

nRF52833

Product Specification

v1.3

Feature list

Features:

- **Bluetooth® 5.1**, IEEE 802.15.4-2006, 2.4 GHz transceiver
 - -96 dBm sensitivity in 1 Mbps **Bluetooth®** low energy mode
 - -103 dBm sensitivity in 125 kbps **Bluetooth®** low energy mode (long range)
 - -20 to +8 dBm TX power, configurable in 4 dB steps
 - On-air compatible with nRF52, nRF51, nRF24L, and nRF24AP Series
 - Supported data rates:
 - **Bluetooth®** 5.1 – 2 Mbps, 1 Mbps, 500 kbps, and 125 kbps
 - IEEE 802.15.4-2006 – 250 kbps
 - Proprietary 2.4 GHz – 2 Mbps, 1 Mbps
 - Angle-of-arrival (AoA) and angle-of-departure (AoD) direction finding using **Bluetooth®**
 - Single-ended antenna output (on-chip balun)
 - 128-bit AES/ECB/CCM/AAR co-processor (on-the-fly packet encryption)
 - 4.9 mA peak current in TX (0 dBm)
 - 4.6 mA peak current in RX
 - RSSI (1 dB resolution)
- **ARM® Cortex® -M4** 32-bit processor with FPU, 64 MHz
 - 217 EEMBC CoreMark® score running from flash memory
 - 52 µA/MHz running CoreMark from flash memory
 - 38 µA/MHz running CoreMark from RAM
 - Watchpoint and trace debug modules (DWT, ETM, and ITM)
 - Serial wire debug (SWD)
- Rich set of security features
 - Secure boot ready
 - Flash access control list (ACL)
 - Debug control and configuration
 - Access port protection (CTRL-AP)
 - Secure erase
- Flexible power management
 - 1.7 V to 5.5 V supply voltage range
 - On-chip DC/DC and LDO regulators with automated low current modes
 - Automated peripheral power management
 - Fast wake-up using 64 MHz internal oscillator
 - 0.6 µA at 3 V in System OFF mode, no RAM retention
 - 1.5 µA at 3 V in System ON mode, no RAM retention, wake on RTC
- 512 kB flash and 128 kB RAM
- Advanced on-chip interfaces
 - USB 2.0 full speed (12 Mbps) controller
 - High-speed 32 MHz SPI
 - Type 2 near field communication (NFC-A) tag with wake-on field
 - Touch-to-pair support
 - Programmable peripheral interconnect (PPI)
 - 42 general purpose I/O pins
 - EasyDMA automated data transfer between memory and peripherals
- Nordic SoftDevice ready with support for concurrent multiprotocol
- 12-bit, 200 ksp/s ADC – 8 configurable channels with programmable gain
- 64 level comparator
- 15 level low-power comparator with wake-up from System OFF mode
- Temperature sensor
- 4x four channel pulse width modulator (PWM) unit with EasyDMA
- Audio peripherals – I²S, digital microphone interface (PDM)
- 5x 32-bit timer with counter mode
- Up to 4x SPI master/3x SPI slave with EasyDMA
- Up to 2x I²C compatible two-wire master/slave
- 2x UART (CTS/RTS) with EasyDMA
- Quadrature decoder (QDEC)
- 3x real-time counter (RTC)
- Single crystal operation
- Operating temperature from -40 to 105 °C
- Package variants
 - aQFN™ 73 package, 7 x 7 mm
 - QFN40 package, 5 x 5 mm
 - WLCSP package, 3.175 x 3.175 mm

Applications:

- Advanced computer peripherals and I/O devices
 - Mouse
 - Keyboard
 - Multi-touch trackpad
- Advanced wearables
 - Health/fitness sensor and monitor devices
 - Wireless payment enabled devices
- Internet of things (IoT)
 - Smart home sensors and controllers
 - Industrial IoT sensors and controllers
- Interactive entertainment devices
 - Remote controls
 - Gaming controllers

Contents

Feature list	ii
1 Revision history	12
2 About this document	13
2.1 Document status	13
2.2 Peripheral chapters	13
2.3 Register tables	13
2.3.1 Fields and values	14
2.3.2 Permissions	14
2.4 Registers	14
2.4.1 DUMMY	14
3 Block diagram	16
4 Core components	18
4.1 CPU	18
4.1.1 Floating point interrupt	18
4.1.2 CPU and support module configuration	18
4.1.3 Electrical specification	19
4.2 Memory	19
4.2.1 RAM - Random access memory	20
4.2.2 Flash - Non-volatile memory	20
4.2.3 Memory map	20
4.2.4 Instantiation	22
4.3 NVMC — Non-volatile memory controller	23
4.3.1 Writing to flash	23
4.3.2 Erasing a page in flash	24
4.3.3 Writing to user information configuration registers (UICR)	24
4.3.4 Erasing user information configuration registers (UICR)	24
4.3.5 Erase all	24
4.3.6 Access port protection behavior	24
4.3.7 NVMC power failure protection	24
4.3.8 Partial erase of a page in flash	25
4.3.9 Cache	25
4.3.10 Registers	25
4.3.11 Electrical specification	30
4.4 FICR — Factory information configuration registers	30
4.4.1 Registers	30
4.5 UICR — User information configuration registers	40
4.5.1 Registers	40
4.6 EasyDMA	44
4.6.1 EasyDMA error handling	45
4.6.2 EasyDMA array list	45
4.7 AHB multilayer	46
4.8 Debug and trace	47
4.8.1 DAP - Debug access port	48
4.8.2 CTRL-AP - Control access port	48
4.8.3 Debug interface mode	50
4.8.4 Real-time debug	51

4.8.5 Trace	51
5 Power and clock management.	52
5.1 Power management unit (PMU)	52
5.2 Current consumption	52
5.2.1 Electrical specification	53
5.3 POWER — Power supply	58
5.3.1 Main supply	59
5.3.2 USB supply	64
5.3.3 System OFF mode	65
5.3.4 System ON mode	66
5.3.5 RAM power control	66
5.3.6 Reset	67
5.3.7 Registers	68
5.3.8 Electrical specification	78
5.4 CLOCK — Clock control	80
5.4.1 HFCLK controller	80
5.4.2 LFCLK controller	82
5.4.3 Registers	85
5.4.4 Electrical specification	93
6 Peripherals.	96
6.1 Peripheral interface	96
6.1.1 Peripheral ID	96
6.1.2 Peripherals with shared ID	97
6.1.3 Peripheral registers	97
6.1.4 Bit set and clear	97
6.1.5 Tasks	97
6.1.6 Events	97
6.1.7 Shortcuts	98
6.1.8 Interrupts	98
6.2 AAR — Accelerated address resolver	99
6.2.1 EasyDMA	99
6.2.2 Resolving a resolvable address	99
6.2.3 Use case example for chaining RADIO packet reception with address resolution using AAR	100
6.2.4 IRK data structure	100
6.2.5 Registers	100
6.2.6 Electrical specification	104
6.3 ACL — Access control lists	104
6.3.1 Registers	106
6.4 CCM — AES CCM mode encryption	108
6.4.1 Key-stream generation	108
6.4.2 Encryption	109
6.4.3 Decryption	109
6.4.4 AES CCM and RADIO concurrent operation	110
6.4.5 Encrypting packets on-the-fly in radio transmit mode	110
6.4.6 Decrypting packets on-the-fly in radio receive mode	111
6.4.7 CCM data structure	112
6.4.8 EasyDMA and ERROR event	113
6.4.9 Registers	113
6.4.10 Electrical specification	120
6.5 COMP — Comparator	120
6.5.1 Differential mode	122

6.5.2 Single-ended mode	122
6.5.3 Registers	124
6.5.4 Electrical specification	131
6.6 ECB — AES electronic codebook mode encryption	132
6.6.1 Shared resources	132
6.6.2 EasyDMA	132
6.6.3 ECB data structure	132
6.6.4 Registers	133
6.6.5 Electrical specification	135
6.7 EGU — Event generator unit	135
6.7.1 Registers	136
6.7.2 Electrical specification	138
6.8 GPIO — General purpose input/output	138
6.8.1 Pin configuration	139
6.8.2 Registers	141
6.8.3 Electrical specification	145
6.9 GPIOTE — GPIO tasks and events	146
6.9.1 Pin events and tasks	147
6.9.2 Port event	147
6.9.3 Tasks and events pin configuration	148
6.9.4 Registers	148
6.9.5 Electrical specification	152
6.10 I ² S — Inter-IC sound interface	152
6.10.1 Mode	153
6.10.2 Transmitting and receiving	153
6.10.3 Left right clock (LRCK)	154
6.10.4 Serial clock (SCK)	154
6.10.5 Master clock (MCK)	155
6.10.6 Width, alignment and format	155
6.10.7 EasyDMA	157
6.10.8 Module operation	159
6.10.9 Pin configuration	161
6.10.10 Registers	162
6.10.11 Electrical specification	171
6.11 LPCOMP — Low-power comparator	172
6.11.1 Shared resources	173
6.11.2 Pin configuration	173
6.11.3 Registers	174
6.11.4 Electrical specification	180
6.12 MWU — Memory watch unit	180
6.12.1 Registers	181
6.13 NFCT — Near field communication tag	194
6.13.1 Overview	195
6.13.2 Operating states	197
6.13.3 Pin configuration	198
6.13.4 EasyDMA	198
6.13.5 Frame assembler	199
6.13.6 Frame disassembler	200
6.13.7 Frame timing controller	201
6.13.8 Collision resolution	202
6.13.9 Antenna interface	203
6.13.10 NFCT antenna recommendations	203
6.13.11 Battery protection	204
6.13.12 Digital Modulation Signal	204

6.13.13	References	205
6.13.14	Registers	205
6.13.15	Electrical specification	224
6.14	PDM — Pulse density modulation interface	224
6.14.1	Master clock generator	225
6.14.2	Module operation	225
6.14.3	Decimation filter	226
6.14.4	EasyDMA	226
6.14.5	Hardware example	227
6.14.6	Pin configuration	228
6.14.7	Registers	228
6.14.8	Electrical specification	235
6.15	PPI — Programmable peripheral interconnect	235
6.15.1	Pre-programmed channels	237
6.15.2	Registers	237
6.16	PWM — Pulse width modulation	242
6.16.1	Wave counter	243
6.16.2	Decoder with EasyDMA	246
6.16.3	Limitations	253
6.16.4	Pin configuration	253
6.16.5	Registers	254
6.17	QDEC — Quadrature decoder	262
6.17.1	Sampling and decoding	263
6.17.2	LED output	264
6.17.3	Debounce filters	264
6.17.4	Accumulators	265
6.17.5	Output/input pins	265
6.17.6	Pin configuration	265
6.17.7	Registers	266
6.17.8	Electrical specification	277
6.18	RADIO — 2.4 GHz radio	277
6.18.1	Packet configuration	278
6.18.2	Address configuration	279
6.18.3	Data whitening	280
6.18.4	CRC	280
6.18.5	Radio states	281
6.18.6	Transmit sequence	281
6.18.7	Receive sequence	283
6.18.8	Received signal strength indicator (RSSI)	284
6.18.9	Interframe spacing (IFS)	284
6.18.10	Device address match	285
6.18.11	Bit counter	286
6.18.12	Direction finding	286
6.18.13	IEEE 802.15.4 operation	291
6.18.14	EasyDMA	300
6.18.15	Registers	301
6.18.16	Electrical specification	336
6.19	RNG — Random number generator	342
6.19.1	Bias correction	342
6.19.2	Speed	343
6.19.3	Registers	343
6.19.4	Electrical specification	345
6.20	RTC — Real-time counter	345
6.20.1	Clock source	346

6.20.2 Resolution versus overflow and the PRESCALER	346
6.20.3 COUNTER register	347
6.20.4 Overflow features	347
6.20.5 TICK event	347
6.20.6 Event control feature	348
6.20.7 Compare feature	348
6.20.8 TASK and EVENT jitter/delay	350
6.20.9 Reading the COUNTER register	352
6.20.10 Registers	353
6.20.11 Electrical specification	358
6.21 SAADC — Successive approximation analog-to-digital converter	358
6.21.1 Input configuration	359
6.21.2 Reference voltage and gain settings	361
6.21.3 Digital output	361
6.21.4 EasyDMA	361
6.21.5 Continuous sampling	363
6.21.6 Oversampling	363
6.21.7 Event monitoring using limits	363
6.21.8 Calibration	364
6.21.9 Registers	364
6.21.10 Electrical specification	379
6.22 SPI — Serial peripheral interface master	380
6.22.1 Functional description	380
6.22.2 Registers	383
6.22.3 Electrical specification	387
6.23 SPIM — Serial peripheral interface master with EasyDMA	388
6.23.1 SPI master transaction sequence	389
6.23.2 D/CX functionality	390
6.23.3 Pin configuration	391
6.23.4 EasyDMA	391
6.23.5 Low power	392
6.23.6 Registers	393
6.23.7 Electrical specification	404
6.24 SPIS — Serial peripheral interface slave with EasyDMA	405
6.24.1 Shared resources	406
6.24.2 EasyDMA	406
6.24.3 SPI slave operation	406
6.24.4 Pin configuration	408
6.24.5 Registers	409
6.24.6 Electrical specification	420
6.25 SWI — Software interrupts	422
6.25.1 Registers	422
6.26 TEMP — Temperature sensor	422
6.26.1 Registers	422
6.26.2 Electrical specification	428
6.27 TWI — I ² C compatible two-wire interface	429
6.27.1 Functional description	429
6.27.2 Master mode pin configuration	429
6.27.3 Shared resources	430
6.27.4 Master write sequence	430
6.27.5 Master read sequence	431
6.27.6 Master repeated start sequence	432
6.27.7 Low power	433
6.27.8 Registers	433

6.27.9 Electrical specification	441
6.28 TIMER — Timer/counter	441
6.28.1 Capture	443
6.28.2 Compare	443
6.28.3 Task delays	443
6.28.4 Task priority	443
6.28.5 Registers	443
6.29 TWIM — I ² C compatible two-wire interface master with EasyDMA	448
6.29.1 EasyDMA	449
6.29.2 Master write sequence	450
6.29.3 Master read sequence	451
6.29.4 Master repeated start sequence	452
6.29.5 Low power	453
6.29.6 Master mode pin configuration	453
6.29.7 Registers	453
6.29.8 Electrical specification	464
6.29.9 Pullup resistor	465
6.30 TWIS — I ² C compatible two-wire interface slave with EasyDMA	465
6.30.1 EasyDMA	468
6.30.2 TWI slave responding to a read command	468
6.30.3 TWI slave responding to a write command	469
6.30.4 Master repeated start sequence	470
6.30.5 Terminating an ongoing TWI transaction	471
6.30.6 Low power	471
6.30.7 Slave mode pin configuration	471
6.30.8 Registers	472
6.30.9 Electrical specification	482
6.31 UART — Universal asynchronous receiver/transmitter	482
6.31.1 Functional description	483
6.31.2 Pin configuration	483
6.31.3 Shared resources	484
6.31.4 Transmission	484
6.31.5 Reception	484
6.31.6 Suspending the UART	485
6.31.7 Error conditions	485
6.31.8 Using the UART without flow control	485
6.31.9 Parity and stop bit configuration	486
6.31.10 Registers	486
6.31.11 Electrical specification	495
6.32 UARTE — Universal asynchronous receiver/transmitter with EasyDMA	495
6.32.1 EasyDMA	496
6.32.2 Transmission	496
6.32.3 Reception	497
6.32.4 Error conditions	498
6.32.5 Using the UARTE without flow control	498
6.32.6 Parity and stop bit configuration	499
6.32.7 Low power	499
6.32.8 Pin configuration	499
6.32.9 Registers	499
6.32.10 Electrical specification	512
6.33 USB — Universal serial bus device	512
6.33.1 USB device states	513
6.33.2 USB terminology	514
6.33.3 USB pins	515

6.33.4	USB power-up sequence	515
6.33.5	USB pull-up	516
6.33.6	USB reset	516
6.33.7	USB suspend and resume	517
6.33.8	EasyDMA	518
6.33.9	Control transfers	519
6.33.10	Bulk and interrupt transactions	522
6.33.11	Isochronous transactions	525
6.33.12	USB register access limitations	527
6.33.13	Registers	528
6.33.14	Electrical specification	552
6.34	WDT — Watchdog timer	552
6.34.1	Reload criteria	553
6.34.2	Temporarily pausing the watchdog	553
6.34.3	Watchdog reset	553
6.34.4	Registers	553
6.34.5	Electrical specification	556
7	Hardware and layout	557
7.1	Pin assignments	557
7.1.1	aQFN73 ball assignments	557
7.1.2	QFN40 pin assignments	560
7.1.3	WLCSP ball assignments	562
7.2	Mechanical specifications	565
7.2.1	aQFN73 7 x 7 mm package	565
7.2.2	QFN40 5 x 5 mm package	566
7.2.3	WLCSP 3.175 x 3.175 mm package	566
7.3	Reference circuitry	567
7.3.1	Circuit configuration no. 1 for QIAA aQFN73	569
7.3.2	Circuit configuration no. 2 for QIAA aQFN73	570
7.3.3	Circuit configuration no. 3 for QIAA aQFN73	572
7.3.4	Circuit configuration no. 4 for QIAA aQFN73	574
7.3.5	Circuit configuration no. 5 for QIAA aQFN73	576
7.3.6	Circuit configuration no. 6 for QIAA aQFN73	578
7.3.7	Circuit configuration no. 1 for QDAA QFN40	580
7.3.8	Circuit configuration no. 2 for QDAA QFN40	582
7.3.9	Circuit configuration no. 3 for QDAA QFN40	584
7.3.10	Circuit configuration no. 4 for QDAA QFN40	586
7.3.11	Circuit configuration no. 5 for QDAA QFN40	588
7.3.12	Circuit configuration no. 6 for QDAA QFN40	590
7.3.13	Circuit configuration no. 1 for CJAA WLCSP	592
7.3.14	Circuit configuration no. 2 for CJAA WLCSP	594
7.3.15	Circuit configuration no. 3 for CJAA WLCSP	596
7.3.16	Circuit configuration no. 4 for CJAA WLCSP	598
7.3.17	Circuit configuration no. 5 for CJAA WLCSP	600
7.3.18	Circuit configuration no. 6 for CJAA WLCSP	602
7.3.19	PCB guidelines	604
7.3.20	PCB layout example	605
7.4	Package thermal characteristics	607
7.5	Package Variation	607
7.5.1	aQFN73	607
8	Recommended operating conditions	608

8.1 Extended Operating Temperature	608
8.2 WLCSP light sensitivity	608
9 Absolute maximum ratings.	609
10 Ordering information.	611
10.1 IC marking	611
10.2 Box labels	611
10.3 Order code	612
10.4 Code ranges and values	613
10.5 Product options	614
11 Legal notices.	616

1 Revision history

Date	Version	Description
August 2020	1.3	The following content has been added or updated: - Added WLCSP package information - Editorial changes
April 2020	1.2	The following content has been added or updated since the last released version: - Added information for the QFN40 package variant in Pin assignments on page 557, Mechanical specifications on page 565, Reference circuitry on page 567, Package thermal characteristics on page 607, Absolute maximum ratings on page 609 and Ordering information on page 611. - Corrected minimum valid value for EasyDMA MAXCNT and AMOUNT registers in SPIM — Serial peripheral interface master with EasyDMA on page 388, SPIS — Serial peripheral interface slave with EasyDMA on page 405, TWIM — I²C compatible two-wire interface master with EasyDMA on page 448, TWIS — I²C compatible two-wire interface slave with EasyDMA on page 465 and UARTE — Universal asynchronous receiver/transmitter with EasyDMA on page 495. Current consumption on page 52 - Added missing compounded currents. POWER — Power supply on page 58 - Clarified REG0 elspec parameters, by renaming and adding several parameters. RADIO — 2.4 GHz radio on page 277 - Corrected Sensitivity plot. SPIM — Serial peripheral interface master with EasyDMA on page 388 - Corrected parameter $t_{SPIM,CSK}$. Reference circuitry on page 567 - Added optional 4.7 Ω resistor to USB supply in configuration 1 for QIAA package. Recommended operating conditions on page 608 - Added parameter T_J (junction temperature), moved from Absolute maximum ratings on page 609. Absolute maximum ratings on page 609 - Increased aQFN73 CDM to 750 V. Removed parameter T_J (junction temperature), moved to Recommended operating conditions on page 608. Added footnote regarding supply voltages used in HTOL. Legal notices on page 616 - Updated copyright date.
January 2020	1.1	Not released
November 2019	1.0	First release

2 About this document

This document is organized into chapters that are based on the modules and peripherals available in the IC.

2.1 Document status

The document status reflects the level of maturity of the document.

Document name	Description
Objective Product Specification (OPS)	Applies to document versions up to 1.0. This document contains target specifications for product development.
Product Specification (PS)	Applies to document versions 1.0 and higher. This document contains final product specifications. Nordic Semiconductor ASA reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

Table 1: Defined document names

2.2 Peripheral chapters

Every peripheral has a unique capitalized name or an abbreviation of its name, e.g. TIMER, used for identification and reference. This name is used in chapter headings and references, and it will appear in the ARM® Cortex® Microcontroller Software Interface Standard (CMSIS) hardware abstraction layer to identify the peripheral.

The peripheral instance name, which is different from the peripheral name, is constructed using the peripheral name followed by a numbered postfix, starting with 0, for example, TIMER0. A postfix is normally only used if a peripheral can be instantiated more than once. The peripheral instance name is also used in the CMSIS to identify the peripheral instance.

The chapters describing peripherals may include the following information:

- A detailed functional description of the peripheral
- Register configuration for the peripheral
- Electrical specification tables, containing performance data which apply for the operating conditions described in [Recommended operating conditions](#) on page 608.

2.3 Register tables

Individual registers are described using register tables. These tables are built up of two sections. The first three colored rows describe the position and size of the different fields in the register. The following rows describe the fields in more detail.

2.3.1 Fields and values

The **Id (Field Id)** row specifies the bits that belong to the different fields in the register. If a field has enumerated values, then every value will be identified with a unique value id in the **Value Id** column.

A blank space means that the field is reserved and read as undefined, and it also must be written as 0 to secure forward compatibility. If a register is divided into more than one field, a unique field name is specified for each field in the **Field** column. The **Value Id** may be omitted in the single-bit bit fields when values can be substituted with a Boolean type enumerator range, e.g. true/false, disable(d)/enable(d), on/off, and so on.

Values are usually provided as decimal or hexadecimal. Hexadecimal values have a 0x prefix, decimal values have no prefix.

The **Value** column can be populated in the following ways:

- Individual enumerated values, for example 1, 3, 9.
- Range of values, e.g. [0..4], indicating all values from and including 0 and 4.
- Implicit values. If no values are indicated in the **Value** column, all bit combinations are supported, or alternatively the field's translation and limitations are described in the text instead.

If two or more fields are closely related, the **Value Id**, **Value**, and **Description** may be omitted for all but the first field. Subsequent fields will indicate inheritance with '..'.

A feature marked **Deprecated** should not be used for new designs.

2.3.2 Permissions

Different fields in a register might have different access permissions enforced by hardware.

The access permission for each register field is documented in the **Access** column in the following ways:

Access	Description	Hardware behavior
RO	Read-only	Field can only be read. A write will be ignored.
WO	Write-only	Field can only be written. A read will return an undefined value.
RW	Read-write	Field can be read and written multiple times.
W1	Write-once	Field can only be written once per reset. Any subsequent write will be ignored. A read will return an undefined value.
RW1	Read-write-once	Field can be read multiple times, but only written once per reset. Any subsequent write will be ignored.

Table 2: Register field permission schemes

2.4 Registers

Register	Offset	Description
DUMMY	0x514	Example of a register controlling a dummy feature

Table 3: Register overview

2.4.1 DUMMY

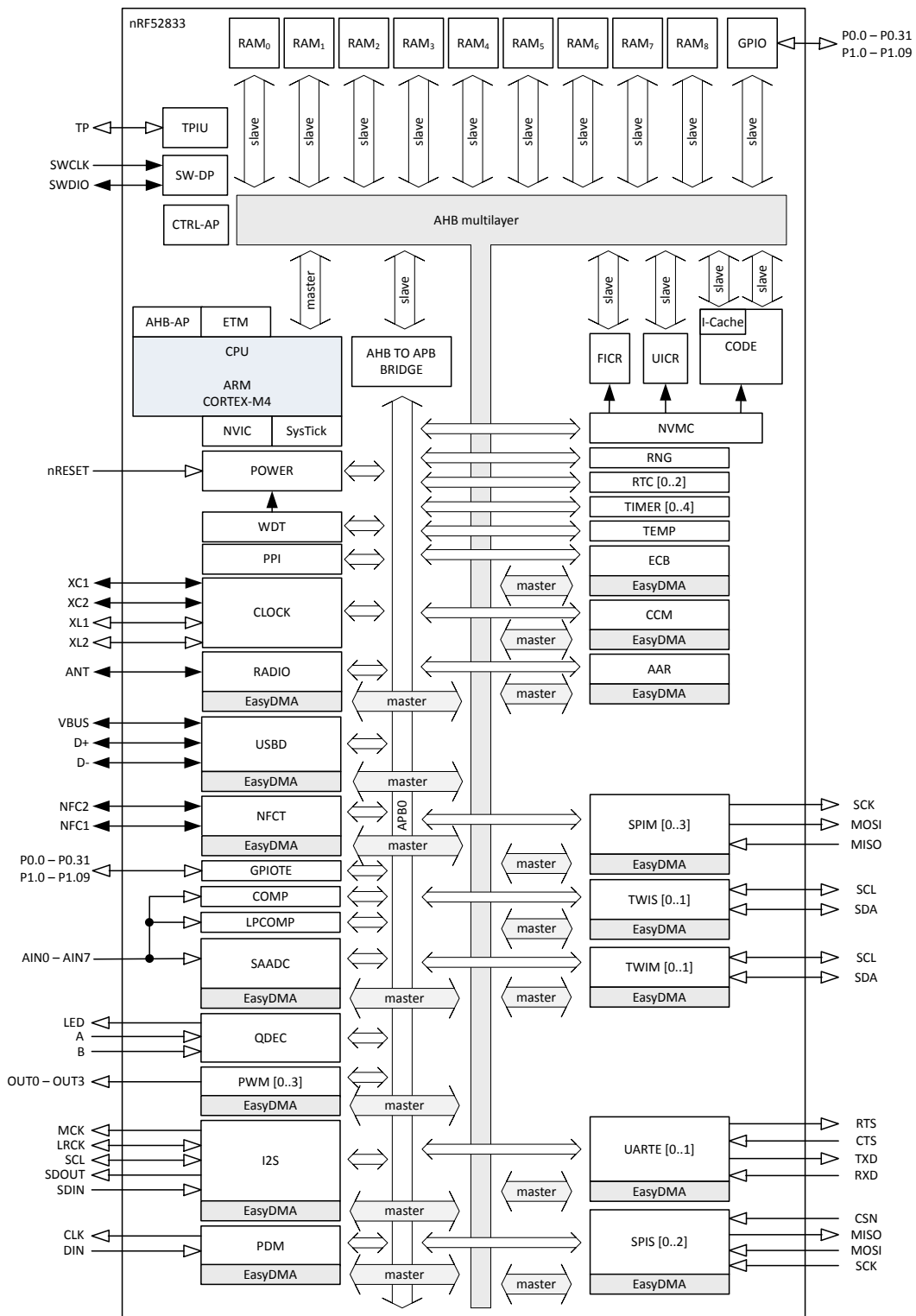
Address offset: 0x514

Example of a register controlling a dummy feature

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID								D	D	D	D					C	C	C					B									A	A						
Reset 0x00050002				0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0			
ID	Acce	Field	Value ID	Value	Description																																		
A	RW	FIELD_A			Example of a read-write field with several enumerated values																																		
			Disabled	0	The example feature is disabled																																		
			NormalMode	1	The example feature is enabled in normal mode																																		
			ExtendedMode	2	The example feature is enabled along with extra functionality																																		
B	RW	FIELD_B			Example of a deprecated read-write field																																		Deprecated
			Disabled	0	The override feature is disabled																																		
			Enabled	1	The override feature is enabled																																		
C	RW	FIELD_C			Example of a read-write field with a valid range of values																																		
			ValidRange	[2..7]	Example of allowed values for this field																																		
D	RW	FIELD_D			Example of a read-write field with no restriction on the values																																		

3 Block diagram

This block diagram illustrates the overall system. Arrows with white heads indicate signals that share physical pins with other signals.



4 Core components

4.1 CPU

The ARM[®] Cortex[®]-M4 processor with floating-point unit (FPU) has a 32-bit instruction set (Thumb[®]-2 technology) that implements a superset of 16- and 32-bit instructions to maximize code density and performance.

This processor implements the following features that enable energy-efficient arithmetic and high-performance signal processing.

- Digital signal processing (DSP) instructions
- Single-cycle multiply and accumulate (MAC) instructions
- Hardware divide
- 8- and 16-bit single instruction multiple data (SIMD) instructions
- Single-precision floating-point unit (FPU)

The ARM[®] Cortex[®] Microcontroller Software Interface Standard (CMSIS) hardware abstraction layer for the ARM[®] Cortex[®] processor series is implemented and available for the M4 CPU.

Real-time execution is highly deterministic in thread mode, to and from sleep modes, and when handling events at configurable priority levels via the nested vectored interrupt controller (NVIC).

Executing code from flash memory will have a wait state penalty on the nRF52 Series. An instruction cache can be enabled to minimize flash wait states when fetching instructions. For more information on cache, see [Cache](#) on page 25. The [Electrical specification](#) on page 19 shows CPU performance parameters including wait states in different modes, CPU current and efficiency, and processing power and efficiency based on the CoreMark[®] benchmark.

The ARM system timer (SysTick) is present on nRF52833. The SysTick's clock will only tick when the CPU is running or when the system is in debug interface mode.

4.1.1 Floating point interrupt

The floating point unit (FPU) may generate exceptions when used due to e.g. overflow or underflow, which in turn will trigger the FPU interrupt.

See [Instantiation](#) on page 22 for more information about the exceptions triggering the FPU interrupt.

To clear the IRQ (interrupt request) line when an exception has occurred, the relevant exception bit within the floating-point status and control register (FPSCR) needs to be cleared. For more information about the FPSCR or other FPU registers, see [Cortex-M4 Devices Generic User Guide](#).

4.1.2 CPU and support module configuration

The ARM[®] Cortex[®]-M4 processor has a number of CPU options and support modules implemented on the IC.

Option / Module	Description	Implemented
Core options		
NVIC	Nested vector interrupt controller	48 vectors
PRIORITIES	Priority bits	3
WIC	Wakeup interrupt controller	NO
Endianness	Memory system endianness	Little endian
Bit-banding	Bit banded memory	NO
DWT	Data watchpoint and trace	YES
SysTick	System tick timer	YES
Modules		
MPU	Memory protection unit	YES
FPU	Floating-point unit	YES
DAP	Debug access port	YES
ETM	Embedded trace macrocell	YES
ITM	Instrumentation trace macrocell	YES
TPIU	Trace port interface unit	YES
ETB	Embedded trace buffer	NO
FPB	Flash patch and breakpoint unit	YES
HTM	AMBA™ AHB trace macrocell	NO

4.1.3 Electrical specification

4.1.3.1 CPU performance

The CPU clock speed is 64 MHz. Current and efficiency data is taken when in System ON and the CPU is executing the CoreMark® benchmark. It includes power regulator and clock base currents. All other blocks are IDLE.

Symbol	Description	Min.	Typ.	Max.	Units
W _{FLASH}	CPU wait states, running CoreMark from flash, cache disabled			2	
W _{FLASHCACHE}	CPU wait states, running CoreMark from flash, cache enabled			3	
W _{RAM}	CPU wait states, running CoreMark from RAM			0	
CM _{FLASH}	CoreMark, running CoreMark from flash, cache enabled		217		Core [®]
CM _{FLASH/MHz}	CoreMark per MHz, running CoreMark from flash, cache enabled		3.4		CoreMark/ MHz
CM _{FLASH/mA}	CoreMark per mA, running CoreMark from flash, cache enabled, DCDC 3V		65.8		Core [®] mA

4.2 Memory

The nRF52833 contains 512 kB of flash memory and 128 kB of RAM that can be used for code and data storage.

The CPU and peripherals with EasyDMA can access memory via the AHB multilayer interconnect. In addition, peripherals are accessed by the CPU via the AHB multilayer interconnect, as shown in the following figure.

See [AHB multilayer](#) on page 46 and [EasyDMA](#) on page 44 for more information about the AHB multilayer interconnect and EasyDMA.

4.2.1 RAM - Random access memory

RAM AHB slaves 0 to 7 are connected to two 4 kB RAM sections each, while RAM AHB slave 8 is connected to two 32 kB sections, as shown in [Memory layout](#) on page 20.

4.2.2 Flash - Non-volatile memory

Writing to flash memory is managed by the non-volatile memory controller (NVMC), see [NVMC — Non-volatile memory controller](#) on page 23.

4.2.3 Memory map

4452_021 v1.3

20



NORDIC

 SEMICONDUCTOR

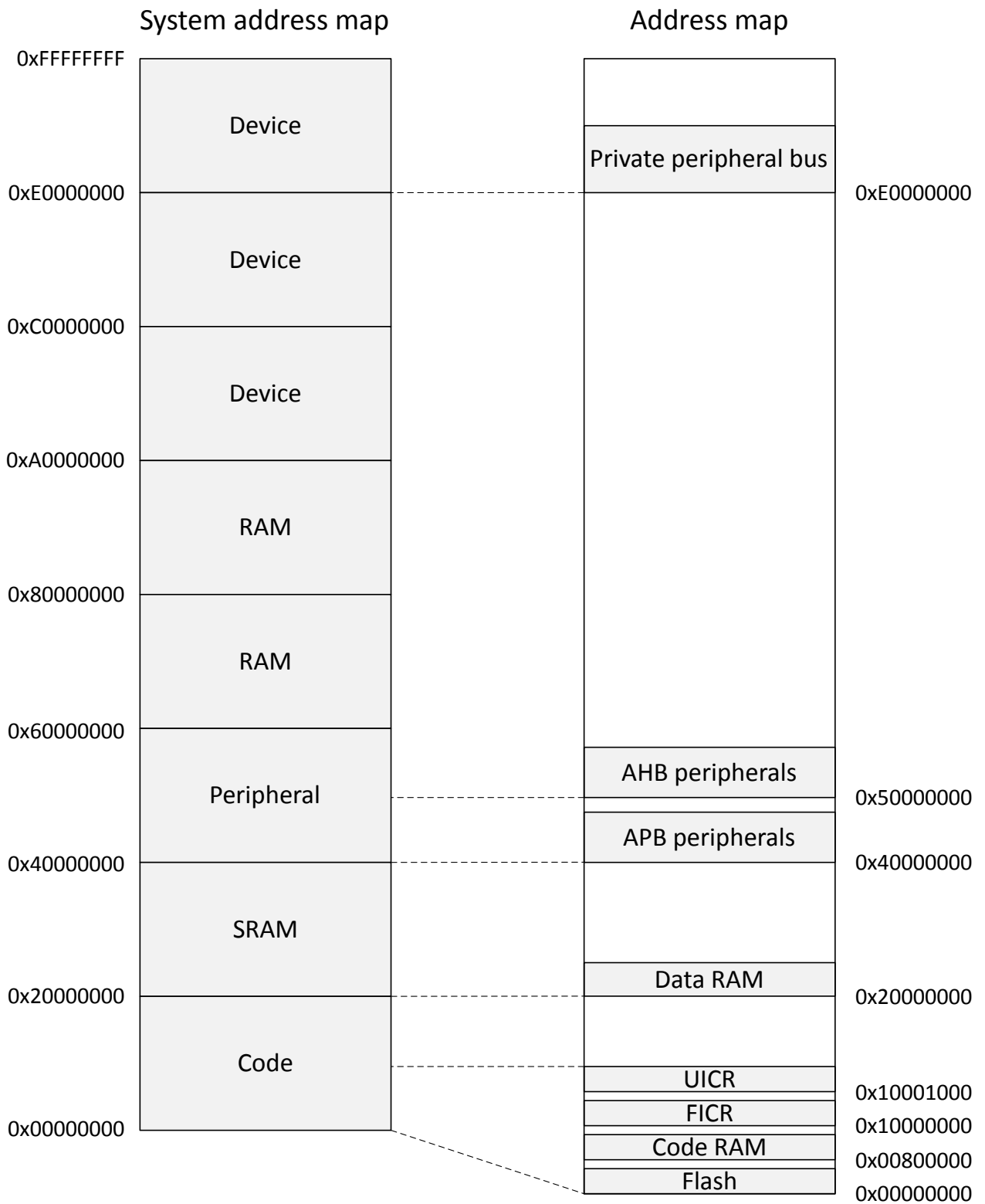


Figure 3: Memory map

4.2.4 Instantiation

ID	Base address	Peripheral	Instance	Description
0	0x40000000	CLOCK	CLOCK	Clock control
0	0x40000000	POWER	POWER	Power control
0	0x50000000	GPIO	GPIO	General purpose input and output Deprecated
0	0x50000000	GPIO	P0	General purpose input and output, port 0
0	0x50000300	GPIO	P1	General purpose input and output, port 1
1	0x40001000	RADIO	RADIO	2.4 GHz radio
2	0x40002000	UART	UART0	Universal asynchronous receiver/transmitter Deprecated
2	0x40002000	UARTE	UARTE0	Universal asynchronous receiver/transmitter with EasyDMA, unit 0
3	0x40003000	SPI	SPI0	SPI master 0 Deprecated
3	0x40003000	SPIM	SPIM0	SPI master 0
3	0x40003000	SPIS	SPIS0	SPI slave 0
3	0x40003000	TWI	TWI0	Two-wire interface master 0 Deprecated
3	0x40003000	TWIM	TWIM0	Two-wire interface master 0
3	0x40003000	TWIS	TWIS0	Two-wire interface slave 0
4	0x40004000	SPI	SPI1	SPI master 1 Deprecated
4	0x40004000	SPIM	SPIM1	SPI master 1
4	0x40004000	SPIS	SPIS1	SPI slave 1
4	0x40004000	TWI	TWI1	Two-wire interface master 1 Deprecated
4	0x40004000	TWIM	TWIM1	Two-wire interface master 1
4	0x40004000	TWIS	TWIS1	Two-wire interface slave 1
5	0x40005000	NFCT	NFCT	Near field communication tag
6	0x40006000	GPIONTE	GPIONTE	GPIO tasks and events
7	0x40007000	SAADC	SAADC	Analog to digital converter
8	0x40008000	TIMER	TIMER0	Timer 0
9	0x40009000	TIMER	TIMER1	Timer 1
10	0x4000A000	TIMER	TIMER2	Timer 2
11	0x4000B000	RTC	RTC0	Real-time counter 0
12	0x4000C000	TEMP	TEMP	Temperature sensor
13	0x4000D000	RNG	RNG	Random number generator
14	0x4000E000	ECB	ECB	AES electronic code book (ECB) mode block encryption
15	0x4000F000	AAR	AAR	Accelerated address resolver
15	0x4000F000	CCM	CCM	AES counter with CBC-MAC (CCM) mode block encryption
16	0x40010000	WDT	WDT	Watchdog timer
17	0x40011000	RTC	RTC1	Real-time counter 1
18	0x40012000	QDEC	QDEC	Quadrature decoder
19	0x40013000	COMP	COMP	General purpose comparator
19	0x40013000	LPCOMP	LPCOMP	Low power comparator
20	0x40014000	EGU	EGU0	Event generator unit 0
20	0x40014000	SWI	SWI0	Software interrupt 0
21	0x40015000	EGU	EGU1	Event generator unit 1
21	0x40015000	SWI	SWI1	Software interrupt 1
22	0x40016000	EGU	EGU2	Event generator unit 2
22	0x40016000	SWI	SWI2	Software interrupt 2
23	0x40017000	EGU	EGU3	Event generator unit 3
23	0x40017000	SWI	SWI3	Software interrupt 3
24	0x40018000	EGU	EGU4	Event generator unit 4
24	0x40018000	SWI	SWI4	Software interrupt 4
25	0x40019000	EGU	EGU5	Event generator unit 5
25	0x40019000	SWI	SWI5	Software interrupt 5

ID	Base address	Peripheral	Instance	Description
26	0x4001A000	TIMER	TIMER3	Timer 3
27	0x4001B000	TIMER	TIMER4	Timer 4
28	0x4001C000	PWM	PWM0	Pulse width modulation unit 0
29	0x4001D000	PDM	PDM	Pulse Density modulation (digital microphone) interface
30	0x4001E000	ACL	ACL	Access control lists
30	0x4001E000	NVMC	NVMC	Non-volatile memory controller
31	0x4001F000	PPI	PPI	Programmable peripheral interconnect
32	0x40020000	MWU	MWU	Memory watch unit
33	0x40021000	PWM	PWM1	Pulse width modulation unit 1
34	0x40022000	PWM	PWM2	Pulse width modulation unit 2
35	0x40023000	SPI	SPI2	SPI master 2
35	0x40023000	SPIM	SPIM2	SPI master 2
35	0x40023000	SPIS	SPIS2	SPI slave 2
36	0x40024000	RTC	RTC2	Real-time counter 2
37	0x40025000	I2S	I2S	Inter-IC sound interface
38	0x40026000	FPU	FPU	FPU interrupt
39	0x40027000	USB	USB	Universal serial bus device
40	0x40028000	UARTE	UARTE1	Universal asynchronous receiver/transmitter with EasyDMA, unit 1
45	0x4002D000	PWM	PWM3	Pulse width modulation unit 3
47	0x4002F000	SPIM	SPIM3	SPI master 3
N/A	0x10000000	FICR	FICR	Factory information configuration
N/A	0x10001000	UICR	UICR	User information configuration

Table 4: Instantiation table

4.3 NVMC — Non-volatile memory controller

The non-volatile memory controller (NVMC) is used for writing and erasing of the internal flash memory and the UICR (user information configuration registers).

The [CONFIG](#) on page 26 is used to enable the NVMC for writing (CONFIG.WEN = Wen) and erasing (CONFIG.WEN = Een).

The CPU must be halted before initiating a NVMC operation from the debug system.

4.3.1 Writing to flash

When write is enabled, full 32-bit words can be written to word-aligned addresses in flash memory.

As illustrated in [Memory](#) on page 19, the flash is divided into multiple pages. The same 32-bit word in flash memory can only be written n_{WRITE} number of times before a page erase must be performed.

The NVMC is only able to write 0 to bits in flash memory that are erased (set to 1). It cannot rewrite a bit back to 1. Only full 32-bit words can be written to flash memory using the NVMC interface. To write less than 32 bits, write the data as a full 32-bit word and set all the bits that should remain unchanged in the word to 1. The restriction on the number of writes (n_{WRITE}) still applies in this case.

Only word-aligned writes are allowed. Byte or half-word-aligned writes will result in a hard fault.

The time it takes to write a word to flash is specified by t_{WRITE} . The CPU is halted if the CPU executes code from the flash while the NVMC is writing to the flash.

NVM writing time can be reduced by using READYNEXT. If this status bit is set to 1, code can perform the next data write to the flash. This write will be buffered and will be taken into account as soon as the ongoing write operation is completed.

4.3.2 Erasing a page in flash

When erase is enabled, the flash memory can be erased page by page using the [ERASEPAGE](#) on page 27.

After erasing a flash page, all bits in the page are set to 1. The time it takes to erase a page is specified by $t_{\text{ERASEPAGE}}$. The CPU is halted if the CPU executes code from the flash while the NVMC is writing to the flash.

See [Partial erase of a page in flash](#) on page 25 for information on dividing the page erase time into shorter chunks.

4.3.3 Writing to user information configuration registers (UICR)

User information configuration registers (UICR) are written in the same way as flash. After UICR has been written, the new UICR configuration will only take effect after a reset.

UICR can only be written n_{WRITE} number of times before an erase must be performed using [ERASEUICR](#) on page 28 or [ERASEALL](#) on page 27. The time it takes to write a word to UICR is specified by t_{WRITE} . The CPU is halted if the CPU executes code from the flash while the NVMC is writing to the UICR.

4.3.4 Erasing user information configuration registers (UICR)

When erase is enabled, UICR can be erased using the [ERASEUICR](#) on page 28.

After erasing UICR, all bits in UICR are set to 1. The time it takes to erase UICR is specified by $t_{\text{ERASEPAGE}}$. The CPU is halted if the CPU executes code from the flash while the NVMC performs the erase operation.

4.3.5 Erase all

When erase is enabled, flash and UICR can be erased completely in one operation by using the [ERASEALL](#) on page 27. This operation will not erase the factory information configuration registers (FICR).

The time it takes to perform an [ERASEALL](#) command is specified by t_{ERASEALL} . The CPU is halted if the CPU executes code from the flash while the NVMC performs the erase operation.

4.3.6 Access port protection behavior

When access port protection is enabled, parts of the NVMC functionality will be blocked in order to prevent intentional or unintentional erase of UICR.

	CTRL-AP ERASEALL	NVMC ERASEPAGE	NVMC ERASEPAGE PARTIAL	NVMC ERASEALL	NVMC ERASEUICR
APPROTECT					
Disabled	Allowed	Allowed	Allowed	Allowed	Allowed
Enabled	Allowed	Allowed	Allowed	Allowed	Blocked

Table 5: NVMC Protection

4.3.7 NVMC power failure protection

NVMC power failure protection is possible through use of power-fail comparator that is monitoring power supply.

