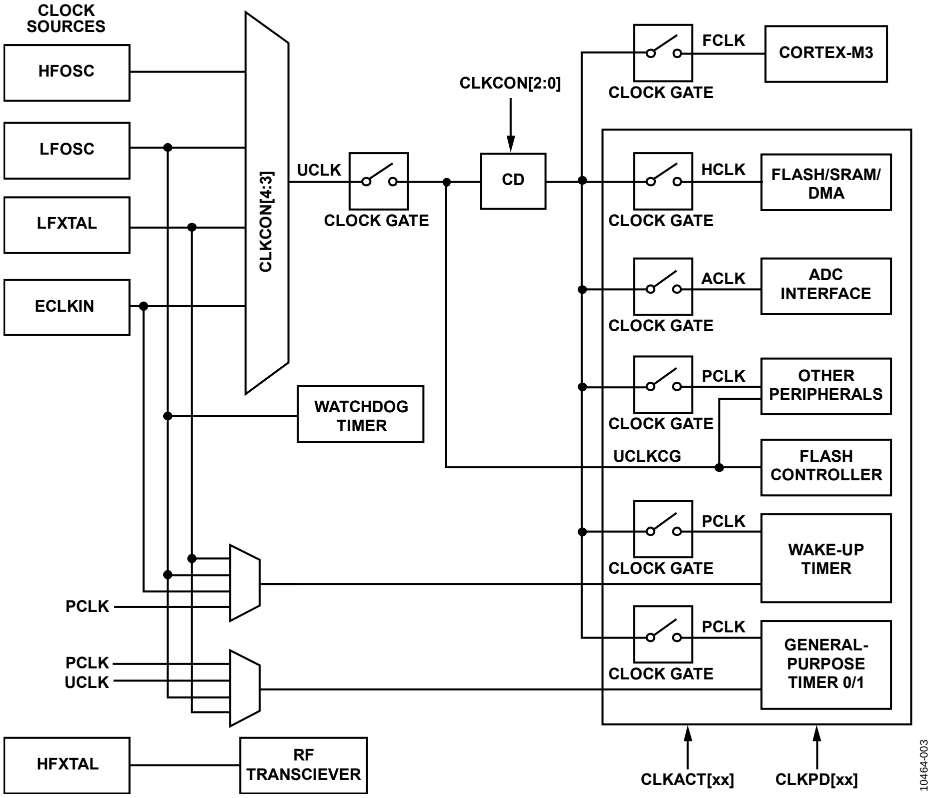


Figure 3. Clocking Architecture Block Diagram

CLOCKING ARCHITECTURE OVERVIEW

Three of the clock sources, HFOSC, LFOSC, and LFXTAL, can be used as system clocks. An external clock on P0.5 (ECLKIN) can also be used for test purposes.

Internally the system clock is divided into five clocks:

- FCLK for the core
- HCLK for the Flash, SRAM, and DMA
- ACLK for the ADC interface
- PCLK for the other peripherals.
- UCLK system clock

Figure 3 shows all the clocks available and includes clock gates for power management. More details on the clock gates can be found in the Power Management Unit section.

CLOCKING ARCHITECTURE OPERATION

At power-up, the core executes from the 16 MHz internal oscillator. User code can select the clock source for the system clock and can divide the clock by a factor of 2^{CD}, where CD is the clock divider bits (CLKCON[2:0]). This allows slower code execution and reduced power consumption.

UCLK is also passed to some of the serial peripherals so that the timings are not affected by CD changes.

For test purposes, the two internal oscillators, external crystal or UCLK can be brought out to P0.4 (ECLKOUT).

An external clock on P0.5 (ECKLIN) can also be used as the system clock: P0.5 must be configured first as a clock input and the external clock applied to P0.5 before switching clock source in the clock control register (CLKCON).

REGISTER SUMMARY (CLOCK CONTROL)

Table 13. Clock Control Register Summary

Address	Name	Description	Reset	RW
0x40002000	CLKCON	System Clocking Architecture Control Register	0x0000	RW
0x40002410	XOSCCON	Crystal Oscillator Control Register	0x00	RW
0x40002480	CLKACT	Clock in Active Mode Enable Register	0x3FFF	RW
0x40002484	CLKPD	Clock in Power-Down Mode Enable Register	0x3FFF	RW

REGISTER DETAILS (CLOCK CONTROL)

System Clocking Architecture Control Register

Address: 0x40002000, Reset: 0x0000, Name: CLKCON

Table 14. Bit Descriptions for CLKCON

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	R
[7:5]	CLKOUT	GPIO output clock multiplexer select bits. 000: UCLKCG. 001: UCLK. 010: PCLK. 101: HFOSC. 110: LFOSC. 111: LFXTAL.	0x0	RW
[4:3]	CLKMUX	Digital subsystem clock source select bits. Notes: These bits select the clock source for the system. The recommended and default clock source is HFOSC. 00: HFOSC. 16 MHz internal oscillator. 01: LFXTAL. 32.768 kHz external crystal. 10: LFOSC. 32.768 kHz internal oscillator. 11: ECLKIN. External clock on P0.5.	0x0	RW
[2:0]	CD	Clock divide bits. 000: DIV1. 001: DIV2. 010: DIV4. 011: DIV8. 100: DIV16. 101: DIV32. 110: DIV64. 111: DIV128.	0x0	RW

Crystal Oscillator Control Register

Address: 0x40002410, Reset: 0x00, Name: XOSCCON

Table 15. Bit Descriptions for XOSCCON

Bits	Bit Name	Description	Reset	Access
[7:1]	RESERVED	Reserved.	0x00	R
0	ENABLE	Crystal oscillator circuit enable bit. 0: DIS. Disables the watch crystal circuitry.(LFXTAL). 1: EN. Enables the watch crystal circuitry.(LFXTAL).	0x0	RW

Clock in Active Mode Enable Register

Address: 0x40002480, Reset: 0x3FFF, Name: CLKACT

Table 16. Bit Descriptions for CLKACT

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	Reserved.	0x0	R
[13:12]	RESERVED	Reserved. 0 should be written to these bits.	0x3	RW
11	T1	Timer 1 clocks enable bit. 0: DIS. Disable T1 clocks. 1: EN. Enable T1 clocks.	0x1	RW
10	T0	Timer 0 clocks enable bit. 0: DIS. Disable T0 clocks. 1: EN. Enable T0 clocks.	0x1	RW
9	PWM	PWM clocks enable bit. 0: DIS. Disable PWM clocks. 1: EN. Enable PWM clocks.	0x1	RW
8	I2C	I2C clocks enable bit. 0: DIS. Disable I2C clocks. 1: EN. Enable I2C clocks.	0x1	RW
7	COM	UART clocks enable bit. 0: DIS. Disable UART clocks. 1: EN. Enable UART clocks.	0x1	RW
6	SPI1	SPI1 clocks enable bit. 0: DIS. Disable SPI1 clocks. 1: EN. Enable SPI1 clocks.	0x1	RW
5	SPI0	SPI0 clocks enable bit. 0: DIS. Disable SPI0 clocks. 1: EN. Enable SPI0 clocks.	0x1	RW
4	T2	Timer 2 clocks enable bit. 0: DIS. Disable T2 clocks. 1: EN. Enable T2 clocks.	0x1	RW
3	ADC	ADC clocks enable bit. 0: DIS. Disable ADC clocks. 1: EN. Enable ADC clocks.	0x1	RW
2	SRAM	SRAM clocks enable bit. Notes: The protection key must be entered in PWRKEY before modifying this bit. Flash and SRAM clocks cannot be disabled at the same time in active mode; at least one active memory is required to execute code. Writing 0 to the FEE and SRAM bits at the same time will not have any effect. 0: DIS. Disable SRAM memory clocks. 1: EN. Enable SRAM memory clocks.	0x1	RW

Bits	Bit Name	Description	Reset	Access
1	FEE	Flash clocks enable bit. Note: the protection key must be entered in PWRKEY before modifying this bit. Flash and SRAM clocks cannot be disabled at the same time in active mode, at least one active memory is required to execute code. Writing 0 to the FEE and SRAM bits at the same time will not have any effect. 0: DIS. Disable Flash memory clocks. 1: EN. Enable Flash memory clocks.	0x1	RW
0	DMA	DMA clock enable bit. 0: DIS.Disable DMA clock. 1: EN. Enable DMA clock.	0x1	RW

Clock In Power-Down Mode Enable Register

Address: 0x40002484, Reset: 0x3FFF, Name: CLKPD

Table 17. Bit Descriptions for CLKPD

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	Reserved.	0x0	R
[13:12]	RESERVED	Reserved. 0 should be written to these bits.	0x3	RW
11	T1	Timer 1 clocks enable bit. 0: DIS. Disable T1 clocks. 1: EN. Enable T1 clocks.	0x1	RW
10	T0	Timer 0 clocks enable bit. 0: DIS. Disable T0 clocks. 1: EN. Enable T0 clocks.	0x1	RW
9	PWM	PWM clocks enable bit. 0: DIS. Disable PWM clocks. 1: EN. Enable PWM clocks.	0x1	RW
8	I2C	I^{2}C clocks enable bit. 0: DIS. Disable I2C clocks. 1: EN. Enable I2C clocks.	0x1	RW
7	COM	UART clocks enable bit. 0: DIS. Disable UART clocks. 1: EN. Enable UART clocks.	0x1	RW
6	SPI1	SPI1 clocks enable bit. 0: DIS. Disable SPI1 clocks. 1: EN. Enable SPI1 clocks.	0x1	RW
5	SPI0	SPI0 clocks enable bit. 0: DIS. Disable SPI0 clocks. 1: EN. Enable SPI0 clocks.	0x1	RW
4	T2	Timer 2 clocks enable bit. 0: DIS. Disable T2 clocks. 1: EN. Enable T2 clocks.	0x1	RW
3	ADC	ADC clocks enable bit. 0: DIS. Disable ADC clocks. 1: EN. Enable ADC clocks.	0x1	RW
2	SRAM	SRAM clocks enable bit. 0: DIS. Disable SRAM memory clocks. 1: EN. Enable SRAM memory clocks.	0x1	RW
1	FEE	Flash clocks enable bit. 0: DIS. Disable Flash memory clocks. 1: EN. Enable Flash memory clocks.	0x1	RW
0	DMA	DMA clock enable bit. 0: DIS. Disable DMA clock. 1: EN. Enable DMA clock.	0x1	RW

GENERAL-PURPOSE TIMERS

GENERAL-PURPOSE TIMERS (TIMER 0/TIMER 1) FEATURES

- Two identical, general-purpose, 16-bit count-up/count-down timers
    - Timer 0 and Timer 1
- Clocked from four different clock sources (can be scaled down using a prescalar of 1, 16, 256, or 32,768)
    - 16 MHz system clock (UCLK)
    - Peripheral clock (PCLK)
    - 32 kHz internal oscillator (LFOSC)
    - 32 kHz external crystal (LFXTAL)
- Two modes
    - Free-running
    - Periodic
- Capture events feature
    - Capability to capture 15 different events on each timer

GENERAL-PURPOSE TIMERS BLOCK DIAGRAM

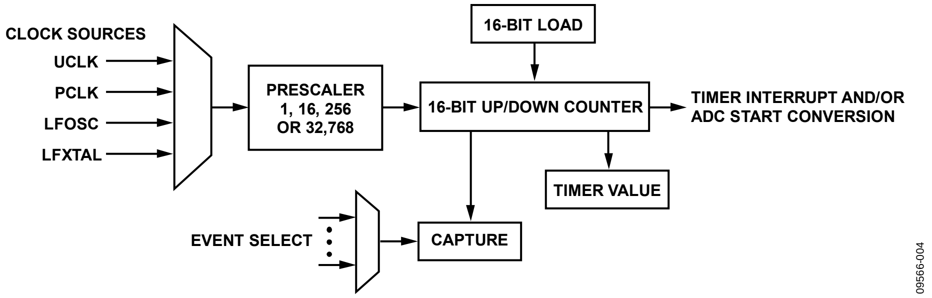


Figure 4. General-Purpose Timers Block Diagram

GENERAL-PURPOSE TIMERS OVERVIEW

Timer 0 and Timer 1 are two identical, general-purpose, 16-bit count-up/count-down timers. They can be clocked from four different clock sources: UCLK, PCLK, 32 kHz internal oscillator (LFOSC), and 32 KHz external crystal (LFXTAL). This clock source can be scaled down using a prescalar of 1, 16, 256, or 32,768.

The timers can be either free-running or periodic:

- In free running mode, the counter decrements from full scale to zero scale or increments from zero scale to full scale and then restarts.
- In periodic mode, the counter decrements or increments from the value in the load register (TxLD MMR) until zero scale or full scale is reached and then restarts at the value stored in the load register.

The value of a counter can be read at any time by accessing its value register (TxVAL).

This register is synchronous to UCLK. Therefore, when a timer is clocked from a clock other than UCLK, TxVAL reflects the latest timer value.

The TxCON register selects the timer mode, configures the clock source, selects count-up or down, starts the counter, and controls the event capture function.

An interrupt signal is generated each time the value of the counter reaches 0 when counting down, or each time the counter value reaches the maximum value when counting up. This interrupt signal is cleared by writing 1 to the clear interrupt register of that particular timer (TxCLRI).

In addition, Timer 0 and Timer 1 have a capture register (TxCAP) that is triggered by a selected interrupt event. When triggered, the current timer value is copied to TxCAP, and the timer continues to run. This feature can be used to determine the assertion of an event with increased accuracy. Each timer can capture 15 different events, listed in Table 18.

The maximum interval that can be achieved is 65,536 seconds using LFOSC or LFXTAL. The maximum resolution is 250 ns using UCLK or an undivided PCLK.

The general-purpose timers can be used in flexi mode but are off in hibernate mode.

Timer 0 and Timer 1 can be used to automatically trigger an ADC Conversion (see the ADC Circuit section).

GENERAL-PURPOSE TIMERS OPERATION

Free Running Mode

In free running mode, the timer is started by setting the enable bit (TxCON[4]) to 1 and the MOD bit (TxCON[3]) to 0. The timer increments from zero scale/full scale to full scale/zero scale if counting up/down. Full scale is 0xFFFF. On reaching full scale (or zero), a timeout interrupt occurs and TxSTA[0] is set. To clear it, user code must write 1 to TxCLRI[0].

Periodic Mode

In periodic mode,the timer is started by setting the enable bit (TxCON[4]) to 1 and the MOD bit (TxCON[3]) to 1. The initial TxLD value should be loaded before starting the timer.The timer value increments from the value stored in the TxLD register to full scale or decrements from the value stored in the TxLD register to zero, depending on the TxCON[2] settings (count-up/down). On reaching full scale or zero, the timer generates an interrupt. The TxLD is reloaded into TxVAL and the timer continues counting up/down. The timer should be disabled prior to changing the TxCON or TxLD register. If the TxLD register is changed while the timer is being reloaded, undefined results can occur. By default, the counter is reloaded automatically when generating the interrupt signal. If TxCON[7] is set to 1, the counter is also reloaded when user code writes TxCLRI. This allows user changes to the TxLD register to take effect immediately and not on the next timeout.

The timer interval is calculated as follows:

If the timer is set to count down then

$$Interval = (TxLD \times Prescaler)/Source\ Clock$$

For example, if TxLD = 0x100, prescaler = 4 and clock source, UCLK =16 MHz, the interval is 64 µs.

If the timer is set to count up then

$$Interval = ((Full\ Scale - TxLD) \times Prescaler)/Source\ Clock.$$

Clock Selection

Clock selection is made by setting TxCON[6:5].

If LFXTAL is selected, ensure that clock circuit is on (XOSCCON[0] = 1).

If the selected clock source is UCLK or PCLK, configuring TxCON[1:0] = 00 results in a prescaler of 4.

Asynchronous Clock Source

Timers are started by setting the enable bit (TxCON[4]) to 1 in the control register of the corresponding timer.

However, when the timer clock source is LFXTAL or LFOSC, some precautions must be taken:

- The control register (TxCON) should not be written if TxSTA[6] is set. Therefore TxSTA should be read prior to configuring the control register(TxCON). When TxSTA[6] is cleared, the register can be modified. This ensures that synchronizing timer control between the core and timer clock domains is complete.
- After clearing the interrupt in TxCLRI, it must be ensured that the register write has fully completed before returning from the interrupt handler. Use the data synchronization barrier (DSB) instruction if necessary and check that TxSTA[7] = 0.
- The value of a counter can be read at any time by accessing its value register (TxVAL). In an asynchronous configuration, TxVAL should always be read twice. If the two readings are different, it should be read a third time to get the correct value.

At any time, TxVAL contains a valid value to be read, synchronized to PCLK.

Capture Event Function

There are 30 interrupt events that can be captured by the general-purpose timers. These events are divided into two groups of 15 inputs for each of the timers, as shown in Table 18. Any one of the 15 events associated with a general-purpose timer can cause a capture of the 16-bit TxVAL register into the 16-bit TxCAP register. TxCON has a 4-bit field selecting which of the 15 events to capture.

When the selected interrupt event occurs, the TxVAL register is copied into the TxCAP register. TxSTA[1] is set indicating a capture event is pending. This bit is cleared by writing 1 to TxCLRI[1]. The TxCAP register also holds its value and cannot be overwritten until a 1 is written to TxCLRI[1].

Table 18. Timer Capture Event

Event Select Bits (TxCON[])	Timer 0 Capture Source	Timer 1 Capture Source
0	Wake-Up Timer	Timer 0
1	External Interrupt 0	SPI0
2	External Interrupt 1	SPI1
3	External Interrupt 2	I^{2}C slave
4	External Interrupt 3	I^{2}C master
5	External Interrupt 4	Reserved
6	External Interrupt 5	DMA error
7	External Interrupt 6	DMA done, any of the DMA channels
8	External Interrupt 7	External Interrupt 1
9	External Interrupt 8	External Interrupt 2
10	Watchdog timer	External Interrupt 3
11	Reserved	PWMTRIP
12	Timer 1	PWM0
13	Analog to Digital Converter	PWM1
14	Flash controller	PWM2
15	UART Peripheral	PWM3

REGISTER SUMMARY (TIMER 0)

Table 19. Timer 0 Register Summary

Address	Name	Description	Reset	RW
0x40000000	T0LD	16-bit Load Value	0x0000	RW
0x40000004	T0VAL	16-bit Timer Value	0x0000	R
0x40000008	T0CON	Control Register	0x000A	RW
0x4000000C	T0CLRI	Clear Interrupt Register	0x0000	RW
0x40000010	T0CAP	Capture Register	0x0000	R
0x4000001C	T0STA	Status Register	0x0000	R

REGISTER DETAILS (TIMER 0)

16-Bit Load Value Register

Address: 0x40000000, Reset: 0x0000, Name: T0LD

Table 20. Bit Descriptions for T0LD

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Load value.	0x0000	RW

16-Bit Timer Value Register

Address: 0x40000004, Reset: 0x0000, Name: T0VAL

Table 21. Bit Descriptions for T0VAL

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Current counter value.	0x0000	R

Control Register

Address: 0x40000008, Reset: 0x000A, Name: T0CON

Table 22. Bit Descriptions for T0CON

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved. 0 should be written to these bits.	0x0	R
12	EVENTEN	Enable event bit. 0: DIS. Cleared by user. 1: EN. Enable time capture of an event.	0x0	RW
[11:8]	EVENT	Event select bits. Settings not described are reserved and should not be used. 0000: T2. Wakeup timer. 0001: EXT0. External Interrupt 0. 0010: EXT1. External Interrupt 1. 0011: EXT2. External Interrupt 2. 0100: EXT3. External Interrupt 3. 0101: EXT4. External Interrupt 4. 0110: EXT5. External Interrupt 5. 0111: EXT6. External Interrupt 6. 1000: EXT7. External Interrupt 7. 1001: EXT8. External Interrupt 8. 1010: T3. Watchdog timer. 1100: T1. Timer 1. 1101: ADC. Analog to digital converter. 1110: FEE. Flash controller. 1111: COM. UART peripheral.	0x0	RW
7	RLD	Reload control bit for periodic mode. 0: DIS. Reload on a time out. 1: EN. Reload the counter on a write to T0CLRI.	0x0	RW
[6:5]	CLK	Clock select. 00: UCLK. Undivided system clock. 01: PCLK. Peripheral clock. 10: LFOSC. 32 kHz internal oscillator. 11: LFXTAL. 32 kHz external crystal.	0x0	RW
4	ENABLE	Timer enable bit. 0: DIS. Disable the timer. Clearing this bit resets the timer, including the T0VAL register. 1: EN. Enable the timer. The timer starts counting from its initial value, 0 if count-up mode or 0xFFFF if count-down mode.	0x0	RW
3	MOD	Timer mode. 0: FREERUN. Operate in free running mode. 1: PERIODIC. Operate in periodic mode.	0x1	RW
2	UP	Count down/up. 0: DIS. Timer to count down. 1: EN. Timer to count up.	0x0	RW
[1:0]	PRE	Prescaler. 00: DIV1. Source clock/1.If the selected clock source is UCLK or PCLK this setting results in a prescaler of 4. 01: DIV16. Source clock/16. 10: DIV256. Source clock/256. 11: DIV32768. Source clock/32768.	0x2	RW

Clear Interrupt Register

Address: 0x4000000C, Reset: 0x0000, Name: T0CLRI

Table 23. Bit Descriptions for T0CLRI

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved. 0 should be written to these bits.	0x0000	R
1	CAP	Clear captured event interrupt. 1: CLR. Clear a captured event interrupt. This bit always reads 0.	0x0	RW
0	TMOUT	Clear timeout interrupt. 1: CLR. Clear a timeout interrupt. This bit always reads 0.	0x0	RW

Capture Register

Address: 0x40000010, Reset: 0x0000, Name: T0CAP

Table 24. Bit Descriptions for T0CAP

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Capture value.T0CAP holds its value until T0CLRI[1] is set by user code. If the same event occurs again, T0CAP will not be overwritten.	0x0000	R

Status Register

Address: 0x4000001C, Reset: 0x0000, Name: T0STA

Table 25. Bit Descriptions for T0STA

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	R
7	CLRI	T0CLRI write sync in progress. 0: CLR. Cleared when the interrupt is cleared in the timer clock domain. 1: SET. Set automatically when the T0CLRI value is being updated in the timer clock domain, indicating that the timer's configuration is not yet valid.	0x0	R
6	CON	T0CON write sync in progress. 0: CLR. Timer ready to receive commands to T0CON. The previous change of T0CON has been synchronized in the timer clock domain. 1: SET. Timer not ready to receive commands to T0CON. Previous change of the T0CON value has not been synchronized in the timer clock domain.	0x0	R
[5:2]	RESERVED	Reserved.	0x0	R
1	CAP	Capture event pending. 0: CLR. No capture event is pending. 1: SET. Capture event is pending.	0x0	R
0	TMOUT	Time out event occurred. 0: CLR. No timeout event has occurred. 1: SET. Timeout event has occurred. For count-up mode, this is when the counter reaches full scale. For count-down mode, this is when the counter reaches 0.	0x0	R

REGISTER SUMMARY (TIMER 1)

Table 26. Timer 1 Register Summary

Address	Name	Description	Reset	RW
0x40000400	T1LD	16-bit load value.	0x0000	RW
0x40000404	T1VAL	16-bit timer value.	0x0000	R
0x40000408	T1CON	Control Register.	0x000A	RW
0x4000040C	T1CLRI	Clear Interrupt Register.	0x0000	RW
0x40000410	T1CAP	Capture Register.	0x0000	R
0x4000041C	T1STA	Status Register.	0x0000	R

REGISTER DETAILS (TIMER 1)

16-Bit Load Value Register

Address: 0x40000400, Reset: 0x0000, Name: T1LD

Table 27. Bit Descriptions for T1LD

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Load value.	0x0000	RW

16-Bit Timer Value Register

Address: 0x40000404, Reset: 0x0000, Name: T1VAL

Table 28. Bit Descriptions for T1VAL

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Current counter value.	0x0000	R

Control Register

Address: 0x40000408, Reset: 0x000A, Name: T1CON

Table 29. Bit Descriptions for T1CON

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved. 0 should be written to these bits.	0x0	R
12	EVENTEN	Enable event bit. 0: DIS. Cleared by user. 1: EN. Enable time capture of an event.	0x0	RW
[11:8]	EVENT	Event select bits. Settings not described are reserved and should not be used. 0000: T0. Timer 0. 0001: SPI0. SPI0 peripheral. 0010: SPI1. SPI1 peripheral. 0011: I2CS. I^{2}C slave peripheral. 0100: I2CM. I^{2}C master peripheral. 0110: DMAERR. DMA error 0111: DMADONE. Completion of transfer on any of the DMA channel. 1000: EXT1. External Interrupt 1. 1001: EXT2. External Interrupt 2. 1010: EXT3. External Interrupt 3. 1011: PWMTRIP. 1100: PWM0. PWM pair 0. 1101: PWM1. PWM pair 1. 1110: PWM2. PWM pair 2. 1111: PWM3. PWM pair 3.	0x0	RW

Bits	Bit Name	Description	Reset	Access
7	RLD	Reload control bit for periodic mode. 0: DIS. Reload on a time out. 1: EN. Reload the counter on a write to T1CLRI.	0x0	RW
[6:5]	CLK	Clock select. 00: UCLK. Undivided system clock. 01: PCLK. Peripheral clock. 10: LFOSC. 32 kHz internal oscillator. 11: LFXTAL. 32 kHz external crystal.	0x0	RW
4	ENABLE	Timer enable bit. 0: DIS. Disable the timer. Clearing this bit resets the timer, including the T1VAL register. 1: EN. Enable the timer. The timer starts counting from its initial value, 0 if count-up mode or 0xFFFF if count-down mode.	0x0	RW
3	MOD	Timer mode. 0: FREERUN. Operate in free running mode. 1: PERIODIC. Operate in periodic mode.	0x1	RW
2	UP	Count down/up. 0: DIS. Timer to count down. 1: EN. Timer to count up.	0x0	RW
[1:0]	PRE	Prescaler. 00: DIV1. Source clock/1.If the selected clock source is UCLK or PCLK this setting results in a prescaler of 4. 01: DIV16. Source clock/16. 10: DIV256. Source clock/256. 11: DIV32768. Source clock/32768.	0x2	RW

Clear Interrupt Register

Address: 0x4000040C, Reset: 0x0000, Name: T1CLRI

Table 30. Bit Descriptions for T1CLRI

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved. 0 should be written to these bits.	0x0000	R
1	CAP	Clear captured event interrupt. 1: CLR. Clear a captured event interrupt. This bit always reads 0.	0x0	RW
0	TMOUT	Clear timeout interrupt. 1: CLR. Clear a timeout interrupt. This bit always reads 0.	0x0	RW

Capture Register

Address: 0x40000410, Reset: 0x0000, Name: T1CAP

Table 31. Bit Descriptions for T1CAP

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Capture value.T1CAP holds its value until T1CLRI[1] is set by user code. If the same event occurs again, T1CAP will not be overwritten.	0x0000	R

Status Register

Address: 0x4000041C, Reset: 0x0000, Name: T1STA

Table 32. Bit Descriptions for T1STA

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	R
7	CLRI	T1CLRI write sync in progress. 0: CLR. Cleared when the interrupt is cleared in the timer clock domain. 1: SET. Set automatically when the T1CLRI value is being updated in the timer clock domain, indicating that the timer's configuration is not yet valid.	0x0	R
6	CON	T1CON write sync in progress. 0: CLR. Timer ready to receive commands to T1CON. The previous change of T1CON has been synchronized in the timer clock domain. 1: SET. Timer not ready to receive commands to T1CON. Previous change of the T1CON value has not been synchronized in the timer clock domain.	0x0	R
[5:2]	RESERVED	Reserved.	0x0	R
1	CAP	Capture event pending. 0: CLR. No capture event is pending. 1: SET. Capture event is pending.	0x0	R
0	TMOUT	Time out event occurred. 0: CLR. No timeout event has occurred. 1: SET. Timeout event has occurred. For count-up mode, this is when the counter reaches full scale. For count-down mode, this is when the counter reaches 0.	0x0	R

WAKE-UP TIMER

WAKE-UP TIMER (TIMER 2) FEATURES

- 32-bit counter (count-up)
- Four clock sources with programmable prescaler (1, 16, 256, or 32,768)
    - Peripheral clock (PCLK)
    - 32 kHz external crystal (LFXTAL)
    - 32 kHz internal oscillator (LFOSC)
    - External clock applied on P0.5 (ECLKIN)
- Can be used to wake-up the device from hibernate or flexi mode
- Four compare points, one automatic increment

WAKE-UP TIMER BLOCK DIAGRAM

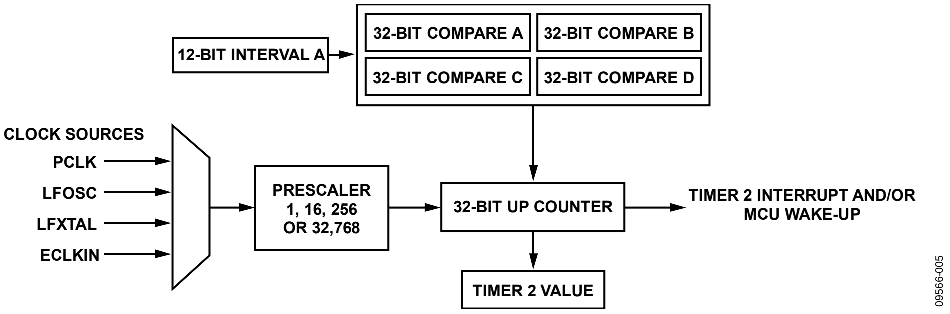


Figure 5. Wake-Up Timer Block Diagram

WAKE-UP TIMER OVERVIEW

The wake-up timer, Timer 2, is a 32-bit wake-up timer (count up) that can be used to wake-up the device from hibernate or flexi modes when clocked by an active clock such as LFOSC or LFXTAL. The timer can also be clocked from PCLK or for debug purposes from an external clock applied on P0.5 (ECLKIN). The clock source can then be scaled down by the prescaler (1, 16, 256 or 32,768).

The timer can be used in free running or periodic mode. In free running mode, the timer counts from 0x00000000 to 0xFFFFFFFF and restarts at 0x00000000. In periodic mode, the timer counts from 0x00000000 to T2WUFD. (T2WUFD is T2WUFD0 combined with T2WUFD1)

In addition, the wake-up timer has four specific time fields to compare with the wake-up counter: T2WUFA, T2WUFB, T2WUFC, and T2WUFD. All four wake-up compare points can generate interrupts or wake-up signals. When in free running mode, T2WUA, T2WUB, T2WUC, andT2WUD must be reconfigured in software to generate a periodic interrupt.

WAKE-UP TIMER OPERATION

The wake-up timer comparator registers must be configured before starting the timer. The timer is started by writing the control enable bit (T2CON[7]). The timer increments until the value reaches full scale in free running mode or when T2WUFD matches the wake-up value, T2VAL.

The wake-up timer is a 32-bit timer. Its current value is stored in two 16-bit registers. T2VAL1 stores the upper 16 bits, and T2VAL0 stores the lower 16 bits.

When T2VAL0 is read, T2VAL1 is frozen at its current value until subsequently read. The control bit FREEZE (T2CON[3]) must be set to freeze the T2VAL register between the lower and upper reads.

Clock Selection

Clock selection is made by setting T2CON[10:9].

If LFXTAL is selected (T2CON[10:9] = 01), it must be ensured that clock circuit is on (XOSCCON[0] = 1).

If PCLK is selected, (T2CON[10:9] = 00) configuring TxCON[1:0] = 00 results in a prescaler of 4.

Synchronization to the LFOSC and LFXTAL clock domain is done automatically by hardware and precautions concerning asynchronous clocks as described in Timer 0, Timer 1, and Timer 3 do not apply.

Compare Field Registers

Hardware Updated Field

T2INC is a 12-bit interval register which is used to update the compare value in T2WUFAx by hardware. When a new value is written in T2INC, bits[16:5] of the internal 32-bit compare register (T2WUFAx) are loaded with the new T2INC value. This 32-bit compare register is automatically incremented with the contents of T2INC (shifted by 5) each time the wake-up counter reaches the value in this compare register, if the new compare is less than the T2WUFD value in periodic mode or 0xFFFFFFFF in free running mode. If the new compare value is greater than these limits, it is recalculated as follows:

In free-running mode, the

$$\textit{New T2WUFA} = \textit{old T2WUFA} + (32 \times \textit{T2INC}) - \text{0xFFFFFFFF}$$

In periodic mode, the

$$\text{New T2WUFA} = \text{old T2WUFA} + (32 \times \text{T2INC}) - \text{T2WUFD}$$

The maximum programmable interval is just above 4 seconds.

The register T2INC contains bits[16:5] of the timer increment and must be shifted left by five bits to get the actual T2WUFA increment. (This is equivalent to the multiplication by 32 in the previous calculation.)

With the default value of 0xC8 (where for calculation purposes 0xC8 = 200 in decimal), a prescaler = 1, and 32 kHz clock selected,

$$\textit{Interval} = ((200 \times 32) + 1) \times 1/32{,}768 = 195.3155 \text{ ms}$$

T2WUFAx registers are read only.

Software Updated Field

T2WUFB, T2WUFC, and T2WUFD are 32-bit values programmed by the user in the T2WUFx0 and T2WUFx1 registers (x = B, C, or D). T2WUFD contains the load value when the wake-up timer is configured in periodic mode.

The T2WUFBx and T2WUFCx registers can be written to at any time, but the corresponding interrupt enable, T2IEN[1] or T2IEN[2], must be disabled. After the register is updated, the interrupt can be reenabled.

In periodic mode, T2WUFDx registers can be only be written to when the timer is disabled.

In free running mode, T2WUFDx registers can be written to while the timer is running.

Before doing so, the corresponding interrupt enable, T2IEN[3], must be disabled. After the register is updated, the interrupt can be reenabled.

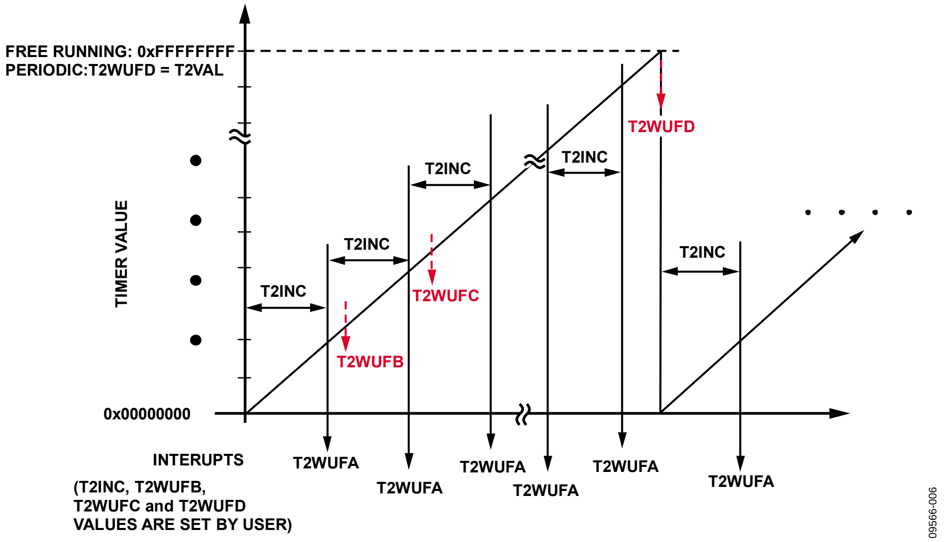


Figure 6. Wake-Up Timer Fields Action

Interrupts/Wake-Up Signals

An interrupt event is generated when the counter value corresponds to any of the compare points or full scale in free running mode. The timer continues counting or is reset to 0.

The wake-up timer generates five maskable interrupts. They are enabled in the T2IEN register. Interrupts can be cleared by setting the corresponding bit in the T2CLRI register.

Note that it takes two cycles of the timer clock source for the interrupt clear to take effect. Ensure that the register write has fully completed before returning from the interrupt handler. Use the data synchronization barrier (DSB) instruction if necessary.

The timer is stopped and reset when clearing the timer enable bit in the T2CON register (T2CON[7]).

REGISTER SUMMARY (WAKE-UP TIMER)

Table 33. Wake-Up Timer Register Summary

Address	Name	Description	Reset	RW
0x40002500	T2VAL0	Current wake-up timer value LSB	0x0000	R
0x40002504	T2VAL1	Current wake-up timer value MSB	0x0000	R
0x40002508	T2CON	Control register	0x0040	RW
0x4000250C	T2INC	12-bit Interval Register for Wake-Up Field A	0x00C8	RW
0x40002510	T2WUFB0	Wake-Up Field B LSB	0x1FFF	RW
0x40002514	T2WUFB1	Wake-Up Field B MSB	0x0000	RW
0x40002518	T2WUFC0	Wake-Up Field C LSB	0x2FFF	RW
0x4000251C	T2WUFC1	Wake-Up Field C MSB	0x0000	RW
0x40002520	T2WUFD0	Wake-UpField D LSB	0x3FFF	RW
0x40002524	T2WUFD1	Wake-Up Field D	0x0000	RW
0x40002528	T2IEN	Interrupt enable	0x0000	RW
0x4000252C	T2STA	Status	0x0000	R
0x40002530	T2CLRI	Clear interrupts	0x0000	W
0x4000253C	T2WUFA0	Wake-Up Field A LSB	0x1900	R
0x40002540	T2WUFA1	Wake-Up Field A MSB	0x0000	R

REGISTER DETAILS (WAKE-UP TIMER)

Current Wake-Up Timer Value LSB Register

Address: 0x40002500, Reset: 0x0000, Name: T2VAL0

Table 34. Bit Descriptions for T2VAL0

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Current wake-up timer value (bits 15 to 0).	0x0000	R

Current Wake-Up Timer Value MSB Register

Address: 0x40002504, Reset: 0x0000, Name: T2VAL1

Table 35. Bit Descriptions for T2VAL1

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Current wake-up timer value (bits 31 to 16).	0x0000	R

Control Register

Address: 0x40002508, Reset: 0x0040, Name: T2CON

Table 36. Bit Descriptions for T2CON

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved. 0 should be written to this bit.	0x0	R
[10:9]	CLK	Clock select. Notes: If Clock prescaler select = 00 (T2CON[1:0] = 00) and if the selected clock source is PCLK (T2CON[10:9] = 00), then this setting results in a prescaler of 4. 00: PCLK. Peripheral clock. 01: LFXTAL. 32 kHz external crystal. 10: LFOSC. 32 kHz internal oscillator. 11: EXTCLK. External clock applied on P0.5.	0x0	RW
8	WUEN	Wake-up enable bits for time field values. 0: DIS. Disable asynchronous wake-up timer. Interrupt conditions will not wake-up the part from sleep mode. 1: EN. Enable asynchronous wake-up timer even when the core clock is off. Once the timer value equals any of the interrupt enabled compare field, a wake-up signal is generated.	0x0	RW
7	ENABLE	Timer enable bit. Notes: This bit must be low when changing any of the control information. 0: DIS. Disable the timer. 1: EN. Enable the timer. When enabled wait for T2STA[8] to clear before continuing.	0x0	RW
6	MOD	Timer free run enable. 0: PERIODIC. Operate in periodic mode. Counts up to the value in T2WUFD 1: FREERUN. Operate in free running mode (default). Counts from 0 to FFFF FFFF and starts again at 0.	0x1	RW
[5:4]	RESERVED	Reserved.	0x0	R
3	FREEZE	Freeze enable bit. 0: DIS. Disable this feature. 1: EN. Enable the freeze of the high 16 bits after the lower bits have been read from T2VAL0. This ensures that the software reads an atomic shot of the timer. The entire T2VAL register unfreezes after the high bits (T2VAL1) have been read.	0x0	RW
2	RESERVED	Reserved. 0 should be written to this bit.	0x0	R

Bits	Bit Name	Description	Reset	Access
[1:0]	PRE	Prescaler. 00: DIV1. Source clock/1. If the selected clock source is PCLK this setting results in a prescaler of 4. 01: DIV16. Source clock/16. 10: DIV256. Source clock/256. 11: DIV32768. Source clock/32768.	0x0	RW

12-Bit Interval Register for Wake-Up Field A

Address: 0x4000250C, Reset: 0x00C8, Name: T2INC

Table 37. Bit Descriptions for T2INC

Bits	Bit Name	Description	Reset	Access
[15:12]	RESERVED	Reserved. 0 should be written to this bit.	0x0	R
[11:0]	VALUE	Wake-up interval.	0x0C8	RW

Wake-Up Field B LSB Register

Address: 0x40002510, Reset: 0x1FFF, Name: T2WUFB0

Table 38. Bit Descriptions for T2WUFB0

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Lower 16 bits of Wake-Up Field B.	0x1FFF	RW

Wake-Up Field B MSB Register

Address: 0x40002514, Reset: 0x0000, Name: T2WUFB1

Table 39. Bit Descriptions for T2WUFB1

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Upper 16 bits of Wake-Up Field B.	0x0000	RW

Wake-Up Field C LSB Register

Address: 0x40002518, Reset: 0x2FFF, Name: T2WUFC0

Table 40. Bit Descriptions for T2WUFC0

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Lower 16 bits of Wake-Up Field C.	0x2FFF	RW

Wake-Up Field C MSB Register

Address: 0x4000251C, Reset: 0x0000, Name: T2WUFC1

Table 41. Bit Descriptions for T2WUFC1

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Upper 16 bits of Wake-Up Field C.	0x0000	RW

Wake-Up Field D LSB Register

Address: 0x40002520, Reset: 0x3FFF, Name: T2WUFD0

Table 42. Bit Descriptions for T2WUFD0

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Lower 16 bits of Wake-Up Field D.	0x3FFF	RW

Wake-Up Field D MSB Register

Address: 0x40002524, Reset: 0x0000, Name: T2WUFD1

Table 43. Bit Descriptions for T2WUFD1

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Upper 16 bits of Wake-Up Field D.	0x0000	RW

Interrupt Enable Register

Address: 0x40002528, Reset: 0x0000, Name: T2IEN

Table 44. Bit Descriptions for T2IEN

Bits	Bit Name	Description	Reset	Access
[15:5]	RESERVED	Reserved. 0 should be written to this bit.	0x000	R
4	ROLL	Interrupt enable bit when the counter rolls over. Only occurs in free running mode. 0: DIS. Disable the roll over interrupt. 1: EN. Generate an interrupt when Timer2 rolls over.	0x0	RW
3	WUFD	T2WUFD interrupt enable. 0: DIS. Disable T2WUFD interrupt. 1: EN. Generate an interrupt when T2VAL reaches T2WUFD.	0x0	RW
2	WUFC	T2WUFC interrupt enable. 0: DIS. Disable T2WUFC interrupt. 1: EN. Generate an interrupt when T2VAL reaches T2WUFC.	0x0	RW
1	WUFB	T2WUFB interrupt enable. 0: DIS. Disable T2WUFB interrupt. 1: EN. Generate an interrupt when T2VAL reaches T2WUFB.	0x0	RW
0	WUFA	T2WUFA interrupt enable. 0: DIS. Disable T2WUFA interrupt. 1: EN. Generate an interrupt when T2VAL reaches T2WUFA.	0x0	RW

Status Register

Address: 0x4000252C, Reset: 0x0000, Name: T2STA

Table 45. Bit Descriptions for T2STA

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x00	R
8	CON	Indicates when a change in the enable bit is synchronized to the 32 kHz clock domain (done automatically). 0: CLR. It returns low when the change in the enable bit has been synchronized to the 32 kHz clock domain. 1: SET. This bit is set high when the Enable bit (Bit 5) in the control register is set or cleared and it is not synchronized to the 32 kHz clock.	0x0	R
7	FREEZE	Status of T2VAL freeze. 0: CLR. Reset low when T2VAL1 is read, indicating T2VAL is unfrozen. 1: SET. Set high when the T2VAL0 is read, indicating T2VAL is frozen.	0x0	R
[6:5]	RESERVED	Reserved.	0x0	R
4	ROLL	Interrupt status bit for instances when counter rolls over. Only occurs in free running mode. This interrupt does not wake up the device in hibernate mode. 0: CLR. Indicate that the timer has not rolled over. 1: SET. Set high when enabled in the interrupt enable register and the T2VALS counter register is equal to all 1s.	0x0	R

Bits	Bit Name	Description	Reset	Access
3	WUFD	T2WUFD interrupt flag. 0: CLR. Cleared after a write to the corresponding bit in T2CLRI. 1: SET. Indicates that a comparator interrupt has occurred.	0x0	R
2	WUFC	T2WUFC interrupt flag. 0: CLR. Cleared after a write to the corresponding bit in T2CLRI. 1: SET. Indicates that a comparator interrupt has occurred.	0x0	R
1	WUFB	T2WUFB interrupt flag. 0: CLR. Cleared after a write to the corresponding bit in T2CLRI. 1: SET. Indicates that a comparator interrupt has occurred.	0x0	R
0	WUFA	T2WUFA interrupt flag. 0: CLR. Cleared after a write to the corresponding bit in T2CLRI. 1: SET. Indicates that a comparator interrupt has occurred.	0x0	R

Clear Interrupts Register

Address: 0x40002530, Reset: 0x0000, Name: T2CLRI

Table 46. Bit Descriptions for T2CLRI

Bits	Bit Name	Description	Reset	Access
[15:5]	RESERVED	Reserved. 0 should be written to this bit.	0x000	R
4	ROLL	Clear interrupt on Rollover. Only occurs in free running mode. 1: CLR. Interrupt clear bit for when counter rolls over.	0x0	W
3	WUFD	T2WUFD interrupt flag. Cleared automatically after synchronization. 1: CLR. Clear the T2WUFD interrupt flag.	0x0	W
2	WUFC	T2WUFC interrupt flag. Cleared automatically after synchronization. 1: CLR. Clear the T2WUFC interrupt flag.	0x0	W
1	WUFB	T2WUFB interrupt flag. Cleared automatically after synchronization. 1: CLR. Clear the T2WUFB interrupt flag.	0x0	W
0	WUFA	T2WUFA interrupt flag. Cleared automatically after synchronization. 1: CLR. Clear the T2WUFA interrupt flag.	0x0	W

Wake-Up Field A LSB Register

Address: 0x4000253C, Reset: 0x1900, Name: T2WUFA0

Table 47. Bit Descriptions for T2WUFA0

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Lower 16 bits of Compare Register A.	0x1900	R

Wake-Up Field A MSB Register

Address: 0x40002540, Reset: 0x0000, Name: T2WUFA1

Table 48. Bit Descriptions for T2WUFA1

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Upper 16 bits of Compare Register A.	0x0000	R

WATCHDOG TIMER

WATCHDOG TIMER (TIMER 3) FEATURES

- 16-bit countdown timer, which can be used to recover from an illegal software state.
- Clocked by the 32 kHz internal oscillator (LFOSC) with a programmable prescaler (1, 16, 256, or 4096).

WATCHDOG TIMER BLOCK DIAGRAM

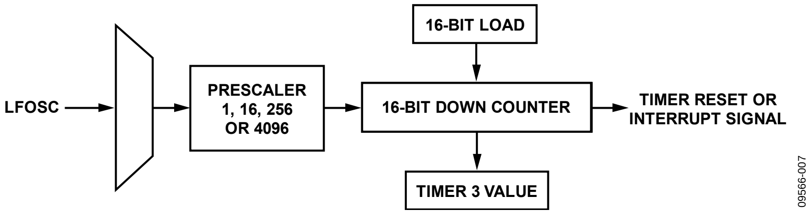


Figure 7. Watchdog Timer Block Diagram

WATCHDOG TIMER OVERVIEW

The watchdog timer, Timer 3, is used to recover from an illegal software state. When enabled, it requires periodic servicing to prevent it from forcing a reset of the device. For debug purposes, the timer can be configured to generate an interrupt instead of a reset.

The watchdog timer is clocked by the internal 32.768 kHz oscillator, LFOSC. It is clocked at all times except during reset and shutdown mode.

The watchdog timer is a 16-bit count-down timer with a programmable prescaler. The prescaler is selectable and can divide LFOSC by factors of 1, 16, 256, or 4096.

WATCHDOG TIMER OPERATION

After a reset or after waking up from SHUTDOWN, the watchdog timer is initialized in hardware as follows:

```
T3CON = 0x00E9
T3LD = 0x1000
T3VAL = 0x1000
```

This enables the watchdog timer with a timeout of 32 seconds.

User code should disable the watchdog timer at the start of user code when debugging or if the watchdog timer is not required.

```
T3CON = 0x00;                                   // Disable watchdog timer
```

A write to T3CON with the timer enable bit set (T3CON[5] = 1) locks the timer configuration after which no further modification is possible. The LOCK bit in the T3STA register (T3STA[4]) indicates if the timer configuration has been locked by a previous write to T3CON.

Programmable Timeout

T3LD is used as the timeout. Once enabled, Timer3 decreases from the value present in the T3LD register to 0. The maximum timeout is ~8192 s (T3LD = 0xFFFF, prescale = 4096).

Refreshing the Timer

When the watchdog timer decrements to 0, a reset (or interrupt signal) is generated. This event can be prevented by writing T3CLRI with 0xCCCC before the timer reaches 0.

A write of 0xCCCC to T3CLRI causes the watchdog timer to reload with the T3LD immediately to begin a new timeout period and start to count again. If any value other than 0xCCCC is written, a reset is generated (or interrupt if selected by T3CON[1]).

User code must ensure that the T3CLRI register write has fully completed before returning from the interrupt handler. Use the data synchronization barrier (DSB) instruction if necessary. Note that the user needs to wait for T3STA[1] bit to be cleared to ensure resetting the timeout period.

Asynchronous Clock Source

The watchdog timer is clocked by a 32 kHz oscillator. Note that 3 bits are provided in the T3STA register to ensure the correct synchronization of the timer clock and core clock domains. After a write to T3CON, T3LD, or T3CLRI, user code should wait until the corresponding status bit is cleared.

The T3VAL register always contains the timer value synchronized to the core clock domain.

REGISTER SUMMARY (WATCHDOG TIMER)

Table 49. Watchdog Timer Register Summary

Address	Name	Description	Reset	RW
0x40002580	T3LD	16-bit load value.	0x1000	RW
0x40002584	T3VAL	16-bit timer value.	0x1000	R
0x40002588	T3CON	Control register.	0x00E9	RW
0x4000258C	T3CLRI	Clear interrupt register.	0x0000	W
0x40002598	T3STA	Status register.	0x0020	R

REGISTER DETAILS (WATCHDOG TIMER)

16-Bit Load Value Register

Address: 0x40002580, Reset: 0x1000, Name: T3LD

Table 50. Bit Descriptions for T3LD

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Load value.	0x1000	RW

16-bit Timer Value Register

Address: 0x40002584, Reset: 0x1000, Name: T3VAL

Table 51. Bit Descriptions for T3VAL

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Current counter value.	0x1000	R

Control Register

Address: 0x40002588, Reset: 0x00E9, Name: T3CON

Table 52. Bit Descriptions for T3CON

Bits	Bit Name	Description	Reset	Access
[15:7]	RESERVED	Reserved. Note that 0 should be written to these bits.	0x001	RW
6	MOD	Timer Mode. 0: Reserved. 1: PERIODIC: Operate in periodic mode.	0x1	RW
5	ENABLE	Timer enable bit. 0: DIS. Disable the timer. Clearing this bit resets the timer, including the T0VAL register. 1: EN. Enable the timer. The timer starts counting from its initial value.	0x1	RW
4	RESERVED	Reserved. Note that 0 should be written to this bit.	0x0	RW
[3:2]	PRE	Prescaler. 00: DIV1. Source clock/1. 01: DIV16. Source clock/16. 10: DIV256. Source clock/256. 11: DIV4096. Source clock/4096.	0x2	RW
1	IRQ	Timer interrupt. 0: DIS. Generate a reset on a timeout. 1: EN. Generate an interrupt when the timer times out. This feature is available in active mode only and can be used to debug the watchdog timeout events.	0x0	RW
0	PD	Stop count in hibernate mode. 0: DIS. The timer continues to count when in hibernate mode. 1: EN. The timer stops counting when in hibernate mode.	0x1	RW

Clear Interrupt Register

Address: 0x4000258C, Reset: 0x0000, Name: T3CLRI

Table 53. Bit Descriptions for T3CLRI

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Clear watchdog. 0xCCCC should be written to this register to reload the watchdog timer. A write of any other value causes the timer to generate a reset or interrupt.	0x0000	W

Status Register

Address: 0x40002598, Reset: 0x0020, Name: T3STA

Table 54. Bit Descriptions for T3STA

Bits	Bit Name	Description	Reset	Access
[15:5]	RESERVED	Reserved.	0x001	R
4	LOCK	Lock status bit. 0: CLR. Cleared after any reset and until user code sets T3CON[5]. 1: SET. Set automatically in hardware when user code sets T3CON[5].	0x0	R
3	CON	T3CON write sync in progress. 0: CLR. Timer ready to receive commands to T3CON. The previous change of T3CON has been synchronized in the timer clock domain. 1: SET. Timer not ready to receive commands to T3CON. Previous change of the T3CON value has not been synchronized in the timer clock domain.	0x0	R
2	LD	T3LD write sync in progress. 0: CLR. The previous change of T3LD has been synchronized in the timer clock domain. 1: SET. Previous change of the T3LD value has not been synchronized in the timer clock domain.	0x0	R
1	CLRI	T3CLRI write sync in progress. 0: CLR. Cleared when the interrupt is cleared in the timer clock domain. 1: SET. Set automatically when the T3CLRI value is being updated in the timer clock domain, indicating that the timer's configuration is not yet valid.	0x0	R
0	IRQ	Interrupt pending. 0: CLR. No timeout event has occurred. 1: SET. A timeout event has occurred.	0x0	R

ADC CIRCUIT

ADC CIRCUIT FEATURES

- 12-bit SAR ADC.
- 6 external single-ended or differential inputs and 5 internal signals multiplexer.
- 1.25 V on-chip reference and 1.8 V internal LDO option for ratiometric measurements.

ADC BLOCK DIAGRAM

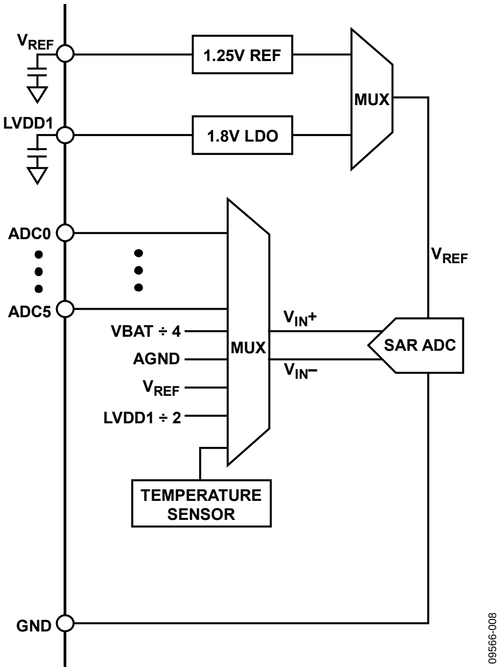


Figure 8. ADC Block Diagram

ADC CIRCUIT OVERVIEW

The analog-to-digital converter (ADC) peripheral provides the user with a multichannel multiplexer, a 1.25 V on-chip reference, and a 12-bit ADC. The ADC is based on a successive approximation register (SAR) ADC architecture with track and hold structure described in the Analog Input Structure section.

In single-ended mode, the converter accepts analog input range of 0 V to V_{REF}, the ADC reference voltage selected in the ADCCFG register.

In differential mode, the input signal must be balanced around a common-mode voltage (V_{CM}) in the $\pm V_{REF} \div 2$ range with a maximum amplitude of V_{REF}.

The absolute maximum rating on the ADC input is 2.1 V.

Single or continuous conversion can be initiated in the software. Timer0 or Timer1 overflow can also be used to trigger an ADC conversion.

Measurements on the external sensors can be ratiometric, as shown in Figure 18. The multiplexer allows selection of the following:

- Up to six external single ended ADC channels or 3 pairs of differential inputs
- Power supply monitoring
- Internal temperature sensor
- Ground and ADC reference, V_{REF}, for calibration purposes

TRANSFER FUNCTION

Single-Ended Mode

The output coding in single-ended mode is straight binary. The LSB weight in single-ended mode is LSB = $V_{REF}/4096$. Figure 9 shows the ideal input/output transfer characteristic in single-ended mode.

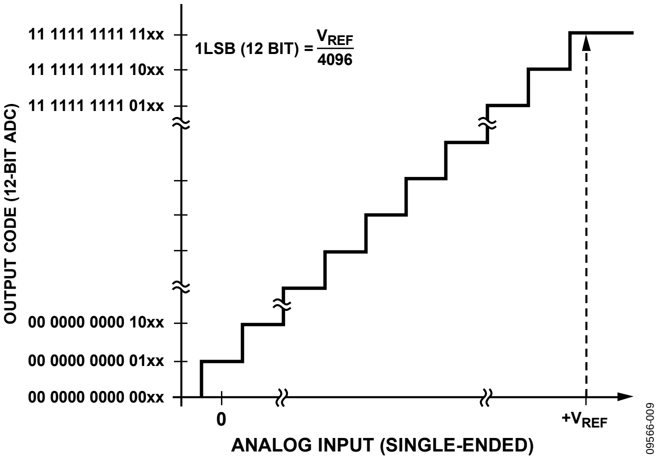


Figure 9. Transfer Function in Single-Ended Mode

Differential Mode

The amplitude of the differential signal is the difference between the signals applied to the V_{IN+} and V_{IN-} input voltage. The maximum amplitude of the differential signal is, therefore, $-V_{REF}$ to $+V_{REF}$ p-p (that is, 2 × V_{REF}). This is regardless of the common mode (CM). The common mode is the average of the two signals, for example, $(V_{IN+} + V_{IN-})/2$, and is, therefore, the voltage that the two inputs are centered on. This results in the span of each input being CM $\pm V_{REF}/2$. This voltage has to be set up externally, and its range varies with V_{REF} (see Driving the Analog Inputs section).

The output coding in differential mode is offset binary with 1 LSB = 2 V_{REF}/4096. The designed code transitions occur midway between successive integer LSB values (that is, 1/2 LSB, 3/2 LSB, 5/2 LSB, … , FS – 3/2 LSB). The ideal input/output transfer characteristic is shown in Figure 10.

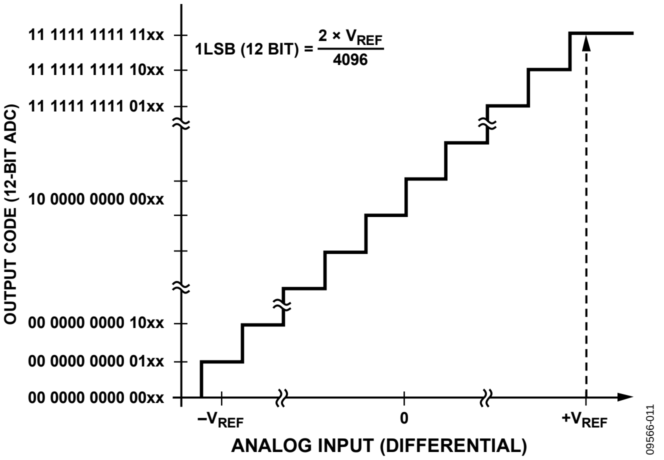


Figure 10. Transfer Function in Differential Mode

CONVERTER OPERATION

The ADC incorporates a successive approximation register(SAR) architecture involving a charge-sampled input stage. This architecture can operate in differential or single-ended mode.

Differential Mode

The successive approximation register architecture is based on two capacitive DACs. Figure 11 and Figure 12 show simplified schematics of the ADC in acquisition and conversion phase, respectively. The ADC comprises control logic, a SAR, and two capacitive DACs. In Figure 11 (the acquisition phase), SW3 is closed and SW1 and SW2 are in Position A. The comparator is held in a balanced condition, and the sampling capacitor arrays acquire the differential signal on the input.

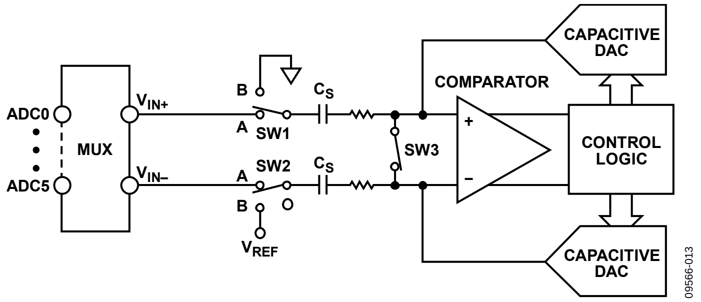


Figure 11. ADC Acquisition Phase

When the ADC starts a conversion, as shown in Figure 12, SW3 opens, and then SW1 and SW2 move to Position B. This causes the comparator to become unbalanced. Both inputs are disconnected once the conversion begins. The control logic and the charge redistribution DACs are used to add and subtract fixed amounts of charge from the sampling capacitor arrays to bring the comparator back into a balanced condition. When the comparator is rebalanced, the conversion is complete. The control logic generates the ADC output code. The output impedances of the sources driving the V_{IN+} and V_{IN-} input voltage pins must be matched; otherwise, the two inputs have different settling times, resulting in errors.

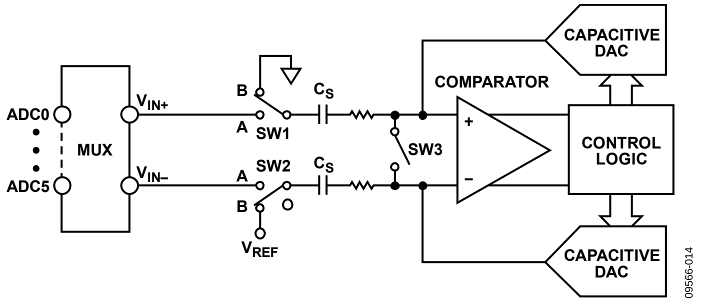


Figure 12. ADC Conversion Phase

Single-Ended Mode

In single-ended mode, the negative input is always connected internally to ground. The V_{IN-} pin can be floating. The input signal range on V_{IN+} is 0 V to V_{REF}.

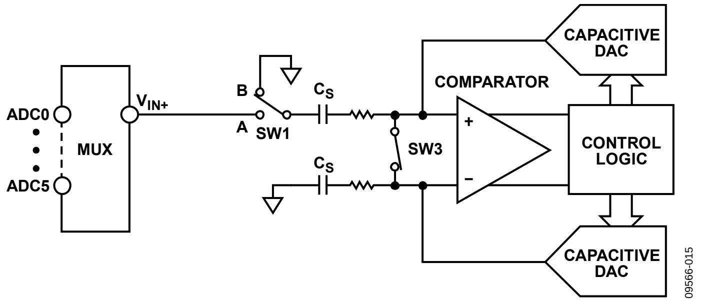


Figure 13. ADC in Single-Ended Mode

Analog Input Structure

Figure 14 shows the equivalent circuit of the analog input structure of the ADC. The four diodes provide ESD protection for the analog inputs. Care must be taken to ensure that the analog input signals never exceed the supply rails by more than 300 mV.

The C1 capacitors in Figure 14 are typically 4 pF and can be primarily attributed to pin capacitance. The resistors are lumped components made up of the on resistance of the switches. The value of these resistors is typically about 750 Ω. The C2 capacitors are the ADC sampling capacitors and typically have a capacitance of 16 pF.

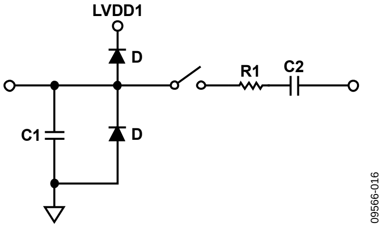


Figure 14. Equivalent Analog Input Circuit Conversion Phase: Switches Open, Track Phase: Switches Closed

ADC Timing

Figure 15 details the ADC conversion phases and timing.

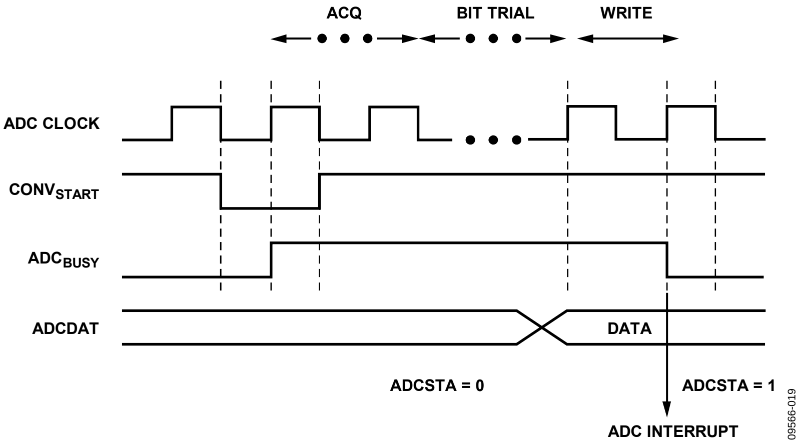


Figure 15. ADC Timing Internal Signals

The number of clocks during the acquisition phase and the ADC clock speed are programmable. By default, the acquisition time is eight ADC clocks and the ADC clock is 4 MHz. The number of extra clocks, for bit trial and register write is set to 18, which gives a default sampling frequency of 154 kHz. The number of acquisition clocks can be extended to 16 when converting a high impedance source, for example, VBAT/4 or LVDD/2 channels. An external capacitor can also be added on the ADCx pin to help the charge of the internal sampling capacitor C2 (see Figure 14).

The structure of the ADC and multiplexer allows one to switch channels while the previous ADC conversion is in the bit trial phase.

DRIVING THE ANALOG INPUTS

For ac applications, removing high frequency components from the analog input signal is recommended by using an RC low-pass filter on the relevant analog input pins. In applications where harmonic distortion and signal-to-noise ratio are critical, the analog input should be driven from a low impedance source. Large source impedances significantly affect the ac performance of the ADC. This can necessitate the use of an input buffer amplifier. The choice of the op amp is a function of the particular application. Figure 16 and Figure 17 give an example of an ADC front end.

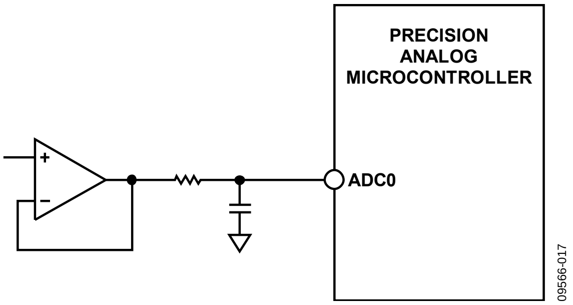


Figure 16. Buffering Single-Ended Input

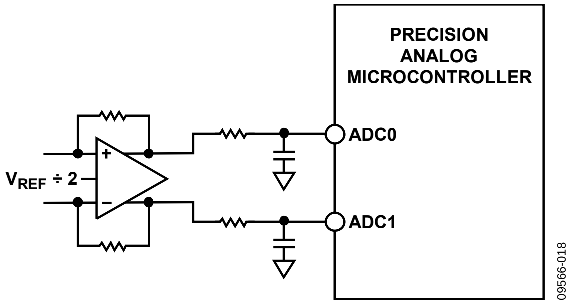


Figure 17. Buffering Differential Inputs

ADC REFERENCE SELECTION

The ADC input range is limited by the selected ADC reference. The ADC performance depends on the reference selected.

Three ADC reference options are available:

- high precision low drift, internal 1.25 V reference
- internal 1.8 V LDO
- external reference on the VREF pin. The external reference voltage is limited to 1.8 V maximum.

Internal Reference

The internal 1.25 V reference is the default and recommended option for most ADC measurements.

However, the internal 1.25 V reference is not intended to drive any external circuitry. An external 0.47 μF capacitor should be connected between the VREF pin and ground.

Internal LDO

The internal LDO can be used for ratio metric measurements, providing supply to the sensor as well as reference to the ADC. In this configuration, the sensor should be connected to P3.4 instead of ground to allow disconnecting the sensor and to reduce current consumption after ADC conversions are complete as shown in Figure 18.