If the power-fail comparator is enabled, and the power supply voltage is below V_{POF} threshold, the power-fail comparator will prevent the NVMC from performing erase or write operations in non-volatile memory (NVM).

If a power failure warning is present at the start of an NVM erase operation, the NVMC operation will be ignored.

If a power failure warning is present at the start of an NVM write operation, the CPU will hardfault.

4.3.8 Partial erase of a page in flash

Partial erase is a feature in the NVMC to split a page erase time into shorter chunks to prevent longer CPU stalls in time-critical applications. Partial erase is only applicable to the code area in flash memory and does not work with UICR.

When erase is enabled, the partial erase of a flash page can be started by writing to [ERASEPAGEPARTIAL](#) on page 28. The duration of a partial erase can be configured in [ERASEPAGEPARTIALCFG](#) on page 28. A flash page is erased when its erase time reaches $t_{\text{ERASEPAGE}}$. Use [ERASEPAGEPARTIAL](#) N number of times so that $N * \text{ERASEPAGEPARTIALCFG} \geq t_{\text{ERASEPAGE}}$, where $N * \text{ERASEPAGEPARTIALCFG}$ gives the cumulative (total) erase time. Every time the cumulative erase time reaches $t_{\text{ERASEPAGE}}$, it counts as one erase cycle.

After the erase is complete, all bits in the page are set to 1. The CPU is halted if the CPU executes code from the flash while the NVMC performs the partial erase operation.

The bits in the page are undefined if the flash page erase is incomplete, i.e. if a partial erase has started but the total erase time is less than $t_{\text{ERASEPAGE}}$.

4.3.9 Cache

An instruction cache (I-Cache) can be enabled for the ICODE bus in the NVMC.

A cache hit is an instruction fetch from the cache, and it has a 0 wait-state delay. The number of wait-states for a cache miss, where the instruction is not available in the cache and needs to be fetched from flash, is shown in [CPU](#) on page 18.

Enabling the cache can increase CPU performance and reduce power consumption by reducing the number of wait cycles and the number of flash accesses. This will depend on the cache hit rate. Cache will use some current when enabled. If the reduction in average current due to reduced flash accesses is larger than the cache power requirement, the average current to execute the program code will decrease.

When disabled, the cache does not use current and does not retain its content.

It is possible to enable cache profiling to analyze the performance of the cache for your program using the [ICACHECNF](#) register. When profiling is enabled, the [IHIT](#) and [IMISS](#) registers are incremented for every instruction cache hit or miss, respectively. The hit and miss profiling registers do not wrap around after reaching the maximum value. If the maximum value is reached, consider profiling for a shorter duration to get correct numbers.

4.3.10 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4001E000	NVMC	NVMC	Non-volatile memory controller	

Table 6: Instances

Register	Offset	Description	
READY	0x400	Ready flag	
READYNEXT	0x408	Ready flag	
CONFIG	0x504	Configuration register	
ERASEPAGE	0x508	Register for erasing a page in code area	
ERASEPCR1	0x508	Register for erasing a page in code area, equivalent to ERASEPAGE	Deprecated
ERASEALL	0x50C	Register for erasing all non-volatile user memory	
ERASEPCR0	0x510	Register for erasing a page in code area, equivalent to ERASEPAGE	Deprecated
ERASEUICR	0x514	Register for erasing user information configuration registers	

Register	Offset	Description
ERASEPAGEPARTIAL	0x518	Register for partial erase of a page in code area
ERASEPAGEPARTIALCFG	0x51C	Register for partial erase configuration
ICACHECNF	0x540	I-code cache configuration register
IHIT	0x548	I-code cache hit counter
IMISS	0x54C	I-code cache miss counter

Table 7: Register overview

4.3.10.1 READY

Address offset: 0x400

Ready flag

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000001				0 0																															

4.3.10.2 READYNEXT

Address offset: 0x408

Ready flag

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A																															
Reset 0x00000001				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
ID	Acce Field		Value ID	Value		Description																													
A	R	READYNEXT				NVMC can accept a new write operation																													
			Busy	0		NVMC cannot accept any write operation																													
			Ready	1		NVMC is ready																													

4.3.10.3 CONFIG

Address offset: 0x504

Configuration register

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
ID																																						A	A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	Acce Field	Value ID	Value	Description																																			
A	RW	WEN		Program memory access mode. It is strongly recommended to only activate erase and write modes when they are actively used.																																			
				Enabling write or erase will invalidate the cache and keep it invalidated.																																			
		Ren	0	Read only access																																			
		Wen	1	Write enabled																																			
		Een	2	Erase enabled																																			

4.3.10.4 ERASEPAGE

Address offset: 0x508

Register for erasing a page in code area

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A			
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
ID	Acce Field	Value ID	Value				Description																														
A	W	ERASEPAGE					Register for starting erase of a page in code area																														
							The value is the address to the page to be erased.																														
							(Addresses of first word in page). The erase must be enabled using CONFIG.WEN before the page can be erased.																														
							Attempts to erase pages that are outside the code area may result in undesirable behavior, e.g. the wrong page may be erased.																														

4.3.10.5 ERASEPCR1 (Deprecated)

Address offset: 0x508

Register for erasing a page in code area, equivalent to ERASEPAGE

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																															
A	W	ERASEPCR1		Register for erasing a page in code area, equivalent to ERASEPAGE																															

4.3.10.6 ERASEALL

Address offset: 0x50C

Register for erasing all non-volatile user memory

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acces Field	Value ID	Value	Description																														
A	W	ERASEALL		Erase all non-volatile memory including UICR registers. The erase must be enabled using CONFIG.WEN before the non-volatile memory can be erased.																														
		NoOperation	0	No operation																														
		Erase	1	Start chip erase																														

4.3.10.7 ERASEPCR0 (Deprecated)

Address offset: 0x510

Register for erasing a page in code area, equivalent to ERASEPAGE

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID		Value				Description																											
A	W	ERASEPCRO						Register for starting erase of a page in code area, equivalent to ERASEPAGE																											

4.3.10.8 ERASEUICR

Address offset: 0x514

Register for erasing user information configuration registers

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
ID																																						A	
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
ID	Acce Field	Value ID	Value		Description																																		
A	W	ERASEUICR			Register starting erase of all user information configuration registers. The erase must be enabled using CONFIG.WEN before the UICR can be erased.																																		
		NoOperation	0		No operation																																		
		Erase	1		Start erase of UICR																																		

4.3.10.9 ERASEPAGEPARTIAL

Address offset: 0x518

Register for partial erase of a page in code area

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value																Description															
A	W	ERASEPAGEPARTIAL																	Register for starting partial erase of a page in code area The value is the address to the page to be partially erased (address of the first word in page). The erase must be enabled using CONFIG.WEN before every erase page partial and disabled using CONFIG.WEN after every erase page partial. Attempts to erase pages that are outside the code area may result in undesirable behavior, e.g. the wrong page may be erased.															

4.3.10.10 ERASEPAGEPARTIALCFG

Address offset: 0x51C

Register for partial erase configuration

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			B																								A							
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value		Description																													
A	RW	CACHEEN			Cache enable																													
		Disabled	0	Disable cache. Invalidates all cache entries.																														
		Enabled	1	Enable cache																														
B	RW	CACHEPROFEN			Cache profiling enable																													
		Disabled	0	Disable cache profiling																														
		Enabled	1	Enable cache profiling																														

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A			
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
ID	Acce	Field	Value	ID	Value	Description																																
A	RW	HITS				Number of cache hits.																																
						Register is writable, but only to '0'.																																

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value				Description																											
A	RW	MISSES					Number of cache misses.																											
				Register is writable, but only to '0'.																														

4.3.11 Electrical specification

4.3.11.1 Flash programming

Symbol	Description	Min.	Typ.	Max.	Units
n_{WRITE}	Number of times a 32-bit word can be written before erase			2	
$n_{\text{ENDURANCE}}$	Erase cycles per page	10000			
t_{WRITE}	Time to write one 32-bit word			42.5 ¹	μs
$t_{\text{ERASEPAGE}}$	Time to erase one page			87.5 ¹	ms
t_{ERASEALL}	Time to erase all flash			173 ¹	ms
$t_{\text{ERASEPAGEPARTIAL,acc}}$	Accuracy of the partial page erase duration. Total execution time for one partial page erase is defined as $\text{ERASEPAGEPARTIALCFG} * t_{\text{ERASEPAGEPARTIAL,acc}}$			1.09 ¹	

4.3.11.2 Cache size

Symbol	Description	Min.	Typ.	Max.	Units
$\text{Size}_{\text{ICODE}}$	I-Code cache size		2048		Bytes

4.4 FICR — Factory information configuration registers

Factory information configuration registers (FICR) are pre-programmed in factory and cannot be erased by the user. These registers contain chip-specific information and configuration.

4.4.1 Registers

Base address	Peripheral	Instance	Description	Configuration
0x10000000	FICR	FICR	Factory information configuration	

Table 8: Instances

Register	Offset	Description
CODEPAGESIZE	0x010	Code memory page size
CODESIZE	0x014	Code memory size
DEVICEID[0]	0x060	Device identifier
DEVICEID[1]	0x064	Device identifier
ER[0]	0x080	Encryption root, word 0
ER[1]	0x084	Encryption root, word 1
ER[2]	0x088	Encryption root, word 2
ER[3]	0x08C	Encryption root, word 3
IR[0]	0x090	Identity Root, word 0
IR[1]	0x094	Identity Root, word 1
IR[2]	0x098	Identity Root, word 2
IR[3]	0x09C	Identity Root, word 3
DEVICEADDRTYPE	0x0A0	Device address type
DEVICEADDR[0]	0x0A4	Device address 0
DEVICEADDR[1]	0x0A8	Device address 1
INFO.PART	0x100	Part code

¹ Applies when HFXO is used. Timing varies according to HFINT accuracy when HFINT is used.

Register	Offset	Description
INFO.VARIANT	0x104	Build code (hardware version and production configuration)
INFO.PACKAGE	0x108	Package option
INFO.RAM	0x10C	RAM variant
INFO.FLASH	0x110	Flash variant
INFO.UNUSED8[0]	0x114	Reserved
INFO.UNUSED8[1]	0x118	Reserved
INFO.UNUSED8[2]	0x11C	Reserved
PRODTTEST[0]	0x350	Production test signature 0
PRODTTEST[1]	0x354	Production test signature 1
PRODTTEST[2]	0x358	Production test signature 2
TEMP.A0	0x404	Slope definition A0
TEMP.A1	0x408	Slope definition A1
TEMP.A2	0x40C	Slope definition A2
TEMP.A3	0x410	Slope definition A3
TEMP.A4	0x414	Slope definition A4
TEMP.A5	0x418	Slope definition A5
TEMP.B0	0x41C	Y-intercept B0
TEMP.B1	0x420	Y-intercept B1
TEMP.B2	0x424	Y-intercept B2
TEMP.B3	0x428	Y-intercept B3
TEMP.B4	0x42C	Y-intercept B4
TEMP.B5	0x430	Y-intercept B5
TEMP.T0	0x434	Segment end T0
TEMP.T1	0x438	Segment end T1
TEMP.T2	0x43C	Segment end T2
TEMP.T3	0x440	Segment end T3
TEMP.T4	0x444	Segment end T4
NFC.TAGHEADER0	0x450	Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST.
NFC.TAGHEADER1	0x454	Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST.
NFC.TAGHEADER2	0x458	Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST.
NFC.TAGHEADER3	0x45C	Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST.

Table 9: Register overview

4.4.1.1 CODEPAGESIZE

Address offset: 0x010

Code memory page size

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce	Field	Value	ID	Value	Description																													
A	R	CODEPAGESIZE	Code memory page size																																

4.4.1.2 CODESIZE

Address offset: 0x014

Code memory size

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A				
Reset 0xFFFFFFFF			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
ID	Acce	Field	Value ID			Value			Description																													
A	R	CODESIZE						Code memory size in number of pages																														
																								Total code space is: CODEPAGESIZE * CODESIZE														

4.4.1.3 DEVICEID[n] (n=0..1)

Address offset: 0x060 + (n × 0x4)

Device identifier

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A																															

4.4.1.4 ER[n] (n=0..3)

Address offset: 0x080 + (n × 0x4)

Encryption root, word n

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field			Value ID			Value			Description																									
A	R	ER					Encryption root, word n																												

4.4.1.5 IR[n] (n=0..3)

Address offset: 0x090 + (n × 0x4)

Identity Root, word n

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID										A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
Reset 0xFFFFFFFF										1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field			Value ID			Value					Description																																
A	R	IR										Identity Root, word n																																

4.4.1.6 DEVICEADDRTYPE

Address offset: 0x0A0

Device address type

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID																																								A				
Reset 0xFFFFFFFF										1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field		Value ID		Value		Description																																					
A	R	DEVICEADDRTYPE					Device address type																																					
				Public		0	Public address																																					
				Random		1	Random address																																					

4.4.1.7 DEVICEADDR[n] (n=0..1)

Address offset: 0x0A4 + (n × 0x4)

Device address n

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
ID	Acce Field	Value ID	Value				Description																											
A	R	DEVICEADDR				48 bit device address																												
				DEVICEADDR[0] contains the least significant bits of the device address. DEVICEADDR[1] contains the most significant bits of the device address. Only bits [15:0] of DEVICEADDR[1] are used.																														

4.4.1.8 INFO.PART

Address offset: 0x100

Part code

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID										A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A		
Reset 0x00052833										0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	1	1
ID	Acce Field		Value ID		Value					Description																																		
A	R	PART							Part code																																			
				N52833		0x52833					nRF52833																																	
				N52840		0x52840					nRF52840																																	
				Unspecified		0xFFFFFFFF					Unspecified																																	

4.4.1.9 INFO.VARIANT

Address offset: 0x104

Build code (hardware version and production configuration)

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field	Value ID	Value				Description																											
A	R	VARIANT					Build code (hardware version and production configuration). Encoded as ASCII.																											
		AAAA	0x41414141				AAAA																											
		AAAB	0x41414142				AAAB																											
		Unspecified	0xFFFFFFFF				Unspecified																											

Address offset: 0x108

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
ID	Acce	Field	Value ID	Value	Description																													
A	R	PACKAGE			Package option																													
			QD	0x2007	QDxx - 40-pin QFN																													
			QI	0x2004	QIxx - 73-pin aQFN																													
			CJ	0x2008	CJxx - WLCSP																													
			Unspecified	0xFFFFFFFF	Unspecified																													

Address offset: 0x10C

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A																															
Reset 0xFFFFFFFF				1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1																															
ID	Acce Field			Value ID	Value		Description																												
A	R	RAM					RAM variant																												
				K16	0x10		16 kByte RAM																												
				K32	0x20		32 kByte RAM																												
				K64	0x40		64 kByte RAM																												
				K128	0x80		128 kByte RAM																												
				K256	0x100		256 kByte RAM																												
				Unspecified	0xFFFFFFFF		Unspecified																												

Address offset: 0x110

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field			Value ID	Value		Description																												
A	R	FLASH			Flash variant																														
				K128	0x80		128 kByte FLASH																												
				K256	0x100		256 kByte FLASH																												
				K512	0x200		512 kByte FLASH																												
				K1024	0x400		1 MByte FLASH																												
				K2048	0x800		2 MByte FLASH																												
				Unspecified	0xFFFFFFFF		Unspecified																												

Address offset: $0x350 + (n \times 0x4)$

Production test signature n

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A																															
Reset 0xFFFFFFFF			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1																															
ID	Acce Field	Value ID	Value				Description																											
A	R	PRODTST					Production test signature n																											
		Done	0xBB42319F				Production tests done																											
		NotDone	0xFFFFFFFF				Production tests not done																											

4.4.1.14 TEMP.A0

Address offset: 0x404

Slope definition A0

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A																															
Reset 0xFFFFFFFF			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1																															
ID	Acce Field	Value ID	Value	Description																														
A	R	A	A (slope definition) register.																															

4.4.1.15 TEMP.A1

Address offset: 0x408

Slope definition A1

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																															
ID																											A								A								A							
Reset 0xFFFFFFFF			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1																																															
ID	Acce Field	Value ID	Value																Description																															
A	R	A																	A (slope definition) register.																															

4.4.1.16 TEMP.A2

Address offset: 0x40C

Slope definition A2

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A																															

4.4.1.17 TEMP.A3

Address offset: 0x410

Slope definition A3

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
ID																								A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						
ID	Acce Field		Value ID	Value				Description																																		
A	R	A						A (slope definition) register.																																		

4.4.1.18 TEMP.A4

Address offset: 0x414

Slope definition A4

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A A																															

4.4.1.19 TEMP.A5

Address offset: 0x418

Slope definition A5

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
ID																												A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1											
ID	Acce Field		Value ID	Value		Description																																									
A	R	A			A (slope definition) register.																																										

4.4.1.20 TEMP.B0

Address offset: 0x41C

Y-intercept B0

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
ID																				A										A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF										1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
ID	Acce Field			Value ID			Value			Description																																						
A	R	B				B (y-intercept)																																										

4.4.1.21 TEMP.B1

Address offset: 0x420

Y-intercept B1

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A																															
Reset 0xFFFFFFFF			1 1																															

4.4.1.22 TEMP.B2

Address offset: 0x424

Y-intercept B2

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
ID																				A										A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF										1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
ID	Acce Field			Value ID			Value			Description																																							
A	R	B								B (y-intercept)																																							

4.4.1.23 TEMP.B3

Address offset: 0x428

Y-intercept B3

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
ID																														A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF										1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
ID	Acce Field			Value ID			Value			Description																																							
A	R	B								B (y-intercept)																																							

4.4.1.24 TEMP.B4

Address offset: 0x42C

Y-intercept B4

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
ID																				A										A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF										1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
ID	Acce Field			Value ID			Value			Description																																							
A	R	B								B (y-intercept)																																							

4.4.1.25 TEMP.B5

Address offset: 0x430

Y-intercept B5

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
ID																				A										A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF										1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
ID	Acce Field			Value ID			Value			Description																																							
A	R	B								B (y-intercept)																																							

4.4.1.26 TEMP.T0

Address offset: 0x434

Segment end T0

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID																												A	A	A	A	A	A	A	A								
Reset 0xFFFFFFFF										1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field			Value ID			Value			Description																																	
A	R	T								T (segment end) register																																	

4.4.1.27 TEMP.T1

Address offset: 0x438

Segment end T1

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A A A A A A																															
Reset 0xFFFFFFFF			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1																															
ID	Acce Field	Value ID	Value				Description																											
A	R	T	T (segment end) register																															

4.4.1.28 TEMP.T2

Address offset: 0x43C

Segment end T2

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																														
ID			A A																														

4.4.1.29 TEMP.T3

Address offset: 0x440

Segment end T3

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																		
ID				A A A A A A A A																																		
Reset 0xFFFFFFFF				1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1																																		
ID	Acce Field			Value ID			Value				Description																											
A	R	T									T (segment end) register																											

4.4.1.30 TEMP.T4

Address offset: 0x444

Segment end T4

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID		A A A A A A A A																															
Reset 0xFFFFFFFF		1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1																															
ID	Acce Field	Value ID		Value				Description																									
A	R	T	T (segment end) register																														

4.4.1.31 NFC.TAGHEADER0

Address offset: 0x450

Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				D D D D D D D D C C C C C C C C B B B B B B B B A A A A A A A A																															
Reset 0xFFFFF5F				1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1																															
ID	Acce	Field	Value	ID	Value	Description																													
A	R	MFGID				Default Manufacturer ID: Nordic Semiconductor ASA has ICM 0x5F																													
B	R	UD1				Unique identifier byte 1																													
C	R	UD2				Unique identifier byte 2																													
D	R	UD3				Unique identifier byte 3																													

4.4.1.32 NFC.TAGHEADER1

Address offset: 0x454

Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST.

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																			
ID		D	D	D	D	D	D	D	D	C	C	C	C	C	C	C	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field	Value ID		Value		Description																															
A-D	R	UD[i] (i=4..7)				Unique identifier byte i																															

4.4.1.33 NFC.TAGHEADER2

Address offset: 0x458

Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST.

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																			
ID		D	D	D	D	D	D	D	D	C	C	C	C	C	C	C	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field	Value ID		Value		Description																															
A-D	R	UD[i] (i=8..11)				Unique identifier byte i																															

4.4.1.34 NFC.TAGHEADER3

Address offset: 0x45C

Default header for NFC tag. Software can read these values to populate NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST.

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																			
ID		D	D	D	D	D	D	D	D	C	C	C	C	C	C	C	B	B	B	B	B	B	B	B	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field	Value ID		Value		Description																															
A-D	R	UD[i] (i=12..15)				Unique identifier byte i																															

4.5 UICR — User information configuration registers

The user information configuration registers (UICRs) are non-volatile memory (NVM) registers for configuring user-specific settings.

For information on writing UICR registers, see the [NVMC — Non-volatile memory controller](#) on page 23 and [Memory](#) on page 19 chapters.

4.5.1 Registers

Base address	Peripheral	Instance	Description	Configuration
0x10001000	UICR	UICR	User information configuration	

Table 10: Instances

Register	Offset	Description	
UNUSED0	0x000		Reserved
UNUSED1	0x004		Reserved
UNUSED2	0x008		Reserved
UNUSED3	0x010		Reserved
NRFFW[0]	0x014	Reserved for Nordic firmware design	
NRFFW[1]	0x018	Reserved for Nordic firmware design	
NRFFW[2]	0x01C	Reserved for Nordic firmware design	
NRFFW[3]	0x020	Reserved for Nordic firmware design	
NRFFW[4]	0x024	Reserved for Nordic firmware design	
NRFFW[5]	0x028	Reserved for Nordic firmware design	
NRFFW[6]	0x02C	Reserved for Nordic firmware design	
NRFFW[7]	0x030	Reserved for Nordic firmware design	
NRFFW[8]	0x034	Reserved for Nordic firmware design	
NRFFW[9]	0x038	Reserved for Nordic firmware design	
NRFFW[10]	0x03C	Reserved for Nordic firmware design	
NRFFW[11]	0x040	Reserved for Nordic firmware design	
NRFFW[12]	0x044	Reserved for Nordic firmware design	
NRFBW[0]	0x050	Reserved for Nordic hardware design	
NRFBW[1]	0x054	Reserved for Nordic hardware design	
NRFBW[2]	0x058	Reserved for Nordic hardware design	
NRFBW[3]	0x05C	Reserved for Nordic hardware design	
NRFBW[4]	0x060	Reserved for Nordic hardware design	
NRFBW[5]	0x064	Reserved for Nordic hardware design	
NRFBW[6]	0x068	Reserved for Nordic hardware design	
NRFBW[7]	0x06C	Reserved for Nordic hardware design	
NRFBW[8]	0x070	Reserved for Nordic hardware design	
NRFBW[9]	0x074	Reserved for Nordic hardware design	
NRFBW[10]	0x078	Reserved for Nordic hardware design	
NRFBW[11]	0x07C	Reserved for Nordic hardware design	
CUSTOMER[0]	0x080	Reserved for customer	
CUSTOMER[1]	0x084	Reserved for customer	
CUSTOMER[2]	0x088	Reserved for customer	
CUSTOMER[3]	0x08C	Reserved for customer	
CUSTOMER[4]	0x090	Reserved for customer	
CUSTOMER[5]	0x094	Reserved for customer	
CUSTOMER[6]	0x098	Reserved for customer	

Register	Offset	Description
CUSTOMER[7]	0x09C	Reserved for customer
CUSTOMER[8]	0x0A0	Reserved for customer
CUSTOMER[9]	0x0A4	Reserved for customer
CUSTOMER[10]	0x0A8	Reserved for customer
CUSTOMER[11]	0x0AC	Reserved for customer
CUSTOMER[12]	0x0B0	Reserved for customer
CUSTOMER[13]	0x0B4	Reserved for customer
CUSTOMER[14]	0x0B8	Reserved for customer
CUSTOMER[15]	0x0BC	Reserved for customer
CUSTOMER[16]	0x0C0	Reserved for customer
CUSTOMER[17]	0x0C4	Reserved for customer
CUSTOMER[18]	0x0C8	Reserved for customer
CUSTOMER[19]	0x0CC	Reserved for customer
CUSTOMER[20]	0x0D0	Reserved for customer
CUSTOMER[21]	0x0D4	Reserved for customer
CUSTOMER[22]	0x0D8	Reserved for customer
CUSTOMER[23]	0x0DC	Reserved for customer
CUSTOMER[24]	0x0E0	Reserved for customer
CUSTOMER[25]	0x0E4	Reserved for customer
CUSTOMER[26]	0x0E8	Reserved for customer
CUSTOMER[27]	0x0EC	Reserved for customer
CUSTOMER[28]	0x0F0	Reserved for customer
CUSTOMER[29]	0x0F4	Reserved for customer
CUSTOMER[30]	0x0F8	Reserved for customer
CUSTOMER[31]	0x0FC	Reserved for customer
PSELRESET[0]	0x200	Mapping of the nRESET function (see POWER chapter for details)
PSELRESET[1]	0x204	Mapping of the nRESET function (see POWER chapter for details)
APPROTECT	0x208	Access port protection
NFCPINS	0x20C	Setting of pins dedicated to NFC functionality: NFC antenna or GPIO
DEBUGCTRL	0x210	Processor debug control
REGOUT0	0x304	Output voltage from REG0 regulator stage. The maximum output voltage from this stage is given as VDDH - VREGODROP.

Table 11: Register overview

4.5.1.1 NRFFW[n] (n=0..12)

Address offset: $0x014 + (n \times 0x4)$

Reserved for Nordic firmware design

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field		Value	ID	Value		Description																												
A	RW NRFFW						Reserved for Nordic firmware design																												

4.5.1.2 NRFHW[n] (n=0..11)

Address offset: $0x050 + (n \times 0x4)$

Reserved for Nordic hardware design

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field		Value ID	Value				Description																											
A	RW NRFHW			Reserved for Nordic hardware design																															

4.5.1.3 CUSTOMER[n] (n=0..31)

Address offset: 0x080 + (n × 0x4)

Reserved for customer

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field	Value ID	Value				Description																											
A	RW	CUSTOMER					Reserved for customer																											

4.5.1.4 PSELRESET[n] (n=0..1)

Address offset: 0x200 + (n × 0x4)

Mapping of the nRESET function (see POWER chapter for details)

All PSELRESET registers have to contain the same value for a pin mapping to be valid. If values are not the same, there will be no nRESET function exposed on a GPIO. As a result, the device will always start independently of the levels present on any of the GPIOs.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																		
ID			C																								B												A	A	A	A	A																									
Reset 0xFFFFFFFF			1																																1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1							
ID	Acce	Field	Value	ID	Value																																Description																															
A	RW	PIN	18																																	GPIO pin number onto which nRESET is exposed																																
B	RW	PORT	0																																	Port number onto which nRESET is exposed																																
C	RW	CONNECT																																	Connection																																	
		Disconnected	1																																	Disconnect																																
		Connected	0																																	Connect																																

4.5.1.5 APPROTECT

Address offset: 0x208

Access port protection

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID																											A	A	A	A	A	A	A	A	A
Reset 0xFFFFFFFF			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
ID	Acce Field	Value ID	Value		Description																														
A	RW	PALL			Enable or disable access port protection.																														
					See Debug and trace on page 47 for more information.																														
		Disabled	0xFF		Disable																														
		Enabled	0x00		Enable																														

See [Debug and trace](#) on page 47 for more information.

4.5.1.6 NFCPINS

Address offset: 0x20C

Setting of pins dedicated to NFC functionality: NFC antenna or GPIO

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0xFFFFFFFF			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1																															
ID	Acce Field	Value ID	Value	Description																														
A	RW PROTECT			Setting of pins dedicated to NFC functionality																														
		Disabled	0	Operation as GPIO pins. Same protection as normal GPIO pins																														
		NFC	1	Operation as NFC antenna pins. Configures the protection for NFC operation																														

4.5.1.7 DEBUGCTRL

Address offset: 0x210

Processor debug control

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			B B B B B B B B A A A A A A A A																															
Reset 0xFFFFFFFF			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1																															
ID	Acce Field	Value ID	Value	Description																														
A	RW CPUNIDEN			Configure CPU non-intrusive debug features																														
		Enabled	0xFF	Enable CPU ITM and ETM functionality (default behavior)																														
		Disabled	0x00	Disable CPU ITM and ETM functionality																														
B	RW CPUFPBEN			Configure CPU flash patch and breakpoint (FPB) unit behavior																														
		Enabled	0xFF	Enable CPU FPB unit (default behavior)																														
		Disabled	0x00	Disable CPU FPB unit. Writes into the FPB registers will be ignored.																														

4.5.1.8 REGOUT0

Address offset: 0x304

Output voltage from REG0 regulator stage. The maximum output voltage from this stage is given as VDDH - VREG0DROP.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A																															
Reset 0xFFFFFFFF			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1																															
ID	Acce Field	Value ID	Value	Description																														
A	RW VOUT			Output voltage from REG0 regulator stage.																														
		1V8	0	1.8 V																														
		2V1	1	2.1 V																														
		2V4	2	2.4 V																														
		2V7	3	2.7 V																														
		3V0	4	3.0 V																														
		3V3	5	3.3 V																														
		DEFAULT	7	Default voltage: 1.8 V																														

4.6 EasyDMA

EasyDMA is a module implemented by some peripherals to gain direct access to Data RAM.

EasyDMA is an AHB bus master similar to CPU and is connected to the AHB multilayer interconnect for direct access to Data RAM. EasyDMA is not able to access flash.

A peripheral can implement multiple EasyDMA instances to provide dedicated channels. For example, for reading and writing of data between the peripheral and RAM. This concept is illustrated in [EasyDMA example](#) on page 44.

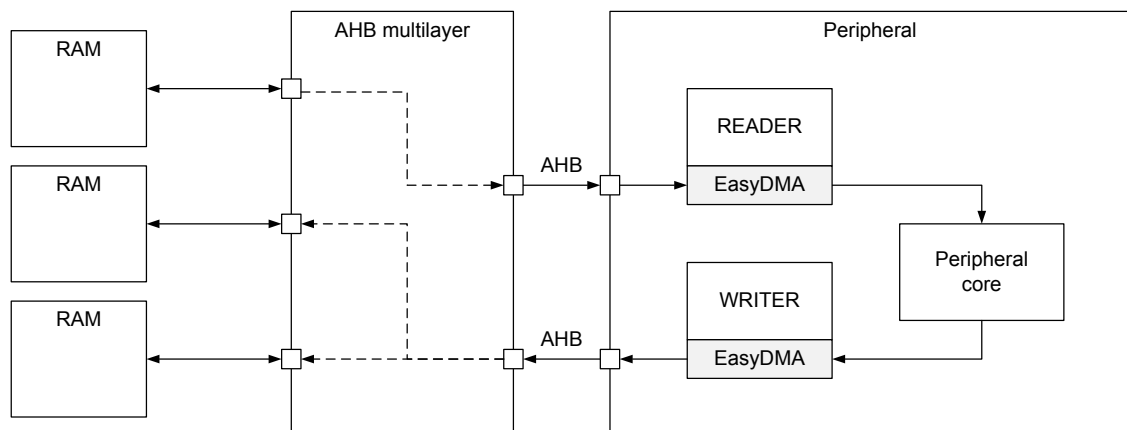


Figure 4: EasyDMA example

An EasyDMA channel is implemented in the following way, but some variations may occur:

```

READERBUFFER_SIZE 5
WRITERBUFFER_SIZE 6

uint8_t readerBuffer[READERBUFFER_SIZE] __at__ 0x20000000;
uint8_t writerBuffer[WRITERBUFFER_SIZE] __at__ 0x20000005;

// Configuring the READER channel
MYPERIPHERAL->READER.MAXCNT = READERBUFFER_SIZE;
MYPERIPHERAL->READER.PTR = &readerBuffer;

// Configure the WRITER channel
MYPERIPHERAL->WRITER.MAXCNT = WRITERBUFFER_SIZE;
MYPERIPHERAL->WRITER.PTR = &writerBuffer;

```

This example shows a peripheral called MYPERIPHERAL that implements two EasyDMA channels - one for reading called READER, and one for writing called WRITER. When the peripheral is started, it is assumed that the peripheral will perform the following tasks:

- Read 5 bytes from the readerBuffer located in RAM at address 0x20000000
- Process the data
- Write no more than 6 bytes back to the writerBuffer located in RAM at address 0x20000005

The memory layout of these buffers is illustrated in [EasyDMA memory layout](#) on page 45.

0x20000000	readerBuffer[0]	readerBuffer[1]	readerBuffer[2]	readerBuffer[3]
0x20000004	readerBuffer[4]	writerBuffer[0]	writerBuffer[1]	writerBuffer[2]
0x20000008	writerBuffer[3]	writerBuffer[4]	writerBuffer[5]	

Figure 5: EasyDMA memory layout

The WRITER.MAXCNT register should not be specified larger than the actual size of the buffer (writerBuffer). Otherwise, the channel would overflow the writerBuffer.

Once an EasyDMA transfer is completed, the AMOUNT register can be read by the CPU to see how many bytes were transferred. For example, CPU can read MYPERIPHERAL->WRITER.AMOUNT register to see how many bytes WRITER wrote to RAM.

Note: The PTR register of a READER or WRITER must point to a valid memory region before use. The reset value of a PTR register is not guaranteed to point to valid memory. See [Memory](#) on page 19 for more information about the different memory regions and EasyDMA connectivity.

4.6.1 EasyDMA error handling

Some errors may occur during DMA handling.

If READER.PTR or WRITER.PTR is not pointing to a valid memory region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 19 for more information about the different memory regions.

If several AHB bus masters try to access the same AHB slave at the same time, AHB bus congestion might occur. An EasyDMA channel is an AHB master. Depending on the peripheral, the peripheral may either stall and wait for access to be granted, or lose data.

4.6.2 EasyDMA array list

EasyDMA is able to operate in Array List mode.

The Array List mode is implemented in channels where the LIST register is available.

The array list does not provide a mechanism to explicitly specify where the next item in the list is located. Instead, it assumes that the list is organized as a linear array where items are located one after the other in RAM.

The EasyDMA Array List can be implemented by using the data structure `ArrayList_type` as illustrated in the code example below using a `READER` EasyDMA channel as an example:

```
#define BUFFER_SIZE 4

typedef struct ArrayList
{
    uint8_t buffer[BUFFER_SIZE];
} ArrayList_type;

ArrayList_type ReaderList[3] __at__ 0x20000000;

MYPERIPHERAL->READER.MAXCNT = BUFFER_SIZE;
MYPERIPHERAL->READER.PTR = &ReaderList;
MYPERIPHERAL->READER.LIST = MYPERIPHERAL_READER_LIST_ArrayList;
```

The data structure only includes a buffer with size equal to the size of `READER.MAXCNT` register. EasyDMA uses the `READER.MAXCNT` register to determine when the buffer is full.

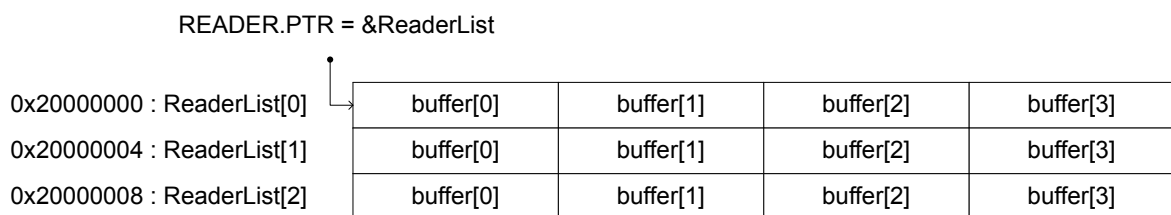


Figure 6: EasyDMA array list

4.7 AHB multilayer

AHB multilayer enables parallel access paths between multiple masters and slaves in a system. Access is resolved using priorities.

Each bus master is connected to all the slave devices using an interconnection matrix. The bus masters are assigned priorities, which are used to resolve access when two (or more) bus masters request access to the same slave device. When that occurs, the following rules apply:

- If two (or more) bus masters request access to the same slave device, the master with the highest priority is granted the access first.
- Bus masters with lower priority are stalled until the higher priority master has completed its transaction.
- If the higher priority master pauses at any point during its transaction, the lower priority master in queue is temporarily granted access to the slave device until the higher priority master resumes its activity.
- Bus masters that have the same priority are mutually exclusive, thus cannot be used concurrently.

Some peripherals, such as `RADIO`, do not have a safe stalling mechanism (no internal data buffering, or opportunity to pause incoming data). Being a low priority bus master might cause loss of data for such peripherals upon bus contention. To avoid AHB bus contention when using multiple bus masters, follow these guidelines:

- Avoid situations where more than one bus master is accessing the same slave.
- If more than one bus master is accessing the same slave, make sure that the bus bandwidth is not exhausted.

Below is a list of bus masters in the system and their priorities.

Bus master name	Description
CPU	
CTRL-AP	
USB	
SPIM1/SPIS1/TWIM1/TWIS1	Same priority and mutually exclusive
RADIO	
CCM/ECB/AAR	Same priority and mutually exclusive
SAADC	
UARTE0	
SPIM0/SPIS0/TWIM0/TWIS0	Same priority and mutually exclusive
SPIM2/SPIS2	Same priority and mutually exclusive
NFCT	
I2S	
PDM	
PWM0	
PWM1	
PWM2	
PWM3	
UARTE1	
SPIM3	

Table 12: AHB bus masters (listed from highest to lowest priority)

Defined bus masters are the CPU and peripherals with implemented EasyDMA. The available slaves are RAM AHB slaves. How the bus masters and slaves are connected using the interconnection matrix is illustrated in [Memory](#) on page 19.

4.8 Debug and trace

Debug and trace system offers a flexible and powerful mechanism for non-intrusive debugging.

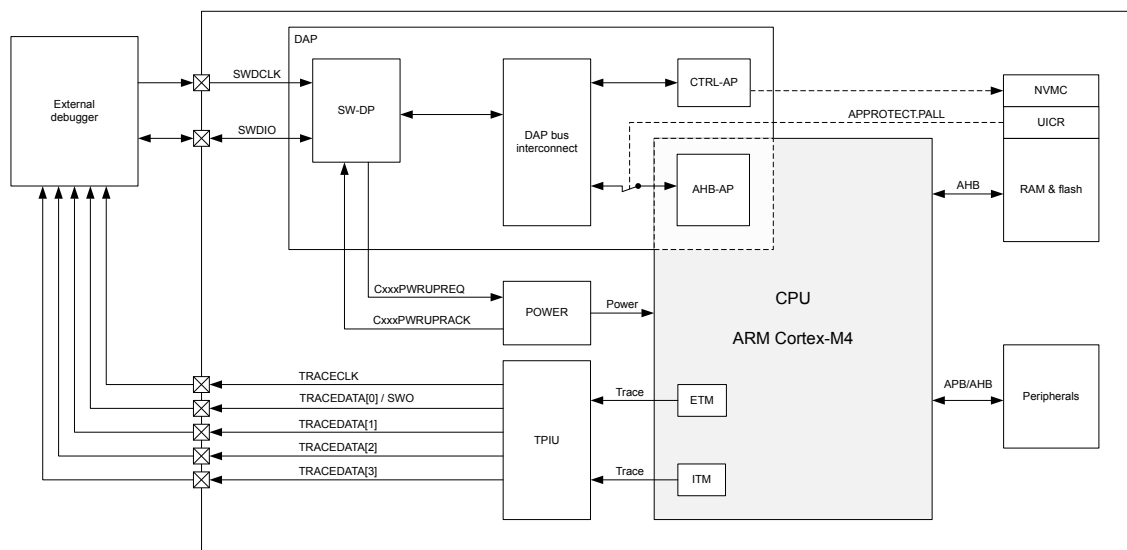


Figure 7: Debug and trace overview

The main features of the debug and trace system are the following:

- Two-pin serial wire debug (SWD) interface
- Flash patch and breakpoint (FPB) unit that supports:
 - Two literal comparators
 - Six instruction comparators
- Data watchpoint and trace (DWT) unit with four comparators
- Instrumentation trace macrocell (ITM)
- Embedded trace macrocell (ETM)
- Trace port interface unit (TPIU)
 - 4-bit parallel trace of ITM and ETM trace data
 - Serial wire output (SWO) trace of ITM data

4.8.1 DAP - Debug access port

An external debugger can access the device via the DAP.

The debug access port (DAP) implements a standard ARM® CoreSight™ serial wire debug port (SW-DP), which implements the serial wire debug protocol (SWD). SWD is a two-pin serial interface, see SWDCLK and SWDIO in [Debug and trace overview](#) on page 47.

In addition to the default access port in CPU (AHB-AP), the DAP includes a custom control access port (CTRL-AP). The CTRL-AP is described in more detail in [CTRL-AP - Control access port](#) on page 48.

Note:

- The SWDIO line has an internal pull-up resistor.
- The SWDCLK line has an internal pull-down resistor.

4.8.2 CTRL-AP - Control access port

The control access port (CTRL-AP) is a custom access port that enables control of the device when other access ports in the DAP are disabled by the access port protection.

Access port protection blocks the debugger from read and write access to all CPU registers and memory-mapped addresses. See the UICR register [APPROTECT](#) on page 42 for more information on enabling access port protection.

Control access port has the following features:

- Soft reset, see [Reset](#) on page 67 for more information
- Disabling of access port protection, which is the reason why CTRL-AP allows control of the device even when all other access ports in the DAP are disabled by the access port protection

Access port protection is disabled by issuing an ERASEALL command via CTRL-AP. This command will erase the flash, UICR, and RAM.

4.8.2.1.4 APPROTECTSTATUS

Address offset: 0x00C

Status register for access port protection

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	R	APPROTECTSTATUS		Status register for access port protection																														
		Enabled	0	Access port protection enabled																														
		Disabled	1	Access port protection not enabled																														

4.8.2.1.5 IDR

Address offset: 0x0FC

CTRL-AP identification register, IDR

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			E E E E D D D C C C C C C C B B B B A A A A A A A A																															
Reset 0x02880000			0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A	R	APID				AP identification																												
B	R	CLASS				Access port (AP) class																												
			NotDefined	0x0	No defined class																													
			MEMAP	0x8	Memory access port																													
C	R	JEP106ID				JEDEC JEP106 identity code																												
D	R	JEP106CONT				JEDEC JEP106 continuation code																												
E	R	REVISION				Revision																												

4.8.2.2 Electrical specification

4.8.2.2.1 Control access port

Symbol	Description	Min.	Typ.	Max.	Units
R_{pull}	Internal SWDIO and SWDCLK pull up/down resistance		13		k Ω
f_{SWDCLK}	SWDCLK frequency	0.125		8	MHz

4.8.3 Debug interface mode

Before an external debugger can access either CPU's access port (AHB-AP) or the control access port (CTRL-AP), the debugger must first request the device to power up via CxxxPWRUPREQ in the SWJ-DP.

If the device is in System OFF when power is requested via CxxxPWRUPREQ, the system will wake up and the DIF flag in [RESETREAS](#) on page 73 will be set. The device is in the debug interface mode as long as the debugger is requesting power via CxxxPWRUPREQ. Once the debugger stops requesting power via CxxxPWRUPREQ, the device is back in normal mode. Some peripherals behave differently in Debug Interface mode compared to normal mode. These differences are described in more detail in the chapters of the peripherals that are affected.

When a debug session is over, the external debugger must make sure to put the device back into normal mode since the overall power consumption is higher in debug interface mode than in normal mode.

For details on how to use the debug capabilities, read the debug documentation of your IDE.

4.8.4 Real-time debug

The nRF52833 supports real-time debugging.

Real-time debugging allows interrupts to execute to completion in real time when breakpoints are set in thread mode or lower priority interrupts. This enables developers to set breakpoints and single-step through the code without the risk of real-time event-driven threads running at higher priority failing. For example, this enables the device to continue to service the high-priority interrupts of an external controller or sensor without failure or loss of state synchronization while the developer steps through code in a low-priority thread.

4.8.5 Trace

The device supports ETM and ITM trace.

Trace data from the ETM and the ITM is sent to an external debugger via a 4-bit wide parallel trace port interface unit (TPIU), see TRACEDATA[0] through TRACEDATA[3] and TRACECLK in [Debug and trace overview](#) on page 47.

In addition to parallel trace, the TPIU supports serial trace via the serial wire output (SWO) trace protocol. Parallel and serial trace cannot be used at the same time. ETM trace is only supported in Parallel Trace mode, while ITM trace is supported in both Parallel and Serial Trace modes.

For details on how to use the trace capabilities, read the debug documentation of your IDE.

TPIU's trace pins are multiplexed with GPIOs. SWO and TRACEDATA[0] use the same GPIO. See [Pin assignments](#) on page 557 for more information.

Trace speed is configured in register [TRACECONFIG](#) on page 93. The speed of the trace pins depends on the DRIVE setting of the GPIOs that the trace pins are multiplexed with. Only S0S1 and H0H1 drives are suitable for debugging. S0S1 is the default DRIVE setting at reset. If parallel or serial trace port signals are not fast enough with the default settings, all GPIOs in use for tracing should be set to high drive (H0H1). The DRIVE setting for these GPIOs should not be overwritten by firmware during the debugging session.