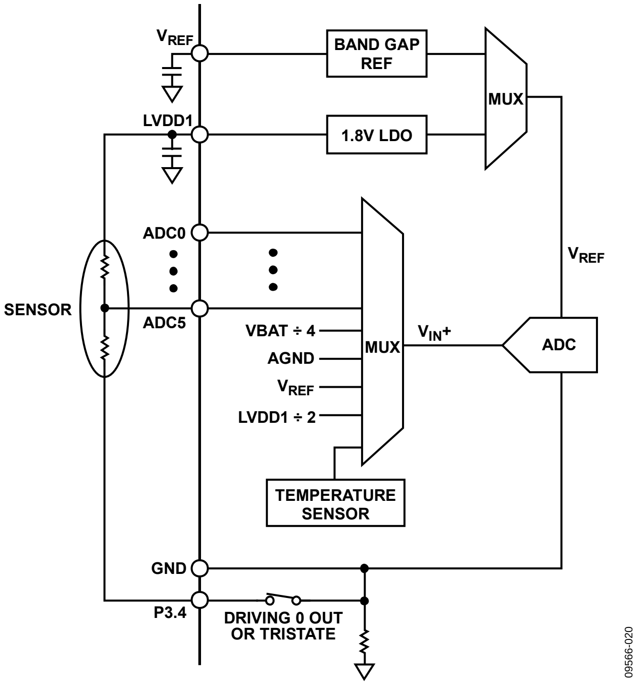


Figure 18. Ratiometric Measurement Typical Connection Diagram

The internal LDO can also be used for converting signals between 1.25 V and 1.8 V. The LVDD internal channel selection accommodates measuring the accuracy of the LDO output vs. the reference. This measurement can be used for software compensations when using the internal LDO as a reference.

External Reference

An external reference can be connected to the VREF pin for converting signals between 1.25 V and 1.8 V. The maximum voltage allowed on the VREF pin is 1.8 V and the internal reference must be turned off.

INTERNAL SIGNALS

Temperature Sensor

An internal temperature sensor is provided on chip to indicate the die temperature. The temperature sensor is not calibrated. Its accuracy is approximately ±10°C. The temperature sensor buffer turns on/off automatically when selecting the temperature sensor channel.

To calculate the temperature from the ADC reading, use the following formula:

$$\textit{Temperature (in }{}^{0}\textit{C)} = 0\text{x}878 \times \textit{ADC reading in mV} - 358$$

This formula is derived from the temperature sensor voltage output at 25°C and voltage TC specifications in the data sheet.

VBAT and LVDD Attenuators

The ADC can measure two attenuated supplies, VBAT and LVDD1. The attenuators are powered up/down automatically when selecting the corresponding channel in the ADCCFG MMR. The resistor size for these attenuators is of the order of 384 kΩ and 128 kΩ for VBAT and 256 kΩ for LVDD1. Due to the high impedance of these inputs, the ADC acquisition clocks should be set to the maximum.

ADC OPERATION

ADC Initialization

To ensure the validity of the first ADC sample, the following steps must be performed:

1. Enable the ADC reference by enabling the ADC (ADCCON = 0x80) or select the temperature sensor in the configuration register (this also turns on the reference).
2. Wait ~2.55 ms.
3. Enable the reference buffer by writing ADCCON =0x00.
4. Wait ~1.95 ms.
5. Start ADC conversions in ADCCON.

Once configured via the ADC configuration and control registers, the ADC converts the analog input and provides a 12-bit result in the ADC data register. The status bit is set when the result is available.

ADC Interrupt Generation

The status bit is set when new data is available and it stays set in between concurrent new data if the data register is not read. Only a read of the data register, a power down of the ADC or a reset of the device, clears the status bit.

If enabled in the NVIC and in the ADC control register, an interrupt occurs each time there is new data, regardless of whether the status bit is set or not.

DMA Request

The DMA operation from the ADC is setup by configuring the DMA peripheral, enabling the DMA ADC interrupt and setting ADCCON[6] = 1(DMA enable) and ADCCON[0] = 1 (ADC conversion start). The ADC conversion modes can be continuous, from Timer0/1 or from P0.3 configured as ADCCONVST in the Digital Port Multiplex section.

ADC Calibration

By default, the factory-set values written to the ADC offset (ADCOF) and gain coefficient registers (ADCGN) yield optimum performance in terms of end-point errors and linearity for standalone operation of the part.

The offset and gain registers are not automatically applied to the ADC conversion result in the ADCDAT register and should be applied by user code to obtain minimum end point errors. The offset should be compensated before the gain calibration as demonstrated in code example provided as part of the development system.

Turning Off the ADC

Ensure the channel selected is not the temperature sensor before powering down the ADC.

REGISTER SUMMARY (ANALOG-TO-DIGITAL CONVERTER)

Table 55. Analog-to-Digital Converter Register Summary

Address	Name	Description	Reset	RW
0x40050000	ADCCFG	ADC configuration register.	0x0A00	RW
0x40050004	ADCCON	ADC control register.	0x90	RW
0x40050008	ADCSTA	ADC status register.	0x00	R
0x4005000C	ADCDAT	ADC data register.	0x0000	R
0x40050010	ADCGN	ADC gain register.	0x0000	RW
0x40050014	ADCOF	ADC offset register.	0x0000	RW

REGISTER DETAILS (ANALOG-TO-DIGITAL CONVERTER)

ADC Configuration Register

Address: 0x40050000, Reset: 0x0A00, Name: ADCCFG

Table 56. Bit Descriptions for ADCCFG

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	Reserved. 0 should be written to these bits.	0x0	R
13	REF	Reference select. 0: INTERNAL125V. Select the internal 1.25 V reference as the ADC reference. 1: LVDD. Select the 1.8 V regulator output (LVDD1) as the ADC reference.	0x0	RW
[12:10]	CLK	ADC clock frequency. 000: Reserved. 001: Reserved. 010: FCOREDIV4. 011: FCOREDIV8. 100: FCOREDIV16. 101: FCOREDIV32.	0x2	RW
[9:8]	ACQ	Acquisition clocks. 00: 2. 01: 4. 10: 8. 11: 16.	0x2	RW
[7:4]	RESERVED	0 should be written to these bits.	0x0	R
[3:0]	CHSEL	Channel select. 0000: ADC0. Single ended ADC0 input. 0001: ADC1. Single ended ADC1 input. 0010: ADC2. Single ended ADC2 input. 0011: ADC3. Single ended ADC3 input. 0100: ADC4. Single ended ADC4 input. 0101: ADC5. Single ended ADC5 input. 0110: DIFF0. Differential ADC0 - ADC1 inputs. 0111: DIFF1. Differential ADC2 - ADC3 inputs. 1000: DIFF2. Differential ADC4 - ADC5 inputs. 1001: TEMP. Internal temperature sensor. 1010: VBATDIV4. Internal supply divided by 4. 1011: LVDDDIV2. Internal 1.8V regulator output (LVDD1) divided by 2. 1100: VREF. Internal ADC reference input for gain calibration. 1101: AGND. Internal ADC ground input for offset calibration.	0x0	RW

ADC Control Register

Address: 0x40050004, Reset: 0x90, Name: ADCCON

Table 57. Bit Descriptions for ADCCON

Bits	Bit Name	Description	Reset	Access
7	REFBUF	Reference buffer enable bit. 0: EN. Turn on the reference buffer. The reference buffer takes 5 ms to settle and consumes approximately 210 µA. 1: DIS. Turn off the reference buffer. The internal reference buffer must be turned off if using an external reference.	0x1	RW
6	DMA	DMA transfer enable bit. 0: DIS. Disable DMA transfer. 1: EN. Enable DMA transfer.	0x0	RW
5	IEN	Interrupt enable. 0: DIS. Disable the ADC interrupt. 1: EN. Enable the ADC interrupt. An interrupt is generated when new data is available.	0x0	RW
4	ENABLE	ADC enable. 0: EN. Enable the ADC. 1: DIS. Disable the ADC.	0x1	RW
[3:1]	MOD	Conversion mode. 000: SOFT. Software trigger, used in conjunction with the START bit. 001: CONT. Continuous convert mode. 011: T0OVF. Timer0 overflow. 100: T1OVF. Timer1 overflow. 101: GPIO. ADC conversion triggered by P0.3 input.	0x0	RW
0	START	ADC conversion start. 0: DIS. Has no effect. 1: EN. Start conversion when SOFT conversion mode is selected. This bit does not clear after a single software conversion.	0x0	RW

ADC Status Register

Address: 0x40050008, Reset: 0x00, Name: ADCSTA

Table 58. Bit Descriptions for ADCSTA

Bits	Bit Name	Description	Reset	Access
[7:1]	RESERVED	Reserved.	0x00	R
0	READY	ADC ready bit. 0: CLR. Cleared automatically when ADCDAT is read. 1: EN. Set by the ADC when a conversion is complete. This bit generates an interrupt if enabled (IEN set in ADCCON).	0x0	R

ADC Data Register

Address: 0x4005000C, Reset: 0x0000, Name: ADCDAT

Table 59. Bit Descriptions for ADCDAT

Bits	Bit Name	Description	Reset	Access
[15:14]	RESERVED	Reserved.	0x0	R
[13:2]	VALUE	ADC result.	0x000	R
[1:0]	VALUE_RESERVED	Reserved.	0x0	R

ADC Gain Register

Address: 0x40050010, Reset: 0x0000, Name: ADCGN

Table 60. Bit Descriptions for ADCGN

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Gain.	0x0000	RW

ADC Offset Register

Address: 0x40050014, Reset: 0x0000, Name: ADCOF

Table 61. Bit Descriptions for ADCOF

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Offset.	0x0000	RW

RF TRANSCEIVER

RF TRANSCEIVER FEATURES

- Ultralow power, high performance transceiver
- Frequency bands:
  - 431 MHz to 464 MHz
  - 862 MHz to 928 MHz
- Optimized transceiver configurations available
  - 1 kbps data rate with 10 kHz frequency deviation
  - 38.4 kbps data rate with 20 kHz frequency deviation
  - 300 kbps data rate with 75 kHz frequency deviation
- Single-ended and differential power amplifiers (PAs)
- Low system current consumption:
- Selectable RF output power using single-ended PA and differential PA
- Patented fast settling automatic frequency control (AFC)
- Digital received signal strength indication (RSSI)
- Packet management
- Insertion/detection of preamble/sync word/CRC/address
- Manchester data encoding and decoding
- Data whitening
- Optimized image rejection calibration routine
- 240-byte packet buffer for Tx/Rx data

RF TRANSCEIVER OPERATION

This section of the user guide is a short primer on how the RF transceiver operates. The goal of this section is to explain in simple terms what it takes to get the transceiver transmitting and receiving data. The major topics of modulation, packet structure, transceiver states, example code and other important considerations are discussed. For advanced users who want to use more of the flexibility that the transceiver has available, refer to the RF Transceiver Advanced Features section.

The RF transceiver allows the transmission and reception of data packets by the Cortex-M3 processor.

Data Transmission

Modulation

The first step to transmitting data is to modulate the binary data onto an RF signal. The primary modulation type supported by the transceiver is 2FSK (two level Frequency Shift Keying). The transceiver can also supports the Gaussian frequency shift keying (GFSK) modulation schemes.

The FSK types of modulation of an RF signal involve the movement of an RF signal about a carrier frequency. The RF signal will deviate around this carrier frequency to signify a 0 or 1. The offset from the center frequency is known as the frequency deviation. The rate at which it deviates around this carrier frequency is known as the data rate.

To start using a particular modulation scheme, a frequency deviation and data rate will have to be chosen. In the case of 2FSK modulation, the frequency deviation is the frequency shift above and below the carrier frequency that will represent the 0's and 1's of the binary data to be transmitted. Figure 19 shows an idealized 2FSK transmit spectrum with the frequency deviation marked. The distance between the two peaks is the frequency deviation × 2 with the 0 kHz point on the graph being the carrier frequency. The exact shape of the transmit spectrum is heavily dependent on the data rate and deviation frequency used.

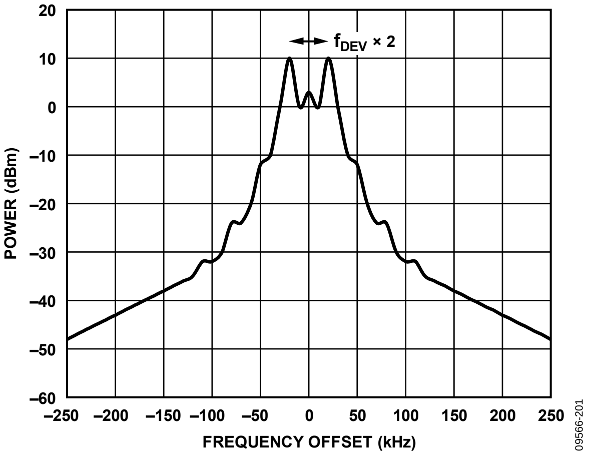


Figure 19. Idealized 2FSK Transmit Spectrum Example at 868 MHz. Data Rate = 38.4 kbps, Frequency Deviation (F_{DEV}) = 20 kHz

To ease the process of optimally configuring the transceiver three recommended options for transceiver frequency deviation and data rate are provided (see Table 62). These combinations are optimized to address the parameters of maximum transmission range and minimum power consumption. Code examples using these transceiver configurations are available and come supplied with the ADuCRF101 evaluation kit.

Table 62. Default Transceiver Configuration Options

Transceiver Configuration	Data Rate (kSPS)	Deviation (kHz)	IF Filter Bandwidth (kHz)	Configuration Target
1	1.0	4.8	100	Maximizing transmission range. Achievable due to higher radio sensitivity.
2	300	75	300	Minimizing power consumption. Achievable due to shorter transmission times.
3	38.4	20	100	Typical standard based setup. A compromise in range and power consumption.

The operation of the transceiver is supported by the provided radio interface functions. Configuring the radio for a specific transceiver configuration is achieved by a single function call with the appropriate parameter matching the desired transceiver configuration.

```
if (RIE_Response == RIE_Success)
     RIE_Response = RadioInit(DR_38_4kbps_Dev20kbps);     // Initialize the radio
```

Selection between GFSK or FSK modulation type is supported by the provided radio interface functions. Configuring the radio for a specific modulation type is achieved by a single function call with the appropriate parameter matching the desired modulation type.

```
if (RIE_Response == RIE_Success)
     RIE_Response = RadioSetModulationType (FSK_Modulation);
```

Transmit Packet Format

The transceiver uses a packet approach to transmit and receive data. The first step to transmitting data is to first load it into the on-board payload buffer (240 bytes maximum). Next, the transceiver's on-board packet handler formats the payload buffer information into the required packet structure and controls the transmission process. Conversely, on receiving data, the received packet's payload information is loaded into the payload buffer where the user can access it. The packet approach makes for a very easy to use and reliable transmission and reception system. Figure 20 shows the general flow of how user data is handled by the transceiver.

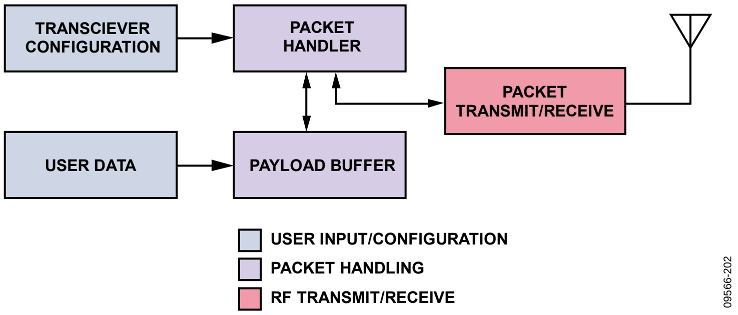


Figure 20. Transceiver Packet Handling Operation

The structure of the transmit/receive packet itself is as in Table 63. The preamble is the first part of the packet, which is mandatory, consists of 01 (0x55) transitions. The preamble can be thought of as a training signal for the transceiver to detect in receive mode, it is used to execute automatic frequency correction (AFC), calibrate the LNA gain, and configure clock and data recovery. The required preamble length depends on the transceiver configuration being used. See the Packet Mode section and Table 90 for more details.

Table 63. Transmit Packet Format

	Preamble	Sync Word	Payload			CRC	Postamble
Field			**Length**	**Address**	**User Data**		
Field Length (Bytes)	1 to 256	1 to 3	1	1 to 9	0 to 240	2 bytes	2 bytes

After the preamble comes the sync word. The function of the sync word is to mark the end of the preamble, the beginning of the payload, and it is used by the transceiver for byte level synchronization. See the Packet Mode section and Table 90 for more details.

The payload comes next. It can contain up to 240 bytes of user definable data. Note that 10 bytes of the payload are reserved for optional address data and one byte is reserved for a data length flag (indicating either a variable data length type or fixed data length type). See the Packet Mode section for more details.

After the payload is the optional CRC (cyclical redundancy check) data. It is 2 bytes in length. The CRC data is used by the transceiver to check the integrity of received data.

Finally, the packet ends with the postamble which is 2 bytes of 01 transitions. The first byte of the postamble is transmitted to mark the end of the CRC and the second byte is transmitted to indicate the PA is ramping down.

Transmission of a packet is supported by the provided radio interface functions. An example of the transmission of a fixed length packet is shown here. Variable length packets are also supported.

```
if (RIE_Response == RIE_Success)
     RIE_Response =  RadioTxPacketFixedLen(12, "HELLO WORLD");
```

RSSI

RSSI stands for receive signal strength indication. When a valid packet is received, the RSSI level is automatically detected during postamble and the value is stored in the RSSI_READBACK register. The RSSI detection range is from −97 dBm to −26 dBm with an accuracy of ±3 dBm. Refer to the RSSI description in the RF Transceiver Advanced Features section for more details.

Reading of the RSSI is supported by the provided radio interface functions. The RSSI of a received packet can be read back when retrieving the received packet.

```
if (RIE_Response == RIE_Success)
      RIE_Response = RadioRxPacketRead(sizeof(Buffer), &PktLen, Buffer, &RSSI);
```

Example Program

In this section, code used to setup the transceiver to transmit and receive the text “HELLO WORLD” is described. The void transmit (void) function below configures the transceiver for transmission and issues a transmit packet command. It first calls the RadioInit() function to setup the data rate and frequency deviation of the transceiver. In this example, a data rate of 38.4 kbps and a frequency deviation of 20 kHz are used The next function called is RadioSetFrequency() which sets the carrier frequency, here it is set to 915 MHz. Next, the RadioTxSetPA() function is called; this selects which PA is to be used (single ended or differential) and at what output power. Here the differential PA is used with a maximum output power setting of 15 chosen (~10 dBm). Finally, the packet is transmitted using the RadioTxPacketFixedLen() function. In this example, “HELLO WORLD” is transmitted using 12 bytes of payload data.

```
void transmit(void)
{
  RIE_Responses RIE_Response = RIE_Success;

  if (RIE_Response == RIE_Success)
     RIE_Response = RadioInit(DR_38_4kbps_Dev20kbps);          // Initialize radio
  if (RIE_Response == RIE_Success)
     RIE_Response = RadioSetFrequency(915000000);               // Set the carrier frequency
  if (RIE_Response == RIE_Success)
     RIE_Response = RadioTxSetPA(DifferentialPA,PowerLevel15); // Set the PA and 
power level
  if (RIE_Response == RIE_Success)
     RIE_Response = RadioTxPacketFixedLen(12, "HELLO WORLD");  // Transmit a fixed 
packet length
}
```

The void receive (void) function is an example of how to program the transceiver to receive a packet. A variable array called Buffer is first defined; this array will collect the received payload data from the packet. Packet length and RSSI level variables are also defined. The first function to be called is the RadioInit() function to setup the transceiver. As in the transmit case, it is setup with a data rate of 38.4 kbps and a frequency deviation of 20 kHz. The next function called is RadioSetFrequency() which sets the carrier frequency, here it is set to 915 MHz. Next, the RadioRxPacketFixedLen() function is called; this instructs the packet handler how many payload bytes the packet will have. The next function called is the RadioRxPacketAvailable() function; it is in a while loop. This function continually checks to see if a packet has arrived. On the arrival of a packet the RadioRxPacketRead() function is called. The payload data of the packet is moved into the Buffer array and the RSSI level read. Finally, the printf() function is used to print out the payload and RSSI level via the UART. Typically a Hyperterminal session is used to view printf() output.

```
void receive(void)
{
  RIE_Responses RIE_Response = RIE_Success;
  unsigned char  Buffer[0x20];
  RIE_U8         PktLen;
  RIE_S8         RSSI;

  if (RIE_Response == RIE_Success)
     RIE_Response = RadioInit(DR_38_4kbps_Dev20kbps);   // Initialize radio
  if (RIE_Response == RIE_Success)
     RIE_Response = RadioSetFrequency(915000000);        // Set the carrier
frequency
  if (RIE_Response == RIE_Success)
     RIE_Response = RadioRxPacketFixedLen(12);           // Receive a fixed packet
length
  if (RIE_Response == RIE_Success)
     {
     while (!RadioRxPacketAvailable());                  // Wait to receive a
packet
     }
  if (RIE_Response == RIE_Success)
     RIE_Response = RadioRxPacketRead(sizeof(Buffer), &PktLen, Buffer, &RSSI);
                                                         // Read the packet payload
                                                            into the Buffer array

  if (RIE_Response == RIE_Success)
     printf("\n\r-> %s @ RSSI %d",Buffer,(int)RSSI);     //Print the RSSI level and
the contents of Buffer out to the UART
  else
     printf("\n\r-> ERROR");
}
```

Other Considerations

Transceiver power consumption is an important factor in many wireless applications especially when battery operated. The transceiver has two PA options to allow greater usage flexibility. The single-ended PA can output up to 13 dBm of RF power and the differential PA can output up to 10 dBm. The output power of the PA is configured using the RadioTxSetPA() function, as shown in the previous code example code. For convenience there are 16 possible power settings [0-15] with 15 being the highest power output setting. For illustration Table 64 shows a summary of some of the PA output powers versus transceiver Idd current consumption, for completeness the receive mode current consumption is also shown.

Table 64. PA Output Power vs. Transceiver Idd Current Consumption Summary

Transceiver State (868 MHz/915 MHz)	Output Power (dBm)	Typical Idd (mA)
Single-ended PA, Tx Mode	−10	10.3
	0	13.3
	10	24.1
	13	32.1
Differential PA, Tx Mode	−10	9.3
	0	12
	10	28
Rx Mode	–	12.8

AFC

In the ideal case, both the transmitter and receiver are operating at the exact same frequency. The receiver knows the correct frequency spectrum locations to look for the modulated data.

This is not always the case. Receiver sensitivity (that is, the ability to receive packets) is degraded when there is a difference between transmitter and receiver.

To optimize receiver sensitivity AFC is available. AFC compensates for the frequency offset between the transmitted frequency and the received frequency. AFC works during a package preamble; it monitors the frequency error between the internal RF synthesizer frequency and the received signal. The synthesizer's local oscillator is adjusted to lock onto the received signal and thus optimize receiver sensitivity. The number of preamble bits required to pull in the frequency, that is, achieve lock, will depend on the data rate. See Table 90 for more details.

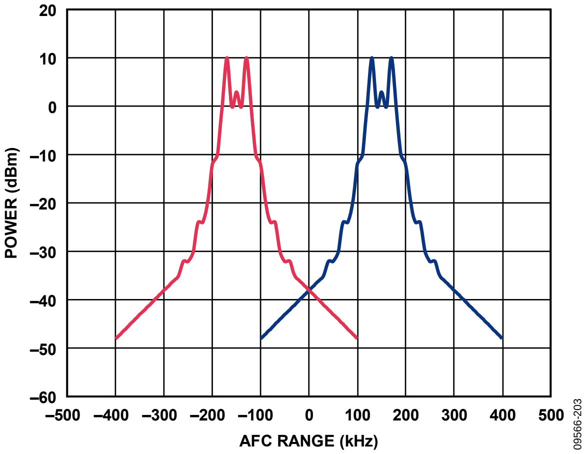


Figure 21. Maximum AFC Frequency Pull-In Range, ±150 kHz, IF Bandwidth = 300 kHz

The frequency pull-in range is set automatically depending on the IF filter bandwidth setting being used. See Table 88 in the AFC section for more details. For illustration, Figure 21 shows the maximum allowed frequency offset that the receiver using AFC can compensate for, +150 kHz and −150 kHz. To achieve this maximum pull in range the IF bandwidth must be set equal to 300 kHz. AFC is automatically configured and available in all the provided transceiver configurations.

RF TRANSCEIVER ADVANCED FEATURES

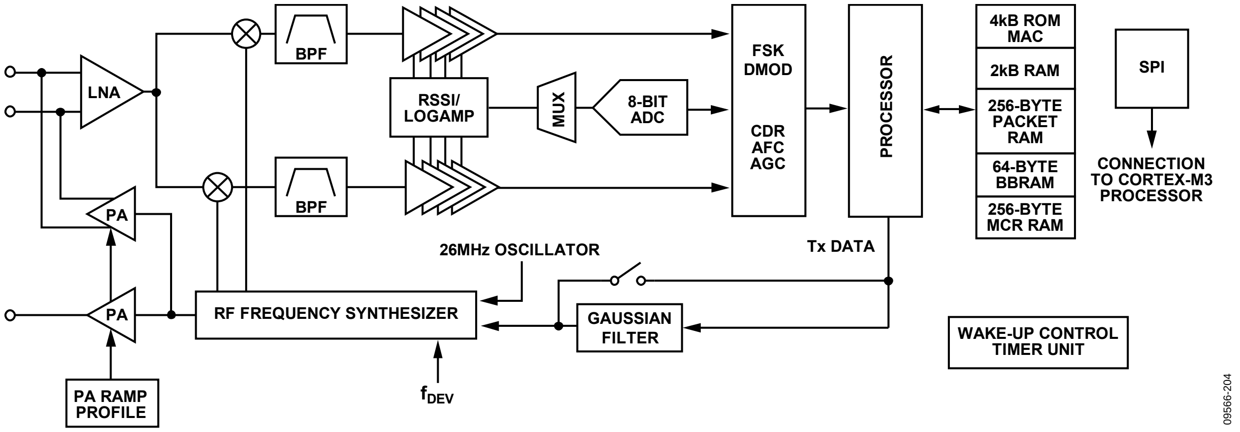


Figure 22. Functional Block Diagram

DETAILED DESCRIPTION

The RF transceiver is a very low power, high performance, highly integrated 2FSK/GFSK/MSK/GMSK transceiver designed for operation in the 862 MHz to 928 MHz and 431 MHz to 464 MHz bands.

The transmit RF synthesizer contains a VCO and a low noise fractional-N phase locked loop (PLL) with an output channel frequency resolution of 400 Hz. The VCO operates at twice the fundamental frequency to reduce spurious emissions. The receive and transmit synthesizer bandwidths are automatically, and independently, configured to achieve optimum phase noise, modulation quality, and settling time. The transmitter output power is programmable from −20 dBm to +13.5 dBm, with automatic PA ramping to meet transient spurious specifications. The part possesses both single-ended and differential PAs, which allow for Tx antenna diversity.

The part is extremely resilient to the presence of interferers in spectrally noisy environments. The receiver features a novel, high speed, AFC loop, allowing the PLL to find and correct any RF frequency errors in the recovered packet. A patent pending image rejection calibration scheme is available by downloading the image rejection calibration firmware module to program RAM. The algorithm does not require the use of an external RF source nor does it require any user intervention once initiated. The results of the calibration can be stored in nonvolatile memory for use on subsequent power-ups of the transceiver.

The RF transceiver can enter a low power sleep mode in which the configuration settings are retained in the battery backup random access memory (BBRAM).

The RF transceiver features an ultralow power, on-chip, communications processor. The communications processor, which is an 8-bit RISC processor, performs the radio control and packet management functionality. The communications processor eases the processing burden of the companion processor by integrating the lower layers of a typical communication protocol stack.

The communications processor provides a simple command-based radio control interface for the Cortex-M3 processor. A single-byte command transitions the radio between states or performs a radio function.

The communications processor provides support for generic packet formats. In transmit mode, the communications adds preamble, sync word, and CRC to the payload data stored in packet RAM. In receive mode, the communications processor can detect and interrupt the Cortex-M3 processor on reception of preamble, sync word, address, and CRC and store the received payload to packet RAM. The RF transceiver uses an efficient interrupt system comprising MAC level interrupts and PHY level interrupts that can be individually set. The payload data plus the 16-bit CRC can be encoded/decoded using Manchester. Alternatively, data whitening and dewhitening can be applied.

RADIO CONTROL

The RF transceiver has five radio states designated PHY_SLEEP, PHY_OFF, PHY_ON, PHY_RX, and PHY_TX. The Cortex-M3 processor can transition the RF transceiver between states by issuing single byte commands over the SPI interface. The various commands and states are illustrated in Figure 23. The communications processor handles the sequencing of various radio circuits and critical timing functions, thereby simplifying radio operation and easing the burden on the Cortex-M3 processor.

RADIO STATES

PHY_SLEEP

In this state, the device is in a low power sleep mode. To enter the state, issue the CMD_PHY_SLEEP command, either from the PHY_OFF or PHY_ON state. To wake the radio from the state, set the $\overline{\text{CS}}$ pin low. If retention of BBRAM contents is not required, Deep

Sleep Mode 2 can be used to further reduce the PHY_SLEEP state current consumption. Deep Sleep Mode 2 is entered by issuing the CMD_HW_RESET command. The options for the PHY_SLEEP state are detailed in Table 65. When in PHY_SLEEP, the P2.4/IRQ8 interrupt pin is held at logic low while the other GPx pins are in a high impedance state.

PHY_OFF

In the PHY_OFF state, the 26 MHz crystal, the digital regulator, and the synthesizer regulator are powered up. All memories are fully accessible. The BBRAM registers must be valid before exiting this state.

PHY_ON

In the PHY_ON state, along with the crystal, the digital regulator and the synthesizer regulator, VCO, and RF regulators are powered up. A baseband filter calibration is performed when this state is entered from the PHY_OFF state. The device is ready to operate, and the PHY_TX and PHY_RX states can be entered.

PHY_TX

In the PHY_TX state, the synthesizer is enabled and calibrated. The power amplifier is enabled, and the device transmits at the channel frequency defined by the CHANNEL_FREQ[23:0] setting (Address 0x109 to Address 0x10B). The state is entered by issuing the CMD_PHY_TX command. The device automatically transmits the transmit packet stored in the packet RAM. After transmission of the packet, the PA is disabled and the device automatically returns to the PHY_ON state and can, optionally, generate an interrupt.

PHY_RX

In the PHY_RX state, the synthesizer is enabled and calibrated. The RSSI, IF filter, mixer, and LNA are enabled. The radio is in receive mode on the channel frequency defined by the CHANNEL_FREQ [23:0] setting (Address 0x109 to Address 0x10B).

After reception of a valid packet, the device returns to the PHY_ON state and can, optionally, generate an interrupt.

Current Consumption

The typical current consumption in each state is detailed in Table 65.

Table 65. Current Consumption Contribution of RF Transceiver Radio States

State	Current (Typical)	Conditions
PHY_SLEEP (Deep Sleep Mode 2)	0.18 µA	Wake-up timer off, BBRAM contents not retained, entered by issuing CMD_HW_RESET
PHY_SLEEP (Deep Sleep Mode 1)	0.33 µA	BBRAM contents retained
PHY_OFF	1.0 mA	
PHY_ON	1.0 mA	
PHY_TX	24.1 mA	10 dBm, single-ended PA, 868 MHz
PHY_RX	12.8 mA	

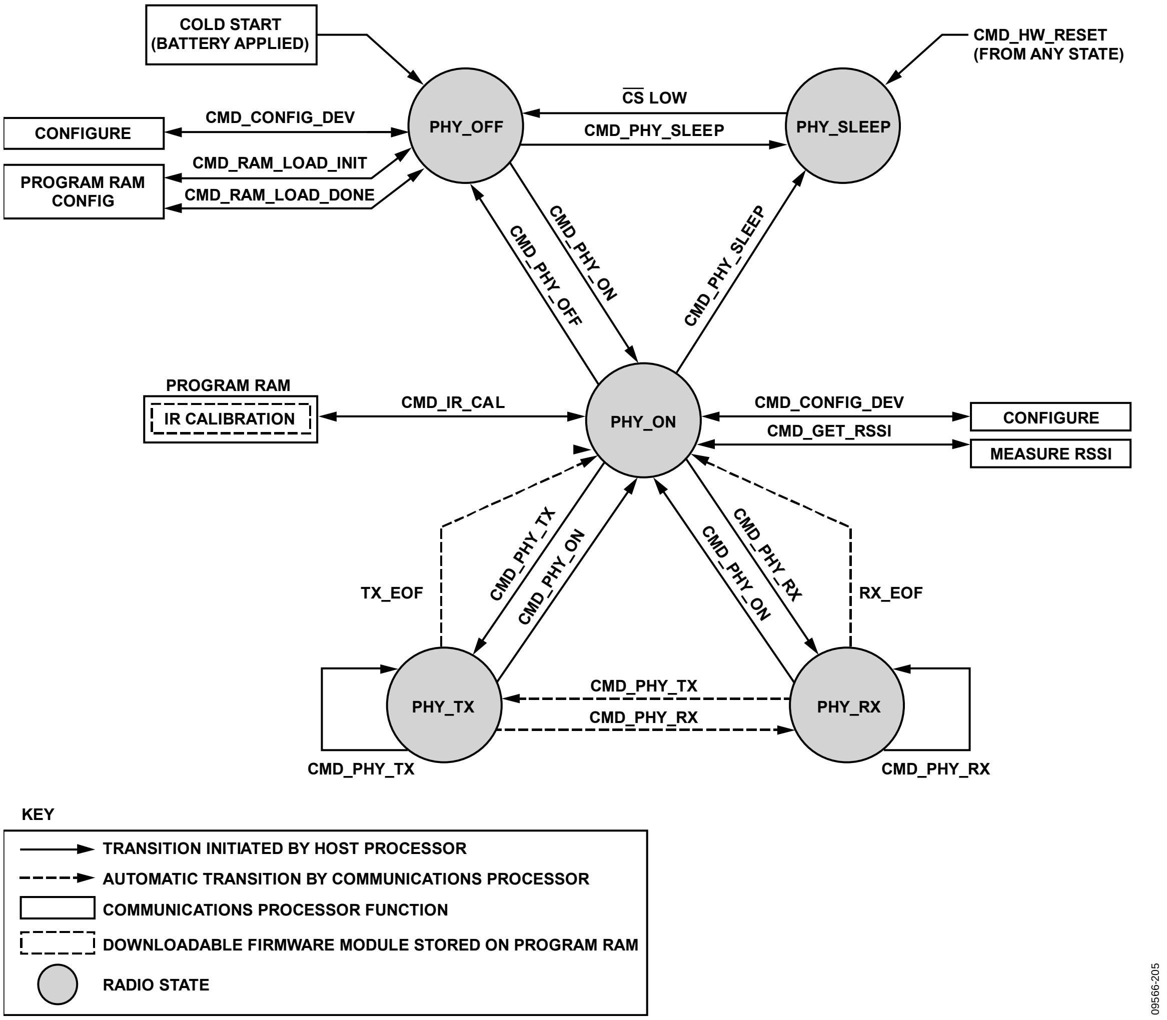


Figure 23. Radio State Diagram

INITIALIZATION

Initialization after Application of Power

When power is applied to the RF transceiver, it registers a power-on reset event (POR) and transitions to the PHY_OFF state. The BBRAM memory is unknown, the packet RAM memory is cleared to 0x00, and the MCR memory is reset to its default values. The Cortex-M3 processor should use the following procedure to complete the initialization sequence:

1. Bring the $\overline{\text{CS}}$ line of the SPI low and wait until the MISO output goes high.
2. Poll status word and wait for the CMD_READY bit to go high.
3. Configure the part by writing to all 64 of the BBRAM registers.
4. Issue the CMD_CONFIG_DEV command so that the radio settings are updated using the BBRAM values.

The RF transceiver is now configured in the PHY_OFF state.

Initialization after Issuing the CMD_HW_RESET Command

The CMD_HW_RESET command performs a full power-down of all radio transceiver hardware, and the radio transceiver enters the PHY_SLEEP state. To complete the hardware reset, the Cortex-M3 processor should complete the following procedure:

1. Wait for 1 ms.
2. Bring the $\overline{\text{CS}}$ line of the SPI low and wait until the MISO output goes high. The RF transceiver registers a POR and enters the PHY_OFF state.
3. Poll status word and wait for the CMD_READY bit to go high.
4. Configure the part by writing to all 64 of the BBRAM registers.
5. Issue the CMD_CONFIG_DEV command so that the radio settings are updated using the BBRAM values.

The RF transceiver is now configured in the PHY_OFF state.

Initialization on Transitioning from PHY_SLEEP (After $\overline{\text{CS}}$ Is Brought Low)

The Cortex-M3 processor can bring $\overline{\text{CS}}$ low at any time to wake the RF transceiver from the PHY_SLEEP state. This event is not registered as a POR event because the BBRAM contents are valid. The following is the procedure that the Cortex-M3 processor is required to follow:

1. Bring the $\overline{\text{CS}}$ line of the SPI low and wait until the MISO output goes high. The RF transceiver enters the PHY_OFF state.
2. Poll status word and wait for the CMD_READY bit to go high.
3. Issue the CMD_CONFIG_DEV command so that the radio settings are updated using the BBRAM values.

The RF transceiver is now configured and ready to transition to the PHY_ON state.

COMMANDS

The commands that are supported by the radio controller are detailed in this section. They initiate transitions between radio states or perform tasks as indicated in Figure 23.

CMD_PHY_OFF (0xB0)

This command transitions the RF transceiver to the PHY_OFF state. It can be issued in the PHY_ON state. It powers down the RF and VCO regulators.

CMD_PHY_ON (0xB1)

This command transitions the RF transceiver to the PHY_ON state.

If the command is issued in the PHY_OFF state, it powers up the RF and VCO regulators and performs an IF filter calibration.

If the command is issued from the PHY_TX state, the Cortex-M3 processor performs the following procedure:

1. Ramp down the PA.
2. Turn off the digital transmit clocks.
3. Power down the synthesizer.
4. Set FW_STATE = PHY_ON.

If the command is issued from the PHY_RX state, the communications processor performs the following procedure:

1. Copy the measured RSSI to the RSSI_READBACK register.
2. Turn off the digital receiver clocks.
3. Power down the synthesizer and the receiver circuitry (ADC, RSSI, IF filter, mixer, and LNA).
4. Set FW_STATE = PHY_ON.

CMD_PHY_SLEEP (0xBA)

This command transitions the RF transceiver to the very low power PHY_SLEEP state in which the WUC is operational (if enabled), and the BBRAM contents are retained. It can be issued from the PHY_OFF or PHY_ON state.

CMD_PHY_RX (0xB2)

This command can be issued in the PHY_ON, PHY_RX, or PHY_TX state. If the command is issued in the PHY_ON state, the communications processor performs the following procedure:

1. Power up the synthesizer.
2. Power up the receiver circuitry (RSSI, IF filter, mixer, and LNA).
3. Set the RF channel based on the CHANNEL_FREQ[23:0] setting in BBRAM.
4. Set the synthesizer bandwidth.
5. Do VCO calibration.
6. Delay for synthesizer settling.
7. Enable the digital receiver blocks.
8. Set FW_STATE = PHY_RX.

If the command is issued in the PHY_RX state, the communications processor performs the following procedure:

1. Unlock the AFC and AGC.
2. Turn off the receive blocks.
3. Set the RF channel based on the CHANNEL_FREQ[23:0] setting in BBRAM.
4. Set the synthesizer bandwidth.
5. Do VCO calibration.
6. Delay for synthesizer settling.
7. Enable the digital receiver blocks.
8. Set FW_STATE = PHY_RX.

If the command is issued in the PHY_TX state, the communications processor performs the following procedure:

1. Ramp down the PA.
2. Turn off the digital transmit blocks.
3. Power up the receiver circuitry (ADC, RSSI, IF filter, mixer, and LNA).
4. Set the RF channel based on the CHANNEL_FREQ[23:0] setting in BBRAM.
5. Set the synthesizer bandwidth.
6. Do VCO calibration.
7. Delay for synthesizer settling.
8. Enable the digital receiver blocks.
9. Set FW_STATE = PHY_RX

CMD_PHY_TX (0xB5)

This command can be issued in the PHY_ON, PHY_TX, or PHY_RX state. If the command is issued in the PHY_ON state, the communications processor performs the following procedure:

1. Power up the synthesizer.
2. Set the RF channel based on the CHANNEL_FREQ[23:0] setting in BBRAM.
3. Set the synthesizer bandwidth.
4. Do VCO calibration.
5. Delay for synthesizer settling.
6. Enable the digital transmit blocks.
7. Ramp up the PA.
8. Set FW_STATE = PHY_TX.
9. Transmit data.

If the command is issued in the PHY_TX state, the communications processor performs the following procedure:

1. Ramp down the PA.
2. Turn off the digital transmit blocks.
3. Set the RF channel based on the CHANNEL_FREQ[23:0] setting in BBRAM.
4. Set the synthesizer bandwidth.
5. Do VCO calibration.
6. Delay for synthesizer settling.
7. Enable the digital transmit blocks.
8. Ramp up the PA.
9. Set FW_STATE = PHY_TX.
10. Transmit data.

If the command is issued in the PHY_RX state, the communications processor performs the following procedure:

1. Unlock the AFC and AGC.
2. Turn off the receive blocks.
3. Power down the receiver circuitry (RSSI, IF filter, mixer, and LNA).
4. Set the RF channel based on the CHANNEL_FREQ[23:0] setting in BBRAM.
5. Set the synthesizer bandwidth.
6. Delay for synthesizer settling.
7. Enable the digital transmit blocks.
8. Ramp up the PA.

9. Set FW_STATE = PHY_TX.
10. Transmit data.

CMD_CONFIG_DEV (0xBB)

This command interprets the BBRAM contents and configures each of the radio parameters based on these contents. It can be issued from the PHY_OFF or PHY_ON state. The only radio parameter that is not configured on this command is the CHANNEL_FREQ[23:0] setting, which instead is configured as part of a CMD_PHY_TX or CMD_PHY_RX command.

The user should write to the entire 64 bytes of the BBRAM and then issue the CMD_CONFIG_DEV command, which can be issued in the PHY_OFF or PHY_ON state.

CMD_GET_RSSI (0xBC)

This command turns on the receiver, performs an RSSI measurement on the current channel, and returns the RF transceiver to the PHY_ON state. The command can be issued from the PHY_ON state. The RSSI result is saved to the RSSI_READBACK register (Address 0x312). This command can be issued from the PHY_ON state only.

CMD_HW_RESET (0xC8)

The command performs a full power-down of all hardware, and the device enters the PHY_SLEEP state. This command can be issued in any state and is independent of the state of the communications processor. The procedure for initialization of the device after a CMD_HW_RESET command is described in detail in the Initialization section.

CMD_RAM_LOAD_INIT (0xBF)

This command prepares the communications processor for a subsequent download of a software module to program RAM. This command should be issued only prior to the program RAM being written to by the Cortex-M3 processor.

CMD_RAM_LOAD_DONE (0xC7)

This command is required only after download of a software module to program RAM. It indicates to the communications processor that a software module is loaded to program RAM. The CMD_RAM_LOAD_DONE command can be issued only in the PHY_OFF state. The command resets the communications processor and the packet RAM.

CMD_IR_CAL (0xBD)

This command performs a fully automatic image rejection calibration on the RF transceiver receiver.

This command requires that the IR calibration firmware module has been loaded to the RF transceiver program RAM.

AUTOMATIC STATE TRANSITIONS

On certain events, the communications processor can automatically transition the RF transceiver between states. These automatic transitions are illustrated as dashed lines in Figure 23 and are explained in this section.

TX_EOF

The communications processor automatically transitions the device from the PHY_TX state to the PHY_ON state at the end of a packet transmission. On the transition, the communications processor performs the following actions:

1. Ramps down the PA.
2. Disables the digital transmitter blocks.
3. Powers down the synthesizer.
4. Sets FW_STATE = PHY_ON.

RX_EOF

The communications processor automatically transitions the device from the PHY_RX state to the PHY_ON state at the end of a packet reception. On the transition, the communications processor performs the following actions:

1. Copies the measured RSSI to the RSSI_READBACK register (Address 0x312).
2. Disables the digital receiver blocks.
3. Powers down the synthesizer and the receiver circuitry (ADC, RSSI, IF filter, mixer, and LNA).
4. Sets FW_STATE = PHY_ON.

STATE TRANSITION AND COMMAND TIMING

The execution times for all radio state transitions are detailed in Table 66 and Table 67. Note that these times are typical and can vary, depending on the BBRAM configuration.

For normal transition times, set TRANSITION_CLOCK_DIV (Location 0x13A) to 0x04. For fast transition times, set TRANSITION_CLOCK_DIV to 0x01. It is recommended to enable fast transition times to reduce system power consumption.

As stated in the SPI Interface section, commands are executed on the last positive $\overline{\text{SCLK}}$ edge of the command. For the values given, there is an additional 200 ns between the last positive SCLK edge and the rising edge of $\overline{\text{CS}}$ that is related to the SPI rate used.

Table 66. RF Transceiver Command Execution Times and State Transition Times Not Related to PHY_TX or PHY_RX

Command/Bit	Command Initiated By	Present State	Next State	Normal Transition Time (µs), Typical	Fast Transition Time (µs), Typical	Condition
CMD_HW_RESET	processor	Any	PHY_SLEEP	1	1	
CMD_PHY_SLEEP	processor	PHY_OFF	PHY_SLEEP	22.3	22.3	
CMD_PHY_SLEEP	processor	PHY_ON	PHY_SLEEP	24.1	24.1	
CMD_PHY_OFF	processor	PHY_ON	PHY_OFF	24	11	From rising edge of $\overline{\text{CS}}$ to CMD_FINISHED interrupt
CMD_PHY_ON	processor	PHY_OFF	PHY_ON	258/73	213/28	From rising edge of $\overline{\text{CS}}$ to CMD_FINISHED interrupt; IF filter calibration enabled/disabled
CMD_GET_RSSI	processor	PHY_ON	PHY_ON	631/450	523/353	RSSI_WAIT_TIME (Address 0x138) = 0xA7/0x37
CMD_CONFIG_DEV	processor	PHY_OFF	PHY_OFF	72	23	From rising edge of $\overline{\text{CS}}$ to CMD_FINISHED interrupt
CMD_CONFIG_DEV	processor	PHY_ON	PHY_ON	75.5	24.5	From rising edge of $\overline{\text{CS}}$ to CMD_FINISHED interrupt
Wake-Up from PHY_SLEEP, ($\overline{\text{CS}}$ Low)	processor	PHY_SLEEP	PHY_OFF	304	304	7 pF load capacitance, T_A = 25°C
Cold Start	application of power	N/A	PHY_OFF	304	304	7 pF load capacitance, T_A = 25°C

Table 67. RF Transceiver State Transition Times Related to PHY_TX and PHY_RX

Command/Bit/ Automatic Transition	Present State	Next State	Normal Transition Time (µs)$^{1, 2}$, Typical	Fast Transition Time (µs)$^{1, 2,}$ Typical	Condition
CMD_PHY_ON	PHY_TX	PHY_ON	T_{EOP} + T_{PARAMP_DOWN} + T_{BYTE} + 43	T_{EOP} + T_{PARAMP_DOWN} + T_{BYTE} + 15	From rising edge of $\overline{\text{CS}}$ to CMD_FINISHED interrupt
CMD_PHY_ON	PHY_RX	PHY_ON	T_{BYTE} + 48	T_{BYTE} + 21	From rising edge of $\overline{\text{CS}}$ to CMD_FINISHED interrupt, CMD_PHY_ON issued during search for preamble
			50.5	23	From rising edge of $\overline{\text{CS}}$ to CMD_FINISHED interrupt, CMD_PHY_ON issued during preamble qualification
			50.5	23	From rising edge of $\overline{\text{CS}}$ to CMD_FINISHED interrupt, CMD_PHY_ON issued during sync word qualification
			T_{EOP} + 62.5	T_{EOP} + 18	From rising edge of $\overline{\text{CS}}$ to CMD_FINISHED interrupt, CMD_PHY_ON issued during Rx data (after a sync word)
CMD_PHY_TX	PHY_ON	PHY_TX	306	237	From rising edge of $\overline{\text{CS}}$ to CMD_FINISHED interrupt; PA ramp up starts 3.4 µs after the interrupt; first bit of user data is transmitted 1.5 × T_{BIT} + 2.3 µs following the interrupt

Command/Bit/ Automatic Transition	Present State	Next State	Normal Transition Time (µs)[1, 2], Typical	Fast Transition Time (µs)[1, 2,] Typical	Condition
CMD_PHY_TX	PHY_RX	PHY_TX	T_{BYTE} + 324.5	T_{BYTE} + 248	From rising edge of $\overline{CS}$ to CMD_FINISHED interrupt, CMD_PHY_TX issued during search for preamble; PA ramp up starts 3.4 µs after the interrupt; first bit of user data is transmitted 1.5 × T_{BIT} + 2.3 µs following the interrupt
			322.5	245.5	From rising edge of $\overline{CS}$ to CMD_FINISHED interrupt, CMD_PHY_TX issued during preamble qualification; PA ramp up starts 3.4 µs after the interrupt; first bit of user data is transmitted 1.5 × T_{BIT} + 2.3 µs following the interrupt
			322.5	245.5	From rising edge of $\overline{CS}$ to CMD_FINISHED interrupt, CMD_PHY_TX issued during sync word qualification; PA ramp up starts 3.4 µs after the interrupt; first bit of user data is transmitted 1.5 × T_{BIT} + 2.3 µs following the interrupt
			T_{EOP} + 281	T_{EOP} + 263	From rising edge of $\overline{CS}$ to CMD_FINISHED interrupt, CMD_PHY_TX issued during Rx data (after a sync word); PA ramp up starts 3.4 µs after the interrupt; first bit of user data is transmitted 1.5 × T_{BIT} + 2.3 µs following the interrupt
CMD_PHY_TX	PHY_TX	PHY_TX	T_{EOP} + T_{PARAMP_DOWN} + T_{BYTE} + 310	T_{EOP} + T_{PARAMP_DOWN} + T_{BYTE} + 236	From rising edge of $\overline{CS}$ to CMD_FINISHED interrupt. CMD_PHY_TX issued during packet transmission; PA ramp up starts 3.4 µs after the interrupt; first bit of user data is transmitted 1.5 × T_{BIT} + 2.3 µs following the interrupt
CMD_PHY_RX	PHY_ON	PHY_RX	327	241	From rising edge of $\overline{CS}$ to CMD_FINISHED interrupt
CMD_PHY_RX	PHY_TX	PHY_RX	T_{EOP} + T_{PARAMP_DOWN} + T_{BYTE} + 336	T_{EOP} + T_{PARAMP_DOWN} + T_{BYTE} + 241	From rising edge of $\overline{CS}$ to CMD_FINISHED interrupt; CMD_PHY_RX issued during packet transmission
CMD_PHY_RX	PHY_RX	PHY_RX	T_{BYTE} + 341.5	T_{BYTE} + 249.5	From rising edge of $\overline{CS}$ to CMD_FINISHED interrupt, CMD_PHY_RX issued during search for preamble
			339.5	249	From rising edge of $\overline{CS}$ to CMD_FINISHED interrupt, CMD_PHY_RX issued during preamble qualification
			339.5	249	From rising edge of $\overline{CS}$ to CMD_FINISHED interrupt, CMD_PHY_RX issued during sync word qualification
			T_{EOP} + 354	T_{EOP} + 246	From rising edge of $\overline{CS}$ to CMD_FINISHED interrupt, CMD_PHY_RX issued during Rx data (after a sync word)
TX_EOF	PHY_TX	PHY_ON	T_{PARAMP_DOWN} + T_{BYTE} + 25	T_{PARAMP_DOWN} + T_{BYTE} + 5	From TX_EOF interrupt to CMD_FINISHED interrupt
RX_EOF	PHY_RX	PHY_ON	46	10	From INTERRUPT_CRC_CORRECT to CMD_FINISHED interrupt

[1] $T_{PARAMP\_DOWN} = T_{PARAMP\_UP} = \frac{PA\_LEVEL\_M\ CR}{2^{(9 - PA\_RAMP)} \times DATA\_RATE \times 100}$, where *PA_LEVEL_MCR* sets the maximum PA output power (PA_LEVEL_MCR register, Address 0x307), *PA_RAMP* sets the PA ramp rate (RADIO_CFG_8 register, Address 0x114), and *DATA_RATE* sets the transmit data rate (RADIO_CFG_0 register, Address 0x10C and RADIO_CFG_1 register, Address 0x10D).

[2] T_{BIT} = one bit period (µs), T_{BYTE} = one byte period (µs), T_{EOP} = time to end of packet (µs).

PACKET MODE

The on-chip communications processor can be configured for use with a wide variety of packet-based radio protocols using 2FSK/GFSK /MSK/GMSK modulation. The general packet format is shown in Table 69; 240 bytes of dedicated packet RAM are available to store, transmit, and receive packets. In transmit mode, preamble, sync word, and CRC can be added by the communications processor to the data stored in the packet RAM for transmission. In addition, all packet data after the sync word can be optionally whitened or Manchester encoded on transmission and decoded on reception.

In receive mode, the communications processor can be used to qualify received packets based on the preamble detection, sync word detection, CRC detection, or address match and generate an interrupt on the P2.4/IRQ8 pin. On reception of a valid packet, the received payload data is loaded to packet RAM memory. More information on interrupts is contained in the Interrupt Generation section.

PREAMBLE

The preamble is a mandatory part of the packet that is automatically added by the communications processor when transmitting a packet and removed after receiving a packet.
The preamble is a 0x55 sequence, with a programmable length between 1 byte and 256 bytes, that is set in the PREAMBLE_LEN register (Address 0x11D). It is necessary to have preamble at the beginning of the packet to allow time for the receiver AGC, AFC, and clock and data recovery circuitry to settle before the start of the sync word. The required preamble length depends on the radio configuration. See the Radio Blocks section for more details.

In receive mode, the RF transceiver can use a preamble qualification circuit to detect preamble and interrupt the Cortex-M3 processor. The preamble qualification circuit tracks the received frame as a sliding window. The window is three bytes in length, and the preamble pattern is fixed at 0x55. The preamble bits are examined in 01 pairs. If either bit or both bits are in error, the pair is deemed erroneous. The possible erroneous pairs are 00, 11, and 10. The number of erroneous pairs tolerated in the preamble can be set using the PREAMBLE_MATCH register value (Address 0x11B) according to Table 68.

Table 68. Preamble Detection Tolerance (PREAMBLE_ MATCH, Address 0x11B)

Value	Description
0x0C	No errors allowed.
0x0B	One erroneous bit-pair allowed in 12 bit-pairs.
0x0A	Two erroneous bit-pairs allowed in 12 bit-pairs.
0x09	Three erroneous bit-pairs allowed in 12 bit-pairs.
0x08	Four erroneous bit-pairs allowed in 12 bit-pairs.
0x00	Preamble detection disabled.

Table 69. RF transceiver Packet Structure Description[1]

	Packet Structure						
	Preamble	Sync	Payload			CRC	Postamble
Packet Format Options			Length	Address	Payload Data		
Field Length	1 byte to 256 bytes	1 bit to 24 bits	1 byte	1 byte to 9 bytes	0 bytes to 240 bytes	2 bytes	2 bytes
Optional Field in Packet Structure	X	X	Yes	Yes	Yes	Yes	X
Comms Processor Adds in Tx, Removes in Rx	Yes	Yes	X	X	X	Yes	Yes
Host Writes These Fields to Packet RAM	X	X	Yes	Yes	Yes	X	X
Whitening/Dewhitening (Optional)	X	X	Yes	Yes	Yes	Yes	X
Manchester Encoding/ Decoding (Optional)	X	X	Yes	Yes	Yes	Yes	X

[1] Yes indicates that the packet format option is supported; X indicates that the packet format option is not supported.

If PREAMBLE_MATCH is set to 0x0C, the RF transceiver must receive 12 consecutive 01 pairs (three bytes) to confirm that valid preamble has been detected. The user can select the option to automatically lock the AFC and/or AGC once the qualified preamble is detected. The AFC lock on preamble detection can be enabled by setting AFC_LOCK_MODE = 3 in the RADIO_CFG_10 register (Address 0x116:). The AGC lock on preamble detection can be enabled by setting AGC_LOCK_ MODE = 3 in the RADIO_CFG_7 register (Address 0x113).

After the preamble is detected and the end of preamble has been reached, the communications processor searches for the sync word. The search for the sync word lasts for a duration equal to the sum of the number of programmed sync word bits, plus the preamble matching tolerance (in bits) plus 16 bits. If the sync word routine is detected during this duration, the communications processor loads the received payload to packet RAM and computes the CRC (if enabled). If the sync word routine is not detected during this duration, the communications processor continues searching for the preamble.

Preamble detection can be disabled by setting the PREAMBLE_ MATCH register to 0x00. To enable an interrupt upon preamble detection, the user must set INTERRUPT_PREAMBLE_DETECT =1 in the INTERRUPT_MASK_0 register (Address 0x100).

SYNC WORD

Sync word is the synchronization word used by the receiver for byte level synchronization, while also providing an optional interrupt on detection. It is automatically added to the packet by the communications processor in transmit mode and removed during reception of a packet.

The value of the sync word is set in the SYNC_BYTE_0, SYNC_BYTE_1, and SYNC_BYTE_2 registers (Address 0x121, Address 0x122, and Address 0x123, respectively). The sync word is transmitted most significant bit first starting with SYNC_BYTE_0. The sync word matching length at the receiver is set using SYNC_WORD_LENGTH in the SYNC_CONTROL register (Address 0x120) and can be one bit to 24 bits long; the transmitted sync word is a multiple of eight bits. Therefore, for nonbyte length sync words, the transmitted sync pattern should be appended with the preamble pattern as described in Table 71 and Figure 24. In receive mode, the RF transceiver can provide an interrupt on reception of the sync word sequence programmed in the SYNC_BYTE_0, SYNC_BYTE_1, and SYNC_BYTE_2 registers. This feature can be used to alert the Cortex-M3 processor that a qualified sync word has been received. An error tolerance parameter can also be programmed that accepts a valid match when up to three bits of the sync word sequence are incorrect. The error tolerance value is set using the SYNC_ERROR_TOL setting in the SYNC_CONTROL register (Address 0x120), as described in Table 70.

Table 70. Sync Word Detection Tolerance (SYNC_ERROR_ TOL, Address 0x120)

Value	Description
00	No bit errors allowed.
01	One bit error allowed.
10	Two bit errors allowed.
11	Three bit errors allowed.

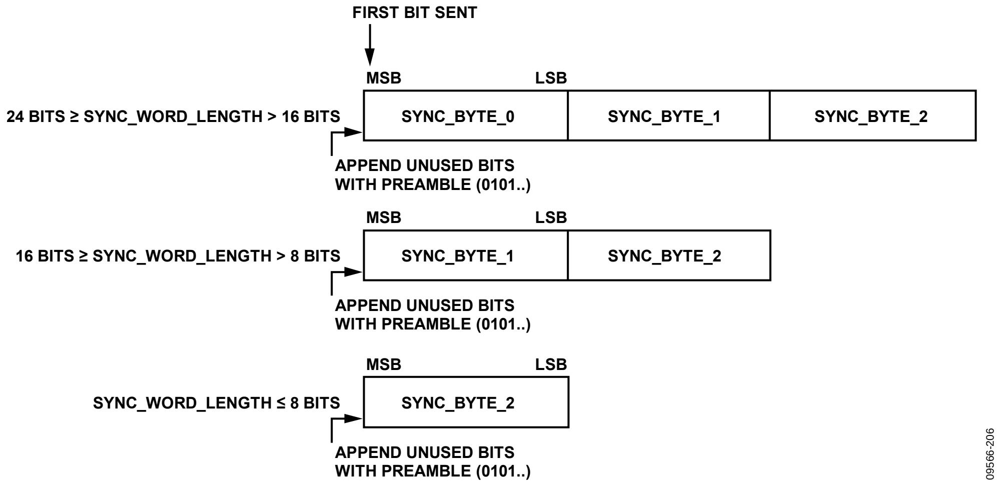


Figure 24. Transmit Sync Word Configuration

Table 71. Sync Word Programming Examples

Required Sync Word (Binary, First Bit Being First in Time)	SYNC_WORD_ LENGTH Bits in SYNC_CONTROL REGISTER (0x120)	SYNC_ BYTE_0[1]	SYNC_ BYTE_1[1]	SYNC_ BYTE_2	Transmitted Sync Word (Binary, First Bit Being First in Time)	Receiver Sync Word Match Length (Bits)
000100100011010001010110	24	0x12	0x34	0x56	0001_0010_0011_0100_0101_0110	24
111010011100101000100	21	0x5D	0x39	0x44	0101_1101_0011_1001_0100_0100	21
0001001000110100	16	0xXX	0x12	0x34	0001_0010_0011_0100	16
011100001110	12	0xXX	0x57	0x0E	0101_0111_0000_1110	12
00010010	8	0xXX	0xXX	0x12	0001_0010	8
011100	6	0xXX	0xXX	0x5C	0101_1100	6

[1] X = Don't care.

Choice of Sync Word

The sync word should be chosen to have low correlation with the preamble and have good autocorrelation properties. When the AFC is set to lock on detection of sync word (AFC_LOCK_ MODE = 3 and PREAMBLE_MATCH = 0), the sync word should be chosen to be dc free, and it should have a run length limit not greater than four bits.

PAYLOAD

The Cortex-M3 processor writes the transmit data payload to the packet RAM. The location of the transmit data in the packet RAM is defined by the TX_BASE_ADR value register (Address 0x124). The TX_BASE_ADR value is the location of the first byte of the transmit payload data in the packet RAM. On reception of a valid sync word, the communications processor automatically loads the receive payload to the packet RAM. The RX_BASE_ADR register value (Address 0x125) sets the location in the packet RAM of the first byte of the received payload. For more details on packet RAM memory, see the RF Transceiver Memory Map section.

Byte Orientation

The over-the-air arrangement of each transmitted packet RAM byte can be set to MSB first or LSB first using the DATA_BYTE setting in the PACKET_LENGTH_CONTROL register (Address 0x126). The same orientation setting should be used on the transmit and receive sides of the RF link.

Packet Length Modes

The RF transceiver can be used in both fixed and variable length packet systems. Fixed or variable length packet mode is set using the PACKET_LEN variable setting in the PACKET_ LENGTH_CONTROL register (Address 0x126).

For a fixed packet length system, the length of the transmit and received payload is set by the PACKET_LENGTH_MAX register (Address 0x127). The payload length is defined as the number of bytes from the end of the sync word to the start of the CRC.

In variable packet length mode, the communications processor extracts the length field from the received payload data. In transmit mode, the length field must be the first byte in the transmit payload.

The communications processor calculates the actual received payload length as

$$RxPayload\ Length = Length + LENGTH\_OFFSET - 4$$

where:
Length is the length field (the first byte in the received payload).
LENGTH_OFFSET is a programmable offset (set in the PACKET_LENGTH_CONTROL register (Address 0x126).

The LENGTH_OFFSET value allows compatibility with systems where the length field in the proprietary packet may also include the length of the CRC and/or the sync word. The RF transceiver defines the payload length as the number of bytes from the end of the sync word to the start of the CRC. In variable packet length mode, the PACKET_LENGTH_MAX value defines the maximum packet length that can be received, as described in Figure 25.

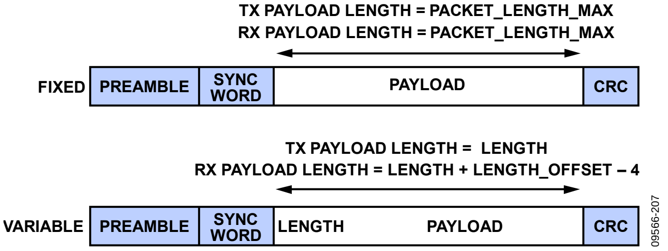

TX PAYLOAD LENGTH = PACKET_LENGTH_MAX
RX PAYLOAD LENGTH = PACKET_LENGTH_MAX
FIXED
PREAMBLE
SYNC WORD
PAYLOAD
CRC
TX PAYLOAD LENGTH = LENGTH
RX PAYLOAD LENGTH = LENGTH + LENGTH_OFFSET – 4
VARIABLE
PREAMBLE
SYNC WORD
LENGTH
PAYLOAD
CRC
09566-207

Figure 25. Payload Length in Fixed and Variable Length Packet Modes

Addressing

The RF transceiver provides a very flexible address matching scheme, allowing matching of a single address, multiple addresses, and broadcast addresses. Addresses up to 32 bits in length are supported. The address information can be included at any section of the transmit payload. The location of the starting byte of the address data in the received payload is set in the ADDRESS_MATCH_OFFSET register (Address 0x129), as illustrated in Figure 26. The number of bytes in the first address field is set in the ADDRESS_LENGTH register (Address 0x12A). These settings allow the communications processor to extract the address information from the received packet.

The address data is then compared against a list of known addresses that are stored in BBRAM (Address 0x12B to Address 0x137). Each stored address byte has an associated mask byte, thereby allowing matching of partial sections of the address bytes, which is useful for checking broadcast addresses or a family of addresses that have a unique identifier in the address sequence. The format and placement of the address information in the payload data should match the address check settings at the receiver to ensure exact address detection and qualification. Table 72 shows the register locations in the BBRAM that are used for setup of the address checking. When Register 0x12A (number of bytes in the first address field) is set to 0x00, address checking is disabled. Note that if static register fixes are employed (see Table 130), the space available for address matching is reduced.

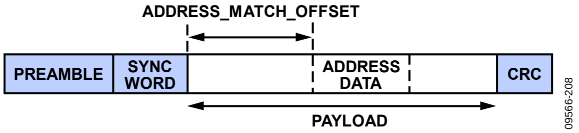


Figure 26. Address Match Offset

Table 72. Address Check Register Setup

Address (BBRAM)	Description[1]
0x129, ADDRESS_MATCH_OFFSET	Position of first address byte in the received packet (first byte after sync word = 0)
0x12A, ADDRESS_LENGTH	Number of bytes in the first address field (N_{ADR_1})
0x12B	Address Match Byte 0
0x12C	Address Mask Byte 0
0x12D	Address Match Byte 1
0x12E	Address Mask Byte 1
…	…
	Address Match Byte $N_{ADR\_1} - 1$
	Address Mask Byte $N_{ADR\_1} - 1$
	0x00 to end or N_{ADR_2} for another address check sequence

[1] N_{ADR_1} = the number of bytes in the first address field; N_{ADR_2} = the number of bytes in the second address field.

The Cortex-M3 processor should set the INTERRUPT_ADDRESS_ MATCH bit in the INTERRUPT_SOURCE_0 register (Address 0x336) if an interrupt is required on the P2.4/IRQ8 pin. Additional information on interrupts is contained in the Interrupt Generation section.

Example Address Check

Consider a system with 16-bit address lengths, in which the first byte is located in the 10th byte of the received payload data. The system also uses broadcast addresses in which the first byte is always 0xAA. To match the exact address, 0xABCD or any broadcast address in the form 0xAAXX, the RF transceiver must be configured as shown in Table 73.

Table 73. Example Address Check Configuration

BBRAM Address	Value	Description
0x129	0x09	Location in payload of the first address byte
0x12A	0x02	Number of bytes in the first address field, $N_{ADR\_1} = 2$
0x12B	0xAB	Address Match Byte 0
0x12C	0xFF	Address Mask Byte 0
0x12D	0xCD	Address Match Byte 1
0x12E	0xFF	Address Mask Byte 1
0x12F	0x02	Number of bytes in the second address field, $N_{ADR\_2} = 2$
0x130	0xAA	Address Match Byte 0
0x131	0xFF	Address Mask Byte 0
0x132	0x00	Address Match Byte 1
0x133	0x00	Address Mask Byte 1
0x134	0x00	End of addresses (indicated by 0x00)
0x135	0xXX	Don't care
0x136	0xXX	Don't care
0x137	0xXX	Don't care

CRC

An optional CRC-16 can be appended to the packet by setting CRC_EN =1 in the PACKET_LENGTH_CONTROL register (Address 0x126). In receive mode, this bit enables CRC detection on the received packet. A default polynomial is used if PROG_CRC_EN = 0 in the SYMBOL_MODE register (Address 0x11C). The default CRC polynomial is

$$g(x) = x^{16} + x^{12} + x^{5} + 1$$

Any other 16-bit polynomial can be used if PROG_CRC_EN = 1, and the polynomial is set in CRC_POLY_0 and CRC_POLY_1 (Address 0x11E and Address 0x11F, respectively). The setup of the CRC is described in Table 74.The CRC is initialized with 0x0000.

Table 74.CRC Setup

CRC_EN Bit in the PACKET_LENGTH CONTROL Register	PROG_CRC_EN Bit in the SYMBOL_MODE Register	Description
0	X[1]	CRC is disabled in transmit, and CRC detection is disabled in receive.
1	0	CRC is enabled in transmit, and CRC detection is enabled in receive, with the default CRC polynomial.
1	1	CRC is enabled in transmit, and CRC detection is enabled in receive, with the CRC polynomial defined by CRC_POLY_0 and CRC_POLY_1.

[1] X = Don't care.

To convert a user-defined polynomial to the 2-byte value, the polynomial should be written in binary format. The x^{16} coefficient is assumed equal to 1 and is, therefore, discarded. The remaining 16 bits then make up CRC_POLY_0 (most significant byte) and CRC_POLY_1 (least significant byte). Two examples of setting common 16-bit CRCs are shown in Table 75.

Table 75. Example: Programming of CRC_POLY_0 and CRC_POLY_1

Polynomial	Binary Format	CRC_POLY_0	CRC_POLY_1
$x^{16} + x^{15} + x^2 + 1$ (CRC-16-IBM)	1_1000_0000_0000_0101	0x80	0x05
$x^{16} + x^{13} + x^{12} + x^{11}\ x^{10} + x^8 + x^6 + x^5 + x^2 + 1$ (CRC-16-DNP)	1_0011_1101_0110_0101	0x3D	0x65

To enable CRC detection on the receiver, with the default CRC or user-defined 16-bit CRC, CRC_EN in the PACKET_LENGTH_CONTROL register (Address 0x126) should be set to 1. An interrupt can be generated on reception of a CRC verified packet.

POSTAMBLE

The communications processor automatically appends two bytes of postamble to the end of the transmitted packet. Each byte of the postamble is 0x55. The first byte is transmitted immediately after the CRC. The PA ramp-down begins immediately after the first postamble byte. The second byte is transmitted while the PA is ramping down.

On the receiver, if the received packet is valid, the RSSI is automatically measured during the first postamble byte, and the result is stored in the RSSI_READBACK register (Address 0x312). The RSSI is measured by the communications processor 17 μs after the last CRC bit.

TRANSMIT PACKET TIMING

The PA ramp timing in relation to the transmit packet data is described in Figure 27. After the CMD_PHY_TX command is issued, a VCO calibration is carried out, followed by a delay for synthesizer settling. The PA ramp follows the synthesizer settling. After the PA is ramped up to the programmed rate, there is 1-byte delay before the start of modulation (preamble). At the beginning of the second byte of postamble, the PA ramps down. The communications processor then transitions to the PHY_ON state.

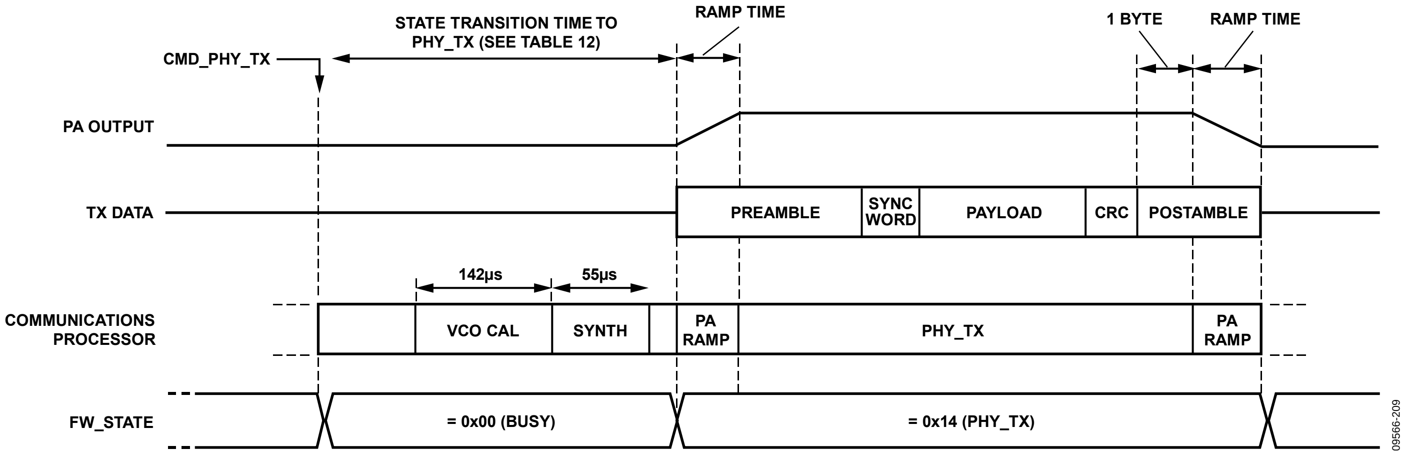


Figure 27. Transmit Packet Timing

DATA WHITENING

Data whitening can be employed to avoid long runs of 1s or 0s in the transmitted data stream. This ensures sufficient bit transitions in the packet, which aids in receiver clock and data recovery because the encoding breaks up long runs of 1s or 0s in the transmit packet. The data, excluding the preamble and sync word, is automatically whitened before transmission by XOR'ing the data with an 8-bit pseudorandom sequence. At the receiver, the data is XOR'ed with the same pseudorandom sequence, thereby reversing the whitening. The linear feedback shift register polynomial used is $x^7 + x^1 + 1$. Data whitening and dewhitening are enabled by setting DATA_WHITENING = 1 in the SYMBOL_MODE register (Address 0x11C).

MANCHESTER ENCODING

Manchester encoding can be used to ensure a dc-free (zero mean) transmission. The encoded over-the-air bit rate (chip rate) is double the rate set by the DATA_RATE variable (Address 0x10C and Address 0x10D). A Binary 0 is mapped to 10, and a Binary 1 is mapped to 01. Manchester encoding and decoding are applied to the payload data and the CRC Manchester encoding and decoding are enabled by setting MANCHESTER_ENC = 1 in the SYMBOL_MODE register (Address 0x11C).

PROCESSOR INTERRUPT GENERATION

The RF transceiver uses a highly flexible, powerful interrupt system with support for MAC level interrupts and PHY level interrupts. To enable an interrupt source, the corresponding mask bit must be set. When an enabled interrupt occurs, the P2.4/IRQ8 pin goes high, and the interrupt bit of the status word is set to Logic 1. The Cortex-M3 processor can use either the P2.4/IRQ8 pin or the status word to check for an interrupt. After an interrupt is asserted, the RF transceiver continues operations unaffected, unless it is directed to do otherwise by the Cortex-M3 processor. An outline of the interrupt source and mask system is shown in Table 76.

MAC interrupts can be enabled by writing a Logic 1 to the relevant bits of the INTERRUPT_MASK_0 register (Address 0x100) and PHY level interrupts by writing a Logic 1 to the relevant bits of the INTERRUPT_MASK_1 register (Address 0x101). The structure of these memory locations is described in Table 76.

In the case of an interrupt condition, the interrupt source can be determined by reading the INTERRUPT_SOURCE_0 register (Address 0x336) and the INTERRUPT_SOURCE_1 register (Address 0x337). The bit that corresponds to the relevant interrupt condition is high. The structure of these two registers is shown in Table 77.

Following an interrupt condition, the Cortex-M3 processor should clear the relevant interrupt flag so that further interrupts assert the P2.4/IRQ8 pin. This is performed by writing a Logic 1 to the bit that is high in either the INTERRUPT_SOURCE_0 or INTERRUPT_SOURCE_1 register. If multiple bits in the interrupt source registers are high, they can be cleared individually or altogether by writing Logic 1 to them. The P2.4/IRQ8 pin goes low when all the interrupt source bits are cleared.

Table 76. Structure of the Interrupt Mask Registers

Register	Bit	Name	Description
INTERRUPT_MASK_0, Address 0x100	7:5	Reserved	Set to 0
	4	INTERRUPT_TX_EOF	Interrupt when a packet has finished transmitting 1: interrupt enabled; 0: interrupt disabled
	3	INTERRUPT_ADDRESS_MATCH	Interrupt when a received packet has a valid address match 1: interrupt enabled; 0: interrupt disabled
	2	INTERRUPT_CRC_CORRECT	Interrupt when a received packet has the correct CRC 1: interrupt enabled; 0: interrupt disabled
	1	INTERRUPT_SYNC_DETECT	Interrupt when a qualified sync word has been detected in the received packet 1: interrupt enabled; 0: interrupt disabled
	0	INTERRUPT_PREAMBLE_DETECT	Interrupt when a qualified preamble has been detected in the received packet 1: interrupt enabled; 0: interrupt disabled
INTERRUPT_MASK_1, Address 0x101	7	Reserved	Set to 0
	6	CMD_READY	Interrupt when the communications processor is ready to load a new command; mirrors the CMD_READY bit of the status word 1: interrupt enabled; 0: interrupt disabled
	5	Reserved	Set to 0
	4	Reserved	Set to 0
	3	Reserved	Set to 0
	2	Reserved	Set to 0
	1	SPI_READY	Interrupt when the SPI is ready for access 1: interrupt enabled; 0: interrupt disabled
	0	CMD_FINISHED	Interrupt when the communications processor has finished performing a command 1: interrupt enabled; 0: interrupt disabled

Table 77. Structure of the Interrupt Source Registers

Register	Bit	Name	Interrupt Description
INTERRUPT_SOURCE_0, Address: 0x336	7	Reserved	
	6	Reserved	
	5	Reserved	
	4	INTERRUPT_TX_EOF	Asserted when a packet has finished transmitting (packet mode only).
	3	INTERRUPT_ADDRESS_MATCH	Asserted when a received packet has a valid address match (packet mode only).
	2	INTERRUPT_CRC_CORRECT	Asserted when a received packet has the correct CRC (packet mode only).
	1	INTERRUPT_SYNC_DETECT	Asserted when a qualified sync word has been detected in the received packet.
	0	INTERRUPT_PREAMBLE_DETECT	Asserted when a qualified preamble has been detected in the received packet.
INTERRUPT_SOURCE_1, Address: 0x337	7		Reserved
	6	CMD_READY	Asserted when the communications processor is ready to load a new command; mirrors the CMD_READY bit of the status word.
	5	Reserved	
	4	Reserved	
	3	Reserved	
	2	Reserved	
	1	SPI_READY	Asserted when the SPI is ready for access.
	0	CMD_FINISHED	Asserted when the communications processor has finished performing a command. If the CMD_FINISHED interrupt is enabled, following the issue of CMD_PHY_TX, the first bit of user data is transmitted 1.5 × T_{BIT} + 2.3 µs following the interrupt. The PA ramp starts 3.4 µs after the interrupt. (T_{BIT} is the time taken to transmit one bit.)

RF TRANSCEIVER MEMORY MAP

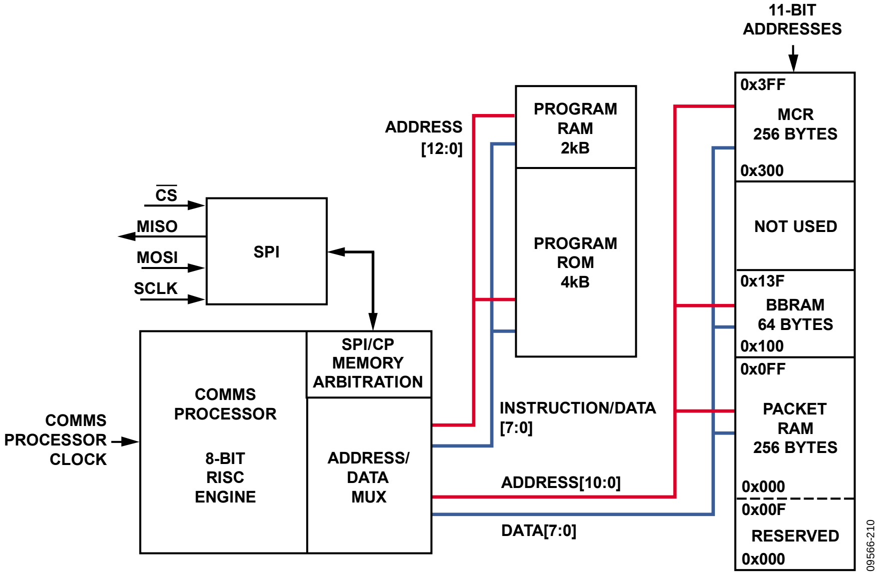


Figure 28. RF Transceiver Memory Map

MEMORY LOCATIONS

This section describes the various memory locations used by the RF transceiver. The radio control and packet management are realized through the use of an integrated RISC processor, which executes instructions stored in the embedded program ROM. There is also a local RAM, subdivided into three sections, that is used as a data packet buffer, both for transmitted and received data (packet RAM), and for storing the radio and packet management configuration (BBRAM and MCR). The RAM addresses of these memory banks are 11 bits long.

BBRAM

The battery backup RAM (BBRAM) contains the main radio and packet management registers used to configure the radio. On application of battery power to the RF transceiver for the first time, the entire BBRAM should be initialized by the Cortex-M3 processor with the appropriate settings. After the BBRAM has been written to, the CMD_CONFIG_DEV command should be issued to update the radio and communications processor with the current BBRAM settings. The CMD_CONFIG_DEV command can be issued in the PHY_OFF state or the PHY_ON state only.

The BBRAM is used to maintain settings needed at wake-up from sleep mode by the wake-up controller. Upon wake-up from sleep, in smart wake mode, the BBRAM contents are read by the on-chip processor to recover the packet management and radio parameters.

Modem Configuration RAM (MCR)

The 256-byte modem configuration RAM (MCR) contains the various registers used for direct control or observation of the physical layer radio blocks of the RF transceiver. The contents of the MCR are not retained in the PHY_SLEEP state.

Program ROM

The program ROM consists of 4 kB of nonvolatile memory. It contains the firmware code for radio control and packet management.

Program RAM

The program RAM consists of 2 kB of volatile memory. This memory space is used for a software module, such as IR calibration. The software module is down-loaded to the program RAM memory space over the SPI by the Cortex-M3 processor. See the Downloadable Firmware Modules section for details on loading a firmware module to program RAM.

Packet RAM

The packet RAM consists of 256 bytes of memory space. The first 16 bytes of this memory space are allocated for use by the on-chip processor. The remaining 240 bytes of this memory space are allocated for storage of data from valid received packets and packet data to be transmitted. The communications processor stores received payload data at the memory location indicated by the value of the RX_BASE_ADR register (Address 0x125), the receive address pointer. The value of the TX_BASE_ADR register (Address 0x124), the transmit address pointer, determines the start address of data to be transmitted by the communications processor. This memory can be arbitrarily assigned to store single or multiple transmit or receive packets, with and without overlap. The RX_BASE_ADR value should be chosen to ensure that there is enough allocated packet RAM space for the maximum receiver payload length.

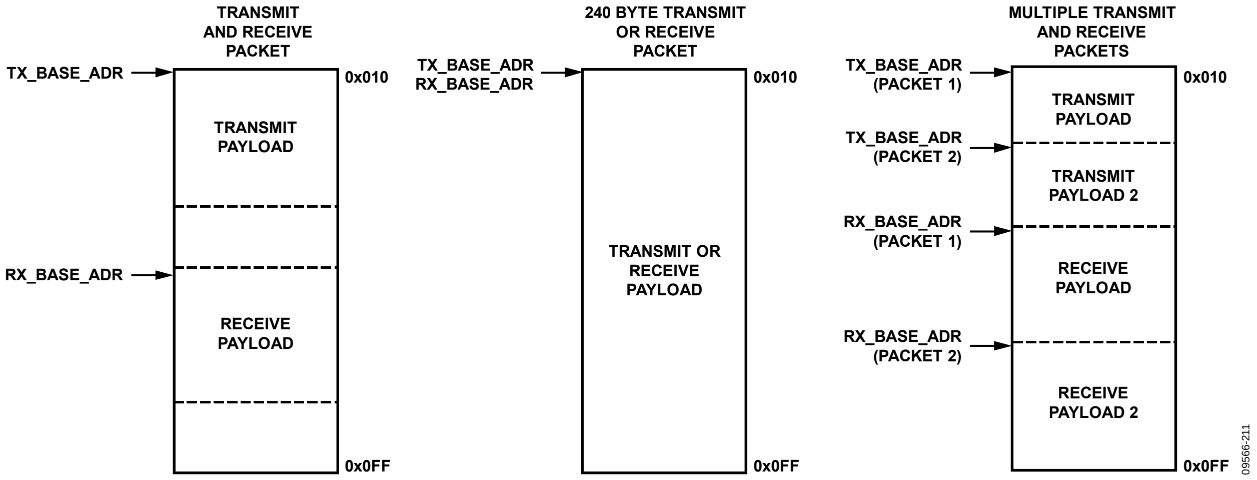


Figure 29. Example Packet RAM Configurations Using the Tx Packet and Rx Packet Address Pointers

SPI INTERFACE

General Characteristics

The RF transceiver is equipped with a 4-wire SPI interface, using the SCLK, MISO, MOSI, and $\overline{\text{CS}}$ line. The RF transceiver always acts as a slave to the Cortex-M3 processor. Figure 30 shows an example connection diagram between the processor and the RF transceiver. The diagram also shows the direction of the signal flow for each pin. The SPI interface is active, and the MISO outputs enabled, only while the $\overline{\text{CS}}$ input is low. The interface uses a word length of eight bits, which is compatible with the SPI hardware of most processors. The data transfer through the SPI interface occurs with the most significant bit first. The MOSI input is sampled at the rising edge of SCLK. As commands or data are shifted in from the MOSI input at the SCLK rising edge, the status word or data is shifted out at the MISO pin synchronous with the SCLK clock falling edge. If $\overline{\text{CS}}$ is brought low, the most significant bit of the status word appears on the MISO output without the need for a rising clock edge on the SCLK input.

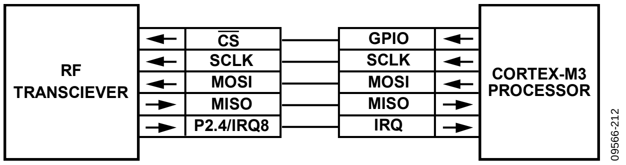


Figure 30. SPI Interface Connections

Command Access

The RF transceiver is controlled through commands. Command words are single octet instructions that control the state transitions of the communications processor and access to the registers and packet RAM. The complete list of valid commands is given in the Command Reference section. Commands that have a CMD prefix are handled by the communications processor. Memory access commands have an SPI prefix and are handled by an independent controller. Thus, SPI commands can be issued independent of the state of the communications processor.

A command is initiated by bringing $\overline{\text{CS}}$ low and shifting in the command word over the SPI, as shown in Figure 31. All commands are executed on the last positive SCLK edge of the command. The $\overline{\text{CS}}$ input must be brought high again after a command has been shifted into the RF transceiver to enable the recognition of successive command words. This is because a single command can be issued only during a $\overline{\text{CS}}$ low period (with the exception of a double NOP command).

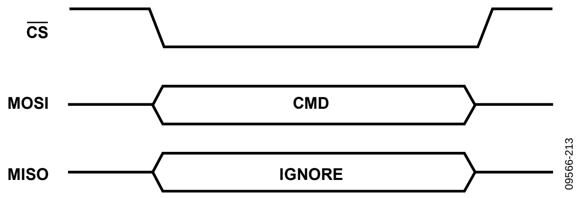


Figure 31. Command Write (No Parameters)

Status Word

The status word of the RF transceiver is automatically returned over the MISO each time a byte is transferred over the MOSI. Shifting in double SPI_NOP commands (see Table 80) causes the status word to be shifted out as shown in Figure 32. The meaning of the various bit fields is illustrated in Table 78. The FW_STATE variable can be used to read the current state of the communications processor and is described in Table 79. If it is busy performing an action or state transition, FW_STATE is busy. The FW_STATE variable also indicates the current state of the radio.

The SPI_READY variable is used to indicate when the SPI is ready for access. The CMD_READY variable is used to indicate when the communications processor is ready to accept a new command. The status word should be polled and the CMD_READY bit examined before issuing a command to ensure that the communications processor is ready to accept a new command. It is not necessary to check the CMD_READY bit before issuing a SPI memory access command. It is possible to queue one command while the communications processor is busy. This is discussed in the Command Queuing section.

The RF transceiver interrupt handler can be also be configured to generate an interrupt signal on P2.4/IRQ8 when the communications processor is ready to accept a new command (CMD_ READY in the INTERRUPT_SOURCE_1 register (Address 0x337)) or when it has finished processing a command (CMD_FINISHED in the INTERRUPT_SOURCE_1 register (Address 0x337)).

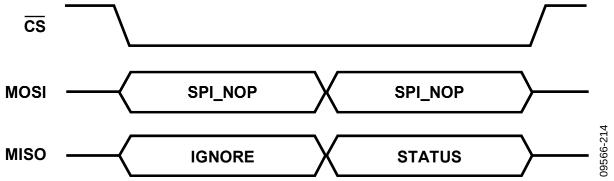


Figure 32. Reading the Status Word Using a Double SPI_NOP Command

Table 78. Status Word

Bit	Name	Description
[7]	SPI_READY	0: SPI is not ready for access. 1: SPI is ready for access.
[6]	IRQ_STATUS	0: no pending interrupt condition. 1: pending interrupt condition (mirrors the P2.4/IRQ8 pin).
[5]	CMD_READY	0: the radio controller is not ready to receive a radio controller command. 1: the radio controller is ready to receive a radio controller command.
[4:0]	FW_STATE	Indicates the RF transceiver state (see Table 79).

Table 79. FW_STATE Description

Value	State
0x0F	Initializing
0x00	Busy, performing a state transition
0x11	PHY_OFF
0x12	PHY_ON
0x13	PHY_RX
0x14	PHY_TX
0x06	PHY_SLEEP
0x05	Performing CMD_GET_RSSI
0x07	Performing CMD_IR_CAL

Command Queuing

The CMD_READY status bit is used to indicate that the command queue used by the communications processor is empty. The queue is one command deep. The FW_STATE bit is used to indicate the state of the communications processor. The operation of the status word and these bits is illustrated in Figure 33 when a CMD_PHY_ON command is issued in the PHY_OFF state.

Operation of the status word when a command is being queued is illustrated in Figure 34 when a CMD_PHY_ON command is issued in the PHY_OFF state followed quickly by a CMD_ PHY_RX command. The CMD_PHY_RX command is issued while FW_STATE is busy (that is, transitioning between the PHY_OFF and PHY_ON states) but the CMD_READY bit is high, indicating that the command queue is empty. After the CMD_PHY_RX command is issued, the CMD_READY bit transitions to a logic low, indicating that the command queue is full. After the PHY_OFF to PHY_ON transition is finished, the PHY_RX command is processed immediately by the communications processor, and the CMD_READY bit goes high, indicating that the command queue is empty and another command can be issued.

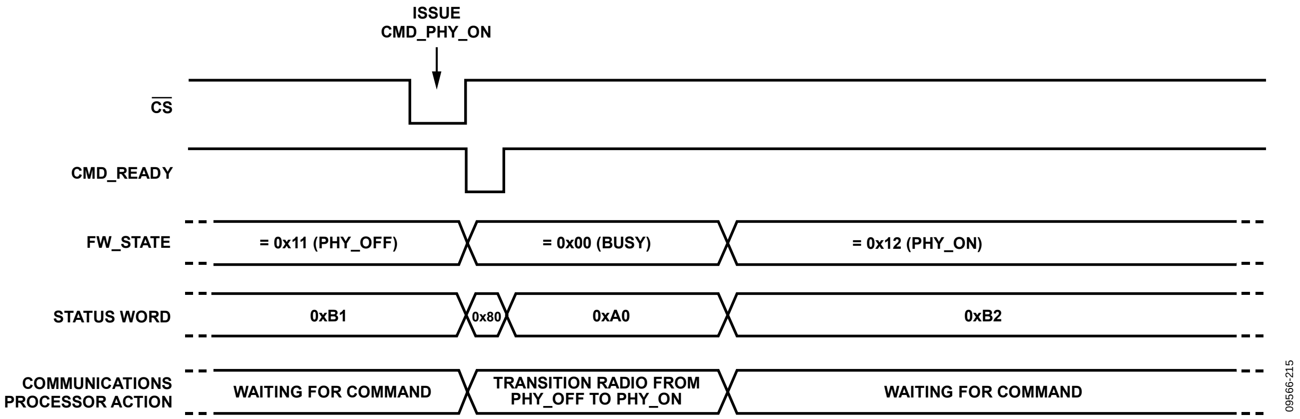


Figure 33. Operation of the CMD_READY and FW_STATE Bits in Transitioning the RF Transceiver from the PHY_OFF State to the PHY_ON State

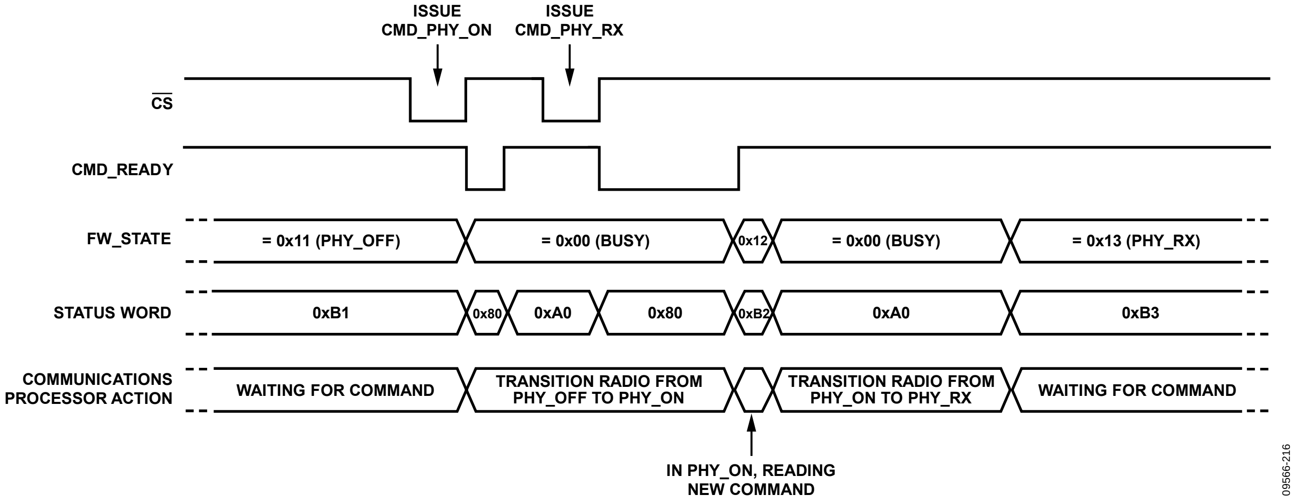


Figure 34. Command Queuing and Operation of the CMD_READY and FW_STATE Bits in Transitioning the RF Transceiver from the PHY_OFF State to the PHY_ON State and Then to the PHY_RX State

MEMORY ACCESS

Memory locations are accessed by invoking the relevant SPI command. An 11-bit address is used to identify registers or locations in the memory space. The most significant three bits of the address are incorporated into the SPI command by appending them as the LSBs of the command word. Figure 35 illustrates command, address, and data partitioning. The various SPI memory access commands are different, depending on the memory location being accessed (see Table 80).

An SPI command should be issued only if the SPI_READY bit in the INTERRUPT_SOURCE_1 register (Address 0x337) of the status word bit is high. The RF transceiver interrupt handler can be also be configured to generate an interrupt signal on P2.4/IRQ8 when the SPI_READY bit is high.

An SPI command should not be issued while the communications processor is initializing (FW_STATE = 0x0F). SPI commands can be issued in any other communications processor state, including the busy state (FW_STATE = 0x00). This allows the RF transceiver memory to be accessed while the radio is transitioning between states.

Block Write

MCR, BBRAM, and packet RAM memory locations can be written to in block format using the SPI_MEM_WR command. The SPI_MEM_WR command code is 00011xxxb, where xxxb represent Bits[10:8] of the first 11-bit address. If more than one data byte is written, the write address is automatically incremented for every byte sent until $\overline{\text{CS}}$ is set high, which terminates the memory access command (see Figure 36 for more details). The maximum block write for the MCR, packet RAM, and BBRAM memories is 256 bytes, 256 bytes, and 64 bytes, respectively. These maximum block-write lengths should not be exceeded.

Example

Write 0x3F to the PA_LEVEL_MCR register (Address 0x307).

- The first five bits of the SPI_MEM_WR command are 00011.
- The 11-bit address of PA_LEVEL_MCR is 01100000111.
- The first byte sent is 00011011 or 0x1B.
- The second byte sent is 00000111 or 0x07.
- The third byte sent is 0x3F.

Thus, 0x1B, 0x07, 0x3F is written to the part.

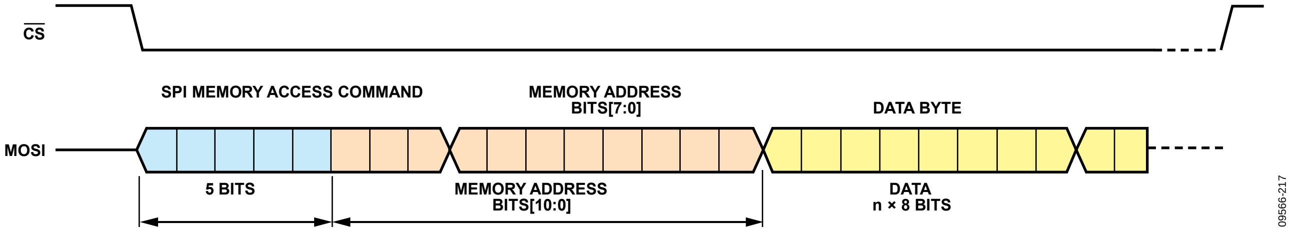


Figure 35. SPI Memory Access Command/Address Format

Table 80. Summary of SPI Memory Access Commands

SPI Command	Command Value	Description
SPI_MEM_WR	0x18 (packet RAM) 0x19 (BBRAM) 0x1B (MCR) 0x1E (program RAM)	Write data to BBRAM, MCR, or packet RAM sequentially. An 11-bit address is used to identify memory locations. The most significant three bits of the address are incorporated into the command (xxxb). This command is followed by the remaining eight bits of the address.
SPI_MEM_RD	0x38 (packet RAM) 0x39 (BBRAM) 0x3B (MCR)	Read data from BBRAM, MCR, or packet RAM sequentially. An 11-bit address is used to identify memory locations. The most significant three bits of the address are incorporated into the command (xxxb). This command is followed by the remaining eight bits of the address, which is subsequently followed by the appropriate number of SPI_NOP commands.
SPI_MEMR_WR	0x08 (packet RAM) 0x09 (BBRAM) 0x0B (MCR)	Write data to BBRAM, MCR, or packet RAM nonsequentially.
SPI_MEMR_RD	0x28 (packet RAM) 0x29 (BBRAM) 0x2B (MCR)	Read data from BBRAM, MCR, or packet RAM nonsequentially.
SPI_NOP	0xFF	No operation. Use for dummy writes when polling the status word. Also used as dummy data on the MOSI line when performing a memory read.

Random Address Write

MCR, BBRAM, and packet RAM memory locations can be written to in a nonsequential manner using the SPI_MEMR_WR command. The SPI_MEMR_WR command code is 00001xxxb, where xxxb represent Bits[10:8] of the 11-bit address. The lower eight bits of the address should follow this command and then the data byte to be written to the address. The lower eight bits of the next address are entered, followed by the data for that address until all required addresses within that block are written, as shown in Figure 37.

Program RAM Write

The program RAM can be written to only by using the memory block write, as illustrated in Figure 36. SPI_MEM_WR should be set to 0x1E. See the Downloadable Firmware Modules section for details on loading a firmware module to program RAM.

Block Read

MCR, BBRAM, and packet RAM memory locations can be read from in block format using the SPI_MEM_RD command. The SPI_MEM_RD command code is 00111xxxb, where xxxb represent Bits[10:8] of the first 11-bit address. This command is followed by the remaining eight bits of the address to be read and then two SPI_NOP commands (dummy byte). The first byte available after writing the address should be ignored, with the second byte constituting valid data. If more than one data byte is to be read, the write address is automatically incremented for subsequent SPI_NOP commands sent. See Figure 38 for more details.

Random Address Read

MCR, BBRAM, and packet RAM memory locations can be read from memory in a nonsequential manner using the SPI_MEMR_RD command. The SPI_MEMR_RD command code is 00101xxxb, where xxxb represent Bits[10:8] of the 11-bit address. This command is followed by the remaining eight bits of the address to be written. Each subsequent address byte is then written. The last address byte to be written should be followed by two SPI_NOP commands, as shown in Figure 39. The data bytes from memory, starting at the first address location, are available after the second status byte.

Example

Read the value stored in the PA_LEVEL_MCR register.

- The first five bits of the SPI_MEM_RD command are 00111.
- The 11-bit address of PA_LEVEL_MCR is 01100000111.
- The first byte sent is 00111011 or 0x3B.
- The second byte sent is 00000111 or 0x07.
- The third byte sent is 0xFF (SPI_NOP).
- The fourth byte sent is 0xFF.

Thus, 0x3B07FFFF is written to the part.

The value shifted out on the MISO line while the fourth byte is sent is the value stored in the PA_LEVEL_MCR register.

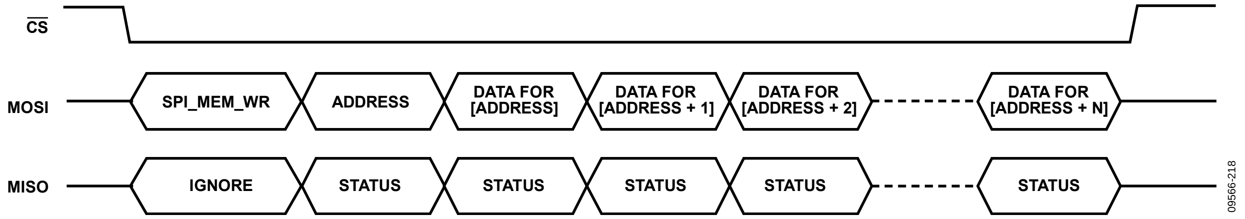


Figure 36. Memory (MCR, BBRAM, or Packet RAM) Block Write

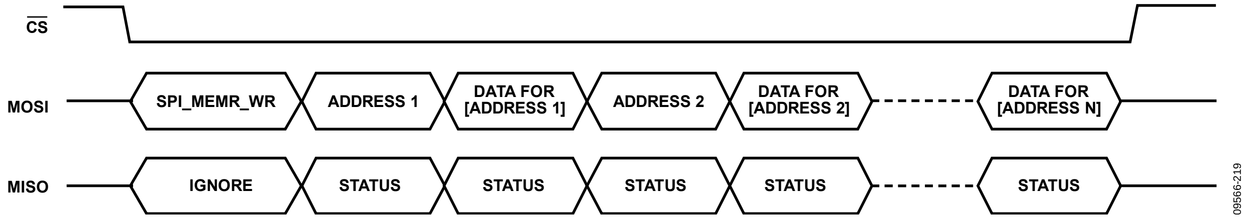


Figure 37. Memory (MCR, BBRAM, or Packet RAM) Random Address Write

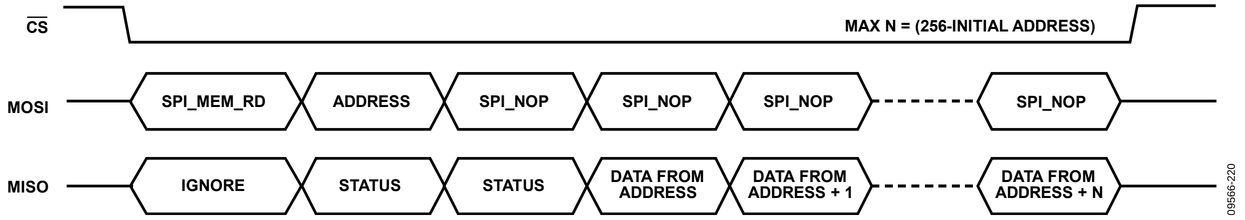


Figure 38. Memory (MCR, BBRAM, or Packet RAM) Block Read

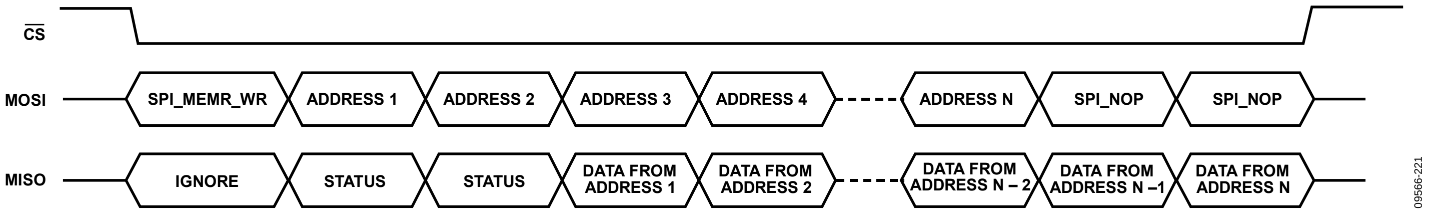


Figure 39. Memory (MCR, BBRAM, or Packet RAM) Random Address Read

LOW POWER MODES

There are two low power modes available on the RF transceiver.

Deep Sleep Mode (Powered down)

Deep Sleep Mode 2 is suitable for applications where the Cortex-M3 processor controls the low power mode timing and the lowest possible RF transceiver sleep current is required.

In this low power mode, the RF transceiver is in the PHY_SLEEP state. The BBRAM contents are not retained. This low power mode is entered by issuing the CMD_HW_RESET command from any radio state. To wake the part from the PHY_SLEEP state, the $\overline{\text{CS}}$ line should be set low. The initialization routine after a CMD_HW_RESET command should be followed as detailed in the Radio Control section.

Deep Sleep Mode (State Retained)

Deep Sleep Mode 1 is suitable for applications where the Cortex-M3 processor controls the low power mode timing and the RF transceiver configuration is retained during the PHY_SLEEP state.

In this low power mode, the RF transceiver is in the PHY_SLEEP state with the BBRAM contents retained. Before entering the PHY_SLEEP state, set WUC_BBRAM_EN (Address 0x30D) to 1 to ensure that the BBRAM is retained. This low power mode is entered by issuing the CMD_PHY_SLEEP command from either the PHY_OFF or PHY_ON state. To exit the PHY_SLEEP state, the $\overline{\text{CS}}$ line can be set low. Then, follow the $\overline{\text{CS}}$ low initialization routine, as detailed in the Radio Control section.

Exiting Low Power Mode

The RF transceiver waits for a host command on any of the termination conditions of the low power mode. It is also possible to perform an asynchronous exit from low power mode using the following procedure:

1. Bring the $\overline{\text{CS}}$ pin of the SPI low and wait until the MISO output goes high.
2. Issue a CMD_HW_RESET command.

The Cortex-M3 processor should then follow the initialization procedure after a CMD_HW_RESET command, as described in the Initialization section.

DOWNLOADABLE FIRMWARE MODULES

The program RAM memory of the RF transceiver can be used to store a firmware module for the communications processor to calibrate the image rejection for additional rejection over the default values. This functionality is available through the radio interface API.

Writing a Module to Program RAM

The sequence to write a firmware module to program RAM is as follows:

1. Ensure that the RF transceiver is in PHY_OFF.
2. Issue the CMD_RAM_LOAD_INIT command.
3. Write the module to program RAM using an SPI memory block write (see the SPI Interface section).
4. Issue the CMD_RAM_LOAD_DONE command.
5. The firmware module is now stored on program RAM.

Image Rejection Calibration Module

The calibration system initially disables the RF transceiver receiver, and an internal RF source is applied to the RF input at the image frequency. The algorithm then maximizes the receiver image rejection performance by iteratively minimizing the quadrature gain and phase errors in the polyphase filter.

The calibration algorithm takes its initial estimates for quadrature phase correction (Address 0x118) and quadrature gain correction (Address 0x119) from BBRAM. After calibration, new optimum values of phase and gain are loaded back into these locations. These calibration values are maintained in BBRAM during sleep mode and are automatically reapplied from a wake-up event, which keeps the number of calibrations required to a minimum.

Depending on the initial values of quadrature gain and phase correction, the calibration algorithm can take approximately 20 ms to find the optimum image rejection performance. However, the calibration time can be significantly less than this when the seed values used for gain and phase corrections are close to optimum.

The image rejection performance is also dependent on temperature. To maintain optimum image rejection performance, a calibration should be activated whenever a temperature change of more than 10°C occurs.

RADIO BLOCKS

Frequency Synthesizer

A fully integrated RF frequency synthesizer is used to generate both the transmit signal and the receiver's local oscillator (LO) signal. The architecture of the frequency synthesizer is shown in Figure 40.

The receiver uses a fractional-N frequency synthesizer to generate the mixer's LO for down conversion to the intermediate frequency (IF) of 200 kHz or 300 kHz. In transmit mode, a high resolution sigma-delta (Σ-Δ) modulator is used to generate the required frequency deviations at the RF output when FSK data is transmitted. To reduce the occupied FSK bandwidth, the transmitted bit stream can be filtered using a digital Gaussian filter, which is enabled via the RADIO_CFG_9 register (Address 0x115). The Gaussian filter uses a bandwidth time (BT) of 0.5.

The VCO and the PLL loop filter of the RF transceiver are fully integrated. To reduce the effect of pulling of the VCO by the power-up of the PA and to minimize spurious emissions, the VCO operates at twice or four times the RF frequency. The VCO signal is then divided by 2 or 4, giving the required frequency for the transmitter and the required LO frequency for the receiver.

A high speed, fully automatic calibration scheme is used to ensure that the frequency and amplitude characteristic of the VCO are maintained over temperature, supply voltage, and process variations.

The calibration is automatically performed when the CMD_PHY_RX or CMD_PHY_TX command is issued. The calibration duration is 142 μs, and if required, the CALIBRATION_STATUS register (Address 0x339) can be polled to indicate the completion of the VCO self-calibration. After the VCO is calibrated, the frequency synthesizer settles to within ±5 ppm of the target frequency in 56 μs.

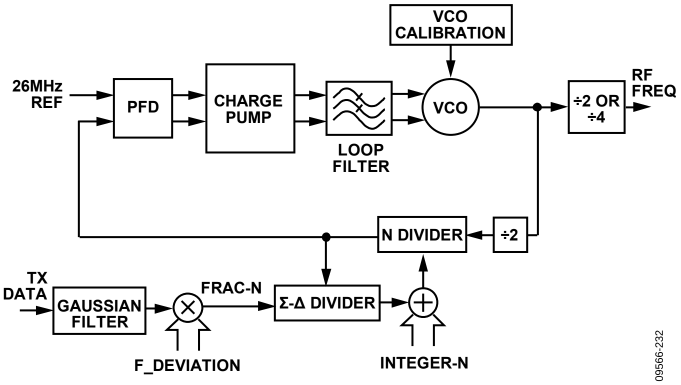


Figure 40. RF Frequency Synthesizer Architecture

Synthesizer Bandwidth

The synthesizer loop filter is fully integrated on chip and has a programmable bandwidth. The communications processor automatically sets the bandwidth of the synthesizer when the device enters PHY_TX or PHY_RX state. On entering the PHY_TX state, the communications processor chooses the bandwidth based on the programmed modulation scheme (2FSK, GFSK) and the data rate. This ensures optimum modulation quality for each data rate. On entering the PHY_RX state, the communications processor sets a narrow bandwidth to ensure best receiver rejection. In all, there are eight bandwidth configurations. Each synthesizer bandwidth setting is described in Table 81.

Table 81. Automatic Synthesizer Bandwidth Selections

Description	Data Rate (kbps)	Closed Loop Synthesizer Bandwidth (kHz)
Rx 2FSK/GFSK/MSK/GMSK	All	92
Tx 2FSK/GFSK/MSK/GMSK	1 to 49.5	130
Tx 2FSK/GFSK/MSK/GMSK	49.6 to 99.1	174
Tx 2FSK/GFSK/MSK/GMSK	99.2 to 129.5	174
Tx 2FSK/GFSK/MSK/GMSK	129.6 to 179.1	226
Tx 2FSK/GFSK/MSK/GMSK	179.2 to 239.9	305
Tx 2FSK/GFSK/MSK/GMSK	240 to 300	382

Synthesizer Settling

After the VCO calibration, a 56 μs delay is allowed for synthesizer settling. This delay is fixed at 56 μs by default and ensures that the synthesizer has fully settled when using any of the default synthesizer bandwidths. However, in some cases, it may be necessary to use a custom synthesizer settling delay.

Crystal Oscillator

A 26 MHz crystal oscillator operating in parallel mode must be connected between the XOSC26P and XOSC26N pins. Two parallel loading capacitors are required for oscillation at the correct frequency. Their values are dependent upon the crystal specification. They should be chosen to ensure that the shunt value of capacitance added to the PCB track capacitance and the input pin capacitance of the RF transceiver equals the specified load capacitance of the crystal, usually 10 pF to 20 pF. Track capacitance values vary from 2 pF to 5 pF, depending on board layout. The total load capacitance is described by

$$C_{LOAD} = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2}} + \frac{C_{PIN}}{2} + C_{PCB}$$

where:
C_{LOAD} is the total load capacitance.
$C1$ and $C2$ are the external crystal load capacitors.
C_{PIN} is the RF transceiver input capacitance of the XOSC26P and XOSC26N pins and is equal to 2.1 pF.
C_{PCB} is the PCB track capacitance.

When possible, choose capacitors that have a very low temperature coefficient to ensure stable frequency operation over all conditions.

The crystal frequency error can be corrected by means of an integrated digital tuning varactor. For a typical crystal load capacitance of 10 pF, a tuning range of +15 ppm to −11.25 ppm is available via programming of a 3-bit DAC, according to Table 82. The 3-bit value should be written to XOSC_CAP_DAC in the OSC_CONFIG register (Address 0x3D2).

Alternatively, any error in the RF frequency due to crystal error can be adjusted for by offsetting the RF channel frequency using the RF channel frequency setting in BBRAM memory.

Table 82. Crystal Frequency Pulling Programming

XOSC_CAP_DAC	Pulling (ppm)
000	+15
001	+11.25
010	+7.5
011	+3.75
100	0
101	−3.75
110	−7.5
111	−11.25

Modulation

The RF transceiver supports binary frequency shift keying (2FSK), minimum shift keying (MSK), binary level Gaussian filtered 2FSK (GFSK), Gaussian filtered MSK (GMSK), The desired transmit and receive modulation formats are set in the RADIO_CFG_9 register (Address 0x115).

When using 2FSK/GFSK/MSK/GMSK modulation, the frequency deviation can be set using the FREQ_DEVIATION[11:0] parameter in the RADIO_CFG_1 register (Address 0x10D) and RADIO_CFG_2 register (Address 0x10E). The data rate can be set using the DATA_RATE[11:0] parameter in the RADIO_CFG_0 register (Address 0x10C) and RADIO_CFG_1 register (Address 0x10D). For GFSK/GMSK modulation, the Gaussian filter uses a fixed bandwidth time (BT) product of 0.5.

RF Output Stage

Power Amplifier (PA)

The RF transceiver PA can be configured for single-ended or differential output operation using the PA_SINGLE_DIFF_SEL bit in the RADIO_CFG_8 register (Address 0x114). The PA level is set by the PA_LEVEL bit in the RADIO_CFG_8 register and has a range of 0 to 15. For finer control of the output power level, the PA_LEVEL_MCR register (Address 0x307) can be used. It offers more resolution with a setting range of 0 to 63. The relationship between the PA_LEVEL and PA_LEVEL_MCR settings is given by

$$PA\_LEVEL\_MCR = 4 \times PA\_LEVEL + 3$$

The single-ended configuration can deliver 13.0 dBm output power. The differential PA can deliver 10 dBm output power and allows a straightforward interface to dipole antennae. The two PA configurations offer a Tx antenna diversity capability. Note that the two PAs cannot be enabled at the same time.

Automatic PA Ramp

The RF transceiver has built-in up and down PA ramping for both single-ended and differential PAs. There are eight ramp rate settings, with the ramp rate defined as a certain number of PA power level settings per data bit period. The PA_RAMP variable in the RADIO_CFG_8 register (Address 0x114) sets this PA ramp rate, as illustrated in Figure 41.

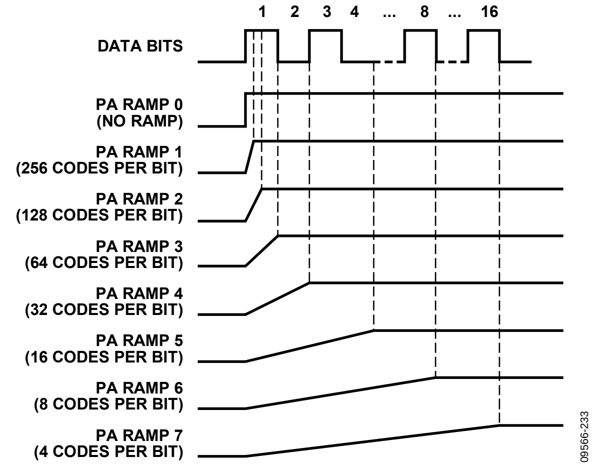


Figure 41. PA Ramp for Different PA_RAMP Settings

The PA ramps to the level set by the PA_LEVEL or PA_LEVEL_ MCR settings. Enabling the PA ramp reduces spectral splatter and helps meet radio regulations (for example, the ETSI EN 300 220 standard), which limit PA transient spurious emissions. To ensure optimum performance, an adequately long PA ramp rate is required based on the data rate and the PA output power setting. The PA_RAMP setting should, therefore, be set such that

$$Ramp\ Rate\ (Codes/Bit) \leq 10{,}000 \times \frac{PA\_LEVEL\_M\ \ CR[5{:}0]}{DATA\_RATE\ \ [11{:}0]}$$

where *PA_LEVEL_MCR* is related to the PA_LEVEL setting by *PA_LEVEL_MCR* = 4 × PA_LEVEL + 3.

PA/LNA Interface

The RF transceiver supports both single-ended and differential PA outputs. Only one PA can be active at one time. The differential PA and LNA share the same pins, RFIO_1P and RFIO_1N, which facilitate a simpler antenna interface. The single-ended PA output is available on the RFO2 pin. A number of PA/LNA antenna matching options are possible and are described in the PA/LNA section.

Receive Channel Filter

The receiver's channel filter is a fourth order, active polyphase Butterworth filter with programmable bandwidths of 100 kHz, 150 kHz, 200 kHz, and 300 kHz. The fourth order filter gives very good interference suppression of adjacent and neighboring channels and also suppresses the image channel by approximately 36 dB at a 100 kHz IF bandwidth and an RF frequency of 868 MHz or 915 MHz.

For channel bandwidths of 100 kHz to 200 kHz, an IF frequency of 200 kHz is used, which results in an image frequency located 400 kHz below the wanted RF frequency. When the 300 kHz bandwidth is selected, an IF frequency of 300 kHz is used, and the image frequency is located at 600 kHz below the wanted frequency.

The bandwidth and center frequency of the IF filter are calibrated automatically after entering the PHY_ON state. The filter calibration time takes 100 µs.

The IF bandwidth is programmed by setting the IFBW field in the RADIO_CFG_9 register (Address 0x115). The filter's pass band is centered at an IF frequency of 200 kHz when bandwidths of 100 kHz to 200 kHz are used and centered at 300 kHz when an IF bandwidth of 300 kHz is used.

Image Channel Rejection

The RF transceiver is capable of providing improved receiver image rejection performance by the use of a fully integrated image rejection calibration system under the control of the on-chip communications processor. To operate the calibration system, a firmware module is downloaded to the on-chip program RAM. The firmware download is supplied by Analog Devices and described in the Downloadable Firmware Modules section.

To achieve the typical uncalibrated image attenuation values given in the Specifications section of the datasheet, it is required to use recommended default values for IMAGE_REJECT_CAL_PHASE (Address 0x118) and IMAGE_REJECT_CAL_AMPLITUDE (Address 0x119).

To achieve the specified uncalibrated image attenuation at 433 MHz, set IMAGE_REJECT_CAL_AMPLITUDE = 0x03 and IMAGE_REJECT_CAL_PHASE = 0x08.

To achieve the specified uncalibrated image attenuation at 868 MHz/915 MHz, set IMAGE_REJECT_CAL_AMPLITUDE = 0x07 and IMAGE_REJECT_CAL_PHASE = 0x16.

Automatic Gain Control (AGC)

AGC is enabled by default, and keeps the receiver gain at the correct level by selecting the LNA, mixer, and filter gain settings based on the measured RSSI level. The LNA has three gain levels, the mixer has gain two levels, and the filter has three gain levels. In all, there are six AGC stages, which are defined in Table 83.

Table 83. AGC Gain Modes

Gain Mode	LNA Gain	Mixer Gain	Filter Gain
1	High	High	High
2	High	Low	High
3	Medium	Low	High
4	Low	Low	High
5	Low	Low	Medium
6	Low	Low	Low

The AGC remains at each gain stage for a time defined by the AGC_CLOCK_DIVIDE register (Address 0x32F). The default value of AGC_CLOCK_DIVIDE = 0x28 gives an AGC delay of 25 µs. When the RSSI is above AGC_HIGH_THRESHOLD (Address 0x35F), the gain is reduced. When the RSSI is below AGC_LOW_THRESHOLD (Address 0x35E), the gain is increased.

The AGC can be configured to remain active while in the PHY_RX state or can be locked on preamble detection. The AGC can also be set to manual mode, in which case the Cortex-M3 processor must set the LNA, filter, and mixer gains by writing to the AGC_MODE register (Address 0x35D). The AGC operation is set by the AGC_LOCK_MODE setting in the RADIO_CFG_7 register (Address 0x113) and is described in Table 84.

The LNA, filter, and mixer gains can be read back through the AGC_GAIN_STATUS register (Address 0x360).

Table 84. AGC Operation

AGC_LOCK_MODE Bits in RADIO_CFG_7 Register	Description
0	AGC is free running.
1	AGC is disabled. Gains must be set manually.
2	AGC is held at the current gain level.
3	AGC is locked on preamble detection.

RSSI

The RSSI is based on a successive compression, log-amp architecture following the analog channel filter. The analog RSSI level is digitized by an 8-bit SAR ADC for user readback and for use by the digital AGC controller.

The RF transceiver has a total of four RSSI measurement functions that support a wide range of applications. These functions can be used to implement carrier sense (CS) or clear channel assessment (CCA). In packet mode, the RSSI is automatically recorded in MCR memory and is available for user readback after receipt of a packet.

Table 86 details the four RSSI measurement methods.

RSSI Method 1

When a valid packet is received in packet mode, the RSSI level during postamble is automatically loaded to the RSSI_READBACK register (Address 0x312) by the communications processor. The RSSI_READBACK register contains a twos complement value and can be converted to input power in dBm using

$$RSSI(dBm) = RSSI\_READBACK - 107$$

To extend the linear range of RSSI measurement down to an input power of −110 dBm, a cosine adjustment can be applied using the following formula:

$$RSSI(dBm) = COS\left(\frac{8}{RSSI\_READBACK}\right) \times RSSI\_READBACK - 106$$

where *COS(X)* is the cosine of Angle X (radians).

RSSI Method 2

The CMD_GET_RSSI command can be used from the PHY_ON state to read the RSSI. This RSSI measurement method uses additional low pass filtering, resulting in a more accurate RSSI reading. The RSSI result is loaded to the RSSI_READBACK register (Address 0x312) by the communications processor. The RSSI_READBACK register contains a twos complement value and can be converted to input power in dBm using the following formula:

$$RSSI(dBm) = RSSI\_READBACK - 107$$

RSSI Method 3

This method supports the measurement of RSSI by the Cortex-M3 processor at any time while in the PHY_RX state. The receiver input power can be calculated using the following procedure:

1. Set AGC to hold by setting the AGC_MODE register (Address 0x35D) = 0x40 (only necessary if AGC has not been locked on the preamble or sync word).
2. Read back the AGC gain settings (AGC_GAIN_STATUS register, Address 0x360).
3. Read the ADC_READBACK[7:0] value (Address 0x327 and Address 0x328).
4. Re-enable the AGC by setting the AGC_MODE register (Address 0x35D) = 0x00 (only necessary if AGC has not already been locked on the preamble or sync word).
5. Calculate the RSSI in dBm as follows:

$$RSSI(dBm) =$$

$$\left(ADC\_READBACK[7:0] \times \frac{1}{7} + Gain\_Correction \right) - 109$$

where *Gain_Correction* is determined by the value of the AGC_GAIN_STATUS register (Address 0x360).

Table 85. Gain Mode Correction for 2FSK/GFSK/MSK/GMSK RSSI

AGC_GAIN_STATUS (Address 0x360)	GAIN_CORRECTION
0x00	44
0x01	35
0x02	26
0x0A	17
0x12	10
0x16	0

To simplify the RSSI calculation, the following approximation can be used by the Cortex-M3 processor:

$$\frac{1}{7} \approx \frac{1}{8}\left(1 + \frac{1}{8} + \frac{1}{64}\right)$$

Table 86. Summary of RSSI Measurement Methods

RSSI Method	RSSI Type	Modulation	Description
1	Automatic end of packet RSSI	2FSK/GFSK/ MSK/GMSK	Automatic RSSI measurement during reception of the postamble in packet mode. The RSSI result is available in the RSSI_READBACK register (Address 0x312).
2	CMD_GET_RSSI command from PHY_ON	2FSK/GFSK/ MSK/GMSK	Automatic RSSI measurement from PHY_ON using CMD_GET_RSSI. The RSSI result is available in the RSSI_READBACK register (Address 0x312).
3	RSSI via ADC and AGC readback, FSK	2FSK/GFSK/ MSK/GMSK	RSSI measurement based on the ADC and AGC gain readbacks. The Cortex-M3 processor calculates RSSI in dBm.

2FSK/GFSK/MSK/GMSK DEMODULATION

A correlator demodulator is used for 2FSK, GFSK, MSK, and GMSK demodulation. The quadrature outputs of the IF filter are first limited and then fed to a digital frequency correlator that performs filtering and frequency discrimination of the 2FSK/GFSK/MSK/ GMSK spectrum. Data is recovered by comparing the output levels from two correlators.

The performance of this frequency discriminator approximates that of a matched filter detector, which is known to provide optimum detection in the presence of additive white gaussian noise (AWGN). This method of 2FSK/GFSK/MSK/GMSK demodulation provides approximately 3 dB to 4 dB better sensitivity than a linear frequency discriminator. The 2FSK/GFSK/MSK/GMSK demodulator architecture is shown in Figure 42. The RF transceiver is configured for 2FSK/GFSK/MSK/ GMSK demodulation by setting DEMOD_SCHEME = 0 in the RADIO_CFG_9 register (Address 0x115).

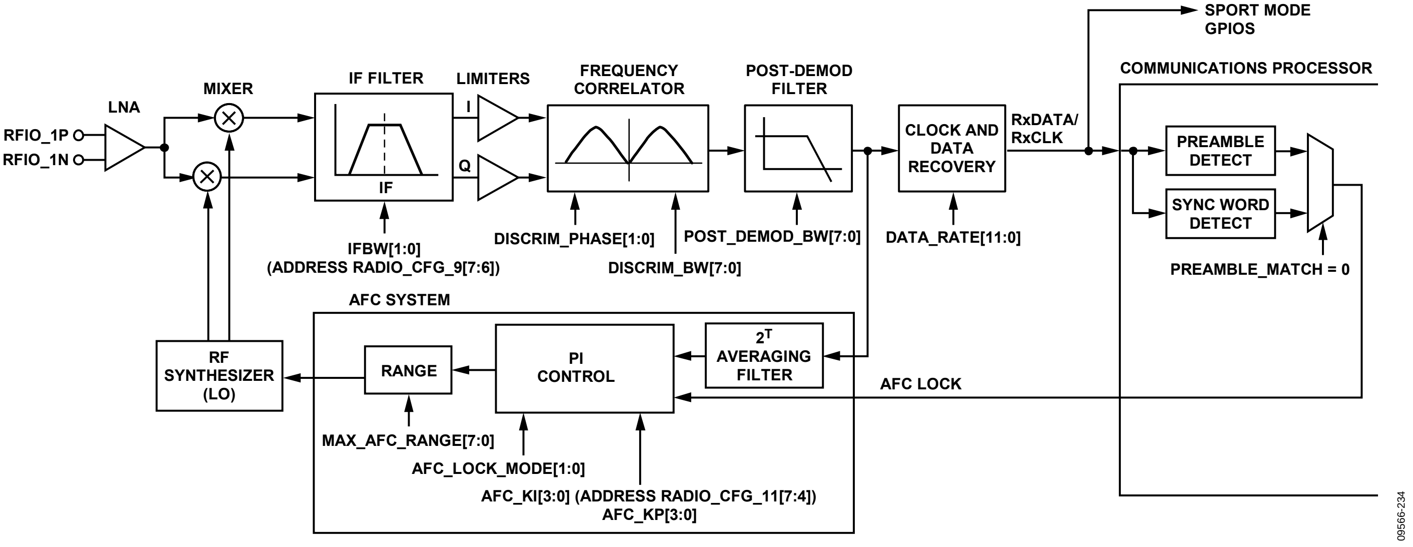


Figure 42. 2FSK/GFSK/MSK/GMSK Demodulation and AFC Architecture

AFC

The RF transceiver features an internal real-time automatic frequency control loop. In receive, the control loop automatically monitors the frequency error during the packet preamble sequence and adjusts the receiver synthesizer local oscillator using proportional integral (PI) control. The AFC frequency error measurement bandwidth is targeted specifically at the packet preamble sequence (dc free). AFC is supported during 2FSK/GFSK/MSK/GMSK demodulation.

AFC can be configured to lock on detection of the qualified preamble or on detection of the qualified sync word. To lock AFC on detection of the qualified preamble, set AFC_LOCK_MODE = 3 (Address 0x116) and ensure that preamble detection is enabled in the PREAMBLE_MATCH register (Address 0x11B). AFC lock is released if the sync word is not detected immediately after the end of the preamble. In packet mode, if the qualified preamble is followed by a qualified sync word, the AFC lock is maintained for the duration of the packet.

To lock AFC on detection of the qualified sync word, set AFC_LOCK_MODE = 3 and ensure that preamble detection is disabled in the PREAMBLE_MATCH register (Address 0x11B). If this mode is selected, consideration must be given to the selection of the sync word. The sync word should be dc free and have short run lengths yet low correlation with the preamble sequence. See the sync word description in the Packet Mode section for further details. After lock on detection of the qualified sync word, the AFC lock is maintained for the duration of the packet. AFC is enabled by setting AFC_LOCK_MODE in the RADIO_CFG_10 register (Address 0x116), as described in Table 87.

Table 87. AFC Mode

AFC_LOCK_MODE[1:0]	Mode
0	Free running: AFC is free running.
1	Disabled: AFC is disabled.
2	Hold: AFC is paused.
3	Lock: AFC locks after the preamble or sync word.

The bandwidth of the AFC loop can be controlled by the AFC_KI and AFC_KP parameters in the RADIO_CFG_11 register (Address 0x117).

The maximum AFC pull-in range is automatically set based on the programmed IF filter bandwidth (IFBW in the RADIO_ CFG_9 register (Address 0x115).

Table 88. Maximum AFC Pull-In Range

IF Bandwidth	Max AFC Pull-In Range
100 kHz	±50 kHz
150 kHz	±75 kHz
200 kHz	±100 kHz
300 kHz	±150 kHz

AFC and Preamble Length

The AFC requires a certain number of the received preamble bits to correct the frequency error between the transmitter and the receiver. The number of preamble bits required depends on the data rate and whether the AFC is locked on detection of the qualified preamble or locked on detection of the qualified sync word. This is discussed in more detail in the Recommended Receiver Settings for 2FSK/GFSK/MSK/GMSK section.

AFC Readback

The frequency error between the received carrier and the receiver local oscillator can be measured when AFC is enabled. The error value can be read from the FREQUENCY_ERROR_ READBACK register (Address 0x372), where each LSB equates to 1 kHz. The value is a twos complement number. The FREQUENCY_ERROR_READBACK value is valid in the PHY_RX state after the AFC has been locked. The value is retained in the FREQUENCY_ERROR_READBACK register after recovering a packet and transitioning back to the PHY_ON state.

Post-Demodulator Filter

A second-order, digital low-pass filter removes excess noise from the demodulated bit stream at the output of the discriminator. The bandwidth of this post-demodulator filter is programmable and must be optimized for the user's data rate and received modulation type. If the bandwidth is set too narrow, performance degrades due to InterSymbol Interference (ISI). If the bandwidth is set too wide, excess noise degrades the performance of the receiver. For optimum performance, the post-demodulator filter bandwidth should be set close to 0.75 times the data rate (when using FSK/GFSK/MSK/GMSK modulation). The actual bandwidth of the post-demodulator filter is given by

$$\textit{Post-Demodulator Filter Bandwidth (kHz)} = \textit{POST\_DEMOD\_BW} \times 2$$

where *POST_DEMOD_BW* is set in the RADIO_CFG_4 register (Address 0x110).

Clock Recovery

An oversampled digital Clock and Data Recovery (CDR) PLL is used to resynchronize the received bit stream to a local clock in all modulation modes. The maximum symbol rate tolerance of the CDR PLL is determined by the number of bit transitions in the transmitted bit stream. For example, during reception of a 010101 preamble, the CDR achieves a maximum data rate tolerance of ±3.0%. However, this tolerance is reduced during recovery of the remainder of the packet where symbol transitions may not be guaranteed to occur at regular intervals during the payload data. To maximize data rate tolerance of the receiver's CDR, 8b/10b encoding or Manchester encoding should be enabled, which guarantees a maximum number of contiguous bits in the transmitted bit stream. Data whitening can also be enabled on the RF transceiver to break up long sequences of contiguous data bit patterns.

Using 2FSK/GFSK/MSK/GMSK modulation, it is also possible to tolerate uncoded payload data fields and payload data fields with long run length coding constraints if the data rate tolerance and packet length are both constrained. More details of CDR operation using uncoded packet formats are discussed in the AN-915 Application Note.

The RF transceiver's CDR PLL is optimized for fast acquisition of the recovered symbols during preamble and typically achieves bit synchronization within five symbol transitions of preamble.

Recommended Receiver Settings for 2FSK/GFSK/MSK/GMSK

To optimize the RF transceiver receiver performance and to ensure the lowest possible packet error rate, it is recommended to use the following configurations:

- Set the recommended AGC low and high thresholds and the AGC clock divide.
- Set the recommended AFC Ki and Kp parameters.
- Use a preamble length ≥ the minimum recommended preamble length.
- When the AGC is configured to lock on the sync word at data rates greater than 200 kbps, it is recommended to set the sync word error tolerance to one bit.

The recommended settings for AGC, AFC, preamble length, and sync word are summarized in Table 90.

Recommended AGC Settings

To optimize the receiver for robust packet error rate performance, when using minimum preamble length over the full input power range, it is recommended to overwrite the default AGC settings in the MCR memory. The recommended settings are as follows:

- AGC_HIGH_THRESHOLD (Address 0x35F) = 0x78
- AGC_LOW_THRESHOLD (Address 0x35E) = 0x46

AGC_CLOCK_DIVIDE (Address 0x32F) = 0x0F or 0x19 (depends on the data rate; see Table 90).

MCR memory is not retained in PHY_SLEEP; therefore, to allow the use of these optimized AGC settings in low power mode applications, a static register fix can be used. An example static register fix to write to the AGC settings in MCR memory is shown in Table 89.

Table 89. Example Static Register Fix for AGC Settings

BBRAM Register	Data	Description
0x128 (STATIC_REG_FIX)	0x2B	Pointer to BBRAM Address 0x12B
0x12B	0x5E	MCR Address 0x35E
0x12C	0x46	Data to write to MCR Address 0x35E (sets AGC low threshold)
0x12D	0x5F	MCR Address 0x35F
0x12E	0x78	Data to write to MCR Address 0x35F (sets AGC high threshold)
0x12F	0x2F	MCR Address 0x32F
0x130	0x0F	Data to write to MCR Address 0x32F (sets AGC clock divide)
0x131	0x00	Ends static MCR register fixes

Recommended AFC Settings

The bandwidth of the AFC loop is controlled by the AFC_KI and AFC_KP parameters in the RADIO_CFG_11 register (Address 0x117). To ensure optimum AFC accuracy while minimizing the AFC settling time (and thus the required preamble length), the AFC_KI and AFC_KP parameters should be set as outlined in Table 90.