4.8.5.1 Electrical specification

4.8.5.1.1 Trace port

Symbol	Description	Min.	Typ.	Max.	Units
T _{cyc}	Clock period as defined by Arm in the Timing specifications for Trace Port Physical Interface of the Embedded Trace Macrocell Architecture Specification	62.5		500	ns

5 Power and clock management

5.1 Power management unit (PMU)

Power and clock management in nRF52833 is designed to automatically ensure maximum power efficiency.

The core of the power and clock management system is the power management unit (PMU) illustrated in the following figure.

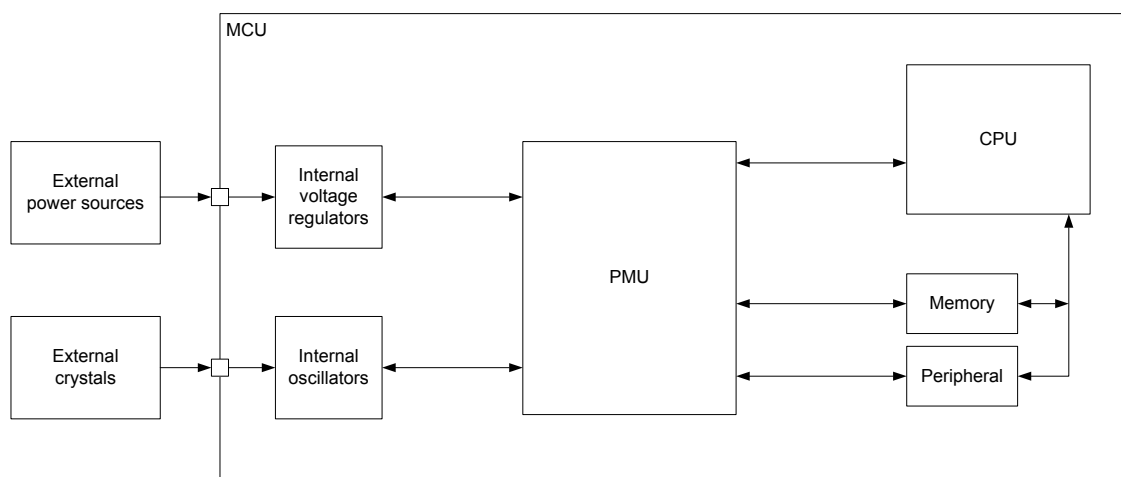


Figure 8: Power management unit

The PMU automatically detects which power and clock resources are required by the different system components at any given time. The PMU will then automatically start/stop and choose operation modes in supply regulators and clock sources, to achieve the lowest power consumption possible.

5.2 Current consumption

Because the system is continually being tuned by the [Power management unit \(PMU\)](#) on page 52, estimating an application's current consumption can be challenging when measurements cannot be directly performed on the hardware. To facilitate the estimation process, a set of current consumption scenarios are provided to show the typical current drawn from the VDD supply.

Each scenario specifies a set of operations and conditions applying to the given scenario. The following table shows a set of common conditions used in all scenarios, unless otherwise stated in the description of a given scenario. All scenarios are listed in [Electrical specification](#) on page 53.

Condition	Value
Supply	3 V on VDD/VDDH (Normal voltage mode)
Temperature	25°C
CPU	WFI (wait for interrupt)/WFE (wait for event) sleep
Peripherals	All idle
Clock	Not running
Regulator	LDO
RAM	In System ON, full 128 kB powered. In System OFF, full 128 kB retention.
Compiler	GCC v7.3.1 20180622 (release) [ARM/embedded-7-branch revision 261907] (GNU Tools for Arm Embedded Processors 7-2018-q3-update). Compiler flags: -O0 -falign-functions=16 -fno-strict-aliasing -mthumb -mcpu=cortex-m4 -mfloat-abi=hard -mfpu=fpv4-sp-d16.
Compiler for CPU Running and Compounded	ARMCC v6.13. - Compiler flags: -xc -std=gnu99 --target=arm-arm-none-eabi -mcpu=cortex-m4 -mfpu=none -mfloat-abi=soft -c -fno-rtti -funsigned-char -gdwarf-3 -fropi -Ofast -ffunction-sections -Omax - Linker flags: --cpu=Cortex-M4 --fpu=SoftVFP --strict -Omax
Cache enabled ²	Yes
32 MHz crystal ³	SMD 2520, 32 MHz, 10 pF +/- 10 ppm

Table 14: Current consumption scenarios, common conditions

5.2.1 Electrical specification

5.2.1.1 Sleep

Symbol	Description	Min.	Typ.	Max.	Units
I _{ON_RAMOFF_EVENT}	System ON, no RAM retention, wake on any event		1.1		μA
I _{ON_RAMON_EVENT}	System ON, full 128 kB RAM retention, wake on any event		1.8		μA
I _{ON_RAMON_POF}	System ON, full 128 kB RAM retention, wake on any event, power-fail comparator enabled		1.9		μA
I _{ON_RAMON_GPIOTE}	System ON, full 128 kB RAM retention, wake on GPIOTE input (event mode)		7.4		μA
I _{ON_RAMON_GPIOTEPORT}	System ON, full 128 kB RAM retention, wake on GPIOTE PORT event		1.8		μA
I _{ON_RAMOFF_RTC}	System ON, no RAM retention, wake on RTC (running from LFRC clock)		1.5		μA
I _{ON_RAMON_RTC}	System ON, full 128 kB RAM retention, wake on RTC (running from LFRC clock)		2.6		μA

² Applies only when CPU is running from flash memory

³ Applies only when HFXO is running

Symbol	Description	Min.	Typ.	Max.	Units
$I_{\text{OFF_RAMOFF_RESET}}$	System OFF, no RAM retention, wake on reset		0.6		μA
$I_{\text{OFF_RAMOFF_LPCOMP}}$	System OFF, no RAM retention, wake on LPCOMP		0.9		μA
$I_{\text{OFF_RAMON_RESET}}$	System OFF, full 128 kB RAM retention, wake on reset		1.3		μA
$I_{\text{ON_RAMOFF_EVENT_5V}}$	System ON, no RAM retention, wake on any event, 5 V supply on VDDH, REG0 output = 3.3 V		1.3		μA
$I_{\text{OFF_RAMOFF_RESET_5V}}$	System OFF, no RAM retention, wake on reset, 5 V supply on VDDH, REG0 output = 3.3 V		1.0		μA

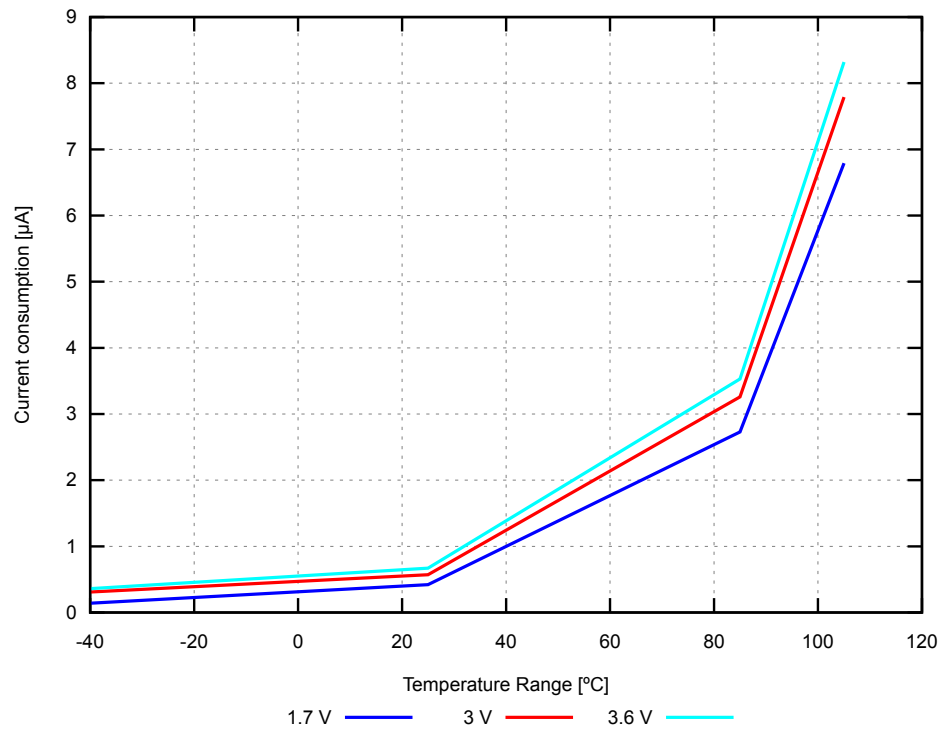


Figure 9: System OFF, no RAM retention, wake on reset (typical values)

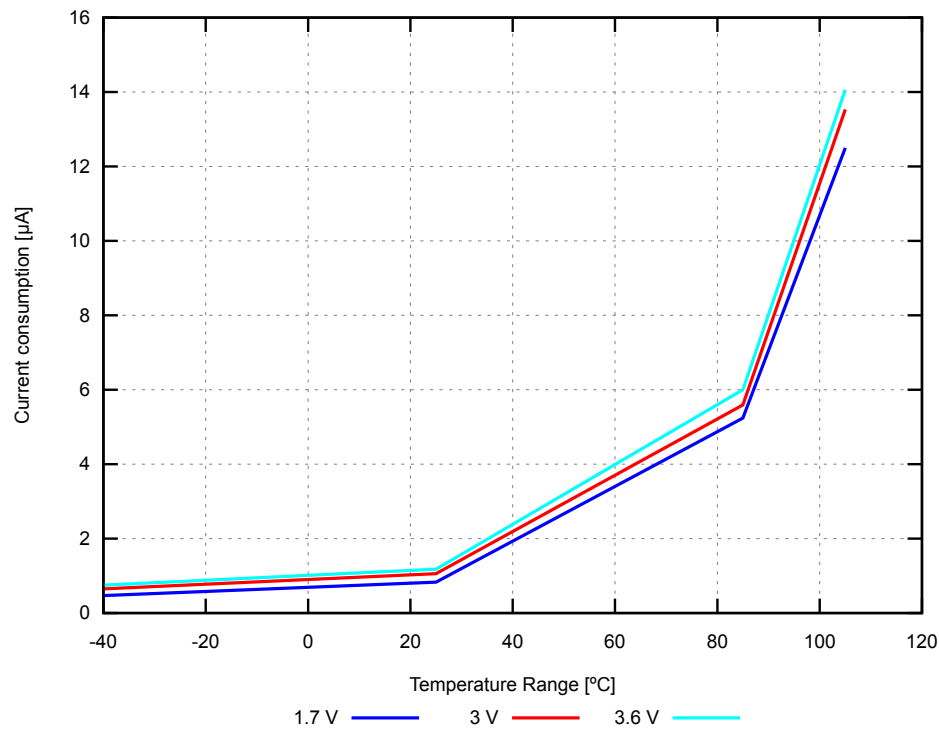


Figure 10: System ON, no RAM retention, wake on any event (typical values)

5.2.1.2 COMP active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{\text{COMP,LP}}$	COMP enabled, low power mode		22.7		μA
$I_{\text{COMP,NORM}}$	COMP enabled, normal mode		26.4		μA
$I_{\text{COMP,HS}}$	COMP enabled, high-speed mode		33.0		μA

5.2.1.3 CPU running

Symbol	Description	Min.	Typ.	Max.	Units
I_{CPU0}	CPU running CoreMark @64 MHz from flash, Clock = HFXO, Regulator = DC/DC		3.3		mA
I_{CPU1}	CPU running CoreMark @64 MHz from flash, Clock = HFXO		5.6		mA
I_{CPU2}	CPU running CoreMark @64 MHz from RAM, Clock = HFXO, Regulator = DC/DC		2.4		mA
I_{CPU3}	CPU running CoreMark @64 MHz from RAM, Clock = HFXO		4.7		mA
I_{CPU4}	CPU running CoreMark @64 MHz from flash, Clock = HFINT, Regulator = DC/DC		3.1		mA

5.2.1.4 NFCT active

Symbol	Description	Min.	Typ.	Max.	Units
I_{sense}	Current in SENSE STATE ⁴		100		nA
$I_{\text{activated}}$	Current in ACTIVATED STATE		400		μA

⁴ This current does not apply when in NFC field

5.2.1.5 Radio transmitting/receiving

Symbol	Description	Min.	Typ.	Max.	Units
$I_{\text{RADIO_TX0}}$	Radio transmitting @ 8 dBm output power, 1 Mbps <i>Bluetooth</i> ® low energy (BLE) mode, Clock = HFXO, Regulator = DC/DC		15.5		mA
$I_{\text{RADIO_TX1}}$	Radio transmitting @ 0 dBm output power, 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC		6.0		mA
$I_{\text{RADIO_TX2}}$	Radio transmitting @ -40 dBm output power, 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC		3.5		mA
$I_{\text{RADIO_TX3}}$	Radio transmitting @ 0 dBm output power, 1 Mbps BLE mode, Clock = HFXO		11.0		mA
$I_{\text{RADIO_TX4}}$	Radio transmitting @ -40 dBm output power, 1 Mbps BLE mode, Clock = HFXO		5.4		mA
$I_{\text{RADIO_TX5}}$	Radio transmitting @ 0 dBm output power, 250 kbit/s IEE 802.15.4-2006 mode, Clock = HFXO, Regulator = DC/DC		6.0		mA
$I_{\text{RADIO_RX0}}$	Radio receiving @ 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC		6.0		mA
$I_{\text{RADIO_RX1}}$	Radio receiving @ 1 Mbps BLE mode, Clock = HFXO		10.5		mA
$I_{\text{RADIO_RX2}}$	Radio receiving @ 250 kbit/s IEE 802.15.4-2006 mode, Clock = HFXO, Regulator = DC/DC		6.2		mA

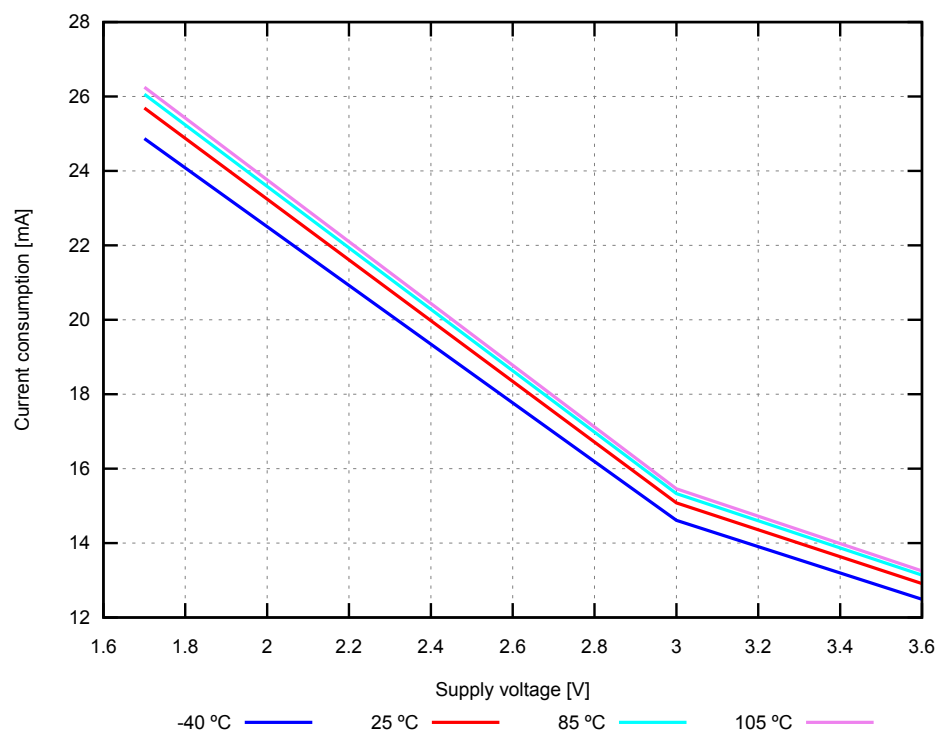


Figure 11: Radio transmitting @ 8 dBm output power, 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC (typical values)

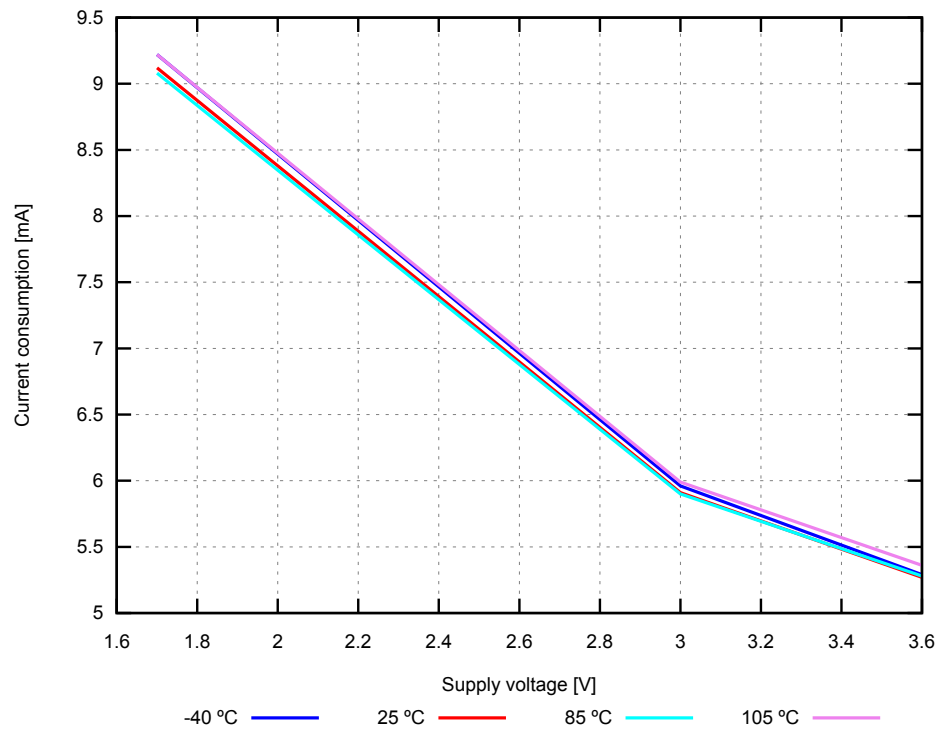


Figure 12: Radio transmitting @ 0 dBm output power, 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC (typical values)

5.2.1.6 RNG active

Symbol	Description	Min.	Typ.	Max.	Units
I _{RNG0}	RNG running		539		μA

5.2.1.7 SAADC active

Symbol	Description	Min.	Typ.	Max.	Units
I _{SAADC, RUN}	SAADC sampling @ 16 kbps, Acquisition time = 20 μs, Clock = HFXO, Regulator = DC/DC		1.37		mA

5.2.1.8 TEMP active

Symbol	Description	Min.	Typ.	Max.	Units
I _{TEMP0}	TEMP started		0.92		mA

5.2.1.9 TIMER running

Symbol	Description	Min.	Typ.	Max.	Units
I _{TIMER0}	One TIMER instance running @ 1 MHz, Clock = HFINT		342		μA
I _{TIMER1}	Two TIMER instances running @ 1 MHz, Clock = HFINT		341		μA
I _{TIMER2}	One TIMER instance running @ 1 MHz, Clock = HFXO		573		μA
I _{TIMER3}	One TIMER instance running @ 16 MHz, Clock = HFINT		497		μA
I _{TIMER4}	One TIMER instance running @ 16 MHz, Clock = HFXO		729		μA

5.2.1.10 USB running

Symbol	Description	Min.	Typ.	Max.	Units
$I_{USB,ACTIVE,VBUS}$	Current from VBUS supply, USB active		2.4		mA
$I_{USB,SUSPEND,VBUS}$	Current from VBUS supply, USB suspended, CPU sleeping		262		μ A
$I_{USB,ACTIVE,VDD}$	Current from VDD supply (normal voltage mode), all RAM retained, regulator=LDO, CPU running, USB active		7.73		mA
$I_{USB,SUSPEND,VDD}$	Current from VDD supply (normal voltage mode), all RAM retained, regulator=LDO, CPU sleeping, USB suspended		173		μ A
$I_{USB,ACTIVE,VDDH}$	Current from VDDH supply (high voltage mode), VDD=3 V (REG0 output), all RAM retained, regulator=LDO, CPU running, USB active		7.46		mA
$I_{USB,SUSPEND,VDDH}$	Current from VDDH supply (high voltage mode), VDD=3 V (REG0 output), all RAM retained, regulator=LDO, CPU sleeping, USB suspended		178		μ A
$I_{USB,DISABLED,VDD}$	Current from VDD supply, USB disabled, VBUS supply connected, all RAM retained, regulator=LDO, CPU sleeping		7		μ A

5.2.1.11 WDT active

Symbol	Description	Min.	Typ.	Max.	Units
$I_{WDT,STARTED}$	WDT started		2.5		μ A

5.2.1.12 Compounded

Symbol	Description	Min.	Typ.	Max.	Units
I_{S0}	CPU running CoreMark from flash, Radio transmitting @ 0 dBm output power, 1 Mbps <i>Bluetooth</i> [®] low energy (BLE) mode, Clock = HFXO, Regulator = DC/DC		8.5		mA
I_{S1}	CPU running CoreMark from flash, Radio receiving @ 1 Mbps BLE mode, Clock = HFXO, Regulator = DC/DC		8.3		mA
I_{S2}	CPU running CoreMark from flash, Radio transmitting @ 0 dBm output power, 1 Mbps BLE mode, Clock = HFXO		16.7		mA
I_{S3}	CPU running CoreMark from flash, Radio receiving @ 1 Mbps BLE mode, Clock = HFXO		16.2		mA

5.3 POWER — Power supply

The power supply consists of a number of LDO and DC/DC regulators that are utilized to maximize the system's power efficiency.

This device has the following power supply features:

- On-chip LDO and DC/DC regulators
- Global System ON/OFF modes
- Individual RAM section power control for all system modes
- Analog or digital pin wakeup from System OFF
- Supervisor hardware to manage power-on reset, brownout, and power failure
- Auto-controlled refresh modes for LDO and DC/DC regulators to maximize efficiency
- Separate USB supply

5.3.1 Main supply

The main supply voltage is connected to the VDD/VDDH pins. The system will enter one of two supply voltage modes, Normal or High Voltage mode, depending on how the supply voltage is connected to these pins.

The system enters Normal Voltage mode when the supply voltage is connected to both the VDD and VDDH pins (pin VDD shorted to pin VDDH). For the supply voltage range to connect to both VDD and VDDH pins, see parameter V_{DD} .

The system enters High Voltage mode when the supply voltage is only connected to the VDDH pin and the VDD pin is not connected to any voltage supply. For the supply voltage range to connect to the VDDH pin, see parameter V_{DDH} .

The register [MAINREGSTATUS](#) on page 76 can be used to read the current supply voltage mode.

5.3.1.1 Main voltage regulators

The system contains two main supply regulator stages, REG0 and REG1.

REG1 regulator stage has the regulator type options of Low-dropout regulator (LDO) and Buck regulator (DC/DC). REG0 regulator stage has only the option of Low-dropout regulator (LDO).

In Normal Voltage mode, only the REG1 regulator stage is used, and the REG0 stage is automatically disabled. In High Voltage mode, both regulator stages (REG0 and REG1) are used. The output voltage of REG0 can be configured in register [REGOUT0](#) on page 43. This output voltage is connected to VDD and is the input voltage to REG1.

Note: In High Voltage mode, the configured output voltage for REG0 ([REGOUT0](#) on page 43) must not be greater than REG0 input voltage minus the voltage drop in REG0 ($V_{DDH} - V_{REG0,DROP}$).

By default, the LDO regulators are enabled and the DC/DC regulator of REG1 stage is disabled. Register [DCDCEN](#) on page 76 is used to enable the DC/DC regulator for REG1 stage.

When the REG1 DC/DC converter is enabled, the LDO for the REG1 stage will be disabled. External LC filters must be connected for the DC/DC regulator if it is being used. The advantage of using the DC/DC regulator is that the overall power consumption is normally reduced as the efficiency of such a regulator is higher than that of a LDO. The efficiency gained by using the DC/DC regulator is best seen when the regulator voltage drop (difference between input and output voltage) is high. The efficiency of internal regulators vary with the supply voltage and the current drawn from the regulators.

Note: Do not enable the DC/DC regulator without an external LC filter being connected as this will inhibit device operation, including debug access, until an LC filter is connected.

5.3.1.2 GPIO levels

The GPIO high reference voltage is equal to the level on the VDD pin.

In Normal Voltage mode, the GPIO high level equals the voltage supplied to the VDD pin. In High Voltage mode, it equals the level specified in register [REGOUT0](#) on page 43.

5.3.1.3 Regulator configuration examples

The voltage regulators can be configured in several ways, depending on the selected supply voltage mode (Normal/High) and the regulator type option for REG1 (LDO or DC/DC).

Four configuration examples are illustrated in the following figures.

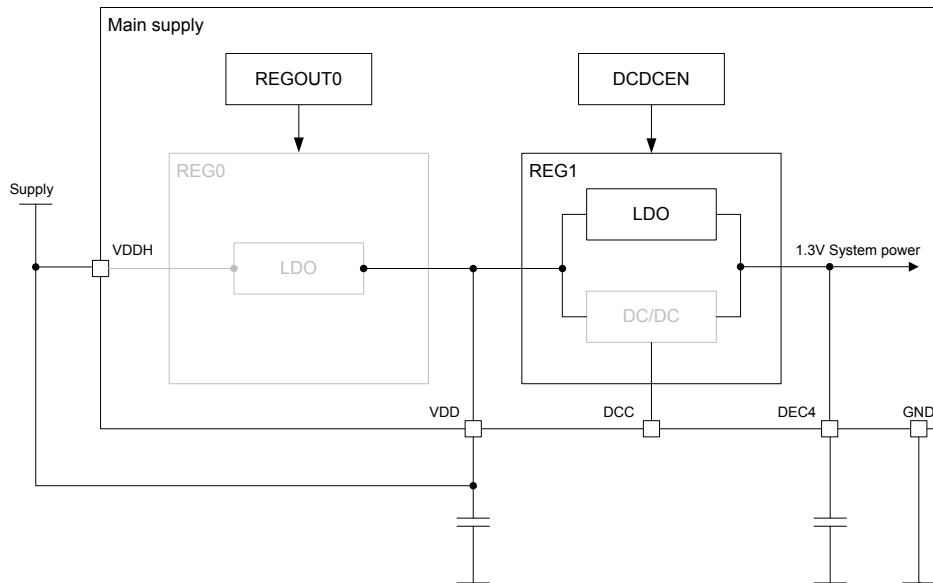


Figure 13: Normal Voltage mode, REG1 LDO

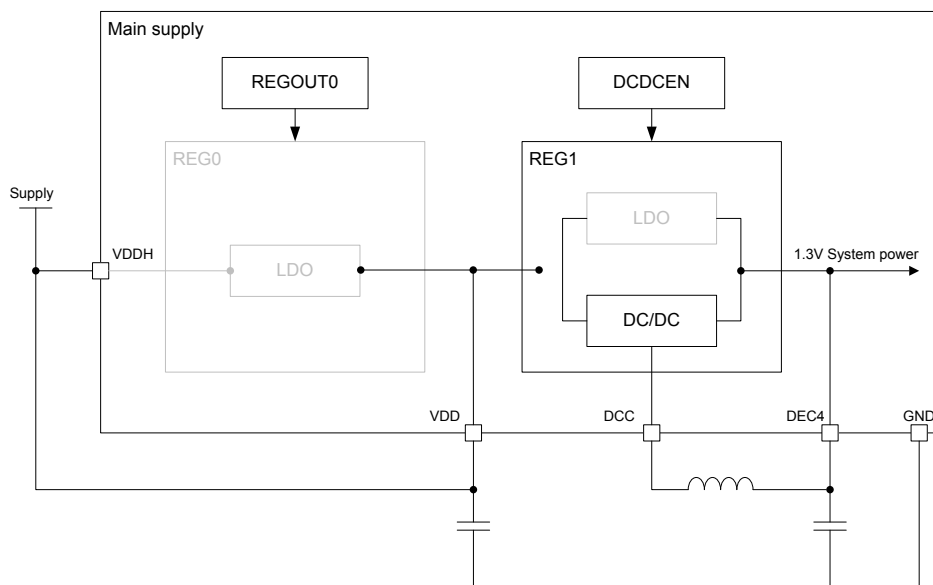


Figure 14: Normal Voltage mode, REG1 DC/DC

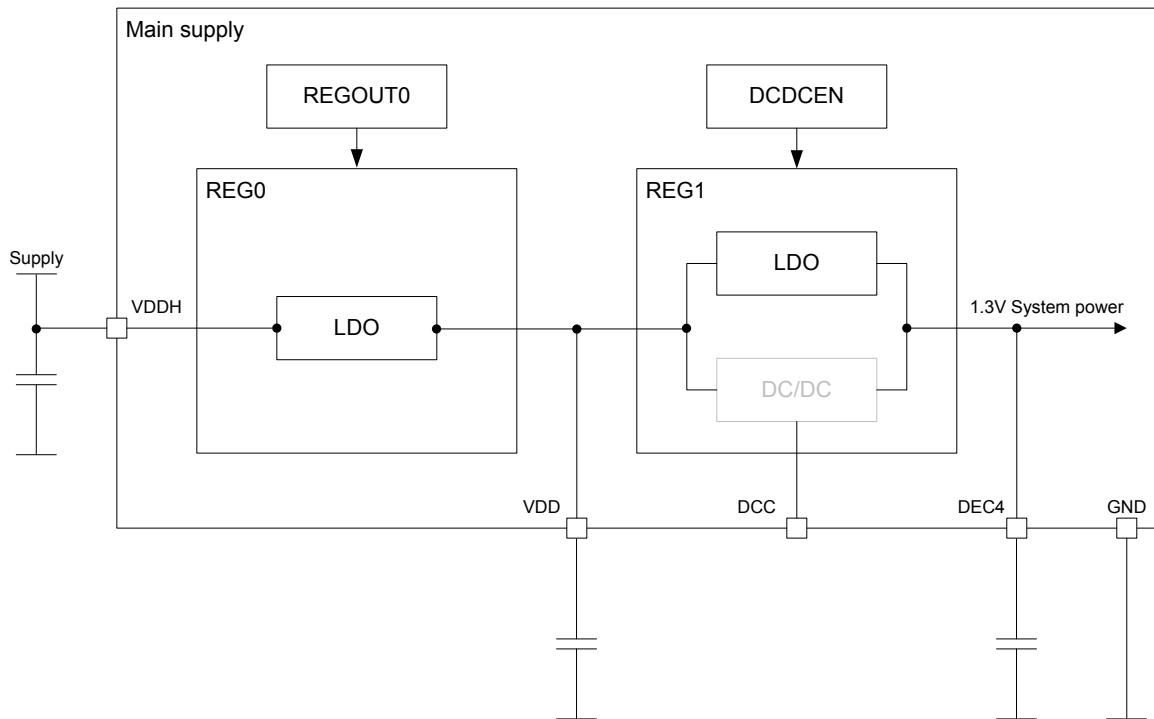


Figure 15: High Voltage mode, REG1 LDO

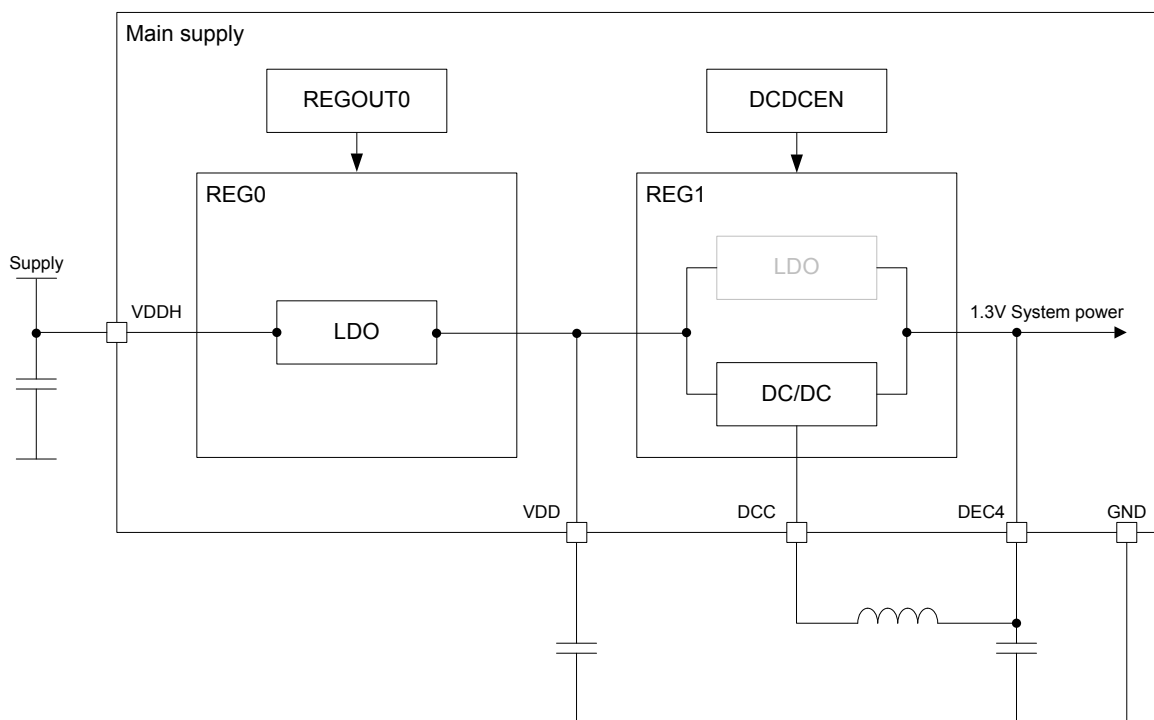


Figure 16: High Voltage mode, REG1 DC/DC

5.3.1.4 Power supply supervisor

The power supply supervisor enables monitoring of the connected power supply.

The power supply supervisor provides the following functionality:

- Power-on reset - signals the circuit when a supply is connected

- An optional power-fail comparator (POF) - signals the application when the supply voltages drop below a configured threshold
- A fixed brownout reset detector - holds the system in reset when the voltage is too low for safe operation

The power supply supervisor is illustrated in the following figure.

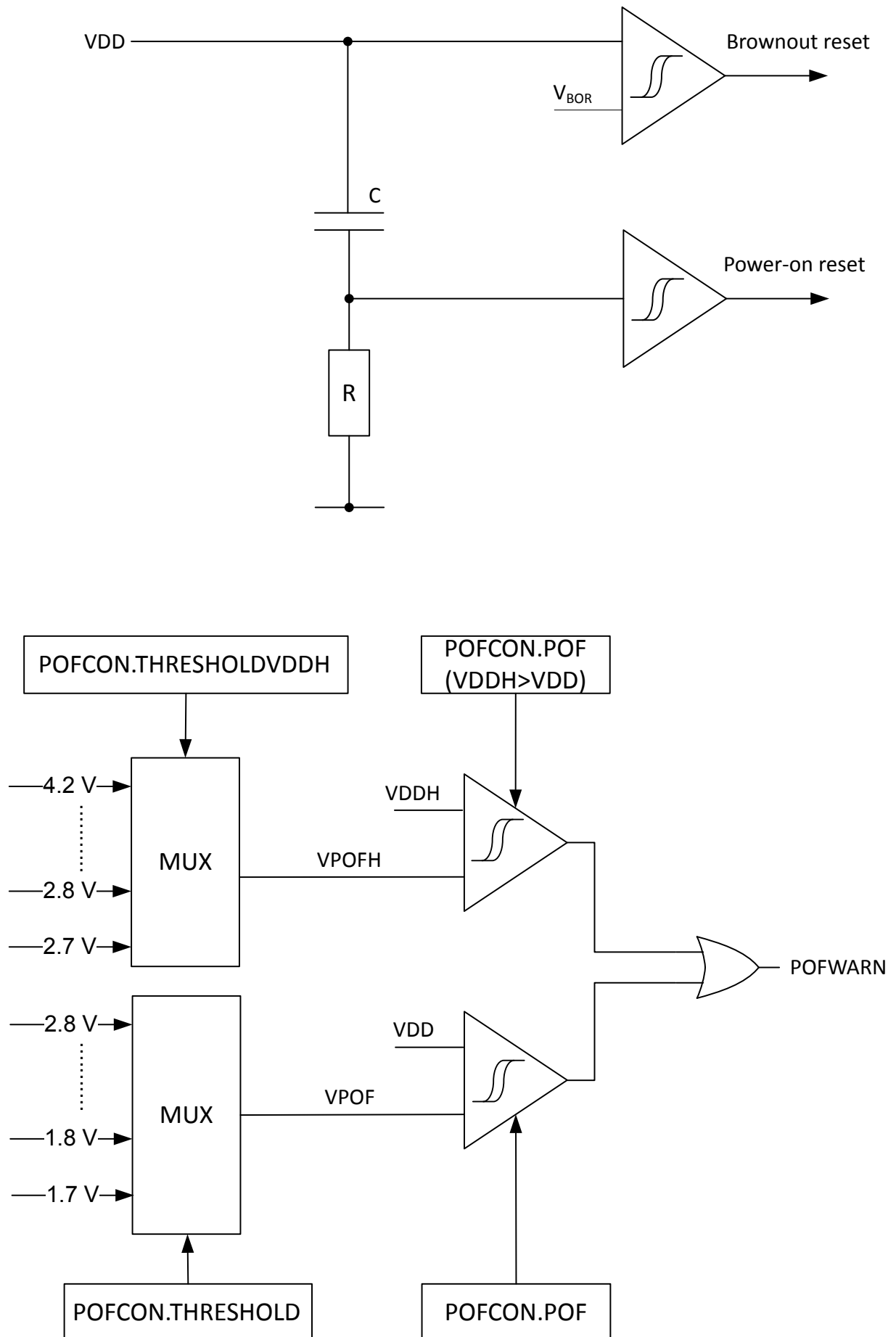


Figure 17: Power supply supervisor

5.3.1.5 Power-fail comparator

Using the power-fail comparator (POF) is optional. When enabled, it can provide an early warning to the CPU of an impending power supply failure.

To enable and configure the power-fail comparator, see the register [POFCON](#) on page 75.

When the supply voltage falls below the defined threshold, the power-fail comparator generates an event (POFWARN) that can be used by an application to prepare for power failure. This event is also generated when the supply voltage is already below the threshold at the time the power-fail comparator is enabled, or if the threshold is re-configured to a level above the supply voltage.

If the power failure warning is enabled, and the supply voltage is below the threshold, the power-fail comparator will prevent the [NVMC](#) from performing write operations to the flash.

The comparator features a hysteresis of V_{HYST} , as illustrated in the following figure.

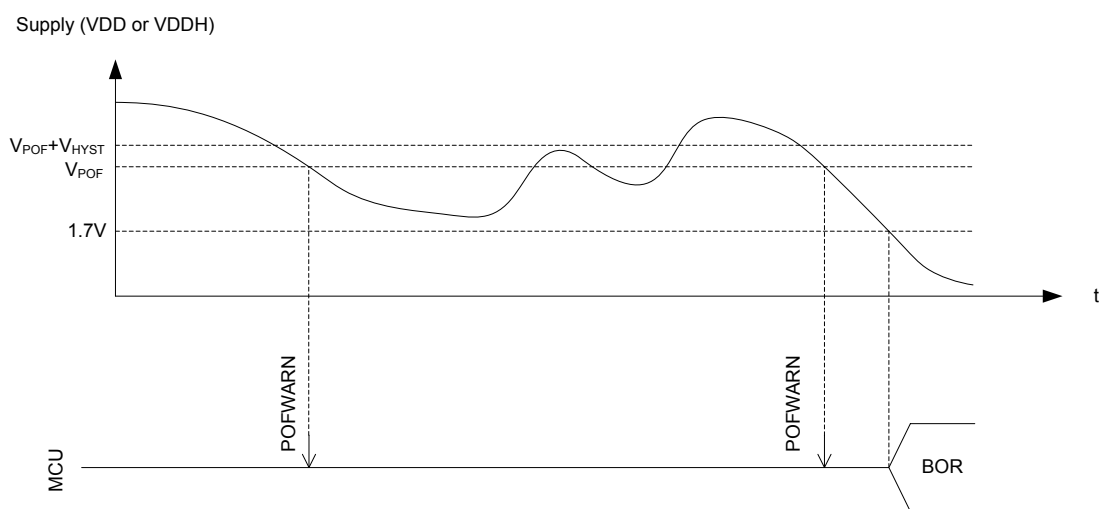


Figure 18: Power-fail comparator (BOR = brownout reset)

To save power, the power-fail comparator is not active in System OFF or System ON when HFCLK is not running.

5.3.2 USB supply

When using the USB peripheral, a 5 V USB supply needs to be provided to the VBUS pin.

The USB peripheral has a dedicated internal voltage regulator for converting the VBUS supply to 3.3 V used by the USB signalling interface (D+ and D- lines, and pull-up on D+). The remainder of the USB peripheral (USBID) is supplied through the main supply like other on-chip features. As a consequence, VBUS and either VDDH or VDD supplies are required for USB peripheral operation.

When VBUS rises into its valid range, the software is notified through a USBDETECTED event. A USBREMOVED event is sent when VBUS goes below its valid range. Use these events to implement the USBID start-up sequence described in the [USBID](#) chapter.

When VBUS rises into its valid range while the device is in System OFF, the device resets and transitions to System ON mode. The [RESETREAS](#) register will have the VBUS bit set to indicate the source of the wake-up.

See [VBUS detection specifications](#) on page 80 for the levels at which the events are sent ($V_{BUS,DETECT}$ and $V_{BUS,REMOVE}$) or at which the system is woken up from System OFF ($V_{BUS,DETECT}$).

When the USB peripheral is enabled through the [ENABLE](#) register, and VBUS is detected, the regulator is turned on. A USBPWRRDY event is sent when the regulator's worst case settling time has elapsed, indicating to the software that it can enable the USB pull-up to signal a USB connection to the host.

The software can read the state of the VBUS detection and regulator output readiness at any time through the [USBREGSTATUS](#) register.

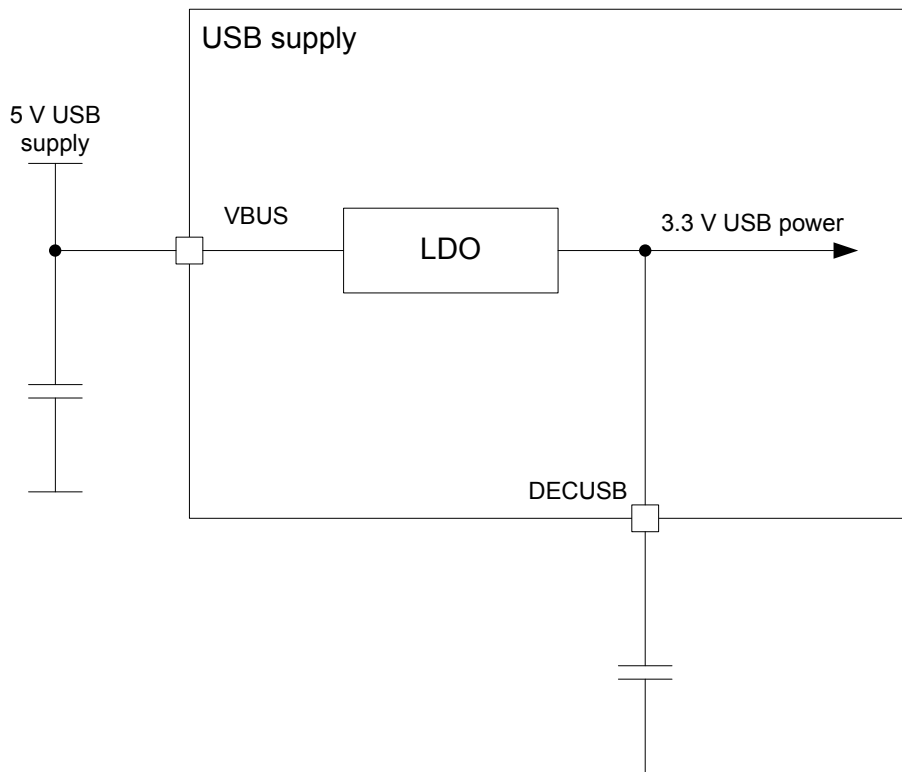


Figure 19: USB voltage regulator

To ensure stability, the input and output of the USB regulator need to be decoupled with a suitable decoupling capacitor. See [Reference circuitry](#) on page 567 for the recommended values.

5.3.3 System OFF mode

System OFF is the deepest power saving mode the system can enter. In this mode, the system's core functionality is powered down and all ongoing tasks are terminated.

The device can be put into System OFF mode using the register [SYSTEMOFF](#) on page 74. When in System OFF mode, the device can be woken up through one of the following signals:

- The DETECT signal, optionally generated by the GPIO peripheral.
- The ANADETECT signal, optionally generated by the LPCOMP module.
- The SENSE signal, optionally generated by the NFC module to wake-on-field.
- Detecting a [valid USB voltage](#) on the VBUS pin ($V_{BUS,DETECT}$).
- A reset.

The system is reset when it wakes up from System OFF mode.