Recommended Preamble Length

When AFC is locked on preamble detection, the minimum preamble length is between 40 and 60 bits depending on the data rate. When AFC is set to lock on sync word detection, the minimum preamble length is between 14 and 32 bits, depending on the data rate. When AFC and preamble detection are disabled, the minimum preamble length is dependent on the AGC settling time and the CDR acquisition time and is between 8 and 24 bits, depending on the data rate. The required preamble length for various data rates and receiver configurations is summarized in Table 90.

Recommended Sync Word Tolerance

At data rates greater than 200 kbps and when the AGC is configured to lock on the sync word, it is recommended to set the sync word error tolerance to one bit (SYNC_ERROR_TOL = 1). This prevents an AGC gain change during sync word reception causing a packet loss by allowing one bit error in the received sync word.

Table 90. Summary of Recommended AGC, AFC, Preamble Length, and Sync Word Error Tolerance for 2FSK/GFSK/MSK/GMSK

Data Rate (kbps)	Freq Deviation (kHz)	IF BW (kHz)	Setup[1]	AGC[2]			AFC[3]			Min. Preamble Length (Bits)[4]	Sync Word Error Tolerance (Bits)[5]
				High Threshold	Low Threshold	Clock Divide	On/ Off	Ki	Kp		
300	75	300	1	0x78	0x46	0x0F	On	7	3	64	0
			2	0x78	0x46	0x19	On	8	3	32	1
			3	0x78	0x46	0x19	Off			24	1
38.4	20	100	1	0x78	0x46	0x19	On	7	3	44	0
			2	0x78	0x46	0x19	On	7	3	14	0
			3	0x78	0x46	0x19	Off			8	0
1	10	100	1	0x78	0x46	0x19	On	7	3	40	0
			3	0x78	0x46	0x19	Off			8	0

[1] Setup 1: AFC and AGC are configured to lock on preamble detection by setting AFC_LOCK_MODE = 3 and AGC_LOCK_MODE = 3.
Setup 2: AFC and AGC are configured to lock on sync word detection by setting AFC_LOCK_MODE = 3, AGC_LOCK_MODE = 3, and PREAMBLE_MATCH = 0.
Setup 3: AFC is disabled and AGC is configured to lock on sync word detection by setting AFC_LOCK_MODE = 1, AGC_LOCK_MODE = 3, and PREAMBLE_MATCH = 0.

[2] The AGC high threshold is configured by writing to the AGC_HIGH_THRESHOLD register (Address 0x35F). The AGC low threshold is configured by writing to the AGC_LOW_THRESHOLD register (Address 0x35E). The AGC clock divide is configured by writing to the AGC_CLOCK_DIVIDE register (Address 0x32F).

[3] The AFC is enabled or disabled by writing to the AFC_LOCK_MODE setting in register RADIO_CFG_10 (Address 0x116). The AFC Ki and Kp parameters are configured by writing to the AFC_KP and AFC_KI settings in the RADIO_CFG_11 register (Address 0x117).

[4] The transmit preamble length (in bytes) is set by writing to the PREAMBLE_LEN register (Address 0x11D).

[5] The sync word error tolerance (in bits) is set by writing to the SYNC_ERROR_TOL setting in the SYNC_CONTROL register (Address 0x120).

Default Transceiver Configurations Register Settings

Table 91 shows the recommended register settings for the Transceiver Configurations specified in Table 62.

Table 91. RADIO_CFG Register Setup Recommendations

Data Rate (kbps)	Frequency Deviation (kHz)	0x10C	0x10D	0x10E	0x10F	0x110	0x111	0x112	0x113	0x114	0x115	0x116	0x117
		RADIO_CFG_00	RADIO_CFG_01	RADIO_CFG_02	RADIO_CFG_03	RADIO_CFG_04	RADIO_CFG_05	RADIO_CFG_06	RADIO_CFG_07	RADIO_CFG_08	RADIO_CFG_09	RADIO_CFG_10	RADIO_CFG_11
300	75	0xB8	0x2B	0xEE	0x0B	0x70	0x00	0x03	0x00	0xFD	0xC0	0x0B	0x37
38.4	20	0x80	0x01	0xC8	0x20	0xE0	0x00	0x00	0x00	0xFD	0x00	0x0B	0x37
1	10	0x0A	0x00	0x64	0x41	0x01	0x00	0x02	0x00	0xFD	0x00	0x0B	0x37

PERIPHERAL FEATURES

Transmit Test Modes

There are two transmit test modes that are enabled by setting the VAR_TX_MODE parameter (Address 0x00D in packet RAM memory), as described in Table 92. VAR_TX_MODE should be set before entering the PHY_TX state.

Table 92. Transmit Test Modes

VAR_TX_MODE	Mode
0	Default; no transmit test mode
1	Transmit random data continuously
2	Transmit the preamble continuously
3	Transmit the carrier continuously
4 to 255	Reserved

APPLICATIONS INFORMATION

Application Circuit

A simplified application circuit for the ADuCRF101 is shown in Figure 43. Power supply decoupling capacitors and ADC input filtering components are not shown. Consult the Power Management Unit section and the ADC Circuit section for more details. This example circuit has been optimized for 915 MHz operation using the differential PA output of the transceiver on an ADuCRF101 evaluation board. A more complete schematic and PCB layout are available in the evaluation board user guide.

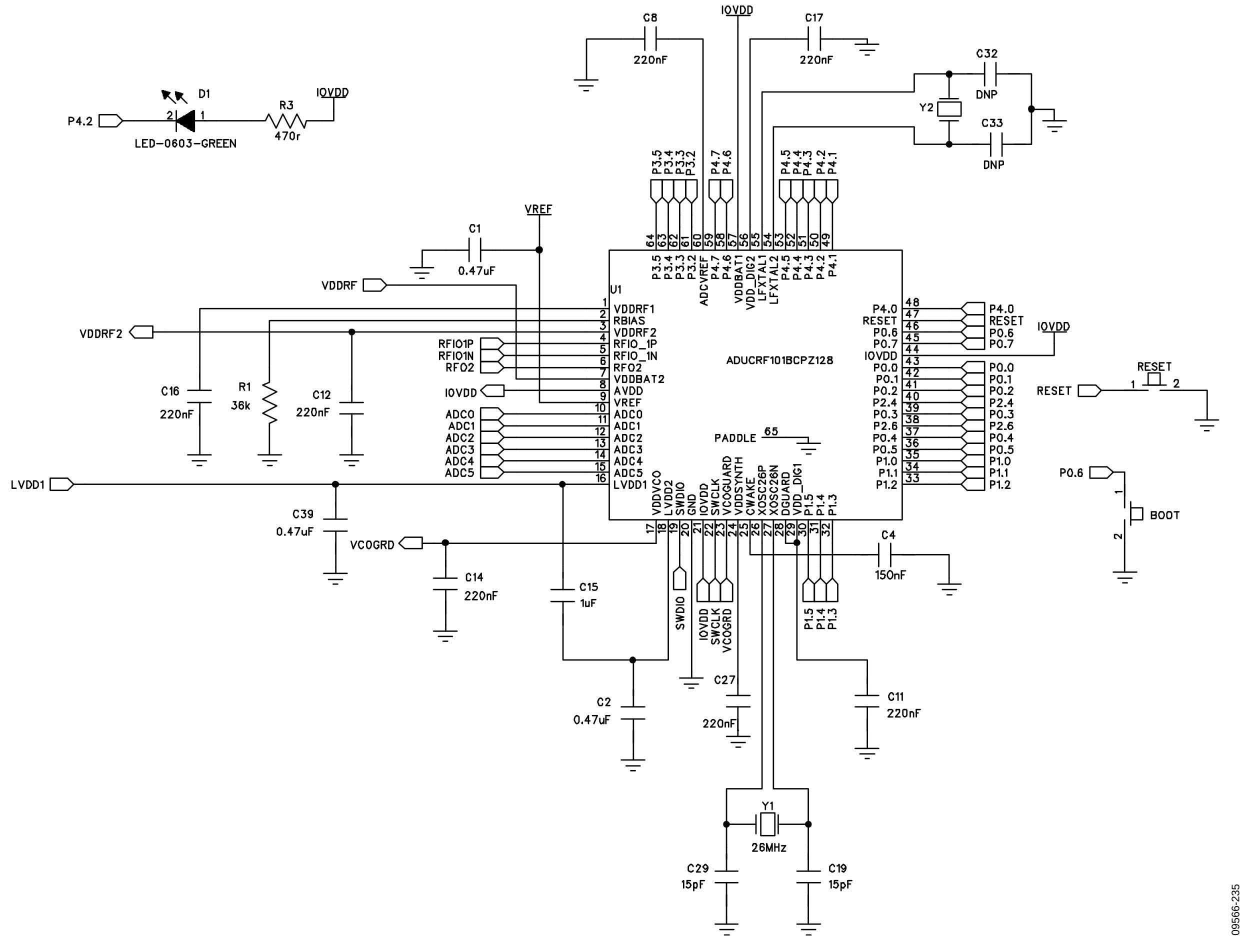


Figure 43. Typical ADuCRF101 Application Circuit Diagram
915 MHz Differential Mode Operation Example

PA/LNA Matching

The RF transceiver has a differential LNA and both a single-ended PA and differential PA. This flexibility allows numerous possibilities in interfacing the RF transceiver to the antenna.

Combined Single-Ended PA and LNA Match

The combined single-ended PA and LNA match allows the transmit and receive paths to be combined without the use of an external transmit/receive switch. The matching network design is shown in Figure 44.

The differential LNA match is a five-element discrete balun giving a single-ended input. The single-ended PA output is a three-element match consisting of the choke inductor to the CREGRF2 regulated supply and an inductor and capacitor series.

The LNA and PA paths are combined, and a T-stage harmonic filter provides attenuation of the transmit harmonics. In a combined match, the off impedances of the PA and LNA must be considered. This can lead to a small loss in transmit power and degradation in receiver sensitivity in comparison with a separate single-ended PA and LNA match. However, with optimum matching, the typical loss in transmit power is <1dB, and the degradation in sensitivity is < 1dB when compared with a separate PA and LNA matching topology.

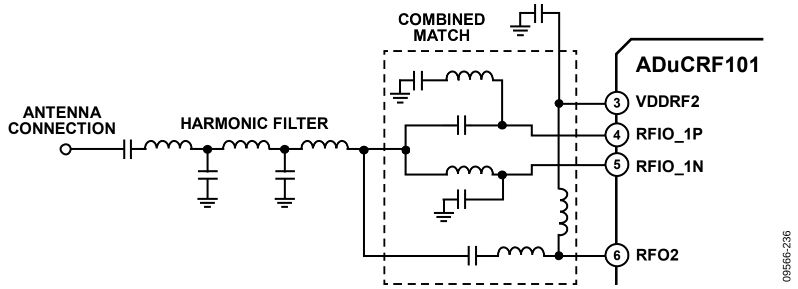


Figure 44. Combined Single-Ended PA and LNA Match

Separate Single-Ended PA/LNA Match

The separate single-ended PA and LNA matching configuration is illustrated in Figure 45. The network is the same as the combined matching network shown in shown in Figure 44 except that the transmit and receive paths are separate. An external transmit/receive antenna switch can be used to combine the transmit and receive paths to allow connection to an antenna. In designing this matching network, it is not necessary to consider the off impedances of the PA and LNA, and, thus, achieving an optimum match is less complex than with the combined single-ended PA and LNA match.

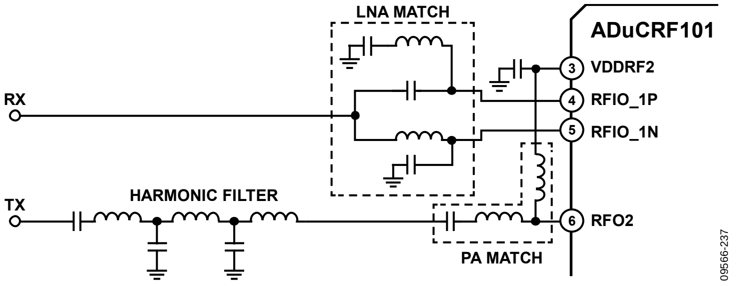


Figure 45. Separate Single-Ended PA and LNA Match
Combined Differential PA/LNA Match

In this matching topology, the single-ended PA is not used. The differential PA and LNA match comprises a five-element discrete balun giving a single-ended input/output as illustrated in Figure 46. The harmonic filter is used to minimize the RF harmonics from the differential PA.

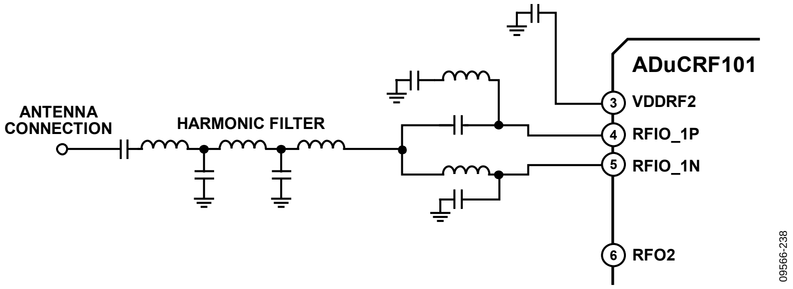


Figure 46. Combined Differential PA and LNA Match Transmit Antenna Diversity

Transmit antenna diversity is possible using the differential PA and single-ended PA. The required matching network is shown in Figure 47.

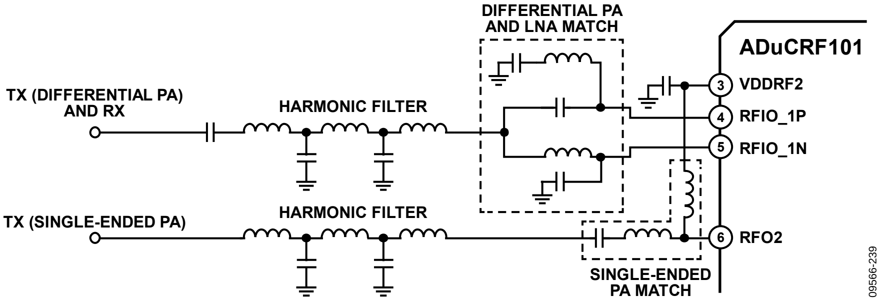


Figure 47. Matching Topology for Transmit Antenna Diversity

COMMAND REFERENCE

Table 93. Radio Controller Commands

Command	Code	Description
CMD_SYNC	0xA2	This is an optional command. It is not necessary to use it during device initialization.
CMD_PHY_OFF	0xB0	Performs a transition of the device into the PHY_OFF state.
CMD_PHY_ON	0xB1	Performs a transition of the device into the PHY_ON state.
CMD_PHY_RX	0xB2	Performs a transition of the device into the PHY_RX state.
CMD_PHY_TX	0xB5	Performs a transition of the device into the PHY_TX state.
CMD_PHY_SLEEP	0xBA	Performs a transition of the device into the PHY_SLEEP state.
CMD_CONFIG_DEV	0xBB	Configures the radio parameters based on the BBRAM values.
CMD_GET_RSSI	0xBC	Performs an RSSI measurement.
CMD_HW_RESET	0xC8	Performs a full hardware reset. The device enters the PHY_SLEEP state.
CMD_RAM_LOAD_INIT	0xBF	Prepares the program RAM for a firmware module download.
CMD_RAM_LOAD_DONE	0xC7	Performs a reset of the communications processor after download of a firmware module to program RAM.
CMD_IR_CAL[1]	0xBD	Initiates an image rejection calibration routine.

[1] The image rejection calibration firmware module must be loaded to program RAM for this command to be functional.

Table 94. SPI Commands

Command	Code	Description
SPI_MEM_WR	00011xxxb = 0x18 (packet RAM) 0x19 (BBRAM) 0x1B (MCR) 0x1E (program RAM)	Writes data to BBRAM, MCR, or packet RAM memory sequentially. An 11-bit address is used to identify memory locations. The most significant three bits of the address are incorporated into the command (xxxb). This command is followed by the remaining eight bits of the address, which are subsequently followed by the data bytes to be written.
SPI_MEM_RD	00111xxxb = 0x38 (packet RAM) 0x39 (BBRAM) 0x3B (MCR)	Reads data from BBRAM, MCR, or packet RAM memory sequentially. An 11-bit address is used to identify memory locations. The most significant three bits of the address are incorporated into the command (xxxb). This command is followed by the remaining eight bits of the address, which are subsequently followed by the appropriate number of SPI_NOP commands.
SPI_MEMR_WR	00001xxxb = 0x08 (packet RAM) 0x09 (BBRAM) 0x0B (MCR)	Writes data to BBRAM, MCR, or packet RAM memory nonsequentially.
SPI_MEMR_RD	00101xxxb = 0x28 (packet RAM) 0x29 (BBRAM) 0x2B (MCR)	Reads data from BBRAM, MCR, or packet RAM memory nonsequentially.
SPI_NOP	0xFF	No operation. Use for dummy writes when polling the status word; used also as dummy data when performing a memory read.

RF TRANSCEIVER REGISTER MAPS

Table 95. Battery Backup Memory (BBRAM)

Address (Hex)	Register	Retained in PHY_SLEEP	Access	Group
0x100	INTERRUPT_MASK_0	Yes	R/W	MAC
0x101	INTERRUPT_MASK_1	Yes	R/W	MAC
0x102 to 0x108	Reserved. Set to 0x00	Not applicable	R/W	Not applicable
0x109	CHANNEL_FREQ_0	Yes	R/W	PHY
0x10A	CHANNEL_FREQ_1	Yes	R/W	PHY
0x10B	CHANNEL_FREQ_2	Yes	R/W	PHY
0x10C	RADIO_CFG_0	Yes	R/W	PHY
0x10D	RADIO_CFG_1	Yes	R/W	PHY
0x10E	RADIO_CFG_2	Yes	R/W	PHY
0x10F	RADIO_CFG_3	Yes	R/W	PHY
0x110	RADIO_CFG_4	Yes	R/W	PHY
0x111	RADIO_CFG_5	Yes	R/W	PHY
0x112	RADIO_CFG_6	Yes	R/W	PHY
0x113	RADIO_CFG_7	Yes	R/W	PHY
0x114	RADIO_CFG_8	Yes	R/W	PHY
0x115	RADIO_CFG_9	Yes	R/W	PHY
0x116	RADIO_CFG_10	Yes	R/W	PHY
0x117	RADIO_CFG_11	Yes	R/W	PHY
0x118	IMAGE_REJECT_CAL_PHASE	Yes	R/W	PHY
0x119	IMAGE_REJECT_CAL_AMPLITUDE	Yes	R/W	PHY
0x11A	Reserved. Set to 0x40	Not applicable	R/W	Not applicable
0x11B	PREAMBLE_MATCH	Yes	R/W	Packet
0x11C	SYMBOL_MODE	Yes	R/W	Packet
0x11D	PREAMBLE_LEN	Yes	R/W	Packet
0x11E	CRC_POLY_0	Yes	R/W	Packet
0x11F	CRC_POLY_1	Yes	R/W	Packet
0x120	SYNC_CONTROL	Yes	R/W	Packet
0x121	SYNC_BYTE_0	Yes	R/W	Packet
0x122	SYNC_BYTE_1	Yes	R/W	Packet
0x123	SYNC_BYTE_2	Yes	R/W	Packet
0x124	TX_BASE_ADR	Yes	R/W	Packet
0x125	RX_BASE_ADR	Yes	R/W	Packet
0x126	PACKET_LENGTH_CONTROL	Yes	R/W	Packet
0x127	PACKET_LENGTH_MAX	Yes	R/W	Packet
0x128	STATIC_REG_FIX	Yes	R/W	PHY
0x129	ADDRESS_MATCH_OFFSET	Yes	R/W	Packet
0x12A	ADDRESS_LENGTH	Yes	R/W	Packet
0x12B to 0x137	Address filtering	Yes	R/W	Packet
0x138	Reserved. Set to 0x00	Not applicable	R/W	Not applicable
0x13A	TRANSITION_CLOCK_DIV	Yes	R/W	PHY
0x13B to 0x13D	Reserved; set to 0x00	Not applicable	R/W	Not applicable

Table 96. Modem Configuration Memory (MCR)

Address (Hex)	Register	Retained in PHY_SLEEP	Access
0x307	PA_LEVEL_MCR	No	R/W
0x30C	WUC_CONFIG_HIGH	No	W
0x30D	WUC_CONFIG_LOW	No	W
0x312	RSSI_READBACK	No	R
0x319	IMAGE_REJECT_CAL_CONFIG	No	R/W
0x32F	AGC_CLOCK_DIVIDE	No	R/W
0x336	INTERRUPT_SOURCE_0	No	R/W
0x337	INTERRUPT_SOURCE_1	No	R/W
0x338	CALIBRATION_CONTROL	No	R/W
0x339	CALIBRATION_STATUS	No	R
0x345	RXBB_CAL_CALWRD_READBACK	No	R
0x346	RXBB_CAL_CALWRD_OVERWRITE	No	RW
0x35D	AGC_MODE	No	R/W
0x35E	AGC_LOW_THRESHOLD	No	R/W
0x35F	AGC_HIGH_THRESHOLD	No	R/W
0x360	AGC_GAIN_STATUS	No	R
0x372	FREQUENCY_ERROR_READBACK	No	R
0x3D2	OSC_CONFIG	No	R/W

Table 97. Packet RAM Memory

Address	Register	Access
0x000	VAR_COMMAND	R/W
0x001 to 0x00B	Reserved	R
0x00D	VAR_TX_MODE	R/W
0x00E to 0x00F	Reserved	R

BBRAM Register Description

Table 98. 0x100: INTERRUPT_MASK_0

Bit	Name	Description	Access
7:5	Reserved		
4	INTERRUPT_TX_EOF	Interrupt when a packet has finished transmitting 1: interrupt enabled; 0: interrupt disabled	R/W
3	INTERRUPT_ADDRESS_MATCH	Interrupt when a received packet has a valid address match 1: interrupt enabled; 0: interrupt disabled	R/W
2	INTERRUPT_CRC_CORRECT	Interrupt when a received packet has the correct CRC 1: interrupt enabled; 0: interrupt disabled	R/W
1	INTERRUPT_SYNC_DETECT	Interrupt when a qualified sync word has been detected in the received packet 1: interrupt enabled; 0: interrupt disabled	R/W
0	INTERRUPT_PREMABLE_DETECT	Interrupt when a qualified preamble has been detected in the received packet 1: interrupt enabled; 0: interrupt disabled	R/W

Table 99. 0x101: INTERRUPT_MASK_1

Bit	Name	Description	Access
7	Reserved		R/W
6	CMD_READY	Interrupt when the communications processor is ready to load a new command; mirrors the CMD_READY bit of the status word 1: interrupt enabled; 0: interrupt disabled	R/W
5:2	Reserved		R/W
1	SPI_READY	Interrupt when the SPI is ready for access 1: interrupt enabled; 0: interrupt disabled	R/W
0	CMD_FINISHED	Interrupt when the communications processor has finished performing a command 1: interrupt enabled; 0: interrupt disabled	R/W

Table 100. 0x109: CHANNEL_FREQ_0

Bit	Name	Description	Access
[7:0]	CHANNEL_FREQ[7:0]	The RF channel frequency in hertz is set according to $Frequency\ (Hz) = F_{PFD} \times \frac{(CHANNEL\_FREQ[23:0])}{2^{16}}$ where F_{PFD} is the PFD frequency and is equal to 26 MHz.	R/W

Table 101. 0x10A: CHANNEL_FREQ_1

Bit	Name	Description	Access
[7:0]	CHANNEL_FREQ[15:8]	See the CHANNEL_FREQ_0 description in Table 100.	R/W

Table 102. 0x10B: CHANNEL_FREQ_2

Bit	Name	Description	Access
[7:0]	CHANNEL_FREQ[23:16]	See the CHANNEL_FREQ_0 description in Table 100.	R/W

Table 103. 0x10C: RADIO_CFG_0

Bit	Name	Description	Access
[7:0]	DATA_RATE[7:0]	The data rate in bps is set according to $Data\ \ Rate\ (bps) = DATA\_RATE\ [11:0] \times 100$	R/W

See Table 91 for the value corresponding to the required transceiver configuration.

Table 104. 0x10D: RADIO_CFG_1

Bit	Name	Description	Access
[7:4]	FREQ_DEVIATION[11:8]	See the FREQ_DEVIATION description in RADIO_CFG_2.	R/W
[3:0]	DATA_RATE[11:8]	See the DATA_RATE description in RADIO_CFG_0 (see Table 103).	R/W

See Table 91 for the value corresponding to the required transceiver configuration.

Table 105. 0x10E: RADIO_CFG_2

Bit	Name	Description	Access
[7:0]	FREQ_DEVIATION[7:0]	The binary level 2FSK/GFSK/MSK/GMSK frequency deviation in hertz (defined as the frequency difference between carrier frequency and 1/0 tones) is set according to $Frequency\ \ Deviation\ \ (Hz)\ =\ FREQ\_DEVIA\ TION\ [11\ :\ 0] \times 100$	R/W

See Table 91 for the value corresponding to the required transceiver configuration.

Table 106. 0x10F: RADIO_CFG_3

Bit	Name	Description	Access
[7:0]	DISCRIM_BW[7:0]	The DISCRIM_BW value sets the bandwidth of the correlator demodulator. See the 2FSK/GFSK/MSK/GMSK Demodulation section for the steps required to set the DISCRIM_BW value.	R/W

See Table 91 for the value corresponding to the required transceiver configuration.

Table 107. 0x110: RADIO_CFG_4

Bit	Name	Description	Access
[7:0]	POST_DEMOD_BW[7:0]	For optimum performance, the post-demodulator filter bandwidth should be set close to 0.75 times the data rate. The actual bandwidth of the post-demod-ulator filter is given by *Post-Demodulator Filter Bandwidth* (kHz) = *POST_DEMOD_BW* × 2 The range of POST_DEMOD_BW is 1 to 255.	R/W

See Table 91 for the value corresponding to the required transceiver configuration.

Table 108. 0x111: RADIO_CFG_5

Bit	Name	Description	Access
[7:0]	Reserved	Set to zero.	R/W

See Table 91 for the value corresponding to the required transceiver configuration.

Table 109. 0x112: RADIO_CFG_6

Bit	Name	Description	Access
[7:2]	SYNTH_LUT_CONFIG_0	If SYNTH_LUT_CONTROL (Address 0x113) = 0 or 2, set SYNTH_LUT_CONFIG_0 = 0. If SYNTH_LUT_CONTROL = 1 or 3, this setting allows the receiver PLL loop bandwidth to be changed to optimize the receiver local oscillator phase noise.	R/W
[1:0]	DISCRIM_PHASE[1:0]	The DISCRIM_PHASE value sets the phase of the correlator demodulator. See the 2FSK/GFSK/MSK/GMSK Demodulation section for the steps required to set the DISCRIM_PHASE value.	R/W

See Table 91 for the value corresponding to the required transceiver configuration.

Table 110. 0x113: RADIO_CFG_7

Bit	Name	Description	Access
[7:6]	AGC_LOCK_MODE	Set to 00: free running 01: manual 10: hold 11: lock after preamble/sync word (only locks on a sync word if PREAMBLE_MATCH = 0)	R/W
[5:4]	Reserved	Set to zero	R/W
[3:0]	Reserved	Set to zero	R/W

See Table 91 for the value corresponding to the required transceiver configuration.

Table 111. 0x114: RADIO_CFG_8

Bit	Name	Description		Access
7	PA_SINGLE_DIFF_SEL	**PA_SINGLE_DIFF_SEL**	**PA**	R/W
		0 1	Single-ended PA enabled Differential PA enabled	
[6:3]	PA_LEVEL	Sets the PA output power. A value of zero sets the minimum RF output power, and a value of 15 sets the maximum PA output power. The PA level can also be set with finer resolution using the PA_LEVEL_MCR setting (Address 0x307). The PA_LEVEL setting is related to the PA_LEVEL_MCR setting by PA_LEVEL_MCR = 4 × PA_LEVEL + 3		R/W
		PA_LEVEL	**PA Level (PA_LEVEL_MCR)**	
		0000 0001 0010 … 1111	Setting 3 Setting 7 Setting 11 … Setting 63	

Bit	Name	Description		Access
[2:0]	PA_RAMP	Sets the PA ramp rate. The PA ramps at the programmed rate until it reaches the level indicated by the PA_LEVEL_MCR (Address 0x307) setting. The ramp rate is dependent on the programmed data rate.		R/W
		PA_RAMP	**Ramp Rate**	
		000	Reserved	
		001	256 codes per data bit	
		010	128 codes per data bit	
		011	64 codes per data bit	
		100	32 codes per data bit	
		101	16 codes per data bit	
		110	Eight codes per data bit	
		111	Four codes per data bit	
		To ensure the correct PA ramp-up and -down timing, the PA ramp rate has a minimum value based on the data rate and the PA_LEVEL or PA_LEVEL_MCR settings. This minimum value is described by $Ramp\ Rate(Codes/Bit) < 10{,}000 \times \frac{PA\_LEVEL\_MCR[5:0]}{DATA\_RATE[11:0]}$ where PA_LEVEL_MCR is related to the PA_LEVEL setting by PA_LEVEL_MCR = 4 × PA_LEVEL + 3.		

See Table 91 for the value corresponding to the required transceiver configuration.

Table 112. 0x115: RADIO_CFG_9

Bit	Name	Description		Access
[7:6]	IFBW	Sets the receiver IF filter bandwidth. Note that setting an IF filter bandwidth of 300 kHz automatically changes the receiver IF frequency from 200 kHz to 300 kHz.		R/W
		IFBW	IF Bandwidth	
		000	100 kHz	
		001	150 kHz	
		010	200 kHz	
		011	300 kHz	
[5:3]	MOD_SCHEME	Sets the transmitter modulation scheme.		R/W
		MOD_SCHEME	Modulation Scheme	
		00	Two-level 2FSK/MSK	
		01	Two-level GFSK/GSMK	
		11	Carrier only	
		0100 to 0111	Reserved	
[2:0]	DEMOD_SCHEME	Sets the receiver demodulation scheme.		R/W
		DEMOD_SCHEME	Demodulation Scheme	
		000	2FSK/GFSK/MSK/GMSK	
		001	Reserved	
		011 to 111	Reserved	

See Table 91 for the value corresponding to the required transceiver configuration.

Table 113. 0x116: RADIO_CFG_10

Bit	Name	Description		R/W
[7:5]	Reserved	Set to 0.		R/W
4	AFC_POLARITY	Set to 0.		R/W
[3:2]	AFC_SCHEME	Set to 2.		R/W
[1:0]	AFC_LOCK_MODE	Sets the AFC mode.		R/W
		AFC_LOCK_MODE	**Mode**	
		000	Free running: AFC is free running.	
		001	Disabled: AFC is disabled.	
		010	Hold AFC: AFC is paused.	
		011	Lock: AFC locks after the preamble or sync word (only locks on a sync word if PREAMBLE_MATCH = 0).	

See Table 91 for the value corresponding to the required transceiver configuration.

Table 114. 0x117: RADIO_CFG_11

Bit	Name	Description	Access
[7:4]	AFC_KP	Sets the AFC PI controller proportional gain in 2FSK/GFSK/MSK/GMSK; the recommended value is 0x3.	R/W
[3:0]	AFC_KI	Sets the AFC PI controller integral gain in 2FSK/GFSK/MSK/GMSK; the recommended value is 0x7.	R/W

See Table 91 for the value corresponding to the required transceiver configuration

Table 115. 0x118: IMAGE_REJECT_CAL_PHASE

Bit	Name	Description	Access
7	Reserved	Set to 0	R/W
[6:0]	IMAGE_REJECT_CAL_PHASE	Sets the I/Q phase adjustment	R/W

Table 116. 0x119: IMAGE_REJECT_CAL_AMPLITUDE

Bit	Name	Description	Access
7	Reserved	Set to 0	R/W
[6:0]	IMAGE_REJECT_CAL_AMPLITUDE	Sets the I/Q amplitude adjustment	R/W

Table 117. 0x11B: PREAMBLE_MATCH

Bit	Name	Description		Access
[7:4]	Reserved	Set to 0		R/W
[3:0]	PREAMBLE_MATCH	**PREAMBLE_MATCH**	**Description**	R/W
		12	0 errors allowed.	
		11	One erroneous bit-pair allowed in 12 bit-pairs.	
		10	Two erroneous bit-pairs allowed in 12 bit-pairs.	
		9	Three erroneous bit-pairs allowed in 12 bit-pairs.	
		8	Four erroneous bit-pairs allowed in 12 bit-pairs.	
		0	Preamble detection disabled.	
		1 to 7	Not recommended.	
		13 to 15	Reserved.	

Table 118. 0x11C: SYMBOL_MODE

Bit	Name	Description	Access
7	Reserved	Set to 0.	R/W
6	MANCHESTER_ENC	1: Manchester encoding and decoding enabled. 0: Manchester encoding and decoding disabled.	R/W
5	PROG_CRC_EN	1: programmable CRC selected. 0: default CRC selected.	R/W
4	Reserved		R/W
3	DATA_WHITENING	1: data whitening and dewhitening enabled. 0: data whitening and dewhitening disabled.	R/W
[2:0]	Reserved		R/W

Table 119. 0x11D: PREAMBLE_LEN

Bit	Name	Description	Access
[7:0]	PREAMBLE_LEN	Length of preamble in bytes. Example: a value of decimal 3 results in a preamble of 24 bits.	R/W

Table 120. 0x11E: CRC_POLY_0

Bit	Name	Description	Access
[7:0]	CRC_POLY[7:0]	Lower byte of CRC_POLY[15:0], which sets the CRC polynomial.	R/W

Table 121. 0x11F: CRC_POLY_1

Bit	Name	Description	Access
[7:0]	CRC_POLY[15:8]	Upper byte of CRC_POLY[15:0], which sets the CRC polynomial. See the Packet Mode section for more details on how to configure a CRC polynomial.	R/W

Table 122. 0x120: SYNC_CONTROL

<table>
<tr><th>Bit</th><th>Name</th><th colspan="2">Description</th><th>Access</th></tr>
<tr><td rowspan="6">[7:6]</td><td rowspan="6">SYNC_ERROR_TOL</td><td colspan="2">Sets the sync word error tolerance in bits.</td><td rowspan="6">R/W</td></tr>
<tr><td><b>SYNC_ERROR_TOL</b></td><td><b>Bit Error Tolerance</b></td></tr>
<tr><td>0</td><td>0 bit errors allowed.</td></tr>
<tr><td>1</td><td>One bit error allowed.</td></tr>
<tr><td>2</td><td>Two bit errors allowed.</td></tr>
<tr><td>3</td><td>Three bit errors allowed.</td></tr>
<tr><td>5</td><td>Reserved</td><td colspan="2">Set to 0.</td><td>R/W</td></tr>
<tr><td rowspan="7">[4:0]</td><td rowspan="7">SYNC_WORD_LENGTH</td><td colspan="2">Sets the sync word length in bits; 24 bits is the maximum. Note that the sync word matching length can be any value up to 24 bits, but the transmitted sync word pattern is a multiple of eight bits. Therefore, for non-byte-length sync words, the transmitted sync pattern should be filled out with the preamble pattern.</td><td rowspan="7">R/W</td></tr>
<tr><td><b>SYNC_WORD_LENGTH</b></td><td><b>Length in Bits</b></td></tr>
<tr><td>00000</td><td>0</td></tr>
<tr><td>00001</td><td>1</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>11000</td><td>24</td></tr>
<tr><td>11001 to 11111</td><td>Reserved</td></tr>
</table>

Table 123. 0x121: SYNC_BYTE_0

Bit	Name	Description	Access
[7:0]	SYNC_BYTE[23:16]	Upper byte of the sync word pattern. The sync word pattern is transmitted most significant bit first starting with SYNC_BYTE_0. For nonbyte length sync words, the reminder of the least significant byte should be stuffed with the preamble. If SYNC_WORD_LENGTH length is >16 bits, SYNC_BYTE_0, SYNC_BYTE_1, and SYNC_BYTE_2 are all transmitted for a total of 24 bits. If SYNC_WORD_LENGTH is between 8 and 15, SYNC_BYTE_1 and SYNC_ BYTE_2 are transmitted. If SYNC_WORD_LENGTH is between 1 and 7, SYNC_BYTE_2 is transmitted for a total of eight bits. If the SYNC WORD LENGTH is 0, no sync bytes are transmitted.	R/W

Table 124. 0x122: SYNC_BYTE_1

Bit	Name	Description	Access
[7:0]	SYNC_BYTE[15:8]	Middle byte of the sync word pattern.	R/W

Table 125. 0x123: SYNC_BYTE_2

Bit	Name	Description	Access
[7:0]	SYNC_BYTE[7:0]	Lower byte of the sync word pattern.	R/W

Table 126. 0x124: TX_BASE_ADR

Bit	Name	Description	Access
[7:0]	TX_BASE_ADR	Address in packet RAM of the transmit packet. This address indicates to the communications processor the location of the first byte of the transmit packet.	R/W

Table 127. 0x125: RX_BASE_ADR

Bit	Name	Description	Access
[7:0]	RX_BASE_ADR	Address in packet RAM of the receive packet. The communications processor writes any qualified received packet to packet RAM, starting at this memory location.	R/W

Table 128. 0x126: PACKET_LENGTH_CONTROL

Bit	Name	Description	Access
7	DATA_BYTE	Over-the-air arrangement of each transmitted packet RAM byte. A byte is transmitted either MSB or LSB first. The same setting should be used on the Tx and Rx sides of the link. 1: data byte MSB first. 0: data byte LSB first.	R/W
6	PACKET_LEN	1: fixed packet length mode. Fixed packet length in Tx and Rx modes, given by PACKET_LENGTH_MAX. 0: variable packet length mode. In Rx mode, packet length is given by the first byte in packet RAM. In Tx mode, the packet length is given by PACKET_LENGTH_MAX.	R/W
5	CRC_EN	1: append CRC in transmit mode. Check CRC in receive mode. 0: no CRC addition in transmit mode. No CRC check in receive mode.	R/W
[4:3]	Reserved	Set to zero	
[2:0]	LENGTH_OFFSET	Offset value in bytes that is added to the received packet length field value (in variable length packet mode) so that the communications processor knows the correct number of bytes to read. The communications processor calculates the actual received payload length as *Rx Payload Length* = *Length* + *LENGTH_OFFSET* − 4 where *Length* is the length field (the first byte in the received payload).	R/W

Table 129. 0x127: PACKET_LENGTH_MAX

Bit	Name	Description	Access
[7:0]	PACKET_LENGTH_MAX	If variable packet length mode is used (PACKET_LENGTH_CONTROL = 0), PACKET_LENGTH_MAX sets the maximum receive packet length in bytes. If fixed packet length mode is used (PACKET_LENGTH_CONTROL = 1), PACKET_LENGTH_MAX sets the length of the fixed transmit and receive packet in bytes. Note that the packet length is defined as the number of bytes from the end of the sync word to the start of the CRC. It also does not include the LENGTH_OFFSET value.	R/W

Table 130. 0x128: STATIC_REG_FIX

<table>
<tr><th>Bit</th><th>Name</th><th colspan="3">Description</th><th>Access</th></tr>
<tr><td rowspan="8">[7:0]</td><td rowspan="8">STATIC_REG_FIX</td><td colspan="3">The RF transceiver has the ability to implement automatic static register fixes from BBRAM memory to MCR memory. This feature allows a maximum of nine MCR registers to be programmed via BBRAM memory. This feature is useful if MCR registers must be configured for optimum receiver performance in low power mode.
The STATIC_REG_FIX value is an address pointer to any BBRAM memory address between 0x12A and 0x13D. For example, to point to BBRAM Address 0x12B, set STATIC_REG_FIX= 0x2B.
• If STATIC_REG_FIX = 0x00, then static register fixes are disabled.
• If STATIC_REG_FIX is nonzero, the communications processor looks for the MCR address and corresponding data at the BBRAM address beginning at STATIC_REG_FIX.
Example: write 0x46 to MCR Register 0x35E and write 0x78 to MCR Register 0x35F. Set STATIC_REG_FIX = 0x2B.</td><td>R/W</td></tr>
<tr><td><b>BBRAM Register</b></td><td><b>Data</b></td><td><b>Description</b></td><td></td></tr>
<tr><td>0x128 (STATIC_REG_FIX)</td><td>0x2B</td><td>Pointer to BBRAM Address 0x12B</td><td></td></tr>
<tr><td>0x12B</td><td>0x5E</td><td>MCR Address 1</td><td></td></tr>
<tr><td>0x12C</td><td>0x46</td><td>Data to write to MCR Address 1</td><td></td></tr>
<tr><td>0x12D</td><td>0x5F</td><td>MCR Address 2</td><td></td></tr>
<tr><td>0x12E</td><td>0x78</td><td>Data to write to MCR Address 2</td><td></td></tr>
<tr><td>0x12F</td><td>0x00</td><td>Ends static MCR register fixes</td><td></td></tr>
</table>

Table 131. 0x129: ADDRESS_MATCH_OFFSET

Bit	Name	Description	Access
[7:0]	ADDRESS_MATCH_OFFSET	Location of first byte of address information in packet RAM	R/W

Table 132. 0x12A: ADDRESS_LENGTH

Bit	Name	Description	Access
[7:0]	ADDRESS_LENGTH	Number of bytes in the first address field (N_{ADR_1}). Set to zero if address filtering is not being used.	R/W

Table 133. 0x12B to 0x137: Address Filtering (or Static Register Fix)

Address	Bit	Description	Access
0x12B	7:0	Address 1 Match Byte 0.	R/W
0x12C	7:0	Address 1 Mask Byte 0.	R/W
0x12D	7:0	Address 1 Match Byte 1.	R/W
0x12E	7:0	Address 1 Mask Byte 1.	R/W
…		…	…
	7:0	Address 1 Match Byte N_{ADR_1}.	R/W
	7:0	Address 1 Mask Byte N_{ADR_1}.	R/W
	7:0	0x00 to end or number of bytes in the second address field (N_{ADR_2})	R/W

Table 134. 0x138: RSSI_WAIT_TIME

Bit	Name	Description	Access
[7:0]	RSSI_WAIT_TIME	Settling time in µs before taking an RSSI measurement in SWM or when using CMD_GET_RSSI. A value of 0xA7 can be used safely in all situations; however, this can be reduced for particular implementations.	R/W

Table 135. 0x13A: TRANSITION_CLOCK_DIV

Bit	Name	Description	R/W
[7:0]	TRANSITION_CLOCK_DIV	0x00: Normal transition times. 0x01: Fast transition times. 0x04: Normal transition times. Else: Reserved.	R/W

MCR REGISTER DESCRIPTION

The MCR register settings are not retained when the device enters the PHY_SLEEP state.

Table 136. 0x307: PA_LEVEL_MCR

Bit	Name	Description	Reset	Access
[5:0]	PA_LEVEL_MCR	Power amplifier level. If PA ramp is enabled, the PA ramps to this target level. The PA level can be set in the 0 to 63 range. The PA level (with less resolution) can also be set via the BBRAM; therefore, the MCR setting should be used only if more resolution is required.	0	R/W

Table 137. 0x30C: WUC_CONFIG_HIGH

Bit	Name	Description	Reset	Access
[7:0]	Reserved	Set to 0.	0	W

Register WUC_CONFIG_LOW should never be written to without updating Register WUC_CONFIG_HIGH first.

Table 138. 0x30D: WUC_CONFIG_LOW

Bit	Name	Description	Reset	Access
[7:4]	Reserved	Set to 0.	0	W
3	WUC_BBRAM_EN	1: enable power to the BBRAM during the PHY_SLEEP state. 0: disable power to the BBRAM during the PHY_SLEEP state.	0	W
[2:0]	Reserved	Set to 0.	0	W

Table 139. 0x312: RSSI_READBACK

Bit	Name	Description	Reset	Access
7:0	RSSI_READBACK	Receive input power. After reception of a packet, the RSSI_READBACK value is valid. *RSSI (dBm) = RSSI_READBACK* – 107	0	R

Table 140. 0x319: IMAGE_REJECT_CAL_CONFIG

Bit	Name	Description	Reset	Access
[7:6]	Reserved		0	R/W
5	IMAGE_REJECT_CAL_OVWRT_EN	Overwrite control for image reject calibration results.	0	R/W
[4:3]	IMAGE_REJECT_FREQUENCY	Set the fundamental frequency of the IR calibration signal source. A harmonic of this frequency can be used as an internal RF signal source for the image rejection calibration. 0: IR calibration source disabled in XTAL divider 1: IR calibration source fundamental frequency = XTAL/4 2: IR calibration source fundamental frequency = XTAL/8 3: IR calibration source fundamental frequency = XTAL/16	0	R/W
[2:0]	IMAGE_REJECT_POWER	Set power level of IR calibration source. 0: IR calibration source disabled at mixer input 1: power level = min 2: power level = min 3: power level = min × 2 4: power level = min × 2 5: power level = min × 3 6: power level = min × 3 7: power level = min × 4	0	R/W

Table 141. 0x32F: AGC_CLOCK_DIVIDE

Bit	Name	Description	Reset	Access
[7:0]	AGC_CLOCK_DIVIDE	AGC clock divider for 2FSK/GFSK/MSK/GMSK mode. The AGC rate is (26 MHz/(16 × AGC_CLOCK_DIVIDE)).	40	R/W

Table 142. 0x336: INTERRUPT_SOURCE_0

Bit	Name	Description	Reset	Access
[7:5]	Reserved		0	R/W
4	INTERRUPT_TX_EOF	Asserted when a packet has finished transmitting (packet mode only)	0	R/W
3	INTERRUPT_ADDRESS_MATCH	Asserted when a received packet has a valid address match (packet mode only)	0	R/W
2	INTERRUPT_CRC_CORRECT	Asserted when a received packet has the correct CRC (packet mode only)	0	R/W
1	INTERRUPT_SYNC_DETECT	Asserted when a qualified sync word has been detected in the received packet	0	R/W
0	INTERRUPT_PREAMBLE_DETECT	Asserted when a qualified preamble has been detected in the received packet	0	R/W

Table 143. 0x337: INTERRUPT_SOURCE_1

Bit	Name	Description	Reset	Access
7	Reserved		0	R/W
6	CMD_READY	Communications processor ready to accept a new command.	0	R/W
5	Unused		0	R/W
4:2	Reserved		0	R/W
1	SPI_READY	SPI ready for access.	0	R/W
0	CMD_FINISHED	Command has finished.	0	R/W

Table 144. 0x338: CALIBRATION_CONTROL

Bit	Name	Description	Reset	Access
[7:2]	Reserved		0	R/W
1	SYNTH_CAL_EN	1: enable the synthesizer calibration state machine. 0: disable the synthesizer calibration state machine.	0	R/W
0	RXBB_CAL_EN	1: enable receiver baseband filter (RXBB) calibration. 0: disable receiver baseband filter (RXBB) calibration.	0	R/W

Table 145. 0x339: CALIBRATION_STATUS

Bit	Name	Description	Reset	Access
[7:3]	Reserved		0	R
2	PA_RAMP_FINISHED		0	R
1	SYNTH_CAL_READY	1: synthesizer calibration finished successfully. 0: synthesizer calibration in progress.	0	R
0	RXBB_CAL_READY	Receive IF filter calibration. 1: complete. 0: in progress (valid while RXBB_CAL_EN = 1).	0	R

Table 146. 0x345: RXBB_CAL_CALWRD_READBACK

Bit	Name	Description	Reset	Access
[5:0]	RXBB_CAL_CALWRD	RXBB reference oscillator calibration word; valid after RXBB calibration cycle has been completed.	0	R

Table 147. 0x346: RXBB_CAL_CALWRD_OVERWRITE

Bit	Name	Description	Reset	Access
[6:1]	RXBB_CAL_DCALWRD_OVWRT_IN	RXBB reference oscillator calibration overwrite word	0	RW
0	RXBB_CAL_DCALWRD_OVWRT_EN	1: enable RXBB reference oscillator calibration word overwrite mode 0: disable RXBB reference oscillator calibration word overwrite mode	0	RW

Table 148. 0x35D: AGC_MODE

Bit	Name	Description	Reset	Access
7	Reserved		0	R/W
[6:5]	AGC_OPERATION_MCR	00: free-running AGC 01: manual AGC 10: hold AGC 11: lock AGC after preamble	0	R/W
4:3	LNA_GAIN	01: low 01: medium 10: high 11: reserved	0	R/W
2	MIXER_GAIN	00: low 01: high	0	R/W
1:0	FILTER_GAIN	00: low 01: medium 10: high 11: reserved	0	R/W

Table 149. 0x35E: AGC_LOW_THRESHOLD

Bit	Name	Description	Reset	Access
[7:0]	AGC_LOW_THRESHOLD	AGC low threshold	55	R/W

Table 150. 0x35F: AGC_HIGH_THRESHOLD

Bit	Name	Description	Reset	Access
[7:0]	AGC_HIGH_THRESHOLD	AGC high threshold	105	R/W

Table 151. 0x360: AGC_GAIN_STATUS

Bit	Name	Description	Reset	Access
[7:5]	Reserved		0	R
[4:3]	LNA_GAIN_READBACK	00: low 01: medium 10: high 11: reserved	0	R
2	MIXER_GAIN_READBACK	0: low 1: high	0	R
[1:0]	FILTER_GAIN_READBACK	00: low 01: medium 10: high 11: reserved	0	R

Table 152. 0x372: FREQUENCY_ERROR_READBACK

Bit	Name	Description	Reset	Access
[7:0]	FREQUENCY_ERROR_READBACK	Frequency error between received signal frequency and receive channel frequency = FREQUENCY_ERROR_READBACK × 1 kHz. The FREQUENCY_ERROR_READBACK value is in twos complement format.	0	R

Table 153. 0x3D2: OSC_CONFIG

Bit	Name	Description	Reset	Access
[7:6]	Reserved	Write 0.	0	R/W
[5:3]	XOSC_CAP_DAC	26 MHz crystal oscillator (XOSC26N) tuning capacitor control word.	0	R/W
[2:0]	Reserved	Write 0.	0	R/W

DIGITAL INPUT/OUTPUTS

DIGITAL I/O FEATURES

Digital input/output features include 28 general-purpose bidirectional input/output (GPIO) pins. All of the GPIO pins have multiple functions, which are selectable by user code.

DIGITAL I/O BLOCK DIAGRAM

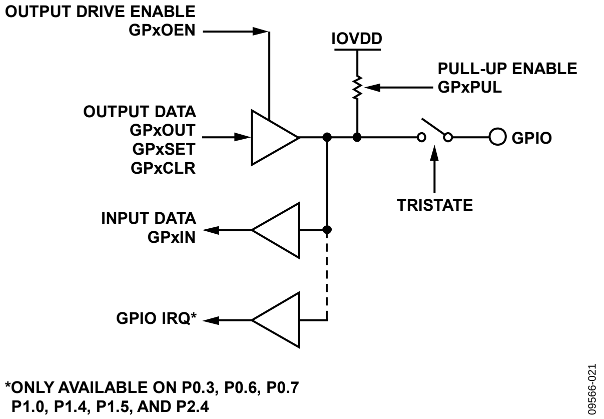


Figure 48. Simplified GPIO Structure

DIGITAL I/O OVERVIEW

Each GPIO can be configured individually as input, output, or have a high impedance state. Each GPIO has an internal pull-up programmable resistor with a drive capability of 4 mA. P3.4 also has a pull down resistor available. All I/O pins are functional over the full supply range (2.2 V to 3.6 V) and the logic input voltages are specified as percentages of the supply.

V_{INL} = 0.2 × IOVDD maximum

V_{INH} = 0.7 × IOVDD minimum

The absolute maximum input voltage is IOVDD + 0.3 V, and the typical leakage current of the GPIOs configured as input or high impedance is 10 nA per GPIO. When the ADuCRF101 enters a power-saving mode, the GPIO pins retain their state. Note that a driving peripheral cannot drive the pin in a power saving mode. That is, if the UART is driving the pin on entry to hibernate, it is isolated from the pin and power is gated. Its state and control is restored on wake-up.

It is recommended to isolate any GPIO from fast switching signals (>1 MHz) by configuring them in high impedance state before entering hibernate mode (see the Power Management Unit section).

DIGITAL I/O OPERATION

The digital I/O pins share multiple functions selectable in the GPxCON registers as per Table 156. External interrupt and input levels are available in any of these configurations except when configured in high impedance state.

Digital I/O Data In

GPIOs power up as inputs. The GPxIN registers reflect the input level of the GPIOs.

Digital I/O Data Out

To configure the GPIOs as an output, the output level should first be configured in the GPxOUT register. The GPxOUT values are reflected on the GPIOs when the corresponding bit in GPxOEN is set. P2.4 should always be configured as an input as it is internally connected to the RF transceiver output interrupt.

Bit Set/Clear/Toggle

GPxSET/CLR/TGL registers are used to set, clear, or toggle one or more GPIOs without affecting others GPIOs within a port. Only the GPIO corresponding with the write data bit equal to one will be set, cleared or toggled, the remaining GPIO will be unaffected.

Digital I/O Pull-Up Resistor

All GPIO pins, except P3.4, configured as inputs, have an internal pull-up resistor with a drive capability of 4 mA. P3.4 has an internal pull down resistor enabled by default. The internal pull-up resistors can be enabled or disabled using the GPxPUL registers. The pull ups are automatically disabled when the GPIO pins are not inputs. The pull-up resistor value varies with the voltage on the pin.

P3.4 Pull-Down Resistor

P3.4 has both an internal pull up resistor and an internal pull down resistor available. On P3.4, the pull up is disabled by default. The pull-up resistors can be controlled by the GPxPUL register. P3.4 has an internal pull-down resistor enabled by default. The pull-down resistor can be controlled in the GPDWN register. This default configuration allows a connection between P3.4 to LVDD1 via a resistive sensor without overdriving the internal LDO. The hardware logic will not allow both the pull-up and pull-down resistors to be enabled at the same time. Valid P3.4 resistors configurations are summarized in Table 5.

Table 154. P3.4 Pull-Up/Pull-Down Configuration Summary

GP3PUL[4]	GPDWN[1]	P3.4 Input Configuration
1 (enabled)	1 (disabled)	Pull-up resistor enabled
1 (enabled)	0 (enabled)	Not allowed, reverted to pull-up configuration
0 (disabled)	1 (disabled)	Nothing connected
0 (disabled)	0 (enabled)	Default, pull down resistor enabled

High Impedance State

The GPIOs can also be in a high impedance state using the GPxOCE register. Table 155 summarizes the possible configurations.

Table 155. GPIO States

GPxOCE	GPxOEN	GPxOUT	GPIO Input States/Interrupts	GPIO Output States
0	0	X	Always available.	Not available. Configured as input.
0	1	X	Always available.	Output level depends on OUT (drive 0 or 1).
1	0	X	Always available.	Not available. Configured as input.
1	1	0	Not available.	Drive 0 out (open collector).
1	1	1	Not available.	Output floating (open collector).

DIGITAL PORT MULTIPLEX

This block provides control over the GPIO functionality of specified pins because some of the pins have a choice to work as GPIO or to have other specific functions. P2.0 toP2.3, P2.5, P2.7, P3.6, and P3.7 are dedicated to the RF transceiver and are not available on external pins. Note that P2.0 toP2.3, P2.5, P2.7, P3.6, and P3.7 are dedicated to the RF transceiver and are not available on external pins.

Table 156. Pin Functions

	Configuration Modes			
Pin	**00**	**01**	**10**	**11**
GP0				
P0.0	GPIO (default)	SPI1MISO	---	---
P0.1	GPIO (default)	SPI1SCLK	---	---
P0.2	GPIO (default)	SPI1MOSI	---	PWM0
P0.3/IRQ1	GPIO (default)	SPI1$\overline{\text{CS0}}$	ADCCONVST	PWM1
P0.4	GPIO (default)	SPI1$\overline{\text{CS1}}$	ECLKOUT	---
P0.5	GPIO (default)	SPI1$\overline{\text{CS2}}$	ECLKIN	---
P0.6/IRQ2	GPIO (default)/BOOT	SPI1$\overline{\text{CS3}}$	UARTRTS	PWM0
P0.7/IRQ3	GPIO (default)	SPI1$\overline{\text{CS4}}$	UARTCTS	---
GP1				
P1.0/IRQ4	GPIO (default)	UARTRXD	SPI1MOSI	PWM2
P1.1	PORB (default)	GPIO	UARTTXD	PWM3
P1.2	--- (default)	GPIO	---	PWM4
P1.3	--- (default)	GPIO	---	PWM5
P1.4/IRQ5	GPIO (default)	I2CSCL	PWM6	---
P1.5/IRQ6	GPIO (default)	I2CSDA	PWM7	---
P1.6	GPIO (default)	ADCCONVST	---	PWMSYNC
GP2				
P2.0	SPI0MISO(default)	GPIO	---	---
P2.1	SPI0SCLK(default)	GPIO	---	---
P2.2	SPI0MOSI(default)	GPIO	---	---
P2.3	SPI0CS(default)	GPIO	---	---
P2.4/IRQ8	GPIO (default)	GPIO	---	---
P2.5	--- (default)	---	---	---
P2.6	--- (default)	GPIO	---	---
GP3				
P3.2	---	GPIO (default)	PWMSYNC	SPI0MISO
P3.3	---	GPIO (default)	PWMTRIP	SPI0SCLK
P3.4	---	GPIO (default)	---	---
P3.5	---	GPIO (default)	---	SPI0MOSI
P3.6	---	GPIO (default)	---	---
P3.7	---	GPIO (default)	---	---
GP4				
P4.0	---	GPIO (default)	PWM0	---
P4.1	---	GPIO (default)	PWM1	---
P4.2	---	GPIO (default)	PWM2	SPI0CS
P4.3	---	GPIO (default)	PWM3	---
P4.4	---	GPIO (default)	PWM4	---
P4.5	---	GPIO (default)	PWM5	---
P4.6	---	GPIO (default)	PWM6	---
P4.7	---	GPIO (default)	PWM7	---

When P1.2 is in its default configuration, this pin reflects the state of SWDIO and will toggle during serial wire communication.

REGISTER SUMMARY (GENERAL-PURPOSE INPUT OUTPUT)

Table 157. General-Purpose Input Output Register Summary

Address	Name	Description	Reset	RW
0x40006000	GP0CON	GPIO Port 0 configuration	0x0000	RW
0x40006004	GP0OEN	GPIO Port 0 output enable	0x00	RW
0x40006008	GP0PUL	GPIO Port 0 pull-up enable	0xFF	RW
0x4000600C	GP0OCE	GPIO Port 0 open collector	0x00	RW
0x40006014	GP0IN	GPIO Port 0 data input	0xFF	R
0x40006018	GP0OUT	GPIO Port 0 data out	0x00	RW
0x4000601C	GP0SET	GPIO Port 0 data out set	0x00	W
0x40006020	GP0CLR	GPIO Port 0 data out clear	0x00	W
0x40006024	GP0TGL	GPIO Port 0 pin toggle	0x00	W
0x40006030	GP1CON	GPIO Port 1 configuration	0x0000	RW
0x40006034	GP1OEN	GPIO Port 1 output enable	0x00	RW
0x40006038	GP1PUL	GPIO Port 1 output pull-up enable	0x7F	RW
0x4000603C	GP1OCE	GPIO Port 1 open collector	0x00	RW
0x40006044	GP1IN	GPIO Port 1 data input	0x7F	R
0x40006048	GP1OUT	GPIO Port 1 data out	0x00	RW
0x4000604C	GP1SET	GPIO Port 1 data out set	0x00	W
0x40006050	GP1CLR	GPIO Port 1 data out clear	0x00	W
0x40006054	GP1TGL	GPIO Port 1 pin toggle.	0x00	W
0x40006060	GP2CON	GPIO Port 2 configuration	0x0000	RW
0x40006064	GP2OEN	GPIO Port 2 output enable	0x00	RW
0x40006068	GP2PUL	GPIO Port 2 output pull-up enable	0xFF	RW
0x4000606C	GP2OCE	GPIO Port 2 open collector	0x00	RW
0x40006074	GP2IN	GPIO port 2 data input	0xFF	R
0x40006078	GP2OUT	GPIO Port 2 data out	0x00	RW
0x4000607C	GP2SET	GPIO Port 2 data out set	0x00	W
0x40006080	GP2CLR	GPIO Port 2 data out clear	0x00	W
0x40006084	GP2TGL	GPIO Port 2 pin toggle	0x00	W
0x40006090	GP3CON	GPIO Port 3 configuration	0x0000	RW
0x40006094	GP3OEN	GPIO Port 3 output enable	0x00	RW
0x40006098	GP3PUL	GPIO Port 3 output pull-up enable	0xFF	RW
0x4000609C	GP3OCE	GPIO Port 3 open collector	0x00	RW
0x400060A4	GP3IN	GPIO Port 3 data input	0xFF	R
0x400060A8	GP3OUT	GPIO Port 3 data out	0x00	RW
0x400060AC	GP3SET	GPIO Port 3 data out set	0x00	W
0x400060B0	GP3CLR	GPIO Port 3 data out clear	0x00	W
0x400060B4	GP3TGL	GPIO Port 3 pin toggle	0x00	W
0x400060C0	GP4CON	GPIO Port 4 configuration	0x0000	RW
0x400060C4	GP4OEN	GPIO Port 4 output enable	0x00	RW
0x400060C8	GP4PUL	GPIO Port 4 output pull-up enable	0xFF	RW
0x400060CC	GP4OCE	GPIO Port 4 open collector	0x00	RW
0x400060D4	GP4IN	GPIO Port 4 data input	0xFF	R
0x400060D8	GP4OUT	GPIO Port 4 data out	0x00	RW
0x400060DC	GP4SET	GPIO Port 4 data out set	0x00	W
0x400060E0	GP4CLR	GPIO Port 4 data out clear	0x00	W
0x400060E4	GP4TGL	GPIO Port 4 pin toggle	0x00	W
0x400060F0	GPDWN	GPIO P3.4 pull-down control	0x01	RW
0x40008824	RFTST	Internal radio test mode access	0x0000	RW

REGISTER DETAILS (GENERAL-PURPOSE INPUT OUTPUT)

GPIO Port Configuration Registers

GP0CON Address: 0x40006000, GP1CON Address: 0x40006030, GP2CON Address: 0x40006060, GP3CON Address: 0x40006090, GP4CON Address: 0x400060C0, Reset: 0x0000, Name: GPxCON

GPIO Port Configuration Registers

Table 158. Bit Descriptions for GPxCON

Bits	Bit Name	Description	Access
[15:14]	CON7	Configuration bits for Px.7 as per Table 156.	RW
[13:12]	CON6	Configuration bits for Px.6 as per Table 156.	RW
[11:10]	CON5	Configuration bits for Px.5 as per Table 156.	RW
[9:8]	CON4	Configuration bits for Px.4 as per Table 156.	RW
[7:6]	CON3	Configuration bits for Px.3 as per Table 156.	RW
[5:4]	CON2	Configuration bits for Px.2 as per Table 156.	RW
[3:2]	CON1	Configuration bits for Px.1 as per Table 156.	RW
[1:0]	CON0	Configuration bits for Px.0 as per Table 156.	RW

GPIO Port Output Enable Registers

GP0OEN Address: 0x40006004, GP1OEN Address: 0x40006034, GP2OEN Address: 0x40006064, GP3OEN Address: 0x40006094, GP4OEN Address: 0x400060C4, Reset: 0x00, Name: GPxOEN

Table 159. Bit Descriptions for GPxOEN

Bits	Bit Name	Description	Access
7	OEN7	Port pin direction. 0: IN. Configures pin as an input only. Disables the output on the corresponding port pin. 1: OUT. Enables the corresponding port pin as an output. The input state/interrupt is still available unless GPxOCE is set. See Table 155.	RW
6	OEN6	Port pin direction. 0: IN. Configures pin as an input only. Disables the output on the corresponding port pin. 1: OUT. Enables the corresponding port pin as an output. The input state/interrupt is still available unless GPxOCE is set. See Table 155.	RW
5	OEN5	Port pin direction. 0: IN. Configures pin as an input only. Disables the output on the corresponding port pin. 1: OUT. Enables the corresponding port pin as an output. The input state/interrupt is still available unless GPxOCE is set. See Table 155.	RW
4	OEN4	Port pin direction. 0: IN. Configures pin as an input only. Disables the output on the corresponding port pin. 1: OUT. Enables the corresponding port pin as an output. The input state/interrupt is still available unless GPxOCE is set. See Table 155.	RW
3	OEN3	Port pin direction. 0: IN. Configures pin as an input only. Disables the output on the corresponding port pin. 1: OUT. Enables the corresponding port pin as an output. The input state/interrupt is still available unless GPxOCE is set. See Table 155.	RW
2	OEN2	Port pin direction. 0: IN. Configures pin as an input only. Disables the output on the corresponding port pin. 1: OUT. Enables the corresponding port pin as an output. The input state/interrupt is still available unless GPxOCE is set. See Table 155.	RW

Bits	Bit Name	Description	Access
1	OEN1	Port pin direction. 0: IN. Configures pin as an input only. Disables the output on the corresponding port pin. 1: OUT. Enables the corresponding port pin as an output. The input state/interrupt is still available unless GPxOCE is set. See Table 155.	RW
0	OEN0	Port pin direction. 0: IN. Configures pin as an input only. Disables the output on the corresponding port pin. 1: OUT. Enables the corresponding port pin as an output. The input state/interrupt is still available unless GPxOCE is set. See Table 155.	RW

GPIO Port Pull-Up Enable Registers

GP0PUL Address: 0x40006008, GP1PUL Address: 0x40006038, GP2PUL Address: 0x40006068, GP3PUL Address: 0x40006098, GP4PUL Address: 0x400060C8, Reset: 0xFF, Name: GPxPUL

Table 160. Bit Descriptions for GPxPUL

Bits	Bit Name	Description	Access
7	PUL7	Pull-up enable for port pin. (Not available for port GPIO1). 0: DIS. Disables the internal pull up on corresponding port pin. 1: EN. Enables the internal pull up on corresponding port pin.	RW
6	PUL6	Pull-up enable for port pin. 0: DIS. Disables the internal pull up on corresponding port pin. 1: EN. Enables the internal pull up on corresponding port pin.	RW
5	PUL5	Pull-up enable for port pin. 0: DIS. Disables the internal pull up on corresponding port pin. 1: EN. Enables the internal pull up on corresponding port pin.	RW
4	PUL4	Pull-up enable for port pin. 0: DIS. Disables the internal pull up on corresponding port pin. 1: EN. Enables the internal pull up on corresponding port pin.	RW
3	PUL3	Pull-up enable for port pin. 0: DIS. Disables the internal pull up on corresponding port pin. 1: EN. Enables the internal pull up on corresponding port pin.	RW
2	PUL2	Pull-up enable for port pin. 0: DIS. Disables the internal pull up on corresponding port pin. 1: EN. Enables the internal pull up on corresponding port pin.	RW
1	PUL1	Pull-up enable for port pin. 0: DIS. Disables the internal pull up on corresponding port pin. 1: EN. Enables the internal pull up on corresponding port pin.	RW
0	PUL0	Pull-up enable for port pin. 0: DIS. Disables the internal pull up on corresponding port pin. 1: EN. Enables the internal pull up on corresponding port pin.	RW

GPIO Port Open Collector Registers

GP0OCE Address: 0x4000600C, GP1OCE Address: 0x4000603C, GP2OCE Address: 0x4000606C, GP3OCE Address: 0x4000609C, GP4OCE Address: 0x400060CC, Reset: 0x00, Name: GPxOCE

Table 161. Bit Descriptions for GPxOCE

Bits	Bit Name	Description	Access
7	OCE7	Open-collector enable for port pin. This register only affects GPIOs configured as output (GPxOEN = 1). 0: DIS. The corresponding port pin has two output states, logic 1 and logic 0 depending on the contents of GPxOUT. The input state/interrupt is always available. 1: EN. The corresponding port pin has two output states, logic 0 when GPxOUT = 0, and high impedance when GPxOUT = 1. See Table 155.	RW
6	OCE6	Open-collector enable for port pin. This register only affects GPIOs configured as output (GPxOEN = 1). 0: DIS. The corresponding port pin has two output states, logic 1 and logic 0 depending on the contents of GPxOUT. The input state/interrupt is always available. 1: EN. The corresponding port pin has two output states, logic 0 when GPxOUT = 0, and high impedance when GPxOUT = 1. See Table 155.	RW
5	OCE5	Open-collector enable for port pin. This register only affects GPIOs configured as output (GPxOEN = 1). 0: DIS. The corresponding port pin has two output states, logic 1 and logic 0 depending on the contents of GPxOUT. The input state/interrupt is always available. 1: EN. The corresponding port pin has two output states, logic 0 when GPxOUT = 0, and high impedance when GPxOUT = 1. See Table 155.	RW
4	OCE4	Open-collector enable for port pin. This register only affects GPIOs configured as output (GPxOEN = 1). 0: DIS. The corresponding port pin has two output states, logic 1 and logic 0 depending on the contents of GPxOUT. The input state/interrupt is always available. 1: EN. The corresponding port pin has two output states, logic 0 when GPxOUT = 0, and high impedance when GPxOUT = 1. See Table 155.	RW
3	OCE3	Open collector enable for port pin. This register only affects GPIOs configured as output (GPxOEN = 1). 0: DIS. The corresponding port pin has two output states, logic 1 and logic 0 depending on the contents of GPxOUT. The input state/interrupt is always available. 1: EN. The corresponding port pin has two output states, logic 0 when GPxOUT = 0, and high impedance when GPxOUT = 1. See Table 155.	RW
2	OCE2	Open collector enable for port pin. This register only affects GPIOs configured as output (GPxOEN = 1). 0: DIS. The corresponding port pin has two output states, logic 1 and logic 0 depending on the contents of GPxOUT. The input state/interrupt is always available. 1: EN. The corresponding port pin has two output states, logic 0 when GPxOUT = 0, and high impedance when GPxOUT = 1. See Table 155.	RW
1	OCE1	Open collector enable for port pin. This register only affects GPIOs configured as output (GPxOEN = 1). 0: DIS. The corresponding port pin has two output states, logic 1 and logic 0 depending on the contents of GPxOUT. The input state/interrupt is always available. 1: EN. The corresponding port pin has two output states, logic 0 when GPxOUT=0, and high impedance when GPxOUT = 1. See Table 155.	RW
0	OCE0	Open collector enable for port pin. This register only affects GPIOs configured as output (GPxOEN = 1). 0: DIS. The corresponding port pin has two output states, logic 1 and logic 0 depending on the contents of GPxOUT. The input state/interrupt is always available. 1: EN. The corresponding port pin has two output states, logic 0 when GPxOUT = 0, and high impedance when GPxOUT = 1. See Table 155.	RW

GPIO Port Data Input Registers

GP0IN Address: 0x40006014, GP1IN Address: 0x40006044, GP2IN Address: 0x40006074, GP3IN Address: 0x400060A4, GP4IN Address: 0x400060D4, Reset: 0xFF, Name: GPxIN

The contents of the GPxIN register depend on the digital level on the corresponding pins.

Table 162. Bit Descriptions for GPxIN

Bits	Bit Name	Description	Access
7	IN7	Reflects the level on the port pin except when corresponding GPxOCE and GPxOEN bits are set. 0: LOW 1: HIGH	R
6	IN6	Reflects the level on the port pin except when corresponding GPxOCE and GPxOEN bits are set. 0: LOW 1: HIGH	R
5	IN5	Reflects the level on the port pin except when corresponding GPxOCE and GPxOEN bits are set. 0: LOW 1: HIGH	R
4	IN4	Reflects the level on the port pin except when corresponding GPxOCE and GPxOEN bits are set. 0: LOW 1: HIGH	R
3	IN3	Reflects the level on the port pin except when corresponding GPxOCE and GPxOEN bits are set. 0: LOW 1: HIGH	R
2	IN2	Reflects the level on the port pin except when corresponding GPxOCE and GPxOEN bits are set. 0: LOW 1: HIGH	R
1	IN1	Reflects the level on the port pin except when corresponding GPxOCE and GPxOEN bits are set. 0: LOW 1: HIGH	R
0	IN0	Reflects the level on the port pin except when corresponding GPxOCE and GPxOEN bits are set. 0: LOW 1: HIGH	R

GPIO Port Data Out Register

GP0OUT Address: 0x40006018, GP1OUT Address: 0x40006048, GP2OUT Address: 0x40006078, GP3OUT Address: 0x400060A8, GP4OUT Address: 0x400060D8, Reset: 0x00, Name: GPxOUT

Table 163. Bit Descriptions for GPxOUT

Bits	Bit Name	Description	Access
7	OUT7	Data out register. 0: LOW. Cleared by user to drive the corresponding port pin low. 1: HIGH. Set by user code to drive the corresponding port pin high.	RW
6	OUT6	Data out register. 0: LOW. Cleared by user to drive the corresponding port pin low. 1: HIGH. Set by user code to drive the corresponding port pin high.	RW
5	OUT5	Data out register. 0: LOW. Cleared by user to drive the corresponding port pin low. 1: HIGH. Set by user code to drive the corresponding port pin high.	RW
4	OUT4	Data out register. 0: LOW. Cleared by user to drive the corresponding port pin low. 1: HIGH. Set by user code to drive the corresponding port pin high.	RW
3	OUT3	Data out register. 0: LOW. Cleared by user to drive the corresponding port pin low. 1: HIGH. Set by user code to drive the corresponding port pin high.	RW

Bits	Bit Name	Description	Access
2	OUT2	Data out register. 0: LOW. Cleared by user to drive the corresponding port pin low. 1: HIGH. Set by user code to drive the corresponding port pin high.	RW
1	OUT1	Data out register. 0: LOW. Cleared by user to drive the corresponding port pin low. 1: HIGH. Set by user code to drive the corresponding port pin high.	RW
0	OUT0	Data out register. 0: LOW. Cleared by user to drive the corresponding port pin low. 1: HIGH. Set by user code to drive the corresponding port pin high.	RW

GPIO Port Data Out Set Registers

GP0SET Address: 0x4000601C, GP1SET Address: 0x4000604C, GP2SET Address: 0x4000607, GP3SET Address: 0x400060AC, GP4SET Address: 0x400060DC, Reset: 0x00, Name: GPxSET

Table 164. Bit Descriptions for GPxSET

Bits	Bit Name	Description	Access
7	SET7	Set output high. 0: No effect. 1: SET. Set by user code to drive the corresponding port pin high.	W
6	SET6	Set output high. 0: No effect. 1: SET. Set by user code to drive the corresponding port pin high.	W
5	SET5	Set output high. 0: No effect. 1: SET. Set by user code to drive the corresponding port pin high.	W
4	SET4	Set output high. 0: No effect. 1: SET. Set by user code to drive the corresponding port pin high.	W
3	SET3	Set output high. 0: No effect. 1: SET. Set by user code to drive the corresponding port pin high.	W
2	SET2	Set output high. 0: No effect. 1: SET. Set by user code to drive the corresponding port pin high.	W
1	SET1	Set output high. 0: No effect. 1: SET. Set by user code to drive the corresponding port pin high. 0: No effect.	W
0	SET0	Set output high. 1: SET. Set by user code to drive the corresponding port pin high.	W

GPIO Port Data Out Clear Registers

GP0CLR Address: 0x40006020, GP1CLR Address: 0x40006050, GP2CLR Address: 0x40006080, GP3CLR Address: 0x400060B0, GP4CLR Address: 0x400060E0, Reset: 0x00, Name: GPxCLR

Table 165. Bit Descriptions for GPxCLR

Bits	Bit Name	Description	Access
7	CLR7	Data out clear. 0: No effect 1: CLR. Set by user code to drive the corresponding port pin low.	W
6	CLR6	Data out clear. 0: No effect 1: CLR. Set by user code to drive the corresponding port pin low.	W
5	CLR5	Data out clear. 0: No effect 1: CLR. Set by user code to drive the corresponding port pin low.	W
4	CLR4	Data out clear. 0: No effect 1: CLR. Set by user code to drive the corresponding port pin low.	W
3	CLR3	Data out clear. 0: No effect 1: CLR. Set by user code to drive the corresponding port pin low.	W
2	CLR2	Data out clear. 0: No effect 1: CLR. Set by user code to drive the corresponding port pin low.	W
1	CLR1	Data out clear. 0: No effect 1: CLR. Set by user code to drive the corresponding port pin low.	W
0	CLR0	Data out clear. 0: No effect 1: CLR. Set by user code to drive the corresponding port pin low.	W

GPIO Port Pin Toggle Registers

GP0TGL Address: 0x40006024, GP1TGL Address: 0x40006054, GP2TGL Address: 0x40006084, GP3TGL Address: 0x400060B4, GP4TGL Address: 0x400060E4, Reset: 0x00, Name: GPxTGL

Table 166. Bit Descriptions for GPxTGL

Bits	Bit Name	Description	Access
7	TGL7	Toggle pin. 0: No effect 1: TGL. Set by user code to invert the corresponding port pin.	W
6	TGL6	Toggle pin. 0: No effect 1: TGL. Set by user code to invert the corresponding port pin.	W
5	TGL5	Toggle pin. 0: No effect 1: TGL. Set by user code to invert the corresponding port pin.	W
4	TGL4	Toggle pin. 0: No effect 1: TGL. Set by user code to invert the corresponding port pin.	W
3	TGL3	Toggle pin. 0: No effect 1: TGL. Set by user code to invert the corresponding port pin.	W
2	TGL2	Toggle pin. 0: No effect 1: TGL. Set by user code to invert the corresponding port pin.	W
1	TGL1	Toggle pin. 0: No effect 1: TGL. Set by user code to invert the corresponding port pin.	W
0	TGL0	Toggle pin. 0: No effect 1: TGL. Set by user code to invert the corresponding port pin.	W

GPIO P3.4 Pull Down Control Register

Address: 0x400060F0, Reset: 0x01, Name: GPDWN

Table 167. Bit Descriptions for GPDWN

Bits	Bit Name	Description	Access
[7:2]	RESERVED	Reserved.	RW
1	DWN1	Pull down resistor control bit. 0: EN to enable the pull down resistor on P3.4 by software. The hardware enables this pull down automatically at power up. 1: DIS to disable the pull down resistor on P3.4. Disabled automatically by hardware if GP3PUL[4] =1 or if GP3OEN[4] =1.	RW
0	RESERVED	Reserved. 1 should be written to this bit.	RW

RF Receive Test Mode Register

Address: 0x40008824, Reset: 0x0000, Name: RFTST

Table 168. Bit Descriptions for RFTST

Bits	Bit Name	Description	Access
[15:5]	DIR	Controls the pins for RF receive test mode. 00000111111 should be written to these bits.	RW
[4:1]	RESERVED	Reserved. 0 should be written to these bits.	R
0	GPX	Connect the internal raw data and clock of the RF transceiver to external GPIOs P2.6 and P0.06. 0: DIS 1: EN. Enables data and clock on P2.6 and P0.6.	RW

Note that enabling data and clock on P2.6 and P0.6 automatically configures these pins as outputs. This overrides the selections made in the associated GPxCON and GPxOEN registers. In addition, Pins P0.7, P1.0, P1.1, P1.4, and P1.5 are outputs and assume different functionality. This overrides the selections made in the associated GPxCON and GPxOEN registers.

I²C SERIAL INTERFACE

I²C FEATURES

- Compliance to the Philips Semiconductors I²C specifications V2.1.
- Master or slave mode with 2-byte transmit and receive FIFOs
- Supports
  - 7-bit and 10-bit addressing for the slave and the master
  - Four 7-bit device addresses or one 10-bit address and two 7-bit addresses in the slave
  - Repeated starts for the slave and the master
  - Clock stretching for the slave and the master
  - Master arbitration
  - Continuous read mode for the master or up to 512 bytes fixed read
  - Internal and external loopback modes
- Support for DMA in master and slave modes
- Software control on the slave of NACK signal
- I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).

I²C OVERVIEW

The I²C data transfer uses a serial clock (I2CSCL) pin and a serial data (I2CSDA) pin. The pins are configured in a wired-AND'ed format that allows arbitration in a multimaster system. Both I2CSDA and I2CSCL are bidirectional and must be connected to a positive supply voltage using a pull-up resistor. Typical pull-up values are 10 kΩ.

The transfer sequence of an I²C system consists of a master device initiating a transfer by generating a start condition while the bus is idle. The master transmits the slave device address and the direction of the data transfer during the initial address transfer. If the master does not lose arbitration and the slave acknowledges, the data transfer is initiated. This continues until the master issues a stop condition and the bus becomes idle. Figure 49 shows a typical I²C transfer.

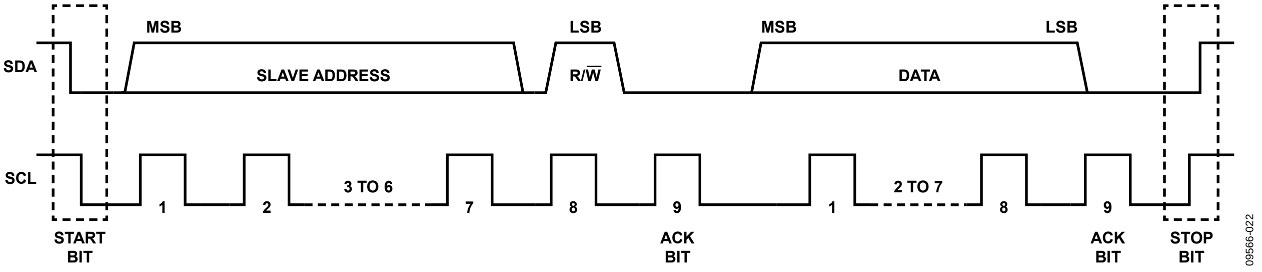


Figure 49. Typical I²C Transfer Sequence

The I²C bus peripheral address in the I²C bus system is programmed by the user. This ID can be modified any time a transfer is not in progress. The user can configure the interface to respond to four slave addresses.

The I²C peripheral can be configured only as a master or slave at any given time. The peripheral is implemented with a 2-byte FIFO for each transmit and receive shift register.

I²C OPERATION

Initialization

The pins P1.4 and P1.5 used for I²C communication must be configured in I²C mode (configuration mode 01) before enabling the I²C peripheral. Their internal pull-up resistors on the I²C pins should be disabled using GPxPUL. The clock for the I²C should also be enabled in the CLKACT register.

Addressing Modes

7-Bit Addressing

The I2CID0, I2CID1, I2CID2, and I2CID3 registers contain the slave device IDs. The device compares the four I2CIDx registers to the address byte. To be correctly addressed, the seven MSBs of either ID register must be identical to that of the seven MSBs of the first received address byte. The LSB of the ID registers (the transfer direction bit) is ignored in the process of address recognition.

The master addresses a device using the I2CADR0 register.

10-Bit Addressing

This feature is enabled by setting I2CSCON[1] for master and slave mode.

The 10-bit address of the slave is stored in I2CID0 and I2CID1, where I2CID0 contains the first byte of the address, and the RW bit and the upper five bits must be programmed to 11110 as shown in Figure 50. I2CID1 contains the remaining eight bits of the 10-bit address. I2CID2 and I2CID3 can still be programmed with 7-bit addresses.

The master communicates to a 10-bit address slave using the I2CADR0/1 registers. The format is described in Figure 50.

I2CADR0 AND I2CID0								I2CADR1 AND I2CID1							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	1	1	1	0	2 MSB		R/W	8 LSB							

09566-023

Figure 50. 10-Bit Address Format

Repeated Start Condition

A repeated start condition occurs when a second start condition is sent to a slave without a stop condition being sent in between. This allows the master to reverse the direction of the transfer, by changing the R/$\overline{W}$ bit without having to give up control of the bus.

An example of a transfer sequence is shown in Figure 51. This is generally used where the first data sent to the part sets up the register address to be read from.

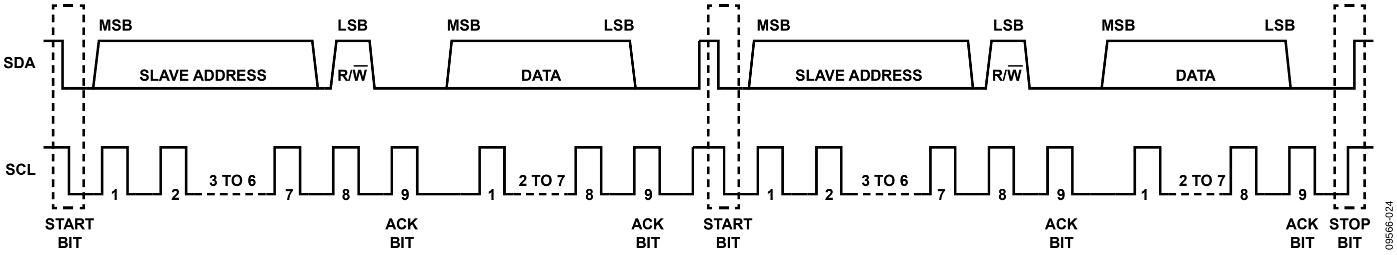


Figure 51. I^{2}C Repeated Start Sequence

On the slave side, an interrupt is generated (if enabled in the I2CSCON register) when a repeated start and a slave address are received. This can be differentiated from receiving a start and slave address by using the status bits START and REPSTART in the I2CSSTA MMR.

On the master side, the master generates a repeated start if the I2CADR0 register is written while the master is still busy with a transaction. After the state machine has started to transmit the device address, it is safe to write to the I2CADR0 register.

For example, if a transaction involving a read, repeated start, or read/write is required, then write to the I2CADR0 register after the state machine starts to transmit the device address or after the first TXREQ interrupt is received. When the transmit FIFO empties, a repeated start is generated.

Similarly, if a transaction involving a read, repeated start, or read/write is required, write to the first master address byte register, I2CADR1, either after the state machine starts to transmit the device address or after the first RXREQ interrupt is received. When the requested 'receive count' is reached, a repeated start is generated.

Serial Clock Generation

The I^{2}C master in the system generates the serial clock for a transfer. The master channel can be configured to operate in fast mode (400 kHz) or standard mode (100 kHz).

The bit rate is defined in the I2CDIV MMR as follows:

$$F_{i2cscl} = f_{UCLK} / (LOW + HIGH + 3)$$

where:

f_{UCLK} = clock before the clock divider.
HIGH is the high period of the clock, I2CDIV[15:8] = (REQD_HIGH_TIME/UCLK_PERIOD) – 2.
LOW is the low period of the clock, I2CDIV[7:0] = (REQD_LOW_TIME/UCLK_PERIOD) − 1.

For 100 kHz I2CSCL operation, with a low time of 5000 ns and a high time of 5000 ns, and a UCLK frequency of 16 MHz,

LOW= (5000 ns/(1/16000000)) − 1 = 79 = 0x4F

$HIGH = (5000\ \text{ns}/(1/16000000)) - 2 = 78 = \text{0x4E}$

$F_{i2cscl} = f_{UCLK}/(\text{LOW} + HIGH + 3) = 16000000/(79+78+3) = 100\ \text{kHz}$

For 400 kHz I2CSCL operation, with a low time of 1250 ns and a high time of 1250 ns, and a UCLK frequency of 16 MHz,

$LOW = (1250\ \text{ns}/(1/16000000)) - 1 = 19 = \text{0x13}$

$HIGH = (1250\ \text{ns}/(1/16000000)) - 2 = 18 = \text{0x12}$

$\text{F}_{\text{i2cscl}} = \text{f}_{\text{UCLK}}/(\text{LOW} + \text{HIGH} + 3) = 16000000/(19 + 18 + 3)\ 400\ \text{kHz}$

The I2CxDIV registers correspond to HIGH: LOW. A master device can be configured to generate the serial clock. The frequency is programmed by the user in the serial clock divisor register.

Clock Stretching

An I^{2}C slave is allowed to hold down the clock if it needs to reduce the bus speed. This is done by a mechanism referred to as clock stretching. The STRECH bit in the slave control register provides this feature.

Operating Modes

Master Transfer Initiation

If the master enable bit (I2CMCON[0], MAS) is set, a master transfer sequence is initiated by writing a value to the I2CADRx register. If there is valid data in the I2CMTX register, it is the first byte transferred in the sequence after the address byte during a write sequence.

Slave Transfer Initiation

If the slave enable bit (I2CSCON[0], SLV) is set, a slave transfer sequence is monitored for the device address in Register I2CID0, Register I2CID1, Register I2CID2, or Register I2CID3. If the device address is recognized, the part participates in the slave transfer sequence.

Note that a slave operation always starts with the assertion of one of three interrupt sources—read request (RXREQ), write request (TXREQ), or general call (GCINT) interrupt—and the software should always look for a stop interrupt to ensure that the transaction has completed correctly and to deassert the stop interrupt status bit.

Rx/Tx Data FIFOs

The transmit data path consists of a master and slave Tx FIFO, each two bytes deep, I2CMTX and I2CSTX, and a transmit shifter. The transmit FIFO status bits in I2CMSTA[1:0] and I2CSSTA[0] denote whether there is valid data in the Tx FIFO. Data from the Tx FIFO is loaded into the Tx shifter when a serial byte begins transmission. If the Tx FIFO is not full during an active transfer sequence, the transmit request bit (TXREQ) in I2CMSTA or I2CSSTA asserts.

In the slave, if there is no valid data to transmit when the Tx shifter is loaded, the transmit underrun status bit (TXUR) asserts (I2CMSTA[12], ISCSSTA[1]).

The master generates a stop condition if there is no data in the transmit FIFO and the master is writing data.

The receive data path consists of a master and slave Rx FIFO, each two bytes deep, I2CMRX and I2CSRX. The receive request interrupt bit (RXREQ) in I2CMSTA or I2CSSTA indicates whether there is valid data in the Rx FIFO. Data is loaded into the Rx FIFO after each byte is received. If valid data in the Rx FIFO is overwritten by the Rx shifter, the receive overflow status bit (RXOF) is asserted (I2CMSTA[9] or I2CSSTA[4]).

There is one shadow register outside the Tx FIFOs into which data is loaded before being transmitted. The Tx FIFO status can be set up to decrement when a byte is unloaded from the Tx FIFO into the shadow register or to decrement when the byte is transmitted. The advantage of decrementing the FIFO status when it is unloaded from the FIFO is that a transmit interrupt asserts earlier and the microcontroller has more time to respond to the interrupt and write a byte to the FIFO.

Master NACK

When receiving data, the master responds with a NACK if its FIFO is full and an attempt is made to write another byte to the FIFO. This last byte received is not written to the FIFO and is lost.

No Acknowledge from the Slave

If the slave does not want to acknowledge a read access, then simply not writing data into the slave transmit FIFO results in a NACK.

If the slave does not want to acknowledge a master write, assert the NACK bit in the slave control register (I2CSCON[7]).

Normally, the slave will ACK all bytes that are written into the receive FIFO. If the receive FIFO fills up the slave cannot write further bytes to it and it will not acknowledge the byte that was not written to the FIFO. The master should then stop the transaction.

The slave does not acknowledge a matching device address if the direction bit (I2CADR0) is 1 (read) and the transmit FIFO is empty. Therefore, there is very little time for the microcontroller to respond to a slave transmit request and the assertion of ACK. It is recommended that EARLYTXR be asserted for this reason.

General Call

If the general call enable bit, GC (I2CSCON[2]) and the slave enable bit, SLVEN (I2CSCON[0]), are set, the device responds to a general call.

An I^{2}C general call is for addressing every device on the I^{2}C bus. A general call address is 0x00 or 0x01. The first byte, address byte is followed by a command byte.

If the address byte is 0x00, then Byte 2, the command byte, can be one of the following:

- 0x6: the I^{2}C interface (master and slave) is reset. The general call interrupt status asserts, and the general call ID bits, GCID (I2CSSTA[9:8]), are 0x1. User code should take corrective action to reset the entire system or simply reenable the I^{2}C interface.
- 0x4: the general call interrupt status bit is asserted, and the general call ID bits (GCID) are 0x2.

If the address byte is 0x01, a hardware general call is issued.

- Byte 2 in this case is the hardware master address.

The general call interrupt status bit is set on any general call after the second byte is received, and user code should take corrective action to reprogram the device address.

If GCis asserted, the slave always acknowledges the first byte of a general call. It acknowledges the second byte of a general call if the second byte is 0x04 or 0x06 or if the second byte is a hardware general call and HGC (I2CSCON[3]) is asserted.

The I2CALT register contains the alternate device ID for a hardware general call sequence. If the hardware general call enable bit, HGC, GC, and SLV are all set, the device recognizes a hardware general call. When a general call sequence is issued and the second byte of the sequence is identical to ALT, the hardware call sequence is recognized for the device.

I^{2}C Reset Mode

The slave state machine is reset when SLV is written to 0 (I2CSCON[0]).

The master state machine is reset when MAS is written to 0 (I2CMCON[0]).

I^{2}C Test Modes

The device can be placed in an internal loopback mode by setting the LOOPBACK bit (I2CMCON[2]). There are four FIFOs (master Tx and Rx, and slave Tx and Rx) so, in effect, the I^{2}C peripheral can be set up to talk to itself. External loopback can be performed if the master is set up to address the slave's address.

I^{2}C Low Power Mode

If the master and slave are both disabled (MAS = SLV = 0), the I^{2}C section is off. To fully power down the I^{2}C block, the clock to the I^{2}C section of the chip should be disabled.

DMA Requests

Four DMA channels are required to service the I^{2}C master and slave. DMA enable bits are provided in the slave control register and in the master control register.

REGISTER SUMMARY (I^{2}C)

Table 169. I^{2}C Register Summary

Address	Name	Description	Reset	RW
0x40003000	I2CMCON	Master Control Register	0x0000	RW
0x40003004	I2CMSTA	Master Status Register	0x0000	R
0x40003008	I2CMRX	Master Receive Data Register	0x0000	R
0x4000300C	I2CMTX	Master Transmit Data Register	0x0000	W
0x40003010	I2CMRXCNT	Master Receive Data Count Register	0x0000	RW
0x40003014	I2CMCRXCNT	Master Current Receive Data Count Register	0x0000	R
0x40003018	I2CADR0	First Master Address Byte Register	0x00	RW
0x4000301C	I2CADR1	Second Master Address Byte Register	0x00	RW
0x40003024	I2CDIV	Serial Clock Period Divisor Register	0x1F1F	RW
0x40003028	I2CSCON	Slave Control Register	0x0000	RW
0x4000302C	I2CSSTA	Slave I^{2}C Status, Error and Interrupt Register	0x0001	R
0x40003030	I2CSRX	Slave Receive Data Register	0x0000	R
0x40003034	I2CSTX	Slave Transmit Data Register	0x0000	W
0x40003038	I2CALT	Hardware General Call ID Register	0x0000	RW
0x4000303C	I2CID0	First Slave Address Device ID	0x0000	RW
0x40003040	I2CID1	Second Slave Address Device ID	0x0000	RW
0x40003044	I2CID2	Third Slave Address Device ID	0x0000	RW
0x40003048	I2CID3	Fourth Slave Address Device ID	0x0000	RW
0x4000304C	I2CFSTA	Master and Slave Rx/Tx FIFO Status Register	0x0000	RW

REGISTER DETAILS (I^{2}C)

Master Control Register

Address: 0x40003000, Reset: 0x0000, Name: I2CMCON

Table 170. Bit Descriptions for I2CMCON

Bits	Bit Name	Description	Reset	Access
[15:12]	RESERVED	Reserved.	0x0	R
11	TXDMA	Enable master Tx DMA request. 0: DIS. Disable Tx DMA mode. 1: EN. Enable I^{2}C master DMA requests.	0x0	RW
10	RXDMA	Enable master Rx DMA request. 0: DIS. Disable Rx DMA mode. 1: EN. Enable I^{2}C master DMA requests.	0x0	RW
9	RESERVED	Reserved.	0x0	R
8	IENCMP	Transaction completed (or stop detected) interrupt enable. 0: DIS. Interrupt disabled. 1: EN. Interrupt enabled. A master I^{2}C interrupt is generated when a STOP is detected. Enables TCOMP to generate an interrupt.	0x0	RW
7	IENNACK	NACK received interrupt enable. 0: DIS. Interrupt disabled. 1: EN, enables NACKADDR(I2CMSTA[4]) and NACKDATA (I2CMSTA[7]) to generate an interrupt.	0x0	RW
6	IENALOST	Arbitration lost interrupt enable. 0: DIS. Interrupt disabled. 1: EN. Interrupt enabled. A master I^{2}C interrupt is generated if the master loses arbitration. Enables ALOST to generate an interrupt.	0x0	RW

Bits	Bit Name	Description	Reset	Access
5	IENTX	Transmit request interrupt enable. 0: DIS. Interrupt disabled. 1: EN. Interrupt enabled. A master I^{2}C interrupt is generated when the Tx FIFO is not full and the direction bit is 0.	0x0	RW
4	IENRX	Receive request interrupt enable. 0: DIS. Interrupt disabled. 1: EN. Interrupt enabled. A master I2C interrupt is generated when data is in the receive FIFO.	0x0	RW
3	STRETCH	Stretch I2CSCL enable. 0: DIS. Disable 1: EN. Setting this bit instructs the device that if I2CSCL is 0, hold it at 0 or if I2CSCL is 1, then when it next goes to 0, hold it at 0.	0x0	RW
2	LOOPBACK	Internal loop back enable. 0: DIS. Disable. 1: EN. I2CSCL and I2CSDA out of the device are muxed onto their corresponding inputs. Note that is also possible for the master to loop back a transfer to the slave as long as the device address corresponds, that is, external loopback.	0x0	RW
1	COMPETE	Start back-off disable. 0: DIS. Disable. 1: EN. Enables the device to compete for ownership even if another device is currently driving a start condition.	0x0	RW
0	MAS	Master enable bit. 0: DIS. The master is disabled. The master state machine is reset. The master should be disabled when not in use. This bit should not be cleared until a transaction has completed. TCOMP in I2CMSTA indicates when a transaction is complete. 1: EN. Enable master.	0x0	RW

Master Status Register

Address: 0x40003004, Reset: 0x0000, Name: I2CMSTA

Table 171. Bit Descriptions for I2CMSTA

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved.	0x0	R
12	TXUR	Master transmit FIFO underrun. 0: CLR. Cleared. 1: SET. Set when the I^{2}C master ends the transaction due to a Tx FIFO empty condition. This bit is only set when IENTX (I2CSCON[10]) is set.	0x0	R
11	MSTOP	STOP driven by the I^{2}C master. 0: CLR. Cleared. 1: SET. Set when the I^{2}C master drives a stop condition on the I^{2}C bus, therefore indicating a transaction completion, Tx underrun, Rx overflow, or a NACK by the slave. It is different from TCOMP because it is not set when the stop condition occurs due to any other master on the I^{2}C bus. This bit does not generate an interrupt. See the TCOMP description for available interrupts related to the stop condition.	0x0	R
10	LINEBUSY	Line is busy. 0: CLR. Cleared when a stop is detected on the I^{2}C bus. 1: SET. Set when a start is detected on the I^{2}C bus.	0x0	R
9	RXOF	Receive FIFO overflow. 0: CLR. Cleared. 1: SET. Set when a byte is written to the receive FIFO when the FIFO is already full.	0x0	R

Bits	Bit Name	Description	Reset	Access
8	TCOMP	Transaction completed (or stop detected). (Can drive an interrupt). 0: CLR. Cleared. 1: SET. Set when a STOP condition is detected on the I²C bus. If IENCMP is 1, an interrupt is generated when this bit asserts. This bit only asserts if the master is enabled (MAS = 1). This bit should be used to determine when it is safe to disable the master. It can also be used to wait for another master's transaction to complete on the I²C bus when this master loses arbitration.	0x0	R
7	NACKDATA	NACK received in response to data write. (Can drive an interrupt). 0: CLR. Cleared on a read of the I2CMSTA register. 1: SET. Set when a NACK is received in response to a data write transfer. If IENNACK is 1, an interrupt is generated when this bit asserts.	0x0	R
6	BUSY	Master busy. 0: CLR. Cleared if the state machine is idle or another device has control of the I²C bus. 1: SET. Set when the master state machine is servicing a transaction.	0x0	R
5	ALOST	Arbitration lost. (Can drive an interrupt). 0: CLR. Cleared on a read of the I2CMSTA register. 1: SET. Set if the master loses arbitration. If IENALOST is 1, an interrupt is generated when this bit asserts.	0x0	R
4	NACKADDR	NACK received in response to an address. (Can drive an interrupt). 0: CLR. Cleared on a read of the I2CMSTA register. 1: SET. Set if a NACK received in response to an address. If IENNACK is 1, an interrupt is generated when this bit asserts.	0x0	R
3	RXREQ	Receive Request. (Can drive an interrupt). 0: CLR. Cleared. 1: SET. Set when there is data in the receive FIFO. If IENRX is 1, an interrupt is generated when this bit asserts.	0x0	R
2	TXREQ	Transmit Request. (Can drive an interrupt). 0: CLR. Cleared when the transmit FIFO underrun condition is not met. 1: SET. Set when the direction bit is 0 and the transmit FIFO is either empty or not full. If IENTX is 1, an interrupt is generated when this bit asserts.	0x0	R
[1:0]	TXFSTA	Transmit FIFO Status. Notes: The meaning of these bits is different than MTXFSTA (I2CFSTA[5:4]). 00: EMPTY. FIFO empty. 10: ONEBYTE. 1 byte in FIFO. 11: FULL. FIFO full.	0x0	R

Master Receive Data Register

Address: 0x40003008, Reset: 0x0000, Name: I2CMRX

Table 172. Bit Descriptions for I2CMRX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	R
[7:0]	VALUE	Receive register. This register allows access to the receive data FIFO. The FIFO can hold two bytes.	0x00	R

Master Transmit Data Register

Address: 0x4000300C, Reset: 0x0000, Name: I2CMTX

Table 173. Bit Descriptions for I2CMTX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	W
[7:0]	VALUE	Transmit register. This register allows access to the transmit data FIFO. The FIFO can hold two bytes.	0x00	W

Master Receive Data Count Register

Address: 0x40003010, Reset: 0x0000, Name: I2CMRXCNT

Table 174. Bit Descriptions for I2CMRXCNT

Bits	Bit Name	Description	Reset	Access
[15:9]	RESERVED	Reserved.	0x00	R
8	EXTEND	Extended read: Use this bit if greater than 256 bytes are required on a read. For example, to receive 412 bytes, write 0x100 (EXTEND = 1) to this register (I2CMRXCNT). Wait for the first byte to be received, then check the I2CMCRXCNT register for every byte received thereafter. When I2CMCRXCNT returns to 0, 256 bytes have been received. Then, write 0x09C (412 – 256 = 156 decimal (equal to 0x9C) – with the EXTEND bit set to 0) to this register (I2CMRXCNT). 0: DIS 1: EN	0x0	RW
[7:0]	COUNT	Receive count. Program the number of bytes required minus one to this register. If just one byte is required write 0 to this register. If greater than 256 bytes are required, then use EXTEND.	0x00	RW

Master Current Receive Data Count Register

Address: 0x40003014, Reset: 0x0000, Name: I2CMCRXCNT

Table 175. Bit Descriptions for I2CMCRXCNT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	R
[7:0]	VALUE	Current receive count. This register gives the total number of bytes received so far. If 256 bytes are requested, then this register reads 0 when the transaction has completed.	0x00	R

First Master Address Byte Register

Address: 0x40003018, Reset: 0x00, Name: I2CADR0

Table 176. Bit Descriptions for I2CADR0

Bits	Bit Name	Description	Reset	Access
[7:0]	VALUE	Address byte. If a 7-bit address is required, then I2CADR0[7:1] is programmed with the address and I2CADR0[0] is programmed with the direction (read or write). If a 10-bit address is required then I2CADR0[7:3] is programmed with '11110', I2CADR0[2:1] is programmed with the two MSBs of the address, and, again, I2CADR0[0] is programmed with the direction bit (read or write).	0x00	RW

Second Master Address Byte Register

Address: 0x4000301C, Reset: 0x00, Name: I2CADR1

Table 177. Bit Descriptions for I2CADR1

Bits	Bit Name	Description	Reset	Access
[7:0]	VALUE	Address byte. This register is only required when addressing a slave with 10-bit addressing. I2CADR1[7:0] is programmed with the lower eight bits of the address.	0x00	RW

Serial Clock Period Divisor Register

Address: 0x40003024, Reset: 0x1F1F, Name: I2CDIV

Table 178. Bit Descriptions for I2CDIV

Bits	Bit Name	Description	Reset	Access
[15:8]	HIGH	Serial clock high time. This register controls the clock high time. See the Serial Clock Generation section for more details.	0x1F	RW
[7:0]	LOW	Serial clock low time. This register controls the clock low time. See the Serial Clock Generation section for more details.	0x1F	RW

Slave Control Register

Address: 0x40003028, Reset: 0x0000, Name: I2CSCON

Table 179. Bit Descriptions for I2CSCON

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
14	TXDMA	Enable slave Tx DMA request. 0: DIS. Disable DMA mode. 1: EN. Enable I^{2}C slave DMA requests.	0x0	RW
13	RXDMA	Enable slave Rx DMA request. 0: DIS. Disable DMA mode. 1: EN. Enable I^{2}C slave DMA requests.	0x0	RW
12	IENREPST	Repeated start interrupt enable. 0: DIS. Disable an interrupt when the REPSTART status bit asserts. 1: EN. Generate an interrupt when the REPSTART status bit asserts.	0x0	RW
11	RESERVED	Reserved. 0 should be written to this bit.	0x0	RW
10	IENTX	Transmit request interrupt enable. 0: DIS. Disable transmit request interrupt. 1: EN. Enable transmit request interrupt.	0x0	RW
9	IENRX	Receive request interrupt enable. 0: DIS. Disable receive request interrupt. 1: EN. Enable receive request interrupt.	0x0	RW
8	IENSTOP	Stop condition detected interrupt enable. 0: DIS. Disable stop condition detect interrupt. 1: EN. Enable stop condition detect interrupt. Enables STOP (I2CSSTA[10]) to generate an interrupt	0x0	RW
7	NACK	NACK next communication. 0: DIS. Disable. 1: EN. Allow the next communication to be NACK'ed. This can be used, for example, if during a 24xx I2C serial EEPROM-style access, an attempt was made to write to a read only or nonexisting location in system memory. That is the indirect address in a 24xx I2C serial eeprom style write pointed to an unwritable memory location.	0x0	RW

Bits	Bit Name	Description	Reset	Access
6	STRETCH	Stretch I2CSCL enable. 0: DIS. Disable. 1: EN. Tell the device that, if I2CSCL is 0, hold it at 0 or if I2CSCL is 1, then when it next goes to 0 hold it at 0.	0x0	RW
5	EARLYTXR	Early transmit request mode. 0: DIS. Disable. 1: EN. Enable a transmit request just after the positive edge of the direction bit (read/write) I2CSCL clock pulse.	0x0	RW
4	GCSB	General call status bit clear. 1: CLR. Clear the general call status and general call ID bits. The general call status and general call ID bits are not reset by anything other than a write to this bit or a full reset.	0x0	RW
3	HGC	Hardware general call enable. 0: DIS. Disable. 1: EN. When this bit and the general call enable bit are set, the device, after receiving a general call, Address 0x00 and a data byte, checks the contents of the I2CALT against the receive shift register. If they match, the device has received a hardware general call. This is used if a device needs urgent attention from a master device without knowing which master it needs to turn to. This is a broadcast message to all master devices in the bus. The device that requires attention embeds its own address into the message. The LSB of the I2CALT register should always be written to a 1.	0x0	RW
2	GC	General call enable. 0: DIS. Disable. 1: EN. Enable the I^{2}C slave to ACK an I^{2}C general call, Address 0x00 (write).	0x0	RW
1	ADR10	Enable 10 bit addressing. 0: DIS. If this bit is clear, the slave can support four slave addresses, programmed in Registers I2CID0 to I2CID3. 1: EN. Enable 10-bit addressing. One 10-bit address is supported by the slave and is stored in I2CID0 and I2CID1, where I2CID0 contains the first byte of the address and the upper five bits must be programmed to 11110' I2CID2 and I2CID3 can be programmed with 7-bit addresses at the same time.	0x0	RW
0	SLV	Slave enable. 0: DIS. Disable the slave and all slave state machine flops are held in reset. 1: EN. Enable slave.	0x0	RW

Slave I²C Status, Error and Interrupt Register

Address: 0x4000302C, Reset: 0x0001, Name: I2CSSTA

Table 180. Bit Descriptions for I2CSSTA

Bits	Bit Name	Description	Reset	Access
15	RESERVED	Reserved.	0x0	R
14	START	Start and matching address. 0: CLR. Cleared on receipt of either a stop or start condition. 1: SET. Set if a start is detected on I2CSCL/I2CSDA and one of the following is true: The device address is matched. A general call (GC = 0000_0000) code is received and GC is enabled. A high speed (HS = 0000_1XXX) code is received. A start byte (0000_0001) is received.	0x0	R
13	REPSTART	Repeated start and matching address. (Can drive an interrupt). 0: CLR. Cleared when read or on receipt of a stop condition. 1: SET. Set if START (I2CSSTA[14]) is already asserted and then a repeated start is detected.	0x0	R
[12:11]	IDMAT	Device ID matched. 00: Set to 00 when received address matched ID Register 0. 01: Set to 01 when received address matched ID Register 1. 02: Set to 10 when received address matched ID Register 2. 03: Set to 11 when received address matched ID Register 3.	0x0	R
10	STOP	Stop after start and matching address. (Can drive an interrupt). Notes: If IENSTOP (I2CSCON[8]) is set, then the slave interrupt request asserted when this bit is set. 0: CLR. Cleared by a read of the status register. 1: SET. Set if the slave device received a stop condition after a previous start condition and a matching address.	0x0	R
[9:8]	GCID	General call ID. Cleared when the GCSB (I2CSCON[4]) is written to 1. These status bits are not cleared by a general call reset. 00: No general call. 01: General call reset and program address. 10: General call program address. 11: General call matching alternative ID.	0x0	R
7	GCINT	General call interrupt. (Always drives an interrupt). 0: CLR. To clear this bit, write 1 to the I2CSCON[4]. If it was a general call reset, all registers are at their default values. If it was a hardware general call, the Rx FIFO holds the second byte of the general call and this can be compared with the ALT register. 1: SET. Set if the slave device receives a general call of any type.	0x0	R
6	BUSY	Slave busy. 0: CLR. Cleared by hardware on any of the following conditions: The address does not match an ID register, the slave device receives a I2C stop condition or if a repeated start address doesn't match. 1: SET. Set if the slave device receives an I2C start condition.	0x0	R
5	NOACK	NACK generated by the slave. 0: CLR. Cleared on a read of the I2CSSTA register. 1: SET. Set to indicate that the slave responded to its device address with a NACK. Set under any of the following conditions: If there was no data to transmit and sequence was a slave read, the device address is NACK'ed or if the NACK bit was set in the slave control register and the device was addressed.	0x0	R
4	RXOF	Receive FIFO overflow. 0: CLR. Cleared. 1: SET. Set when a byte is written to the receive FIFO when the FIFO is already full.	0x0	R

Bits	Bit Name	Description	Reset	Access
3	RXREQ	Receive request. (Can drive an interrupt). 0: CLR. Cleared when the receive FIFO is read or flushed. 1: SET. Set when the receive FIFO is not empty. Set on the falling edge of the I2CSCL clock pulse that clocks in the last data bit of a byte.	0x0	R
2	TXREQ	Transmit request. (Can drive an interrupt). 0: CLR. This bit is cleared on a read of the I2CSSTA register. 1: SET. If EARLYTXR = 0, TXREQ is set when the direction bit for a transfer is received high. Thereafter, as long as the transmit FIFO is not full, this bit remains asserted. Initially, it is asserted on the negative edge of the SCL pulse that clocks in the direction bit (if the device address matched also). If EARLYTXR = 1, TXREQ is set when the direction bit for a transfer is received high. Thereafter, as long as the transmit FIFO is not full, this bit will remain asserted. Initially, it is asserted after the positive edge of the SCL pulse that clocks in the direction bit (if the device address matched also).	0x0	R
1	TXUR	Transmit FIFO underflow. 0: CLR. Cleared. 1: SET. Set to 1 if a master requests data from the device and the Tx FIFO is empty for the rising edge of SCL.	0x0	R
0	TXFSEREQ	Tx FIFO status. 0: CLR. Cleared. 1: SET. Set whenever the slave Tx FIFO is empty.	0x1	R

Slave Receive Data Register

Address: 0x40003030, Reset: 0x0000, Name: I2CSRX

Table 181. Bit Descriptions for I2CSRX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	R
[7:0]	VALUE	Receive register.	0x00	R

Slave Transmit Data Register

Address: 0x40003034, Reset: 0x0000, Name: I2CSTX

Table 182. Bit Descriptions for I2CSTX

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	W
[7:0]	VALUE	Transmit register.	0x00	W

Hardware General Call ID Register

Address: 0x40003038, Reset: 0x0000, Name: I2CALT

Table 183. Bit Descriptions for I2CALT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	R
[7:0]	VALUE	ALT register.This register is used in conjunction with HGC (I2CSCON[3]) to match a master generating a hardware general call. It is used in the case where a master device cannot be programmed with a slave's address and, instead, the slave must recognize the master's address.	0x00	RW

First Slave Address Device ID Register

Address: 0x4000303C, Reset: 0x0000, Name: I2CID0

Table 184. Bit Descriptions for I2CID0

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	R
[7:0]	VALUE	Slave ID.	0x00	RW

Second Slave Address Device ID Register

Address: 0x40003040, Reset: 0x0000, Name: I2CID1

Table 185. Bit Descriptions for I2CID1

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	R
[7:0]	VALUE	Slave ID.	0x00	RW

Third Slave Address Device ID Register

Address: 0x40003044, Reset: 0x0000, Name: I2CID2

Table 186. Bit Descriptions for I2CID2

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	R
[7:0]	VALUE	Slave ID.	0x00	RW

Fourth Slave Address Device ID Register

Address: 0x40003048, Reset: 0x0000, Name: I2CID3

Table 187. Bit Descriptions for I2CID3

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	R
[7:0]	VALUE	Slave ID.	0x00	RW

Master and Slave Rx/Tx FIFO Status Register

Address: 0x4000304C, Reset: 0x0000, Name: I2CFSTA

Table 188. Bit Descriptions for I2CFSTA

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved.	0x00	R
9	MFLUSH	Master Transmit FIFO Flush. 0: DIS. For normal FIFO operation. 1: EN. FIFO flush enabled, to keep the FIFO empty.	0x0	RW
8	SFLUSH	Slave Transmit FIFO Flush. 0: DIS. For normal FIFO operation. 1: EN. FIFO flush enabled, to keep the FIFO empty.	0x0	RW
[7:6]	MRXFSTA	Master Receive FIFO Status. 00: Empty 01: ONEBYTE 10: TWOBYTES	0x0	RW

Bits	Bit Name	Description	Reset	Access
[5:4]	MTXFSTA	Master Transmit FIFO Status. 00: Empty 01: ONEBYTE 10: TWOBYTES	0x0	RW
[3:2]	SRXFSTA	Slave Receive FIFO Status. 00: Empty 01: ONEBYTE 10: TWOBYTES	0x0	RW
[1:0]	STXFSTA	Slave Transmit FIFO Status. 00: Empty 01: ONEBYTE 10: TWOBYTES	0x0	RW

SERIAL PERIPHERAL INTERFACES

SPI FEATURES

SPI is a synchronous, serial interface that allows eight bits of data to be synchronously transmitted and simultaneously received. The features included on the two SPI interfaces are

- Master or slave mode with separate 4-byte Rx and Tx FIFOs
- Flexible interrupt modes
- Rx and Tx DMA channels
- Maximum bit rate (with 16 MHz system clock):
    - 5 Mbps in slave mode, 8 Mbps in master mode.
    - 8 Mbps using DMA transfer half duplex.
- Programmable options
    - Serial clock phase and polarity
    - LSB first or MSB first transfer option
    - Open-circuit data output mode
    - Continuous transfer mode

SPI OVERVIEW

The ADuCRF101 integrates two identical hardware serial peripheral interfaces (SPI): SPI0 is reserved for the internal RF link interface, SPI1 is available to interface with external components. It is also possible to use SPI0 to interface to external component. See the SPI0 Configuration for Communication with External Devices section for more details.

The SPI port can be configured for master or slave operation and consists of four pins: MISO, MOSI, SCLK, and $\overline{\text{CS}}$. The pins for SPI0 are SPI0MISO, SPI0MOSI, SPI0SCLK, and SPI0$\overline{\text{CS}}$0. The pins for SPI1 are SPI1MISO, SPI1MOSI, SPI1SCLK, SPI1$\overline{\text{CS}}$0, SPI1$\overline{\text{CS}}$1, SPI1$\overline{\text{CS}}$2, SPI1$\overline{\text{CS}}$3, and SPI1$\overline{\text{CS}}$4.

The GPIOs used for SPI communication must be configured in SPI mode before enabling the SPO peripheral. P0.0 to P0.3 are used for communication with external components. Four additional $\overline{\text{CS}}$ can also be configured in master mode on P0.4 to P0.7 to communicate with multiple slaves. See the Digital Port Multiplex section for more details.

Each SPI interface has two DMA channels, one for transmit and one for receive.

SPI OPERATION

Transmission Format

In both master and slave mode, data is transmitted on one edge of the SCLK signal and sampled on the other. Therefore, it is important that the polarity and phase be configured the same for the master and slave devices (SPIxCON register).

The SPI transfer protocol diagrams (see Figure 52) illustrate the data transfer protocol for the SPI and the effects of CPHA and CPOL bits in the control register on that protocol.

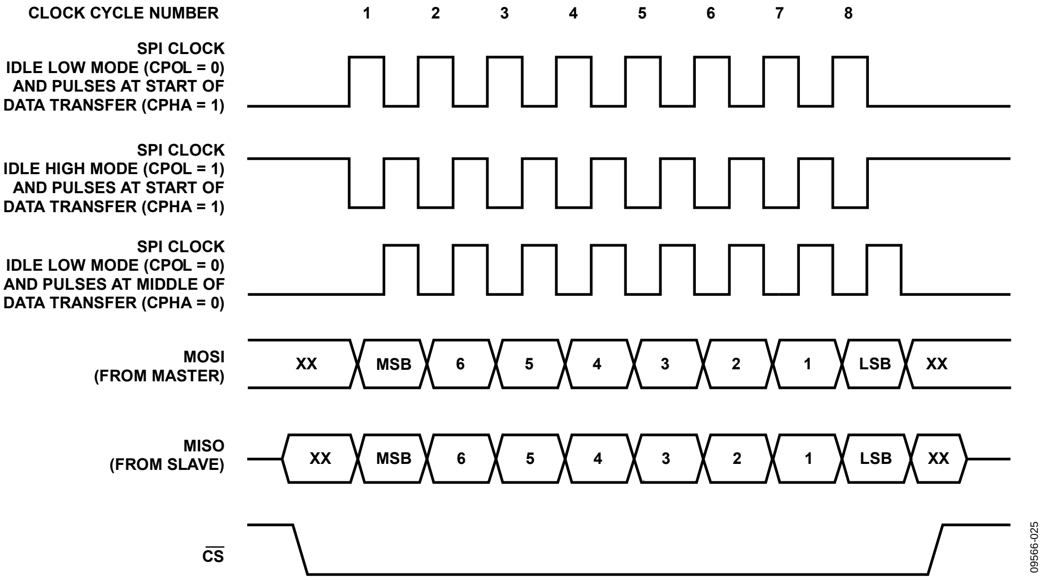

CLOCK CYCLE NUMBER
1
2
3
4
5
6
7
8
SPI CLOCK IDLE LOW MODE (CPOL = 0) AND PULSES AT START OF DATA TRANSFER (CPHA = 1)
SPI CLOCK IDLE HIGH MODE (CPOL = 1) AND PULSES AT START OF DATA TRANSFER (CPHA = 1)
SPI CLOCK IDLE LOW MODE (CPOL = 0) AND PULSES AT MIDDLE OF DATA TRANSFER (CPHA = 0)
MOSI (FROM MASTER)
XX
MSB
6
5
4
3
2
1
LSB
XX
MISO (FROM SLAVE)
XX
MSB
6
5
4
3
2
1
LSB
XX
$\overline{CS}$
09566-025

Figure 52. SPI Transfer Protocol

The SPIxCON register also control if the LSB or MSB is transfer first.

$\overline{\text{CS}}$ can be configured to frame each byte or to stay asserted for multiple bytes using the SPIxCON[11] bit.

Multimaster/Multislave Configuration

Multislave configuration is shown in Figure 53. Five pins can be configured as $\overline{\text{CS}}$. All $\overline{\text{CS}}$ signals are internally controlled together and user code must configure the correct GPIO to control the correct slave. $\overline{\text{CS0}}$ is only available in slave mode. Note that it is necessary to always configure P0.3 as SPI1CS0 for correct operation in master mode. Therefore, it is recommended to use SPI1CS1, SPI1CS2, SPI1CS3, and SPI1CS4 when multiple components need to be selected individually.

In slave mode, only $\overline{\text{CS0}}$ is available.

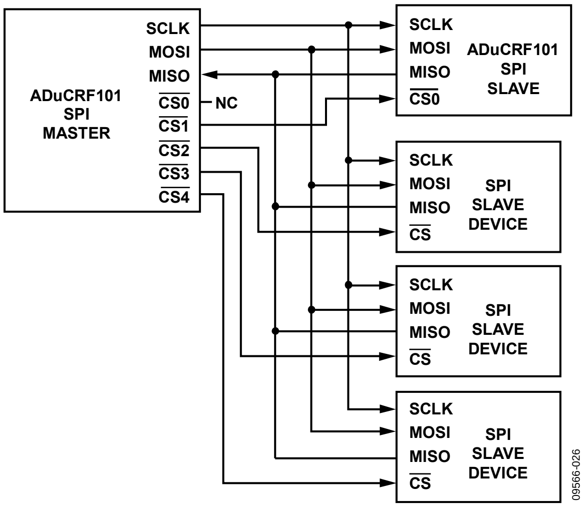


Figure 53. Multislave Configuration Using ADuCRF101 in Both Master and Slave Mode

Wired OR Mode

To prevent contention when the SPI is used in a multimaster or multislave system, the data output pins, MOSI and MISO, can be configured to behave as open-circuit drivers.

In master mode, when a 0 is being transmitted on the MOSI pin, the output driver is enabled. In master mode, when a 1 is being transmitted on the MOSI pin, the output driver is disabled and an external pull-up resister is required to pull the pin high. The typical resistor value is 1 kΩ.

In slave mode, when a 0 is being transmitted on the MISO pin, the output driver is enabled. In slave mode, when a 1 is being transmitted on the MISO pin, the output driver is disabled and an external pull-up resister is required to pull the pin high. The typical resistor value is 1 kΩ.

The WOM bit (SPIxCON[4]) controls the pad enable outputs for the data lines.

Clock Generation

The master serial clock (SCLK) synchronizes the data being transmitted and received. The SCLK pin is automatically configured as an output in master mode and as an input in slave mode.

The bit rate is defined in the SPIxDIV register as follows:

$$f_{SERIALCLOC\ K} = \frac{UCLK}{2 \times (1 + SPIxDIV\ )}$$

When the ADuCRF101 peripheral clock PCLK is set to 4 MHz or less by setting CLKCON, there are limitations on the SPI serial clock frequency as summarized in Table 189.

Table 189. Maximum SPI Serial Clock Frequency vs. Clock Divider Bits

CD Setting	Resulting PCLK (kHz)	Maximum SPIxDIV Setting	Resulting fSCLK (kHz)
2	4000	0	8000
3	2000	1	4000
4	1000	3	2000
5	500	7	1000
6	250	15	500
7	125	31	250

Note the BCRST bit in SPIxDIV[7] always needs to be set in both master and slave mode.

Master Mode

The master has full control of the data flow. The slave cannot transmit data without the master transmitting to the slave.

The $\overline{CS}$ asserts itself automatically at the beginning of a transfer and deasserts itself upon completion. Continuous transfer can be selected in the SPIxCON register, where the $\overline{CS}$ stays asserted until there are no valid data in the master Tx FIFO.

P0.3 must be configured in Mode 1 (as SPI1CS0) for correct operation of the master.

The transfer and interrupt mode bit (SPIxCON[6]) determines the manner in which an SPI serial transfer is initiated:

Tx Initiated Transfer

The SPI starts transmitting as soon as the first byte is written to the FIFO.

If the continuous transfer is enabled (SPIxCON[11]), the transfer continues until no valid data is available in the Tx FIFO. $\overline{CS}$ is asserted at the beginning of the first byte and remains asserted until the Tx FIFO is empty. Values are entered into the Tx FIFO by writing to the SPIxTX register. User code, when it is intended to transmit multiple bytes without the deassertion of $\overline{CS}$, must ensure that the rate writing to the SPIxTX register is faster than the emptying of the FIFO or that this process is not interrupted to ensure that the $\overline{CS}$ does not get deasserted unintentionally during the transfer. Manual control of the $\overline{CS}$ is also possible.

If the continuous transfer is disabled, each byte transfer is framed by the assertion and deassertion of $\overline{CS}$.

Rx Initiated Transfer

Transfers are initiated by a read of the SPIxRX register. The number of transfers is configured by the MOD bits (SPIxCON[15:14]).

For example, if SPIxCON[15:14] is set to 0x3 and user code reads the SPIxRX register, the SPI initiates a 4-byte transfer. If continuous mode is set,(SPIxCON[11]= 0x1), the four bytes occur continuously with no deassertion of $\overline{CS}$ between bytes. If continuous mode is not set, (SPIxCON[11] = 0x0), the four bytes occur with stall periods between transfers where the $\overline{CS}$ is deasserted. A read of the SPIxRX register while the SPI is receiving data does not initiate another transfer after the present transfer is complete.

Full-Duplex Operation

Simultaneous read/writes are supported on the SPI.

When implementing full-duplex transfers in master mode, use the following procedure to handle the availability delay in the SPIxRX register associated with the reception and transfer into the SPIxRX of the data from the input holding register:

- Initiate transfer sequence via a transmit on the MOSI pin (SPIxCON[6] = 1). If interrupts are enabled, interrupts are triggered when a byte is transmitted rather than when a byte is received.
- The SPI Tx interrupt indicated by SPIxSTA[5], or Tx FIFO under run interrupt, (SPIxSTA[4] is asserted approximately 3 × SCLK to 4 × SCLK periods into the transfer of the first byte. Reload a byte into the Tx FIFO if necessary by writing to SPIxTX.
- The first byte received via the MISO pin will not update the Rx FIFO status bits (SPIxSTA[10:8]) until 12 ×SCLK periods after $\overline{\text{CS}}$ has gone low. Therefore two transmit interrupts may occur before the first received byte is ready to be handled.
- After the last transmit interrupt has occurred, it may be necessary to read two more bytes. It is recommended that SPIxSTA[10:8] are polled outside of the SPI interrupt handler after the last transmit interrupt is handled.

Slave Mode

In slave mode, a transfer is initiated by the assertion of the device's $\overline{\text{CS}}$. The SPI port then transmits and receives 8-bit data until the transfer is concluded by deassertion of $\overline{\text{CS}}$. In slave mode, $\overline{\text{CS}}$ is always an input. Note that only SPI0$\overline{\text{CS0}}$ is active in slave mode on SPI1. If the master is transmitting in continuous mode or the slave's $\overline{\text{CS}}$ is connected to GND, the slave should be configured in continuous mode (SPIxCON[11]). Otherwise, it expects $\overline{\text{CS}}$ to go high after each byte.

The SPIxCON[6] configures the slave's type of transfer, Rx or Tx initiated, and interrupt.

Data Underrun and Overflow

For any setting of SPIxCON[6] and in either master or slave mode, the SPI simultaneously receives and transmits data. Therefore, during data transmission, the SPI is also receiving data and filling up the Rx FIFO. If the data is not read from the Rx FIFO, the overflow interrupt occurs when the FIFO starts to overflow.

If the user does not want to read the Rx data or receive overflow interrupts, SPIxCON[12] can be set and the receive data is not saved to the Rx FIFO. Similarly, when the user only wants to receive data and does not want to write data to the Tx FIFO SPIxCON[13] can be set to avoid receiving underrun interrupts from the Tx FIFO.

If the Tx underrun mode bit, SPIxCON[7], is cleared, the last byte from the previous transmission is shifted out when a transfer is initiated with no valid data in the FIFO. If SPIxCON[7] is set, 0s are transmitted when a transfer is initiated with no valid data in the FIFO.

If the Rx overflow overwrite enable bit, SPIxCON[8], is set, the valid data in the Rx FIFO is overwritten by the new serial byte received when there is no space left in the FIFO. If SPIxCON[8] is cleared, the new serial byte received is discarded when there is no space left in the FIFO. When valid data is being overwritten in the Rx FIFO, the oldest byte is overwritten first, followed by the next oldest byte, and so on.

SPI INTERRUPTS

There is one interrupt line per SPI and five sources of interrupts. SPIxSTA[0] reflects the state of the interrupt line, and SPIxSTA[12] and SPIxSTA[7:4] reflects the state of the five sources.

The SPI generates either TIRQ (transmit) or RIRQ (receive). Both interrupts cannot be enabled at the same time. The appropriate interrupt is enabled using the TIM bit, SPIxCON[6]. If TIM = 1, TIRQ is enabled. If TIM = 0, RIRQ is enabled.

Also note that the SPI0 and SPI1 interrupt source must be enabled in the NVIC Register ISER0 (ISER0[17] = SPI0, ISER0[18] = SPI1).

Rx Interrupt

If SPIxCON[6] is cleared, the Rx FIFO status causes the interrupt. SPIxCON[15:14] control when the interrupt occurs. The interrupt is cleared by a read of SPIxSTA. The status of this interrupt can be read by reading SPIxSTA[6].

Interrupts are only generated when data is written to the FIFO. For example if SPIxCON[15:14] are set to 00, an interrupt is generated after the first byte is received. When the status register is read, the interrupt is deactivated. If the byte is not read from the FIFO the interrupt is not regenerated. Another interrupt is not generated until another byte is received into the FIFO.

The interrupt depends on the number of valid bytes in FIFO and not the number of bytes received. For example, when SPIxCON[15:14] are set to 01, an interrupt is generate after a byte is received when there are two or more bytes in the FIFO. The interrupt is not generated after every two bytes received.

The interrupt is disabled if SPIxCON[12] is left high.

Tx Interrupt

If SPIxCON[6] is set, the Tx FIFO status causes the interrupt. SPIxCON[15:14] control when the interrupt occurs, that is, if the interrupt is generated after 1, 2, 3, or 4 bytes transmitted. The interrupts are generated depending on the number of bytes transmitted and not on the number of bytes in the FIFO. This is unlike the Rx interrupt, which depends on the number of bytes in the Rx FIFO and not the number of bytes received.

The transmit interrupt is cleared by a read of the status register. The status of this interrupt can be read by reading SPIxSTA[5]. The interrupt is disabled if SPIxCON[13] is left high.

A write to the control register, SPIxCON, resets the transmitted byte counter back to zero. For example, in a case where SPIxCON[15:14] is set to 0x3 and SPIxCON is written to after three bytes have been transmitted, then the Tx interrupt does not occur until another four bytes have been transmitted.

Data Underrun and Overflow Interrupts

SPIxSTA[7] and SPIxSTA[4] also generate SPI interrupts.

When a transfer starts with no data in the Tx FIFO, SPIxSTA[4] is set to indicate an underrun condition. This causes an interrupt. The interrupt (and status bit) are cleared on a read of the status register. This interrupt occurs irrespective of SPIxCON[15:14]. This interrupt is disabled if SPIxCON[13] is set.

When data is received and the Rx FIFO is already full, this causes SPIxSTA[7] to go high, indicating an overflow condition. This causes an interrupt. The interrupt (and status bit) are cleared on a read of the status register. This interrupt occurs irrespective of SPIxCON[15:14]. This interrupt is disabled if SPIxCON[12] is set.

All interrupts are cleared by a read of the status register or if SPIxCON[0] is deasserted. The Rx and Tx interrupts are also cleared if the relevant flush bits are asserted. Otherwise, the interrupts stay active even if the SPI is reconfigured.

CSERR Interrupt

The CSERR bit in the SPIxSTA register indicates if an erroneous deassertion of the $\overline{\text{CS}}$ signal before the completion of all the eight SCLK cycles has been detected. This bit generates an interrupt and is available in all modes of operation: slave, master and during DMA transfers.

If an interrupt occurs, generated by the CSERR bit, the SPI ENABLE bit, (SPIxCON[0]) should be disabled and restarted to enable a clean recovery. This ensures that the subsequent transfers are error-free. The BCRST bit, (SPIxDIV[7]) should be set at all times in both slave and master mode except when a mid-byte stall in SPI communication is required. In this case, CSERR bit (SPIxSTA[12]) is set but can be ignored.

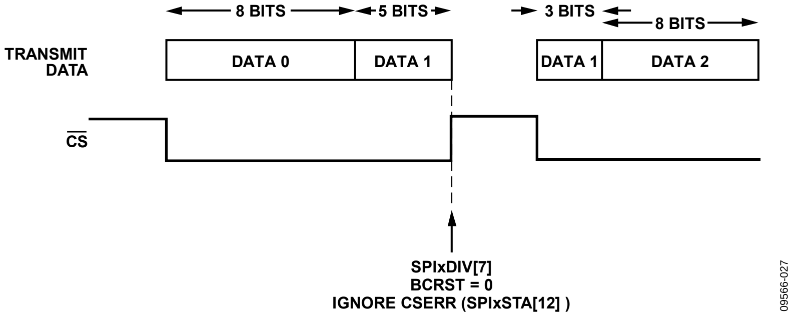


Figure 54. SPI Communication: Mid-Byte Stall

Note the SPI should only be reenabled when the $\overline{\text{CS}}$ signal is high.

SPI DMA

DMA operation is available independently for transmit and receive on each SPI interface. Each SPI interface has two DMA channels, one for Rx and one for Tx, and associated interrupts in the NVIC. The DMA channels can be configured in the DMA controller. The DMA interrupts should be configured in the NVIC. SPI DMA transfers are enabled in the SPIxDMA register.

The SPI Tx and Rx interrupts are not generated during DMA transfer, but TXUR, RXOF, and CSERR are generated. If only the DMA transmit request (SPIxDMA[1]) is enabled, to avoid generating Rx overflow interrupts, the Rx FIFO flush bit should be set, or the SPI interrupt disabled in the NVIC. If only the DMA receive request (SPIxDMA[2]) is enabled, to avoid Tx underrun interrupts, the Tx FIFO flush bit should be set, or the SPI interrupt disabled in the NVIC.

When using DMA, SPIxCON[15:14] settings become irrelevant but should be set to 00 in receive mode.

DMA Master Transmit Configuration

The following sequence should be implemented in user code:

1. The DMA SPI Tx channel should be configured.
2. The SPI DMA Tx interrupt should enabled in the NVIC.
3. The SPI block should be configured as follows:

```
pADI_SPI1->SPIDIV = SPI_serial_div;     //Configures serial clock frequency

pADI_SPI1->SPICON = 0x1043;             //Enable SPI in master mode and transmit mode, Rx FIFO
                                        flush enabled.

pADI_SPI1->SPIDMA = = 0x3;              //Enable DMA mode, enable Tx DMA request
```

The ENABLE bit (SPIxDMA[0]) controls the start of a DMA transfer in transmit mode. A DMA request from the SPI to the DMA controller is generated each time there is free space in the Tx FIFO to always keep the FIFO full. DMA requests are only generated when ENABLE = 1 (SPIxDMA[0]).

At the end of a DMA transfer, that is, when receiving a DMA SPI transfer interrupt, user code should disable the DMA by clearing SPIxDMA[0] to prevent extra DMA requests to the DMA controller. The data still present in the Tx FIFO is transmitted. User code can check in the FIFO status register the number of byte still present in the FIFO, and ensure the SPI transfer is complete before disabling SPI.

DMA Master Receive Configuration

The following sequence should be implemented in user code:

1. The DMA SPI Rx channel should be configured.
2. The NVIC should be configured to enable DMA Rx master interrupt.
3. The SPI block should be configured as follows:

```
pADI_SPI1->SPIDIV = SPI_serial_freq;    //Configures serial clock frequency where

pADI_SPI1->SPICON = 0x2003;             //Enable SPI in master mode and receive mode,
                                        1 byte transfer.

pADI_SPI1->SPIDMA = 0x5;                //Enable DMA mode, enable Rx DMA request

pADI_SPI1->SPICNT = XXX;                //Number of bytes to transferred + 1

A = pADI_SPI1->SPIRX;                   //Dummy read
```

When using DMA, SPIxCON[15:14] settings become irrelevant, but should be set to 0x0 in receive mode.

The SPIxCNT register is available in DMA receive master mode only. It sets the number of receive bytes required by the SPI master, or the number of clocks that the master needs to generate. When the required number of bytes are received, no more transfers are initiated. To initiate a DMA master receive transfer, a dummy read should be completed by user code. This dummy read should be added to the SPIxCNT number.

The counter counting the bytes as they are received is reset when SPI is disabled in SPIxCON[0], or if SPIxCNT register is modified by user code.