One or more RAM sections can be retained in System OFF mode, depending on the settings in the `RAM[n].POWER` registers. `RAM[n].POWER` are retained registers. These registers are usually overwritten by the start-up code provided with the nRF application examples.

Before entering System OFF mode, all on-going EasyDMA transactions need to have completed. See peripheral specific chapters for more information about how to acquire the status of EasyDMA transactions.

5.3.3.1 Emulated System OFF mode

If the device is in Debug Interface mode, System OFF will be emulated to secure that all required resources needed for debugging are available during System OFF.

Required resources needed for debugging include the following key components:

- [Debug and trace](#) on page 47
- [CLOCK — Clock control](#) on page 80
- [POWER — Power supply](#) on page 58
- [NVMC — Non-volatile memory controller](#) on page 23
- [CPU](#) on page 18
- Flash memory
- RAM

See [Debug and trace](#) on page 47 for more information.

Because the CPU is kept on in an emulated System OFF mode, it is recommended to add an infinite loop directly after entering System OFF, to prevent the CPU from executing code that normally should not be executed.

5.3.4 System ON mode

System ON is the default state after power-on reset. In System ON mode, all functional blocks such as the CPU or peripherals can be in IDLE or RUN mode, depending on the configuration set by the software and the state of the application executing.

Register [RESETREAS](#) on page 73 provides information about the source causing the wakeup or reset.

The system can switch the appropriate internal power sources on and off, depending on the amount of power needed at any given time. The power requirement of a peripheral is directly related to its activity level, and the activity level of a peripheral fluctuates when specific tasks are triggered or events are generated.

5.3.4.1 Sub-power modes

In System ON mode, when the CPU and all peripherals are in IDLE mode, the system can reside in one of the two sub-power modes.

The sub-power modes are:

- Constant Latency
- Low-power

In Constant Latency mode, the CPU wakeup latency and the PPI task response are constant and kept at a minimum. This is secured by forcing a set of basic resources to be turned on while in sleep. The cost of constant and predictable latency is increased power consumption. Constant Latency mode is selected by triggering the CONSTLAT task.

In Low-power mode, the automatic power management system described in [System ON mode](#) on page 66 ensures that the most efficient supply option is chosen to save power. The cost of having the lowest possible power consumption is a varying CPU wakeup latency and PPI task response. Low-power mode is selected by triggering the LOWPWR task.

When the system enters System ON mode, it is by default in the sub-power mode Low-power.

5.3.5 RAM power control

The RAM power control registers are used for configuring the following:

- The RAM sections to be retained during System OFF
- The RAM sections to be retained and accessible during System ON

In System OFF, retention of a RAM section is configured in the RETENTION field of the corresponding register [RAM\[n\].POWER \(n=0..8\)](#) on page 77.

In System ON, retention and accessibility of a RAM section is configured in the RETENTION and POWER fields of the corresponding register [RAM\[n\].POWER \(n=0..8\)](#) on page 77.

The following table summarizes the behavior of these registers.

Configuration			RAM section status	
System on/off	RAM[n].POWER.POWER	RAM[n].POWER.RETENTION	Accessible	Retained
Off	x	Off	No	No
Off	x	On	No	Yes
On	Off	Off	No	No
On	Off ⁵	On	No	Yes
On	On	x	Yes	Yes

Table 15: RAM section configuration

The advantage of not retaining RAM contents is that the overall current consumption is reduced.

See [Memory](#) on page 19 for more information on RAM sections.

5.3.6 Reset

Several sources may trigger a reset.

After a reset has occurred, register [RESETREAS](#) can be read to determine which source triggered the reset.

5.3.6.1 Power-on reset

The power-on reset generator initializes the system at power-on.

The system is held in reset state until the supply has reached the minimum operating voltage and the internal voltage regulators have started.

5.3.6.2 Pin reset

A pin reset is generated when the physical reset pin on the device is asserted.

Pin reset is configured via both registers [PSELRESET\[n\] \(n=0..1\)](#) on page 42.

5.3.6.3 Wakeup from System OFF mode reset

The device is reset when it wakes up from System OFF mode.

The debug access port (DAP) is not reset following a wake up from System OFF mode if the device is in Debug Interface mode. See chapter [Debug and trace](#) on page 47 for more information.

5.3.6.4 Soft reset

A soft reset is generated when the SYSRESETREQ bit of the application interrupt and reset control register (AIRCRR) in the ARM® core is set.

See [ARM documentation](#) for more details.

A soft reset can also be generated via the register [RESET](#) on page 49 in the CTRL-AP.

5.3.6.5 Watchdog reset

A Watchdog reset is generated when the watchdog times out.

⁵ Not useful. RAM section power off gives negligible reduction in current consumption when retention is on.

See chapter [WDT — Watchdog timer](#) on page 552 for more information.

5.3.6.6 Brownout reset

The brownout reset generator puts the system in a reset state if VDD drops below the brownout reset (BOR) threshold.

See section [Power fail comparator](#) on page 79 for more information.

5.3.6.7 Retained registers

A retained register is one that will retain its value in System OFF mode and through a reset, depending on the reset source. See the individual peripheral chapters for information on which of their registers are retained.

5.3.6.8 Reset behavior

The various reset sources and their targets are summarized in the table below.

Reset source	Reset target								
	CPU	Peripherals	GPIO	Debug ⁶	SWJ-DP	RAM	WDT	Retained registers	RESETREAS
CPU lockup ⁷	x	x	x						
Soft reset	x	x	x						
Wakeup from System OFF mode reset	x	x		x ⁸		x ⁹	x		
Watchdog reset ¹⁰	x	x	x	x		x	x	x	
Pin reset	x	x	x	x		x	x	x	
Brownout reset	x	x	x	x	x	x	x	x	x
Power-on reset	x	x	x	x	x	x	x	x	x

Note: The RAM is never reset, but depending on a reset source the content of RAM may be corrupted.

5.3.7 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40000000	POWER	POWER	Power control	

Table 16: Instances

Register	Offset	Description
TASKS_CONSTLAT	0x78	Enable Constant Latency mode
TASKS_LOWPWR	0x7C	Enable Low-power mode (variable latency)

⁶ All debug components excluding SWJ-DP. See [Debug and trace](#) on page 47 for more information about the different debug components.

⁷ Reset from CPU lockup is disabled if the device is in Debug Interface mode. CPU lockup is not possible in System OFF.

⁸ The debug components will not be reset if the device is in Debug Interface mode.

⁹ RAM is not reset on wakeup from System OFF mode. RAM, or certain parts of RAM, may not be retained after the device has entered System OFF mode, depending on the settings in the RAM registers.

¹⁰ Watchdog reset is not available in System OFF.

Register	Offset	Description	
EVENTS_POFWARN	0x108	Power failure warning	
EVENTS_SLEEPENTER	0x114	CPU entered WFI/WFE sleep	
EVENTS_SLEEPEXIT	0x118	CPU exited WFI/WFE sleep	
EVENTS_USBDTECTED	0x11C	Voltage supply detected on VBUS	
EVENTS_USBREMOVED	0x120	Voltage supply removed from VBUS	
EVENTS_USBPWRDY	0x124	USB 3.3 V supply ready	
INTENSET	0x304	Enable interrupt	
INTENCLR	0x308	Disable interrupt	
RESETREAS	0x400	Reset reason	
RAMSTATUS	0x428	RAM status register	Deprecated
USBREGSTATUS	0x438	USB supply status	
SYSTEMOFF	0x500	System OFF register	
POFCON	0x510	Power-fail comparator configuration	
GPREGRET	0x51C	General purpose retention register	
GPREGRET2	0x520	General purpose retention register	
DCDCEN	0x578	Enable DC/DC converter for REG1 stage	
MAINREGSTATUS	0x640	Main supply status	
RAM[0].POWER	0x900	RAM0 power control register	
RAM[0].POWERSET	0x904	RAM0 power control set register	
RAM[0].POWERCLR	0x908	RAM0 power control clear register	
RAM[1].POWER	0x910	RAM1 power control register	
RAM[1].POWERSET	0x914	RAM1 power control set register	
RAM[1].POWERCLR	0x918	RAM1 power control clear register	
RAM[2].POWER	0x920	RAM2 power control register	
RAM[2].POWERSET	0x924	RAM2 power control set register	
RAM[2].POWERCLR	0x928	RAM2 power control clear register	
RAM[3].POWER	0x930	RAM3 power control register	
RAM[3].POWERSET	0x934	RAM3 power control set register	
RAM[3].POWERCLR	0x938	RAM3 power control clear register	
RAM[4].POWER	0x940	RAM4 power control register	
RAM[4].POWERSET	0x944	RAM4 power control set register	
RAM[4].POWERCLR	0x948	RAM4 power control clear register	
RAM[5].POWER	0x950	RAM5 power control register	
RAM[5].POWERSET	0x954	RAM5 power control set register	
RAM[5].POWERCLR	0x958	RAM5 power control clear register	
RAM[6].POWER	0x960	RAM6 power control register	
RAM[6].POWERSET	0x964	RAM6 power control set register	
RAM[6].POWERCLR	0x968	RAM6 power control clear register	
RAM[7].POWER	0x970	RAM7 power control register	
RAM[7].POWERSET	0x974	RAM7 power control set register	
RAM[7].POWERCLR	0x978	RAM7 power control clear register	
RAM[8].POWER	0x980	RAM8 power control register	
RAM[8].POWERSET	0x984	RAM8 power control set register	
RAM[8].POWERCLR	0x988	RAM8 power control clear register	

Table 17: Register overview

5.3.7.1 TASKS_CONSTLAT

Address offset: 0x78

Enable Constant Latency mode

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID																																								A			
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value		Description																																				
A	W	TASKS_CONSTLAT		Trigger		1		Enable Constant Latency mode																																			
								Trigger task																																			

5.3.7.2 TASKS_LOWPWR

Address offset: 0x7C

Enable Low-power mode (variable latency)

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID																																						A				
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value		Description																																			
A	W	TASKS_LOWPWR		Trigger		1		Enable Low-power mode (variable latency)																																		
								Trigger task																																		

5.3.7.3 EVENTS_POFWARN

Address offset: 0x108

Power failure warning

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A																															
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																															
A	RW	EVENTS_POFWARN		Power failure warning																															
		NotGenerated	0	Event not generated																															
		Generated	1	Event generated																															

5.3.7.4 EVENTS_SLEEPENTER

Address offset: 0x114

CPU entered WFI/WFE sleep

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	RW	EVENTS_SLEEPENTER		CPU entered WFI/WFE sleep																															
		NotGenerated	0	Event not generated																															
		Generated	1	Event generated																															

5.3.7.5 EVENTS_SLEEPEXIT

Address offset: 0x118

CPU exited WFI/WFE sleep

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID																																				A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																																
A	RW	EVENTS_SLEEPEXIT		CPU exited WFI/WFE sleep																																
		NotGenerated	0	Event not generated																																
		Generated	1	Event generated																																

5.3.7.6 EVENTS_USBDTECTED

Address offset: 0x11C

Voltage supply detected on VBUS

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID																																				A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																																
A	RW	EVENTS_USBDTECTED		Voltage supply detected on VBUS																																
		NotGenerated	0	Event not generated																																
		Generated	1	Event generated																																

5.3.7.7 EVENTS_USBREMOVED

Address offset: 0x120

Voltage supply removed from VBUS

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID																																							A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
ID	Acce Field	Value ID	Value	Description																																			
A	RW	EVENTS_USBREMOVED		Voltage supply removed from VBUS																																			
		NotGenerated	0	Event not generated																																			
		Generated	1	Event generated																																			

5.3.7.8 EVENTS_USBPWRRDY

Address offset: 0x124

USB 3.3 V supply ready

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID																																							A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
ID	Acce Field	Value ID	Value	Description																																			
A	RW	EVENTS_USBPWRRDY		USB 3.3 V supply ready																																			
		NotGenerated	0	Event not generated																																			
		Generated	1	Event generated																																			

5.3.7.9 INTENSET

Address offset: 0x304

Enable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			F E D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																													
A	RW	POFWARN			Write '1' to enable interrupt for event POFWARN																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
B	RW	SLEEPENTER			Write '1' to enable interrupt for event SLEEPENTER																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
C	RW	SLEEPEXIT			Write '1' to enable interrupt for event SLEEPEXIT																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
D	RW	USBDETECTED			Write '1' to enable interrupt for event USBDETECTED																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
E	RW	USBREMOVED			Write '1' to enable interrupt for event USBREMOVED																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
F	RW	USBPWRRDY			Write '1' to enable interrupt for event USBPWRRDY																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													

5.3.7.10 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			F E D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW POFWARN			Write '1' to disable interrupt for event POFWARN																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
B	RW SLEEPENTER			Write '1' to disable interrupt for event SLEEPENTER																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
C	RW SLEEPEXIT			Write '1' to disable interrupt for event SLEEPEXIT																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
D	RW USBDETECTED			Write '1' to disable interrupt for event USBDETECTED																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																			
ID																											F				E	D	C	B	A			
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																			
ID	Acce Field	Value ID	Value	Description																																		
		Enabled	1	Read: Enabled																																		
E	RW USBREMOVED			Write '1' to disable interrupt for event USBREMOVED																																		
		Clear	1	Disable																																		
		Disabled	0	Read: Disabled																																		
		Enabled	1	Read: Enabled																																		
F	RW USBPWRRDY			Write '1' to disable interrupt for event USBPWRRDY																																		
		Clear	1	Disable																																		
		Disabled	0	Read: Disabled																																		
		Enabled	1	Read: Enabled																																		

5.3.7.11 RESETREAS

Address offset: 0x400

Reset reason

Unless cleared, the RESETREAS register will be cumulative. A field is cleared by writing '1' to it. If none of the reset sources are flagged, this indicates that the chip was reset from the on-chip reset generator, which will indicate a power-on-reset or a brownout reset.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																			
ID																											I H G F E								D C B A			
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																			
ID	Acce Field	Value ID	Value	Description																																		
A	RW RESETPIN			Reset from pin-reset detected																																		
		NotDetected	0	Not detected																																		
		Detected	1	Detected																																		
B	RW DOG			Reset from watchdog detected																																		
		NotDetected	0	Not detected																																		
		Detected	1	Detected																																		
C	RW SREQ			Reset from soft reset detected																																		
		NotDetected	0	Not detected																																		
		Detected	1	Detected																																		
D	RW LOCKUP			Reset from CPU lock-up detected																																		
		NotDetected	0	Not detected																																		
		Detected	1	Detected																																		
E	RW OFF			Reset due to wake up from System OFF mode when wakeup is triggered from DETECT signal from GPIO																																		
		NotDetected	0	Not detected																																		
		Detected	1	Detected																																		
F	RW LPCOMP			Reset due to wake up from System OFF mode when wakeup is triggered from ANADETECT signal from LPCOMP																																		
		NotDetected	0	Not detected																																		
		Detected	1	Detected																																		
G	RW DIF			Reset due to wake up from System OFF mode when wakeup is triggered from entering into debug interface mode																																		
		NotDetected	0	Not detected																																		
		Detected	1	Detected																																		
H	RW NFC			Reset due to wake up from System OFF mode by NFC field detect																																		
		NotDetected	0	Not detected																																		

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID			I H G F E																		D C B A																
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
ID	Acce Field	Value ID	Value		Description																																
		Detected	1		Detected																																
I	RW	VBUS			Reset due to wake up from System OFF mode by VBUS rising into valid range																																
		NotDetected	0		Not detected																																
		Detected	1		Detected																																

5.3.7.12 RAMSTATUS (Deprecated)

Address offset: 0x428

RAM status register

Since this register is deprecated the following substitutions have been made: RAM block 0 is equivalent to a block comprising RAM0.S0 and RAM1.S0, RAM block 1 is equivalent to a block comprising RAM2.S0 and RAM3.S0, RAM block 2 is equivalent to a block comprising RAM4.S0 and RAM5.S0 and RAM block 3 is equivalent to a block comprising RAM6.S0 and RAM7.S0. A RAM block field will indicate ON as long as any of the RAM sections associated with a block are on.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A-D	R	RAMBLOCK[i] (i=0..3)		RAM block i is on or off/powering up																														
		Off	0	Off																														
		On	1	On																														

5.3.7.13 USBREGSTATUS

Address offset: 0x438

USB supply status

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	R	VBUSDETECT		VBUS input detection status (USBDETECTED and USBREMOVED events are derived from this information)																														
		NoVbus	0	VBUS voltage below valid threshold																														
		VbusPresent	1	VBUS voltage above valid threshold																														
B	R	OUTPUTRDY		USB supply output settling time elapsed																														
		NotReady	0	USBREG output settling time not elapsed																														
		Ready	1	USBREG output settling time elapsed (same information as USBPWRRDY event)																														

5.3.7.14 SYSTEMOFF

Address offset: 0x500

System OFF register

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W SYSTEMOFF			Enable System OFF mode																															
		Enter	1	Enable System OFF mode																															

5.3.7.15 POFCON

Address offset: 0x510

Power-fail comparator configuration

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																
ID																											D D D D				B B B B				A
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																
ID	Acce Field	Value ID	Value	Description																															
A	RW POF			Enable or disable power failure warning																															
		Disabled	0	Disable																															
		Enabled	1	Enable																															
B	RW THRESHOLD			Power-fail comparator threshold setting. This setting applies both for normal voltage mode (supply connected to both VDD and VDDH) and high voltage mode (supply connected to VDDH only). Values 0-3 set threshold below 1.7 V and should not be used as brown out detection will be activated before power failure warning on such low voltages.																															
		V17	4	Set threshold to 1.7 V																															
		V18	5	Set threshold to 1.8 V																															
		V19	6	Set threshold to 1.9 V																															
		V20	7	Set threshold to 2.0 V																															
		V21	8	Set threshold to 2.1 V																															
		V22	9	Set threshold to 2.2 V																															
		V23	10	Set threshold to 2.3 V																															
		V24	11	Set threshold to 2.4 V																															
		V25	12	Set threshold to 2.5 V																															
		V26	13	Set threshold to 2.6 V																															
		V27	14	Set threshold to 2.7 V																															
		V28	15	Set threshold to 2.8 V																															
D	RW THRESHOLDVDDH			Power-fail comparator threshold setting for high voltage mode (supply connected to VDDH only). This setting does not apply for normal voltage mode (supply connected to both VDD and VDDH).																															
		V27	0	Set threshold to 2.7 V																															
		V28	1	Set threshold to 2.8 V																															
		V29	2	Set threshold to 2.9 V																															
		V30	3	Set threshold to 3.0 V																															
		V31	4	Set threshold to 3.1 V																															
		V32	5	Set threshold to 3.2 V																															
		V33	6	Set threshold to 3.3 V																															
		V34	7	Set threshold to 3.4 V																															
		V35	8	Set threshold to 3.5 V																															
		V36	9	Set threshold to 3.6 V																															
		V37	10	Set threshold to 3.7 V																															
		V38	11	Set threshold to 3.8 V																															

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																						D	D	D	D					B	B	B	B	A	
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																															
		V39	12	Set threshold to 3.9 V																															
		V40	13	Set threshold to 4.0 V																															
		V41	14	Set threshold to 4.1 V																															
		V42	15	Set threshold to 4.2 V																															

5.3.7.16 GPREGRET

Address offset: 0x51C

General purpose retention register

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID																														A	A	A	A	A	A	A	A						
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value				Description																																		
A	RW		GPREGRET						General purpose retention register																																		
					This register is a retained register																																						

5.3.7.17 GPREGRET2

Address offset: 0x520

General purpose retention register

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
ID																														A										A	A	A	A	A	A	A					
Reset 0x00000000										0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value				Description																																										
A	RW		GPREGRET						General purpose retention register																																										
										This register is a retained register																																									

5.3.7.18 DCDCCEN

Address offset: 0x578

Enable DC/DC converter for REG1 stage

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID																																							A
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																			
ID	Acce Field		Value ID	Value		Description																																	
A	RW DCDCCEN					Enable DC/DC converter for REG1 stage.																																	
			Disabled	0		Disable																																	
			Enabled	1		Enable																																	

5.3.7.19 MAINREGSTATUS

Address offset: 0x640

Main supply status

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID																																				A		
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
ID	Acce Field		Value ID	Value		Description																																
A	R	MAINREGSTATUS				Main supply status																																
				Normal		0	Normal voltage mode. Voltage supplied on VDD.																															
				High		1	High voltage mode. Voltage supplied on VDDH.																															

5.3.7.20 RAM[n].POWER (n=0..8)

Address offset: 0x900 + (n × 0x10)

RAMn power control register

Bit number		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID		f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x0000FFFF		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field	Value ID		Value	Description																												
A-P	RW	S[i]POWER (i=0..15)			Keep RAM section Si on or off in System ON mode.																												
		Off		0	Off																												
		On		1	On																												
Q-f	RW	S[i]RETENTION (i=0..15)			Keep retention on RAM section Si when RAM section is off																												
		Off		0	Off																												
		On		1	On																												

5.3.7.21 RAM[n].POWERSET (n=0..8)

Address offset: 0x904 + (n × 0x10)

RAMn power control set register

When read, this register will return the value of the POWER register.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID			f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A				
Reset 0x0000FFFF			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
ID	Acce Field	Value ID	Value		Description																																	
A-P	W	S[i]POWER (i=0..15)			Keep RAM section Si of RAMn on or off in System ON mode																																	
		On	1		On																																	
Q-f	W	S[i]RETENTION (i=0..15)			Keep retention on RAM section Si when RAM section is switched off																																	
		On	1		On																																	

5.3.7.22 RAM[n].POWERCLR (n=0..8)

Address offset: 0x908 + (n × 0x10)

RAMn power control clear register

When read, this register will return the value of the POWER register.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID				f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A				
Reset 0x0000FFFF				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				
ID	Acce	Field	Value	ID	Value	Description																																	
A-P	W	S[i]POWER (i=0..15)				Keep RAM section Si of RAMn on or off in System ON mode																																	
			Off		1	Off																																	
Q-f	W	S[i]RETENTION (i=0..15)				Keep retention on RAM section Si when RAM section is																																	
						switched off																																	
			Off		1	Off																																	

5.3.8 Electrical specification

5.3.8.1 Regulator operating conditions

Symbol	Description	Min.	Typ.	Max.	Units
V _{DD,POR}	VDD supply voltage needed during power-on reset	1.75			V
V _{DD}	Normal voltage mode operating voltage	1.7	3.0	3.6	V
V _{DDH}	High voltage mode operating voltage	2.5	3.7	5.5	V
C _{VDD}	Effective decoupling capacitance on the VDD pin	2.7	4.7	5.5	μF
C _{DEC4}	Effective decoupling capacitance on the DEC4 pin	0.7	1	1.3	μF

5.3.8.2 Regulator specifications, REG0 stage

Symbol	Description	Min.	Typ.	Max.	Units
V _{REG0OUT}	REG0 output voltage	1.8		3.3	V
V _{REG0OUT,ERR}	REG0 output voltage error (deviation from setting in REG0OUT on page 43)	-10		5	%
V _{REG0OUT,ERR,EXT}	REG0 output voltage error (deviation from setting in REG0OUT on page 43), extended temperature range	-10		7	%
V _{VDDH-VDD}	Required difference between input voltage (VDDH) and output voltage (VDD, configured in REG0OUT on page 43), VDDH > VDD	0.3			V

5.3.8.3 Device startup times

Symbol	Description	Min.	Typ.	Max.	Units
t _{POR}	Time in power-on reset after supply reaches minimum operating voltage, depending on supply rise time				
t _{POR,10μs}	VDD rise time 10 μs ¹¹		1	10	ms
t _{POR,10ms}	VDD rise time 10 ms ¹¹		9		ms
t _{POR,60ms}	VDD rise time 60 ms ¹¹		23	110	ms
t _{RISE,REG0OUT}	REG0 output (VDD) rise time after VDDH reaches minimum VDDH supply voltage ¹¹				
t _{RISE,REG0OUT,10μs}	VDDH rise time 10 μs ¹¹		0.22	1.55	ms
t _{RISE,REG0OUT,10ms}	VDDH rise time 10 ms ¹¹		5		ms
t _{RISE,REG0OUT,100ms}	VDDH rise time 100 ms ¹¹	30	50	80	ms
t _{PINR}	Reset time when using pin reset, depending on pin capacitance				
t _{PINR,500nF}	500 nF capacitance at reset pin			32.5	ms

¹¹ See [Recommended operating conditions](#) on page 608 for more information.

Symbol	Description	Min.	Typ.	Max.	Units
t _{PINR,10μF}	10 μF capacitance at reset pin			650	ms
t _{R2ON}	Time from power-on reset to System ON				
t _{R2ON,NOTCONF}	If reset pin not configured	tPOR			ms
t _{R2ON,CONF}	If reset pin configured	tPOR + tPINR			ms
t _{OFF2ON}	Time from OFF to CPU execute		16.5		μs
t _{IDLE2CPU}	Time from IDLE to CPU execute		3.0		μs
t _{EVTSET,CL1}	Time from HW event to PPI event in Constant Latency System ON mode		0.0625		μs
t _{EVTSET,CL0}	Time from HW event to PPI event in Low Power System ON mode		0.0625		μs

5.3.8.4 Power fail comparator

Symbol	Description	Min.	Typ.	Max.	Units
V _{POF,NV}	Nominal power level warning thresholds (falling supply voltage) in Normal voltage mode (supply on VDD). Levels are configurable between Min. and Max. in 100 mV increments	1.7		2.8	V
V _{POF,HV}	Nominal power level warning thresholds (falling supply voltage) in High voltage mode (supply on VDDH). Levels are configurable in 100 mV increments	2.7		4.2	V
V _{POFTOL}	Threshold voltage tolerance (applies in both Normal voltage mode and High voltage mode)	-5		5	%
V _{POFHYST}	Threshold voltage hysteresis (applies in both Normal voltage mode and High voltage mode)	40	50	60	mV
V _{BOR,OFF}	Brownout reset voltage range System OFF mode. Brownout only applies to the voltage on VDD	1.2		1.62	V
V _{BOR,ON}	Brownout reset voltage range System ON mode. Brownout only applies to the voltage on VDD	1.57	1.6	1.63	V

5.3.8.5 USB operating conditions

Symbol	Description	Min.	Typ.	Max.	Units
V _{BUS}	Supply voltage on VBUS pin	4.35	5	5.5	V
V _{DPDM}	Voltage on D+ and D- lines	VSS - 0.3		VUSB33 + 0.3	V

5.3.8.6 USB regulator specifications

Symbol	Description	Min.	Typ.	Max.	Units
I _{USB,QUIES}	USB regulator quiescent current drawn from VBUS (USB enabled)		170		μA
t _{USBPWRRDY}	Time from USB enabled to USBPWRRDY event triggered, V _{BUS} supply provided		1		ms
V _{USB33}	On voltage at the USB regulator output (DECUSB pin)	3.0	3.3	3.6	V
R _{SOURCE,VBUS}	Maximum source resistance on VBUS, including cable			2	Ω
C _{DECUSB}	Decoupling capacitor on the DECUSB pin	2.35	4.7	5.5	μF

5.3.8.7 VBUS detection specifications

Symbol	Description	Min.	Typ.	Max.	Units
V _{BUS,DETECT}	Voltage at which rising VBUS gets reported by USBDETECTED	3.4	4.0	4.3	V
V _{BUS,REMOVE}	Voltage at which decreasing VBUS gets reported by USBREMOVED	3.0	3.6	3.9	V

5.4 CLOCK — Clock control

The clock control system can source the system clocks from a range of internal or external high and low frequency oscillators and distribute them to modules based upon a module's individual requirements. Clock distribution is automated and grouped independently by module to limit current consumption in unused branches of the clock tree.

Listed here are the main features for CLOCK:

- 64 MHz on-chip oscillator
- 64 MHz crystal oscillator, using external 32 MHz crystal
- 32.768 kHz +/-500 ppm RC oscillator
- 32.768 kHz crystal oscillator, using external 32.768 kHz crystal
- 32.768 kHz oscillator synthesized from 64 MHz oscillator
- Firmware (FW) override control of crystal oscillator activity for low latency start up
- Automatic internal oscillator and clock control, and distribution for ultra-low power

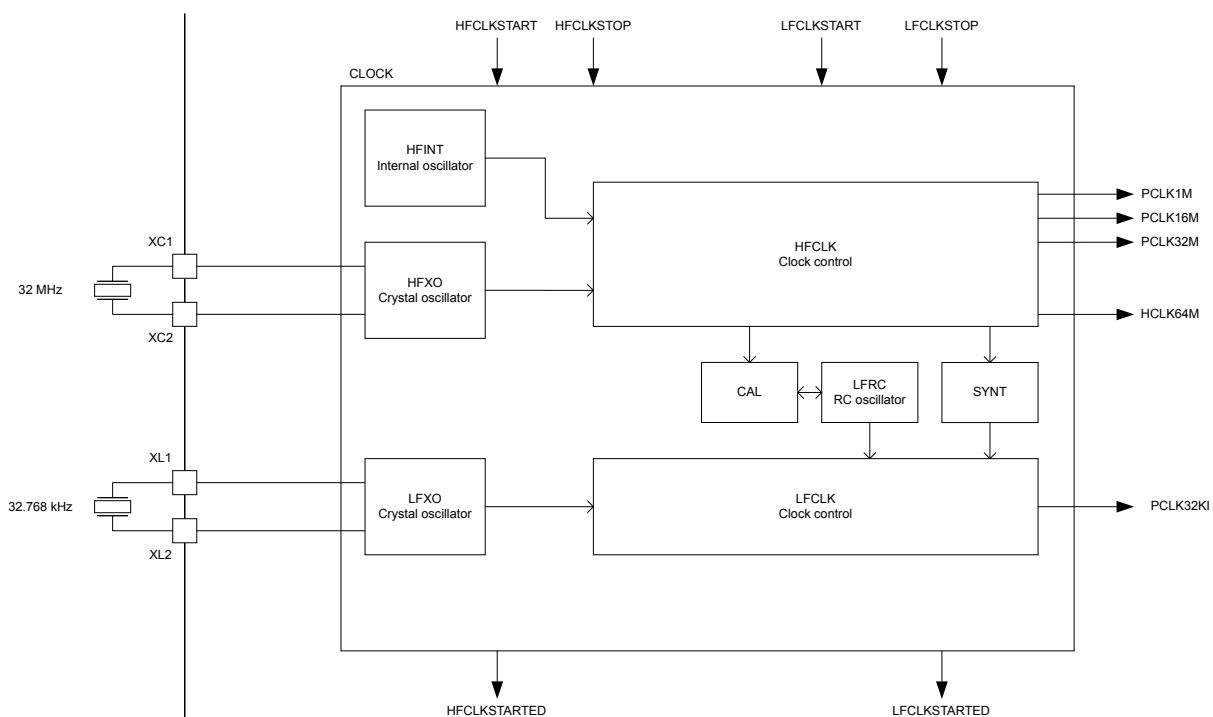


Figure 20: Clock control

5.4.1 HFCLK controller

The HFCLK controller provides several clock signals in the system.

These are as follows:

- HCLK64M: 64 MHz CPU clock
- PCLK1M: 1 MHz peripheral clock
- PCLK16M: 16 MHz peripheral clock
- PCLK32M: 32 MHz peripheral clock

The HFCLK controller uses the following high frequency clock (HFCLK) sources:

- 64 MHz internal oscillator (HFINT)
- 64 MHz crystal oscillator (HFXO)

For illustration, see [Clock control](#) on page 80.

The HFCLK controller will automatically provide the clock(s) requested by the system. If the system does not request any clocks from the HFCLK controller, the controller will enter a power saving mode.

The HFINT source will be used when HFCLK is requested and HFXO has not been started.

The HFXO is started by triggering the HFCLKSTART task and stopped by triggering the HFCLKSTOP task. When the HFCLKSTART task is triggered, the HFCLKSTARTED event is generated once the HFXO startup time has elapsed. The HFXO startup time is given as the sum of the following:

- HFXO power-up time, as specified in [64 MHz crystal oscillator \(HFXO\)](#) on page 93.
- HFXO debounce time, as specified in register [HFXODEBOUNCE](#) on page 92.

The HFXO must be running to use the RADIO or the calibration mechanism associated with the 32.768 kHz RC oscillator.

5.4.1.1 64 MHz crystal oscillator (HFXO)

The 64 MHz crystal oscillator (HFXO) is controlled by a 32 MHz external crystal.

The crystal oscillator is designed for use with an AT-cut quartz crystal in parallel resonant mode. To achieve correct oscillation frequency, the load capacitance must match the specification in the crystal data sheet.

[Circuit diagram of the 64 MHz crystal oscillator](#) on page 81 shows how the 32 MHz crystal is connected to the 64 MHz crystal oscillator.

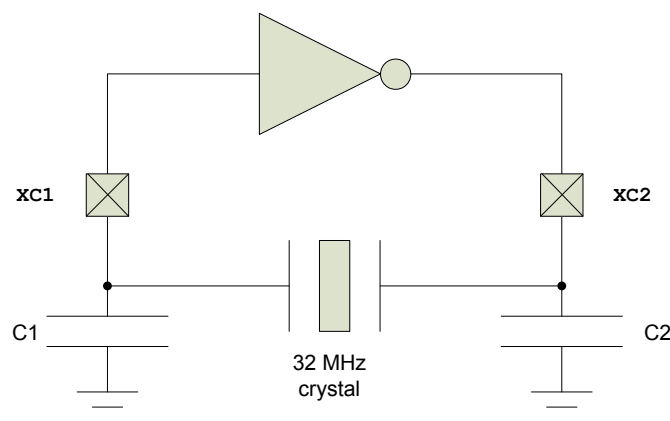


Figure 21: Circuit diagram of the 64 MHz crystal oscillator

The load capacitance (CL) is the total capacitance seen by the crystal across its terminals and is given by:

$$CL = \frac{(C1' \cdot C2')}{(C1' + C2')}$$

$$C1' = C1 + C_{pcb1} + C_{pin}$$

$$C2' = C2 + C_{pcb2} + C_{pin}$$

C1 and C2 are ceramic SMD capacitors connected between each crystal terminal and ground. For more information, see [Reference circuitry](#) on page 567. C_{pcb1} and C_{pcb2} are stray capacitances on the PCB. C_{pin} is the pin input capacitance on the XC1 and XC2 pins. See table [64 MHz crystal oscillator \(HFXO\)](#) on page 93. The load capacitors C1 and C2 should have the same value.

For reliable operation, the crystal load capacitance, shunt capacitance, equivalent series resistance, and drive level must comply with the specifications in table [64 MHz crystal oscillator \(HFXO\)](#) on page 93. It is recommended to use a crystal with lower than maximum load capacitance and/or shunt capacitance. A low load capacitance will reduce both start up time and current consumption.

5.4.2 LFCLK controller

The system supports several low frequency clock sources.

As illustrated in [Clock control](#) on page 80, the system supports the following low frequency clock sources:

- 32.768 kHz RC oscillator (LFRC)
- 32.768 kHz crystal oscillator (LFXO)
- 32.768 kHz synthesized from HFCLK (LFSYNT)

The LFCLK controller and all of the LFCLK clock sources are always switched off when in System OFF mode.

The LFCLK clock is started by first selecting the preferred clock source in register [LFCLKSRC](#) on page 91 and then triggering the LFCLKSTART task. If the LFXO is selected as the clock source, the LFCLK will initially start running from the 32.768 kHz LFRC while the LFXO is starting up and automatically switch to using the LFXO once this oscillator is running. The LFCLKSTARTED event will be generated when the LFXO has been started.

The register [LFXODEBOUNCE](#) on page 92 is used to configure the LFXO debounce time. The register must be modified if operating in the Extended Operating Conditions temperature range, see [Recommended operating conditions](#) on page 608. The LFXO start up time will be increased as a result.

The LFCLK clock is stopped by triggering the LFCLKSTOP task.

Register [LFCLKSRC](#) on page 91 controls the clock source, and its allowed swing. The truth table for various situations is as follows:

SRC	EXTERNAL	BYPASS	Comment
0	0	0	Normal operation, LFRC is source
0	0	1	DO NOT USE
0	1	X	DO NOT USE
1	0	0	Normal XTAL operation
1	1	0	Apply external low swing signal to XL1, ground XL2
1	1	1	Apply external full swing signal to XL1, leave XL2 grounded or unconnected
1	0	1	DO NOT USE
2	0	0	Normal operation, LFSYNT is source
2	0	1	DO NOT USE
2	1	X	DO NOT USE

Table 18: LFCLKSRC configuration depending on clock source

It is not allowed to write to register **LFCLKSRC** on page 91 when the LFCLK is running.

A LFCLKSTOP task will stop the LFCLK oscillator. However, the LFCLKSTOP task can only be triggered after the STATE field in register **LFCLKSTAT** on page 91 indicates LFCLK running state.

The synthesized 32.768 kHz clock depends on the HFCLK to run. If high accuracy is required for the LFCLK running off the synthesized 32.768 kHz clock, the HFCLK must running from the HFXO source.

5.4.2.1 32.768 kHz RC oscillator (LFRC)

The default source of the low frequency clock (LFCLK) is the 32.768 kHz RC oscillator (LFRC).

The LFRC oscillator does not require additional external components.

The LFRC frequency will be affected by variation in temperature. The LFRC oscillator can be calibrated to improve accuracy by using the HFXO as a reference oscillator during calibration.

5.4.2.2 Calibrating the 32.768 kHz RC oscillator

After the LFRC oscillator is started and running, it can be calibrated by triggering the CAL task.

The LFRC oscillator will then temporarily request the HFCLK to be used as a reference for the calibration. A DONE event will be generated when calibration has finished. The HFCLK crystal oscillator has to be started (by triggering the HFCLKSTART task) in order for the calibration mechanism to work.

It is not allowed to stop the LFRC during an ongoing calibration.

5.4.2.3 Calibration timer

The calibration timer can be used to time the calibration interval of the 32.768 kHz RC oscillator.

The calibration timer is started by triggering the CTSTART task and stopped by triggering the CTSTOP task. The calibration timer will always start counting down from the value specified in **CTIV (Retained)** on page 93 and generate a CTTO event when it reaches 0. The calibration timer will automatically stop when it reaches 0.

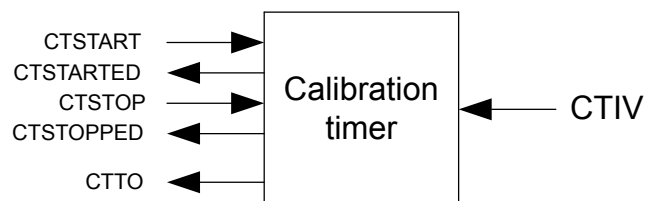


Figure 22: Calibration timer

After a CTSTART task has been triggered, the calibration timer will ignore further tasks until it has returned the CTSTARTED event. Likewise, after a CTSTOP task has been triggered, the calibration timer will ignore further tasks until it has returned a CTSTOPPED event. Triggering CTSTART while the calibration timer

is running will immediately return a CTSTARTED event. Triggering CTSTOP when the calibration timer is stopped will immediately return a CTSTOPPED event.

5.4.2.4 32.768 kHz crystal oscillator (LFXO)

For higher LFCLK accuracy (when better than +/- 500 ppm accuracy is required), the low frequency crystal oscillator (LFXO) must be used.

The following external clock sources are supported:

- Low swing clock signal applied to the XL1 pin. The XL2 pin shall then be grounded.
- Rail-to-rail clock signal applied to the XL1 pin. The XL2 pin shall then be grounded or left unconnected.

To achieve correct oscillation frequency, the load capacitance must match the specification in the crystal data sheet. [Circuit diagram of the 32.768 kHz crystal oscillator](#) on page 84 shows the LFXO circuitry.

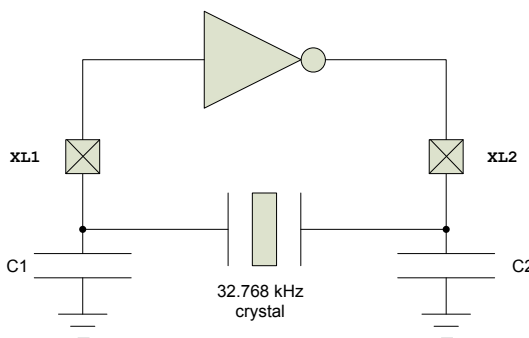


Figure 23: Circuit diagram of the 32.768 kHz crystal oscillator

The load capacitance (CL) is the total capacitance seen by the crystal across its terminals and is given by:

$$CL = \frac{(C1' \cdot C2')}{(C1' + C2')}$$

$$C1' = C1 + C_{pcb1} + C_{pin}$$

$$C2' = C2 + C_{pcb2} + C_{pin}$$

C1 and C2 are ceramic SMD capacitors connected between each crystal terminal and ground. C_{pcb1} and C_{pcb2} are stray capacitances on the PCB. C_{pin} is the pin input capacitance on the XC1 and XC2 pins (see [Low frequency crystal oscillator \(LFXO\)](#) on page 94). The load capacitors C1 and C2 should have the same value.

For more information, see [Reference circuitry](#) on page 567.

5.4.2.5 32.768 kHz synthesized from HFCLK (LFSYNT)

LFCLK can also be synthesized from the HFCLK clock source. The accuracy of LFCLK will then be the accuracy of the HFCLK.

Using the LFSYNT clock avoids the requirement for a 32.768 kHz crystal, but increases average power consumption as the HFCLK will need to be requested in the system.

5.4.3 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40000000	CLOCK	CLOCK	Clock control	

Table 19: Instances

Register	Offset	Description	
TASKS_HFCLKSTART	0x000	Start HFXO crystal oscillator	
TASKS_HFCLKSTOP	0x004	Stop HFXO crystal oscillator	
TASKS_LFCLKSTART	0x008	Start LFCLK	
TASKS_LFCLKSTOP	0x00C	Stop LFCLK	
TASKS_CAL	0x010	Start calibration of LFRC	
TASKS_CTSTART	0x014	Start calibration timer	
TASKS_CTSTOP	0x018	Stop calibration timer	
EVENTS_HFCLKSTARTED	0x100	HFXO crystal oscillator started	
EVENTS_LFCLKSTARTED	0x104	LFCLK started	
EVENTS_DONE	0x10C	Calibration of LFRC completed	
EVENTS_CTTO	0x110	Calibration timer timeout	
EVENTS_CTSTARTED	0x128	Calibration timer has been started and is ready to process new tasks	
EVENTS_CTSTOPPED	0x12C	Calibration timer has been stopped and is ready to process new tasks	
INTENSET	0x304	Enable interrupt	
INTENCLR	0x308	Disable interrupt	
HFCLKRUN	0x408	Status indicating that HFCLKSTART task has been triggered	
HFCLKSTAT	0x40C	HFCLK status	
LFCLKRUN	0x414	Status indicating that LFCLKSTART task has been triggered	
LFCLKSTAT	0x418	LFCLK status	
LFCLKSRCCOPY	0x41C	Copy of LFCLKSRC register, set when LFCLKSTART task was triggered	
LFCLKSRC	0x518	Clock source for the LFCLK	
HFXODEBOUNCE	0x528	HFXO debounce time. The HFXO is started by triggering the TASKS_HFCLKSTART task.	
LFXODEBOUNCE	0x52C	LFXO debounce time. The LFXO is started by triggering the TASKS_LFCLKSTART task when the LFCLKSRC register is configured for Xtal.	
CTIV	0x538	Calibration timer interval	Retained
TRACECONFIG	0x55C	Clocking options for the trace port debug interface	

Table 20: Register overview

5.4.3.1 TASKS_HFCLKSTART

Address offset: 0x000

Start HFXO crystal oscillator

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field		Value ID	Value	Description																														
A	W	TASKS_HFCLKSTART			Start HFXO crystal oscillator																														
			Trigger	1	Trigger task																														

5.4.3.2 TASKS_HFCLKSTOP

Address offset: 0x004

Stop HFXO crystal oscillator

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W	TASKS_HFCLKSTOP		Stop HFXO crystal oscillator																															
		Trigger	1	Trigger task																															

5.4.3.3 TASKS_LFCLKSTART

Address offset: 0x008

Start LFCLK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W	TASKS_LFCLKSTART		Start LFCLK																															
		Trigger	1	Trigger task																															

5.4.3.4 TASKS_LFCLKSTOP

Address offset: 0x00C

Stop LFCLK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W	TASKS_LFCLKSTOP		Stop LFCLK																															
		Trigger	1	Trigger task																															

5.4.3.5 TASKS_CAL

Address offset: 0x010

Start calibration of LFRC

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W	TASKS_CAL		Start calibration of LFRC																															
		Trigger	1	Trigger task																															

5.4.3.6 TASKS_CTSTART

Address offset: 0x014

Start calibration timer

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID																																								A			
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value		Description																																				
A	W	TASKS_CTSTART						Start calibration timer																																			
				Trigger	1		Trigger task																																				

5.4.3.7 TASKS_CTSTOP

Address offset: 0x018

Stop calibration timer

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID																																								A		
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field			Value ID			Value			Description																																
A	W	TASKS_CTSTOP									Stop calibration timer																															
					Trigger			1			Trigger task																															

5.4.3.8 EVENTS_HFCLKSTARTED

Address offset: 0x100

HFXO crystal oscillator started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	RW	EVENTS_HFCLKSTARTED		HF XO crystal oscillator started																															
		NotGenerated	0	Event not generated																															
		Generated	1	Event generated																															

5.4.3.9 EVENTS_LFCLKSTARTED

Address offset: 0x104

LFCLK started

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	RW	EVENTS_LFCLKSTARTED		LFCLK started																															
		NotGenerated	0	Event not generated																															
		Generated	1	Event generated																															

5.4.3.10 EVENTS_DONE

Address offset: 0x10C

Calibration of LFRC completed

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID																																							A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
ID	Acce Field	Value ID	Value	Description																																			
A	RW	EVENTS_DONE		Calibration of LFRC completed																																			
		NotGenerated	0	Event not generated																																			
		Generated	1	Event generated																																			

5.4.3.11 EVENTS_CTTO

Address offset: 0x110

Calibration timer timeout

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID																																								A				
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value		Description																																					
A	RW	EVENTS_CTTO				Calibration timer timeout																																						
				NotGenerated		0	Event not generated																																					
				Generated		1	Event generated																																					

5.4.3.12 EVENTS_CTSTARTED

Address offset: 0x128

Calibration timer has been started and is ready to process new tasks

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A																															
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																															
A	RW	EVENTS_CTSTARTED		Calibration timer has been started and is ready to process new tasks																															
		NotGenerated	0	Event not generated																															
		Generated	1	Event generated																															

5.4.3.13 EVENTS_CTSTOPPED

Address offset: 0x12C

Calibration timer has been stopped and is ready to process new tasks

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID				A																																			
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	Acce Field		Value ID	Value		Description																																	
A	RW		EVENTS_CTSTOPPED		Calibration timer has been stopped and is ready to process new tasks																																		
			NotGenerated	0		Event not generated																																	
			Generated	1		Event generated																																	

5.4.3.14 INTENSET

Address offset: 0x304

Enable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			F E D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																													
A	RW	HFCLKSTARTED			Write '1' to enable interrupt for event HFCLKSTARTED																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
B	RW	LFCLKSTARTED			Write '1' to enable interrupt for event LFCLKSTARTED																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
C	RW	DONE			Write '1' to enable interrupt for event DONE																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
D	RW	CTTO			Write '1' to enable interrupt for event CTTO																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
E	RW	CTSTARTED			Write '1' to enable interrupt for event CTSTARTED																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
F	RW	CTSTOPPED			Write '1' to enable interrupt for event CTSTOPPED																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													

5.4.3.15 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																	
ID																											F	E					D	C	B	A
Reset 0x00000000			0 0																																	

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID																											F	E					D	C			B	A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
ID	Acce Field	Value ID	Value		Description																																	
		Clear	1		Disable																																	
		Disabled	0		Read: Disabled																																	
		Enabled	1		Read: Enabled																																	
E	RW	CTSTARTED		Write '1' to disable interrupt for event CTSTARTED																																		
		Clear	1		Disable																																	
		Disabled	0		Read: Disabled																																	
		Enabled	1		Read: Enabled																																	
F	RW	CTSTOPPED		Write '1' to disable interrupt for event CTSTOPPED																																		
		Clear	1		Disable																																	
		Disabled	0		Read: Disabled																																	
		Enabled	1		Read: Enabled																																	

5.4.3.16 HFCLKRUN

Address offset: 0x408

Status indicating that HFCLKSTART task has been triggered

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID																																								A					
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value		Description																																						
A	R	STATUS						HFCLKSTART task triggered or not																																					
				NotTriggered		0		Task not triggered																																					
				Triggered		1		Task triggered																																					

5.4.3.17 HFCLKSTAT

Address offset: 0x40C

HFCLK status

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID										B																				A														
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value		Description																																					
A	R	SRC				Source of HFCLK																																						
				RC		0	64 MHz internal oscillator (HFINT)																																					
				Xtal		1	64 MHz crystal oscillator (HFXO)																																					
B	R	STATE				HFCLK state																																						
				NotRunning		0	HFCLK not running																																					
				Running		1	HFCLK running																																					

5.4.3.18 LFCLKRUN

Address offset: 0x414

Status indicating that LFCLKSTART task has been triggered

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID																																								A				
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value		Description																																					
A	R	STATUS		NotTriggered		0		LFCLKSTART task triggered or not Task not triggered																																				
				Triggered		1		Task triggered																																				

5.4.3.19 LFCLKSTAT

Address offset: 0x418

LFCLK status

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID				B																												A	A					
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
ID	Acce	Field	Value ID	Value	Description																																	
A	R	SRC			Source of LFCLK																																	
			RC	0	32.768 kHz RC oscillator (LFRC)																																	
			Xtal	1	32.768 kHz crystal oscillator (LFXO)																																	
			Synth	2	32.768 kHz synthesized from HFCLK (LFSYNT)																																	
B	R	STATE			LFCLK state																																	
			NotRunning	0	LFCLK not running																																	
			Running	1	LFCLK running																																	

5.4.3.20 LFCLKSRCCOPY

Address offset: 0x41C

Copy of LFCLKSRC register, set when LFCLKSTART task was triggered

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID																														A	A												
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value		Description																																				
A	R	SRC						Clock source																																			
				RC	0		32.768 kHz RC oscillator (LFRC)																																				
				Xtal	1		32.768 kHz crystal oscillator (LFXO)																																				
				Synth	2		32.768 kHz synthesized from HFCLK (LFSYNT)																																				

5.4.3.21 LFCLKSRC

Address offset: 0x518

Clock source for the LFCLK

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			C B A A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW SRC			Clock source																														
		RC	0	32.768 kHz RC oscillator (LFRC)																														
		Xtal	1	32.768 kHz crystal oscillator (LFXO)																														
		Synth	2	32.768 kHz synthesized from HFCLK (LFSYNT)																														
B	RW BYPASS			Enable or disable bypass of LFCLK crystal oscillator with external clock source																														
		Disabled	0	Disable (use with Xtal or low-swing external source)																														
		Enabled	1	Enable (use with rail-to-rail external source)																														
C	RW EXTERNAL			Enable or disable external source for LFCLK																														
		Disabled	0	Disable external source (use with Xtal)																														
		Enabled	1	Enable use of external source instead of Xtal (SRC needs to be set to Xtal)																														

5.4.3.22 HFXODEBOUNCE

Address offset: 0x528

HFXO debounce time. The HFXO is started by triggering the TASKS_HFCLKSTART task.

The EVENTS_HFCLKSTARTED event is generated after the HFXO power up time + the HFXO debounce time has elapsed. It is not allowed to change the value of this register while the HFXO is starting.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A A A A A A																															
Reset 0x00000010			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW HFXODEBOUNCE		0x01..0xFF	HFXO debounce time. Debounce time = HFXODEBOUNCE * 16 μs.																														
		Db256us	0x10	256 μs debounce time. Recommended for 1.6 mm x 2.0 mm crystals and larger.																														
		Db1024us	0x40	1024 μs debounce time. Recommended for 1.6 mm x 1.2 mm crystals and smaller.																														

5.4.3.23 LFXODEBOUNCE

Address offset: 0x52C

LFXO debounce time. The LFXO is started by triggering the TASKS_LFCLKSTART task when the LFCLKSRC register is configured for Xtal.

The EVENTS_LFCLKSTARTED event is generated after the LFXO debounce time has elapsed. It is not allowed to change the value of this register while the LFXO is starting.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW LFXODEBOUNCE			LFXO debounce time.																														
		Normal	0	8192 32.768 kHz periods, or 0.25 s. Recommended for normal Operating Temperature conditions.																														
		Extended	1	16384 32.768 kHz periods, or 0.5 s. Recommended for Extended Operating Temperature conditions.																														

5.4.3.24 CTIV (Retained)

Address offset: 0x538

This register is a retained register

Calibration timer interval

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
ID																												A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A

5.4.3.25 TRACECONFIG

Address offset: 0x55C

Clocking options for the trace port debug interface

This register is a retained register. Reset behavior is the same as debug components.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																											
ID			B																B	A										A																															
Reset 0x00000000			0																																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce	Field	Value		ID	Value	Description																																																						
A	RW	TRACEPORTSPEED					Speed of trace port clock. Note that the TRACECLK pin will output this clock divided by two.																																																						
			32MHz	0	32 MHz trace port clock (TRACECLK = 16 MHz)																																																								
			16MHz	1	16 MHz trace port clock (TRACECLK = 8 MHz)																																																								
			8MHz	2	8 MHz trace port clock (TRACECLK = 4 MHz)																																																								
			4MHz	3	4 MHz trace port clock (TRACECLK = 2 MHz)																																																								
B	RW	TRACEMUX					Pin multiplexing of trace signals. See pin assignment chapter for more details.																																																						
			GPIO	0	No trace signals routed to pins. All pins can be used as regular GPIOs.																																																								
			Serial	1	SWO trace signal routed to pin. Remaining pins can be used as regular GPIOs.																																																								
			Parallel	2	All trace signals (TRACECLK and TRACEDATA[n]) routed to pins.																																																								

5.4.4 Electrical specification

5.4.4.1 64 MHz internal oscillator (HFINT)

Symbol	Description	Min.	Typ.	Max.	Units
f_{NOM_HFINT}	Nominal output frequency		64		MHz
f_{TOL_HFINT}	Frequency tolerance		±1.5	±8	%
$f_{TOL_HFINT,EXT}$	Frequency tolerance, extended temperature range			±9	%

5.4.4.2 64 MHz crystal oscillator (HFXO)

Symbol	Description	Min.	Typ.	Max.	Units
$f_{\text{NOM_HFXO}}$	Nominal output frequency		64		MHz
$f_{\text{XTAL_HFXO}}$	External crystal frequency		32		MHz
$f_{\text{TOL_HFXO}}$	Frequency tolerance requirement for 2.4 GHz proprietary radio applications			±60	ppm
$f_{\text{TOL_HFXO_BLE}}$	Frequency tolerance requirement, Bluetooth low energy applications, packet length ≤ 200 bytes			±40	ppm
$f_{\text{TOL_HFXO_BLE_LP}}$	Frequency tolerance requirement, Bluetooth low energy applications, packet length > 200 bytes			±30	ppm
$C_{\text{L_HFXO}}$	Load capacitance			12	pF
$C_{\text{O_HFXO}}$	Shunt capacitance			7	pF
$R_{\text{S_HFXO_7PF}}$	Equivalent series resistance 3 pF < $C_{\text{O}} \leq 7$ pF			60	Ω
$R_{\text{S_HFXO_3PF}}$	Equivalent series resistance $C_{\text{O}} \leq 3$ pF			100	Ω
$P_{\text{D_HFXO}}$	Drive level			100	μW
$C_{\text{PIN_HFXO}}$	Input capacitance XC1 and XC2		3		pF
$I_{\text{STBY_X32M}}$	Core standby current for various crystals				
$I_{\text{STBY_X32M_X0}}$	Typical parameters for a given 2.5mm x 2.0mm crystal: $CL_{\text{HFXO}} = 8$ pF, $CO_{\text{HFXO}} = 1$ pF, $LM_{\text{HFXO}} = 7$ mH, $RS_{\text{HFXO}} = 20$ Ω		65		μA
$I_{\text{STBY_X32M_X1}}$	Typical parameters for a given 1.6mm x 1.2mm crystal: $CL_{\text{HFXO}} = 8$ pF, $CO_{\text{HFXO}} = 0.4$ pF, $LM_{\text{HFXO}} = 20$ mH, $RS_{\text{HFXO}} = 40$ Ω		110		μA
$I_{\text{START_X32M}}$	Average startup current for various crystals, first 1 ms				
$I_{\text{START_X32M_X0}}$	Typical parameters for a given 2.5mm x 2.0mm crystal: $CL_{\text{HFXO}} = 8$ pF, $CO_{\text{HFXO}} = 1$ pF, $LM_{\text{HFXO}} = 7$ mH, $RS_{\text{HFXO}} = 20$ Ω		360		μA
$I_{\text{START_X32M_X1}}$	Typical parameters for a given 1.6mm x 1.2mm crystal: $CL_{\text{HFXO}} = 8$ pF, $CO_{\text{HFXO}} = 0.4$ pF, $LM_{\text{HFXO}} = 20$ mH, $RS_{\text{HFXO}} = 40$ Ω		785		μA
$t_{\text{POWERUP_X32M}}$	Power-up time for various crystals				
$t_{\text{POWERUP_X32M_X0}}$	Typical parameters for a given 2.5mm x 2.0mm crystal: $CL_{\text{HFXO}} = 8$ pF, $CO_{\text{HFXO}} = 1$ pF, $LM_{\text{HFXO}} = 7$ mH, $RS_{\text{HFXO}} = 20$ Ω		60		μs
$t_{\text{POWERUP_X32M_X1}}$	Typical parameters for a given 1.6mm x 1.2mm crystal: $CL_{\text{HFXO}} = 8$ pF, $CO_{\text{HFXO}} = 0.4$ pF, $LM_{\text{HFXO}} = 20$ mH, $RS_{\text{HFXO}} = 40$ Ω		200		μs

5.4.4.3 Low frequency crystal oscillator (LFXO)

Symbol	Description	Min.	Typ.	Max.	Units
$f_{\text{NOM_LFXO}}$	Crystal frequency		32.768		kHz
$f_{\text{TOL_LFXO_BLE}}$	Frequency tolerance requirement for BLE stack			±500	ppm
$f_{\text{TOL_LFXO_ANT}}$	Frequency tolerance requirement for ANT stack			±50	ppm
$C_{\text{L_LFXO}}$	Load capacitance			12.5	pF
$C_{\text{O_LFXO}}$	Shunt capacitance			2	pF
$R_{\text{S_LFXO}}$	Equivalent series resistance			100	kΩ
$P_{\text{D_LFXO}}$	Drive level			0.5	μW
C_{pin}	Input capacitance on XL1 and XL2 pads		4		pF
I_{LFXO}	Run current for 32.768 kHz crystal oscillator		0.23		μA
$t_{\text{START_LFXO}}$	Startup time for 32.768 kHz crystal oscillator		0.25		s

Symbol	Description	Min.	Typ.	Max.	Units
t _{START_LFXO_EXT}	Startup time for 32.768 kHz crystal oscillator when CLOCK.LFXODEBOUNCE configured for Extended debounce time		0.5		s

5.4.4.4 Low frequency RC oscillator (LFRC)

Symbol	Description	Min.	Typ.	Max.	Units
f _{NOM_LFRC}	Nominal frequency		32.768		kHz
f _{TOL_LFRC}	Frequency tolerance, uncalibrated			±5	%
f _{TOL_CAL_LFRC}	Frequency tolerance after calibration ¹²			±500	ppm
I _{LFRC}	Run current		0.7		μA
t _{START_LFRC}	Startup time		1000		μs

5.4.4.5 Synthesized low frequency clock (LFSYNT)

Symbol	Description	Min.	Typ.	Max.	Units
f _{NOM_LFSYNT}	Nominal frequency		32.768		kHz

¹² Constant temperature within ±0.5 °C, calibration performed at least every 8 seconds, averaging interval > 7.5 ms, defined as 3 sigma

6 Peripherals

6.1 Peripheral interface

Peripherals are controlled by the CPU by writing to configuration registers and task registers. Peripheral events are indicated to the CPU by event registers and interrupts if they are configured for a given event.

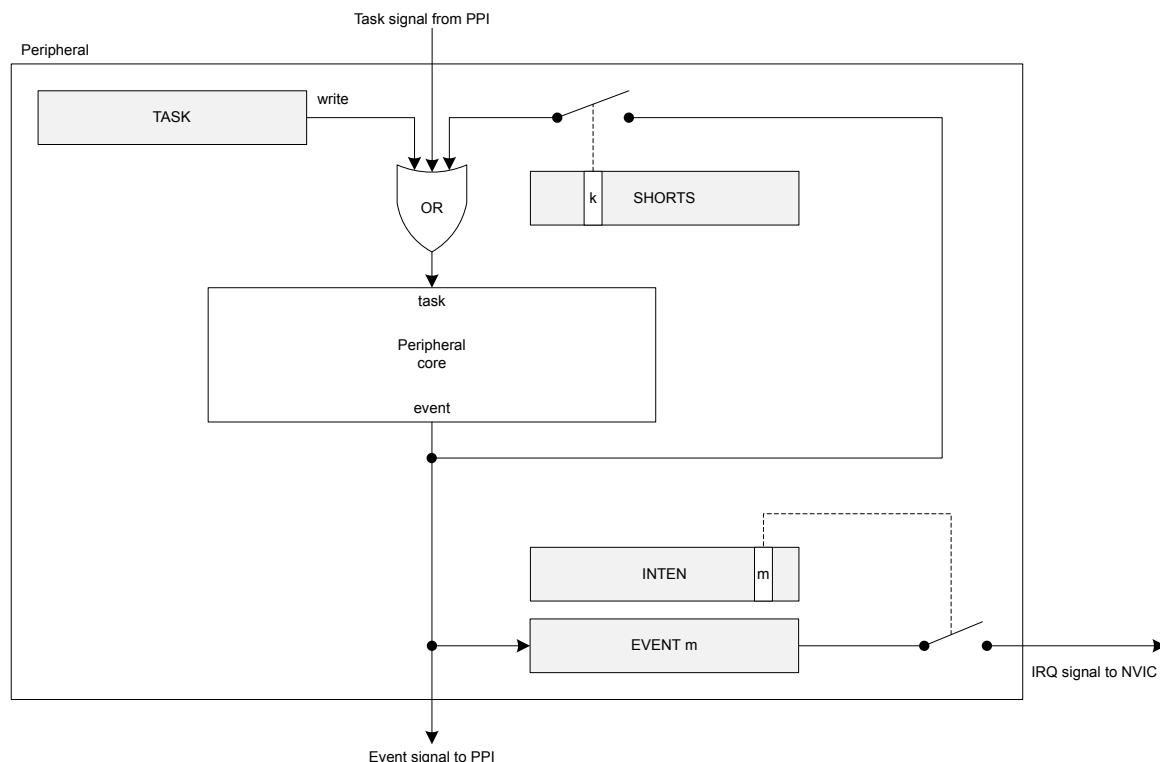


Figure 24: Tasks, events, shortcuts, and interrupts

6.1.1 Peripheral ID

Every peripheral is assigned a fixed block of 0x1000 bytes of address space, which is equal to 1024 x 32 bit registers.

See [Instantiation](#) on page 22 for more information about which peripherals are available and where they are located in the address map.

There is a direct relationship between peripheral ID and base address. For example, a peripheral with base address 0x40000000 is assigned ID=0, a peripheral with base address 0x40001000 is assigned ID=1, and a peripheral with base address 0x4001F000 is assigned ID=31.

Peripherals may share the same ID, which may impose one or more of the following limitations:

- Some peripherals share some registers or other common resources.
- Operation is mutually exclusive. Only one of the peripherals can be used at a time.
- Switching from one peripheral to another must follow a specific pattern (disable the first, then enable the second peripheral).

6.1.2 Peripherals with shared ID

In general (with the exception of ID 0), peripherals sharing an ID and base address may not be used simultaneously. The user can only enable one peripheral at the time on this specific ID.

When switching between two peripherals sharing an ID, the user should do the following to prevent unwanted behavior:

1. Disable the previously used peripheral.
2. Remove any programmable peripheral interconnect (PPI) connections set up for the peripheral that is being disabled.
3. Clear all bits in the INTEN register, i.e. `INTENCLR = 0xFFFFFFFF`.
4. Explicitly configure the peripheral that you are about to enable and do not rely on configuration values that may be inherited from the peripheral that was disabled.
5. Enable the now configured peripheral.

See which peripherals are sharing ID in [Instantiation](#) on page 22.

6.1.3 Peripheral registers

Most peripherals feature an ENABLE register. Unless otherwise specified in the relevant chapter, the peripheral registers (in particular the PSEL registers) must be configured before enabling the peripheral.

The peripheral must be enabled before tasks and events can be used.

6.1.4 Bit set and clear

Registers with multiple single-bit bit fields may implement the set-and-clear pattern. This pattern enables firmware to set and clear individual bits in a register without having to perform a read-modify-write operation on the main register.

This pattern is implemented using three consecutive addresses in the register map, where the main register is followed by dedicated SET and CLR registers (in that exact order).

The SET register is used to set individual bits in the main register while the CLR register is used to clear individual bits in the main register. Writing 1 to a bit in SET or CLR register will set or clear the same bit in the main register respectively. Writing 0 to a bit in SET or CLR register has no effect. Reading the SET or CLR register returns the value of the main register.

Note: The main register may not be visible and hence not directly accessible in all cases.

6.1.5 Tasks

Tasks are used to trigger actions in a peripheral, for example to start a particular behavior. A peripheral can implement multiple tasks with each task having a separate register in that peripheral's task register group.

A task is triggered when firmware writes 1 to the task register, or when the peripheral itself or another peripheral toggles the corresponding task signal. See [Tasks, events, shortcuts, and interrupts](#) on page 96.

6.1.6 Events

Events are used to notify peripherals and the CPU about events that have happened, for example a state change in a peripheral. A peripheral may generate multiple events with each event having a separate register in that peripheral's event register group.

An event is generated when the peripheral itself toggles the corresponding event signal, and the event register is updated to reflect that the event has been generated. See [Tasks, events, shortcuts, and interrupts](#) on page 96. An event register is only cleared when firmware writes 0 to it.

Events can be generated by the peripheral even when the event register is set to 1.

6.1.7 Shortcuts

A shortcut is a direct connection between an event and a task within the same peripheral. If a shortcut is enabled, the associated task is automatically triggered when its associated event is generated.

Using a shortcut is the equivalent to making the same connection outside the peripheral and through the PPI. However, the propagation delay through the shortcut is usually shorter than the propagation delay through the PPI.

Shortcuts are predefined, which means their connections cannot be configured by firmware. Each shortcut can be individually enabled or disabled through the shortcut register, one bit per shortcut, giving a maximum of 32 shortcuts for each peripheral.

6.1.8 Interrupts

All peripherals support interrupts. Interrupts are generated by events.

A peripheral only occupies one interrupt, and the interrupt number follows the peripheral ID. For example, the peripheral with ID=4 is connected to interrupt number 4 in the nested vectored interrupt controller (NVIC).

Using the INTEN, INTENSET, and INTENCLR registers, every event generated by a peripheral can be configured to generate that peripheral's interrupt. Multiple events can be enabled to generate interrupts simultaneously. To resolve the correct interrupt source, the event registers in the event group of peripheral registers will indicate the source.

Some peripherals implement only INTENSET and INTENCLR registers, and the INTEN register is not available on those peripherals. See the individual peripheral chapters for details. In all cases, reading back the INTENSET or INTENCLR register returns the same information as in INTEN.

Each event implemented in the peripheral is associated with a specific bit position in the INTEN, INTENSET, and INTENCLR registers.

The relationship between tasks, events, shortcuts, and interrupts is shown in [Tasks, events, shortcuts, and interrupts](#) on page 96.

Interrupt clearing

Clearing an interrupt by writing 0 to an event register, or disabling an interrupt using the INTENCLR register, can take up to four CPU clock cycles to take effect. This means that an interrupt may reoccur immediately, even if a new event has not come, if the program exits an interrupt handler after the interrupt is cleared or disabled but before four clock cycles have passed.

Note: To avoid an interrupt reoccurring before a new event has come, the program should perform a read from one of the peripheral registers. For example, the event register that has been cleared, or the INTENCLR register that has been used to disable the interrupt. This will cause a one to three-cycle delay and ensure the interrupt is cleared before exiting the interrupt handler.

Care should be taken to ensure the compiler does not remove the read operation as an optimization. If the program can guarantee a four-cycle delay after an event is cleared or an interrupt is disabled, then a read of a register is not required.

6.2 AAR — Accelerated address resolver

Accelerated address resolver is a cryptographic support function for implementing the Resolvable Private Address Resolution Procedure described in the *Bluetooth Core specification v4.0*. Resolvable Private Address generation should be achieved using ECB and is not supported by AAR.

The procedure allows two devices that share a secret key to generate and resolve a hash based on their device address. The AAR block enables real-time address resolution on incoming packets when configured as described in this chapter. This allows real-time packet filtering (whitelisting) using a list of known shared keys (Identity Resolving Keys (IRK) in *Bluetooth*).

6.2.1 EasyDMA

The AAR implements EasyDMA for reading and writing to the RAM. The EasyDMA will have finished accessing the RAM when the END, RESOLVED, and NOTRESOLVED events are generated.

If the [IRKPTR](#) on page 104, [ADDRPTR](#) on page 104, and the [SCRATCHPTR](#) on page 104 is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 19 for more information about the different memory regions.

6.2.2 Resolving a resolvable address

As per *Bluetooth* specification, a private resolvable address is composed of six bytes.

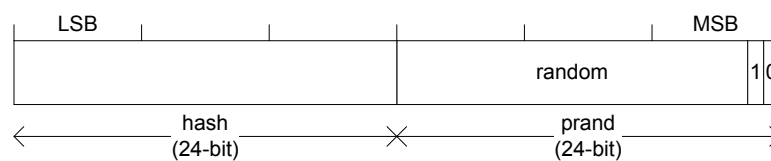


Figure 25: Resolvable address

To resolve an address the register [ADDRPTR](#) on page 104 must point to the start of the packet. The resolver is started by triggering the START task. A RESOLVED event is generated when the AAR manages to resolve the address using one of the Identity Resolving Keys (IRK) found in the IRK data structure. The AAR will use the IRK specified in the register IRK0 to IRK15 starting from IRK0. The register [NIRK](#) on page 103 specifies how many IRKs should be used. The AAR module will generate a NOTRESOLVED event if it is not able to resolve the address using the specified list of IRKs.

The AAR will go through the list of available IRKs in the IRK data structure and for each IRK try to resolve the address according to the Resolvable Private Address Resolution Procedure described in the *Bluetooth Core specification v4.0* [Vol 3] chapter 10.8.2.3. The time it takes to resolve an address varies due to the location in the list of the resolvable address. The resolution time will also be affected by RAM accesses performed by other peripherals and the CPU. See the [Electrical specifications](#) for more information about resolution time.

The AAR only compares the received address to those programmed in the module without checking the address type.

The AAR will stop as soon as it has managed to resolve the address, or after trying to resolve the address using NIRK number of IRKs from the IRK data structure. The AAR will generate an END event after it has stopped.

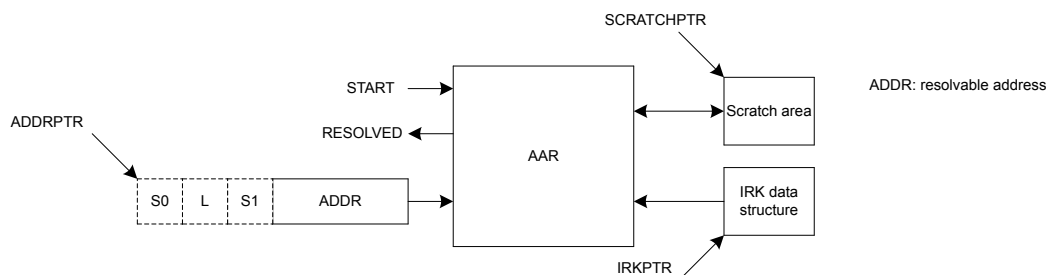


Figure 26: Address resolution with packet preloaded into RAM

6.2.3 Use case example for chaining RADIO packet reception with address resolution using AAR

The AAR may be started as soon as the 6 bytes required by the AAR have been received by the RADIO and stored in RAM. The ADDRPTR pointer must point to the start of packet.

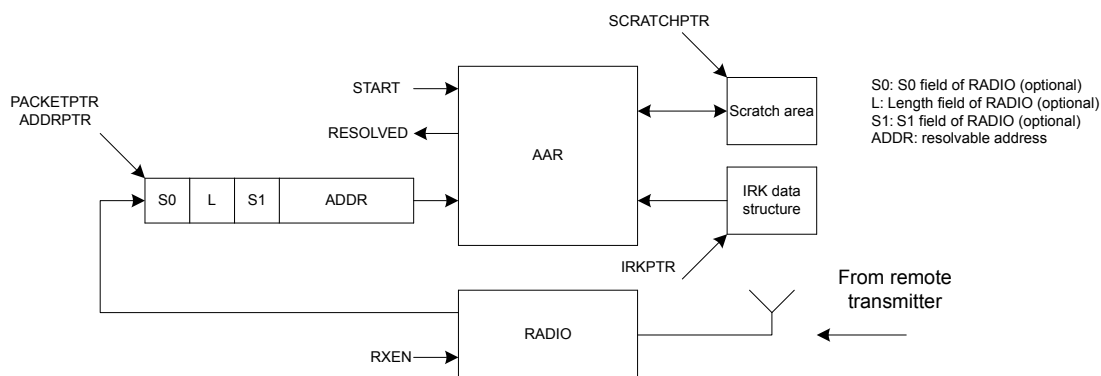


Figure 27: Address resolution with packet loaded into RAM by the RADIO

6.2.4 IRK data structure

The IRK data structure is located in RAM at the memory location specified by the IRKPTR register.

Property	Address offset	Description
IRK0	0	IRK number 0 (16 bytes)
IRK1	16	IRK number 1 (16 bytes)
..	..	..
IRK15	240	IRK number 15 (16 bytes)

Table 21: IRK data structure overview

6.2.5 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4000F000	AAR	AAR	Accelerated address resolver	

Table 22: Instances

Register	Offset	Description
TASKS_START	0x000	Start resolving addresses based on IRKs specified in the IRK data structure
TASKS_STOP	0x008	Stop resolving addresses
EVENTS_END	0x100	Address resolution procedure complete

6.2.5.1 TASKS_START

Start resolving addresses based on IRKs specified in the IRK data structure

6.2.5.2 TASKS_STOP

Stop resolving addresses

6.2.5.3 EVENTS_END

Address resolution procedure complete

4452 021 v1.3

6.2.5.4 EVENTS_RESOLVED

Address offset: 0x104

Address resolved

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A																															
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value		Description																													
A	RW	EVENTS_RESOLVED			Address resolved																													
		NotGenerated	0	Event not generated																														
		Generated	1	Event generated																														

6.2.5.5 EVENTS_NOTRESOLVED

Address offset: 0x108

Address not resolved

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																	A	
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ID	Acce Field	Value ID	Value		Description																													
A	RW	EVENTS_NOTRESOLVED			Address not resolved																													
		NotGenerated	0		Event not generated																													
		Generated	1		Event generated																													

6.2.5.6 INTENSET

Address offset: 0x304

Enable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW END			Write '1' to enable interrupt for event END																														
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
B	RW RESOLVED			Write '1' to enable interrupt for event RESOLVED																														
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
C	RW NOTRESOLVED			Write '1' to enable interrupt for event NOTRESOLVED																														
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														

6.2.5.7 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID																																				C	B	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
ID	Acce	Field	Value	ID	Value	Description																																
A	RW	END				Write '1' to disable interrupt for event END																																
			Clear	1	Disable																																	
			Disabled	0	Read: Disabled																																	
			Enabled	1	Read: Enabled																																	
B	RW	RESOLVED				Write '1' to disable interrupt for event RESOLVED																																
			Clear	1	Disable																																	
			Disabled	0	Read: Disabled																																	
			Enabled	1	Read: Enabled																																	
C	RW	NOTRESOLVED				Write '1' to disable interrupt for event NOTRESOLVED																																
			Clear	1	Disable																																	
			Disabled	0	Read: Disabled																																	
			Enabled	1	Read: Enabled																																	

6.2.5.8 STATUS

Address offset: 0x400

Resolution status

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID																														A				A	A	A								
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value		Description																																					
A	R	STATUS		[0..15]		The IRK that was used last time an address was resolved																																						

6.2.5.9 ENABLE

Address offset: 0x500

Enable AAR

Bit number					31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID																																					A	A
Reset 0x00000000					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ID	Acce Field		Value	ID	Value		Description																															
A	RW	ENABLE			Enable or disable AAR																																	
			Disabled	0	Disable																																	
			Enabled	3	Enable																																	

6.2.5.10 NIRK

Address offset: 0x504

Number of IRKs

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID																																	A	A	A	A	A
Reset 0x00000001			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1			
ID	Acce Field	Value ID	Value		Description																																
A	RW NIRK		[1..16]		Number of Identity root keys available in the IRK data structure																																

6.2.5.11 IRKPTR

Address offset: 0x508

Pointer to IRK data structure

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID										A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID			Value			Description																																		
A	RW IRKPTR								Pointer to the IRK data structure																																		

6.2.5.12 ADDRPTR

Address offset: 0x510

Pointer to the resolvable address

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value																Description															
A	RW	ADDRPTR	Pointer to the resolvable address (6-bytes)																															

6.2.5.13 SCRATCHPTR

Address offset: 0x514

Pointer to data area used for temporary storage

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A				
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	Acce Field	Value ID	Value		Description																																	
A	RW	SCRATCHPTR			Pointer to a scratch data area used for temporary storage during resolution. A space of minimum 3 bytes must be reserved.																																	

6.2.6 Electrical specification

6.2.6.1 AAR Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{AAR}	Address resolution time per IRK. Total time for several IRKs is given as $(1 \mu s + n * t_{AAR})$, where n is the number of IRKs. (Given priority to the actual destination RAM block).			6	μs
$t_{AAR,8}$	Time for address resolution of 8 IRKs. (Given priority to the actual destination RAM block).			49	μs

6.3 ACL — Access control lists

The Access control lists (ACL) peripheral is designed to assign and enforce access permissions to different regions of the on-chip flash memory map.

Flash memory regions can be assigned individual ACL permission schemes. The following registers are involved:

- PERM register, where the permissions are configured.
- ADDR register, where the word-aligned start address for the flash page is defined.
- SIZE register, where the size of the region the permissions are applied to is determined.

Note: The size of the region in bytes is restricted to a multiple of the flash page size, and the maximum region size is limited to half the flash size. See [Memory](#) on page 19 for more information.

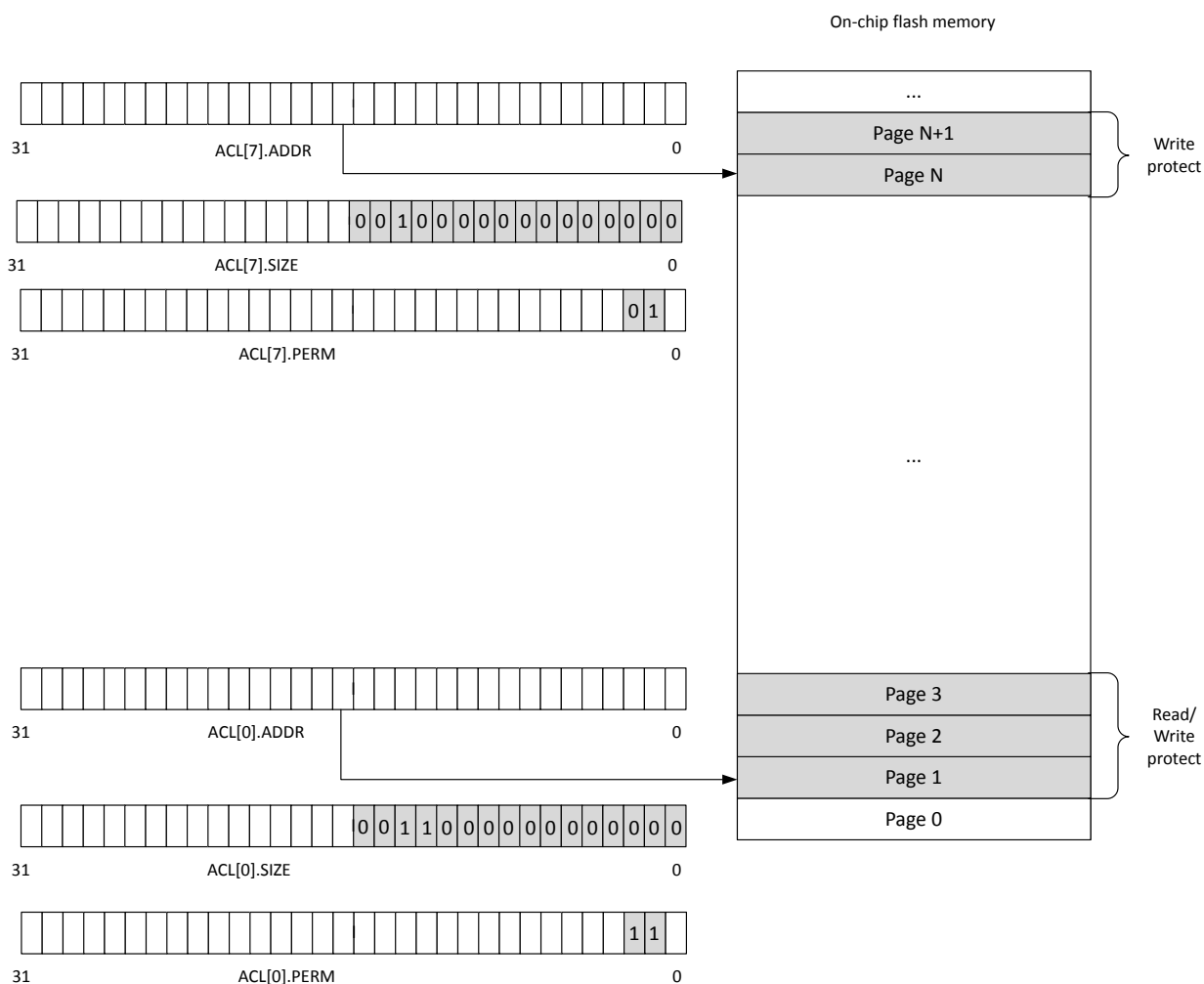


Figure 28: Protected regions of on-chip flash memory

There are four defined ACL permission schemes, with different combinations of read/write permissions:

Read	Write	Protection description
0	0	No protection. Entire region can be executed, read, written or erased.
0	1	Region can be executed and read, but not written or erased.
1	0	Region can be written and erased, but not executed or read.
1	1	Region is locked for all access until next reset.

Table 24: Permission schemes

Note: If a permission violation to a protected region is detected by the ACL peripheral, the request is blocked and a Bus Fault exception is triggered.

Access control to a configured region is enforced by the hardware two CPU clock cycles after the ADDR, SIZE, and PERM registers for an ACL instance have been successfully written. The protection is only enforced if a valid start address of the flash page boundary is written into the ADDR register, and the values of the SIZE and PERM registers are not zero.

The ADDR, SIZE, and PERM registers can only be written once. All ACL configuration registers are cleared on reset (by resetting the device from any reset source), which is also the only way of clearing the configuration registers. To ensure that the desired permission schemes are always enforced by the ACL peripheral, the device boot sequence must perform the necessary configuration.

Debugger read access to a read-protected region will be Read-As-Zero (RAZ), while debugger write access to a write-protected region will be Write-Ignored (WI).

6.3.1 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4001E000	ACL	ACL	Access control lists	

Table 25: Instances

Register	Offset	Description
ACL[0].ADDR	0x800	Start address of region to protect. The start address must be word-aligned.
ACL[0].SIZE	0x804	Size of region to protect counting from address ACL[0].ADDR. Write '0' as no effect.
ACL[0].PERM	0x808	Access permissions for region 0 as defined by start address ACL[0].ADDR and size ACL[0].SIZE
ACL[0].UNUSED0	0x80C	Reserved
ACL[1].ADDR	0x810	Start address of region to protect. The start address must be word-aligned.
ACL[1].SIZE	0x814	Size of region to protect counting from address ACL[1].ADDR. Write '0' as no effect.
ACL[1].PERM	0x818	Access permissions for region 1 as defined by start address ACL[1].ADDR and size ACL[1].SIZE
ACL[1].UNUSED0	0x81C	Reserved
ACL[2].ADDR	0x820	Start address of region to protect. The start address must be word-aligned.
ACL[2].SIZE	0x824	Size of region to protect counting from address ACL[2].ADDR. Write '0' as no effect.
ACL[2].PERM	0x828	Access permissions for region 2 as defined by start address ACL[2].ADDR and size ACL[2].SIZE
ACL[2].UNUSED0	0x82C	Reserved
ACL[3].ADDR	0x830	Start address of region to protect. The start address must be word-aligned.
ACL[3].SIZE	0x834	Size of region to protect counting from address ACL[3].ADDR. Write '0' as no effect.
ACL[3].PERM	0x838	Access permissions for region 3 as defined by start address ACL[3].ADDR and size ACL[3].SIZE
ACL[3].UNUSED0	0x83C	Reserved
ACL[4].ADDR	0x840	Start address of region to protect. The start address must be word-aligned.
ACL[4].SIZE	0x844	Size of region to protect counting from address ACL[4].ADDR. Write '0' as no effect.
ACL[4].PERM	0x848	Access permissions for region 4 as defined by start address ACL[4].ADDR and size ACL[4].SIZE
ACL[4].UNUSED0	0x84C	Reserved
ACL[5].ADDR	0x850	Start address of region to protect. The start address must be word-aligned.
ACL[5].SIZE	0x854	Size of region to protect counting from address ACL[5].ADDR. Write '0' as no effect.
ACL[5].PERM	0x858	Access permissions for region 5 as defined by start address ACL[5].ADDR and size ACL[5].SIZE
ACL[5].UNUSED0	0x85C	Reserved
ACL[6].ADDR	0x860	Start address of region to protect. The start address must be word-aligned.
ACL[6].SIZE	0x864	Size of region to protect counting from address ACL[6].ADDR. Write '0' as no effect.
ACL[6].PERM	0x868	Access permissions for region 6 as defined by start address ACL[6].ADDR and size ACL[6].SIZE
ACL[6].UNUSED0	0x86C	Reserved
ACL[7].ADDR	0x870	Start address of region to protect. The start address must be word-aligned.
ACL[7].SIZE	0x874	Size of region to protect counting from address ACL[7].ADDR. Write '0' as no effect.
ACL[7].PERM	0x878	Access permissions for region 7 as defined by start address ACL[7].ADDR and size ACL[7].SIZE

Register	Offset	Description	
ACL[7].UNUSED0	0x87C		Reserved

Table 26: Register overview

6.3.1.1 ACL[n].ADDR (n=0..7)

Address offset: $0x800 + (n \times 0x10)$

Start address of region to protect. The start address must be word-aligned.

This register can only be written once.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																												
A	RW1 ADDR			Start address of flash region n. The start address must point to a flash page boundary.																												

6.3.1.2 ACL[n].SIZE (n=0..7)

Address offset: $0x804 + (n \times 0x10)$

Size of region to protect counting from address ACL[n].ADDR. Write '0' as no effect.

This register can only be written once.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																												
A	RW1 SIZE			Size of flash region n in bytes. Must be a multiple of the flash page size.																												

6.3.1.3 ACL[n].PERM (n=0..7)

Address offset: $0x808 + (n \times 0x10)$

Access permissions for region n as defined by start address ACL[n].ADDR and size ACL[n].SIZE

This register can only be written once.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																															C	B
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																												
B	RW1 WRITE			Configure write and erase permissions for region n. Write '0' has no effect.																												
		Enable	0	Allow write and erase instructions to region n																												
		Disable	1	Block write and erase instructions to region n																												
C	RW1 READ			Configure read permissions for region n. Write '0' has no effect.																												
		Enable	0	Allow read instructions to region n																												
		Disable	1	Block read instructions to region n																												

6.4 CCM — AES CCM mode encryption

Cipher block chaining - message authentication code (CCM) mode is an authenticated encryption algorithm designed to provide both authentication and confidentiality during data transfer. CCM combines counter mode encryption and CBC-MAC authentication. The CCM terminology "Message authentication code (MAC)" is called the "Message integrity check (MIC)" in *Bluetooth* terminology and also in this document.

The CCM block generates an encrypted keystream that is applied to input data using the XOR operation and generates the 4 byte MIC field in one operation. The CCM and radio can be configured to work synchronously. The CCM will encrypt in time for transmission and decrypt after receiving bytes into memory from the radio. All operations can complete within the packet RX or TX time. CCM on this device is implemented according to *Bluetooth* requirements and the algorithm as defined in IETF [RFC3610](#), and depends on the AES-128 block cipher. A description of the CCM algorithm can also be found in [NIST Special Publication 800-38C](#). The *Bluetooth* specification describes the configuration of counter mode blocks and encryption blocks to implement compliant encryption for BLE.

The CCM block uses EasyDMA to load key, counter mode blocks (including the nonce required), and to read/write plain text and cipher text.

The AES CCM supports three operations: key-stream generation, packet encryption, and packet decryption. All these operations are done in compliance with the *Bluetooth* specification.¹³

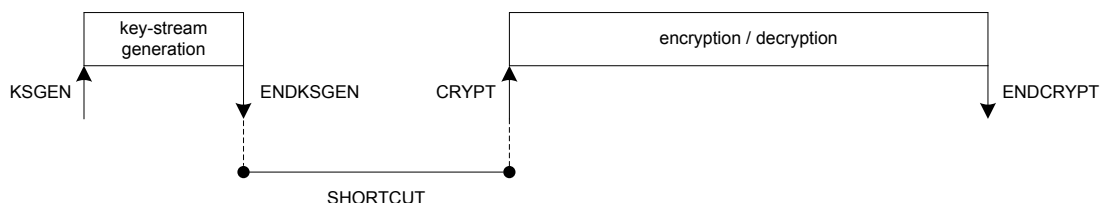


Figure 29: Key-stream generation followed by encryption or decryption. The shortcut is optional.

6.4.1 Key-steam generation

A new key-stream needs to be generated before a new packet encryption or packet decryption operation can be started.