The DMA transfer stops when the number of clock has been generated. Note that the DMA buffer must be of the same size as SPIxCNT to generate a DMA interrupt when the transfer is complete.

SPI AND POWER-DOWN MODES

In master mode, before entering power-down mode it is recommended to disable the SPI block in SPIxCON[0].

In slave mode, in either mode of operation, interrupt driven or DMA, the $\overline{\text{CS line}}$ level should be checked via the GPIO registers to ensure the SPI is not communicating and the SPI block should be disabled while the $\overline{\text{CS}}$ line is high. At power-up, the SPI block can be reenabled.

OTHER CONSIDERATIONS

When changing the SPI configuration, care must be taken not to change it during a data transfer to avoid corrupting the data. It is recommended to change configuration when the module is disabled (SPIxCON[0] = 0) , then reconfigure and then reenable the SPI (SPIxCON[0] = 1).

- When reconfiguring from slave mode to master mode, or vice versa, both FIFOs must be empty and flushed, if necessary.

SPI0 Configuration for RF Link

The SPI0 interface is used to communicate with the RF link. It should be configured in master mode, clock polarity low and phase with sample leading, MSB first. The speed should be 4 MHz maximum.

```
pADI_SPI0->SPICON =  SPI0CON_MASEN  |    // Master mode
SPI0CON_TIM    |                         // Interrupt on transmit
SPI0CON_ENABLE;
```

SPI0 Configuration for Communication with External Devices

In applications that require two SPI interfaces, SPI0 can also be diverted from the RF link interface and used with external devices on P3.2 (SPI0MISO), P3.3 (SPI0SCLK), P3.5 (SPI0MOSI), and P4.2 (SPI0CS). GP3CON and GP4CON need to be configured to switch between the two configurations. For slave operation, GP2CON must be set to 0x55.

REGISTER SUMMARY (SERIAL PERIPHERAL INTERFACE)

Table 190. Serial Peripheral Interface Register Summary

Address	Name	Description	Reset	RW
0x40004000	SPI0STA	SPI0 Status Register	0x0000	R
0x40004004	SPI0RX	SPI0 Receive Register	0x00	R
0x40004008	SPI0TX	SPI0 Transmit Register	0x00	W
0x4000400C	SPI0DIV	SPI0 Bit Rate Selection Register	0x0000	RW
0x40004010	SPI0CON	SPI0 Configuration Register	0x0000	RW
0x40004014	SPI0DMA	SPI0 DMA Enable Register	0x0000	RW
0x40004018	SPI0CNT	SPI0 DMA Master Received Byte Count Register	0x0000	RW
0x40004400	SPI1STA	SPI1 Status Register	0x0000	R
0x40004404	SPI1RX	SPI1 Receive Register	0x00	R
0x40004408	SPI1TX	SPI1 Transmit Register	0x00	W
0x4000440C	SPI1DIV	SPI1 Bit Rate Selection Register	0x0000	RW
0x40004410	SPI1CON	SPI1 Configuration Register	0x0000	RW
0x40004414	SPI1DMA	SPI1 DMA Enable Register	0x0000	RW
0x40004418	SPI1CNT	SPI1 DMA Master Receive Byte Count Register	0x0000	RW

REGISTER DETAILS (SERIAL PERIPHERAL INTERFACE)

SPI0 Status Register

Address: 0x40004000, Reset: 0x0000, Name: SPI0STA

Table 191. Bit Descriptions for SPI0STA

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved.	0x0	R
12	CSERR	$\overline{\text{CS}}$ error status bit. This bit generates an interrupt when detecting an abrupt CS desassertion before the full byte of data is transmitted completely. 0: CLR: Cleared when no $\overline{\text{CS}}$ error is detected. Cleared to 0 on a read of SPI0STA register. 1: SET: Set when the $\overline{\text{CS}}$ line is deasserted abruptly.	0x0	R
11	RXS	Rx FIFO excess bytes present. 0: CLR. When the number of bytes in the FIFO is equal or less than the number in SPI0CON[15:14]. This bit is not cleared on a read of SPI0STA register. 1: SET. When there are more bytes in the Rx FIFO than configured in MOD(SPI0CON[15:14]). For example, if MOD = TX1RX1, RXS is set when there are two or more bytes in the Rx FIFO. This bit does not dependent on SPI0CON[6] and does not cause an interrupt.	0x0	R

Bits	Bit Name	Description	Reset	Access
[10:8]	RXFSTA	Rx FIFO status bits, indicates how many valid bytes are in the Rx FIFO. 000: EMPTY. When Rx FIFO is empty. 001: ONEBYTE. When 1 valid byte is in the FIFO. 010: TWOBYTES. When 2 valid bytes are in the FIFO. 011: THREEBYTES. When 3 valid bytes are in the FIFO. 100: FOURBYTES. When 4 valid bytes are in the FIFO.	0x0	R
7	RXOF	Rx FIFO overflow status bit. This bit generates an interrupt. 0: CLR. Cleared to 0 on a read of SPI0STA register. 1: SET. Set when the Rx FIFO is already full when new data is loaded to the FIFO. This bit generates an interrupt except when RFLUSH is set (SPI0CON[12]).	0x0	R
6	RX	Rx interrupt status bit. This bit generates an interrupt, except when DMA transfer is enabled. 0: CLR. Cleared to 0 on a read of SPI0STA register. 1: SET. Set when a receive interrupt occurs. This bit is set when TIM (SPI0CON[6]) is cleared and the required number of bytes have been received.	0x0	R
5	TX	Tx interrupt status bit. This bit generates an interrupt, except when DMA transfer is enabled. 0: CLR. Cleared to 0 on a read of SPI0STA register. 1: SET. Set when a transmit interrupt occurs. This bit is set when TIM (SPI0CON[6]) set and the required number of bytes have been transmitted.	0x0	R
4	TXUR	Tx FIFO underrun. This bit generates an interrupt. 0: CLR. Cleared to 0 on a read of SPI0STA register. 1: SET. Set when a transmit is initiated without any valid data in the Tx FIFO. This bit generates an interrupt except when TFLUSH is set in SPI0CON.	0x0	R
[3:1]	TXFSTA	Tx FIFO status bits, indicates how many valid bytes are in the Tx FIFO. 000: EMPTY. Tx FIFO is empty. 001: ONEBYTE. 1 valid byte is in the FIFO. 010: TWOBYTES. 2 valid bytes are in the FIFO. 011: THREEBYTES. 3 valid bytes are in the FIFO. 100: FOURBYTES . 4 valid bytes are in the FIFO.	0x0	R
0	IRQ	Interrupt status bit. 0: CLR. Cleared to 0 on a read of SPI0STA register. 1: SET. Set to 1 when an SPI0 based interrupt occurs.	0x0	R

SPI0 Receive Register

Address: 0x40004004, Reset: 0x00, Name: SPI0RX

Table 192. Bit Descriptions for SPI0RX

Bits	Bit Name	Description	Reset	Access
[7:0]	VALUE	8-bit receive register. A read of the Rx FIFO returns the next byte to be read from the FIFO. A read of the FIFO when it is empty returns zero.	0x00	R

SPI0 Transmit Register

Address: 0x40004008, Reset: 0x00, Name: SPI0TX

Table 193. Bit Descriptions for SPI0TX

Bits	Bit Name	Description	Reset	Access
[7:0]	VALUE	8-bit transmit register. A write to the Tx FIFO address space writes data to the next available location in the Tx FIFO. If the FIFO is full, the oldest byte of data in the FIFO is overwritten. A read from this address location returns zero.	0x00	W

SPI0 Bit Rate Selection Register

Address: 0x4000400C, Reset: 0x0000, Name: SPI0DIV

Table 194. Bit Descriptions for SPI0DIV

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved. 0 should be written to these bits.	0x00	R
7	BCRST	Configures the behavior of SPI communication after an abrupt deassertion of $\overline{CS}$. This bit should be set in slave and master mode. 0: DIS. Resumes communication from where it stopped when the CS is deasserted. The rest of the bits are then received/transmitted when CS returns low. User code should ignore the CSERR interrupt. 1: EN. Enabled for a clean restart of SPI transfer after a CSERR condition. User code must also clear the SPI enable bit in SPI0CON during the CSERR interrupt.	0x0	RW
6	RESERVED	Reserved. 0 should be written to this bit.	0x0	R
[5:0]	DIV	Factor used to divide UCLK in the generation of the master mode serial clock.	0x00	RW

SPI0 Configuration Register

Address: 0x40004010, Reset: 0x0000, Name: SPI0CON

Table 195. Bit Descriptions for SPI0CON

Bits	Bit Name	Description	Reset	Access
[15:14]	MOD	IRQ mode bits. When TIM is set these bits configure when the Tx/Rx interrupts occur in a transfer. For a DMA Rx transfer, these bits should be 00. 00: TX1RX1. Tx/Rx interrupt occurs when 1 byte has been transmitted/received from/into the FIFO. 01: TX2RX2. Tx/Rx interrupt occurs when 2 bytes have been transmitted/received from/into the FIFO. 10: TX3RX3. Tx/Rx interrupt occurs when 3 bytes have been transmitted/received from/into the FIFO. 11: TX4RX4. Tx/Rx interrupt occurs when 4 bytes have been transmitted/received from/into the FIFO.	0x0	RW
13	TFLUSH	Tx FIFO flush enable bit. 0: DIS. Disable Tx FIFO flushing. 1: EN. Flush the Tx FIFO. This bit does not clear itself and should be toggled if a single flush is required. If this bit is left high, then either the last transmitted value or 0x00 is transmitted depending on the ZEN bit (SPI0CON[7]). Any writes to the Tx FIFO are ignored while this bit is set.	0x0	RW
12	RFLUSH	Rx FIFO flush enable bit. 0: DIS. Disable Rx FIFO flushing. 1: EN. If this bit is set, all incoming data is ignored and no interrupts are generated. If set and TIM = 0 (SPI0CON[6]), a read of the Rx FIFO initiates a transfer.	0x0	RW
11	CON	Continuous transfer enable bit. 0: DIS. Disable continuous transfer. Each transfer consists of a single 8-bit serial transfer. If valid data exists in the SPIxTX register, then a new transfer is initiated after a stall period of one serial clock cycle. The CS line is deactivated for this one serial clock cycle. 1: EN. Enable continuous transfer. In master mode, the transfer continues until no valid data is available in the Tx register. CS is asserted and remains asserted for the duration of each 8-bit serial transfer until Tx is empty.	0x0	RW
10	LOOPBACK	Loopback enable bit. 0: DIS. Normal mode. 1: EN. Connect MISO to MOSI, thus, data transmitted from Tx register is looped back to the Rx register. SPI must be configured in master mode.	0x0	RW

Bits	Bit Name	Description	Reset	Access
9	SOEN	Slave output enable bit. 0: DIS. Disable the output driver on the MISO pin. The MISO pin is open-circuit when this bit is clear. 1: EN. MISO operates as normal.	0x0	RW
8	RXOF	RX overflow overwrite enable bit. 0: DIS. The new serial byte received is discarded when there is no space left in the FIFO 1: EN. The valid data in the Rx register is overwritten by the new serial byte received when there is no space left in the FIFO.	0x0	RW
7	ZEN	Transmit underrun. Transmit 0s when the Tx FIFO is empty. 0: DIS. The last byte from the previous transmission is shifted out when a transfer is initiated with no valid data in the FIFO. 1: EN. Transmit 0x00 when a transfer is initiated with no valid data in the FIFO.	0x0	RW
6	TIM	Transfer and interrupt mode bit. 1: TXWR. Initiate transfer with a write to the SPIxTX register. Interrupt only occurs when Tx is empty. 0: RXRD. Initiate transfer with a read of the SPIxRX register. The read must be done while the SPI interface is idle. Interrupt only occurs when Rx is full.	0x0	RW
5	LSB	LSB first transfer enable bit. 0: DIS. MSB is transmitted first. 1: EN. LSB is transmitted first.	0x0	RW
4	WOM	Wired OR enable bit. 0: DIS. Normal driver output operation. 1: EN. Enable open circuit data output for multimaster/multislave configuration.	0x0	RW
3	CPOL	Serial clock polarity mode bit. 0: LOW. Serial clock idles low. 1: HIGH. Serial clock idles high.	0x0	RW
2	CPHA	Serial clock phase mode bit. 0: SAMPLELEADING. Serial clock pulses at the middle of the first data bit transfer. 1: SAMPLETRAILING. Serial clock pulses at the start of the first data bit.	0x0	RW
1	MASEN	Master mode enable bit. 0: DIS. Configure in slave mode. 1: EN. Configure in master mode.	0x0	RW
0	ENABLE	SPI enable bit. 0: DIS. Disable the SPI. Clearing this bit will also reset all the FIFO related logic to enable a clean start. 1: EN. Enable the SPI.	0x0	RW

SPI0 DMA Enable Register

Address: 0x40004014, Reset: 0x0000, Name: SPI0DMA

Table 196. Bit Descriptions for SPI0DMA

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved. 0 should be written to these bits.	0x0000	R
2	IENRXDMA	Receive DMA request enable bit. 0: DIS. Disable Rx DMA requests. 1: EN. Enable Rx DMA requests.	0x0	RW
1	IENTXDMA	Transmit DMA request enable bit. 0: DIS. Disable Tx DMA requests. 1: EN. Enable Tx DMA requests.	0x0	RW
0	ENABLE	DMA data transfer enable bit. 0: DIS. Disable DMA transfer. This bit needs to be cleared to prevent extra DMA request to the µDMA controller. 1: EN. Enable a DMA transfer. Starts the transfer of a master configured to initiate transfer on transmit.	0x0	RW

SPI0 DMA Master Received Byte Count Register

Address: 0x40004018, Reset: 0x0000, Name: SPI0CNT

Table 197. Bit Descriptions for SPI0CNT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	R
[7:0]	VALUE	Number of bytes requested by the SPI master during DMA transfer, when configured to initiate a transfer on a read of SPI0RX. This register is only used in DMA, master, Rx mode.	0x00	RW

SPI1 Status Register

Address: 0x40004400, Reset: 0x0000, Name: SPI1STA

Table 198. Bit Descriptions for SPI1STA

Bits	Bit Name	Description	Reset	Access
[15:13]	RESERVED	Reserved.	0x0	R
12	CSERR	CS error status bit. This bit generates an interrupt when detecting an abrupt CS desassertion before the full byte of data is transmitted completely. 0: CLR: Cleared when no CS error is detected. Cleared to 0 on a read of SPI1STA register. 1: SET. Set when the CS line is deasserted abruptly.	0x0	R
11	RXS	Rx FIFO excess bytes present. 0: CLR. When the number of bytes in the FIFO is equal or less than the number in SPI0CON[15:14]. This bit is not cleared on a read of SPI0STA register. 1: SET. When there are more bytes in the Rx FIFO than configured in MOD (SPI1CON[15:14]). For example, if MOD = TX1RX1, RXS is set when there are two or more bytes in the Rx FIFO. This bit does not dependent on SPI1CON[6] and does not cause an interrupt.	0x0	R
[10:8]	RXFSTA	Rx FIFO status bits, indicates how many valid bytes are in the Rx FIFO. 000: EMPTY. When Rx FIFO is empty. 001: ONEBYTE. When 1 valid byte is in the FIFO. 010: TWOBYTES. When 2 valid bytes are in the FIFO. 011: THREEBYTES. When 3 valid bytes are in the FIFO. 100: FOURBYTES. When 4 valid bytes are in the FIFO.	0x0	R

Bits	Bit Name	Description	Reset	Access
7	RXOF	Rx FIFO overflow status bit. This bit generates an interrupt. 0: CLR.Cleared to 0 on a read of SPI1STA register. 1: SET. Set when the Rx FIFO is already full when new data is loaded to the FIFO. This bit generates an interrupt except when RFLUSH is set. (SPI1CON[12]).	0x0	R
6	RX	Rx interrupt status bit. This bit generates an interrupt, except when DMA transfer is enabled. 0: CLR. Cleared to 0 on a read of SPI1STA register. 1: SET. Set when a receive interrupt occurs. This bit is set when TIM (SPI1CON[6]) is cleared and the required number of bytes have been received.	0x0	R
5	TX	Tx interrupt status bit. This bit generates an interrupt, except when DMA transfer is enabled. 0: CLR. Cleared to 0 on a read of SPI1STA register. 1: SET. Set when a transmit interrupt occurs. This bit is set when TIM (SPI1CON[6]) is set and the required number of bytes have been transmitted.	0x0	R
4	TXUR	Tx FIFO Underrun. This bit generates an interrupt. 0: CLR. Cleared to 0 on a read of SPI1STA register. 1: SET. Set when a transmit is initiated without any valid data in the Tx FIFO. This bit generates an interrupt except when TFLUSH is set in SPI1CON.	0x0	R
[3:1]	TXFSTA	Tx FIFO status bits, indicates how many valid bytes are in the Tx FIFO. 000: EMPTY. When Tx FIFO is empty. 001: ONEBYTE. 1 valid byte is in the FIFO. 010: TWOBYTES. 2 valid bytes are in the FIFO. 011: THREEBYTES. 3 valid bytes are in the FIFO. 100: FOURBYTES. 4 valid bytes are in the FIFO.	0x0	R
0	IRQ	Interrupt status bit. 0: CLR. Cleared to 0 on a read of SPI1STA register. 1: SET. Set to 1 when an SPI1 based interrupt occurs.	0x0	R

SPI1 Receive Register

Address: 0x40004404, Reset: 0x00, Name: SPI1RX

Table 199. Bit Descriptions for SPI1RX

Bits	Bit Name	Description	Reset	Access
[7:0]	VALUE	8-bit receive register. A read of the Rx FIFO returns the next byte to be read from the FIFO. A read of the FIFO when it is empty returns zero.	0x00	R

SPI1 Transmit Register

Address: 0x40004408, Reset: 0x00, Name: SPI1TX

Table 200. Bit Descriptions for SPI1TX

Bits	Bit Name	Description	Reset	Access
[7:0]	VALUE	8-bit transmit register. A write to the Tx FIFO address space writes data to the next available location in the Tx FIFO. If the FIFO is full, the oldest byte of data in the FIFO is overwritten. A read from this address location return zero.	0x00	W

SPI1 Bit Rate Selection Register

Address: 0x4000440C, Reset: 0x0000, Name: SPI1DIV

Table 201. Bit Descriptions for SPI1DIV

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved. 0 should be written to these bits.	0x00	R
7	BCRST	Configures the behavior of SPI communication after an abrupt deassertion of CS. This bit should be set in slave and master mode. 0: DIS. Resumes communication from where it stopped when the CS is deasserted. The rest of the bits are then received/transmitted when CS returns low. User code should ignore the CSERR interrupt. 1: EN. Enabled for a clean restart of SPI transfer after a CSERR condition. User code must also clear the SPI enable bit in SPI1CON during the CSERR interrupt.	0x0	RW
6	RESERVED	Reserved. 0 should be written to this bit.	0x0	R
[5:0]	DIV	Factor used to divide UCLK in the generation of the master mode serial clock.	0x00	RW

SPI1 Configuration Register

Address: 0x40004410, Reset: 0x0000, Name: SPI1CON

Table 202. Bit Descriptions for SPI1CON

Bits	Bit Name	Description	Reset	Access
[15:14]	MOD	IRQ mode bits. When TIM is set these bits configure when the Tx/Rx interrupts occur in a transfer. For a DMA Rx transfer, these bits should be 00. 00: TX1RX1. Tx/Rx interrupt occurs when 1 byte has been transmitted/received from/into the FIFO. 01: TX2RX2. Tx/Rx interrupt occurs when 2 bytes have been transmitted/received from/into the FIFO. 10: TX3RX3. Tx/Rx interrupt occurs when 3 bytes have been transmitted/received from/into the FIFO. 11: TX4RX4. Tx/Rx interrupt occurs when 4 bytes have been transmitted/received from/into the FIFO.	0x0	RW
13	TFLUSH	Tx FIFO flush enable bit. 0: DIS. Disable Tx FIFO flushing. 1: EN. Flush the Tx FIFO. This bit does not clear itself and should be toggled if a single flush is required. If this bit is left high, then either the last transmitted value or 0x00 is transmitted depending on the ZEN bit (SPI1CON[7]). Any writes to the Tx FIFO are ignored while this bit is set.	0x0	RW
12	RFLUSH	Rx FIFO flush enable bit. 0: DIS. Disable Rx FIFO flushing. 1: EN. If this bit is set, all incoming data is ignored and no interrupts are generated. If set and TIM = 0 (SPI1CON[6]), a read of the Rx FIFO initiates a transfer.	0x0	RW
11	CON	Continuous transfer enable bit. 0: DIS. Disable continuous transfer. Each transfer consists of a single 8-bit serial transfer. If valid data exists in the SPIxTX register, then a new transfer is initiated after a stall period of one serial clock cycle. The CS line is deactivated for this one serial clock cycle. 1: EN. Enable continuous transfer. In master mode, the transfer continues until no valid data is available in the Tx register. CS is asserted and remains asserted for the duration of each 8-bit serial transfer until Tx is empty.	0x0	RW
10	LOOPBACK	Loopback enable bit. 0: DIS. Normal mode. 1: EN. Connect MISO to MOSI, thus, data transmitted from Tx register is looped back to the Rx register. SPI must be configured in master mode.	0x0	RW

Bits	Bit Name	Description	Reset	Access
9	SOEN	Slave output enable bit. 0: DIS. Disable the output driver on the MISO pin. The MISO pin is open-circuit when this bit is clear. 1: EN. MISO operates as normal.	0x0	RW
8	RXOF	RX overflow overwrite enable bit. 0: DIS. The new serial byte received is discarded when there is no space left in the FIFO. 1: EN. The valid data in the Rx register is overwritten by the new serial byte received when there is no space left in the FIFO.	0x0	RW
7	ZEN	TX underrun. Transmit 0s when Tx FIFO is empty. 0: DIS. The last byte from the previous transmission is shifted out when a transfer is initiated with no valid data in the FIFO. 1: EN. Transmit 0x00 when a transfer is initiated with no valid data in the FIFO.	0x0	RW
6	TIM	Transfer and interrupt mode bit. 1: TXWR. Initiate transfer with a write to the SPIxTX register. Interrupt only occurs when Tx is empty. 0: RXRD. Initiate transfer with a read of the SPIxRX register. The read must be done while the SPI interface is idle. Interrupt only occurs when Rx is full.	0x0	RW
5	LSB	LSB first transfer enable bit. 0: DIS. MSB is transmitted first. 1: EN. LSB is transmitted first.	0x0	RW
4	WOM	Wired OR enable bit. 0: DIS. Normal driver output operation. 1: EN. Enable open circuit data output for multimaster/multislave configuration.	0x0	RW
3	CPOL	Serial clock polarity mode bit. 0: LOW. Serial clock idles low. 1: HIGH. Serial clock idles high.	0x0	RW
2	CPHA	Serial clock phase mode bit. 0: SAMPLELEADING. Serial clock pulses at the middle of the first data bit transfer. 1: SAMPLETRAILING. Serial clock pulses at the start of the first data bit.	0x0	RW
1	MASEN	Master mode enable bit. 0: DIS. Configure in slave mode. 1: EN. Configure in master mode.	0x0	RW
0	ENABLE	SPI enable bit. 0: DIS. Disable the SPI. Clearing this bit will also reset all the FIFO related logic to enable a clean start. 1: EN. Enable the SPI.	0x0	RW

SPI1 DMA Enable Register

Address: 0x40004414, Reset: 0x0000, Name: SPI1DMA

Table 203. Bit Descriptions for SPI1DMA

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved. 0 should be written to these bits.	0x0000	R
2	IENRXDMA	Receive DMA request enable bit. 0: DIS. Disable Rx DMA requests. 1: EN. Enable Rx DMA requests.	0x0	RW
1	IENTXDMA	Transmit DMA request enable bit. 0: DIS. Disable Tx DMA requests. 1: EN. Enable Tx DMA requests.	0x0	RW
0	ENABLE	DMA data transfer enable bit. 0: DIS. Disable DMA transfer. This bit needs to be cleared to prevent extra DMA request to the µDMA controller. 1: EN. Enable a DMA transfer. Starts the transfer of a master configured to initiate transfer on transmit.	0x0	RW

SPI1 DMA Master Receive Byte Count Register

Address: 0x40004418, Reset: 0x0000, Name: SPI1CNT

Table 204. Bit Descriptions for SPI1CNT

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x00	R
[7:0]	VALUE	Number of bytes requested by the SPI master during DMA transfer, when configured to initiate a transfer on a read of SPI1RX. This register is only used in DMA, master, Rx mode.	0x00	RW

UART SERIAL INTERFACE

UART FEATURES

- Industry standard, 16450 UART peripheral
- Support for DMA

UART OVERVIEW

The UART peripheral is a full-duplex universal asynchronous receiver/transmitter (UART), compatible with the industry standard, 16450. The UART is responsible for converting data between serial and parallel formats. The serial communication follows an asynchronous protocol, supporting various word length, stop bits, and parity generation options.

This UART also contains modem control and interrupt handling hardware. The UART features a fractional divider that facilitates high accuracy baud rate generation.

Interrupts can be generated from a number of unique events, such as full/empty data buffer, transfer error detection, and break detection.

UART OPERATION

Serial Communications

An asynchronous serial communication protocol is followed with these options:

- Five to eight data bits
- One, two, or 1½ stop bits
- None, even, or odd parity
- Baud rate = UCLK ÷ (2 × 16 × COMDIV) ÷ (M + N ÷ 2048), where COMDIV = 1 to 65536, M = 1 to 3, and N = 0 to 2047

All data words require a start bit and at least one stop bit. This creates a range from seven bits to 12 bits for each word. Transmit operation is initiated by writing to the transmit holding register (COMTX). After a synchronization delay, the data is moved to the internal transmit shift register (TSR) where it is shifted out at a baud (bit) rate equal to UCLK ÷ (2 × 16 × COMDIV) ÷ (M + N ÷ 2048) with start, stop, and parity bits appended as required. All data words begin with a low going start bit. The transfer of COMTX to the TSR causes the transmit register empty status flag to be set.

Receive operation uses the same data format as the transmit configuration except for the number of stop bits, which is always one. After detection of the start bit, the received word is shifted in the internal receive shift register (RSR). After the appropriate number of bits (including stop bits) are received, the data and any status is updated, and the RSR is transferred to the receive buffer register (COMRX). The receive buffer register full status flag is updated upon the transfer of the received word to this buffer and the appropriate synchronization delay.

A sampling clock equal to 16 times the baud rate is used to sample the data as close to the midpoint of the bit as possible. A receive filter is also present that removes spurious pulses of less than two times the sampling clock period.

Data is transmitted and received least significant bit first. This is often not the assumed case by the user. However, it is standard for the protocol.

For power saving purposes, it is possible to reduce the clock to the UART block via the CLKCON register. By default, the clock to the UART is enabled (CLKACT[7] = 1).

PROGRAMMED I/O MODE

In this mode, the software is responsible for moving data to and from the UART. This is typically accomplished by interrupt service routines that respond to the transmit and receive interrupts by either reading or writing data as appropriate. This mode puts certain constraints on the software itself in that the software must respond within a certain time to prevent overflow errors from occurring in the receive channel.

Programmed I/O mode also includes polling the status flags to determine when it is okay to move data.

Polling the status flag is processor intensive and not typically used unless the system can tolerate the overhead. Interrupts can be disabled using the COMIEN register.

Writing the COMTX when it is not empty or reading the COMRX when it is not full produces incorrect results and should not be done. In the former case, the COMTX is overwritten by the new word and the previous word is never transmitted. In the latter case, the previously received word is read again. Both of these errors must be avoided in software by correctly using either interrupts or the status register polling. These errors are not detected in hardware.

DMA MODE

In this mode, user code does not move data to and from the UART. DMA request signals going to external DMA block indicate that the UART is ready to transmit or receive data. These DMA request signals can be disabled in the COMIEN register.

In DMA mode, modem functionality is not available.

ENABLE/DISABLE BIT

Before the ADuCRF101 enters power-down mode, it is recommended to disable the serial interfaces. A bit is provided in the UART control register to disable the UART serial peripheral. This bit disables the clock to the peripheral. When setting this bit, care must be taken in software that no data is being transmitted or received. If set during communication, the data transfer does not complete; the receive or transmit register contains only part of the data.

INTERRUPTS

The UART peripheral has one interrupt output to the interrupt controller for both Rx and Tx interrupts. The COMIIR register must be read by software to determine the cause of the interrupt.

In I/O mode, when receiving, the interrupt is generated for the following cases:

- COMRX full
- Receive overflow error
- Receive parity error
- Receive framing error
- Break interrupt (UART input (UARTRXD) held low)
- Modem status interrupt (changes to CTS)
- COMTX empty

UART Transmit and Receive Registers

COMRX and COMTX share the same address while they are implemented as different registers. If written to, the user accesses the transmit holding register (COMTX); if read, the user accesses the receive buffer register (COMRX).

COMRX

This is an 8-bit register that the user can read received data from. If the ERBFI bit is set in COMIEN register, then an interrupt is generated when this register is fully loaded with the received data via serial input port.

Note that when the user sets the ERBFI bit while COMRX is already full, an interrupt is generated immediately.

COMTX

This is an 8-bit register that the user can write to with the data to be sent. If the ETBEI bit is set in the COMIEN register, an interrupt is generated when COMTX is empty.

Note that if the user sets ETBEI while COMTX is already empty, an interrupt is generated immediately.

UART Interrupt Enable Register

COMIEN is the interrupt enable register that is used to configure which interrupt source generates the interrupt. Only the lowest four bits in this register enable interrupts. Bit 4 and Bit 5 enable UART DMA signals. The UART DMA channel and interrupt must be configured in the DMA block.

Table 205: Interrupt Identification Table

COMIIR				
Bits[2:1], STA	**Bit 0, NINT**	**Priority**	**Definition**	**Clearing Operation**
00	1	N/A	No interrupt	N/A
11	0	1	Receive line status interrupt	Read COMLSR register
10	0	2	Receive buffer full interrupt	Read COMRX register
01	0	3	Transmit buffer empty interrupt	Write data to COMTX or read COMIIR
00	0	4	Modem status interrupt	Read COMMSR register

UART Divider Register

The baud rate divider register is a 16-bit register used to generate the baud rate for UART data transfer. The baud rate without fractional divider is a divided down version of the master clock, as shown in Figure 55.

$$Baud\ rate = UCLK \div (2 \times 16 \times COMDIV)$$

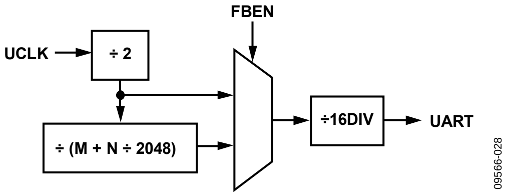


Figure 55. Baud Rate Generation

The generating of a fractional baud rate can be described by the following formula. The final baud rate of UART operation is shown in Figure 55.

$$Baudrate = \frac{\dfrac{UCLK}{2 \times (M + N/2048)}}{16 \times COMDIV}$$

REGISTER SUMMARY (UART)

Table 206. UART Register Summary

Address	Name	Description	Reset	RW
0x40005000	COMTX	Transmit Holding Register	0x00	W
0x40005000	COMRX	Receive Buffer Register	0x00	R
0x40005004	COMIEN	Interrupt Enable Register	0x00	RW
0x40005008	COMIIR	Interrupt Identification Register	0x01	R
0x4000500C	COMLCR	Line Control Register	0x00	RW
0x40005010	COMMCR	Module Control Register	0x00	RW
0x40005014	COMLSR	Line Status Register	0x60	R
0x40005018	COMMSR	Modem Status Register	0x00	R
0x40005024	COMFBR	Fractional Baud Rate Register.	0x0000	RW
0x40005028	COMDIV	Baud Rate Divider Register	0x0001	RW

REGISTER DETAILS (UART)

Transmit Holding Register

Address: 0x40005000, Reset: 0x00, Name: COMTX

Table 207. Bit Descriptions for COMTX

Bits	Bit Name	Description	Reset	Access
[7:0]	VALUE	Transmit Holding Register.	0x00	W

Receive Buffer Register

Address: 0x40005000, Reset: 0x00, Name: COMRX

Table 208. Bit Descriptions for COMRX

Bits	Bit Name	Description	Reset	Access
[7:0]	VALUE	Receive Buffer Register.	0x00	R

Interrupt Enable Register

Address: 0x40005004, Reset: 0x00, Name: COMIEN

Table 209. Bit Descriptions for COMIEN

Bits	Bit Name	Description	Reset	Access
[7:6]	RESERVED	Reserved.	0x0	R
5	EDMAR	DMA requests in transmit mode. 0: DIS. Disable. 1: EN. Enable.	0x0	RW
4	EDMAT	DMA requests in receive mode. 0: DIS. Disable. 1: EN. Enable.	0x0	RW
3	EDSSI	Modem status interrupt. Notes: This interrupt is generated when COMMSR[0] is set. 0: DIS. Disable. 1: EN. Enable.	0x0	RW
2	ELSI	Rx status interrupt. Notes: This interrupt is generated when any bit of COMLSR[4:1] is set. 0: DIS. Disable. 1: EN. Enable.	0x0	RW
1	ETBEI	Transmit buffer empty interrupt. 0: DIS. Disable. 1: EN. Enable the transmit interrupt. An interrupt is generated when the COMTX register is empty. Note that if the COMTX is already empty when enabling this bit, an interrupt is generated immediately.	0x0	RW
0	ERBFI	Receive buffer full interrupt. 0: DIS. Disable. 1: EN. Enable the receive interrupt. An interrupt is generated when the COMRX register is loaded with the received data. Note that if the COMRX is already full when enabling this bit, an interrupt is generated immediately.	0x0	RW

Interrupt Identification Register

Address: 0x40005008, Reset: 0x01, Name: COMIIR

Table 210. Bit Descriptions for COMIIR

Bits	Bit Name	Description	Reset	Access
[7:3]	RESERVED	Reserved.	0x00	R
[2:1]	STA	Status bits. 00: MODEMSTATUS. Modem status interrupt. 01: TXBUFEMPTY. Transmit buffer empty interrupt. 10: RXBUFFULL. Receive buffer full interrupt. Read COMRX register to clear. 11: RXLINESTATUS. Receive line status interrupt. Read COMLSR register to clear.	0x0	R
0	NINT	Interrupt flag. 0: CLR. Indicates any of the following: receive buffer full, transmit buffer empty, line status, or modem status interrupt occurred. 1: SET. There is no interrupt (default).	0x1	R

Line Control Register

Address: 0x4000500C, Reset: 0x00, Name: COMLCR

Table 211. Bit Descriptions for COMLCR

Bits	Bit Name	Description	Reset	Access
7	RESERVED	Reserved.	0x0	R
6	BRK	Set Break. 0: DIS to operate in normal mode. 1: EN to force TxD to 0.	0x0	RW
5	SP	Stick Parity. 0: DIS. Parity is not forced based on EPS and PEN values. 1: EN. Force parity to defined values based on EPS and PEN values. EPS = 1 and PEN = 1, parity forced to 1 EPS = 0 and PEN = 1, parity forced to 0 EPS = X and PEN = 0, no parity transmitted.	0x0	RW
4	EPS	Even Parity Select Bit. 0: DIS. Odd parity. 1: EN. Even parity.	0x0	RW
3	PEN	Parity Enable Bit. 0: DIS. No parity transmission or checking. 1: EN. Transmit and check the parity bit.	0x0	RW
2	STOP	Stop Bit. 0: DIS. Generate one stop bit in the transmitted data. 1: EN. Transmit 1.5 stop bits if the word length is 5 bits, or 2 stop bits if the word length is 6, 7, or 8 bits. The receiver checks the first stop bit only, regardless of the number of stop bits selected.	0x0	RW
[1:0]	WLS	Word Length Select Bits. 00: 5BITS. 5 bits. 01: 6BITS. 6 bits. 10: 7BITS. 7 bits. 11: 8BITS. 8 bits.	0x0	RW

Module Control Register

Address: 0x40005010, Reset: 0x00, Name: COMMCR

Table 212. Bit Descriptions for COMMCR

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved. 0 should be written to these bits.	0x0	R
4	LOOPBACK	Loop Back. 0: DIS. Normal mode. 1: EN. Enable loopback mode.	0x0	RW
[3:2]	RESERVED	Reserved. 0 should be written to these bits.	0x0	RW
1	RTS	Request to send output control bit. 0: DIS. Force the RTS output to 1. 1: EN. Force the RTS output to 0.	0x0	RW
0	RESERVED	Reserved. 0 should be written to this bit.	0x0	RW

Line Status Register

Address: 0x40005014, Reset: 0x60, Name: COMLSR

Table 213. Bit Descriptions for COMLSR

Bits	Bit Name	Description	Reset	Access
7	RESERVED	Reserved.	0x0	R
6	TEMT	COMTX and Shift Register Empty Status Bit. 0: CLR. Cleared when writing to COMTX. 1: SET. If COMTX and the shift register are empty, this bit indicates that the data has been transmitted, that is, it is no longer present in the shift register (default).	0x1	R
5	THRE	COMTX Empty Status Bit. 0: CLR. Cleared when writing to COMTX. 1: SET. If COMTX is empty, COMTX can be written as soon as this bit is set. The previous data may not have been transmitted yet and can still be present in the shift register (default).	0x1	R
4	BI	Break Indicator. 0: CLR. Cleared automatically. 1: SET. Set when UART RXD is held low for more than the maximum word length.	0x0	R
3	FE	Framing Error. 0: CLR. Cleared automatically. 1: SET. Set when the stop bit is invalid.	0x0	R
2	PE	Parity Error. 0: CLR. Cleared automatically. 1: SET. Set when a parity error occurs.	0x0	R
1	OE	Overrun Error. 0: CLR. Cleared automatically. 1: SET. Set automatically if data is overwritten before being read.	0x0	R
0	DR	Data Ready. 0: CLR. Cleared by reading COMRX. 1: SET. Set automatically when COMRX is full.	0x0	R

Modem Status Register

Address: 0x40005018, Reset: 0x00, Name: COMMSR

Table 214. Bit Descriptions for COMMSR

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved. 0 Should be written to these bits.	0x0	RW
4	CTS	Clear to send signal status bit. 0: CLR. Cleared to 0 when CTS input is logic high. 1: SET. Set to 1 when CTS input is logic low.	0x0	RW
[3:1]	RESERVED	Reserved. 0 Should be written to these bits.	0x0	RW
0	DCTS	Delta CTS. 0: DIS. Cleared automatically by reading COMMSR. 1: EN. Set automatically if CTS changed state since COMMSR last read.	0x0	RW

Fractional Baud Rate Register.

Address: 0x40005024, Reset: 0x0000, Name: COMFBR

Table 215. Bit Descriptions for COMFBR

Bits	Bit Name	Description	Reset	Access
15	ENABLE	Fractional baud rate generator enable bit. Used for more accurate baud rate generation. 0: DIS. Disable. 1: EN. Enable.	0x0	RW
[14:13]	RESERVED	Reserved.	0x0	R
[12:11]	DIVM	Fractional baud rate M divide bits (1 to 3). These bits should not be set to 0.	0x0	RW
[10:0]	DIVN	Fractional baud rate N divide bits (0 to 2047).	0x000	RW

Baud Rate Divider Register

Address: 0x40005028, Reset: 0x0001, Name: COMDIV

Table 216. Bit Descriptions for COMDIV

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Sets the baud rate. The COMDIV register should not be 0.	0x0001	RW

FLASH CONTROLLER

FLASH CONTROLLER FEATURES

- 128 kB Flash/EE
- 2 kB information space
- Flash controller

FLASH CONTROLLER OVERVIEW

The ADuCRF101 includes 128 kB of Flash/EE memory.

- User flash area of 128 kB, organized as 256 512 byte pages
- Minimum write size of four bytes
- Minimum erase block size is a 512-byte page
- 2 kB of information space, which contains factory code
- Flash controller

Read and write to flash are executed by direct access.

Commands Supported

- Mass erase and page erase
- Generation of signatures for single or multiple pages
- Command abort

Any access that the processor makes to the flash memory while a command is in progress is stalled until that command completes.

Flash Protection

- Write protection for user space
- Ability to lock serial wire interface for read protection

Flash Integrity

- Automatic signature check of kernel space on reset
- User signature for application code

FLASH MEMORY ORGANIZATION

The ADuCRF101 controller supports 128 kB of user flash and 2 kB of information space. The information space is memory mapped above user flash space as shown in Figure 56. The main 128 kB space is organized into 32 × flash blocks. Each flash block contains eight flash pages, each page is 512 bytes in size. The flash is protected in block segments, but it can be erased/written in page segments.

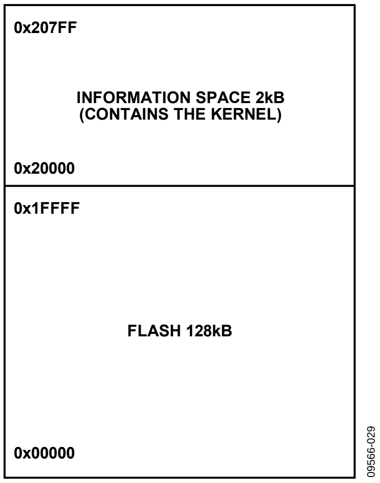
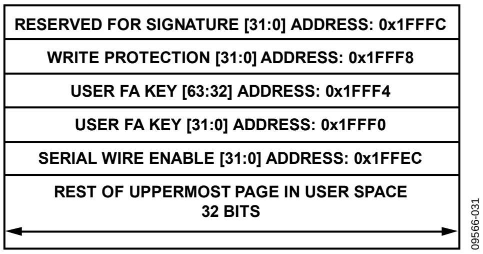


Figure 56. Information and User Space Memory Map on ADuCRF101

User Space

The top 20 bytes are shown in Figure 57. If a user tries to read to or write from a portion of memory that is not available, then a bus error is returned.



RESERVED FOR SIGNATURE [31:0] ADDRESS: 0x1FFFC

WRITE PROTECTION [31:0] ADDRESS: 0x1FFF8

USER FA KEY [63:32] ADDRESS: 0x1FFF4

USER FA KEY [31:0] ADDRESS: 0x1FFF0

SERIAL WIRE ENABLE [31:0] ADDRESS: 0x1FFEC

REST OF UPPERMOST PAGE IN USER SPACE
32 BITS

Figure 57. Uppermost Page of User Memory

Kernel Space

The kernel space is reserved for test data and the serial downloader and is read protected. Only 16 bytes, or four (32-bit) locations in the kernel space are user readable. These locations are at Address 0x207E8, Address 0x207EC, Address 0x207F0, and Address 0x207F4. Byte Location 0x207E8 is used to store kernel revision in ASCII format. Location 0x207EC is not used and reads 0xFF. Location 0x207F0 and Location 0x207F4 are used to identify a device, as described in Table 217.

Table 217. Silicon Identification Register Bit Descriptions, Address: 0x207F4 and Address: 0x207F0

Bits	Name	Description
[63:58]	Lot ID Character 6	Lot ID characters are encoded in six bits. A to Z or a to z = 0 to 25 0 to 9 = 26 to 36 Blank space or 0xFF = 37
[57:52]	Lot ID Character 5	
[51:46]	Lot ID Character 4	
[45:40]	Lot ID Character 3	
[39:34]	Lot ID Character 2	
[33:28]	Lot ID Character 1	
[27:22]	Lot ID Character 0	
[21:16]	Wafer	Wafer number encoded in six bits = 0 to 63
[15:8]	Y	Y-coordinate encoded in eight bits = 0 to 255 (−127 to +127)
[7:0]	X	X-coordinate encoded in eight bits = 0 to 255 (−127 to +127)

WRITING TO FLASH/EE MEMORY

Flash is written directly, similarly to SRAM. To enable writes to the flash memory, user code must first set FEECON0[2] to 1. When the flash write sequence is complete as indicated by FEESTA[3], user code should then clear FEECON0[2]. After these bits are configured and flash protection is disabled, user code can then write to the flash. When a flash erase or write operation is in progress, any access by the Cortex-M3 processor to the flash is stalled until the erase/write operation is completed.

Note that only 32-bit writes are supported. 16-bit and 8-bit writes are not supported. Burst writes to flash are supported.

If the addressed location is write-protected, the controller responds with a bus fault system exception error. This will be escalated to a hard fault exception if the bus fault exception handler is not enabled.

If a write is followed immediately by a second write to the next location in flash and this is in the same row as the previous write, then the flash write time is faster as the controller does not need to change the row address.

Note that only a store multiple assembly instruction takes advantage of this flash feature.

Before performing a write to Flash/EE memory, the FEESTA register should be read and checked to ensure that no other commands or writes are being executed.

ERASING FLASH/EE MEMORY

User code can call two flash erase commands:

- Mass erase. This command erases the entire user flash memory. After entering the user protection key into the FEEKEY register, write the mass erase command to FEECMD.
- Page erase. This command erases the 512-byte flash page selected by FEEADR0L/FEEADR0H. After entering the user protection key into the FEEKEY register, load the FEEADR0L/FEEADR0H registers with the page address to be erased. Finally, write the page erase command to FEECMD. CMDDONE (FEESTA[2]) indicates that the page erase command is completed.

During a page or mass erase sequence, the flash controller and flash block consume extra current for the duration of the flash erase sequence.

FLASH CONTROLLER PERFORMANCE AND COMMAND DURATION

Direct single write access (32-bit location): 46 µs.

Mass erase: 21 ms.

Page erase: 21 ms.

Direct write of a page (128 32-bit locations): 3.04 ms.

Mass verify for all of user space: 2.05 ms (16 MHz HCLK).

Sign for all of user space: 2.05 ms (16 MHz HCLK).

Note that these flash timings are not dependent on the clock dividers.

FLASH PROTECTION

There are three types of protection implemented:

- Key protection
- Read protection
- Write protection

Flash Protection: Key Protection

Some of the flash controller registers are key-protected to avoid accidental writes to these registers.

The user key is 0xF123F456. It is entered via the 16-bit Key Register, 0xF456 first, followed by 0xF123. This key must be entered to run certain user commands or write to certain locations in flash or to enable write access to the setup register. Once entered, the key remains asserted unless a command is written to the FEECMD register. When the command completes, the key clears automatically. If this key is entered to enable write access to the setup register or to enable writes to certain locations in flash, it needs to be cleared by user code afterwards. To clear it, write any 16-bit value to the key register.

Flash Protection: User Read Protection

User space read protection is provided by disabling serial wire access. Two methods are provided to disable serial wire access, a permanent method using flash location 0x1FFEC, or a temporary method using the FEECON1.

Temporary User Read Protection: Using the FEECON1 Register

User code can write 0 to DBG in the flash FEECON1 register. The read protection takes immediate effect and if the write to FEECON1 is executed during a debugging session, the session will be aborted. When using this method, it is recommended to write protect the page where the write to FEECON1 instruction is stored, as well as the last page of the user flash. This ensures that the protections cannot be erased. This method is temporary as the FEECON1 register returns to 1 after exiting the kernel after reset.

Permanent User Read Protection: Using Flash Location 0x1FFEC

Serial wire access is temporarily disabled during kernel execution, after power up or reset. Before exiting and starting user code execution, the kernel reads the serial wire enable location 0x1FFEC:

- If 0x16032010 is present, the FEECON1 register is not modified and serial wire access stays disabled.
- If any other value is present, it enables serial wire access by writing 1 to DBG in the flash FEECON1 register.

To write 0x16032010 at Address 0x1FFEC, there are two methods:

- This value can be declared as a constant in the startup code and included as part of the file programmed into Flash.
- User code can write the value at that location. Note that the location is key protected.

To ensure read protection is permanent, the last block of Flash that contains the flash protection should also be write protected.

Flash Protection: User Write Protection

User write protection is provided to prevent accidental writes to pages in user space and to protect blocks of user code when downloading extra code to flash. If a write or erase of a protected location is detected, then the ADuCRF101 flash controller generates an exception.

For write protection, memory is split into 32 blocks of eight pages each, or 32 blocks of 4 k bytes (1 page being 512 bytes).

Like for the user read protection, two methods are provided for the read protection, a permanent method using flash location 0x1FFFF8 and a temporary method using the FEEPRO register.

Temporary User Write Protection: Using the FEEPRO Register

If the permanent write protection in flash has not been programmed, then the FEEPRO register can be written to directly from user code. This allows a user to verify the write protection before committing it to flash.

Permanent User Write Protection: Using Flash Location 0x1FFF8

The permanent write protection is stored at the top of user space. The top four bytes are reserved for a signature and the next four bytes are for write protection. The write protection is uploaded by the flash controller into local registers after reset. After uploading, the 32 write protection bits can be read via memory mapped Register FEEPRO[31:0]. To write to the write protection bits, a user must first write 0xF456, followed by 0xF123 to the key register. After the write protection has been written, it cannot be rewritten without a full erase of user space. After a mass erase, the device must be reset to de-assert the uploaded copy of the write protection bits.

The following is the sequence to program the permanent write protection word:

- Write 0xF456 followed by 0xF123 to the key register.
- Write the required write protection directly to flash. Write 0 to enable protection. The write protection address is 0x1FFF8.
- Verify that the write completed successfully by polling the status register, FEESTA[3], or by enabling a write complete interrupt.
- Reset the device and the Write Protection field is uploaded from the kernel space and activated by the flash controller.

If the write protection in flash has been programmed, then the MSB of the write protection must be programmed to 0 to prevent erasing of the write protection block.

If an attempt is made to write to the write protection word in flash without setting the FEEKEY first, a bus error is generated.

FLASH CONTROLLER FAILURE ANALYSIS KEY

It may be necessary to perform failure analysis on parts even though read protection is enabled. A method has been provided to allow Analog Devices to subsequently access the memory when provided with the correct 64 bit key by the user.

This 64-bit key that is stored at the top of user space in flash as shown in Figure 57.

It is used to gain access to user code if the serial wire interface has been locked. It is the user's responsibility to program this key to a value. The key must be given to Analog Devices to enable access to user code. Note that these two locations are key protected if written by user code. If an attempt to write these locations without the FEEKEY is made, a bus error is generated. The FA key can also be programmed directly by defining it as constant in the start-up file.

FLASH INTEGRITY SIGNATURE FEATURE

The signature is used to check the integrity of the flash device. Software can call a signature check command occasionally or whenever a new block of code is about to be executed. The signature is a 24-bit CRC with the polynomial $x^{24} + x^{23} + x^6 + x^5 + x + 1$.

The sign command can be used to generate a signature and check the signature of a block of code, where a block can be a single page or multiple pages. A 24-bit LFSR is used to generate the signature. The hardware assumes that the signature for a block is stored in the upper four bytes of the most significant page of a block; therefore, these four bytes are not included when generating the signature.

Use the following procedure to generate a signature:

1. Write the start address of the block to the FEEADR0L/FEEADR0L H register.
2. Write the end address of the block to the FEEADR1L/ FEEADR1LH register.
3. Write the Sign command to the command register (FEECMD = 0x2).

When the command has completed, the signature is available in the sign register. The signature is compared with the data stored in the upper four bytes of the uppermost page of the block. If the data does not match the signature, a fail status is returned in the status register (FEESTA[5:4] = 10).

While the signature is being computed all other accesses to flash are stalled. For a 128 kB block, that is 32 k reads.

Note that the FEEADR0L/FEEADR0H and FEEADR1L/FEEADR1H addresses are byte addresses, but only pages need to be identified, that is, the lower nine bits are ignored by the hardware.

Note that the user must run the CRC polynomial in user code first to generate the CRC value and must write this to the upper four bytes of the uppermost page of a block. When this operation is complete, any call of the signature feature compares this 4-byte value to the result of the signature check function.

INTEGRITY OF THE KERNEL

The hardware automatically checks the integrity of the kernel after reset. In the event of a failure, FEESTA[6] is set and user code cannot run. This bit can only be read via a serial wire read if the serial wire interface is enabled.

ABORT USING INTERRUPTS

Commands (erase, sign, or mass verify) and writes can be aborted on receipt of an interrupt as listed in the System Exceptions and Peripheral Interrupts section. Aborts are also possible by writing an abort command to the FEECMD register. However, if flash is being programmed and the routine controlling the programming is in flash, it is not possible to use the abort command to abort the cycle because instructions cannot be read. Therefore, the ability to abort a cycle on the assertion of any system interrupt is provided. The FEEAENx register is used to enable aborts on receipt of an interrupt.

When a command or write is aborted via a system interrupt, FEESTA[5:4] indicates an abort (FEESTA[5:4] = 0x3).

It is not possible to abort a flash write or erase cycle without waiting for the high voltage in the flash core to discharge first, that is, a write cycle must finish with a waiting time of 6.6 µs for the high voltage to discharge. An erase cycle must finish with a waiting time of 6.6 µs. A mass erase cycle must finish with a waiting time of 121 µs.

Depending on the state that a write cycle is in when the abort asserts, the write cycle may or may not complete. If the write or erase cycle did not complete successfully, a fail status of aborted can be read in the status register.

If an immediate response to an interrupt is required during an erase or program cycle then the interrupt service routine and the interrupt vector table must be moved to SRAM for the duration of the cycle.

If the DMA engine is set up to write a block of data to flash, then an interrupt can be set up to abort the current write; however, the DMA engine starts the next write immediately. The interrupt causing the abort stays asserted so that there is a number of aborted write cycles in this case before the processor gains access to flash.

When an abort is triggered by an interrupt, all commands are repeatedly aborted until the appropriate FEEAENx bit is cleared or the interrupt source is cleared.

REGISTER SUMMARY (FLASH CONTROLLER)

Table 218. Flash Controller Register Summary

Address	Name	Description	Reset	RW
0x40002800	FEESTA	Status Register	0x0000	R
0x40002804	FEECON0	Command Control Register	0x0000	RW
0x40002808	FEECMD	Command Register	0x0000	RW
0x40002810	FEEADR0L	Address 0 LSB	0x0000	RW
0x40002814	FEEADR0H	Address 0 MSB	0x0000	RW
0x40002818	FEEADR1L	Address1 LSB	0x0000	RW
0x4000281C	FEEADR1H	Address1 MSB	0x0000	RW
0x40002820	FEEKEY	Key Register	0x0000	W
0x40002828	FEEPROL	Write Protection Register LSB	0xFFFF	RW
0x4000282C	FEEPROH	Write Protection Register MSB	0xFFFF	RW
0x40002830	FEESIGL	Signature LSB	0xFFFF	R
0x40002834	FEESIGH	Signature MSB	0xFFFF	R
0x40002838	FEECON1	User Setup Register	0x0001	RW
0x40002848	FEEADRAL	Abort Address Register LSB	0x0800	R
0x4000284C	FEEADRAH	Abort Address Register MSB	0x0002	R
0x40002878	FEEAEN0	Interrupt Abort Register (Interrupt 15 to Interrupt 0)	0x0000	RW
0x4000287C	FEEAEN1	Interrupt Abort Register (Interrupt 31 to Interrupt 16)	0x0000	RW
0x40002880	FEEAEN2	Interrupt Abort Register (Interrupt 42 to Interrupt 32)	0x0000	RW

REGISTER DETAILS (FLASH CONTROLLER)

Status Register

Address: 0x40002800, Reset: 0x0000, Name: FEESTA

Table 219. Bit Descriptions for FEESTA

Bits	Bit Name	Description	Reset	Access
[15:7]	RESERVED	Reserved.	0x000	R
6	SIGNERR	Kernel space signature check on reset error. 0: CLR. Cleared, if the signature check of the kernel passes. 1: SET. Set, if the signature check of the kernel fails. User code does not execute.	0x0	R
[5:4]	CMDRES	These two bits indicate the status of a command on completion or the status of a write. If multiple commands are executed or there are multiple writes without a read of the status register, then the first error encountered is stored. 00: SUCCESS. Indicates a successful completion of a command or a write. Also cleared after a read of FEESTA. 01: PROTECTED. Indicates an attempted erase of a protected location. 10: VERIFYERR. Indicates a read verify error. After an erase the controller reads the corresponding word(s) to verify that the transaction completed successfully. If data read is not all 'F's this is the resulting status. If the sign command is executed and the resulting signature does not match the data in the upper 4 bytes of the upper page in a block, then this is the resulting status. 11: ABORT. Indicates that a command or a write was aborted by an abort command or a system interrupt has caused an abort.	0x0	R
3	WRDONE	Write complete. 0: CLR. Cleared after a read of FEESTA. 1: SET. Set when a write completes. If there are multiple writes or a burst write, this status bit asserts after the first long word written and stays asserted until read. If there is a burst write to flash, then this bit asserts after every long word written, assuming that user code read FEESTA after every long word written.	0x0	R
2	CMDDONE	Command complete. 0: CLR. Cleared after a read of FEESTA. 1: SET. Set when a command completes. If there are multiple commands, this status bit asserts after the first command completes and stays asserted until read.	0x0	R
1	WRBUSY	Write busy. 0: CLR. Cleared after a read of FEESTA. 1: SET. Set when the flash block is executing a write.	0x0	R
0	CMDBUSY	Command busy. 0: CLR. Cleared after a read of FEESTA. 1: SET. Set when the flash block is executing any command entered via the command register.	0x0	R

Command Control Register

Address: 0x40002804, Reset: 0x0000, Name: FEECON0

Table 220. Bit Descriptions for FEECON0

Bits	Bit Name	Description	Reset	Access
[15:3]	RESERVED	Reserved.	0x0000	R
2	WREN	Write enable bit. 0: DIS. Disables flash writes. A flash write when this bit is 0 results in a bus fault system exception error and the write does not take place. 1: EN. Enables flash writes.	0x0	RW

Bits	Bit Name	Description	Reset	Access
1	IENERR	Error interrupt enable bit. 0: DIS. Disables the Flash error interrupt. 1: EN. An interrupt is generated when a command or flash write completes with an error status.	0x0	RW
0	IENCMD	Command complete interrupt enable bit. 0: DIS. Disables the Flash command complete interrupt. 1: EN. An interrupt is generated when a command or flash write completes.	0x0	RW

Command Register

Address: 0x40002808, Reset: 0x0000, Name: FEECMD

Table 221. Bit Descriptions for FEECMD

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x000	R
[3:0]	CMD	Commands supported by the flash controller. Note that any core access to the flash is stalled until the command completes. 0000: IDLE. No command executed. 0001: ERASEPAGE. Write the address of the page to be erased to FEEADR0L/H, then write this code to the FEECMD register and the flash erases the page. When the erase is complete, the flash reads every location in the page to verify that all words in the page are erased. If there is a read verify error, this is indicated in FEESTA. To erase multiple pages, wait until a previous page erase has completed. Check the status, and then issue a command to start the next page erase. Before entering this command, 0xF456 followed by 0xF123 must be written to the key register. 0010: SIGN. Use this command to generate a signature for a block of data. The signature is generated on a page-by-page basis. To generate a signature, the address of the first page of the block is entered in FEEADR0L/FEEADR0H. The address of the last page is written to FEEADR1L/FEEADR1H. Then write this code to the FEECMD register. When the command has completed, the signature is available for reading in FEESIGL/FEESIGH. The last four bytes of the last page in a block is reserved for storing the signature. Before entering this command, 0xF456 followed 0xF123 must be written to the key register. 0011: MASSERASE. Erase all of user space. To enable this operation, 0xF456 followed by 0xF123 must first be written to FEEKEY (this is to prevent accidental erases). When the mass erase has completed, the controller reads every location to verify that all locations are 0xFFFFFFFF. If there is a read verify error this is indicated in FEESTA. 0100: ABORT. If this command is issued, then any command currently in progress is stopped. The status indicates command completed with an error status in FEESTA[5:4]. Note that this is the only command that can be issued while another command is already in progress. This command can also be used to stop a write that may be in progress. If a write is aborted, the address of the location being written can be read via the FEEADRAL/ FEEADRAH register. While the flash controller is writing one long word, another long word write may be in the pipeline from the Cortex-M3 or DMA engine (depending on how the software implements writes). Therefore, both writes may need to be aborted. If a write or erase is aborted, then the flash timing is violated and it is not possible to determine if the write or erase completed successfully. To enable this operation, 0xF456 followed by 0xF123 must first be written to FEEKEY (this is to prevent accidental aborts).	0x0	RW

Address 0 LSB Register

Address: 0x40002810, Reset: 0x0000, Name: FEEADR0L

Table 222. Bit Descriptions for FEEADR0L

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Used in conjunction with FEEADR0H, to indicate the page to be erased, or the start of a section to be signed. The address of a memory location inside the page should be stored in FEEADR0L/H, bits[15:0] in FEEADR0L, and bits[17:16] in FEEADR0H. The 9 LSBs of the address are ignored.	0x0000	RW

Address 0 MSB Register

Address: 0x40002814, Reset: 0x0000, Name: FEEADR0H

Table 223. Bit Descriptions for FEEADR0H

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved.	0x0000	R
[1:0]	VALUE	Used in conjunction with FEEADR0L, to indicate the page to be erased, or the start of a section to be signed. The address of a memory location inside the page should be stored in FEEADR0L/H, Bits[15:0] in FEEADR0L, and Bits[17:16] in FEEADR0H.	0x0	RW

Address1 LSB Register

Address: 0x40002818, Reset: 0x0000, Name: FEEADR1L

Table 224. Bit Descriptions for FEEADR1L

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Used in conjunction with FEEADR1H, to identify the last page used by the sign command. The address of a memory location inside the page should be stored in FEEADR1L/H, Bits[15:0] in FEEADR1L, and Bits[17:16] in FEEADR1H. The 9 LSBs of the address are ignored.	0x0000	RW

Address1 MSB Register

Address: 0x4000281C, Reset: 0x0000, Name: FEEADR1H

Table 225. Bit Descriptions for FEEADR1H

Bits	Bit Name	Description	Reset	Access
[15:2]	RESERVED	Reserved.	0x0000	R
[1:0]	VALUE	Used in conjunction with FEEADR1L, to identify the last page used by the sign command. The address of a memory location inside the page should be stored in FEEADR1L/H, Bits[15:0] in FEEADR1L, and Bits[17:16] in FEEADR1H.	0x0	RW

Key Register

Address: 0x40002820, Reset: 0x0000, Name: FEEKEY

Table 226. Bit Descriptions for FEEKEY

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Enter 0xF456 followed by 0xF123. Returns 0x0 if read. 1111010001010110: USERKEY1 1111000100100011: USERKEY2	0x0000	W

Write Protection Register LSB

Address: 0x40002828, Reset: 0xFFFF, Name: FEEPROL

Table 227. Bit Descriptions for FEEPROL

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Lower 16 bits of the write protection. This register is read only if the write protection in flash has been programmed, that is, FEEPROH/FEEPROL have previously been configured to protect pages. 0: Protect a section of flash. 1: Leave a flash block unprotected.	0xFFFF	RW

Write Protection Register MSB

Address: 0x4000282C, Reset: 0xFFFF, Name: FEEPROH

Table 228. Bit Descriptions for FEEPROH

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Upper 16 bits of the write protection. This register is read only if the write protection in flash has been programmed, that is, FEEPROH/FEEPROL have previously been configured to protect pages. 0: Protect a section of flash. 1: Leave a flash block unprotected.	0xFFFF	RW

Signature LSB Register

Address: 0x40002830, Reset: 0xFFFF, Name: FEESIGL

Table 229. Bit Descriptions for FEESIGL

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Lower 16 bits of the signature. Signature Bits[15:0].	0xFFFF	R

Signature MSB Register

Address: 0x40002834, Reset: 0xFFFF, Name: FEESIGH

Table 230. Bit Descriptions for FEESIGH

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0xFF	R
[7:0]	VALUE	Upper eight bits of the signature. Signature Bits[23:16].	0xFF	R

User Setup Register

Address: 0x40002838, Reset: 0x0001, Name: FEECON1

Note that the FEECON1 register is key protected. The key must be entered in FEEKEY. After writing to FEECON1, a 16-bit value must be written again to FEEKEY, to reassert the key protection.