A key-stream is generated by triggering the KSGEN task and an ENDKSGEN event will be generated when the key-stream has been generated.

Key-stream generation, packet encryption, and packet decryption operations utilize the configuration specified in the data structure pointed to by [CNFPTR](#) on page 118. It is necessary to configure this pointer and its underlying data structure, and the [MODE](#) on page 117 register before the KSGEN task is triggered.

The key-stream will be stored in the AES CCM's temporary memory area, specified by the [SCRATCHPTR](#) on page 118, where it will be used in subsequent encryption and decryption operations.

For default length packets (MODE.LENGTH = Default) the size of the generated key-stream is 27 bytes. When using extended length packets (MODE.LENGTH = Extended) the [MAXPACKETSIZE](#) on page 119 register specifies the length of the key-stream to be generated. The length of the generated key-stream must be greater or equal to the length of the subsequent packet payload to be encrypted or decrypted. The maximum length of the key-stream in extended mode is 251 bytes, which means that the maximum packet payload size is 251.

¹³ *Bluetooth* AES CCM 128 bit block encryption, see *Bluetooth* Core specification Version 4.0.

If a shortcut is used between ENDKSGEN event and CRYPT task, the [INPTR](#) on page 118 pointer and the [OUTPTR](#) on page 118 pointers must also be configured before the KSGEN task is triggered.

6.4.2 Encryption

During packet encryption, the AES CCM will read the unencrypted packet located in RAM at the address specified in the INPTR pointer, encrypt the packet and append a four byte long Message Integrity Check (MIC) field to the packet.

Encryption is started by triggering the CRYPT task with the [MODE](#) on page 117 register set to ENCRYPTION. An ENDCRYPT event will be generated when packet encryption is completed

The AES CCM will also modify the length field of the packet to adjust for the appended MIC field, that is, add four bytes to the length, and store the resulting packet back into RAM at the address specified in the [OUTPTR](#) on page 118 pointer, see [Encryption](#) on page 109.

Empty packets (length field is set to 0) will not be encrypted but instead moved unmodified through the AES CCM.

The CCM supports different widths of the LENGTH field in the data structure for encrypted packets. This is configured in the [MODE](#) on page 117 register.

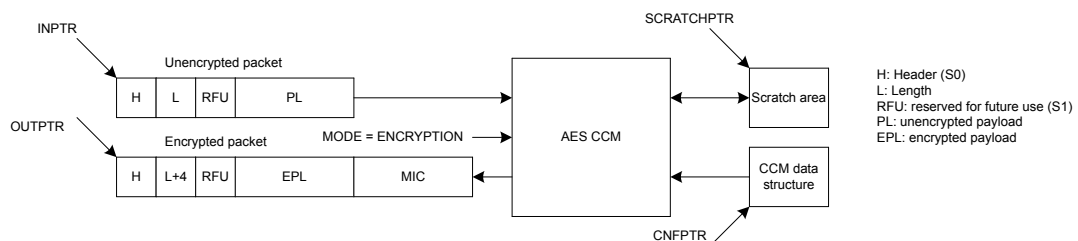


Figure 30: Encryption

6.4.3 Decryption

During packet decryption, the AES CCM will read the encrypted packet located in RAM at the address specified in the INPTR pointer, decrypt the packet, authenticate the packet's MIC field and generate the appropriate MIC status.

Decryption is started by triggering the CRYPT task with the [MODE](#) on page 117 register set to DECRYPTION. An ENDCRYPT event will be generated when packet decryption is completed

The AES CCM will also modify the length field of the packet to adjust for the MIC field, that is, subtract four bytes from the length, and then store the decrypted packet into RAM at the address pointed to by the OUTPTR pointer, see [Decryption](#) on page 110.

The CCM is only able to decrypt packet payloads that are at least 5 bytes long, that is, 1 byte or more encrypted payload (EPL) and 4 bytes of MIC. The CCM will therefore generate a MIC error for packets where the length field is set to 1, 2, 3 or 4.

Empty packets (length field is set to 0) will not be decrypted but instead moved unmodified through the AES CCM, these packets will always pass the MIC check.

The CCM supports different widths of the LENGTH field in the data structure for decrypted packets. This is configured in the [MODE](#) on page 117 register.

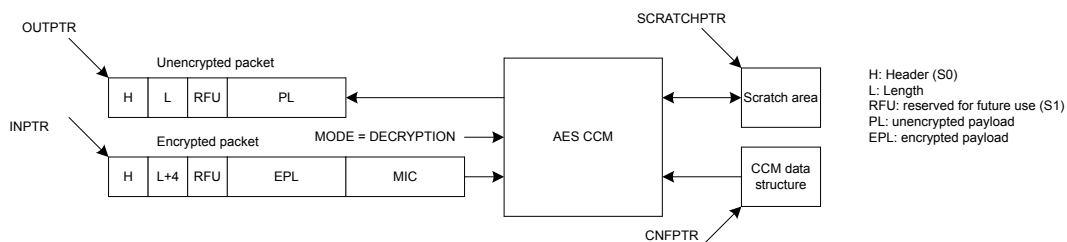


Figure 31: Decryption

6.4.4 AES CCM and RADIO concurrent operation

The CCM module is able to encrypt/decrypt data synchronously to data being transmitted or received on the radio.

In order for the CCM module to run synchronously with the radio, the data rate setting in the **MODE** on page 117 register needs to match the radio data rate. The settings in this register apply whenever either the KSGEN or CRYPT tasks are triggered.

The data rate setting of the **MODE** on page 117 register can also be overridden on-the-fly during an ongoing encrypt/decrypt operation by the contents of the **RATEOVERRIDE** on page 119 register. The data rate setting in this register applies whenever the RATEOVERRIDE task is triggered. This feature can be useful in cases where the radio data rate is changed during an ongoing packet transaction.

6.4.5 Encrypting packets on-the-fly in radio transmit mode

When the AES CCM is encrypting a packet on-the-fly at the same time as the radio is transmitting it, the radio must read the encrypted packet from the same memory location as the AES CCM is writing to.

The **OUTPTR** on page 118 pointer in the AES CCM must therefore point to the same memory location as the **PACKETPTR** pointer in the radio, see [Configuration of on-the-fly encryption](#) on page 110.

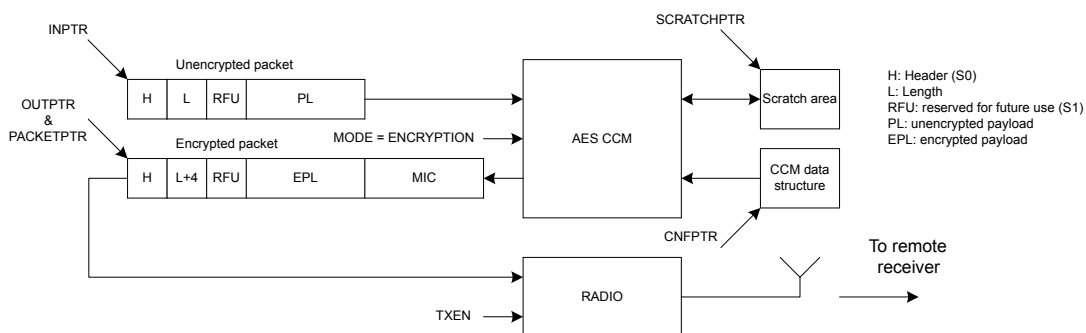


Figure 32: Configuration of on-the-fly encryption

In order to match the RADIO's timing, the KSGEN task must be triggered early enough to allow the key-stream generation to complete before the encryption of the packet shall start.

For short packets (**MODE.LENGTH** = Default) the KSGEN task must be triggered no later than when the **START** task in the RADIO is triggered. In addition the shortcut between the **ENDKSGEN** event and the **CRYPT** task must be enabled. This use-case is illustrated in [On-the-fly encryption of short packets \(MODE.LENGTH = Default\) using a PPI connection](#) on page 111 using a PPI connection between the **READY** event in the RADIO and the **KSGEN** task in the AES CCM.

For long packets (**MODE.LENGTH** = Extended) the key-stream generation will need to be started even earlier, for example at the time when the **TXEN** task in the RADIO is triggered.

Important: Refer to [Timing specification](#) on page 120 for information about the time needed for generating a key-stream.

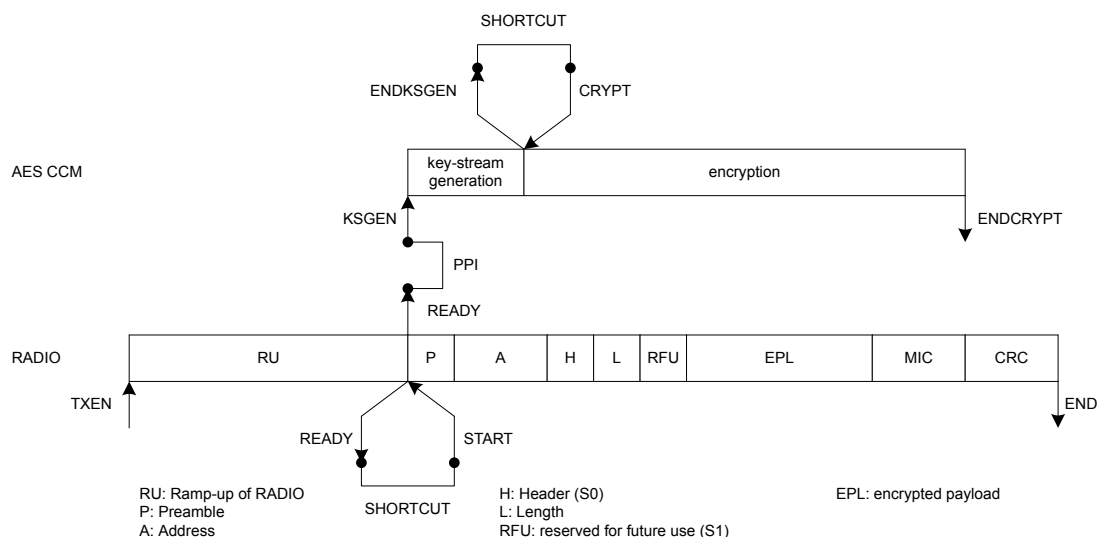


Figure 33: On-the-fly encryption of short packets (MODE.LENGTH = Default) using a PPI connection

6.4.6 Decrypting packets on-the-fly in radio receive mode

When the AES CCM is decrypting a packet on-the-fly at the same time as the RADIO is receiving it, the AES CCM must read the encrypted packet from the same memory location as the RADIO is writing to.

The **INPTR** on page 118 pointer in the AES CCM must therefore point to the same memory location as the **PACKETPTR** pointer in the RADIO, see [Configuration of on-the-fly decryption](#) on page 111.

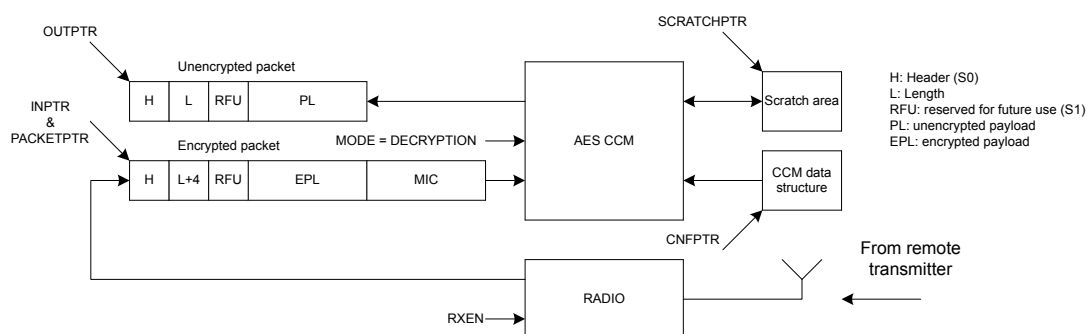


Figure 34: Configuration of on-the-fly decryption

In order to match the RADIO's timing, the KSGEN task must be triggered early enough to allow the key-stream generation to complete before the decryption of the packet shall start.

For short packets (MODE.LENGTH = Default) the KSGEN task must be triggered no later than when the START task in the RADIO is triggered. In addition, the CRYPT task must be triggered no earlier than when the ADDRESS event is generated by the RADIO.

If the CRYPT task is triggered exactly at the same time as the ADDRESS event is generated by the RADIO, the AES CCM will guarantee that the decryption is completed no later than when the END event in the RADIO is generated.

This use-case is illustrated in [On-the-fly decryption of short packets \(MODE.LENGTH = Default\) using a PPI connection](#) on page 112 using a PPI connection between the ADDRESS event in the RADIO and the CRYPT task in the AES CCM. The KSGEN task is triggered from the READY event in the RADIO through a PPI connection.

For long packets (MODE.LENGTH = Extended) the key-stream generation will need to be started even earlier, for example at the time when the RXEN task in the RADIO is triggered.

Important: Refer to [Timing specification](#) on page 120 for information about the time needed for generating a key-stream.

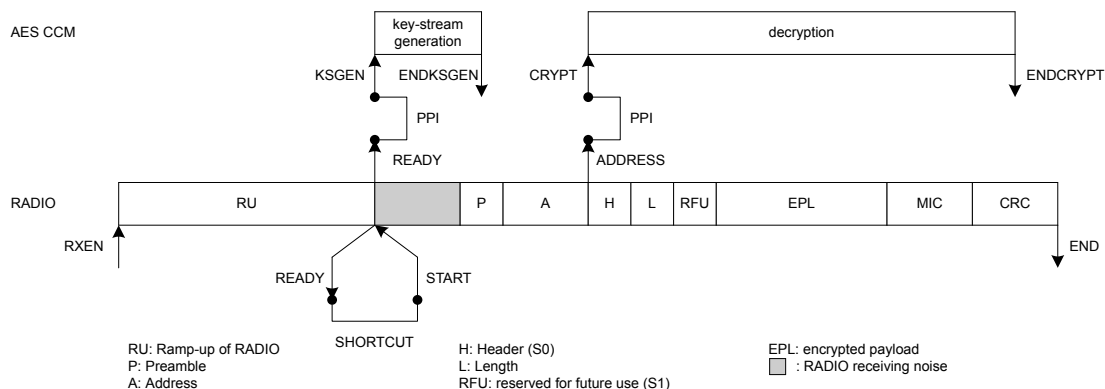


Figure 35: On-the-fly decryption of short packets (*MODE.LENGTH = Default*) using a PPI connection

6.4.7 CCM data structure

The CCM data structure is located in Data RAM at the memory location specified by the CNFPTR pointer register.

Property	Address offset	Description
KEY	0	16 byte AES key
PKTCTR	16	Octet0 (LSO) of packet counter
	17	Octet1 of packet counter
	18	Octet2 of packet counter
	19	Octet3 of packet counter
	20	Bit 6 – Bit 0: Octet4 (7 most significant bits of packet counter, with Bit 6 being the most significant bit) Bit7: Ignored
	21	Ignored
	22	Ignored
	23	Ignored
	24	Bit 0: Direction bit Bit 7 – Bit 1: Zero padded
IV	25	8 byte initialization vector (IV) Octet0 (LSO) of IV, Octet1 of IV, ... , Octet7 (MSO) of IV

Table 27: CCM data structure overview

The NONCE vector (as specified by the *Bluetooth* Core Specification) will be generated by hardware based on the information specified in the CCM data structure from [CCM data structure overview](#) on page 112 .

Property	Address offset	Description
HEADER	0	Packet Header
LENGTH	1	Number of bytes in unencrypted payload
RFU	2	Reserved Future Use
PAYLOAD	3	Unencrypted payload

Table 28: Data structure for unencrypted packet

Property	Address offset	Description
HEADER	0	Packet Header
LENGTH	1	Number of bytes in encrypted payload including length of MIC
		Important: LENGTH will be 0 for empty packets since the MIC is not added to empty packets
RFU	2	Reserved Future Use
PAYLOAD	3	Encrypted payload
MIC	3 + payload length	ENCRYPT: 4 bytes encrypted MIC
		Important: MIC is not added to empty packets

Table 29: Data structure for encrypted packet

6.4.8 EasyDMA and ERROR event

The CCM implements an EasyDMA mechanism for reading and writing to the RAM.

In cases where the CPU and other EasyDMA enabled peripherals are accessing the same RAM block at the same time, a high level of bus collisions may cause too slow operation for correct on the fly encryption. In this case the ERROR event will be generated.

The EasyDMA will have finished accessing the RAM when the ENDKSGEN and ENDCRYPT events are generated.

If the CNFPTR, SCRATCHPTR, INPTR and the OUTPTR are not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 19 for more information about the different memory regions.

6.4.9 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4000F000	CCM	CCM	AES counter with CBC-MAC (CCM) mode block encryption	

Table 30: Instances

Register	Offset	Description	
TASKS_KSGEN	0x000	Start generation of key-stream. This operation will stop by itself when completed.	
TASKS_CRYPT	0x004	Start encryption/decryption. This operation will stop by itself when completed.	
TASKS_STOP	0x008	Stop encryption/decryption	
TASKS_RATEOVERRIDE	0x00C	Override DATARATE setting in MODE register with the contents of the RATEOVERRIDE register for any ongoing encryption/decryption	
EVENTS_ENDKSGEN	0x100	Key-stream generation complete	
EVENTS_ENDCRYPT	0x104	Encrypt/decrypt complete	
EVENTS_ERROR	0x108	CCM error event	Deprecated
SHORTS	0x200	Shortcuts between local events and tasks	
INTENSET	0x304	Enable interrupt	
INTENCLR	0x308	Disable interrupt	
MICSTATUS	0x400	MIC check result	
ENABLE	0x500	Enable	
MODE	0x504	Operation mode	
CNFPTR	0x508	Pointer to data structure holding AES key and NONCE vector	
INPTR	0x50C	Input pointer	
OUTPTR	0x510	Output pointer	

Register	Offset	Description
SCRATCHPTR	0x514	Pointer to data area used for temporary storage
MAXPACKETSIZE	0x518	Length of key-stream generated when MODE.LENGTH = Extended.
RATEOVERRIDE	0x51C	Data rate override setting.

Table 31: Register overview

6.4.9.1 TASKS_KSGEN

Address offset: 0x000

Start generation of key-stream. This operation will stop by itself when completed.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W	TASKS_KSGEN		Start generation of key-stream. This operation will stop by itself when completed.																															
		Trigger	1	Trigger task																															

6.4.9.2 TASKS_CRYPT

Address offset: 0x004

Start encryption/decryption. This operation will stop by itself when completed.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A																															
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value		Description																													
A	W	TASKS_CRYPT			Start encryption/decryption. This operation will stop by itself when completed.																													
		Trigger	1		Trigger task																													

6.4.9.3 TASKS_STOP

Address offset: 0x008

Stop encryption/decryption

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value		Description																													
A	W	TASKS_STOP			Stop encryption/decryption																													
		Trigger	1		Trigger task																													

6.4.9.4 TASKS_RATEOVERRIDE

Address offset: 0x00C

Override DATARATE setting in MODE register with the contents of the RATEOVERRIDE register for any ongoing encryption/decryption

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																		A
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A	W	TASKS_RATEOVERRIDE				Override DATARATE setting in MODE register with the contents of the RATEOVERRIDE register for any ongoing encryption/decryption																												
		Trigger	1			Trigger task																												

6.4.9.5 EVENTS_ENDKSGEN

Address offset: 0x100

Key-stream generation complete

Bit number					31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID					A																																			
Reset 0x00000000					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	Acce Field		Value ID		Value		Description																																	
A	RW		EVENTS_ENDKSGEN				Key-stream generation complete																																	
			NotGenerated		0		Event not generated																																	
			Generated		1		Event generated																																	

6.4.9.6 EVENTS ENDCRYPT

Address offset: 0x104

Encrypt/decrypt complete

Bit number					31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID					A																															
Reset 0x00000000					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value				Description																											
A	RW		EVENTS_ENDCRYPT						Encrypt/decrypt complete																											
			NotGenerated		0				Event not generated																											
			Generated		1				Event generated																											

6.4.9.7 EVENTS ERROR (Deprecated)

Address offset: 0x108

CCM error event

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID																																				A			
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	Acces	Field	Value	ID	Value	Description																																	
A	RW	EVENTS_ERROR				CCM error event																																	Deprecated
			NotGenerated	0		Event not generated																																	
			Generated	1		Event generated																																	

6.4.9.8 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																		A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value		Description																													
A	RW	ENDKSGEN_CRYPT			Shortcut between event ENDKSGEN and task CRYPT																													
		Disabled	0		Disable shortcut																													
		Enabled	1		Enable shortcut																													

6.4.9.9 INTENSET

Address offset: 0x304

Enable interrupt

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
ID																																						C	B	A	
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
ID	Acce Field	Value ID	Value	Description																																					
A	RW	ENDKSGEN		Write '1' to enable interrupt for event ENDKSGEN																																					
		Set	1	Enable																																					
		Disabled	0	Read: Disabled																																					
		Enabled	1	Read: Enabled																																					
B	RW	ENDCRYPT		Write '1' to enable interrupt for event ENDCRYPT																																					
		Set	1	Enable																																					
		Disabled	0	Read: Disabled																																					
		Enabled	1	Read: Enabled																																					
C	RW	ERROR		Write '1' to enable interrupt for event ERROR																																			Deprecated		
		Set	1	Enable																																					
		Disabled	0	Read: Disabled																																					
		Enabled	1	Read: Enabled																																					

Deprecated

6.4.9.10 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	ENDKSGEN		Write '1' to disable interrupt for event ENDKSGEN																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
B	RW	ENDCRYPT		Write '1' to disable interrupt for event ENDCRYPT																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
C	RW	ERROR		Write '1' to disable interrupt for event ERROR																													Deprecated	
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														

Deprecated

6.4.9.11 MICSTATUS

Address offset: 0x400

MIC check result

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A																															
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value			Description																												
A	R	MICSTATUS				The result of the MIC check performed during the previous decryption operation																												
		CheckFailed	0			MIC check failed																												
		CheckPassed	1			MIC check passed																												

6.4.9.12 ENABLE

Address offset: 0x500

Enable

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID																																								A	A		
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field			Value ID			Value			Description																																	
A	RW	ENABLE									Enable or disable CCM																																
		Disabled			0						Disable																																
		Enabled			2						Enable																																

6.4.9.13 MODE

Address offset: 0x504

Operation mode

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
ID				C										B										B										A									
Reset 0x00000001				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1																																							
ID	Acce	Field	Value	ID	Value	Description																																					
A	RW	MODE				The mode of operation to be used. The settings in this register apply whenever either the KSGEN or CRYPT tasks are triggered.																																					
			Encryption	0		AES CCM packet encryption mode																																					
			Decryption	1		AES CCM packet decryption mode																																					
B	RW	DATARATE				Radio data rate that the CCM shall run synchronous with																																					
			1Mbit	0		1 Mbps																																					
			2Mbit	1		2 Mbps																																					
			125Kbps	2		125 Kbps																																					
			500Kbps	3		500 Kbps																																					
C	RW	LENGTH				Packet length configuration																																					
			Default	0		Default length. Effective length of LENGTH field in encrypted/decrypted packet is 5 bits. A key-stream for packet payloads up to 27 bytes will be generated.																																					

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
ID			C										B										B										A									
Reset 0x00000001			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1							
ID	Acce Field	Value ID	Value										Description																													
		Extended	1										Extended length. Effective length of LENGTH field in encrypted/decrypted packet is 8 bits. A key-stream for packet payloads up to MAXPACKETSIZE bytes will be generated.																													

6.4.9.14 CNFPTR

Address offset: 0x508

Pointer to data structure holding AES key and NONCE vector

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID	Value				Description																											
A	RW	CNFPTR						Pointer to the data structure holding the AES key and the CCM NONCE vector (see Table 1 CCM data structure overview)																											

6.4.9.15 INPTR

Address offset: 0x50C

Input pointer

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID	Value				Description																											
A	RW INPTR							Input pointer																											

6.4.9.16 OUTPTR

Address offset: 0x510

Output pointer

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A																															

6.4.9.17 SCRATCHPTR

Address offset: 0x514

Pointer to data area used for temporary storage

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ID	Acce Field	Value ID	Value	Description
A	RW	SCRATCHPTR		Pointer to a scratch data area used for temporary storage during key-stream generation, MIC generation and encryption/decryption. The scratch area is used for temporary storage of data during key-stream generation and encryption. When MODE.LENGTH = Default, a space of 43 bytes is required for this temporary storage. MODE.LENGTH = Extended (16 + MAXPACKETSIZE) bytes of storage is required.

6.4.9.18 MAXPACKETSIZE

Address offset: 0x518

Length of key-stream generated when MODE.LENGTH = Extended.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																										A	A	A	A	A	A	A
Reset 0x000000FB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1

ID	Acce Field	Value ID	Value	Description
A	RW	MAXPACKETSIZE	[0x001B..0x00FB]	Length of key-stream generated when MODE.LENGTH = Extended. This value must be greater or equal to the subsequent packet payload to be encrypted/decrypted.

6.4.9.19 RATEOVERRIDE

Address offset: 0x51C

Data rate override setting.

Override value to be used instead of the setting of MODE.DATARATE. This override value applies when the RATEOVERRIDE task is triggered.

Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																															A	A
Reset 0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ID	Acce Field	Value ID	Value	Description
A	RW	RATEOVERRIDE		Data rate override setting.
		1Mbit	0	1 Mbps
		2Mbit	1	2 Mbps
		125Kbps	2	125 Kbps
		500Kbps	3	500 Kbps

6.4.10 Electrical specification

6.4.10.1 Timing specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{kgen}	Time needed for key-stream generation (given priority access to destination RAM block).			50	μs

6.5 COMP — Comparator

The comparator (COMP) compares an input voltage (VIN+) against a second input voltage (VIN-). VIN+ can be derived from an analog input pin (AIN0-AIN7). VIN- can be derived from multiple sources depending on the operation mode of the comparator.

Main features of the comparator are:

- Input range from 0 V to VDD
- Single-ended mode
 - Fully flexible hysteresis using a 64-level reference ladder
- Differential mode
 - Configurable hysteresis
- Reference inputs (VREF):
 - VDD
 - External reference from AIN0 to AIN7 (between 0 V and VDD)
 - Internal references 1.2 V, 1.8 V and 2.4 V
- Three speed/power consumption modes: low-power, normal and high-speed
- Event generation on output changes
 - UP event on VIN- > VIN+
 - DOWN event on VIN- < VIN+
 - CROSS event on VIN+ and VIN- crossing
 - READY event on core and internal reference (if used) ready

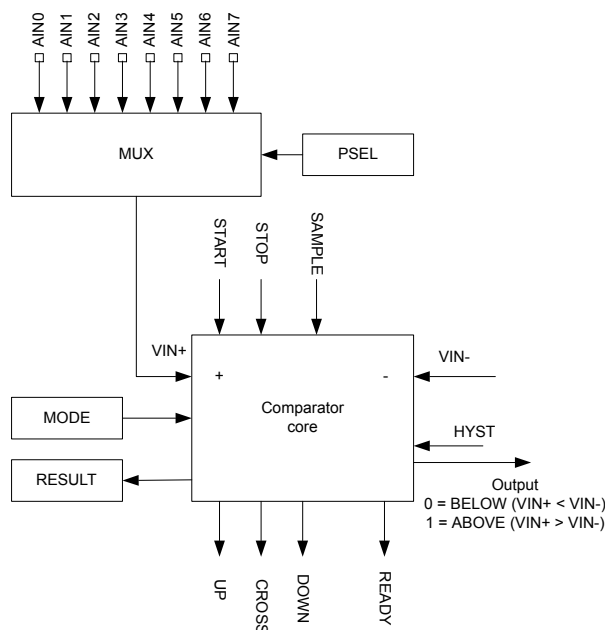


Figure 36: Comparator overview

Once enabled (using the [ENABLE](#) register), the comparator is started by triggering the START task and stopped by triggering the STOP task. The comparator will generate a READY event to indicate when it is ready for use and the output is correct. The delay between START and READY is $t_{INT_REF,START}$ if an internal reference is selected, or $t_{COMP,START}$ if an external reference is used. When the COMP module is started, events will be generated every time VIN+ crosses VIN-.

Operation modes

The comparator can be configured to operate in two main operation modes, differential mode and single-ended mode. See the [MODE](#) register for more information. In both operation modes, the comparator can operate in different speed and power consumption modes (low-power, normal and high-speed). High-speed mode will consume more power compared to low-power mode, and low-power mode will result in slower response time compared to high-speed mode.

Use the [PSEL](#) register to select any of the AIN0-AIN7 pins as VIN+ input, regardless of the operation mode selected for the comparator. The source of VIN- depends on which operation mode is used:

- Differential mode: Derived directly from AIN0 to AIN7
- Single-ended mode: Derived from VREF. VREF can be derived from VDD, AIN0-AIN7 or internal 1.2 V, 1.8 V and 2.4 V references.

The selected analog pins will be acquired by the comparator once it is enabled.

An optional hysteresis on VIN+ and VIN- can be enabled when the module is used in differential mode through the [HYST](#) register. In single-ended mode, VUP and VDOWN thresholds can be set to implement a hysteresis using the reference ladder (see [Comparator in single-ended mode](#) on page 123). This hysteresis is in the order of magnitude of $V_{DIFFHYST}$, and shall prevent noise on the signal to create unwanted events. See [Hysteresis example where VIN+ starts below VUP](#) on page 124 for illustration of the effect of an active hysteresis on a noisy input signal.

An upward crossing will generate an UP event and a downward crossing will generate a DOWN event. The CROSS event will be generated every time there is a crossing, independent of direction.

The immediate value of the comparator can be sampled to [RESULT](#) register by triggering the SAMPLE task.

6.5.1 Differential mode

In differential mode, the reference input VIN- is derived directly from one of the AINx pins.

Before enabling the comparator via the **ENABLE** register, the following registers must be configured for the differential mode:

- **PSEL**
- **MODE**
- **EXTREFSEL**

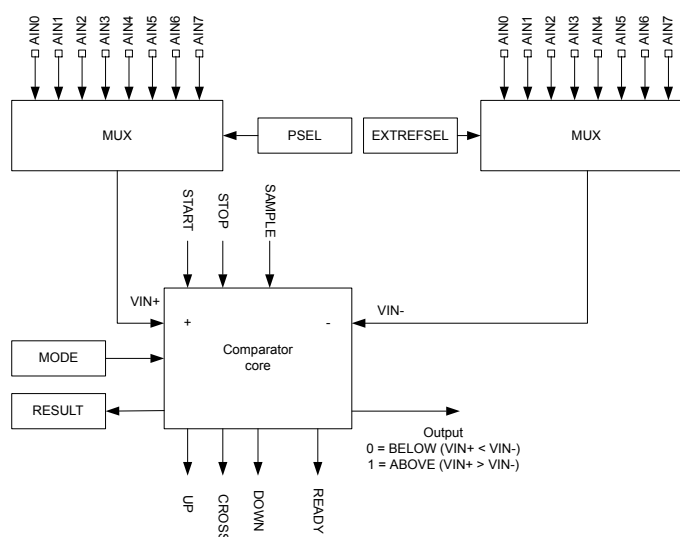


Figure 37: Comparator in differential mode

Note: Depending on the device, not all the analog inputs may be available for each MUX. See definitions for **PSEL** and **EXTREFSEL** for more information about which analog pins are available on a particular device.

When **HYST** register is turned on while in this mode, the output of the comparator (and associated events) will change from ABOVE to BELOW whenever VIN+ becomes lower than VIN- - ($V_{DIFFHYST} / 2$). It will also change from BELOW to ABOVE whenever VIN+ becomes higher than VIN- + ($V_{DIFFHYST} / 2$). This behavior is illustrated in **Hysteresis enabled in differential mode** on page 122.

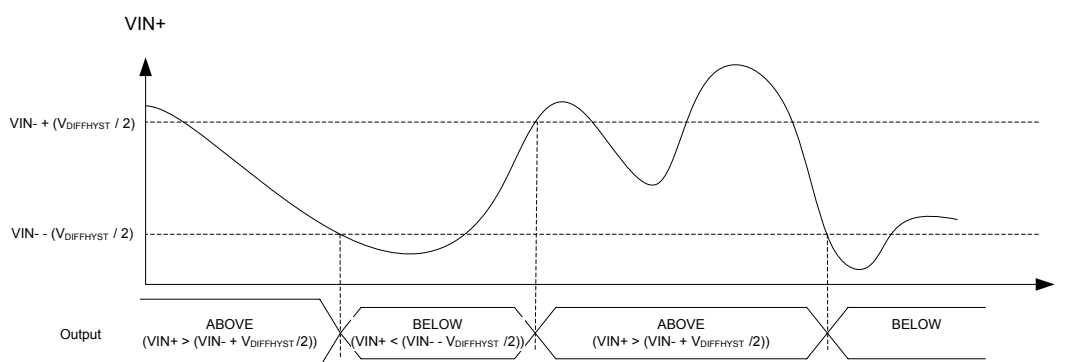


Figure 38: Hysteresis enabled in differential mode

6.5.2 Single-ended mode

In single-ended mode, VIN- is derived from the reference ladder.

Before enabling the comparator via the [ENABLE](#) register, the following registers must be configured for the single-ended mode:

- [PSEL](#)
- [MODE](#)
- [REFSEL](#)
- [EXTREFSEL](#)
- [TH](#)

The reference ladder uses the reference voltage (VREF) to derive two new voltage references, VUP and VDOWN. VUP and VDOWN are configured using THUP and THDOWN respectively in the [TH](#) register. VREF can be derived from any of the available reference sources, configured using the [EXTREFSEL](#) and [REFSEL](#) registers as illustrated in [Comparator in single-ended mode](#) on page 123. When AREF is selected in the [REFSEL](#) register, the [EXTREFSEL](#) register is used to select one of the AIN0-AIN7 analog input pins as reference input. The selected analog pins will be acquired by the comparator once it is enabled.

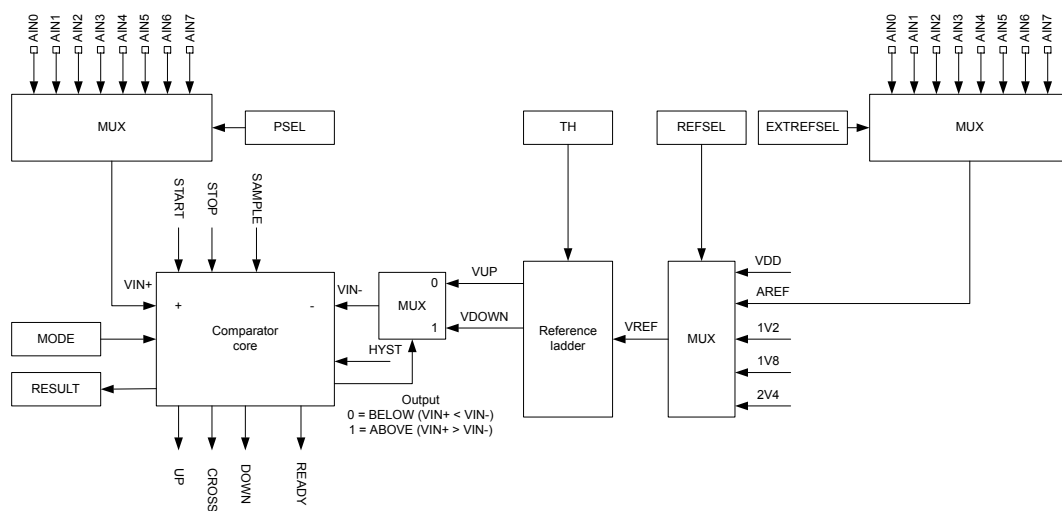


Figure 39: Comparator in single-ended mode

Note: Depending on the device, not all the analog inputs may be available for each MUX. See definitions for [PSEL](#) and [EXTREFSEL](#) for more information about which analog pins are available on a particular device.

When the comparator core detects that $VIN+ > VIN-$, i.e. ABOVE as per the [RESULT](#) register, $VIN-$ will switch to VDOWN. When $VIN+$ falls below $VIN-$ again, $VIN-$ will be switched back to VUP. By specifying VUP larger than VDOWN, a hysteresis can be generated as illustrated in [Hysteresis example where \$VIN+\$ starts below VUP](#) on page 124 and [Hysteresis example where \$VIN+\$ starts above VUP](#) on page 124.

Writing to [HYST](#) has no effect in single-ended mode, and the content of this register is ignored.

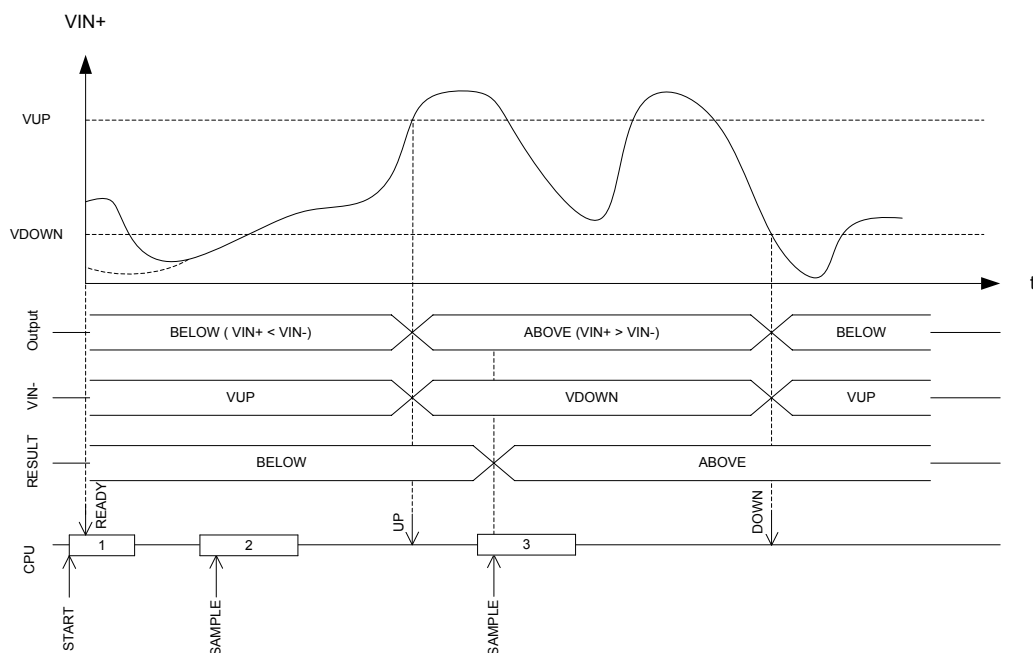


Figure 40: Hysteresis example where $VIN+$ starts below VUP

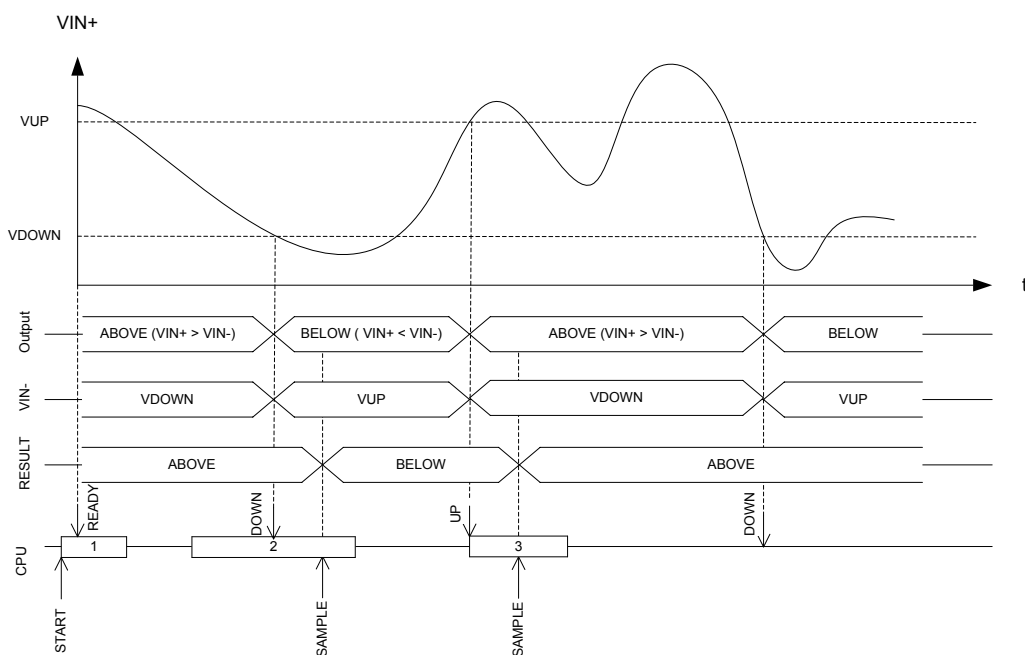


Figure 41: Hysteresis example where $VIN+$ starts above VUP

6.5.3 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40013000	COMP	COMP	General purpose comparator	

Table 32: Instances

Register	Offset	Description
TASKS_START	0x000	Start comparator

Register	Offset	Description
TASKS_STOP	0x004	Stop comparator
TASKS_SAMPLE	0x008	Sample comparator value
EVENTS_READY	0x100	COMP is ready and output is valid
EVENTS_DOWN	0x104	Downward crossing
EVENTS_UP	0x108	Upward crossing
EVENTS_CROSS	0x10C	Downward or upward crossing
SHORTS	0x200	Shortcuts between local events and tasks
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
RESULT	0x400	Compare result
ENABLE	0x500	COMP enable
PSEL	0x504	Pin select
REFSEL	0x508	Reference source select for single-ended mode
EXTREFSEL	0x50C	External reference select
TH	0x530	Threshold configuration for hysteresis unit
MODE	0x534	Mode configuration
HYST	0x538	Comparator hysteresis enable

Table 33: Register overview

6.5.3.1 TASKS_START

Address offset: 0x000

Start comparator

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																													
A	W	TASKS_START				Start comparator																													
			Trigger	1		Trigger task																													

6.5.3.2 TASKS_STOP

Address offset: 0x004

Stop comparator

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																													
A	W	TASKS_STOP				Stop comparator																													
			Trigger	1		Trigger task																													

6.5.3.3 TASKS_SAMPLE

Address offset: 0x008

Sample comparator value

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID																																				A
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																
ID	Acce Field		Value ID	Value				Description																												
A	W	TASKS_SAMPLE						Sample comparator value																												
		Trigger	1	Trigger task																																

6.5.3.4 EVENTS READY

Address offset: 0x100

COMP is ready and output is valid

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A																															
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																															
A	RW	EVENTS_READY		COMP is ready and output is valid																															
		NotGenerated	0	Event not generated																															
		Generated	1	Event generated																															

6.5.3.5 EVENTS_DOWN

Address offset: 0x104

Downward crossing

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A																															
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value				Description																											
A	RW	EVENTS_DOWN					Downward crossing																											
		NotGenerated	0				Event not generated																											
		Generated	1				Event generated																											

6.5.3.6 EVENTS_UP

Address offset: 0x108

Upward crossing

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID																																				A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ID	Acce Field	Value ID	Value	Description																																
A	RW	EVENTS_UP		Upward crossing																																
		NotGenerated	0	Event not generated																																
		Generated	1	Event generated																																

6.5.3.7 EVENTS CROSS

Address offset: 0x10C

Downward or upward crossing

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A																															
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																															
A	RW	EVENTS_CROSS		Downward or upward crossing																															
		NotGenerated	0	Event not generated																															
		Generated	1	Event generated																															

6.5.3.8 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				E D C B A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																													
A	RW	READY_SAMPLE				Shortcut between event READY and task SAMPLE																													
			Disabled	0	Disable shortcut																														
			Enabled	1	Enable shortcut																														
B	RW	READY_STOP				Shortcut between event READY and task STOP																													
			Disabled	0	Disable shortcut																														
			Enabled	1	Enable shortcut																														
C	RW	DOWN_STOP				Shortcut between event DOWN and task STOP																													
			Disabled	0	Disable shortcut																														
			Enabled	1	Enable shortcut																														
D	RW	UP_STOP				Shortcut between event UP and task STOP																													
			Disabled	0	Disable shortcut																														
			Enabled	1	Enable shortcut																														
E	RW	CROSS_STOP				Shortcut between event CROSS and task STOP																													
			Disabled	0	Disable shortcut																														
			Enabled	1	Enable shortcut																														

6.5.3.9 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID				D C B A																																		
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																		
ID	Acce Field		Value ID	Value		Description																																
A	RW	READY				Enable or disable interrupt for event READY																																
			Disabled	0	Disable																																	
			Enabled	1	Enable																																	
B	RW	DOWN				Enable or disable interrupt for event DOWN																																
			Disabled	0	Disable																																	
			Enabled	1	Enable																																	
C	RW	UP				Enable or disable interrupt for event UP																																
			Disabled	0	Disable																																	
			Enabled	1	Enable																																	
D	RW	CROSS				Enable or disable interrupt for event CROSS																																
			Disabled	0	Disable																																	

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			D C B A																															
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value				Description																											
		Enabled	1				Enable																											

6.5.3.10 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				D C B A																															
Reset 0x00000000				0 0																															

6.5.3.11 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID																																				D	C	B	A
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																			
ID	Acce	Field	Value	ID	Value	Description																																	
A	RW	READY				Write '1' to disable interrupt for event READY																																	
			Clear	1	Disable																																		
			Disabled	0	Read: Disabled																																		
			Enabled	1	Read: Enabled																																		
B	RW	DOWN				Write '1' to disable interrupt for event DOWN																																	
			Clear	1	Disable																																		
			Disabled	0	Read: Disabled																																		
			Enabled	1	Read: Enabled																																		
C	RW	UP				Write '1' to disable interrupt for event UP																																	
			Clear	1	Disable																																		
			Disabled	0	Read: Disabled																																		
			Enabled	1	Read: Enabled																																		

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
D	RW CROSS			Write '1' to disable interrupt for event CROSS																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														

6.5.3.12 RESULT

Address offset: 0x400

Compare result

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	R	RESULT		Result of last compare. Decision point SAMPLE task.																															
		Below	0	Input voltage is below the threshold (VIN+ < VIN-)																															
		Above	1	Input voltage is above the threshold (VIN+ > VIN-)																															

6.5.3.13 ENABLE

Address offset: 0x500

COMP enable

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW ENABLE			Enable or disable COMP																														
		Disabled	0	Disable																														
		Enabled	2	Enable																														

6.5.3.14 PSEL

Address offset: 0x504

Pin select

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A	RW	PSEL				Analog pin select																												
			AnalogInput0	0	AIN0 selected as analog input																													
			AnalogInput1	1	AIN1 selected as analog input																													
			AnalogInput2	2	AIN2 selected as analog input																													
			AnalogInput3	3	AIN3 selected as analog input																													
			AnalogInput4	4	AIN4 selected as analog input																													
			AnalogInput5	5	AIN5 selected as analog input																													
			AnalogInput6	6	AIN6 selected as analog input																													

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID																																		A	A	A		
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	Acce Field	Value ID	Value		Description																																	
		AnalogInput7	7		AIN7 selected as analog input																																	

6.5.3.15 REFSEL

Address offset: 0x508

Reference source select for single-ended mode

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A A A																															
Reset 0x00000004				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0																															
ID	Acce	Field	Value	ID	Value	Description																													
A	RW	REFSEL				Reference select																													
			Int1V2	0	VREF = internal 1.2 V reference (VDD >= 1.7 V)																														
			Int1V8	1	VREF = internal 1.8 V reference (VDD >= VREF + 0.2 V)																														
			Int2V4	2	VREF = internal 2.4 V reference (VDD >= VREF + 0.2 V)																														
			VDD	4	VREF = VDD																														
			ARef	5	VREF = AREF																														

6.5.3.16 EXTREFSEL

Address offset: 0x50C

External reference select

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A A A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																													
A	RW	EXTREFSEL				External analog reference select																													
			AnalogReference0	0		Use AIN0 as external analog reference																													
			AnalogReference1	1		Use AIN1 as external analog reference																													
			AnalogReference2	2		Use AIN2 as external analog reference																													
			AnalogReference3	3		Use AIN3 as external analog reference																													
			AnalogReference4	4		Use AIN4 as external analog reference																													
			AnalogReference5	5		Use AIN5 as external analog reference																													
			AnalogReference6	6		Use AIN6 as external analog reference																													
			AnalogReference7	7		Use AIN7 as external analog reference																													

6.5.3.17 TH

Address offset: 0x530

Threshold configuration for hysteresis unit

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				B B B B B B B B A A A A A A A A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	RW THDOWN		[63:0]	VDOWN = (THDOWN+1)/64*VREF																															
B	RW THUP		[63:0]	VUP = (THUP+1)/64*VREF																															

6.5.3.18 MODE

Address offset: 0x534

Mode configuration

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID																											B				A			
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A	RW	SP				Speed and power modes																												
			Low	0	Low-power mode																													
			Normal	1	Normal mode																													
			High	2	High-speed mode																													
B	RW	MAIN				Main operation modes																												
			SE	0	Single-ended mode																													
			Diff	1	Differential mode																													

6.5.3.19 HYST

Address offset: 0x538

Comparator hysteresis enable

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW HYST			Comparator hysteresis																														
		NoHyst	0	Comparator hysteresis disabled																														
		Hyst50mV	1	Comparator hysteresis enabled																														

6.5.4 Electrical specification

6.5.4.1 COMP Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$t_{\text{PROPDLV,LP}}$	Propagation delay, low-power mode ¹		0.6		μs
$t_{\text{PROPDLV,N}}$	Propagation delay, normal mode ¹		0.2		μs
$t_{\text{PROPDLV,HS}}$	Propagation delay, high-speed mode ¹		0.1		μs
V_{DIFFHYST}	Optional hysteresis applied to differential input	10	30	90	mV
$V_{\text{VDD-VREF}}$	Required difference between VDD and a selected VREF, VDD > VREF	0.3			V
$t_{\text{INT_REF,START}}$	Startup time for the internal bandgap reference		50	80	μs
$E_{\text{INT_REF}}$	Internal bandgap reference error	-3		3	%
$V_{\text{INPUTOFFSET}}$	Input offset	-15		15	mV
$t_{\text{COMP,START}}$	Startup time for the comparator core		3		μs

¹ Propagation delay is with 10 mV overdrive.

6.6 ECB — AES electronic codebook mode encryption

The AES electronic codebook mode encryption (ECB) can be used for a range of cryptographic functions like hash generation, digital signatures, and keystream generation for data encryption/decryption. The ECB encryption block supports 128 bit AES encryption (encryption only, not decryption).

AES ECB operates with EasyDMA access to system Data RAM for in-place operations on cleartext and ciphertext during encryption. ECB uses the same AES core as the CCM and AAR blocks and is an asynchronous operation which may not complete if the AES core is busy.

AES ECB features:

- 128 bit AES encryption
- Supports standard AES ECB block encryption
- Memory pointer support
- DMA data transfer

AES ECB performs a 128 bit AES block encrypt. At the STARTECB task, data and key is loaded into the algorithm by EasyDMA. When output data has been written back to memory, the ENDECB event is triggered.

AES ECB can be stopped by triggering the STOPECB task.

6.6.1 Shared resources

The ECB, CCM, and AAR share the same AES module. The ECB will always have lowest priority and if there is a sharing conflict during encryption, the ECB operation will be aborted and an ERRORECB event will be generated.

6.6.2 EasyDMA

The ECB implements an EasyDMA mechanism for reading and writing to the Data RAM. This DMA cannot access the program memory or any other parts of the memory area except RAM.

If the ECBDATAPTR is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 19 for more information about the different memory regions.

The EasyDMA will have finished accessing the Data RAM when the ENDECB or ERRORECB is generated.

6.6.3 ECB data structure

Input to the block encrypt and output from the block encrypt are stored in the same data structure. ECBDATAPTR should point to this data structure before STARTECB is initiated.

Property	Address offset	Description
KEY	0	16 byte AES key
CLEARTTEXT	16	16 byte AES cleartext input block
CIPHERTEXT	32	16 byte AES ciphertext output block

Table 34: ECB data structure overview

6.6.4 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4000E000	ECB	ECB	AES electronic code book (ECB) mode block encryption	

Table 35: Instances

Register	Offset	Description
TASKS_STARTECB	0x000	Start ECB block encrypt
TASKS_STOPECB	0x004	Abort a possible executing ECB operation
EVENTS_ENDECB	0x100	ECB block encrypt complete
EVENTS_ERRORECB	0x104	ECB block encrypt aborted because of a STOPECB task or due to an error
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ECBDATAPTR	0x504	ECB block encrypt memory pointers

Table 36: Register overview

6.6.4.1 TASKS_STARTECB

Address offset: 0x000

Start ECB block encrypt

If a crypto operation is already running in the AES core, the STARTECB task will not start a new encryption and an ERRORECB event will be triggered

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																														
A	W	TASKS_STARTECB			Start ECB block encrypt																														
					If a crypto operation is already running in the AES core, the STARTECB task will not start a new encryption and an ERRORECB event will be triggered																														
Trigger				1	Trigger task																														

6.6.4.2 TASKS_STOPECB

Address offset: 0x004

Abort a possible executing ECB operation

If a running ECB operation is aborted by STOPECB, the ERRORECB event is triggered.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																													
A	W	TASKS_STOPECB				Abort a possible executing ECB operation																													
						If a running ECB operation is aborted by STOPECB, the ERRORECB event is triggered.																													
			Trigger	1		Trigger task																													

6.6.4.3 EVENTS_ENDECB

Address offset: 0x100

ECB block encrypt complete

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
ID		A	
Reset 0x00000000		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
ID	Access Field	Value ID	Description
A	RW	EVENTS_ENDECB	ECB block encrypt complete
		NotGenerated	0 Event not generated
		Generated	1 Event generated

6.6.4.4 EVENTS_ERRORECB

Address offset: 0x104

ECB block encrypt aborted because of a STOPECB task or due to an error

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
ID		A	
Reset 0x00000000		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
ID	Access Field	Value ID	Description
A	RW	EVENTS_ERRORECB	ECB block encrypt aborted because of a STOPECB task or due to an error
		NotGenerated	0 Event not generated
		Generated	1 Event generated

6.6.4.5 INTENSET

Address offset: 0x304

Enable interrupt

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
ID		B A	
Reset 0x00000000		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
ID	Access Field	Value ID	Description
A	RW	ENDECB	Write '1' to enable interrupt for event ENDECB
		Set	1 Enable
		Disabled	0 Read: Disabled
		Enabled	1 Read: Enabled
B	RW	ERRORECB	Write '1' to enable interrupt for event ERRORECB
		Set	1 Enable
		Disabled	0 Read: Disabled
		Enabled	1 Read: Enabled

6.6.4.6 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
ID																																						B	A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	Acce Field	Value ID	Value		Description																																		
A	RW	ENDECB		Write '1' to disable interrupt for event ENDECB																																			
		Clear	1	Disable																																			
		Disabled	0	Read: Disabled																																			
		Enabled	1	Read: Enabled																																			
B	RW	ERRORECB		Write '1' to disable interrupt for event ERRORECB																																			
		Clear	1	Disable																																			
		Disabled	0	Read: Disabled																																			
		Enabled	1	Read: Enabled																																			

6.6.4.7 ECBDATAPTR

Address offset: 0x504

ECB block encrypt memory pointers

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID		Value				Description																											
A	RW	ECBDATAPTR						Pointer to the ECB data structure (see Table 1 ECB data structure overview)																											

6.6.5 Electrical specification

6.6.5.1 ECB Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t _{ECB}	Run time per 16 byte block in all modes			7.2	μs

6.7 EGU — Event generator unit

Event generator unit (EGU) provides support for interlayer signaling. This means providing support for atomic triggering of both CPU execution and hardware tasks, from both firmware (by CPU) and hardware (by PPI). This feature can be used for triggering CPU execution at a lower priority execution from a higher priority execution, or to handle a peripheral's interrupt service routine (ISR) execution at a lower priority for some of its events. However, triggering any priority from any priority is possible.

Listed here are the main EGU features:

- Software-enabled interrupt triggering
- Separate interrupt vectors for every EGU instance
- Up to 16 separate event flags per interrupt for multiplexing

Each instance of EGU implements a set of tasks which can individually be triggered to generate the corresponding event. For example, the corresponding event for TASKS_TRIGGER[n] is EVENTS_TRIGGERED[n]. See [Instances](#) on page 136 for a list of EGU instances.

6.7.1 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40014000	EGU	EGU0	Event generator unit 0	
0x40015000	EGU	EGU1	Event generator unit 1	
0x40016000	EGU	EGU2	Event generator unit 2	
0x40017000	EGU	EGU3	Event generator unit 3	
0x40018000	EGU	EGU4	Event generator unit 4	
0x40019000	EGU	EGU5	Event generator unit 5	

Table 37: Instances

Register	Offset	Description
TASKS_TRIGGER[0]	0x000	Trigger 0 for triggering the corresponding TRIGGERED[0] event
TASKS_TRIGGER[1]	0x004	Trigger 1 for triggering the corresponding TRIGGERED[1] event
TASKS_TRIGGER[2]	0x008	Trigger 2 for triggering the corresponding TRIGGERED[2] event
TASKS_TRIGGER[3]	0x00C	Trigger 3 for triggering the corresponding TRIGGERED[3] event
TASKS_TRIGGER[4]	0x010	Trigger 4 for triggering the corresponding TRIGGERED[4] event
TASKS_TRIGGER[5]	0x014	Trigger 5 for triggering the corresponding TRIGGERED[5] event
TASKS_TRIGGER[6]	0x018	Trigger 6 for triggering the corresponding TRIGGERED[6] event
TASKS_TRIGGER[7]	0x01C	Trigger 7 for triggering the corresponding TRIGGERED[7] event
TASKS_TRIGGER[8]	0x020	Trigger 8 for triggering the corresponding TRIGGERED[8] event
TASKS_TRIGGER[9]	0x024	Trigger 9 for triggering the corresponding TRIGGERED[9] event
TASKS_TRIGGER[10]	0x028	Trigger 10 for triggering the corresponding TRIGGERED[10] event
TASKS_TRIGGER[11]	0x02C	Trigger 11 for triggering the corresponding TRIGGERED[11] event
TASKS_TRIGGER[12]	0x030	Trigger 12 for triggering the corresponding TRIGGERED[12] event
TASKS_TRIGGER[13]	0x034	Trigger 13 for triggering the corresponding TRIGGERED[13] event
TASKS_TRIGGER[14]	0x038	Trigger 14 for triggering the corresponding TRIGGERED[14] event
TASKS_TRIGGER[15]	0x03C	Trigger 15 for triggering the corresponding TRIGGERED[15] event
EVENTS_TRIGGERED[0]	0x100	Event number 0 generated by triggering the corresponding TRIGGER[0] task
EVENTS_TRIGGERED[1]	0x104	Event number 1 generated by triggering the corresponding TRIGGER[1] task
EVENTS_TRIGGERED[2]	0x108	Event number 2 generated by triggering the corresponding TRIGGER[2] task
EVENTS_TRIGGERED[3]	0x10C	Event number 3 generated by triggering the corresponding TRIGGER[3] task
EVENTS_TRIGGERED[4]	0x110	Event number 4 generated by triggering the corresponding TRIGGER[4] task
EVENTS_TRIGGERED[5]	0x114	Event number 5 generated by triggering the corresponding TRIGGER[5] task
EVENTS_TRIGGERED[6]	0x118	Event number 6 generated by triggering the corresponding TRIGGER[6] task
EVENTS_TRIGGERED[7]	0x11C	Event number 7 generated by triggering the corresponding TRIGGER[7] task
EVENTS_TRIGGERED[8]	0x120	Event number 8 generated by triggering the corresponding TRIGGER[8] task
EVENTS_TRIGGERED[9]	0x124	Event number 9 generated by triggering the corresponding TRIGGER[9] task
EVENTS_TRIGGERED[10]	0x128	Event number 10 generated by triggering the corresponding TRIGGER[10] task
EVENTS_TRIGGERED[11]	0x12C	Event number 11 generated by triggering the corresponding TRIGGER[11] task
EVENTS_TRIGGERED[12]	0x130	Event number 12 generated by triggering the corresponding TRIGGER[12] task
EVENTS_TRIGGERED[13]	0x134	Event number 13 generated by triggering the corresponding TRIGGER[13] task
EVENTS_TRIGGERED[14]	0x138	Event number 14 generated by triggering the corresponding TRIGGER[14] task
EVENTS_TRIGGERED[15]	0x13C	Event number 15 generated by triggering the corresponding TRIGGER[15] task
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt

Table 38: Register overview

6.7.1.1 TASKS_TRIGGER[n] (n=0..15)

Address offset: 0x000 + (n × 0x4)

Trigger n for triggering the corresponding TRIGGERED[n] event

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID			A																																	
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
ID	Acce Field	Value ID	Value		Description																															
A	W	TASKS_TRIGGER		Trigger n for triggering the corresponding TRIGGERED[n] event																																
		Trigger	1	Trigger task																																

6.7.1.2 EVENTS_TRIGGERED[n] (n=0..15)

Address offset: 0x100 + (n × 0x4)

Event number n generated by triggering the corresponding TRIGGER[n] task

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	EVENTS_TRIGGERED		Event number n generated by triggering the corresponding TRIGGER[n] task																														
		NotGenerated	0	Event not generated																														
		Generated	1	Event generated																														

6.7.1.3 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				P O N M L K J I H G F E D C B A																															
Reset 0x00000000				0 0																															

6.7.1.4 INTENSET

Address offset: 0x304

Enable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				P O N M L K J I H G F E D C B A																															
Reset 0x00000000				0 0																															

6.7.1.5 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			P O N M L K J I H G F E D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A-P	RW	TRIGGERED[i] (i=0..15)		Write '1' to disable interrupt for event TRIGGERED[i]																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														

6.7.2 Electrical specification

6.7.2.1 EGU Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$t_{\text{EGU,EVT}}$	Latency between setting an EGU event flag and the system setting an interrupt		1		cycles

6.8 GPIO — General purpose input/output

The general purpose input/output pins (GPIOs) are grouped as one or more ports, with each port having up to 32 GPIOs.

The number of ports and GPIOs per port varies with product variant and package. Refer to [Registers](#) on page 141 and [Pin assignments](#) on page 557 for more information about the number of GPIOs that are supported.

GPIO has the following user-configurable features:

- Up to 32 GPIO pins per GPIO port
- Output drive strength
- Internal pull-up and pull-down resistors
- Wake-up from high or low level triggers on all pins
- Trigger interrupt on state changes on any pin
- All pins can be used by the PPI task/event system
- One or more GPIO outputs can be controlled through the PPI and GPIOTE channels
- Any pin can be mapped to a peripheral for layout flexibility
- GPIO state changes captured on the SENSE signal can be stored by the LATCH register

The GPIO port peripheral implements up to 32 pins, PIN_0 through PIN_{31} . Each of these pins can be individually configured in the $\text{PIN_CNF}[n]$ registers ($n=0..31$).

The following parameters can be configured through these registers:

- Direction
- Drive strength
- Enabling of pull-up and pull-down resistors
- Pin sensing
- Input buffer disconnect

- Analog input (for selected pins)

The PIN_CNF registers are retained registers. See [POWER — Power supply](#) on page 58 for more information about retained registers.

6.8.1 Pin configuration

Pins can be individually configured through the SENSE field in the PIN_CNF[n] register to detect either a high or low level input.

When the correct level is detected on a configured pin, the sense mechanism will set the DETECT signal high. Each pin has a separate DETECT signal. Default behavior, defined by the DETECTMODE register, combines all DETECT signals from the pins in the GPIO port into one common DETECT signal and routes it through the system to be utilized by other peripherals. This mechanism is functional in both System ON and System OFF mode. See [GPIO port and the GPIO pin details](#) on page 139.

The following figure illustrates the GPIO port containing 32 individual pins, where PIN0 is shown in more detail for reference. All signals on the left side of the illustration are used by other peripherals in the system and therefore not directly available to the CPU.

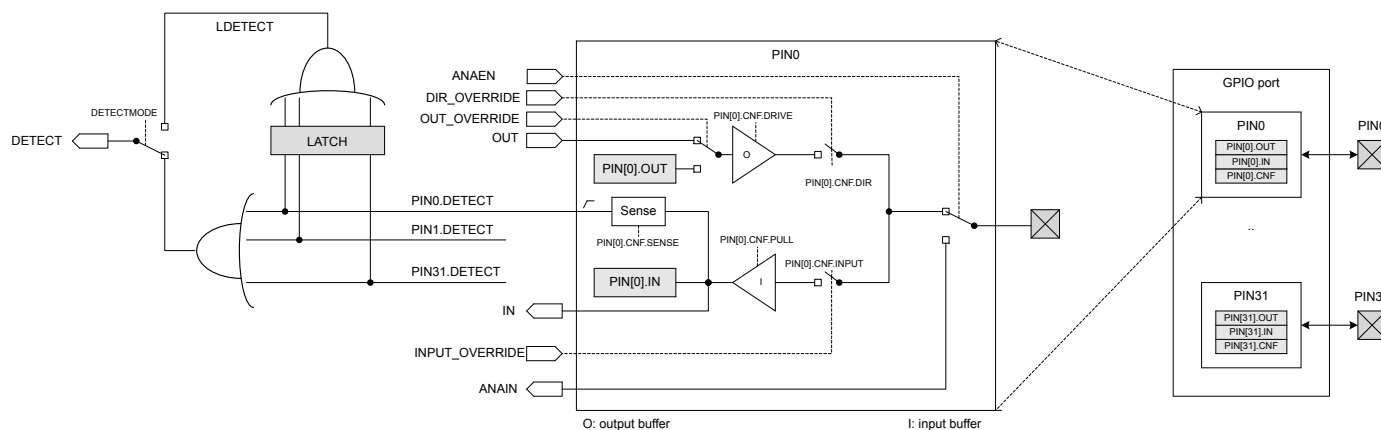


Figure 42: GPIO port and the GPIO pin details

Pins should be in a level that cannot trigger the sense mechanism before being enabled. If the SENSE condition configured in the PIN_CNF registers is met when the sense mechanism is enabled, the DETECT signal will immediately go high. A PORT event is triggered if the DETECT signal was low before enabling the sense mechanism. See [GPIOTE — GPIO tasks and events](#) on page 146.

See the following peripherals for more information about how the DETECT signal is used:

- [POWER — Power supply](#) on page 58 - uses the DETECT signal to exit from System OFF mode.
- [GPIOTE — GPIO tasks and events](#) on page 146 - uses the DETECT signal to generate the PORT event.

When a pin's PINx.DETECT signal goes high, a flag is set in the LATCH register. For example, when the PIN0.DETECT signal goes high, bit 0 in the LATCH register is set to 1. If the CPU performs a clear operation on a bit in the LATCH register when the associated PINx.DETECT signal is high, the bit in the LATCH register will not be cleared. The LATCH register will only be cleared if the CPU explicitly clears it by writing a 1 to the bit that shall be cleared, i.e. the LATCH register will not be affected by a PINx.DETECT signal being set low.

The LDETECT signal will be set high when one or more bits in the LATCH register are 1. The LDETECT signal will be set low when all bits in the LATCH register are successfully cleared to 0.

If one or more bits in the LATCH register are 1 after the CPU has performed a clear operation on the LATCH register, a rising edge will be generated on the LDETECT signal. This is illustrated in [DETECT signal behavior](#) on page 140.

Note: The CPU can read the LATCH register at any time to check if a SENSE condition has been met on any of the GPIO pins. This is still valid if that condition is no longer met at the time the CPU queries the LATCH register. This mechanism will work even if the LDETECT signal is not used as the DETECT signal.

The LDETECT signal is by default not connected to the GPIO port's DETECT signal, but via the DETECTMODE register. It is possible to change from default behavior to the DETECT signal that is derived directly from the LDETECT signal. See [GPIO port and the GPIO pin details](#) on page 139. The following figure illustrates the DETECT signal behavior for these two alternatives.

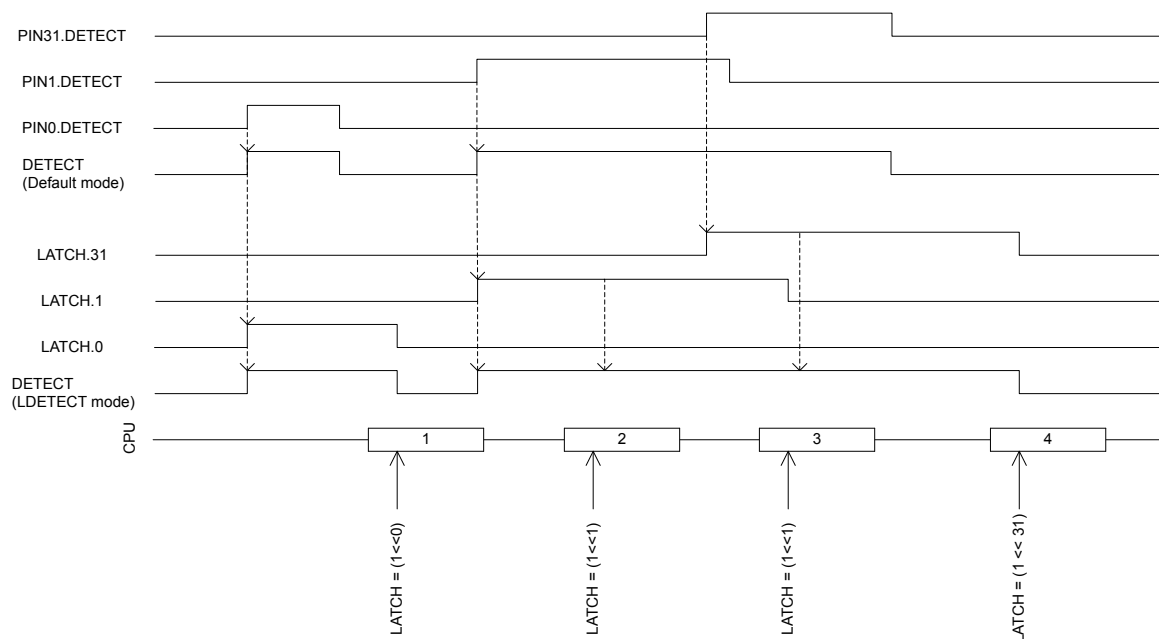


Figure 43: DETECT signal behavior

A GPIO pin input buffer can be disconnected from the pin to enable power savings when the pin is not used as an input, see [GPIO port and the GPIO pin details](#) on page 139. Input buffers must be connected to get a valid input value in the IN register, and for the sense mechanism to get access to the pin.

Other peripherals in the system can connect to GPIO pins and override their output value and configuration, or read their analog or digital input value. See [GPIO port and the GPIO pin details](#) on page 139.

Selected pins also support analog input signals, see ANAIN in [GPIO port and the GPIO pin details](#) on page 139. The assignment of the analog pins can be found in [Pin assignments](#) on page 557.

Note: When a pin is configured as digital input, increased current consumption occurs when the input voltage is between V_{IL} and V_{IH} . It is good practice to ensure that the external circuitry does not drive that pin to levels between V_{IL} and V_{IH} for a long period of time.

6.8.2 Registers

Base address	Peripheral	Instance	Description	Configuration
0x50000000	GPIO	GPIO	General purpose input and output	Deprecated
0x50000000	GPIO	P0	General purpose input and output, port 0	P0.00 to P0.31 implemented
0x50000300	GPIO	P1	General purpose input and output, port 1	P1.00 to P1.09 implemented

Table 39: Instances

Register	Offset	Description
OUT	0x504	Write GPIO port
OUTSET	0x508	Set individual bits in GPIO port
OUTCLR	0x50C	Clear individual bits in GPIO port
IN	0x510	Read GPIO port
DIR	0x514	Direction of GPIO pins
DIRSET	0x518	DIR set register
DIRCLR	0x51C	DIR clear register
LATCH	0x520	Latch register indicating what GPIO pins that have met the criteria set in the PIN_CNF[n].SENSE registers
DETECTMODE	0x524	Select between default DETECT signal behavior and LDETECT mode
PIN_CNF[0]	0x700	Configuration of GPIO pins
PIN_CNF[1]	0x704	Configuration of GPIO pins
PIN_CNF[2]	0x708	Configuration of GPIO pins
PIN_CNF[3]	0x70C	Configuration of GPIO pins
PIN_CNF[4]	0x710	Configuration of GPIO pins
PIN_CNF[5]	0x714	Configuration of GPIO pins
PIN_CNF[6]	0x718	Configuration of GPIO pins
PIN_CNF[7]	0x71C	Configuration of GPIO pins
PIN_CNF[8]	0x720	Configuration of GPIO pins
PIN_CNF[9]	0x724	Configuration of GPIO pins
PIN_CNF[10]	0x728	Configuration of GPIO pins
PIN_CNF[11]	0x72C	Configuration of GPIO pins
PIN_CNF[12]	0x730	Configuration of GPIO pins
PIN_CNF[13]	0x734	Configuration of GPIO pins
PIN_CNF[14]	0x738	Configuration of GPIO pins
PIN_CNF[15]	0x73C	Configuration of GPIO pins
PIN_CNF[16]	0x740	Configuration of GPIO pins
PIN_CNF[17]	0x744	Configuration of GPIO pins
PIN_CNF[18]	0x748	Configuration of GPIO pins
PIN_CNF[19]	0x74C	Configuration of GPIO pins
PIN_CNF[20]	0x750	Configuration of GPIO pins
PIN_CNF[21]	0x754	Configuration of GPIO pins
PIN_CNF[22]	0x758	Configuration of GPIO pins
PIN_CNF[23]	0x75C	Configuration of GPIO pins
PIN_CNF[24]	0x760	Configuration of GPIO pins
PIN_CNF[25]	0x764	Configuration of GPIO pins
PIN_CNF[26]	0x768	Configuration of GPIO pins
PIN_CNF[27]	0x76C	Configuration of GPIO pins
PIN_CNF[28]	0x770	Configuration of GPIO pins
PIN_CNF[29]	0x774	Configuration of GPIO pins
PIN_CNF[30]	0x778	Configuration of GPIO pins

Register	Offset	Description
PIN_CNF[31]	0x77C	Configuration of GPIO pins

Table 40: Register overview

6.8.2.1 OUT

Address offset: 0x504

Write GPIO port

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce	Field	Value	ID	Value	Description																													
A-f	RW	PIN[i] (i=0..31)				Pin i																													
		Low	0			Pin driver is low																													
		High	1			Pin driver is high																													

6.8.2.2 OUTSET

Address offset: 0x508

Set individual bits in GPIO port

Read: reads value of OUT register.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce	Field	Value	ID	Value	Description																													
A-f	RW	PIN[i] (i=0..31)				Pin i																													
		Low	0			Read: pin driver is low																													
		High	1			Read: pin driver is high																													
		Set	1			Write: a '1' sets the pin high; a '0' has no effect																													

6.8.2.3 OUTCLR

Address offset: 0x50C

Clear individual bits in GPIO port

Read: reads value of OUT register.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value			Description																												
A-f	RW	PIN[i] (i=0..31)				Pin i																												
		Low	0			Read: pin driver is low																												
		High	1			Read: pin driver is high																												
		Clear	1			Write: a '1' sets the pin low; a '0' has no effect																												

6.8.2.4 IN

Address offset: 0x510

Read GPIO port

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																															
A-f	R	PIN[i] (i=0..31)		Pin i																															
		Low	0	Pin input is low																															
		High	1	Pin input is high																															

6.8.2.5 DIR

Address offset: 0x514

Direction of GPIO pins

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																															
A-f	RW	PIN[i] (i=0..31)		Pin i																															
		Input	0	Pin set as input																															
		Output	1	Pin set as output																															

6.8.2.6 DIRSET

Address offset: 0x518

DIR set register

Read: reads value of DIR register.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																															
A-f	RW	PIN[i] (i=0..31)		Set as output pin i																															
		Input	0	Read: pin set as input																															
		Output	1	Read: pin set as output																															
		Set	1	Write: a '1' sets pin to output; a '0' has no effect																															

6.8.2.7 DIRCLR

Address offset: 0x51C

DIR clear register

Read: reads value of DIR register.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			E E																D D D			C C B A												
Reset 0x00000002			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0																															
ID	Acce	Field	Value ID	Value	Description																													
C	RW	PULL			Pull configuration																													
			Disabled	0	No pull																													
			Pulldown	1	Pull down on pin																													
			Pullup	3	Pull up on pin																													
D	RW	DRIVE			Drive configuration																													
			S0S1	0	Standard '0', standard '1'																													
			H0S1	1	High drive '0', standard '1'																													
			S0H1	2	Standard '0', high drive '1'																													
			H0H1	3	High drive '0', high 'drive '1"																													
			D0S1	4	Disconnect '0' standard '1' (normally used for wired-or connections)																													
			D0H1	5	Disconnect '0', high drive '1' (normally used for wired-or connections)																													
			S0D1	6	Standard '0'. disconnect '1' (normally used for wired-and connections)																													
			H0D1	7	High drive '0', disconnect '1' (normally used for wired-and connections)																													
E	RW	SENSE			Pin sensing mechanism																													
			Disabled	0	Disabled																													
			High	2	Sense for high level																													
			Low	3	Sense for low level																													

6.8.3 Electrical specification

6.8.3.1 GPIO Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
V_{IH}	Input high voltage	0.7 x VDD		VDD	V
V_{IL}	Input low voltage	VSS		0.3 x VDD	V
$V_{OH,SD}$	Output high voltage, standard drive, 0.5 mA, VDD $\geq$ 1.7	VDD - 0.4		VDD	V
$V_{OH,HDL}$	Output high voltage, high drive, 5 mA, VDD $\geq$ 2.7 V	VDD - 0.4		VDD	V
$V_{OH,HDL}$	Output high voltage, high drive, 3 mA, VDD $\geq$ 1.7 V	VDD - 0.4		VDD	V
$V_{OL,SD}$	Output low voltage, standard drive, 0.5 mA, VDD $\geq$ 1.7	VSS		VSS + 0.4	V
$V_{OL,HDL}$	Output low voltage, high drive, 5 mA, VDD $\geq$ 2.7 V	VSS		VSS + 0.4	V
$V_{OL,HDL}$	Output low voltage, high drive, 3 mA, VDD $\geq$ 1.7 V	VSS		VSS + 0.4	V
$I_{OL,SD}$	Current at VSS+0.4 V, output set low, standard drive, VDD $\geq$ 1.7	1	2	4	mA
$I_{OL,HDL}$	Current at VSS+0.4 V, output set low, high drive, VDD $\geq$ 2.7 V	6	10	15	mA
$I_{OL,HDL}$	Current at VSS+0.4 V, output set low, high drive, VDD $\geq$ 1.7 V	3			mA
$I_{OH,SD}$	Current at VDD-0.4 V, output set high, standard drive, VDD $\geq$ 1.7	1	2	4	mA
$I_{OH,HDL}$	Current at VDD-0.4 V, output set high, high drive, VDD $\geq$ 2.7 V	6	9	14	mA
$I_{OH,HDL}$	Current at VDD-0.4 V, output set high, high drive, VDD $\geq$ 1.7 V	3			mA

Symbol	Description	Min.	Typ.	Max.	Units
$t_{RF,15pF}$	Rise/fall time, standard drive mode, 10-90%, 15 pF load ¹⁴		9		ns
$t_{RF,25pF}$	Rise/fall time, standard drive mode, 10-90%, 25 pF load ¹⁴		13		ns
$t_{RF,50pF}$	Rise/fall time, standard drive mode, 10-90%, 50 pF load ¹⁴		25		ns
$t_{HRF,15pF}$	Rise/Fall time, high drive mode, 10-90%, 15 pF load ¹⁴		4		ns
$t_{HRF,25pF}$	Rise/Fall time, high drive mode, 10-90%, 25 pF load ¹⁴		5		ns
$t_{HRF,50pF}$	Rise/Fall time, high drive mode, 10-90%, 50 pF load ¹⁴		8		ns
R_{PU}	Pull-up resistance	11	13	16	k Ω
R_{PD}	Pull-down resistance	11	13	16	k Ω
C_{PAD}	Pad capacitance		3		pF

6.8.3.2 NFC Pads Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
C_{PAD_NFC}	Pad capacitance on NFC pads		4		pF
I_{NFC_LEAK}	Leakage current between NFC pads when driven to different states		1	10	μ A
$I_{NFC_LEAK_EXT}$	Leakage current between NFC pads when driven to different states, extended temperature range		1	15	μ A

6.9 GPIOTE — GPIO tasks and events

The GPIO tasks and events (GPIOTE) module provides functionality for accessing GPIO pins using tasks and events. Each GPIOTE channel can be assigned to one pin.

A GPIOTE block enables GPIOs to generate events on pin state change which can be used to carry out tasks through the PPI system. A GPIO can also be driven to change state on system events using the PPI system. Tasks and events are briefly introduced in [Peripheral interface](#) on page 96, and GPIO is described in more detail in [GPIO — General purpose input/output](#) on page 138.

Low power detection of pin state changes is possible when in System ON or System OFF.

Instance	Number of GPIOTE channels
GPIOTE	8

Table 41: GPIOTE properties

Up to three tasks can be used in each GPIOTE channel for performing write operations to a pin. Two tasks are fixed (SET and CLR), and one (OUT) is configurable to perform following operations:

- Set
- Clear
- Toggle

An event can be generated in each GPIOTE channel from one of the following input conditions:

- Rising edge
- Falling edge
- Any change

¹⁴ Rise and fall times based on simulations

6.9.1 Pin events and tasks

The GPIOTE module has a number of tasks and events that can be configured to operate on individual GPIO pins.

The tasks SET[n], CLR[n], and OUT[n] can write to individual pins, and events IN[n] can be generated from input changes of individual pins.

The SET task will set the pin selected in GPIOTE.CONFIG[n].PSEL to high. The CLR task will set the pin low.

The effect of the OUT task on the pin is configurable in CONFIG[n].POLARITY. It can set the pin high, set it low, or toggle it.

Tasks and events are configured using the CONFIG[n] registers. One CONFIG[n] register is associated with a set of SET[n], CLR[n], and OUT[n] tasks and IN[n] events.

As long as a SET[n], CLR[n], and OUT[n] task or an IN[n] event is configured to control pin **n**, the pin's output value will only be updated by the GPIOTE module. The pin's output value, as specified in the GPIO, will be ignored as long as the pin is controlled by GPIOTE. Attempting to write to the pin as a normal GPIO pin will have no effect. When the GPIOTE is disconnected from a pin, the associated pin gets the output and configuration values specified in the GPIO module, see MODE field in CONFIG[n] register.

When conflicting tasks are triggered simultaneously (i.e. during the same clock cycle) in one channel, the priority of the tasks is as described in the following table.

Priority	Task
1	OUT
2	CLR
3	SET

Table 42: Task priorities

When setting the CONFIG[n] registers, MODE=Disabled does not have the same effect as MODE=Task and POLARITY=None. In the latter case, a CLR or SET task occurring at the exact same time as OUT will end up with no change on the pin, based on the priorities described in the table above.

When a GPIOTE channel is configured to operate on a pin as a task, the initial value of that pin is configured in the OUTINIT field of CONFIG[n].

6.9.2 Port event

PORT is an event that can be generated from multiple input pins using the GPIO DETECT signal.

The event will be generated on the rising edge of the DETECT signal. See [GPIO — General purpose input/output](#) on page 138 for more information about the DETECT signal.

The GPIO DETECT signal will not wake the system up again if the system is put into System ON IDLE while the DETECT signal is high. Clear all DETECT sources before entering sleep. If the LATCH register is used as a source, a new rising edge will be generated on DETECT if any bit in LATCH is still high after clearing all or part of the register. This could occur if one of the PINx.DETECT signals is still high, for example. See [Pin configuration](#) on page 139 for more information.

Setting the system to System OFF while DETECT is high will cause a wakeup from System OFF reset.

This feature is always enabled even if the peripheral itself appears to be IDLE, meaning no clocks or other power intensive infrastructure have to be requested to keep this feature enabled. This feature can therefore be used to wake up the CPU from a WFI or WFE type sleep in System ON when all peripherals and the CPU are idle, meaning the lowest power consumption in System ON mode.

In order to prevent spurious interrupts from the PORT event while configuring the sources, the following must be performed:

1. Disable interrupts on the PORT event (through INTENCLR.PORT).
2. Configure the sources (PIN_CNF[n].SENSE).
3. Clear any potential event that could have occurred during configuration (write '0' to EVENTS_PORT).
4. Enable interrupts (through INTENSET.PORT).

6.9.3 Tasks and events pin configuration

Each GPIOTE channel is associated with one physical GPIO pin through the CONFIG.PSEL field.

When Event mode is selected in CONFIG.MODE, the pin specified by CONFIG.PSEL will be configured as an input, overriding the DIR setting in GPIO. Similarly, when Task mode is selected in CONFIG.MODE, the pin specified by CONFIG.PSEL will be configured as an output overriding the DIR setting and OUT value in GPIO. When Disabled is selected in CONFIG.MODE, the pin specified by CONFIG.PSEL will use its configuration from the PIN[n].CNF registers in GPIO.

Note: A pin can only be assigned to one GPIOTE channel at a time. Failing to do so may result in unpredictable behavior.

6.9.4 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40006000	GPIOTE	GPIOTE	GPIO tasks and events	

Table 43: Instances

Register	Offset	Description
TASKS_OUT[0]	0x000	Task for writing to pin specified in CONFIG[0].PSEL. Action on pin is configured in CONFIG[0].POLARITY.
TASKS_OUT[1]	0x004	Task for writing to pin specified in CONFIG[1].PSEL. Action on pin is configured in CONFIG[1].POLARITY.
TASKS_OUT[2]	0x008	Task for writing to pin specified in CONFIG[2].PSEL. Action on pin is configured in CONFIG[2].POLARITY.
TASKS_OUT[3]	0x00C	Task for writing to pin specified in CONFIG[3].PSEL. Action on pin is configured in CONFIG[3].POLARITY.
TASKS_OUT[4]	0x010	Task for writing to pin specified in CONFIG[4].PSEL. Action on pin is configured in CONFIG[4].POLARITY.
TASKS_OUT[5]	0x014	Task for writing to pin specified in CONFIG[5].PSEL. Action on pin is configured in CONFIG[5].POLARITY.
TASKS_OUT[6]	0x018	Task for writing to pin specified in CONFIG[6].PSEL. Action on pin is configured in CONFIG[6].POLARITY.
TASKS_OUT[7]	0x01C	Task for writing to pin specified in CONFIG[7].PSEL. Action on pin is configured in CONFIG[7].POLARITY.
TASKS_SET[0]	0x030	Task for writing to pin specified in CONFIG[0].PSEL. Action on pin is to set it high.
TASKS_SET[1]	0x034	Task for writing to pin specified in CONFIG[1].PSEL. Action on pin is to set it high.
TASKS_SET[2]	0x038	Task for writing to pin specified in CONFIG[2].PSEL. Action on pin is to set it high.
TASKS_SET[3]	0x03C	Task for writing to pin specified in CONFIG[3].PSEL. Action on pin is to set it high.
TASKS_SET[4]	0x040	Task for writing to pin specified in CONFIG[4].PSEL. Action on pin is to set it high.
TASKS_SET[5]	0x044	Task for writing to pin specified in CONFIG[5].PSEL. Action on pin is to set it high.
TASKS_SET[6]	0x048	Task for writing to pin specified in CONFIG[6].PSEL. Action on pin is to set it high.
TASKS_SET[7]	0x04C	Task for writing to pin specified in CONFIG[7].PSEL. Action on pin is to set it high.
TASKS_CLR[0]	0x060	Task for writing to pin specified in CONFIG[0].PSEL. Action on pin is to set it low.
TASKS_CLR[1]	0x064	Task for writing to pin specified in CONFIG[1].PSEL. Action on pin is to set it low.
TASKS_CLR[2]	0x068	Task for writing to pin specified in CONFIG[2].PSEL. Action on pin is to set it low.

6.9.4.1 TASKS OUT[n] (n=0..7)

Task for writing to pin specified in CONFIG[n].PSEL. Action on pin is configured in CONFIG[n].POLARITY.

6.9.4.2 TASKS SET[n] (n=0..7)

Task for writing to pin specified in CONFIG[n].PSEL. Action on pin is to set it high.



NORDIC
SEMICONDUCTOR

6.9.4.3 TASKS_CLR[n] (n=0..7)

Address offset: $0x060 + (n \times 0x4)$

Task for writing to pin specified in CONFIG[n].PSEL. Action on pin is to set it low.