Table 231. Bit Descriptions for FEECON1

Bits	Bit Name	Description	Reset	Access
[15:1]	RESERVED	Reserved.	0x0000	R
0	DBG	Serial wire debug enable. Note that the kernel sets this bit to 1 when it has finished executing, thus enabling debug access to a user. If the serial wire enable location in the last page of flash is set to 0x16032010 by user code when the kernel loads, it disables serial wire access while user code is running, therefore, the kernel sets this bit to 0 when it has finished executing. If the serial wire enable location is not set to 0x16032010 by user code when the kernel loads, it enables serial wire access while user code is running, therefore, the kernel sets this bit to 1 when it has finished executing. 0: DIS. Disable access via the serial wire debug interface. 1: EN. Enable access via the serial wire debug interface.	0x1	RW

Abort Address Register LSB

Address: 0x40002848, Reset: 0x0800, Name: FEEADRAL

Table 232. Bit Descriptions for FEEADRAL

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Lower 16 bits of the FEEADRA register. If a write is aborted then this location contains the address of the location being written when the write was aborted.	0x0800	R

Abort Address Register MSB

Address: 0x4000284C, Reset: 0x0002, Name: FEEADRAH

Table 233. Bit Descriptions for FEEADRAH

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Upper 16 bits of the FEEADRA register.	0x0002	R

Interrupt Abort Register (Interrupt 15 to Interrupt 0)

Address: 0x40002878, Reset: 0x0000, Name: FEEAEN0

Table 234. Bit Descriptions for FEEAEN0

Bits	Bit Name	Description	Reset	Access
15	FEE	Flash controller interrupt abort enable bit. 0: DIS. Flash controller interrupt abort disabled. 1: EN. Flash controller interrupt abort enabled.	0x0	RW
14	ADC	ADC interrupt abort enable bit. 0: DIS. ADC interrupt abort disabled. 1: EN. ADC interrupt abort enabled.	0x0	RW
13	T1	Timer1 interrupt abort enable bit. 0: DIS. Timer1 interrupt abort disabled. 1: EN. Timer1 interrupt abort enabled.	0x0	RW
12	T0	Timer0 interrupt abort enable bit. 0: DIS. Timer0 interrupt abort disabled. 1: EN. Timer0 interrupt abort enabled.	0x0	RW
11	RESERVED	Reserved.	0x0	R
10	T3	Timer3 interrupt abort enable bit. 0: DIS. Timer3 interrupt abort disabled. 1: EN. Timer3 interrupt abort enabled.	0x0	RW

Bits	Bit Name	Description	Reset	Access
9	EXTINT8	External Interrupt 8 abort enable bit. 0: DIS. External Interrupt 8 abort disabled. 1: EN. External Interrupt 8 abort enabled.	0x0	RW
8	EXTINT7	External Interrupt 7 abort enable bit. 0: DIS. External Interrupt 7 abort disabled. 1: EN. External Interrupt 7 abort enabled.	0x0	RW
7	EXTINT6	External Interrupt 6 abort enable bit. 0: DIS. External Interrupt 6 abort disabled. 1: EN. External Interrupt 6 abort enabled.	0x0	RW
6	EXTINT5	External Interrupt 5 abort enable bit. 0: DIS. External Interrupt 5 abort disabled. 1: EN. External Interrupt 5 abort enabled.	0x0	RW
5	EXTINT4	External Interrupt 4 abort enable bit. 0: DIS. External Interrupt 4 abort disabled. 1: EN. External Interrupt 4 abort enabled.	0x0	RW
4	EXTINT3	External Interrupt 3 abort enable bit. 0: DIS. External Interrupt 3 abort disabled. 1: EN. External Interrupt 3 abort enabled.	0x0	RW
3	EXTINT2	External Interrupt 2 abort enable bit. 0: DIS. External Interrupt 2 abort disabled. 1: EN. External Interrupt 2 abort enabled.	0x0	RW
2	EXTINT1	External Interrupt 1 abort enable bit. 0: DIS. External Interrupt 1 abort disabled. 1: EN. External Interrupt 1 abort enabled.	0x0	RW
1	EXTINT0	External Interrupt 0 abort enable bit. 0: DIS. External Interrupt 0 abort disabled. 1: EN. External Interrupt 0 abort enabled.	0x0	RW
0	T2	Timer 2 interrupt abort enable bit. 0: DIS. Timer 2 interrupt abort disabled. 1: EN. Timer 2 interrupt abort enabled	0x0	RW

Interrupt Abort Register (Interrupt 31 to Interrupt 16)

Address: 0x4000287C, Reset: 0x0000, Name: FEEAEN1

Table 235. Bit Descriptions for FEEAEN1

Bits	Bit Name	Description	Reset	Access
15	DMAI2CMRX	I^{2}C master Rx DMA interrupt abort enable bit. 0: DIS. I^{2}C master Rx DMA interrupt abort disabled. 1: EN. I^{2}C master Rx DMA interrupt abort enabled.	0x0	RW
14	DMAI2CMTX	I^{2}C master Tx DMA interrupt abort enable bit. 0: DIS. I^{2}C master Tx DMA interrupt abort disabled. 1: EN. I^{2}C master Tx DMA interrupt abort enabled.	0x0	RW
13	DMAI2CSRX	I^{2}C slave Rx DMA interrupt abort enable bit. 0: DIS. I^{2}C slave Rx DMA interrupt abort disabled. 1: EN. I^{2}C slave Rx DMA interrupt abort enabled.	0x0	RW
12	DMAI2CSTX	I^{2}C slave Tx DMA interrupt abort enable bit. 0: DIS. I^{2}C slave Tx DMA interrupt abort disabled. 1: EN. I^{2}C slave Tx DMA interrupt abort enabled.	0x0	RW
11	DMAUARTRX	UARTRx DMA interrupt abort enable bit. 0: DIS. UARTRx DMA interrupt abort disabled. 1: EN. UARTRx DMA interrupt abort enabled.	0x0	RW

Bits	Bit Name	Description	Reset	Access
10	DMAUARTTX	UARTTx DMA interrupt abort enable bit. 0: DIS. UARTTx DMA interrupt abort disabled. 1: EN. UARTTx DMA interrupt abort enabled.	0x0	RW
9	DMASPI1RX	SPI1RXx DMA interrupt abort enable bit. 0: DIS. SPI1Rx DMA interrupt abort disabled. 1: EN. SPI1Rx DMA interrupt abort enabled.	0x0	RW
8	DMASPI1TX	SPI1Tx DMA interrupt abort enable bit. 0: DIS. SPI1Tx DMA interrupt abort disabled. 1: EN. SPI1Tx DMA interrupt abort enabled.	0x0	RW
7	DMAERROR	DMA error interrupt abort enable bit. 0: DIS. DMA error interrupt abort disabled. 1: EN. DMA error interrupt abort enabled.	0x0	RW
[6:5]	RESERVED	Reserved.	0x0	R
4	I2CM	I^{2}C master interrupt abort enable bit. 0: DIS. I^{2}C slave interrupt abort disabled. 1: EN. I^{2}C master interrupt abort enabled.	0x0	RW
3	I2CS	I^{2}C slave interrupt abort enable bit. 0: DIS. I^{2}C slave interrupt abort disabled. 1: EN. I^{2}C slave interrupt abort enabled.	0x0	RW
2	SPI1	SPI1 interrupt abort enable bit. 0: DIS. SPI1 interrupt abort disabled. 1: EN. SPI1 interrupt abort enabled.	0x0	RW
1	SPI0	SPI0 interrupt abort enable bit. 0: DIS. SPI0 interrupt abort disabled. 1: EN. SPI0 interrupt abort enabled.	0x0	RW
0	UART	UART interrupt abort enable bit. 0: DIS. UART interrupt abort disabled. 1: EN. UART interrupt abort enabled.	0x0	RW

Interrupt Abort Register (Interrupt 42 to Interrupt 32)

Address: 0x40002880, Reset: 0x0000, Name: FEEAEN2

Table 236. Bit Descriptions for FEEAEN2

Bits	Bit Name	Description	Reset	Access
[15:11]	RESERVED	Reserved.	0x00	R
10	PWM3	PWM3 interrupt abort enable bit. 0: DIS. PWM3 interrupt abort disabled. 1: EN. PWM3 interrupt abort enabled.	0x0	RW
9	PWM2	PWM2 interrupt abort enable bit. 0: DIS. PWM2 interrupt abort disabled. 1: EN. PWM2 interrupt abort enabled.	0x0	RW
8	PWM1	PWM1 interrupt abort enable bit. 0: DIS. PWM1 interrupt abort disabled. 1: EN. PWM1 interrupt abort enabled.	0x0	RW
7	PWM0	PWM0 interrupt abort enable bit. 0: DIS. PWM0 interrupt abort disabled. 1: EN. PWM0 interrupt abort enabled.	0x0	RW
6	PWMTRIP	PWMTRIP interrupt abort enable bit. 0: DIS. PWMTRIP interrupt abort disabled. 1: EN. PWMTRIP interrupt abort enabled.	0x0	RW

Bits	Bit Name	Description	Reset	Access
5	DMASPI0RX	SPI0Rx DMA interrupt abort enable bit. 0: DIS. SPI0Rx DMA interrupt abort disabled. 1: EN. SPI0RxDMA interrupt abort enabled.	0x0	RW
4	DMASPI0TX	SPI0Tx DMA interrupt abort enable bit. 0: DIS. SPI0Tx DMA interrupt abort disabled. 1: EN. SPI0Tx DMA interrupt abort enabled.	0x0	RW
3	DMAADC	ADC DMA interrupt abort enable bit. 0: DIS. ADC DMA interrupt abort disabled. 1: EN. ADC DMA interrupt abort enabled.	0x0	RW
[2:0]	RESERVED	Reserved.	0x0	R

DMA CONTROLLER

DMA FEATURES

There are 11 dedicated DMA channels.

DMA OVERVIEW

Direct memory access (DMA) is used to provide high speed data transfer between peripherals and memory. Data can be moved quickly by DMA without any processor actions. This keeps the processor resources free for other operations.

The DMA controller has 14 channels in total, three of which are reserved. The 11 used channels are dedicated to managing DMA requests from specific peripherals. Channels are assigned as shown in Table 237.

Note that CLKACT[0] must be set to 1 to enable the system clock to the DMA block, thus enabling the DMA operation.

Table 237. DMA Channel Assignment

Channel	Peripheral
0	SPI1 Tx
1	SPI1 Rx
2	UART Tx
3	UART Rx
4	I^{2}C slave Tx
5	I^{2}C slave Rx
6	I^{2}C master Tx
7	I^{2}C master Rx
8-10	Reserved
11	ADC
12	SPI0 Tx
13	SPI0 Rx

The channels are connected to dedicated hardware DMA requests; a software trigger is also supported on each channel. This configuration is done by software.

Each DMA channel has a programmable priority level default or high. Within a priority level, arbitration is done using a fixed priority that is determined by the DMA channel number.

The DMA controller supports multiple DMA transfer data widths: independent source and destination transfer size (byte, half word, and word). Source/destination addresses must be aligned on the data size.

The DMA transfer error interrupt is generated when a bus error condition occurs during a DMA transfer.

The DMA controller supports peripheral-to-memory, memory-to-peripheral and memory-to-memory transfers and access to flash or SRAM, as source and destination.

DMA OPERATION

The DMA controller performs direct memory transfer by sharing the system bus with the Cortex-M3 processor. The DMA request may stall the processor access to the system bus for some bus cycles, when the processor and DMA are targeting the same destination (memory or peripheral).

ERROR MANAGEMENT

A DMA transfer error can be generated by reading from or writing to a reserved address space. When a DMA transfer error occurs during a DMA read or a write access, the faulty channel is automatically disabled. If the DMA error interrupt is enabled in the NVIC, the error also generates an interrupt.

INTERRUPTS

An interrupt can be produced when a transfer is complete for each DMA channel. Separate interrupt enable bits are available in the NVIC for each of the DMA channels.

The DMA controller fetches channel control data structures located in the SRAM memory to perform data transfers. When enabled to use DMA operation, the DMA-capable peripherals request the DMA controller for transfer. At the end of the programmed number of DMA transfers for a channel, the DMA controller generates an interrupt corresponding to that channel. This interrupt indicates the completion of the DMA transfer.

DMA PRIORITY

The priority of a channel is determined by its number and priority level. Each channel can have two priority levels: default or high. All channels at high priority level have higher priority than all channels at default priority level. At the same priority level, a channel with a lower channel number has higher priority than a channel with a higher channel number. The DMA channel priority levels can be changed by writing into the appropriate bit in the DMAPRISET register.

CHANNEL CONTROL DATA STRUCTURE

Every channel has two control data structures associated with it: primary data structure and an alternate data structure. For simple transfer modes, the DMA controller uses either the primary or the alternate data structure. For more complex data transfer modes such as ping-pong or scatter-gather, the DMA controller uses both the primary and alternate data structures. Each control data structure (primary or alternate) occupies four 32-bit locations in the memory as shown in Table 238. The entire channel control data structure is shown in Table 239.

Table 238. Channel Control Data Structure

Offset	Name	Description
0x00	SRC_END_PTR	Source end pointer
0x04	DST_END_PTR	Destination end pointer
0x08	CHNL_CFG	Control data configuration
0x0C	Reserved	Reserved

Before the controller can perform a DMA transfer, the data structure related to the DMA channel must be programmed at the designated location in system memory, SRAM.

- The source end pointer memory location contains the end address of the source data.
- The destination end pointer memory location contains the end address of the destination data.
- The control data configuration memory location contains the channel configuration control data.

The programming determines the source and destination data size, number of transfers, and the number of arbitrations.

Table 239. Memory Map of Primary and Alternate DMA Structures

Channel	Primary Structures		Alternate Structures	
Channel 13	Reserved; set to 0	0x0DC	Reserved; set to 0	0x1DC
	Control	0x0D8	Control	0x1D8
	Destination end pointer	0x0D4	Destination end pointer	0x1D4
	Source end pointer	0x0D0	Source end pointer	0x1D0
Channel 12	Reserved; set to 0	0x0CC	Reserved; set to 0	0x1CC
	Control	0x0C8	Control	0x1C8
	Destination end pointer	0x0C4	Destination end pointer	0x1C4
	Source end pointer	0x0C0	Source end pointer	0x1C0
Channel 11	Reserved; set to 0	0x0BC	Reserved; set to 0	0x1BC
	Control	0x0B8	Control	0x1B8
	Destination end pointer	0x0B4	Destination end pointer	0x1B4
	Source end pointer	0x0B0	Source end pointer	0x1B0
Channel 10 to Channel 8	Reserved		Reserved	

Channel	Primary Structures		Alternate Structures	
Channel 7	Reserved; set to 0	0x07C	Reserved; set to 0	0x17C
	Control	0x078	Control	0x178
	Destination end pointer	0x074	Destination end pointer	0x174
	Source end pointer	0x070	Source end pointer	0x170
Channel 6	Reserved; set to 0	0x06C	Reserved; set to 0	0x16C
	Control	0x068	Control	0x168
	Destination end pointer	0x064	Destination end pointer	0x164
	Source end pointer	0x060	Source end pointer	0x160
:	…	…	…	…
Channel 1	Reserved; set to 0	0x01C	Reserved; set to 0	0x11C
	Control	0x018	Control	0x118
	Destination end pointer	0x014	Destination end pointer	0x114
	Source end pointer	0x010	Source end pointer	0x110
Channel 0	Reserved; set to 0	0x00C	Reserved; set to 0	0x10C
	Control	0x008	Control	0x108
	Destination end pointer	0x004	Destination end pointer	0x104
	Source end pointer	0x000	Source end pointer	0x100

Control Data Configuration

For each DMA transfer, the CHNL_CFG memory location provides the control information for the DMA transfer to the controller.

Table 240. Control Data Configuration

Bits	Name	Description
31 to 30	DST_INC	Destination address increment. The address increment depends on the source data width as follows: Source data width: byte 00: byte. 01: half word. 10: word. 11: no increment. Address remains set to the value that the DST_END_PTR memory location contains. Source data width: half word 00: Reserved. 01: half word. 10: word. 11: no increment. Address remains set to the value that the DST_END_PTR memory location contains. Source data width: word 00: Reserved. 01: Reserved. 10: word. 11: no increment. Address remains set to the value that the DST_END_PTR memory location contains.
29 to 28	Reserved	Undefined. Write as zero.

Bits	Name	Description
27 to 26	SRC_INC	Source address increment. The address increment depends on the source data width as follows: Source data width: byte 00: byte. 01: half word. 10: word. 11: no increment. Address remains set to the value that the SRC_END_PTR memory location contains. Source data width: half word 00: Reserved. 01: half word. 10: word. 11: no increment. Address remains set to the value that the SRC_END_PTR memory location contains. Source data width: word 00: Reserved. 01: Reserved. 10: word. 11: no increment. Address remains set to the value that the SRC_END_PTR memory location contains.
25 to 24	SRC_SIZE	Size of the source data. 00: byte 01: halfword 10: word 11: Reserved.
23 to 18	Reserved	Undefined. Write as zero.
17 to 14	R_Power	Set these bits to control how many DMA transfers can occur before the controller rearbitrates. Must be set to 0000 for all DMA transfers involving peripherals. Note that the operation of the DMA is indeterminate if a value other than 0000 is programmed in this location for DMA transfers involving peripherals.
13 to 4	N_minus_1	The number of configured transfers minus 1 for that channel. The 10-bit value indicates the number of DMA transfers (not the total number of bytes), minus one. The possible values are 0x000: 1 DMA transfer 0x001: 2 DMA transfers 0x002: 3 DMA transfers … 0x3FF: 1024 DMA transfers.
3	Reserved	Undefined. Write as zero.
2 to 0	Cycle_ctrl	The transfer types of the DMA cycle. 000: Stop (Invalid) 001: Basic 010: Autorequest 011: Ping-pong 100: Memory scatter-gather primary 101: Memory scatter-gather alternate 110: Peripheral scatter-gather primary 111: Peripheral scatter-gather alternate

During the DMA transfer process, but before arbitration, CHNL_CFG is written back to system memory with the N_MINUS_1 field changed to reflect the number of transfers yet to be completed.

At the end when the whole DMA cycle is complete, the CYCLE_CTRL bits are made invalid to indicate the completion of the transfer.

DMA Transfer Types (CHNL_CFG[2:0])

The DMA controller supports five types of DMA transfer. The various types are selected by programming the appropriate values into the cycle_ctrl bits (Bits[2:0]) in the CHNL_CFG location of the control data structure.

Invalid (CHNL_CFG[2:0] = 000)

This means no DMA transfer is enabled for the channel. After the controller completes a DMA cycle, it sets the cycle type to invalid, to prevent it from repeating the same DMA cycle.

Basic (CHNL_CFG[2:0] = 001)

In this mode, the controller can be configured to use either the primary, or alternate data structure. The peripheral must present a request for every data transfer. After the channel is enabled, when the controller receives a request, it performs the following operations:

1. The controller performs a transfer. If the number of transfers remaining is 0 the flow continues at Step 3.
2. The controller arbitrates
   a. If a higher-priority channel is requesting service then the controller services that channel.
   b. If the peripheral or software signals a request to the controller, then it continues at Step 1.
3. At the end of the transfer, the controller generates the corresponding DMA channel interrupt in the NVIC.

Autorequest (CHNL_CFG[2:0] = 010)

When the controller operates in this mode, it is only necessary for the controller to receive a single request to enable it to complete the entire DMA cycle. This allows a large data transfer to occur, without significantly increasing the latency for servicing higher priority requests, or requiring multiple requests from the processor or peripheral. This mode is very useful for a memory-to-memory copy application.

Autorequest is not suitable for peripheral use.

In this mode, the controller can be configured to use either the primary or alternate data structure. After the channel is enabled, when the controller receives a request, it performs the following operations:

1. The controller performs min(2^{R_power}, N) transfers for the channel, where R_POWER is Bits[17:14] of the control data configuration register and N is the number of transfers. If the number of transfers remaining is 0, the flow continues at Step 3.
2. A request for the channel is automatically generated. The controller arbitrates. If the channel has the highest priority, the DMA cycle continues at Step 1.
3. At the end of the transfer, the controller generates an interrupt for the corresponding DMA channel.

Ping-Pong (CHNL_CFG[2:0] = 011)

In ping-pong mode, the controller performs a DMA cycle using one of the data structures and then performs a DMA cycle using the other data structure. The controller continues to switch from primary to alternate to primary until it reads a data structure that is invalid, or until the host processor disables the channel.

This mode is useful for transferring data from peripheral to memory using different buffers in the memory. In a typical application, the host must configure both primary and alternate data structures before starting the transfer. As the transfer progresses, the host can subsequently configure primary or alternate control data structures in the interrupt service routine when the corresponding transfer ends.

The DMA controller interrupts the processor after the completion of transfers associated with each control data structure. The individual transfers using either the primary or alternate control data structure work exactly the same as a basic DMA transfer.

Memory Scatter-Gather (CHNL_CFG[2:0] = 100 or 101)

In memory scatter-gather mode, the controller must be configured to use both the primary and alternate data structures. The controller uses the primary data structure to program the control configuration for alternate data structure. The alternate data structure is used for actual data transfers, which are similar to an autorequest DMA transfer. The controller arbitrates after every primary transfer. The controller only needs one request to complete the entire transfer. This mode is used when performing multiple memory-to-memory copy tasks. The processor can configure all of the tasks simultaneously and does not need to intervene in between each task. The controller generates the corresponding DMA channel interrupt in the NVIC when the entire scatter-gather transaction completes using a basic cycle.

In this mode, the controller receives an initial request and then performs four DMA transfers using the primary data structure to program the control structure of the alternate data structure. After this transfer completes, the controller starts a DMA cycle using the alternate data structure. After the cycle completes, the controller performs another four DMA transfers using the primary data structure. The controller continues to switch from primary to alternate to primary, until the processor configures the alternate data structure for a basic cycle or the DMA reads an invalid data structure.

Table 241 lists the fields of the CHNL_CFG memory location for the primary data structure, which must be programmed with constant values for the memory scatter-gather mode.

Table 241. CHNL_CFG for Primary Data Structure in Memory Scatter-Gather Mode, CHNL_CFG[2:0] = 100

Bits	Name	Description
31 to 30	DST_INC	10: Configures the controller to use word increments for the address.
29 to 28	Reserved	Undefined. Write as 0.
27 to 26	SRC_INC	10: Configures the controller to use word increments for the address.
25 to 24	SRC_SIZE	10: Configures the controller to use word transfers.
23 to 18	Reserved	Undefined. Write as 0.
17 to 14	R_POWER	0010: Indicates that the DMA controller is to perform 4 transfers.
13 to 4	N_MINUS_1	Configures the controller to perform *N* DMA transfers, where *N* is a multiple of 4.
3	Reserved	Undefined. Write as 0.
2 to 0	CYCLE_CTRL	100: Configures the controller to perform a memory scatter-gather DMA cycle.

Peripheral Scatter-Gather (CHNL_CFG[2:0] = 110 or 111)

In peripheral scatter-gather mode, the controller must be configured to use both the primary and alternate data structure. The controller uses the primary data structure to program the control structure of the alternate data structure. The alternate data structure is used for actual data transfers and each transfer takes place using the alternate data structure with a basic DMA transfer. The controller does not arbitrate after every primary transfer. This mode is used when there are multiple peripheral-to-memory DMA tasks to be performed. The Cortex-M3 can configure all of the tasks simultaneously and does not need to intervene in between each task. This is very similar to memory scatter-gather mode except for arbitration and request requirements. The controller generates the corresponding DMA channel interrupt in the NVIC when the entire scatter-gather transaction completes using a basic cycle.

In peripheral scatter-gather mode, the controller receives an initial request from a peripheral and then performs four DMA transfers using the primary data structure to program the alternate control data structure. The controller then immediately starts a DMA cycle using the alternate data structure, without rearbitrating.

After this cycle completes, the controller rearbitrates and if it receives a request from the peripheral that has the highest priority, it performs another four DMA transfers using the primary data structure. It then immediately starts a DMA cycle using the alternate data structure without rearbitrating. The controller continues to switch from primary to alternate to primary until the processor configures the alternate data structure for a basic cycle or the DMA reads an invalid data structure.

Table 242 lists the fields of the CHNL_CFG memory location for the primary data structure, which must be programmed with constant values for the peripheral scatter-gather mode.

Table 242. CHNL_CFG For Primary Data Structure in Peripheral Scatter Gather Mode, CHNL_CFG[2:0] = 110

Bits	Name	Description
31 to 30	DST_INC	10: Configures the controller to use word increments for the address.
29 to 28	Reserved	Undefined. Write as 0.
27 to 26	SRC_INC	10: Configures the controller to use word increments for the address.
25 to 24	SRC_SIZE	10: Configures the controller to use word transfers.
23 to 18	Reserved	Undefined. Write as 0.
17 to 14	R_POWER	0010: Indicates that the DMA controller performed four transfers without rearbitration.
13 to 4	N_MINUS_1	Configures the controller to perform *N* DMA transfers, where *N* is a multiple of 4.
3	Reserved	Undefined. Write as 0.
2 to 0	CYCLE_CTRL	110: Configures the controller to perform a memory scatter-gather DMA cycle.

Address Calculation

The DMA controller calculates the source read address based on the content of SRC_END_PTR, the source address increment setting in CHNL_CFG, and the current value of the N_MINUS_1 (CHNL_CFG[13:4]).

Similarly the destination write address is calculated based on the content of DST_END_PTR, the destination address increment setting in CHNL_CFG, and the current value of the N_MINUS_1 (CHNL_CFG[13:4]).

Source Read Address = *SRC_END_PTR* − (*N_MINUS_1* << (*SRC_INC*)) for SRC_INC = 0, 1, 2

Source Read Address = *SRC_END_PTR* for SRC_INC = 3

Destination Write Address = *DST_END_PTR* − (*N_MINUS_1* << (*DST_INC*)) for DST_INC = 0, 1, 2

Destination Write Address = *DST_END_PTR* for *DST_INC* = 3

where *N_MINUS_1* is the number of configured transfers minus 1 for that channel.

REGISTER SUMMARY (DIRECT MEMORY ACCESS)

Table 243. Direct Memory Access Register Summary

Address	Name	Description	Reset	RW
0x40010000	DMASTA	Status Register	0x000D0000	R
0x40010004	DMACFG	Configuration Register	0x00000000	W
0x40010008	DMAPDBPTR	Primary Control Database Pointer Register	0x00000000	RW
0x4001000C	DMAADBPTR	Alternate Control Database Pointer Register	0x00000100	R
0x40010014	DMASWREQ	Channel Software Request Register	0x00000000	W
0x40010020	DMARMSKSET	Channel Request Mask Set Register	0x00000000	RW
0x40010024	DMARMSKCLR	Channel Request Mask Clear Register	0x00000000	W
0x40010028	DMAENSET	Channel Enable Set Register	0x00000000	RW
0x4001002C	DMAENCLR	Channel Enable Clear Register	0x00000000	W
0x40010030	DMAALTSET	Channel Primary-Alternate Set Register	0x00000000	RW
0x40010034	DMAALTCLR	Channel Primary-Alternate Clear Register	0x00000000	W
0x40010038	DMAPRISET	Channel Priority Set Register	0x00000000	RW
0x4001003C	DMAPRICLR	Channel Priority Clear Register	0x00000000	W
0x4001004C	DMAERRCLR	Bus Error Clear Register	0x00000000	RW

REGISTER DETAILS (DIRECT MEMORY ACCESS)

Status Register

Address: 0x40010000, Reset: 0x000D0000, Name: DMASTA

Returns the status of the controller when not in the reset state.

Table 244. Bit Descriptions for DMASTA

Bits	Bit Name	Description	Reset	Access
[31:21]	RESERVED	Reserved.	0x000	R
[20:16]	CHNLSMINUS1	Number of available DMA channels minus 1. For example, if there are 14 channels available, the register reads back 0xD for these bits. 01101: FOURTEENCHNLS–Controller configured to use 14 DMA channels.	0x0D	R
[15:8]	RESERVED	Reserved.	0x00	R
[7:4]	STATE	Current state of DMA controller state machine. Provides insight into the operation performed by the DMA at the time this register is read. 0000: IDL. Idle. 0001: RDCHNLDATA. Reading channel controller data. 0010: RDSRCENDPTR. Reading source data end pointer. 0011: RDDSTENDPTR. Reading destination end pointer. 0100: RDSRCDATA. Reading source data. 0101: WRDSTDATA. Writing destination data. 0110: WAITDMAREQCLR. Waiting for DMA request to clear. 0111: WRCHNLDATA. Writing channel controller data. 1000: STALLED. Stalled. 1001: DONE. Done. 1010: SCATRGATHR. Peripheral scatter-gather transition.	0x0	R
[3:1]	RESERVED	Reserved.	0x0	R
0	ENABLE	Enable status of the controller. 0: CLR. Controller is disabled. 1: SET. Controller is enabled.	0x0	R

Configuration Register

Address: 0x40010004, Reset: 0x00000000, Name: DMACFG

Table 245. Bit Descriptions for DMACFG

Bits	Bit Name	Description	Reset	Access
[31:1]	RESERVED	Reserved.	0x00000000	W
0	ENABLE	Controller enable. 0: DIS. Controller is disabled. 1: EN. Controller is enabled.	0x0	W

Primary Control Database Pointer Register

Address: 0x40010008, Reset: 0x00000000, Name: DMAPDBPTR

Table 246. Bit Descriptions for DMAPDBPTR

Bits	Bit Name	Description	Reset	Access
[31:0]	CTRLBASEPTR	Pointer to the base address of the primary data structure. 5 + log (2)M LSBs are reserved and must be written 0. M is the number of channels. Note that the DMAPDBPTR register must be programmed to point to the primary channel control base pointer in the system memory. The amount of system memory that must be assigned to the DMA controller depends on the number of DMA channels used and whether the alternate channel control data structure is used. This register cannot be read when the DMA controller is in the reset state.	0x00000000	R

Alternate Control Database Pointer Register

Address: 0x4001000C, Reset: 0x00000100, Name: DMAADBPTR

Table 247. Bit Descriptions for DMAADBPTR

Bits	Bit Name	Description	Reset	Access
[31:0]	ALTCBPTR	Base address of the alternate data structure. Notes: The DMAADBPTR read-only register returns the base address of the alternate channel control data structure. This register removes the necessity for application software to calculate the base address of the alternate data structure. This register cannot be read when the DMA controller is in the reset state.	0x00000100	RW

Channel Software Request Register

Address: 0x40010014, Reset: 0x00000000, Name: DMASWREQ

Used to generate a software request.

Set the appropriate bit to generate a software DMA request on the corresponding DMA channel. These bits are automatically cleared by the hardware after the corresponding software request completes.

Table 248. Bit Descriptions for DMASWREQ

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved.	0x00000	W
13	SPI0RX	DMA SPI0 Rx. 0: DIS. Does not create a DMA request for SPI0RX. 1: EN. Generates a DMA request for SPI0RX.	0x0	W
12	SPI0TX	DMA SPI0 Tx. 0: DIS. Does not create a DMA request for SPI0TX. 1: EN. Generates a DMA request for SPI0TX.	0x0	W
11	ADC	DMA ADC. 0: DIS. Does not create a DMA request for ADC. 1: EN. Generates a DMA request for ADC.	0x0	W
[10:8]	RESERVED	Reserved.	0x0	W
7	I2CMRX	DMA I^{2}C Master Rx. 0: DIS. Does not create a DMA request for I2CMRX. 1: EN. Generates a DMA request for I2CMRX.	0x0	W
6	I2CMTX	DMA I2C Master Tx. 0: DIS. Does not create a DMA request for I2CMTX. 1: EN. Generates a DMA request for I2CMTX.	0x0	W
5	I2CSRX	DMA I2C Slave Rx. 0: DIS. Does not create a DMA request for I2CSRX. 1: EN. Generates a DMA request for I2CSRX.	0x0	W
4	I2CSTX	DMA I2C Slave Tx. 0: DIS. Does not create a DMA request for I2CSTX. 1: EN. Generates a DMA request for I2CSTX.	0x0	W
3	UARTRX	DMA UART Rx. 0: DIS. Does not create a DMA request for UARTRX. 1: EN. Generates a DMA request for UARTRX.	0x0	W
2	UARTTX	DMA UART Tx. 0: DIS. Does not create a DMA request for UARTTX. 1: EN. Generates a DMA request for UARTTX.	0x0	W
1	SPI1RX	DMA SPI 1 Rx. 0: DIS. Does not create a DMA request for SPI1RX. 1: EN. Generates a DMA request for SPI1RX.	0x0	W
0	SPI1TX	DMA SPI 1 Tx. 0: DIS. Does not create a DMA request for SPI1TX. 1: EN. Generates a DMA request for SPI1TX.	0x0	W

Channel Request Mask Set Register

Address: 0x40010020, Reset: 0x00000000, Name: DMARMSKSET

This register disables DMA requests from peripherals. Each bit of the register represents the corresponding channel number in the DMA controller.

Set the appropriate bit to mask the request from the corresponding DMA channel.

Table 249. Bit Descriptions for DMARMSKSET

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved.	0x00000	R
13	SPI0RX	DMA SPI0 Rx. 0: DIS. When read: requests are enabled for SPI0RX. When written: no effect. Use the DMARMSKCLR register to enable DMA requests. 1: EN. When read: requests are disabled for SPI0RX. When written: disables peripheral associated with SPI0RX from generating DMA requests.	0x0	RW
12	SPI0TX	DMA SPI0 Tx. 0: DIS. When read: requests are enabled for SPI0TX. When written: no effect. Use the DMARMSKCLR register to enable DMA requests. 1: EN. When read: requests are disabled for SPI0TX. When written: disables peripheral associated with SPI0TX from generating DMA requests.	0x0	RW
11	ADC	DMA ADC. 0: DIS. When read: requests are enabled for ADC. When written: no effect. Use the DMARMSKCLR register to enable DMA requests. 1: EN. When read: requests are disabled for ADC. When written: disables peripheral associated with ADC from generating DMA requests.	0x0	RW
[10:8]	RESERVED	Reserved.	0x0	R
7	I2CMRX	DMA I^{2}C Master Rx. 0: DIS. When read: requests are enabled for I2CMRX. When written: no effect. Use the DMARMSKCLR register to enable DMA requests. 1: EN. When read: requests are disabled for I2CMRX. When written: disables peripheral associated with I2CMRX from generating DMA requests.	0x0	RW
6	I2CMTX	DMA I^{2}C Master Tx. 0: DIS. When read: requests are enabled for I2CMTX. When written: no effect. Use the DMARMSKCLR register to enable DMA requests. 1: EN. When read: requests are disabled for I2CMTX. When written: disables peripheral associated with I2CMTX from generating DMA requests.	0x0	RW
5	I2CSRX	DMA I^{2}C Slave Rx. 0: DIS. When read: requests are enabled for I2CSRX. When written: no effect. Use the DMARMSKCLR register to enable DMA requests. 1: EN. When read: requests are disabled for I2CSRX. When written: disables peripheral associated with I2CSRX from generating DMA requests.	0x0	RW
4	I2CSTX	DMA I^{2}C Slave Tx. 0: DIS. When read: requests are enabled for I2CSTX. When written: no effect. Use the DMARMSKCLR register to enable DMA requests. 1: EN. When read: requests are disabled for I2CSTX. When written: disables peripheral associated with I2CSTX from generating DMA requests.	0x0	RW
3	UARTRX	DMA UART Rx. 0: DIS. When read: requests are enabled for UARTRX. When written: no effect. Use the DMARMSKCLR register to enable DMA requests. 1: EN. When read: requests are disabled for UARTRX. When written: disables peripheral associated with UARTRX from generating DMA requests.	0x0	RW
2	UARTTX	DMA UART Tx. 0: DIS. When read: requests are enabled for UARTTX. When written: no effect. Use the DMARMSKCLR register to enable DMA requests. 1: EN. When read: Requests are disabled for UARTTX. When written: disables peripheral associated with UARTTX from generating DMA requests.	0x0	RW

Bits	Bit Name	Description	Reset	Access
1	SPI1RX	DMA SPI 1 Rx. 0: DIS. When read: requests are enabled for SPI1RX. When written: no effect. Use the DMARMSKCLR register to enable DMA requests. 1: EN. When read: requests are disabled for SPI1RX. When written: disables peripheral associated with SPI1RX from generating DMA requests.	0x0	RW
0	SPI1TX	DMA SPI 1 Tx. 0: DIS. When read: requests are enabled for SPI1TX. When written: no effect. Use the DMARMSKCLR register to enable DMA requests. 1: EN. When read: requests are disabled for SPI1TX When written: disables peripheral associated with SPI1TX from generating DMA requests.	0x0	RW

Channel Request Mask Clear Register

Address: 0x40010024, Reset: 0x00000000, Name: DMARMSKCLR

This register enables DMA requests from peripherals by clearing the mask set in the DMARMSKSET register. Each bit of the register represents the corresponding channel number in the DMA controller.

Set the appropriate bit to clear the corresponding bits in the DMARMSKSET register.

Table 250. Bit Descriptions for DMARMSKCLR

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved.	0x00000	W
13	SPI0RX	DMA SPI0 Rx. 0: DIS. No effect. Use the DMARMSKSET register to disable DMA requests. 1: EN. Enables peripheral associated with SPI0RX to generate DMA requests.	0x0	W
12	SPI0TX	DMA SPI0 Tx. 0: DIS. No effect. Use the DMARMSKSET register to disable DMA requests. 1: EN. Enables peripheral associated with SPI0TX to generate DMA requests.	0x0	W
11	ADC	DMA ADC. 0: DIS. No effect. Use the DMARMSKSET register to disable DMA requests. 1: EN. Enables peripheral associated with ADC to generate DMA requests.	0x0	W
[10:8]	RESERVED	Reserved.	0x0	W
7	I2CMRX	DMA I^{2}C Master Rx. 0: DIS. No effect. Use the DMARMSKSET register to disable DMA requests. 1: EN. Enables peripheral associated with I2CMRX to generate DMA requests.	0x0	W
6	I2CMTX	DMA I^{2}C Master Tx. 0: DIS. No effect. Use the DMARMSKSET register to disable DMA requests. 1: EN. Enables peripheral associated with I2CMTX to generate DMA requests.	0x0	W
5	I2CSRX	DMA I^{2}C Slave Rx. 0: DIS. No effect. Use the DMARMSKSET register to disable DMA requests. 1: EN. Enables peripheral associated with I2CSRX to generate DMA requests.	0x0	W
4	I2CSTX	DMA I^{2}C Slave Tx. 0: DIS. No effect. Use the DMARMSKSET register to disable DMA requests. 1: EN. Enables peripheral associated with I2CSTX to generate DMA requests.	0x0	W
3	UARTRX	DMA UART Rx. 0: DIS. No effect. Use the DMARMSKSET register to disable DMA requests. 1: EN. Enables peripheral associated with UARTRX to generate DMA requests.	0x0	W

Bits	Bit Name	Description	Reset	Access
2	UARTTX	DMA UART Tx. 0: DIS. No effect. Use the DMARMSKSET register to disable DMA requests. 1: EN. Enables peripheral associated with UARTTX to generate DMA requests.	0x0	W
1	SPI1RX	DMA SPI 1 Rx. 0: DIS. No effect. Use the DMARMSKSET register to disable DMA requests. 1: EN. Enables peripheral associated with SPI1RX to generate DMA requests.	0x0	W
0	SPI1TX	DMA SPI 1 Tx. 0: DIS. No effect. Use the DMARMSKSET register to disable DMA requests. 1: EN. Enables peripheral associated with SPI1TX to generate DMA requests.	0x0	W

Channel Enable Set Register

Address: 0x40010028, Reset: 0x00000000, Name: DMAENSET

Enable DMA channels.

This register allows for the enabling of DMA channels. Reading the register returns the enable status of the channels. Each bit of the register represents the corresponding channel number in the DMA controller. Set the appropriate bit to enable the corresponding channel.

Table 251. Bit Descriptions for DMAENSET

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved.	0x00000	R
13	SPI0RX	DMA SPI0 Rx. 0: DIS. No effect. Use the DMAENCLR register to disable the channel. 1: EN. Enables SPI0RX.	0x0	RW
12	SPI0TX	DMA SPI0 Tx. 0: DIS. No effect. Use the DMAENCLR register to disable the channel. 1: EN. Enables SPI0TX.	0x0	RW
11	ADC	DMA ADC. 0: DIS. No effect. Use the DMAENCLR register to disable the channel. 1: EN. Enables ADC.	0x0	RW
[10:8]	RESERVED	Reserved.	0x0	R
7	I2CMRX	DMA I^{2}C Master Rx. 0: DIS. No effect. Use the DMAENCLR register to disable the channel. 1: EN. Enables I2CMRX.	0x0	RW
6	I2CMTX	DMA I^{2}C Master Tx. 0: DIS. No effect. Use the DMAENCLR register to disable the channel. 1: EN. Enables I2CMTX.	0x0	RW
5	I2CSRX	DMA I^{2}C Slave Rx. 0: DIS. No effect. Use the DMAENCLR register to disable the channel. 1: EN. Enables I2CSRX.	0x0	RW
4	I2CSTX	DMA I^{2}C Slave Tx. 0: DIS. No effect. Use the DMAENCLR register to disable the channel. 1: EN. Enables I2CSTX.	0x0	RW
3	UARTRX	DMA UART Rx. 0: DIS. No effect. Use the DMAENCLR register to disable the channel. 1: EN. Enables UARTRX.	0x0	RW
2	UARTTX	DMA UART Tx. 0: DIS. No effect. Use the DMAENCLR register to disable the channel. 1: EN. Enables UARTTX.	0x0	RW

Bits	Bit Name	Description	Reset	Access
1	SPI1RX	DMA SPI 1 Rx. 0: DIS. No effect. Use the DMAENCLR register to disable the channel. 1: EN. Enables SPI1RX.	0x0	RW
0	SPI1TX	DMA SPI 1 Tx. 0: DIS. No effect. Use the DMAENCLR register to disable the channel. 1: EN. Enables SPI1TX.	0x0	RW

Channel Enable Clear Register

Address: 0x4001002C, Reset: 0x00000000, Name: DMAENCLR

Disable DMA channels.

This register allows for the disabling of DMA channels. Each bit of the register represents the corresponding channel number in the DMA controller.

Note that the controller disables a channel automatically by setting the appropriate bit when it completes the DMA cycle. Set the appropriate bit to disable the corresponding channel.

Table 252. Bit Descriptions for DMAENCLR

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved.	0x00000	W
13	SPI0RX	DMA SPI0 Rx. 0: DIS. No effect. Use the DMAENSET register to enable the channel. 1: EN. Disables SPI0RX.	0x0	W
12	SPI0TX	DMA SPI0 Tx. 0: DIS. No effect. Use the DMAENSET register to enable the channel. 1: EN. Disables SPI0TX.	0x0	W
11	ADC	DMA ADC. 0: DIS. No effect. Use the DMAENSET register to enable the channel. 1: EN. Disables ADC.	0x0	W
[10:8]	RESERVED	Reserved.	0x0	W
7	I2CMRX	DMA I^{2}C Master Rx. 0: DIS. No effect. Use the DMAENSET register to enable the channel. 1: EN. Disables I2CMRX.	0x0	W
6	I2CMTX	DMA I^{2}C Master Tx. 0: DIS. No effect. Use the DMAENSET register to enable the channel. 1: EN. Disables I2CMTX.	0x0	W
5	I2CSRX	DMA I^{2}C Slave Rx. 0: DIS. No effect. Use the DMAENSET register to enable the channel. 1: EN. Disables I2CSRX.	0x0	W
4	I2CSTX	DMA I^{2}C Slave Tx. 0: DIS. No effect. Use the DMAENSET register to enable the channel. 1: EN. Disables I2CSTX.	0x0	W
3	UARTRX	DMA UART Rx. 0: DIS. No effect. Use the DMAENSET register to enable the channel. 1: EN. Disables UARTRX.	0x0	W
2	UARTTX	DMA UART Tx. 0: DIS. No effect. Use the DMAENSET register to enable the channel. 1: EN. Disables UARTTX.	0x0	W
1	SPI1RX	DMA SPI 1 Rx. 0: DIS. No effect. Use the DMAENSET register to enable the channel. 1: EN. Disables SPI1RX.	0x0	W

Bits	Bit Name	Description	Reset	Access
0	SPI1TX	DMA SPI 1 Tx. 0: DIS. No effect. Use the DMAENSET register to enable the channel. 1: EN. Disables SPI1TX.	0x0	W

Channel Primary-Alternate Set Register

Address: 0x40010030, Reset: 0x00000000, Name: DMAALTSET

Control structure status/select alternate structure.

Returns the channel control data structure status, or selects the alternate data structure for the corresponding DMA channel.

Note that the DMA controller sets/clears these bits automatically as necessary for ping-pong, memory scatter-gather, and peripheral scatter-gather transfers.

Table 253. Bit Descriptions for DMAALTSET

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved.	0x00000	R
13	SPI0RX	DMA SPI0 Rx. 0: DIS. When read: DMA SPI0RX is using the primary data structure. When written: no effect. Use the DMAALTCLR register to set SPI0RX to 0. 1: EN. When read: DMA SPI0RX is using the alternate data structure. When written: selects the alternate data structure for SPI0RX.	0x0	RW
12	SPI0TX	DMA SPI0 Tx. 0: DIS. When read: DMA SPI0TX is using the primary data structure. When written: no effect. Use the DMAALTCLR register to set SPI0TX to 0. 1: EN. When read: DMA SPI0TX is using the alternate data structure. When written: selects the alternate data structure for SPI0TX.	0x0	RW
11	ADC	DMA ADC. 0: DIS. When read: DMA ADC is using the primary data structure. When written: no effect. Use the DMAALTCLR register to set ADC to 0. 1: EN. When read: DMA ADC is using the alternate data structure. When written: selects the alternate data structure for ADC.	0x0	RW
[10:8]	RESERVED	Reserved.	0x0	R
7	I2CMRX	DMA I^{2}C Master Rx. 0: DIS. When read: DMA I2CMRX is using the primary data structure. When written: No effect. Use the DMAALTCLR register to set I2CMRX to 0. 1: EN. When read: DMA I2CMRX is using the alternate data structure. When written: Selects the alternate data structure for I2CMRX.	0x0	RW
6	I2CMTX	DMA I^{2}C Master Tx. 0: DIS. When read: DMA I2CMTX is using the primary data structure. When written: no effect. Use the DMAALTCLR register to set I2CMTX to 0. 1: EN. When read: DMA I2CMTX is using the alternate data structure. When written: selects the alternate data structure forI2CMTX.	0x0	RW
5	I2CSRX	DMA I^{2}C Slave Rx. 0: DIS. When read: DMA I2CSRX is using the primary data structure. When written: no effect. Use the DMAALTCLR register to set I2CSRX to 0. 1: EN. When read: DMA I2CSRX is using the alternate data structure. When written: selects the alternate data structure for I2CSRX.	0x0	RW
4	I2CSTX	DMA I^{2}C Slave Tx. 0: DIS. When read: DMA I2CSTX is using the primary data structure. When written: no effect. Use the DMAALTCLR register to set I2CSTX to 0. 1: EN. When read: DMA I2CSTX is using the alternate data structure. When written: selects the alternate data structure for I2CSTX.	0x0	RW

Bits	Bit Name	Description	Reset	Access
3	UARTRX	DMA UART Rx. 0: DIS. When read: DMA UARTRX is using the primary data structure. When written: no effect. Use the DMAALTCLR register to set UARTRX to 0. 1: EN. When read: DMA UARTRX is using the alternate data structure. When written: selects the alternate data structure for UARTRX.	0x0	RW
2	UARTTX	DMA UART Tx. 0: DIS. When read: DMA UARTTX is using the primary data structure. When written: no effect. Use the DMAALTCLR register to set UARTTX to 0. 1: EN. When read: DMA UARTTX is using the alternate data structure. When written: selects the alternate data structure for UARTTX.	0x0	RW
1	SPI1RX	DMA SPI 1 Rx. 0: DIS. When read: DMA SPI1RX is using the primary data structure. When written: no effect. Use the DMAALTCLR register to set SPI1RX to 0. 1: EN. When read: DMA SPI1RX is using the alternate data structure. When written: selects the alternate data structure for SPI1RX.	0x0	RW
0	SPI1TX	DMA SPI 1 Tx. 0: DIS. When read: DMA SPI1TX is using the primary data structure. When written: no effect. Use the DMAALTCLR register to set SPI1TX to 0. 1: EN. When read: DMA SPI1TX is using the alternate data structure. When written: selects the alternate data structure for SPI1TX.	0x0	RW

Channel Primary-Alternate Clear Register

Address: 0x40010034, Reset: 0x00000000, Name: DMAALTCLR

Select primary data structure.

Set the appropriate bit to select the primary data structure for the corresponding DMA channel.

Note that the DMA controller sets/clears these bits automatically as necessary for ping-pong, memory scatter-gather, and peripheral scatter-gather transfers.

Table 254. Bit Descriptions for DMAALTCLR

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved.	0x00000	W
13	SPI0RX	DMA SPI0 Rx. 0: DIS. No effect. Use the DMAALTSET register to select the alternate data structure. 1: EN. Selects the primary data structure for SPI0RX.	0x0	W
12	SPI0TX	DMA SPI0 Tx. 0: DIS. No effect. Use the DMAALTSET register to select the alternate data structure. 1: EN. Selects the primary data structure for SPI0TX.	0x0	W
11	ADC	DMA ADC. 0: DIS. No effect. Use the DMAALTSET register to select the alternate data structure. 1: EN. Selects the primary data structure for ADC.	0x0	W
[10:8]	RESERVED	Reserved.	0x0	W
7	I2CMRX	DMA I^{2}C Master Rx. 0: DIS. No effect. Use the DMAALTSET register to select the alternate data structure. 1: EN. Selects the primary data structure for I2CMRX.	0x0	W
6	I2CMTX	DMA I^{2}C Master Tx. 0: DIS. No effect. Use the DMAALTSET register to select the alternate data structure. 1: EN. Selects the primary data structure for I2CMTX.	0x0	W

Bits	Bit Name	Description	Reset	Access
5	I2CSRX	DMA I^{2}C Slave Rx. 0: DIS. No effect. Use the DMAALTSET register to select the alternate data structure. 1: EN. Selects the primary data structure for I2CSRX.	0x0	W
4	I2CSTX	DMA I^{2}C Slave Tx. 0: DIS. No effect. Use the DMAALTSET register to select the alternate data structure. 1: EN. Selects the primary data structure for I2CSTX.	0x0	W
3	UARTRX	DMA UART Rx. 0: DIS. No effect. Use the DMAALTSET register to select the alternate data structure. 1: EN. Selects the primary data structure for UARTRX.	0x0	W
2	UARTTX	DMA UART Tx. 0: DIS. No effect. Use the DMAALTSET register to select the alternate data structure. 1: EN. Selects the primary data structure for UARTTX.	0x0	W
1	SPI1RX	DMA SPI 1 Rx. 0: DIS. No effect. Use the DMAALTSET register to select the alternate data structure. 1: EN. Selects the primary data structure for SPI1RX.	0x0	W
0	SPI1TX	DMA SPI 1 Tx. 0: DIS. No effect. Use the DMAALTSET register to select the alternate data structure. 1: EN. Selects the primary data structure for SPI1TX.	0x0	W

Channel Priority Set Register

Address: 0x40010038, Reset: 0x00000000, Name: DMAPRISET

Configure channel for high priority.

This register enables the user to configure a DMA channel to use the high priority level.

Reading the register returns the status of the channel priority mask. Each bit of the register represents the corresponding channel number in the DMA controller.

Returns the channel priority mask status, or sets the channel priority to high.

Table 255. Bit Descriptions for DMAPRISET

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved.	0x00000	R
13	SPI0RX	DMA SPI0 Rx. 0: DIS. When read: DMA SPI0RX is using the default priority level. When written: no effect. Use the DMAPRICLR register to set SPI0RX to the default priority level. 1: EN. When read: DMA SPI0RX is using a high priority level. When written: SPI0RX uses the high priority level.	0x0	RW
12	SPI0TX	DMA SPI0 Tx. 0: DIS. When read: DMA SPI0TX is using the default priority level. When written: no effect. Use the DMAPRICLR register to set SPI0TX to the default priority level. 1: EN. When read: DMA SPI0TX is using a high priority level. When written: SPI0TX uses the high priority level.	0x0	RW
11	ADC	DMA ADC. 0: DIS. When read: DMA ADC is using the default priority level. When written: no effect. Use the DMAPRICLR register to set ADC to the default priority level. 1: EN. When read: DMA ADCs using a high priority level. When written: ADC uses the high priority level.	0x0	RW
[10:8]	RESERVED	Reserved.	0x0	R

Bits	Bit Name	Description	Reset	Access
7	I2CMRX	DMA I^{2}C Master Rx. 0: DIS. When read: DMA I2CMRX is using the default priority level. When written: no effect. Use the DMAPRICLR register to set I2CMRX to the default priority level. 1: EN. When read: DMA I2CMRX is using a high priority level. When written: I2CMRX uses the high priority level.	0x0	RW
6	I2CMTX	DMA I^{2}C Master Tx. 0: DIS. When read: DMA I2CMTX is using the default priority level. When written: no effect. Use the DMAPRICLR register to set I2CMTX to the default priority level. 1: EN. When read: DMA I2CMTX is using a high priority level. When written: I2CMTX uses the high priority level.	0x0	RW
5	I2CSRX	DMA I^{2}C Slave Rx. 0: DIS. When read: DMA I2CSRX is using the default priority level. When written: no effect. Use the DMAPRICLR register to set I2CSRX to the default priority level. 1: EN. When read: DMA I2CSRX is using a high priority level. When written: I2CSRX uses the high priority level.	0x0	RW
4	I2CSTX	DMA I^{2}C Slave Tx. 0: DIS. When read: DMA I2CSTX is using the default priority level. When written: no effect. Use the DMAPRICLR register to set I2CSTX to the default priority level. 1: EN. When read: DMA I2CSTX is using a high priority level. When written: I2CSTX uses the high priority level.	0x0	RW
3	UARTRX	DMA UART Rx. 0: DIS. When read: DMA UARTRX is using the default priority level. When written: No effect. Use the DMAPRICLR register to set UARTRX to the default priority level. 1: EN. When read: DMA UARTRX is using a high priority level. When written: UARTRX uses the high priority level.	0x0	RW
2	UARTTX	DMA UART Tx. 0: DIS. When read: DMA UARTTX is using the default priority level. When written: no effect. Use the DMAPRICLR register to set UARTTX to the default priority level. 1: EN. When read: DMA UARTTX is using a high priority level. When written: UARTTX uses the high priority level.	0x0	RW
1	SPI1RX	DMA SPI 1 Rx. 0: DIS. When read: DMA SPI1RX is using the default priority level. When written: no effect. Use the DMAPRICLR register to set SPI1RX to the default priority level. 1: EN. When read: DMA SPI1RX is using a high priority level. When written: SPI1RX uses the high priority level.	0x0	RW
0	SPI1TX	DMA SPI 1 Tx. 0: DIS. When read: DMA SPI1TX is using the default priority level. When written: No effect. Use the DMAPRICLR register to set SPI1TX to the default priority level. 1: EN. When read: DMA SPI1TX is using a high priority level. When written: SPI1TX uses the high priority level.	0x0	RW

Channel Priority Clear Register

Address: 0x4001003C, Reset: 0x00000000, Name: DMAPRICLR

Configure channel for default priority level.

The DMAPRICLR write-only register enables the user to configure a DMA channel to use the default priority level. Each bit of the register represents the corresponding channel number in the DMA controller. Set the appropriate bit to select the default priority level for the specified DMA channel.

Table 256. Bit Descriptions for DMAPRICLR

Bits	Bit Name	Description	Reset	Access
[31:14]	RESERVED	Reserved.	0x00000	W
13	SPI0RX	DMA SPI0 Rx. 0: DIS. No effect. Use the DMAPRISET register to set SPI0RX to the high priority level. 1: EN. SPI0RX uses the default priority level.	0x0	W
12	SPI0TX	DMA SPI0 Tx. 0: DIS. No effect. Use the DMAPRISET register to set SPI0TX to the high priority level. 1: EN. SPI0TX uses the default priority level.	0x0	W
11	ADC	DMA ADC. 0: DIS. No effect. Use the DMAPRISET register to set ADC to the high priority level. 1: EN. ADC uses the default priority level.	0x0	W
[10:8]	RESERVED	Reserved.	0x0	W
7	I2CMRX	DMA I^{2}C Master Rx. 0: DIS. No effect. Use the DMAPRISET register to set I2CMRX to the high priority level. 1: EN. I2CMRX uses the default priority level.	0x0	W
6	I2CMTX	DMA I^{2}C Master Tx. 0: DIS. No effect. Use the DMAPRISET register to set I2CMTX to the high priority level. 1: EN. I2CMTX uses the default priority level.	0x0	W
5	I2CSRX	DMA I^{2}C Slave Rx. 0: DIS. No effect. Use the DMAPRISET register to set I2CSRX to the high priority level. 1: EN. I2CSRX uses the default priority level.	0x0	W
4	I2CSTX	DMA I^{2}C Slave Tx. 0: DIS. No effect. Use the DMAPRISET register to set I2CSTX to the high priority level. 1: EN. I2CSTX uses the default priority level.	0x0	W
3	UARTRX	DMA UART Rx. 0: DIS. No effect. Use the DMAPRISET register to set UARTRX to the high priority level. 1: EN. UARTRX uses the default priority level.	0x0	W
2	UARTTX	DMA UART Tx. 0: DIS. No effect. Use the DMAPRISET register to set UARTTX to the high priority level. 1: EN. UARTTX uses the default priority level.	0x0	W
1	SPI1RX	DMA SPI 1 Rx. 0: DIS. No effect. Use the DMAPRISET register to set SPI1RX to the high priority level. 1: EN. SPI1RX uses the default priority level.	0x0	W
0	SPI1TX	DMA SPI 1 Tx. 0: DIS. No effect. Use the DMAPRISET register to set SPI1TX to the high priority level. 1: EN. SPI1TX uses the default priority level.	0x0	W

Bus Error Clear Register

Address: 0x4001004C, Reset: 0x00000000, Name: DMAERRCLR

Table 257. Bit Descriptions for DMAERRCLR

Bits	Bit Name	Description	Reset	Access
[31:1]	RESERVED	Reserved.	0x00000000	R
0	ERROR	DMA Bus Error status. Note that this register is used to read and clear the DMA bus error status. The error status is set if the controller encounters a bus error while performing a transfer. If a bus error occurs on a channel, that channel is automatically disabled by the controller. The other channels are unaffected. Write 1 to clear bits. 0: DIS. When read: no bus error occurred. When Written: no effect. 1: EN. When read: a bus error is pending. When Written: bit is cleared.	0x0	RW

PWM

PWM FEATURES

- 4 pairs of PWM outputs individually controlled
- H-Bridge mode supported on 2 pairs

PWM OVERVIEW

The ADuCRF101 integrates an 8-channel PWM interface. The eight outputs are grouped as four pairs (0 to 3). Pair 0 and pair 1 can be configured in standard mode or to drive an H-bridge. Pair 2 and pair 3 can be configured in standard mode only. The PWM pairs and modes are summarized in Table 258.

Table 258. PWM Channel Grouping

Pair	Port Name	Description	PWM Mode Available
0	PWM0	High-side PWM output	H-Bridge and standard
	PWM1	Low-side PWM output	H-Bridge and standard
1	PWM2	High-side PWM output	H-Bridge and standard
	PWM3	Low-side PWM output	H-Bridge and standard
2	PWM4	High-side PWM output	Standard
	PWM5	Low-side PWM output	Standard
3	PWM6	High-side PWM output	Standard
	PWM7	Low-side PWM output	Standard

On power-up, the PWM outputs default to H-bridge mode for pair 0 and pair 1.

In all modes, users have control over the period of each pair of outputs and over the duty cycle of each active individual outputs.

In the event of external fault conditions, a falling edge on the $\mathrm{PWM}_{\mathrm{TRIP}}$ pin provides an instantaneous shutdown of the PWM controller. All PWM outputs are placed in the off state, that is, in low state for the low side and high state for the high side, and a PWMTRIP interrupt can be generated.

PWM OPERATION

The PWM peripheral clock is selectable via PWMCON0 with one of the following values: UCLK divided by 2, 4, 8, 16, 32, 64, 128, or 256.

In all modes, the PWMxCOMx MMRs controls the point at which the PWM output changes state. An example is shown in Figure 58. Each pair has an associated counter. The length of the PWM period is defined by PWMxLEN.

The PWM waveforms are set by the count value of the 16-bit timer and the compare register contents.

An example for PWM Pair 0 (Ports PWM0 and PWM1) is

- The low-side waveform, PWM1, goes high when the timer count reaches PWM0LEN, and it goes low when the timer count reaches the value held in PWM0COM2 or when the high-side waveform PWM0 goes low.
- The high-side waveform, PWM0, goes high when the timer count reaches the value held in PWM0COM0, and it goes low when the timer count reaches the value held in PWM0COM1.

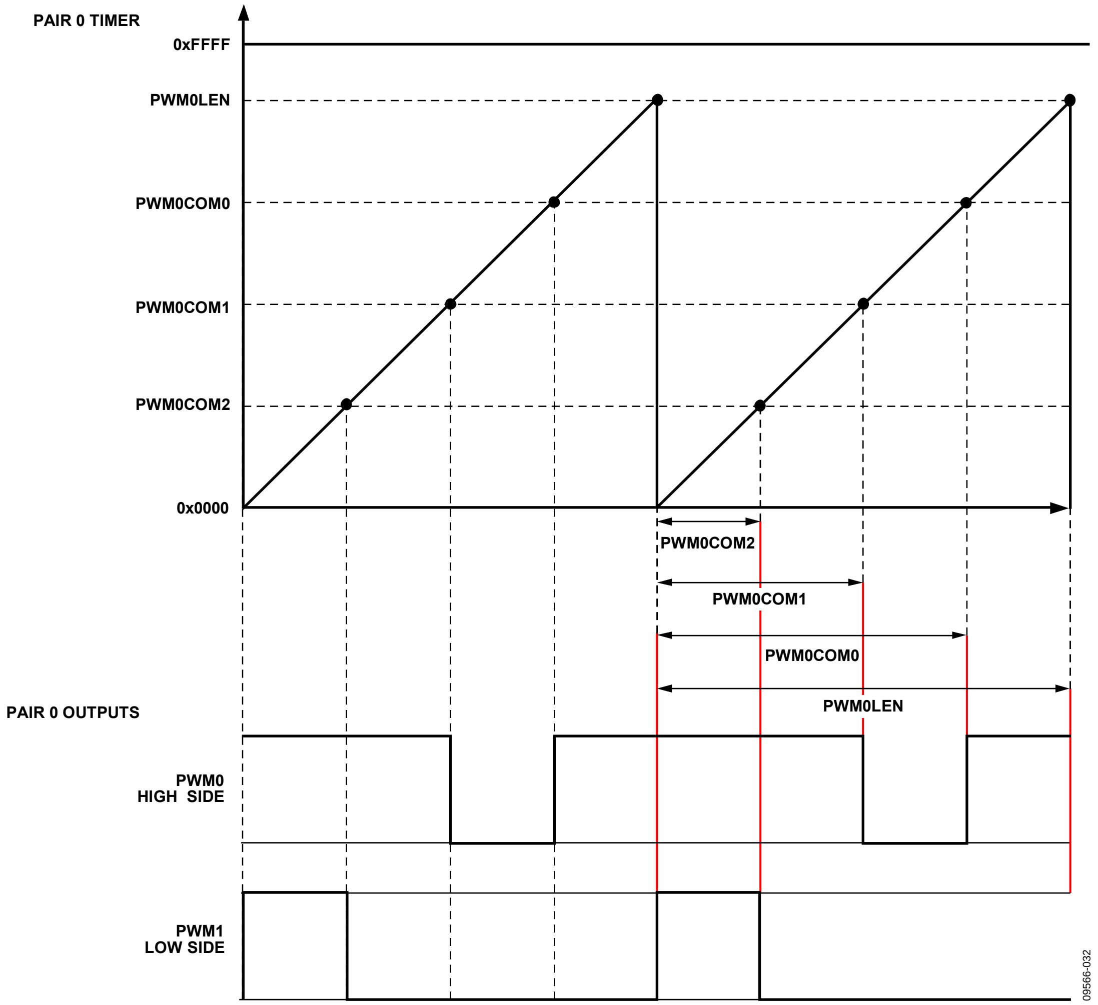


Figure 58. Waveform of PWM Channel Pair in Standard Mode

Note that the high-side PWM output for each channel must have a high duration period greater than or equal to the high period duration of the low-side output. For example, the high period for PWM0 must be equal to or, greater than the high period of PWM1.

Table 259 lists equations for the period and duration of the low side and high side of a pair.

Table 259. PWM Equations Example for Pair 0

Pair 0	Period	Duration
Low Side (PWM1)	t_{UCLK} × (PWM0LEN + 1) × $N_{PRESCALE}$	High duration If (PWM0COM2 < PWM0COM1), then t_{UCLK} × (PWM0LEN-PWM0COM2) × $N_{PRESCALE}$ Else t_{UCLK} × (PWM0LEN-PWM0COM1) × $N_{PRESCALE}$
High Side (PWM0)	t_{UCLK} × (PWM0LEN + 1) × $N_{PRESCALE}$	Low duration t_{UCLK} × (PWM0COM0 – PWM0COM1) × $N_{PRESCALE}$

Notes:

- t_{UCLK} = Period of UCLK clock input
- $N_{PRESCALE}$ = Prescalar value as determined by PWMCON0[8:6]

Standard Mode

In standard mode each pair is controlled by a set of registers as summarized in Table 260.

Table 260: Compare Register Descriptions

Pair	Name	Description
0	PWM0COM0	PWM0 output goes high when the PWM timer reaches the count value stored in this register
	PWM0COM1	PWM0 output goes low when the PWM timer reaches the count value stored in this register
	PWM0COM2	PWM1 output goes low when the PWM timer reaches the count value stored in this register
	PWM0LEN	PWM1 output goes high when the PWM timer reaches the count value stored in this register
1	PWM1COM0	PWM2 output goes high when the PWM timer reaches the count value stored in this register
	PWM1COM1	PWM2 output goes low when the PWM timer reaches the count value stored in this register
	PWM1COM2	PWM3 output goes low when the PWM timer reaches the count value stored in this register
	PWM1LEN	PWM3 output goes high when the PWM timer reaches the count value stored in this register
2	PWM2COM0	PWM4 output goes high when the PWM timer reaches the count value stored in this register
	PWM2COM1	PWM4 output goes low when the PWM timer reaches the count value stored in this register
	PWM2COM2	PWM5 output goes low when the PWM timer reaches the count value stored in this register
	PWM2LEN	PWM5 output goes high when the PWM timer reaches the count value stored in this register
3	PWM3COM0	PWM6 output goes high when the PWM timer reaches the count value stored in this register
	PWM3COM1	PWM6 output goes low when the PWM timer reaches the count value stored in this register
	PWM3COM2	PWM7 output goes low when the PWM timer reaches the count value stored in this register
	PWM3LEN	PWM7 output goes high when the PWM timer reaches the count value stored in this register

H-Bridge Mode

In H-bridge mode, the period and duty cycle of the 4 outputs are controlled using pair 0 registers: PWM0COM0, PWM0COM1, PWM0COM2 and PWM0LEN. In addition the state of the output is controlled by PWMCON0 Bit 9, Bit 5, Bit 4, and Bit 2 as summarized in Table 261.

An example of a motor in H-bridge configuration is shown in Figure 59. Note that only PWM0 to PWM3 participate in H-bridge mode; other outputs (PWM4 to PWM7) do not and continue to generate standard mode output.

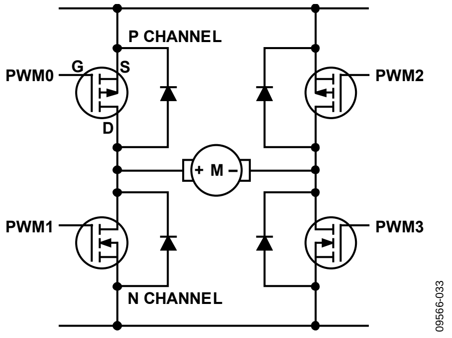


Figure 59. Example H-Bridge Configuration

Table 261. PWM Output in H-Bridge Mode

PWMCON Bits				PWM Outputs				
ENA Bit 9	POINV Bit 5	HOFF Bit 4	DIR Bit 2	PWM0	PWM1	PWM2	PWM3	State of Motor
0	X	0	X	1 (Disable)	1 (Enable)	1 (Disable)	1 (Enable)	Brake
X	X	1	X	1 (Disable)	0 (Disable)	1 (Disable)	0 (Disable)	Free Run
1	0	0	0	0 (Enable)	0 (Disable)	HS	LS	Move controlled by LS on PWM3
1	0	0	1	HS	LS	0 (Enable)	0 (Disable)	Move controlled by LS on PWM1
1	1	0	0	~LS	~HS	1 (Disable)	1 (Enable)	Move controlled by ~LS on PWM0
1	1	0	1	1 (Disable)	1 (Enable)	~LS	~HS	Move controlled by ~LS on PWM2

Note that HS = high-side, LS = low side, ~ = inverse, as programmed in PWM0 registers.

PWM INTERRUPT GENERATION

PWM Trip Enable Pin

When the PWM trip function is enabled (TRIPEN, PWMCON1[6]) and the PWM trip input signal goes low (falling edge), the PWM disables itself. It also generates the PWM trip interrupt. The interrupt is cleared by setting PWMCLRI[4].

PWM Interrupts

The PWM has four interrupts: IRQPWM0, IRQPWM1, IRQPWM2, and IRQPWM3.

When the interrupt generation is enabled (PWMCON0[10]) and the counter value for Pair 0 changes from PWM0LEN to 0, it also generates the IRQPWM0 interrupt.

The interrupt is cleared by setting PWMCLRI[0].

When the interrupt generation is enabled (PWMCON0[10]) and the counter value for Pair 1 changes from PWM1LEN to 0, it also generates the IRQPWM1 interrupt.

The interrupt is cleared by setting PWMCLRI[1].

When the interrupt generation is enabled (PWMCON0[10]) and the counter value for Pair 2 changes from PWM2LEN to 0, it also generates the IRQPWM2 interrupt.

The interrupt is cleared by setting PWMCLRI[2].

When the interrupt generation is enabled (PWMCON0[10]) and the counter value for Pair 3 changes from PWM3LEN to 0, it also generates the IRQPWM3 interrupt.

The interrupt is cleared by setting PWMCLRI[3].

REGISTER SUMMARY (PULSE-WIDTH MODULATION)

Table 262. Pulse Width Modulation Register Summary

Address	Name	Description	Reset	RW
0x40001000	PWMCON0	PWM Control Register	0x0012	RW
0x40001004	PWMCON1	Trip Control Register	0x00	RW
0x40001008	PWMCLRI	PWM Interrupt Clear	0x0000	W
0x40001010	PWM0COM0	Compare Register 0 for Pair 0 (PWM0 and PWM1)	0x0000	RW
0x40001014	PWM0COM1	Compare Register 1 for Pair 0 (PWM0 and PWM1)	0x0000	RW
0x40001018	PWM0COM2	Compare Register 2 for Pair 0 (PWM0 and PWM1)	0x0000	RW
0x4000101C	PWM0LEN	Period Value Register for Pair 0 (PWM0 and PWM1)	0x0000	RW
0x40001020	PWM1COM0	Compare Register 0 for Pair 1 (PWM2 and PWM3)	0x0000	RW
0x40001024	PWM1COM1	Compare Register 1 for Pair 1 (PWM2 and PWM3)	0x0000	RW
0x40001028	PWM1COM2	Compare Register 2 for Pair 1 (PWM2 and PWM3)	0x0000	RW
0x4000102C	PWM1LEN	Period Value Register for Pair 1 (PWM2 and PWM3)	0x0000	RW
0x40001030	PWM2COM0	Compare Register 0 for Pair 2 (PWM4 and PWM5)	0x0000	RW
0x40001034	PWM2COM1	Compare Register 1 for Pair 2 (PWM4 and PWM5)	0x0000	RW
0x40001038	PWM2COM2	Compare Register 2 for Pair 2 (PWM4 and PWM5)	0x0000	RW
0x4000103C	PWM2LEN	Period Value Register for Pair 2 (PWM4 and PWM5)	0x0000	RW
0x40001040	PWM3COM0	Compare Register 0 for Pair 3 (PWM6 and PWM7)	0x0000	RW
0x40001044	PWM3COM1	Compare Register 1 for Pair 3 (PWM6 and PWM7)	0x0000	RW
0x40001048	PWM3COM2	Compare Register 2 for Pair 3 (PWM6 and PWM7)	0x0000	RW
0x4000104C	PWM3LEN	Period Value Register for Pair 3 (PWM6 and PWM7)	0x0000	RW

REGISTER DETAILS (PULSE-WIDTH MODULATION)

PWM Control Register

Address: 0x40001000, Reset: 0x0012, Name: PWMCON0

Table 263. Bit Descriptions for PWMCON0

Bits	Bit Name	Description	Reset	Access
15	SYNC	PWM Synchronization. 0: DIS. Ignore transitions on the PWMSYNC pin. 1: EN. All PWM counters are reset on the next clock edge after the detection of a falling edge on the PWMSYNC pin.	0x0	RW
14	PWM7INV	Inversion of PWM output. Available in standard mode only. 0: DIS. PWM7 is normal. 1: EN. Invert PWM7.	0x0	RW
13	PWM5INV	Inversion of PWM output. Available in standard mode only. 0: DIS. PWM5 is normal. 1: EN. Invert PWM5.	0x0	RW
12	PWM3INV	Inversion of PWM output. Available in standard mode only. 0: DIS. PWM3 is normal. 1: EN. Invert PWM3.	0x0	RW
11	PWM1INV	Inversion of PWM output. Available in standard mode only. 0: DIS. PWM1 is normal. 1: EN. Invert PWM1.	0x0	RW
10	PWMIEN	Enable PWM interrupts. 0: DIS. Disable PWM interrupts. 1: EN. Enable PWM interrupts.	0x0	RW
9	ENA	Enable PWM outputs. Available in H-Bridge mode only. 0: DIS. Disable PWM outputs. 1: EN. Enable PWM outputs.	0x0	RW

Bits	Bit Name	Description	Reset	Access
[8:6]	PWMCP	PWM Clock Prescaler. Sets UCLK divider. 000: UCLK/2. 001: UCLK/4. 010: UCLK/8. 011: UCLK/16. 100: UCLK/32. 101: UCLK/64. 110: UCLK/128. 111: UCLK/256.	0x0	RW
5	POINV	Invert all PWM outputs. Available in H-bridge mode only. 0: DIS. PWM outputs as normal. 1: EN. Invert all PWM outputs.	0x0	RW
4	HOFF	High Side Off. Available in H-bridge mode only. 0: DIS. PWM outputs as normal. 1: EN. Force PWM0 and PWM2 outputs high and PWM1 and PWM3 low.	0x1	RW
3	LCOMP	Load Compare Registers. 0: DIS. Use the values previously stored in the internal compare registers. 1: EN. Load the internal compare registers with the values in PWMxCOMx on the next transition of the PWM timer from 0x00 to 0x01.	0x0	RW
2	DIR	Direction Control. Available in H-bridge mode only. 0: DIS. Enable PWM2 and PWM3 as the output signals while PWM0 and PWM1 are held low. 1: EN. Enable PWM0 and PWM1 as the output signals while PWM2 and PWM3 are held low.	0x0	RW
1	HMODE	Enable PWM mode of operation. 0: DIS. The PWM operates in standard mode. 1: EN. The PWM is configured in H-bridge mode.	0x1	RW
0	PWMEN	Enable all PWM outputs. 0: DIS. Disables all PWM outputs. 1: EN. Enables all PWM outputs.	0x0	RW

Note that except for LCOMP, all other bits of the PWMCON0 register can only be changed when PWMEN is low. When LCOMP is set to 1, it stays at that value until the new value is loaded in the compare register for all the channels.

Trip Control Register

Address: 0x40001004, Reset: 0x00, Name: PWMCON1

Table 264. Bit Descriptions for PWMCON1

Bits	Bit Name	Description	Reset	Access
7	RESERVED	Reserved. 0 should be written to these bits.	0x0	RW
6	TRIPEN	Enable PWM trip functionality. 0: DIS. Disable PWM trip functionality. 1: EN. Enable PWM trip functionality.	0x0	RW
[5:0]	RESERVED	Reserved. 0 should be written to these bits.	0x00	RW

PWM Interrupt Clear Register

Address: 0x40001008, Reset: 0x0000, Name: PWMCLRI

Table 265. Bit Descriptions for PWMCLRI

Bits	Bit Name	Description	Reset	Access
[15:5]	RESERVED	Reserved. These bits always read 0.	0x000	W
4	TRIP	Clear the latched trip interrupt. This bit always reads 0. 1: EN. Clear the latched PWMTRIP interrupt.	0x0	W
3	IRQPWM3	Clear the latched PWM3 interrupt. This bit always reads 0. 1: EN. Clear the latched IRQPWM3 interrupt.	0x0	W
2	IRQPWM2	Clear the latched PWM2 interrupt. This bit always reads 0. 1: EN. Clear the latched IRQPWM2 interrupt.	0x0	W
1	IRQPWM1	Clear the latched PWM1 interrupt. This bit always reads 0. 1: EN. Clear the latched IRQPWM1 interrupt.	0x0	W
0	IRQPWM0	Clear the latched PWM0 interrupt. This bit always reads 0. 1: EN. Clear the latched IRQPWM0 interrupt.	0x0	W

Compare Register 0 for Pair 0

Address: 0x40001010, Reset: 0x0000, Name: PWM0COM0

Table 266. Bit Descriptions for PWM0COM0

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM0 output pin goes high when the PWM timer reaches the count value stored in this register.	0x0000	RW

Compare Register 1 for Pair 0

Address: 0x40001014, Reset: 0x0000, Name: PWM0COM1

Table 267. Bit Descriptions for PWM0COM1

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM0 output pin goes low when the PWM timer reaches the count value stored in this register.	0x0000	RW

Compare Register 2 for Pair 0

Address: 0x40001018, Reset: 0x0000, Name: PWM0COM2

Table 268. Bit Descriptions for PWM0COM2

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM1 output pin goes low when the PWM timer reaches the count value stored in this register.	0x0000	RW

Period Value Register for Pair 0

Address: 0x4000101C, Reset: 0x0000, Name: PWM0LEN

Table 269. Bit Descriptions for PWM0LEN

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM1 output pin goes high when the PWM timer reaches the value stored in this register.	0x0000	RW

Compare Register 0 for Pair 1

Address: 0x40001020, Reset: 0x0000, Name: PWM1COM0

Table 270. Bit Descriptions for PWM1COM0

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM2 output pin goes high when the PWM timer reaches the count value stored in this register.	0x0000	RW

Compare Register 1 for Pair 1

Address: 0x40001024, Reset: 0x0000, Name: PWM1COM1

Table 271. Bit Descriptions for PWM1COM1

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM2 output pin goes low when the PWM timer reaches the count value stored in this register.	0x0000	RW

Compare Register 2 for Pair 1

Address: 0x40001028, Reset: 0x0000, Name: PWM1COM2

Table 272. Bit Descriptions for PWM1COM2

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM3 output pin goes low when the PWM timer reaches the count value stored in this register.	0x0000	RW

Period Value Register for Pair 1

Address: 0x4000102C, Reset: 0x0000, Name: PWM1LEN

Table 273. Bit Descriptions for PWM1LEN

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM3 output pin goes high when the PWM timer reaches the count value stored in this register.	0x0000	RW

Compare Register 0 for Pair 2

Address: 0x40001030, Reset: 0x0000, Name: PWM2COM0

Table 274. Bit Descriptions for PWM2COM0

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM4 output pin goes high when the PWM timer reaches the count value stored in this register.	0x0000	RW

Compare Register 1 for Pair 2

Address: 0x40001034, Reset: 0x0000, Name: PWM2COM1

Table 275. Bit Descriptions for PWM2COM1

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM4 output pin goes low when the PWM timer reaches the count value stored in this register.	0x0000	RW

Compare Register 2 for Pair 2

Address: 0x40001038, Reset: 0x0000, Name: PWM2COM2

Table 276. Bit Descriptions for PWM2COM2

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM5 output pin goes low when the PWM timer reaches the count value stored in this register.	0x0000	RW

Period Value Register for Pair 2

Address: 0x4000103C, Reset: 0x0000, Name: PWM2LEN

Table 277. Bit Descriptions for PWM2LEN

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM5 output pin goes high when the PWM timer reaches the count value stored in this register.	0x0000	RW

Compare Register 0 for Pair 3

Address: 0x40001040, Reset: 0x0000, Name: PWM3COM0

Table 278. Bit Descriptions for PWM3COM0

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM6 output pin goes high when the PWM timer reaches the count value stored in this register.	0x0000	RW

Compare Register 1 for Pair 3

Address: 0x40001044, Reset: 0x0000, Name: PWM3COM1

Table 279. Bit Descriptions for PWM3COM1

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM6 output pin goes Low when the PWM timer reaches the count value stored in this register.	0x0000	RW

Compare Register 2 for Pair 3

Address: 0x40001048, Reset: 0x0000, Name: PWM3COM2

Table 280. Bit Descriptions for PWM3COM2

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM7 output pin goes low when the PWM timer reaches the count value stored in this register.	0x0000	RW

Period Value Register for Pair 3

Address: 0x4000104C, Reset: 0x0000, Name: PWM3LEN

Table 281. Bit Descriptions for PWM3LEN

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	PWM7 output pin goes high when the PWM timer reaches the count value stored in this register.	0x0000	RW

POWER MANAGEMENT UNIT

POWER MANAGEMENT UNIT FEATURES

- Power supply support circuits
    - Low dropout regulators
    - Power-on-reset
    - Power supply monitor
- Programmable power modes
    - Active mode, where core and any peripherals can be active
    - Flexi mode, where the core is in sleep mode, but any peripheral can be active
    - Hibernate mode, where SRAM content is retained, watchdog and wake-up timers are active
    - Shutdown mode, where everything is disabled except POR and three external interrupts

POWER SUPPLY SUPPORT CIRCUITS OVERVIEW

Low Dropout Regulators

The ADuCRF101 integrates four on-chip regulators (LDO) that are driven directly from the battery voltage to generate a 1.8 V internal supply and 1.32 V internal supply. These regulated supplies are then used as the supply voltage for the Cortex-M3 memories and peripherals, including the precision analog circuits on chip. Additional LDOs are integrated as part of the RF transceiver.

The four LDOs operate in pairs: two LDOs share the same output pin, LVDD1 and the two others share the LVDD2 pin. They all need an external capacitor of 0.47 μF for stability. The LVDD1 pin and LVDD2 pin must be connected together via a 1 μF capacitor for correct operation of the device. For each pair of LDOs, the high power LDO is used in active and flexi modes and the low power LDO is used in hibernate mode. In shutdown mode all LDOs are off. Turning on and off the LDOs is done automatically when selecting the operation mode of the device.

The high power LDO on LVDD1 can supply external sensors for ratiometric measurements up to 20 mA, whereas the low power LDO can only supply 5 μA maximum, so the sensors should be disconnected before entering hibernate mode.

Power-On-Reset

The power-on-reset (POR) function is also integrated to ensure safe operation of the processor. The POR circuit is designed to ensure full functional operation of the Flash/EE memory-based processor during power-on and power-down cycles.

As shown in Figure 60, when the supply voltage on VBAT reaches a minimum voltage below the operating voltage of 2.2 V, a POR signal keeps the core in reset for approximately 48 ms. The output of the POR is available on P1.1.

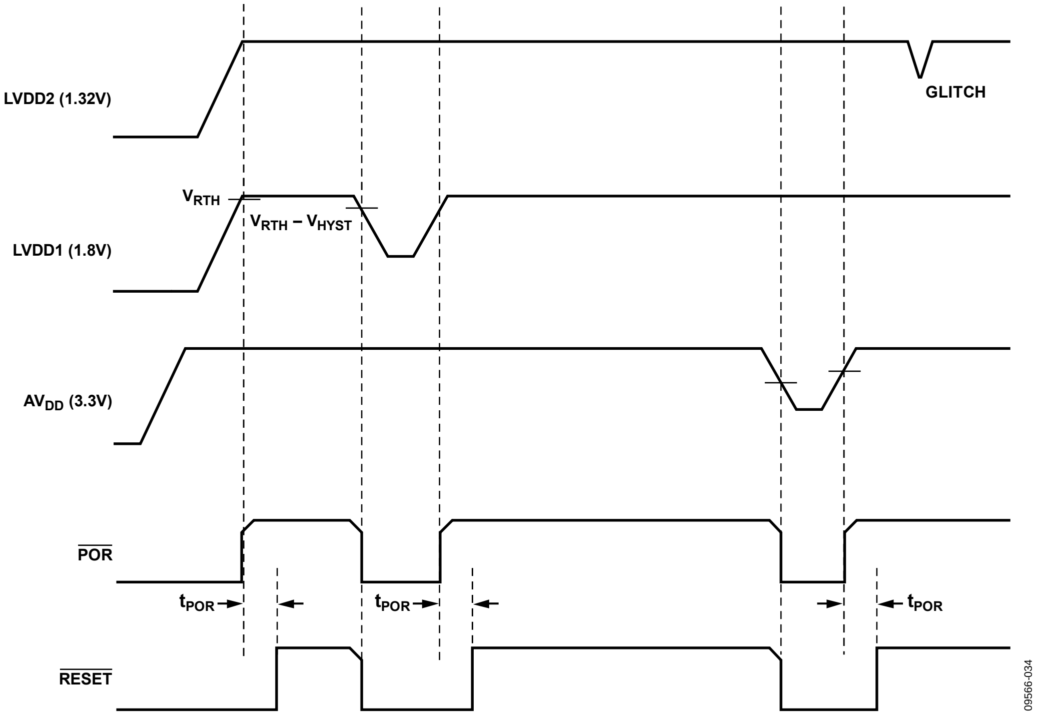


Figure 60. Typical Power-On Cycle

Power Supply Monitor

A power supply monitor (PSM) is also available on chip. It generates a nonmaskable interrupt if VBAT drops below 1.71 V approximately.

The PSM can be disabled by user code by setting PSMCON[1]=1. The PSM feature is not available in shutdown mode. It is available in all other modes, and the NMI wakes up the ADuCRF101 from Hibernate or Flexi modes.

PROGRAMMABLE POWER MODES OVERVIEW

The ADuCRF101 has four modes of operation to best minimize current consumption based on the application. The power management unit (PMU) controls the power modes of the ADuCRF101 with the exception of the RF link power modes, which are described in the RF Transceiver section. The device always starts up in active mode, and returns to active mode after waking up from any of the low power modes.

Active Mode

The Cortex-M3 processor is running and any peripheral can be enabled. The processor consumes <210 µA/ MHz. The default core clock is 16 MHz. Dynamic consumption can be minimized by dividing the system clock or disabling clocks of peripherals that are not in use in the application. This is done through the CLKACT register. This register is described in detail in the Clocking Architecture section.

Flexi Mode

In this mode, the Cortex-M3 processor is in light sleep mode. The user has complete flexibility to leave peripherals, including DMA and memories, on or off and to gate the clock to these peripherals using the CLKPD register described in the Clocking Architecture section. Any interrupt can wake-up the device in this mode. The wake-up time is 3 to 5 core clock cycles.

Hibernate Mode

The Cortex-M3 processor is in deep sleep mode, the processor clock is off. The low power oscillator and watchdog timer are active and GPIOs retain their state. The external crystal and wake-up timer can be enabled. The 16 kB of SRAM memory content is retained. To further reduce the power consumption, the user has the possibility to only retain half the SRAM content by clearing the RETAIN bit in the SRAMRET register. External interrupt, wake-up timer and watchdog timer only can wake-up the device. Wake-up time is ~10 µs.

Note that in this mode, low power LDOs are used. LVDD1 can only supply very low load currents (<5 µA) and should be disconnected from any external loads, for example, external resistive sensors as described in the ADC Circuit section.

It is also recommended to isolate GPIO from fast switching signals (>1 MHz) by configuring them in high impedance state before entering hibernate mode (see Digital Input/Outputs section).

Shutdown Mode

In this mode, only the pads state is retained. All other peripherals are powered down, LDOs are off, and therefore the SRAM content is not retained. External interrupts 0, 1 and 8 can wake-up the part from shutdown mode. Waking up from shutdown mode is equivalent to powering up, the code start executing from reset, the RSTSTA register indicates a POR but the shutdown acknowledge register indicate the source of the wake-up, external interrupt 0, 1 or 8. The device restarts in ~48 ms.

Note that it is recommended to power cycle the device to enter hibernator or shutdown mode after serial wire debug access.

The PMU and Cortex-M3 modes are summarized in Table 282.

Table 282. Power Modes Summary

Clock and Power	Active	Flexi	Hibernate	Shutdown
HP LDOs	On	On	Off	Off
LP LDOs	Off	Off	On	Off
HFOSC	On	On	Off	Off
LFOSC	On	On	On	Off
UCLK	On	On	Off	Off
FCLK	On	On	Off	Off
HCLK	On	On	Off	Off
PCLK	On	On	Off	Off
ACLK	On	On	Off	Off
SRAM	On	On	On	Off
CORTEX-M3	Active	Sleep	Deep sleep	Off

PROGRAMMABLE POWER MODES OPERATION

All communication peripherals should be disabled before entering low power modes where PCLK is disabled. Disabling the peripheral resets the state machine of these peripheral while keeping their configuration. This ensures that, on wake-up, the communication peripherals are in a known state.

See Table 284 for the required sequence to enter power down mode.

Power Management Unit Configuration

The low power mode should be configured in the PWRMOD register (Flexi, hibernate or shutdown). The write to this register must be complete before instructing the Cortex-M3 to enter power saving mode. This is ensured by using the DSB assembly instruction.

Cortex-M3 Configuration

The Cortex-M3 has two power saving modes, sleep mode and deep sleep mode, configured by the sleepdeep bit in the Cortex-M3 system control register (Address 0xE000ED10).

- To enter flexi mode, the Cortex-M3 should be configured by user code in sleep mode by clearing the sleepdeep bit. (However, enabling deep sleep mode in flexi mode does not have any effect.)
- To enter hibernate or shutdown mode, the Cortex-M3 must be configured by user code in deep sleep mode by setting the sleep deep bit.

The WFI instruction instructs the Cortex-M3 to go to sleep. The sleep-on-exit bit in the Cortex-M3 system control register can be used to automatically enter sleep mode on exit of an interrupt handler.

Additional Consideration for Hibernate and Shutdown Modes

When configuring the PWRMOD register for hibernate or shutdown mode, the Cortex-M3 is instructed to get ready to be placed in deep sleep mode. Bit 3 (WICENAK) of the PWRMOD register indicates if the Cortex-M3 is ready for deep sleep mode. The Cortex-M3 should not be instructed to go to sleep until this bit is set. Note that in debug mode, the WICENAK will not be set and user code can check for the presence of a debugger by checking for serial wire activity (SWACT register).

Serial Wire Activity Register

Address: 0x40008830, Reset: 0x0000, Name: SWACT

Table 283. Bit Descriptions for SWACT

Bits	Bit Name	Description	Reset	Access
[7:1]	RESERVED	Reserved.	0x0000	R
0	ACT	Serial wire activity 0: Disable 1: Enable	0x0	RW

Recommended Sequences

To ensure reliable entry into power-down mode, the sequences shown in Table 284 are suggested:

Table 284. Power down sequence

Sequence Description	Flexi Mode	Hibernate/Shutdown Modes

PMU Configuration	PWRMOD = 1
	Asm "DSB"
		SCR = 0x00000004;
		PWRMOD = x (x = 4 or 5)
		Asm "DSB"

Instruct the Cortex-M3 to go to sleep	Asm "WFI"	If no serial wire activity, wait until WICENAK = 1 Asm "WFI"

The debug tools can prevent the Cortex-M3 from fully entering its power saving modes by setting bits in the debug logic. Only a power-on-reset resets the debug logic. Therefore the device should be power cycled after using serial wire debug with application code containing the WFI instruction.

REGISTER SUMMARY (POWER MANAGEMENT UNIT)

Table 285. Power Management Unit Register Summary

Address	Name	Description	Reset	RW
0x40002400	PWRMOD	Power Modes Register	0x0000	RW
0x40002404	PWRKEY	Key Protection for the PWRMOD Register.	0x0000	RW
0x40002408	PSMCON	Power Supply Monitor Control and Status	0x03	RW
0x40002478	SRAMRET	SRAM Retention Register	0x01	RW
0x4000247C	SHUTDOWN	Shutdown Acknowledge Register	0x00	R

REGISTER DETAILS (POWER MANAGEMENT UNIT)

Power Modes Register

Address: 0x40002400, Reset: 0x0100, Name: PWRMOD

Table 286. Bit Descriptions for PWRMOD

Bits	Bit Name	Description	Reset	Access
[15:10]	RESERVED	Reserved. 0 should be written to these bits.	0x00	R
3	WICENACK	WIC acknowledge, for Cortex-M3 deep sleep mode. 0: CLR. Cleared automatically by hardware when the Cortex-M3 processor is not ready to enter deep sleep mode including if serial wire activity is detected. 1: SET. Set automatically by the Cortex-M3 processor when ready to enter sleep deep mode.	0x0	R
[2:0]	MOD	Low Power Mode. 000: FLEXI. 101: HIBERNATE. 110: SHUTDOWN.	0x0	RW

Key Protection for the PWRMOD Register.

Address: 0x40002404, Reset: 0x0000, Name: PWRKEY

Table 287. Bit Descriptions for PWRKEY

Bits	Bit Name	Description	Reset	Access
[15:0]	VALUE	Power control key register.. Two writes to the key are necessary to change the value in the key protected register, first 0x4859, and then 0xF27B. The protected register should be written next. A write to any other register returns the protection to the lock state. 0x4859: KEY1 0xF27B: KEY2 The PWRKEY register is used with the following protected registers: PWRMOD CLKACT[2] and CLKACT[1] SRAMRET	0x0000	RW

Power Supply Monitor Control and Status Register

Address: 0x40002408, Reset: 0x03, Name: PSMCON

Table 288. Bit Descriptions for PSMCON

Bits	Bit Name	Description	Reset	Access
[7:2]	RESERVED	Reserved. 0 should be written to these bits.	0x00	R
1	PD	Power Supply Monitor power down bit. 0: DIS. Power up the PSM. 1: EN. Power down the PSM.	0x1	RW
0	RESERVED	Reserved. 1 should be written to this bit.	0x1	RW

SRAM Retention Register

Address: 0x40002478, Reset: 0x01, Name: SRAMRET

Table 289. Bit Descriptions for SRAMRET

Bits	Bit Name	Description	Reset	Access
[7:1]	RESERVED	Reserved. 0 should be written to these bits.	0x00	R
0	RETAIN	SRAM retention enable bit. This bit selects the amount of SRAM memory that retains its contents during HIBERNATE mode. This register is key protected with PWRKEY. 0: DIS. To retain contents of the bottom 8 kB of SRAM only. 1: EN. To retain contents of the entire 16 kB of SRAM.	0x1	RW

Shutdown Acknowledge Register

Address: 0x4000247C, Reset: 0x00, Name: SHUTDOWN

Table 290. Bit Descriptions for SHUTDOWN

Bits	Bit Name	Description	Reset	Access
[7:3]	RESERVED	Reserved. 0 should be written to these bits.	0x00	R
2	EINT8	External Interrupt 8 detected during SHUTDOWN mode. 0: CLR. Cleared automatically by hardware when clearing IRQ8 in EICLR. 1: SET Indicates the interrupt was detected.	0x0	R
1	EINT1	External Interrupt 1 detected during SHUTDOWN mode. 0: CLR. Cleared automatically by hardware when clearing IRQ1 in EICLR. 1: SET Indicates the interrupt was detected.	0x0	R
0	EINT0	External Interrupt 0 detected during SHUTDOWN mode. 0: CLR. Cleared automatically by hardware when clearing IRQ0 in EICLR. 1: SET Indicates the interrupt was detected.	0x0	R

RESET

RESET FEATURES

There are four resets:

- External reset
- Power-on reset
- Watchdog timeout
- Software system reset

RESET OPERATION

The software system reset is provided as part of the Cortex-M3 processor. To generate a software system reset, the application interrupt/reset control register must be written with 0x05FA0004. This register is part of the NVIC register and is located at Address 0xE000ED0C. The RSTSTA register stores the cause for the reset until it is cleared by writing the RSTCLR register. RSTSTA and RSTCLR can be used during a reset exception service routine to identify the source of the reset.

Table 291. Device Reset Implications

	Impact						
Reset	**Reset External Pins to Default State**[1]	**Reset Debug Logic**	**Execute Kernel**	**Reset All MMRs Except RSTSTA**	**Reset All Peripherals**	**Valid SRAM**[2]	**RSTSTA After Reset Event**
SWRST	Yes	No	Yes	Yes	Yes	Yes/no	RSTSTA[4] = 1
WDRST	Yes	No	Yes	Yes	Yes	Yes/no	RSTSTA[3] = 1
EXTRST	Yes	No	Yes	Yes	Yes	Yes/no	RSTSTA[2] = 1
PORHV	Yes	Yes	Yes	Yes	Yes	No	RSTSTA[1] = 1
PORLV	Yes	Yes	Yes	Yes	Yes	No	RSTSTA[0] = 1

[1] P1.1returns to its default state, that is, POR output. It is only low in the case of a POR event; in all other cases, it remains high.
[2] RAM is not valid in the case of a reset following a UART download.

REGISTER SUMMARY (RESET)

Table 292. Reset Register Summary

Address	Name	Description	Reset	RW
0x40002440	RSTSTA	Reset Status	0x03	R
0x40002440	RSTCLR	Reset Status Clear	0x03	W

REGISTER DETAILS (RESET)

Reset Status Register

Address: 0x40002440, Reset: 0x03, Name: RSTSTA

Table 293. Bit Descriptions for RSTSTA

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved.	0x0	R
4	SWRST	Software reset status bit. 0: CLR. Indicates that no software reset has occurred. 1: SET. Indicates that a software reset has occurred.	0x0	R
3	WDRST	Watchdog reset status bit. 0: CLR. Indicates that no watchdog reset has occurred. 1: SET. Indicates that a watchdog reset has occurred.	0x0	R
2	EXTRST	External reset status bit. 0: CLR. Indicates that no external reset has occurred. 1: SET. Indicates an external reset has occurred.	0x0	R
1	PORHV	Power-on reset status bit HV. 0: CLR. Indicates a POR or wake up from SHUTDOWN has not occurred. 1: SET. This bit indicates that the AVDD supply has dropped below the POR trip point, causing a power on reset. It is also set when waking up from SHUTDOWN mode.	0x1	R
0	PORLV	Power-on reset status bit LV. 0: CLR. Indicates a POR or wake up from SHUTDOWN has not occurred. 1: SET. This bit indicates that the AVDD supply has dropped below the POR trip point, causing a power on reset. It is also set when waking up from SHUTDOWN mode.	0x1	R

Reset Status Clear Register

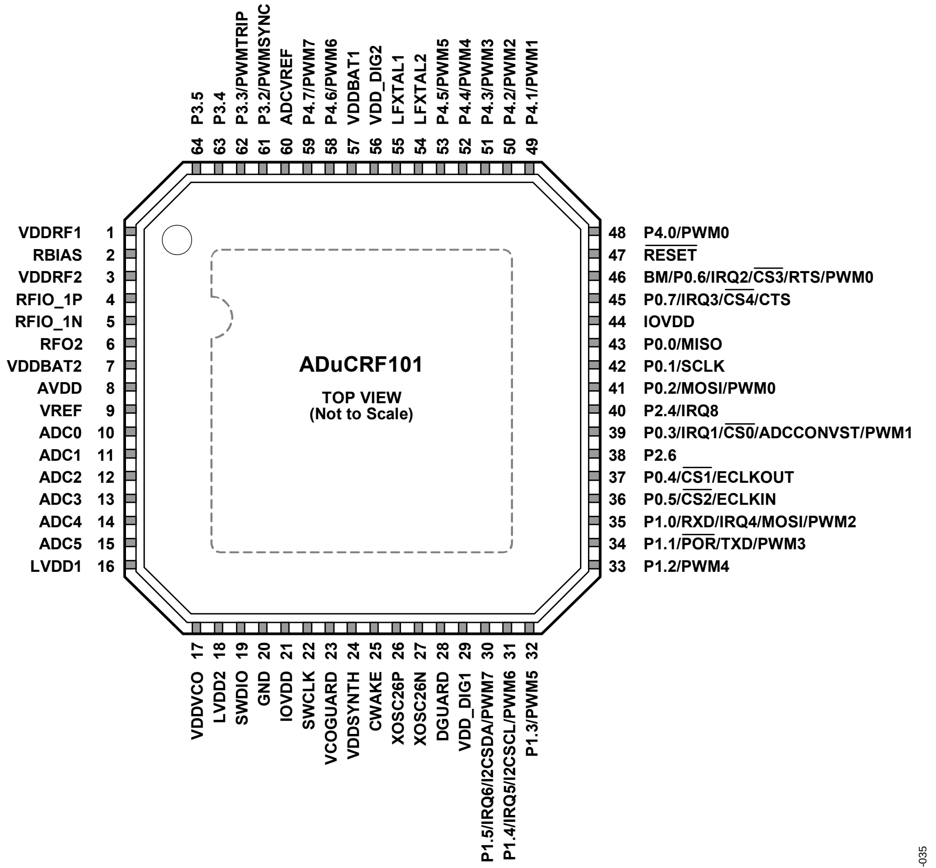
Address: 0x40002440, Reset: 0x03, Name: RSTCLR

Table 294. Bit Descriptions for RSTCLR

Bits	Bit Name	Description	Reset	Access
[7:5]	RESERVED	Reserved. 0 should be written to these bits.	0x0	W
4	SWRST	Software reset clear status bit. 0: DIS. Has no effect. 1: EN. Clears the SWRST status bit in RSTSTA.	0x0	W
3	WDRST	Watchdog reset clear status bit. 0: DIS. Has no effect. 1: EN. Clears the WDRST status bit in RSTSTA.	0x0	W
2	EXTRST	External reset clear status bit. 0: DIS. Has no effect. 1: EN. Clears the EXTRST status bit in RSTSTA.	0x0	W
1	PORHV	Power on reset clear status bit. 0: DIS. Has no effect. 1: EN. Clears PORLV status bit in RSTSTA.	0x1	W
0	PORLV	Power-on reset clear status bit LV. 0: DIS. Has no effect. 1: EN. Clears the PORLV status bit in RSTSTA.	0x1	W

HARDWARE DESIGN CONSIDERATIONS

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS



NOTES
1. THE EXPOSED PACKAGE PADDLE MUST BE SOLDERED TO A METAL PAD ON THE PCB AND CONNECTED TO GROUND.

Figure 61. 64-Lead LFCSP_VQ Pin Configuration

Table 295. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	VDDRF1	Voltage Regulator Output for RF Block. For regulator stability and noise rejection, place a 220 nF capacitor between this pin and ground.
2	RBIAS	External Bias Resistor. Use a 36 kΩ resistor with 2% tolerance.
3	VDDRF2	Voltage Regulator Output for RF Block. For regulator stability and noise rejection, place a 220 nF capacitor between this pin and ground.
4	RFIO_1P	LNA Positive Input in Receive Mode; Differential PA Positive Output in Transmit Mode.
5	RFIO_1N	LNA Negative Input in Receive Mode; Differential PA Negative Output in Transmit Mode.
6	RFO2	Single-Ended PA Output.
7	VDDBAT2	Battery Terminal[1]. Supply for the LDOs used in the RF section of the transceiver.
8	AVDD	Battery Terminal[1]. Supply for the analog circuits such as the ADC and ADC internal reference, POR, PSM, and LDOs.
9	VREF	Internal 1.25 V ADC Reference. Place a 0.47 μF capacitor between this pin and ground.
10	ADC0	ADC Input Channel 0. Input of DIFF0 pair in differential mode.
11	ADC1	ADC Input Channel 1. Input of DIFF0 pair in differential mode.
12	ADC2	ADC Input Channel 2. Input of DIFF1 pair in differential mode.
13	ADC3	ADC Input Channel 3. Input of DIFF1 pair in differential mode.
14	ADC4	ADC Input Channel 4. Input of DIFF2 pair in differential mode.
15	ADC5	ADC Input Channel 5. Input of DIFF2 pair in differential mode.

Pin No.	Mnemonic	Description
16	LVDD1	On-Chip LDO Decoupling Output. Connect a 0.47 µF capacitor to the 1.8 V output to ensure that the core operating voltage is stable. For correct operation, connect a 1 µF capacitor between this pin and LVDD2 (Pin 18).
17	VDDVCO	Voltage Regulator Output for Voltage Controlled Oscillator (VCO). For regulator stability and noise rejection, place a 220 nF capacitor between this pin and ground.
18	LVDD2	On-Chip LDO Decoupling Output. Connect a 0.47 µF capacitor to the 1.32 V output to ensure that the core operating voltage is stable. For correct operation, connect a 1 µF capacitor between this pin and LVDD1(Pin 16).
19	SWDIO	Serial Wire Bidirectional Data.
20	GND	Ground. Connect this pin to the exposed pad.
21	IOVDD	General-Purpose I/O Supply[1]. Connect this pin to the battery terminal.
22	SWCLK	Serial Wire Debug Clock.
23	VCOGUARD	Guard, Screen for VCO. Connect this pin to VDDVCO.
24	VDDSYNTH	Voltage Regulator Output for Synthesizer. For regulator stability and noise rejection, place a 220 nF capacitor between this pin and ground.
25	CWAKE	External Capacitor for Wake-Up Control. Place a 150 nF capacitor between this pin and ground.
26	XOSC26P	Connect the 26 MHz reference crystal between this pin and XOSC26N (HFXTAL).
27	XOSC26N	Connect the 26 MHz reference crystal between this pin and XOSC26P (HFXTAL).
28	DGUARD	Internal Guard, Screen for Digital Cells. Connect this pin to VDD_DIG1.
29	VDD_DIG1	Voltage Regulator Output for the Digital Section of the Transceiver. For regulator stability and noise rejection, place a 220 nF capacitor between this pin and ground.
30	P1.5/IRQ6/I2CSDA/PWM7	General-Purpose Input and Output Port 1.5 (P1.5). External Interrupt 6 (IRQ6). I^{2}C Serial Data (I2CSDA). PWM Channel 7 (PWM7).
31	P1.4/IRQ5/I2CSCL/PWM6	General-Purpose Input and Output Port 1.4 (P1.4). External Interrupt 5 (IRQ5). I^{2}C Serial Clock (I2CSCL). PWM Channel 6 (PWM6).
32	P1.3/PWM5	General-Purpose Input and Output Port 1.3 (P1.3). PWM Channel 5 (PWM5).
33	P1.2/PWM4	General-Purpose Input and Output Port 1.2 (P1.2). PWM Channel 4 (PWM4).
34	P1.1/$\overline{\text{POR}}$/TXD/PWM3	General-Purpose Input and Output Port 1.1 (P1.1). Power-On Reset Output ($\overline{\text{POR}}$). UART TXD (TXD). PWM Channel 3 (PWM3).
35	P1.0/RXD/IRQ4/MOSI/PWM2	General-Purpose Input and Output Port 1.0 (P1.0). UART RXD (RXD). External Interrupt 4 (IRQ4). SPI1 Master Out, Slave In (MOSI). PWM Channel 2 (PWM2).
36	P0.5/$\overline{\text{CS2}}$/ECLKIN	General-Purpose Input and Output Port 0.5 (P0.5). SPI1 Chip Select 2 ($\overline{\text{CS2}}$). External Clock Input (ECLKIN).
37	P0.4/$\overline{\text{CS1}}$/ECLKOUT	General-Purpose Input and Output Port 0.4 (P0.4). SPI1 Chip Select 1 ($\overline{\text{CS1}}$). External Clock Output (ECLKOUT).
38	P2.6	General-Purpose Input and Output Port 2.6. Do not connect this pin. This pin is connected internally to the RF transceiver. It can be used for BER measurements.

Pin No.	Mnemonic	Description
39	P0.3/IRQ1/$\overline{\text{CS0}}$/ADCCONVST/ PWM1	General-Purpose Input and Output Port 0.3 (P0.3). External Interrupt 1 (IRQ1). SPI1 Chip Select 0 ($\overline{\text{CS0}}$). ADC Convert Start (ADCCONVST). PWM Channel 1 (PWM1).
40	P2.4/IRQ8	General-Purpose Input and Output Port 2.4 (P2.4). Do not connect this pin. This pin is connected internally to the RF transceiver and can be used for debug purposes to monitor RF transceiver interrupts. External Interrupt 8 (IRQ8).
41	P0.2/MOSI/PWM0	General-Purpose Input and Output Port 0.2 (P0.2). SPI1 Master Out, Slave In (MOSI). PWM Channel 0 (PWM0).
42	P0.1/SCLK	General-Purpose Input and Output Port 0.1 (P0.1). SPI1 Serial Clock (SCLK).
43	P0.0/MISO	General-Purpose Input and Output Port 0.0 (P0.0). SPI1 Master In, Slave Out (MISO).
44	IOVDD	General-Purpose I/O Supply[1]. Connect this pin to the battery terminal.
45	P0.7/IRQ3/$\overline{\text{CS4}}$/CTS	General-Purpose Input and Output Port 0.7 (P0.7). External Interrupt 3 (IRQ3). SPI1 Chip Select 4 ($\overline{\text{CS4}}$). UART Handshake (CTS).
46	BM/P0.6/IRQ2/$\overline{\text{CS3}}$/RTS/PWM0	Boot Mode (BM). The ADuCRF101 enters serial download mode if P0.6 is low during, and for a short time after, an external reset event. It executes user code after any reset event or if P0.6 is high during an external reset event. General-Purpose Input and Output Port 0.6 (P0.6). External Interrupt 2 (IRQ2). SPI1 Chip Select 3 ($\overline{\text{CS3}}$). UART Handshake (RTS). PWM Channel 0 (PWM0).
47	$\overline{\text{RESET}}$	Reset, Active Low. A low signal on this pin for 24 system clocks causes the device to reset.
48	P4.0/PWM0	General-Purpose Input and Output Port 4.0 (P4.0). PWM Channel 0 (PWM0).
49	P4.1/PWM1	General-Purpose Input and Output Port 4.1 (P4.1). PWM Channel 1 (PWM1).
50	P4.2/PWM2	General-Purpose Input and Output Port 4.2 (P4.2). PWM Channel 2 (PWM2).
51	P4.3/PWM3	General-Purpose Input and Output Port 4.3 (P4.3). PWM Channel 3 (PWM3).
52	P4.4/PWM4	General-Purpose Input and Output Port 4.4 (P4.4).l PWM Channel 4 (PWM4).
53	P4.5/PWM5	General-Purpose Input and Output Port 4.5 (P4.5). PWM Channel 5 (PWM5).
54	LFXTAL2	32.768 kHz Watch Crystal Input for Wake-Up Timers.
55	LFXTAL1	32.768 kHz Watch Crystal Output for Wake-Up Timers.
56	VDD_DIG2	Voltage Regulator Output for the Digital Section of the Transceiver. For regulator stability and noise rejection, place a 220 nF capacitor between this pin and ground.
57	VDDBAT1	Battery Terminal[1]. Supply for the digital section of the transceiver and GPIOs.
58	P4.6/PWM6	General-Purpose Input and Output Port 4.6 (P4.6). PWM Channel 6 (PWM6).
59	P4.7/PWM7	General-Purpose Input and Output Port 4.7 (P4.7). PWM Channel 7 (PWM7).

Pin No.	Mnemonic	Description
60	ADCVREF	Transceiver ADC Reference Output. For adequate noise rejection, place a 220 nF capacitor between this pin and ground
61	P3.2/PWMSYNC	General-Purpose Input and Output Port 3.2 (P3.2). PWM Synchronization (PWMSYNC).
62	P3.3/PWMTRIP	General-Purpose Input and Output Port 3.3 (P3.3). PWM Safety Cutoff (PWMTRIP).
63	P3.4	General-Purpose Input and Output Port 3.4.
64	P3.5	General-Purpose Input and Output Port 3.5.
65	EP	Exposed Pad. The exposed package paddle must be soldered to a metal pad on the PCB and connected to ground.

[1] VDDBAT1, VDDBAT2, AVDD, and IOVDD must all be connected together.

POWER SUPPLY AND GROUND

All the supply pins must be connected together and decoupled to ground. The decoupling capacitor(s) should be placed as close as possible to the five supply pins on the device, AVDD, IOVDD (×2), VDDBAT1, and VDDBAT2. IOVDDs are used for GPIOs only. AVDD supplies the LDOs on LVDD1 and LVDD2 which in turn supply the processor, memories, and peripherals. VDDBAT1 supplies GPIOs and the LDO on VDD_DIG1 and VDDDIG2. Finally, VDDBAT2 supplies the LDOs on VDDVCO, VDDSYNTH, VDDRF1 and VDDRF2.

The two LDO pins, LVDD1 and LVDD2, require a 0.47 μF capacitor to ground and to be connected together by a 1 μF capacitor. Figure 62 shows a typical diagram. The six LDO pins VDD_DIG1, VDDDIG2, VDDVCO, VDDSYNTH, VDDRF1, and VDDRF2 require a 220 nF capacitor to ground.

Note that the paddle and the GND pin should both be connected to ground.

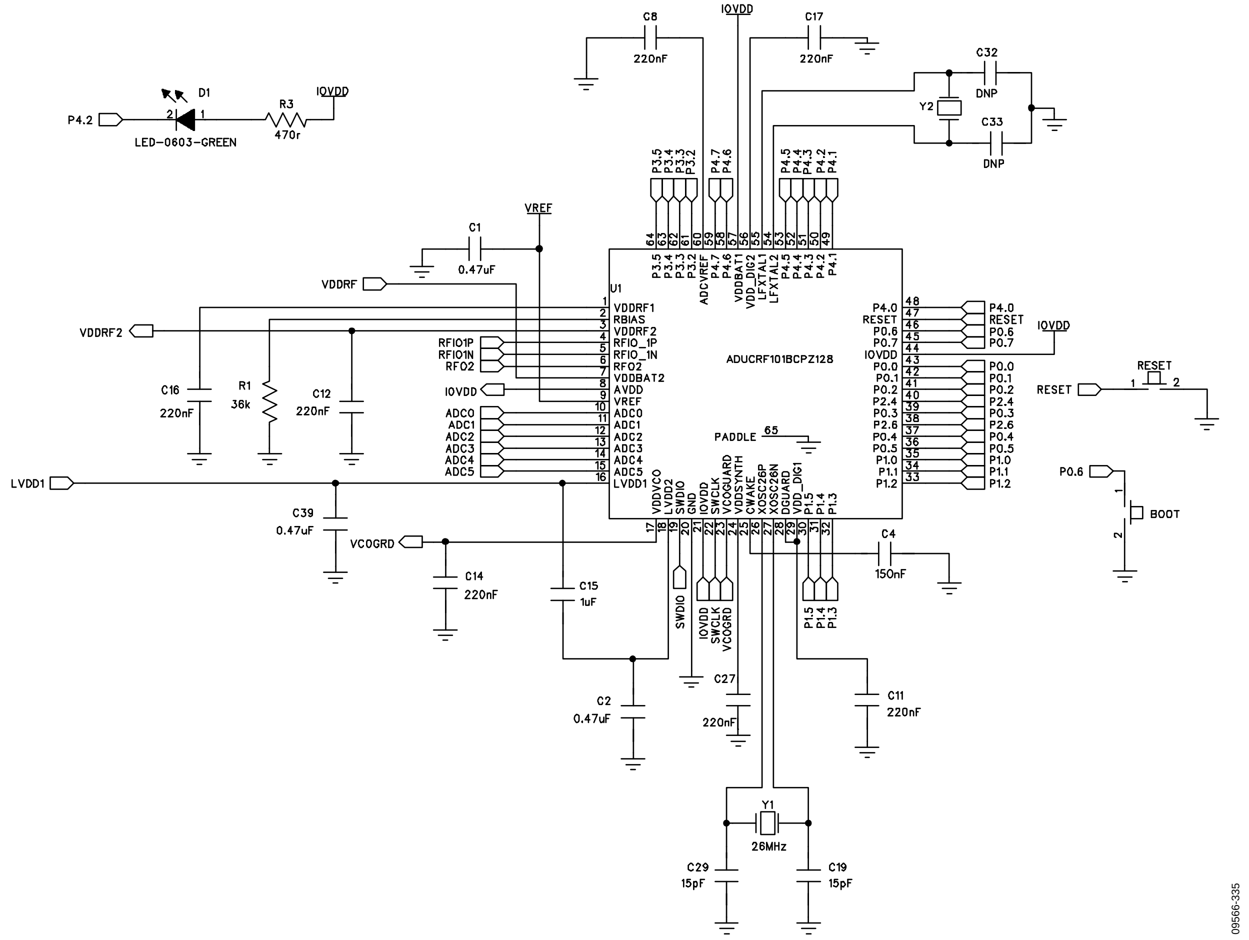


Figure 62. Typical Diagram

Other external components required for correct operation of the device are

- 0.47 μF on VREF pin, for ADC measurements
- 220 nF on ADCVREF pin, 150 nF on CWAKE pin, and 36 kΩ resistor on RBIAS for transceiver operation. The transceiver also requires a 26 MHz crystal, described in the external crystal oscillator section, and a matching network.

SERIAL WIRE DEBUG INTERFACE

Serial wire debug (SWD) provides a debug port for pin limited packages. SWD replaces the 5-pin JTAG port with a clock (SWDCLK) and a single bidirectional data pin (SWDIO), providing all the normal JTAG debug and test functionality. SWDIO and SWCLK are overlaid on the TMS and TCK pins on the ARM 20-PIN JTAG interface as shown in Figure 63 and Table 296.

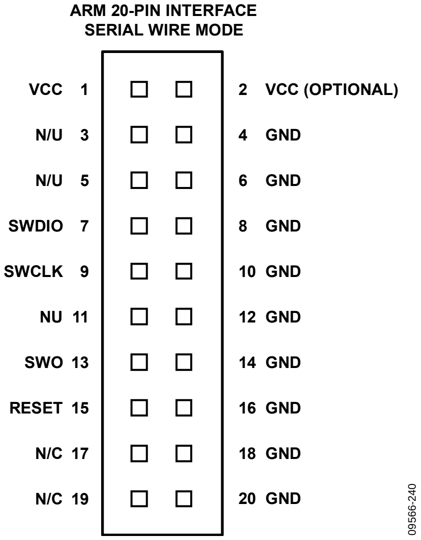


Figure 63. Serial Wire Mode on 20-Pin JTAG Connector

Table 296. SWD Connection

ARM 20-Pin Interface	ADuCRF101 Pins
SWDIO	SWDIO
SWO	No connect
SWCLK	SWCLK
VCC	IOVDD & VDD
GND	GND
RESET	RESET

Note that after serial wire debug access, the serial wire logic may be configured by the debugger to prevent entry of the Cortex-M in deep sleep mode, which in turns prevents the ADuCRF101 from entering hibernate or shutdown correctly. The debug logic is only cleared by a power cycle. It is, therefore, recommended to power cycle the device to enter hibernate or shutdown mode after serial wire debug access.

IN-CIRCUIT SERIAL DOWNLOAD ACCESS

In-circuit download is provided via the UART peripheral on P1.0 and P1.1. To enter serial download mode, P0.6 should be held low for a short time after the external reset pin is released. All other resets are ignored. The serial download protocol is described in details in the AN-1160.

Note that it is not possible to enter download mode directly from shutdown mode. Waking up from external reset is not flagged to the on-chip loader as an external reset event. therefore it is recommended to include a delay or conditions in user code to wake up from shutdown mode during development phase. This delay will allow a subsequent external reset to allow entry to serial download mode, or allow a serial wire debugger time to access the part before entering shutdown mode again.

EXTERNAL CRYSTAL OSCILLATORS

The ADuCRF101 has a total of four clock sources. This section deals with connecting an optional 32 kHz watch crystal to the LFXTAL pins and a 26 MHz crystal to the XOSC26 pins.

32 kHz Watch Crystal:

A 32.768 kHz watch crystal can be used to provide a high accuracy clock to the wakeup timer. The crystal oscillator is designed to support a 32.768 kHz watch crystal with a series resistance of 35 kΩ typically.

The internal structure and connections are shown in Figure 64. Pin capacitance (CPIN) is approximately 2 pF.

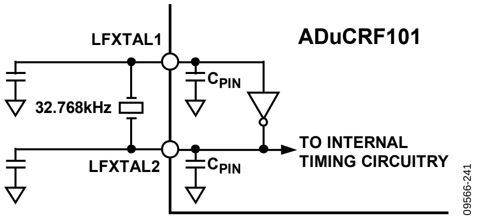


Figure 64: External Parallel Resonant Crystal Connections

26 MHz Crystal

A 26 MHz crystal oscillator operating in parallel mode must be connected between the XOSC26P and XOSC26N for the transceiver to operate. Two parallel loading capacitors are required for oscillation at the correct frequency. Their values are dependent upon the crystal specification. They should be chosen to ensure that the shunt value of capacitance added to the PCB track capacitance and the input pin capacitance of the RF transceiver equals the specified load capacitance of the crystal, usually 10 pF to 20 pF. Track capacitance values vary from 2 pF to 5 pF, depending on board layout. The total load capacitance is described by

$$C_{LOAD} =$$

$$\frac{1}{\frac{1}{C_1}+\frac{1}{C_2}} + \frac{C_{PIN}}{2} + C_{PCB}$$

where:
C_{LOAD} is the total load capacitance.
C1 and *C2* are the external crystal load capacitors.
C_{PIN} is the RF Transceiver input capacitance of the XOSC26P and XOSC26N pins and is equal to 2.1 pF.
C_{PCB} is the PCB track capacitance.

When possible, choose capacitors that have a very low temperature coefficient to ensure stable frequency operation over all conditions.

The crystal frequency error can be corrected by means of an integrated digital tuning varactor. See the RF Transceiver section for details.

Alternatively, any error in the RF frequency due to crystal error can be adjusted for by offsetting the RF channel frequency.

PA/LNA MATCHING

For the transceiver to receive and transmit, external matching network components are required. The different topologies to interface the ADuCRF101 to the antenna are described in the RF Transceiver section.

MEMORY MAPPED REGISTER SUMMARY

Table 297. Timer 0 Register Summary

Address	Name	Description	Reset	RW
0x40000000	T0LD	16-bit Load Value	0x0000	RW
0x40000004	T0VAL	16-bit Timer Value	0x0000	R
0x40000008	T0CON	Control Register	0x000A	RW
0x4000000C	T0CLRI	Clear Interrupt Register	0x0000	RW
0x40000010	T0CAP	Capture Register	0x0000	R
0x4000001C	T0STA	Status Register	0x0000	R

Table 298. Timer 1 Register Summary

Address	Name	Description	Reset	RW
0x40000400	T1LD	16-bit Load Value	0x0000	RW
0x40000404	T1VAL	16-bit Timer Value	0x0000	R
0x40000408	T1CON	Control Register	0x000A	RW
0x4000040C	T1CLRI	Clear Interrupt Register	0x0000	RW
0x40000410	T1CAP	Capture Register	0x0000	R
0x4000041C	T1STA	Status Register	0x0000	R

Table 299. Pulse-Width Modulation Register Summary

Address	Name	Description	Reset	RW
0x40001000	PWMCON0	PWM Control Register	0x0012	RW
0x40001004	PWMCON1	Trip Control Register	0x00	RW
0x40001008	PWMCLRI	PWM Interrupt Clear	0x0000	W
0x40001010	PWM0COM0	Compare Register 0 for PWM0 and PWM1	0x0000	RW
0x40001014	PWM0COM1	Compare Register 1 for PWM0 and PWM1	0x0000	RW
0x40001018	PWM0COM2	Compare Register 2 for PWM0 and PWM1	0x0000	RW
0x4000101C	PWM0LEN	Period Value Register for PWM0 and PWM1	0x0000	RW
0x40001020	PWM1COM0	Compare Register 0 for PWM2 and PWM3	0x0000	RW
0x40001024	PWM1COM1	Compare Register 1 for PWM2 and PWM3	0x0000	RW
0x40001028	PWM1COM2	Compare Register 2 for PWM2 and PWM3	0x0000	RW
0x4000102C	PWM1LEN	Period Value Register for PWM2 and PWM3	0x0000	RW
0x40001030	PWM2COM0	Compare Register 0 for PWM4 and PWM5	0x0000	RW
0x40001034	PWM2COM1	Compare Register 1 for PWM4 and PWM5	0x0000	RW
0x40001038	PWM2COM2	Compare Register 2 for PWM4 and PWM5	0x0000	RW
0x4000103C	PWM2LEN	Period Value Register for PWM4 and PWM5	0x0000	RW
0x40001040	PWM3COM0	Compare Register 0 for PWM6 and PWM7	0x0000	RW
0x40001044	PWM3COM1	Compare Register 1 for PWM6 and PWM7	0x0000	RW
0x40001048	PWM3COM2	Compare Register 2 for PWM6 and PWM7	0x0000	RW
0x4000104C	PWM3LEN	Period Value Register for PWM6 and PWM7	0x0000	RW

Table 300. Clock Control Register Summary

Address	Name	Description	Reset	RW
0x40002000	CLKCON	System Clocking Architecture Control Register	0x0000	RW
0x40002410	XOSCCON	Crystal Oscillator Control Register	0x00	RW
0x40002480	CLKACT	Clock in Active Mode Enable Register	0x3FFF	RW
0x40002484	CLKPD	Clock in Power-Down Mode Enable Register	0x3FFF	RW

Table 301. Power Management Unit Register Summary

Address	Name	Description	Reset	RW
0x40002400	PWRMOD	Power Modes Register	0x0100	RW
0x40002404	PWRKEY	Key Protection for the PWRMOD Register.	0x0000	RW
0x40002408	PSMCON	Power Supply Monitor Control and Status	0x03	RW
0x40002478	SRAMRET	SRAM Retention Register	0x01	RW
0x4000247C	SHUTDOWN	Shutdown Acknowledge Register	0x00	RW

Table 302. Interrupts Register Summary

Address	Name	Description	Reset	RW
0x40002420	EI0CFG	External Interrupt Configuration Register 0	0x0000	RW
0x40002424	EI1CFG	External Interrupt Configuration Register 1	0x0000	RW
0x40002428	EI2CFG	RF Transceiver Interrupt Configuration Register	0x0000	RW
0x40002430	EICLR	External Interrupts Clear Register	0x0000	RW
0x40002434	NMICLR	NMI Clear Register	0x00	RW

Table 303. Reset Register Summary

Address	Name	Description	Reset	RW
0x40002440	RSTSTA	Reset Status	0x03	R
0x40002440	RSTCLR	Reset Status Clear	0x03	W

Table 304. Wake-Up Timer Register Summary

Address	Name	Description	Reset	RW
0x40002500	T2VAL0	Current Wake-Up Timer Value LSB	0x0000	R
0x40002504	T2VAL1	Current Wake-Up Timer Value MSB	0x0000	R
0x40002508	T2CON	Control Register	0x0040	RW
0x4000250C	T2INC	12-bit Interval Register for Wake-Up Field A	0x00C8	RW
0x40002510	T2WUFB0	Wake-Up Field B LSB	0x1FFF	RW
0x40002514	T2WUFB1	Wake-Up Field B MSB	0x0000	RW
0x40002518	T2WUFC0	Wake-Up Field C LSB	0x2FFF	RW
0x4000251C	T2WUFC1	Wake-Up Field C MSB	0x0000	RW
0x40002520	T2WUFD0	Wake-Up Field D LSB	0x3FFF	RW
0x40002524	T2WUFD1	Wake-Up Field D MSB	0x0000	RW
0x40002528	T2IEN	Interrupt Enable	0x0000	RW
0x4000252C	T2STA	Status	0x0000	R
0x40002530	T2CLRI	Clear Interrupts	0x0000	W
0x4000253C	T2WUFA0	Wake-Up Field A LSB	0x1900	R
0x40002540	T2WUFA1	Wake-Up Field A MSB	0x0000	R

Table 305. Watchdog Timer Register Summary

Address	Name	Description	Reset	RW
0x40002580	T3LD	16-bit Load Value	0x1000	RW
0x40002584	T3VAL	16-bit Timer Value	0x1000	R
0x40002588	T3CON	Control Register	0x00E9	RW
0x4000258C	T3CLRI	Clear Interrupt Register	0x0000	W
0x40002598	T3STA	Status Register	0x0020	R

Table 306. Flash Controller Register Summary

Address	Name	Description	Reset	RW
0x40002800	FEESTA	Status Register	0x0000	R
0x40002804	FEECON0	Command Control Register	0x0000	RW
0x40002808	FEECMD	Command Register	0x0000	RW
0x40002810	FEEADR0L	Low Page (Lower 16 bits)	0x0000	RW
0x40002814	FEEADR0H	Low Page (Upper 16 bits)	0x0000	RW
0x40002818	FEEADR1L	Hi Page (Lower 16 bits)	0x0000	RW
0x4000281C	FEEADR1H	Hi Page (Upper 16 bits)	0x0000	RW
0x40002820	FEEKEY	Key Register	0x0000	W
0x40002828	FEEPROL	Write Protection Register (Lower 16 bits)	0xFFFF	RW
0x4000282C	FEEPROH	Write Protection Register (Upper 16 bits)	0xFFFF	RW
0x40002830	FEESIGL	Signature (Lower 16 bits)	0xFFFF	R
0x40002834	FEESIGH	Signature (Upper 16 bits)	0xFFFF	R
0x40002838	FEECON1	User Setup Register	0x0001	RW
0x40002848	FEEADRAL	Abort Address Register (Lower 16 bits)	0x0800	R
0x4000284C	FEEADRAH	Abort Address Register (Upper 16 bits)	0x0002	R
0x40002878	FEEAEN0	Interrupt Abort Register (Interrupt 15 to Interrupt 0)	0x0000	RW
0x4000287C	FEEAEN1	Interrupt Abort Register (Interrupt 31 to Interrupt 16)	0x0000	RW
0x40002880	FEEAEN2	Interrupt Abort Register (Interrupt 47 to Interrupt 32)	0x0000	RW

Table 307. I^{2}C Register Summary

Address	Name	Description	Reset	RW
0x40003000	I2CMCON	Master Control Register	0x0000	RW
0x40003004	I2CMSTA	Master Status Register	0x0000	R
0x40003008	I2CMRX	Master Receive Data Register	0x00	R
0x4000300C	I2CMTX	Master Transmit Data Register	0x00	W
0x40003010	I2CMRXCNT	Master Receive Data Count Register	0x0000	RW
0x40003014	I2CMCRXCNT	Master Current Receive Data Count Register	0x0000	R
0x40003018	I2CADR0	First Master Address Byte Register	0x00	RW
0x4000301C	I2CADR1	Second Master Address Byte Register	0x00	RW
0x40003024	I2CDIV	Serial Clock Period Divisor Register	0x1F1F	RW
0x40003028	I2CSCON	Slave Control Register	0x0000	RW
0x4000302C	I2CSSTA	Slave I2C Status, Error and IRQ Register	0x0001	R
0x40003030	I2CSRX	Slave Receive Data Register	0x0000	R
0x40003034	I2CSTX	Slave Transmit Data Register	0x0000	W
0x40003038	I2CALT	Hardware General Call ID Register	0x0000	RW
0x4000303C	I2CID0	First Slave Address Device ID	0x0000	RW
0x40003040	I2CID1	Second Slave Address Device ID	0x0000	RW
0x40003044	I2CID2	Third Slave Address Device ID	0x0000	RW
0x40003048	I2CID3	Fourth Slave Address Device ID	0x0000	RW
0x4000304C	I2CFSTA	Master and Slave Rx/Tx FIFO Status Register	0x0000	RW

Table 308. Serial Peripheral Interface Register Summary

Address	Name	Description	Reset	RW
0x40004000	SPI0STA	SPI0 Status Register	0x0000	R
0x40004004	SPI0RX	SPI0 Receive Register	0x00	R
0x40004008	SPI0TX	SPI0 Transmit Register	0x00	W
0x4000400C	SPI0DIV	SPI0 Bit Rate Selection Register	0x0000	RW
0x40004010	SPI0CON	SPI0 Configuration Register	0x0000	RW
0x40004014	SPI0DMA	SPI0 DMA Enable Register	0x0000	RW
0x40004018	SPI0CNT	SPI0 DMA Master Received Byte Count Register	0x0000	RW
0x40004400	SPI1STA	SPI1 Status Register	0x0000	R
0x40004404	SPI1RX	SPI1 Receive Register	0x00	R
0x40004408	SPI1TX	SPI1Transmit Register	0x00	W
0x4000440C	SPI1DIV	SPI1 Bit Rate Selection Register	0x0000	RW
0x40004410	SPI1CON	SPI1 Configuration Register	0x0000	RW
0x40004414	SPI1DMA	SPI1 DMA Enable Register	0x0000	RW
0x40004418	SPI1CNT	SPI1 DMA Master Receive Byte Count Register	0x0000	RW

Table 309. UART Register Summary

Address	Name	Description	Reset	RW
0x40005000	COMTX	Transmit Holding Register	0x00	W
0x40005000	COMRX	Receive Buffer Register	0x00	R
0x40005004	COMIEN	Interrupt Enable Register	0x00	RW
0x40005008	COMIIR	Interrupt Identification Register	0x01	R
0x4000500C	COMLCR	Line Control Register	0x00	RW
0x40005010	COMMCR	Module Control Register	0x00	RW
0x40005014	COMLSR	Line Status Register	0x60	R
0x40005018	COMMSR	Modem Status Register	0x00	R
0x40005024	COMFBR	Fractional Baud Rate Register.	0x0000	RW
0x40005028	COMDIV	Baud Rate Divider Register	0x0001	RW

Table 310. General-Purpose Input Output Register Summary

Address	Name	Description	Reset	RW
0x40006000	GP0CON	GPIO Port 0 Configuration	0x0000	RW
0x40006004	GP0OEN	GPIO Port 0 Output Enable	0x00	RW
0x40006008	GP0PUL	GPIO Port 0 Pull-Up Enable	0xFF	RW
0x4000600C	GP0OCE	GPIO Port 0 Open Collector	0x00	RW
0x40006014	GP0IN	GPIO Port 0 Data Input	0xFF	R
0x40006018	GP0OUT	GPIO Port 0 Data Out	0x00	RW
0x4000601C	GP0SET	GPIO Port 0 Data Out Set	0x00	W
0x40006020	GP0CLR	GPIO Port 0 Data Out Clear	0x00	W
0x40006024	GP0TGL	GPIO Port 0 Pin Toggle	0x00	W
0x40006030	GP1CON	GPIO Port 1 Configuration	0x0000	RW
0x40006034	GP1OEN	GPIO Port 1 Output Enable	0x00	RW
0x40006038	GP1PUL	GPIO Port 1 Pull-Up Enable	0x7F	RW
0x4000603C	GP1OCE	GPIO Port 1 Open Collector	0x00	RW
0x40006044	GP1IN	GPIO Port 1 Data Input	0x7F	R
0x40006048	GP1OUT	GPIO Port 1 Data Out	0x00	RW
0x4000604C	GP1SET	GPIO Port 1 Data Out Set	0x00	W
0x40006050	GP1CLR	GPIO Port 1 Data Out Clear	0x00	W
0x40006054	GP1TGL	GPIO Port 1 Pin Toggle.	0x00	W

Address	Name	Description	Reset	RW
0x40006060	GP2CON	GPIO Port 2 Configuration	0x0000	RW
0x40006064	GP2OEN	GPIO Port 2 Output Enable	0x00	RW
0x40006068	GP2PUL	GPIO Port 2 Pull-Up Enable	0xFF	RW
0x4000606C	GP2OCE	GPIO Port 2 Open Collector	0x00	RW
0x40006074	GP2IN	GPIO Port 2 Data Input	0xEF	R
0x40006078	GP2OUT	GPIO Port 2 Data Out	0x00	RW
0x4000607C	GP2SET	GPIO Port 2 Data Out Set	0x00	W
0x40006080	GP2CLR	GPIO Port 2 Data Out Clear	0x00	W
0x40006084	GP2TGL	GPIO Port 2 Pin Toggle	0x00	W
0x40006090	GP3CON	GPIO Port 3 Configuration	0x0550	RW
0x40006094	GP3OEN	GPIO Port 3 Output Enable	0x00	RW
0x40006098	GP3PUL	GPIO Port 3 Pull-Up Enable	0xEF	RW
0x4000609C	GP3OCE	GPIO Port 3 Open Collector	0x00	RW
0x400060A4	GP3IN	GPIO Port 3 Data Input	0xFF	R
0x400060A8	GP3OUT	GPIO Port 3 Data Out	0x00	RW
0x400060AC	GP3SET	GPIO Port 3 Data Out Set	0x00	W
0x400060B0	GP3CLR	GPIO Port 3 Data Out Clear	0x00	W
0x400060B4	GP3TGL	GPIO Port 3 Pin Toggle	0x00	W
0x400060C0	GP4CON	GPIO Port 4 Configuration	0x5555	RW
0x400060C4	GP4OEN	GPIO Port 4 Output Enable	0x00	RW
0x400060C8	GP4PUL	GPIO Port 4 Pull-Up Enable	0xFF	RW
0x400060CC	GP4OCE	GPIO Port 4 Open Collector	0x00	RW
0x400060D4	GP4IN	GPIO Port 4 Data Input	0xFF	R
0x400060D8	GP4OUT	GPIO Port 4 Data Out	0x00	RW
0x400060DC	GP4SET	GPIO Port 4 Data Out Set	0x00	W
0x400060E0	GP4CLR	GPIO Port 4 Data Out Clear	0x00	W
0x400060E4	GP4TGL	GPIO Port 4 Pin Toggle	0x00	W

Table 311. Miscellaneous Registers Register Summary

Address	Name	Description	Reset	RW
0x40008830	SWACT	Serial Wire Activity Register	0x00	RW

Table 312. Direct Memory Access Register Summary

Address	Name	Description	Reset	RW
0x40010000	DMASTA	Status Register	0x000D0000	R
0x40010004	DMACFG	Configuration Register	0x00000000	RW
0x40010008	DMAPDBPTR	Primary Control Database Pointer Register	0x00000000	RW
0x4001000C	DMAADBPTR	Alternate Control Database Pointer Register	0x00000100	R
0x40010014	DMASWREQ	Channel Software Request Register	0x00000000	W
0x40010020	DMARMSKSET	Channel Request Mask Set Register	0x00000000	RW
0x40010024	DMARMSKCLR	Channel Request Mask Clear Register	0x00000000	W
0x40010028	DMAENSET	Channel Enable Set Register	0x00000000	RW
0x4001002C	DMAENCLR	Channel Enable Clear Register	0x00000000	W
0x40010030	DMAALTSET	Channel Primary-Alternate Set Register	0x00000000	RW
0x40010034	DMAALTCLR	Channel Primary-Alternate Clear Register	0x00000000	W
0x40010038	DMAPRISET	Channel Priority Set Register	0x00000000	RW
0x4001003C	DMAPRICLR	Channel Priority Clear Register	0x00000000	W
0x4001004C	DMAERRCLR	Bus Error Clear Register	0x00000000	RW
0x40010FD0	DMAPERID4	DMA Peripheral Identification 4 Register	0x04	R
0x40010FE0	DMAPERID0	DMA Peripheral Identification 0 Register	0x30	R
0x40010FE4	DMAPERID1	DMA Peripheral Identification 1 Register	0xB2	R
0x40010FE8	DMAPERID2	DMA Peripheral Identification 2 Register	0x0B	R
0x40010FEC	DMAPERID3	DMA Peripheral Identification 3 Register	0x00	R
0x40010FF0	DMAPCELLID0	DMA PrimeCell Identification 0 Register	0x0D	R
0x40010FF4	DMAPCELLID1	DMA PrimeCell Identification 1 Register	0xF0	R
0x40010FF8	DMAPCELLID2	DMA PrimeCell Identification 2 Register	0x05	R
0x40010FFC	DMAPCELLID3	DMA PrimeCell Identification 3 Register	0xB1	R

Table 313. Analog-to-Digital Converter Register Summary

Address	Name	Description	Reset	RW
0x40050000	ADCCFG	ADC Configuration Register	0x0A00	RW
0x40050004	ADCCON	ADC Control Register	0x90	RW
0x40050008	ADCSTA	ADC Status Register	0x00	R
0x4005000C	ADCDAT	ADC Data Register	0x0000	R
0x40050010	ADCGN	ADC Gain Register	0x0000	RW
0x40050014	ADCOF	ADC Offset Register	0x0000	RW

RELATED LINKS AND DOCUMENTS

Resource	Description
ADuCRF101	Product page, ADuCRF101 precision analog microcontroller ARM Cortex-M3 RF transceiver.
ADuCRF101 Data Sheet	Available on the product page.

I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).

ESD Caution

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

Legal Terms and Conditions

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners. Information contained within this document is subject to change without notice. Software or hardware provided by Analog Devices may not be disassembled, decompiled or reverse engineered. Analog Devices' standard terms and conditions for products purchased from Analog Devices can be found at: http://www.analog.com/en/content/analog_devices_terms_and_conditions/fca.html.

UG09566-0-11/14(A)

ANALOG DEVICES

www.analog.com