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
ID		A	
Reset 0x00000000		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
ID	Acce Field	Value ID	Description
A	W TASKS_CLR		Task for writing to pin specified in CONFIG[n].PSEL. Action on pin is to set it low.
		Trigger	1 Trigger task

6.9.4.4 EVENTS_IN[n] (n=0..7)

Address offset: $0x100 + (n \times 0x4)$

Event generated from pin specified in CONFIG[n].PSEL

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
ID		A	
Reset 0x00000000		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
ID	Acce Field	Value ID	Description
A	RW EVENTS_IN		Event generated from pin specified in CONFIG[n].PSEL
		NotGenerated	0 Event not generated
		Generated	1 Event generated

6.9.4.5 EVENTS_PORT

Address offset: 0x17C

Event generated from multiple input GPIO pins with SENSE mechanism enabled

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
ID		A	
Reset 0x00000000		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
ID	Acce Field	Value ID	Description
A	RW EVENTS_PORT		Event generated from multiple input GPIO pins with SENSE mechanism enabled
		NotGenerated	0 Event not generated
		Generated	1 Event generated

6.9.4.6 INTENSET

Address offset: 0x304

Enable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			I																								H G F E D C B A							
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A-H	RW	IN[i] (i=0..7)		Write '1' to enable interrupt for event IN[i]																														
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
I	RW	PORT		Write '1' to enable interrupt for event PORT																														
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														

6.9.4.7 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			I																								H G F E D C B A							
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A-H	RW	IN[i] (i=0..7)				Write '1' to disable interrupt for event IN[i]																												
		Clear	1		Disable																													
		Disabled	0		Read: Disabled																													
		Enabled	1		Read: Enabled																													
I	RW	PORT				Write '1' to disable interrupt for event PORT																												
		Clear	1		Disable																													
		Disabled	0		Read: Disabled																													
		Enabled	1		Read: Enabled																													

6.9.4.8 CONFIG[n] (n=0..7)

Address offset: 0x510 + (n × 0x4)

Configuration for OUT[n], SET[n], and CLR[n] tasks and IN[n] event

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																							
ID			E																D D		C B B B B B																				A A	
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																							
ID	Acce	Field	Value	ID	Value	Description																																				
A	RW	MODE				Mode																																				
		Disabled	0	Disabled. Pin specified by PSEL will not be acquired by the GPIOTE module.																																						
		Event	1	Event mode																																						
				The pin specified by PSEL will be configured as an input and the IN[n] event will be generated if operation specified in POLARITY occurs on the pin.																																						

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			E D D C B B B B																A A															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																													
			Task	3	Task mode																													
					The GPIO specified by PSEL will be configured as an output and triggering the SET[n], CLR[n] or OUT[n] task will perform the operation specified by POLARITY on the pin. When enabled as a task the GPIOTE module will acquire the pin and the pin can no longer be written as a regular output pin from the GPIO module.																													
B	RW	PSEL		[0..31]	GPIO number associated with SET[n], CLR[n], and OUT[n] tasks and IN[n] event																													
C	RW	PORT		[0..1]	Port number																													
D	RW	POLARITY			When In task mode: Operation to be performed on output when OUT[n] task is triggered. When In event mode: Operation on input that shall trigger IN[n] event.																													
			None	0	Task mode: No effect on pin from OUT[n] task. Event mode: no IN[n] event generated on pin activity.																													
			LoToHi	1	Task mode: Set pin from OUT[n] task. Event mode: Generate IN[n] event when rising edge on pin.																													
			HiToLo	2	Task mode: Clear pin from OUT[n] task. Event mode: Generate IN[n] event when falling edge on pin.																													
			Toggle	3	Task mode: Toggle pin from OUT[n]. Event mode: Generate IN[n] when any change on pin.																													
E	RW	OUTINIT			When in task mode: Initial value of the output when the GPIOTE channel is configured. When in event mode: No effect.																													
			Low	0	Task mode: Initial value of pin before task triggering is low																													
			High	1	Task mode: Initial value of pin before task triggering is high																													

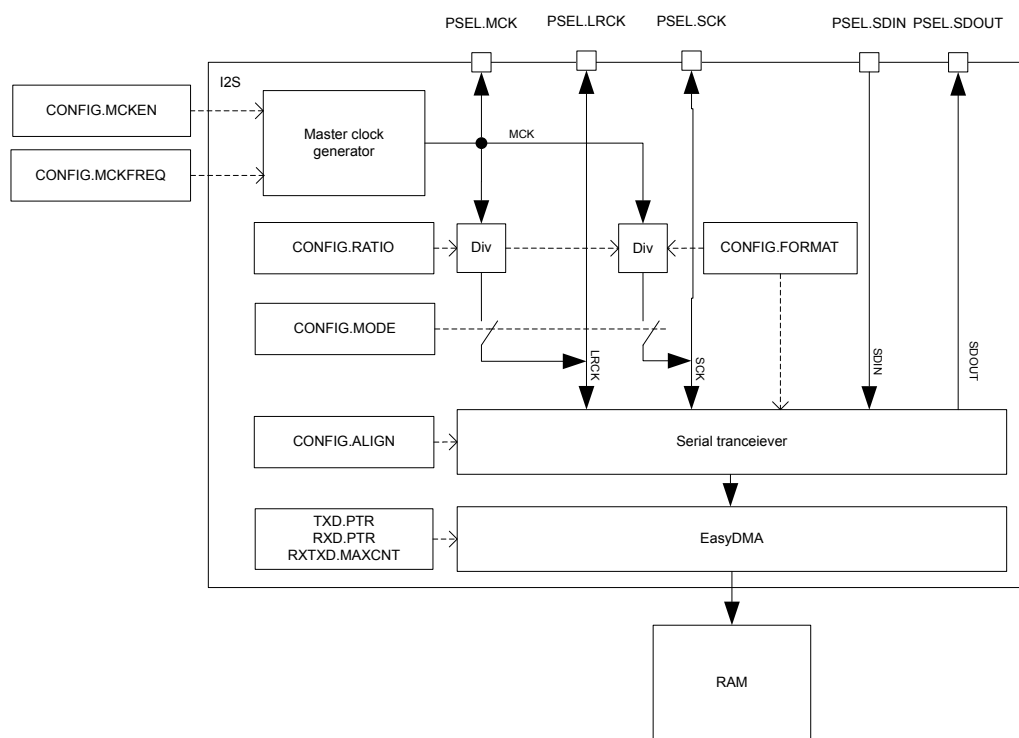
6.9.5 Electrical specification

6.10 I²S — Inter-IC sound interface

The I²S (Inter-IC Sound) module, supports the original two-channel I²S format, and left or right-aligned formats. It implements EasyDMA for sample transfer directly to and from RAM without CPU intervention.

The I²S peripheral has the following main features:

- Master and Slave mode
- Simultaneous bi-directional (TX and RX) audio streaming
- Original I²S and left- or right-aligned format
- 8, 16 and 24-bit sample width
- Low-jitter Master Clock generator
- Various sample rates

Figure 44: I²S master

6.10.1 Mode

The I²S protocol specification defines two modes of operation, Master and Slave.

The I²S mode decides which of the two sides (Master or Slave) shall provide the clock signals LRCK and SCK, and these signals are always supplied by the Master to the Slave.

6.10.2 Transmitting and receiving

The I²S module supports both transmission (TX) and reception (RX) of serial data. In both cases the serial data is shifted synchronously to the clock signals SCK and LRCK.

TX data is written to the SDOUT pin on the falling edge of SCK, and RX data is read from the SDIN pin on the rising edge of SCK. The most significant bit (MSB) is always transmitted first.

TX and RX are available in both Master and Slave modes and can be enabled/disabled independently in the [CONFIG.TXEN](#) on page 166 and [CONFIG.RXEN](#) on page 166.

Transmission and/or reception is started by triggering the START task. When started and transmission is enabled (in [CONFIG.TXEN](#) on page 166), the TXPTRUPD event will be generated for every [RXTXD.MAXCNT](#) on page 169 number of transmitted data words (containing one or more samples). Similarly, when started and reception is enabled (in [CONFIG.RXEN](#) on page 166), the RXPTRUPD event will be generated for every [RXTXD.MAXCNT](#) on page 169 received data words.

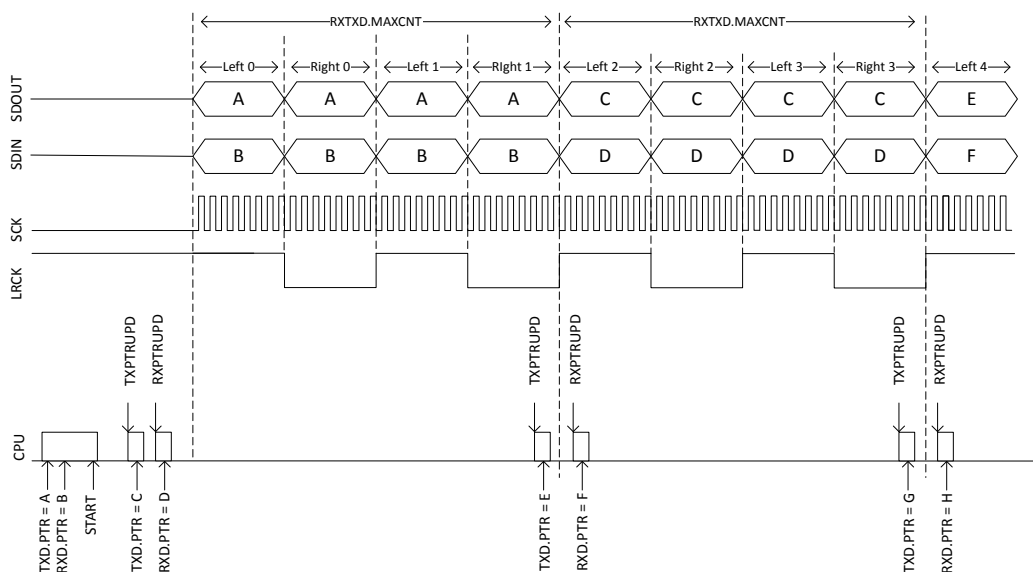


Figure 45: Transmitting and receiving. *CONFIG.FORMAT = Aligned, CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Stereo, RXTXD.MAXCNT = 1.*

6.10.3 Left right clock (LRCK)

The Left Right Clock (LRCK), often referred to as "word clock", "sample clock" or "word select" in I²S context, is the clock defining the frames in the serial bit streams sent and received on SDOUT and SDIN, respectively.

In I2S mode, each frame contains one left and right sample pair, with the left sample being transferred during the low half period of LRCK followed by the right sample being transferred during the high period of LRCK.

In Aligned mode, each frame contains one left and right sample pair, with the left sample being transferred during the high half period of LRCK followed by the right sample being transferred during the low period of LRCK.

Consequently, the LRCK frequency is equivalent to the audio sample rate.

When operating in Master mode, the LRCK is generated from the MCK, and the frequency of LRCK is then given as:

$$\text{LRCK} = \text{MCK} / \text{CONFIG.RATIO}$$

LRCK always toggles around the falling edge of the serial clock SCK.

6.10.4 Serial clock (SCK)

The serial clock (SCK), often referred to as the serial bit clock, pulses once for each data bit being transferred on the serial data lines SDIN and SDOUT.

When operating in Master mode the SCK is generated from the MCK, and the frequency of SCK is then given as:

$$\text{SCK} = 2 * \text{LRCK} * \text{CONFIG.SWIDTH}$$

The falling edge of the SCK falls on the toggling edge of LRCK.

When operating in Slave mode SCK is provided by the external I²S master.

6.10.5 Master clock (MCK)

The master clock (MCK) is the clock from which LRCK and SCK are derived when operating in Master mode.

The MCK is generated by an internal MCK generator. This generator always needs to be enabled when in Master mode, but the generator can also be enabled when in Slave mode. Enabling the generator when in slave mode can be useful in the case where the external Master is not able to generate its own master clock.

The MCK generator is enabled/disabled in the register [CONFIG.MCKEN](#) on page 166, and the generator is started or stopped by the START or STOP tasks.

In Master mode the LRCK and the SCK frequencies are closely related, as both are derived from MCK and set indirectly through [CONFIG.RATIO](#) on page 167 and [CONFIG.SWIDTH](#) on page 168.

When configuring these registers, the user is responsible for fulfilling the following requirements:

1. SCK frequency can never exceed the MCK frequency, which can be formulated as:

$$\text{CONFIG.RATIO} \geq 2 * \text{CONFIG.SWIDTH}$$

2. The MCK/LRCK ratio shall be a multiple of $2 * \text{CONFIG.SWIDTH}$, which can be formulated as:

$$\text{Integer} = (\text{CONFIG.RATIO} / (2 * \text{CONFIG.SWIDTH}))$$

The MCK signal can be routed to an output pin (specified in PSEL.MCK) to supply external I²S devices that require the MCK to be supplied from the outside.

When operating in Slave mode, the I²S module does not use the MCK and the MCK generator does not need to be enabled.

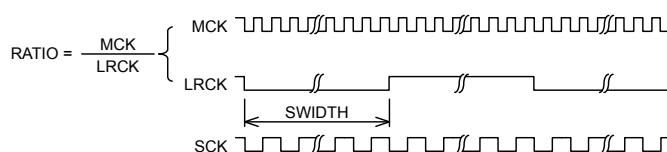


Figure 46: Relation between RATIO, MCK and LRCK.

Desired LRCK [Hz]	CONFIG.SWIDTH	CONFIG.RATIO	CONFIG.MCKF	MCK [Hz]	LRCK [Hz]	LRCK error [%]
16000	16Bit	32X	32MDIV63	507936.5	15873.0	-0.8
16000	16Bit	64X	32MDIV31	1032258.1	16129.0	0.8
16000	16Bit	256X	32MDIV8	4000000.0	15625.0	-2.3
32000	16Bit	32X	32MDIV31	1032258.1	32258.1	0.8
32000	16Bit	64X	32MDIV16	2000000.0	31250.0	-2.3
44100	16Bit	32X	32MDIV23	1391304.3	43478.3	-1.4
44100	16Bit	64X	32MDIV11	2909090.9	45454.5	3.1

Table 45: Configuration examples

6.10.6 Width, alignment and format

The CONFIG.SWIDTH register primarily defines the sample width of the data written to memory. In master mode, it then also sets the amount of bits per frame. In Slave mode it controls padding/trimming if required. Left, right, transmitted, and received samples always have the same width. The CONFIG.FORMAT

register specifies the position of the data frames with respect to the LRCK edges in both Master and Slave modes.

When using I^2S format, the first bit in a half-frame (containing one left or right sample) gets sampled on the second rising edge of the SCK after a LRCK edge. When using Aligned mode, the first bit in a half-frame gets sampled on the first rising edge of SCK following a LRCK edge.

For data being received on SDIN the sample value can be either right or left-aligned inside a half-frame, as specified in [CONFIG.ALIGN](#) on page 168. [CONFIG.ALIGN](#) on page 168 affects only the decoding of the incoming samples (SDIN), while the outgoing samples (SDOUT) are always left-aligned (or justified).

When using left-alignment, each half-frame starts with the MSB of the sample value (both for data being sent on SDOUT and received on SDIN).

When using right-alignment, each half-frame of data being received on SDIN ends with the LSB of the sample value, while each half-frame of data being sent on SDOUT starts with the MSB of the sample value (same as for left-alignment).

In Master mode, the size of a half-frame (in number of SCK periods) equals the sample width (in number of bits), and in this case the alignment setting does not care as each half-frame in any case will start with the MSB and end with the LSB of the sample value.

In slave mode, however, the sample width does not need to equal the frame size. This means you might have extra or fewer SCK pulses per half-frame than what the sample width specified in [CONFIG.SWIDTH](#) requires.

In the case where we use **left-alignment** and the number of SCK pulses per half-frame is **higher** than the sample width, the following will apply:

- For data received on SDIN, all bits after the LSB of the sample value will be discarded.
- For data sent on SDOUT, all bits after the LSB of the sample value will be 0.

In the case where we use **left-alignment** and the number of SCK pulses per frame is **lower** than the sample width, the following will apply:

- Data sent and received on SDOUT and SDIN will be truncated with the LSBs being removed first.

In the case where we use **right-alignment** and the number of SCK pulses per frame is **higher** than the sample width, the following will apply:

- For data received on SDIN, all bits before the MSB of the sample value will be discarded.
- For data sent on SDOUT, all bits after the LSB of the sample value will be 0 (same behavior as for left-alignment).

In the case where we use **right-alignment** and the number of SCK pulses per frame is **lower** than the sample width, the following will apply:

- Data received on SDIN will be sign-extended to "sample width" number of bits before being written to memory.
- Data sent on SDOUT will be truncated with the LSBs being removed first (same behavior as for left-alignment).

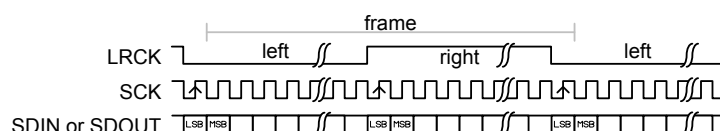


Figure 47: I^2S format. [CONFIG.SWIDTH](#) equalling half-frame size.

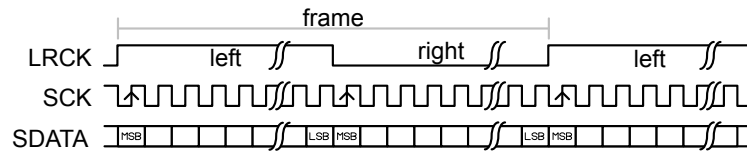


Figure 48: Aligned format. *CONFIG.SWIDTH* equalling half-frame size.

6.10.7 EasyDMA

The I²S module implements EasyDMA for accessing internal Data RAM without CPU intervention.

The source and destination pointers for the TX and RX data are configured in [TXD.PTR](#) on page 169 and [RXD.PTR](#) on page 169. The memory pointed to by these pointers will only be read or written when TX or RX are enabled in [CONFIG.TXEN](#) on page 166 and [CONFIG.RXEN](#) on page 166.

The addresses written to the pointer registers [TXD.PTR](#) on page 169 and [RXD.PTR](#) on page 169 are double-buffered in hardware, and these double buffers are updated for every [RXTXD.MAXCNT](#) on page 169 words (containing one or more samples) read/written from/to memory. The events TXPTRUPD and RXPTRUPD are generated whenever the TXD.PTR and RXD.PTR are transferred to these double buffers.

If [TXD.PTR](#) on page 169 is not pointing to the Data RAM region when transmission is enabled, or [RXD.PTR](#) on page 169 is not pointing to the Data RAM region when reception is enabled, an EasyDMA transfer may result in a HardFault and/or memory corruption. See [Memory](#) on page 19 for more information about the different memory regions.

Due to the nature of I²S, where the number of transmitted samples always equals the number of received samples (at least when both TX and RX are enabled), one common register [RXTXD.MAXCNT](#) on page 169 is used for specifying the sizes of these two memory buffers. The size of the buffers is specified in a number of 32-bit words. Such a 32-bit memory word can either contain four 8-bit samples, two 16-bit samples or one right-aligned 24-bit sample sign extended to 32 bit.

In stereo mode ([CONFIG.CHANNELS](#)=Stereo), the samples are stored as "left and right sample pairs" in memory. Figure [Memory mapping for 8 bit stereo. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Stereo.](#) on page 158, [Memory mapping for 16 bit stereo. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Stereo.](#) on page 158 and [Memory mapping for 24 bit stereo. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Stereo.](#) on page 159 show how the samples are mapped to memory in this mode. The mapping is valid for both RX and TX.

In mono mode ([CONFIG.CHANNELS](#)=Left or Right), RX sample from only one channel in the frame is stored in memory, the other channel sample is ignored. Illustrations [Memory mapping for 8 bit mono. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Left.](#) on page 158, [Memory mapping for 16 bit mono, left channel only. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Left.](#) on page 158 and [Memory mapping for 24 bit mono, left channel only. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Left.](#) on page 159 show how RX samples are mapped to memory in this mode.

For TX, the same outgoing sample read from memory is transmitted on both left and right in a frame, resulting in a mono output stream.

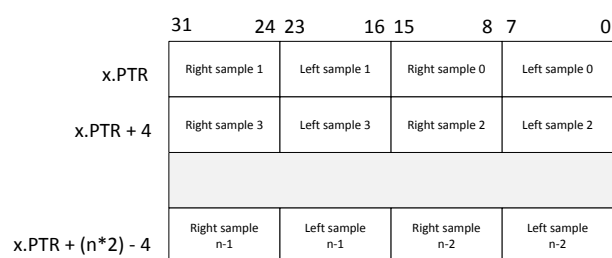


Figure 49: Memory mapping for 8 bit stereo. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Stereo.

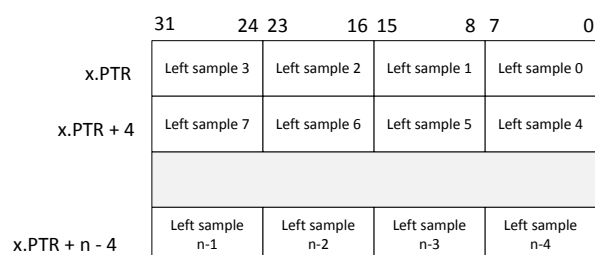


Figure 50: Memory mapping for 8 bit mono. CONFIG.SWIDTH = 8Bit, CONFIG.CHANNELS = Left.

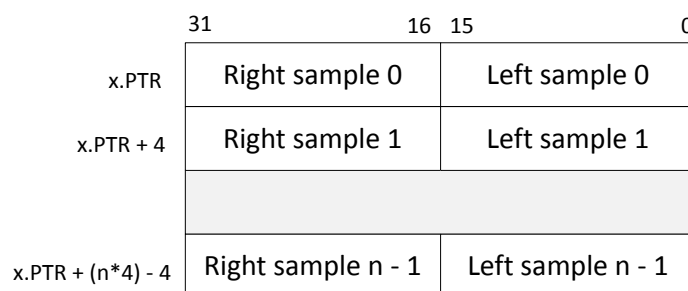


Figure 51: Memory mapping for 16 bit stereo. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Stereo.

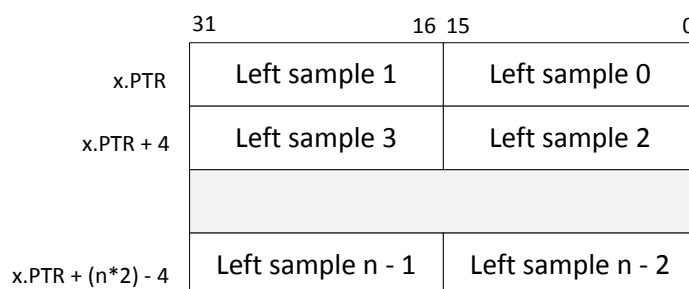


Figure 52: Memory mapping for 16 bit mono, left channel only. CONFIG.SWIDTH = 16Bit, CONFIG.CHANNELS = Left.

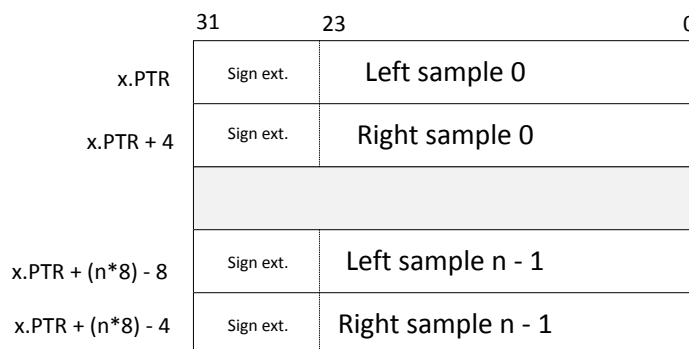


Figure 53: Memory mapping for 24 bit stereo. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Stereo.

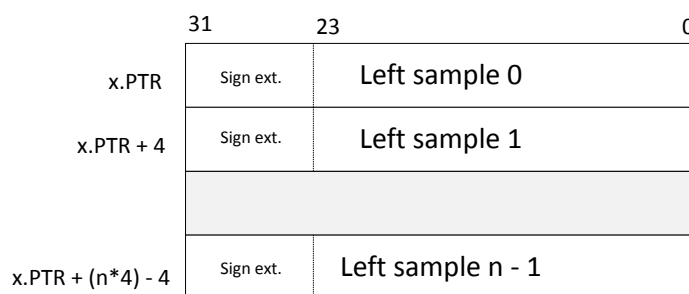


Figure 54: Memory mapping for 24 bit mono, left channel only. CONFIG.SWIDTH = 24Bit, CONFIG.CHANNELS = Left.

6.10.8 Module operation

Described here is a typical operating procedure for the I²S module.

1. Configure the I²S module using the CONFIG registers

```
// Enable reception
NRF_I2S->CONFIG.RXEN = (I2S_CONFIG_RXEN_RXEN_Enabled <<
                        I2S_CONFIG_RXEN_RXEN_Pos);

// Enable transmission
NRF_I2S->CONFIG.TXEN = (I2S_CONFIG_TXEN_TXEN_Enabled <<
                        I2S_CONFIG_TXEN_TXEN_Pos);

// Enable MCK generator
NRF_I2S->CONFIG.MCKEN = (I2S_CONFIG_MCKEN_MCKEN_Enabled <<
                        I2S_CONFIG_MCKEN_MCKEN_Pos);

// MCKFREQ = 4 MHz
NRF_I2S->CONFIG.MCKFREQ = I2S_CONFIG_MCKFREQ_MCKFREQ_32MDIV8 <<
                        I2S_CONFIG_MCKFREQ_MCKFREQ_Pos;

// Ratio = 256
NRF_I2S->CONFIG.RATIO = I2S_CONFIG_RATIO_RATIO_256X <<
                        I2S_CONFIG_RATIO_RATIO_Pos;

// MCKFREQ = 4 MHz and Ratio = 256 gives sample rate = 15.625 ks/s
// Sample width = 16 bit
NRF_I2S->CONFIG.SWIDTH = I2S_CONFIG_SWIDTH_SWIDTH_16Bit <<
                        I2S_CONFIG_SWIDTH_SWIDTH_Pos;

// Alignment = Left
NRF_I2S->CONFIG.ALIGN = I2S_CONFIG_ALIGN_ALIGN_Left <<
                        I2S_CONFIG_ALIGN_ALIGN_Pos;

// Format = I2S
NRF_I2S->CONFIG.FORMAT = I2S_CONFIG_FORMAT_FORMAT_I2S <<
                        I2S_CONFIG_FORMAT_FORMAT_Pos;

// Use stereo
NRF_I2S->CONFIG.CHANNELS = I2S_CONFIG_CHANNELS_CHANNELS_Stereo <<
                        I2S_CONFIG_CHANNELS_CHANNELS_Pos;
```

2. Map IO pins using the PINSEL registers

```
// MCK routed to pin 0
NRF_I2S->PSEL.MCK = (0 << I2S_PSEL_MCK_PIN_Pos) |
                    (I2S_PSEL_MCK_CONNECT_Connected <<
                     I2S_PSEL_MCK_CONNECT_Pos);

// SCK routed to pin 1
NRF_I2S->PSEL.SCK = (1 << I2S_PSEL_SCK_PIN_Pos) |
                    (I2S_PSEL_SCK_CONNECT_Connected <<
                     I2S_PSEL_SCK_CONNECT_Pos);

// LRCK routed to pin 2
NRF_I2S->PSEL.LRCK = (2 << I2S_PSEL_LRCK_PIN_Pos) |
                     (I2S_PSEL_LRCK_CONNECT_Connected <<
                      I2S_PSEL_LRCK_CONNECT_Pos);

// SDOUT routed to pin 3
NRF_I2S->PSEL.SDOUT = (3 << I2S_PSEL_SDOUT_PIN_Pos) |
                      (I2S_PSEL_SDOUT_CONNECT_Connected <<
                       I2S_PSEL_SDOUT_CONNECT_Pos);

// SDIN routed on pin 4
NRF_I2S->PSEL.SDIN = (4 << I2S_PSEL_SDIN_PIN_Pos) |
                     (I2S_PSEL_SDIN_CONNECT_Connected <<
                      I2S_PSEL_SDIN_CONNECT_Pos);
```


3. Configure TX and RX data pointers using the TXD, RXD and RXTXD registers

```
NRF_I2S->TXD.PTR = my_tx_buf;
NRF_I2S->RXD.PTR = my_rx_buf;
NRF_I2S->TXD.MAXCNT = MY_BUF_SIZE;
```

4. Enable the I²S module using the ENABLE register

```
NRF_I2S->ENABLE = 1;
```

5. Start audio streaming using the START task

```
NRF_I2S->TASKS_START = 1;
```

6. Handle received and transmitted data when receiving the TXPTRUPD and RXPTRUPD events

```
if (NRF_I2S->EVENTS_TXPTRUPD != 0)
{
    NRF_I2S->TXD.PTR = my_next_tx_buf;
    NRF_I2S->EVENTS_TXPTRUPD = 0;
}

if (NRF_I2S->EVENTS_RXPTRUPD != 0)
{
    NRF_I2S->RXD.PTR = my_next_rx_buf;
    NRF_I2S->EVENTS_RXPTRUPD = 0;
}
```

6.10.9 Pin configuration

The MCK, SCK, LRCK, SDIN and SDOUT signals associated with the I²S module are mapped to physical pins according to the pin numbers specified in the PSEL.x registers.

These pins are acquired whenever the I²S module is enabled through the register [ENABLE](#) on page 165.

When a pin is acquired by the I²S module, the direction of the pin (input or output) will be configured automatically, and any pin direction setting done in the GPIO module will be overridden. The directions for the various I²S pins are shown below in [GPIO configuration before enabling peripheral \(master mode\)](#) on page 161 and [GPIO configuration before enabling peripheral \(slave mode\)](#) on page 162.

To secure correct signal levels on the pins when the system is in OFF mode, and when the I²S module is disabled, these pins must be configured in the GPIO peripheral directly.

I ² S signal	I ² S pin	Direction	Output value	Comment
MCK	As specified in PSEL.MCK	Output	0	
LRCK	As specified in PSEL.LRCK	Output	0	
SCK	As specified in PSEL.SCK	Output	0	
SDIN	As specified in PSEL.SDIN	Input	Not applicable	
SDOUT	As specified in PSEL.SDOUT	Output	0	

Table 46: GPIO configuration before enabling peripheral (master mode)

I ² S signal	I ² S pin	Direction	Output value	Comment
MCK	As specified in PSEL.MCK	Output	0	
LRCK	As specified in PSEL.LRCK	Input	Not applicable	
SCK	As specified in PSEL.SCK	Input	Not applicable	
SDIN	As specified in PSEL.SDIN	Input	Not applicable	
SDOUT	As specified in PSEL.SDOUT	Output	0	

Table 47: GPIO configuration before enabling peripheral (slave mode)

6.10.10 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40025000	I2S	I2S	Inter-IC sound interface	

Table 48: Instances

Register	Offset	Description
TASKS_START	0x000	Starts continuous I2S transfer. Also starts MCK generator when this is enabled.
TASKS_STOP	0x004	Stops I2S transfer. Also stops MCK generator. Triggering this task will cause the STOPPED event to be generated.
EVENTS_RXPTRUPD	0x104	The RXD.PTR register has been copied to internal double-buffers. When the I2S module is started and RX is enabled, this event will be generated for every RXTXD.MAXCNT words that are received on the SDIN pin.
EVENTS_STOPPED	0x108	I2S transfer stopped.
EVENTS_TXPTRUPD	0x114	The TDX.PTR register has been copied to internal double-buffers. When the I2S module is started and TX is enabled, this event will be generated for every RXTXD.MAXCNT words that are sent on the SDOUT pin.
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ENABLE	0x500	Enable I2S module.
CONFIG.MODE	0x504	I2S mode.
CONFIG.RXEN	0x508	Reception (RX) enable.
CONFIG.TXEN	0x50C	Transmission (TX) enable.
CONFIG.MCKEN	0x510	Master clock generator enable.
CONFIG.MCKFREQ	0x514	Master clock generator frequency.
CONFIG.RATIO	0x518	MCK / LRCK ratio.
CONFIG.SWIDTH	0x51C	Sample width.
CONFIG.ALIGN	0x520	Alignment of sample within a frame.
CONFIG.FORMAT	0x524	Frame format.
CONFIG.CHANNELS	0x528	Enable channels.
RXD.PTR	0x538	Receive buffer RAM start address.
TXD.PTR	0x540	Transmit buffer RAM start address.
RXTXD.MAXCNT	0x550	Size of RXD and TXD buffers.
PSEL.MCK	0x560	Pin select for MCK signal.
PSEL.SCK	0x564	Pin select for SCK signal.
PSEL.LRCK	0x568	Pin select for LRCK signal.
PSEL.SDIN	0x56C	Pin select for SDIN signal.
PSEL.SDOUT	0x570	Pin select for SDOUT signal.

Table 49: Register overview

6.10.10.1 TASKS_START

Address offset: 0x000

Starts continuous I2S transfer. Also starts MCK generator when this is enabled.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID			A																																
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ID	Acce Field	Value ID	Value		Description																														
A	W	TASKS_START			Starts continuous I2S transfer. Also starts MCK generator when this is enabled.																														
		Trigger	1		Trigger task																														

6.10.10.2 TASKS_STOP

Address offset: 0x004

Stops I2S transfer. Also stops MCK generator. Triggering this task will cause the STOPPED event to be generated.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A																															
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																															
A	W	TASKS_STOP		Stops I2S transfer. Also stops MCK generator. Triggering this task will cause the STOPPED event to be generated.																															
		Trigger	1	Trigger task																															

6.10.10.3 EVENTS_RXPTRUPD

Address offset: 0x104

The RXD.PTR register has been copied to internal double-buffers. When the I2S module is started and RX is enabled, this event will be generated for every RXTXD.MAXCNT words that are received on the SDIN pin.

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID		A																															
Reset 0x00000000		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																													
A	RW	EVENTS_RXPTRUPD		The RXD.PTR register has been copied to internal double-buffers. When the I2S module is started and RX is enabled, this event will be generated for every RXTXD.MAXCNT words that are received on the SDIN pin.																													
		NotGenerated	0	Event not generated																													
		Generated	1	Event generated																													

6.10.10.4 EVENTS_STOPPED

Address offset: 0x108

I2S transfer stopped.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID				A																																
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																																
A	RW	EVENTS_STOPPED		I2S transfer stopped.																																
		NotGenerated	0	Event not generated																																
		Generated	1	Event generated																																

6.10.10.5 EVENTS_TXPTRUPD

Address offset: 0x114

The TDX.PTR register has been copied to internal double-buffers. When the I2S module is started and TX is enabled, this event will be generated for every RXTXD.MAXCNT words that are sent on the SDOUT pin.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID			A																																		
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																		
ID	Acce Field	Value ID	Value	Description																																	
A	RW	EVENTS_TXPTRUPD		The TDX.PTR register has been copied to internal double-buffers. When the I2S module is started and TX is enabled, this event will be generated for every RXTXD.MAXCNT words that are sent on the SDOUT pin.																																	
		NotGenerated	0	Event not generated																																	
		Generated	1	Event generated																																	

6.10.10.6 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				F C B																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																													
B	RW	RXPTRUPD				Enable or disable interrupt for event RXPTRUPD																													
			Disabled	0	Disable																														
			Enabled	1	Enable																														
C	RW	STOPPED				Enable or disable interrupt for event STOPPED																													
			Disabled	0	Disable																														
			Enabled	1	Enable																														
F	RW	TXPTRUPD				Enable or disable interrupt for event TXPTRUPD																													
			Disabled	0	Disable																														
			Enabled	1	Enable																														

6.10.10.7 INTENSET

Address offset: 0x304

Enable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																	
ID																															F		C		B	
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																	
ID	Acce Field	Value ID	Value	Description																																
B	RW RXPTRUPD			Write '1' to enable interrupt for event RXPTRUPD																																
		Set	1	Enable																																
		Disabled	0	Read: Disabled																																
		Enabled	1	Read: Enabled																																
C	RW STOPPED			Write '1' to enable interrupt for event STOPPED																																
		Set	1	Enable																																
		Disabled	0	Read: Disabled																																
		Enabled	1	Read: Enabled																																
F	RW TXPTRUPD			Write '1' to enable interrupt for event TXPTRUPD																																
		Set	1	Enable																																
		Disabled	0	Read: Disabled																																
		Enabled	1	Read: Enabled																																

6.10.10.8 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																	
ID																															F		C		B	
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																	
ID	Acce Field	Value ID	Value	Description																																
B	RW RXPTRUPD			Write '1' to disable interrupt for event RXPTRUPD																																
		Clear	1	Disable																																
		Disabled	0	Read: Disabled																																
		Enabled	1	Read: Enabled																																
C	RW STOPPED			Write '1' to disable interrupt for event STOPPED																																
		Clear	1	Disable																																
		Disabled	0	Read: Disabled																																
		Enabled	1	Read: Enabled																																
F	RW TXPTRUPD			Write '1' to disable interrupt for event TXPTRUPD																																
		Clear	1	Disable																																
		Disabled	0	Read: Disabled																																
		Enabled	1	Read: Enabled																																

6.10.10.9 ENABLE

Address offset: 0x500

Enable I2S module.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field		Value ID	Value	Description																														
A	RW	ENABLE			Enable I2S module.																														
			Disabled	0	Disable																														
			Enabled	1	Enable																														

6.10.10.10 CONFIG.MODE

Address offset: 0x504

I2S mode.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW MODE			I2S mode.																														
		Master	0	Master mode. SCK and LRCK generated from internal master clcok (MCK) and output on pins defined by PSEL.xxx.																														
		Slave	1	Slave mode. SCK and LRCK generated by external master and received on pins defined by PSEL.xxx																														

6.10.10.11 CONFIG.RXEN

Address offset: 0x508

Reception (RX) enable.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW RXEN			Reception (RX) enable.																														
		Disabled	0	Reception disabled and now data will be written to the RXD.PTR address.																														
		Enabled	1	Reception enabled.																														

6.10.10.12 CONFIG.TXEN

Address offset: 0x50C

Transmission (TX) enable.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000001			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1																															
ID	Acce Field	Value ID	Value	Description																														
A	RW TXEN			Transmission (TX) enable.																														
		Disabled	0	Transmission disabled and now data will be read from the RXD.TXD address.																														
		Enabled	1	Transmission enabled.																														

6.10.10.13 CONFIG.MCKEN

Address offset: 0x510

Master clock generator enable.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
ID			A																																				
Reset 0x00000001			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1					
ID	Acce Field	Value ID	Value		Description																																		
A	RW MCKEN				Master clock generator enable.																																		
		Disabled	0		Master clock generator disabled and PSEL.MCK not connected(available as GPIO).																																		
		Enabled	1		Master clock generator running and MCK output on PSEL.MCK.																																		

6.10.10.14 CONFIG.MCKFREQ

Address offset: 0x514

Master clock generator frequency.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x20000000			0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value		Description																													
A	RW	MCKFREQ			Master clock generator frequency.																													
		32MDIV8	0x20000000		32 MHz / 8 = 4.0 MHz																													
		32MDIV10	0x18000000		32 MHz / 10 = 3.2 MHz																													
		32MDIV11	0x16000000		32 MHz / 11 = 2.9090909 MHz																													
		32MDIV15	0x11000000		32 MHz / 15 = 2.1333333 MHz																													
		32MDIV16	0x10000000		32 MHz / 16 = 2.0 MHz																													
		32MDIV21	0x0C000000		32 MHz / 21 = 1.5238095																													
		32MDIV23	0x0B000000		32 MHz / 23 = 1.3913043 MHz																													
		32MDIV30	0x08800000		32 MHz / 30 = 1.0666667 MHz																													
		32MDIV31	0x08400000		32 MHz / 31 = 1.0322581 MHz																													
		32MDIV32	0x08000000		32 MHz / 32 = 1.0 MHz																													
		32MDIV42	0x06000000		32 MHz / 42 = 0.7619048 MHz																													
		32MDIV63	0x04100000		32 MHz / 63 = 0.5079365 MHz																													
		32MDIV125	0x020C0000		32 MHz / 125 = 0.256 MHz																													

6.10.10.15 CONFIG.RATIO

Address offset: 0x518

MCK / LRCK ratio.

Bit number		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID																																A	A	A	A
Reset 0x00000006		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0		
ID	Acce Field	Value ID	Value	Description																															
A	RW RATIO			MCK / LRCK ratio.																															
		32X	0	LRCK = MCK / 32																															
		48X	1	LRCK = MCK / 48																															
		64X	2	LRCK = MCK / 64																															
		96X	3	LRCK = MCK / 96																															
		128X	4	LRCK = MCK / 128																															
		192X	5	LRCK = MCK / 192																															
		256X	6	LRCK = MCK / 256																															
		384X	7	LRCK = MCK / 384																															
		512X	8	LRCK = MCK / 512																															

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
ID																																						A	A
Reset 0x00000001			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1				
ID	Acce Field	Value ID	Value		Description																																		
A	RW	WIDTH			Sample width.																																		
		8Bit	0		8 bit.																																		
		16Bit	1		16 bit.																																		
		24Bit	2		24 bit.																																		

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	RW ALIGN			Alignment of sample within a frame.																															
		Left	0	Left-aligned.																															
		Right	1	Right-aligned.																															

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID				A																																	
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
ID	Acce Field		Value ID	Value		Description																															
A	RW		FORMAT			Frame format.																															
			I2S	0		Original I2S format.																															
			Aligned	1		Alternate (left- or right-aligned) format.																															

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																														
A	RW	CHANNELS			Enable channels.																														
			Stereo	0	Stereo.																														
			Left	1	Left only.																														
			Right	2	Right only.																														

6.10.10.20 RXD.PTR

Address offset: 0x538

Receive buffer RAM start address.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID		Value				Description																											
A	RW	PTR						Receive buffer Data RAM start address. When receiving, words containing samples will be written to this address. This address is a word aligned Data RAM address.																											

6.10.10.21 TXD.PTR

Address offset: 0x540

Transmit buffer RAM start address.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID		Value				Description																											
A	RW	PTR						Transmit buffer Data RAM start address. When transmitting, words containing samples will be fetched from this address. This address is a word aligned Data RAM address.																											

6.10.10.22 RXTXD.MAXCNT

Address offset: 0x550

Size of RXD and TXD buffers.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
ID																								A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
ID	Acce	Field	Value	ID	Value	Description																																					
A	RW	MAXCNT				Size of RXD and TXD buffers in number of 32 bit words.																																					

6.10.10.23 PSEL.MCK

Address offset: 0x560

Pin select for MCK signal.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			C																															
Reset 0xFFFFFFFF			1																															
ID	Acce Field	Value ID	Value				Description																											
A	RW PIN		[0..31]				Pin number																											
B	RW PORT		[0..1]				Port number																											
C	RW CONNECT						Connection																											
		Disconnected	1				Disconnect																											
		Connected	0				Connect																											

6.10.10.24 PSEL.SCK

Address offset: 0x564

Pin select for SCK signal.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																							
ID				C																												B	A	A	A	A	A																					
Reset 0xFFFFFFFF				1																												1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field		Value	ID	Value		Description																																																			
A	RW PIN				[0..31]		Pin number																																																			
B	RW PORT				[0..1]		Port number																																																			
C	RW CONNECT						Connection																																																			
			Disconnected		1		Disconnect																																																			
			Connected		0		Connect																																																			

6.10.10.25 PSEL.LRCK

Address offset: 0x568

Pin select for LRCK signal.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID			C																												B	A	A	A	A	A
Reset 0xFFFFFFF			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
ID	Acce Field	Value ID	Value				Description																													
A	RW PIN		[0..31]				Pin number																													
B	RW PORT		[0..1]				Port number																													
C	RW CONNECT						Connection																													
		Disconnected	1				Disconnect																													
		Connected	0				Connect																													

6.10.10.26 PSEL.SDIN

Address offset: 0x56C

Pin select for SDIN signal.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
ID			C																								B												A	A	A	A	A
Reset 0xFFFFFFFF			1																																								
ID	Acce Field	Value ID	Value				Description																																				
A	RW PIN		[0..31]				Pin number																																				
B	RW PORT		[0..1]				Port number																																				
C	RW CONNECT						Connection																																				
		Disconnected	1				Disconnect																																				
		Connected	0				Connect																																				

6.10.10.27 PSEL.SDOUT

Address offset: 0x570

Pin select for SDOUT signal.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
ID			C																										B										A	A	A	A	A
Reset 0xFFFFFFFF			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1								
ID	Acce Field	Value ID	Value				Description																																				
A	RW PIN		[0..31]				Pin number																																				
B	RW PORT		[0..1]				Port number																																				
C	RW CONNECT						Connection																																				
		Disconnected	1				Disconnect																																				
		Connected	0				Connect																																				

6.10.11 Electrical specification

6.10.11.1 I2S timing specification

Symbol	Description	Min.	Typ.	Max.	Units
t_{S_SDIN}	SDIN setup time before SCK rising	20			ns
t_{H_SDIN}	SDIN hold time after SCK rising	15			ns
t_{S_SDOUT}	SDOUT setup time after SCK falling	40			ns
t_{H_SDOUT}	SDOUT hold time before SCK falling	6			ns
t_{SCK_LRCK}	SCLK falling to LRCK edge	-5	0	5	ns
f_{MCK}	MCK frequency			4000	kHz
f_{LRCK}	LRCK frequency			48	kHz
f_{SCK}	SCK frequency			2000	kHz
DC_{CK}	Clock duty cycle (MCK, LRCK, SCK)	45		55	%

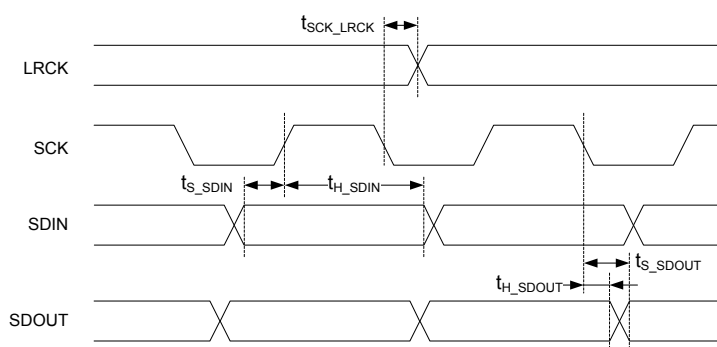


Figure 55: I2S timing diagram

6.11 LPCOMP — Low-power comparator

Low-power comparator (LPCOMP) compares an input voltage against a reference voltage.

Listed here are the main features of LPCOMP:

- 0 - VDD input range
- Ultra-low power
- Eight input options (**AIN0** to **AIN7**)
- Reference voltage options:
 - Two external analog reference inputs, or
 - 15-level internal reference ladder ($VDD/16$)
- Optional hysteresis enable on input
- Can be used as a wakeup source from System OFF mode

In System ON, the LPCOMP can generate separate events on rising and falling edges of a signal, or sample the current state of the pin as being above or below the selected reference. The block can be configured to use any of the analog inputs on the device. Additionally, the low-power comparator can be used as an analog wakeup source from System OFF or System ON. The comparator threshold can be programmed to a range of fractions of the supply voltage.

Note: LPCOMP cannot be used (STARTed) at the same time as COMP. Only one comparator can be used at a time.

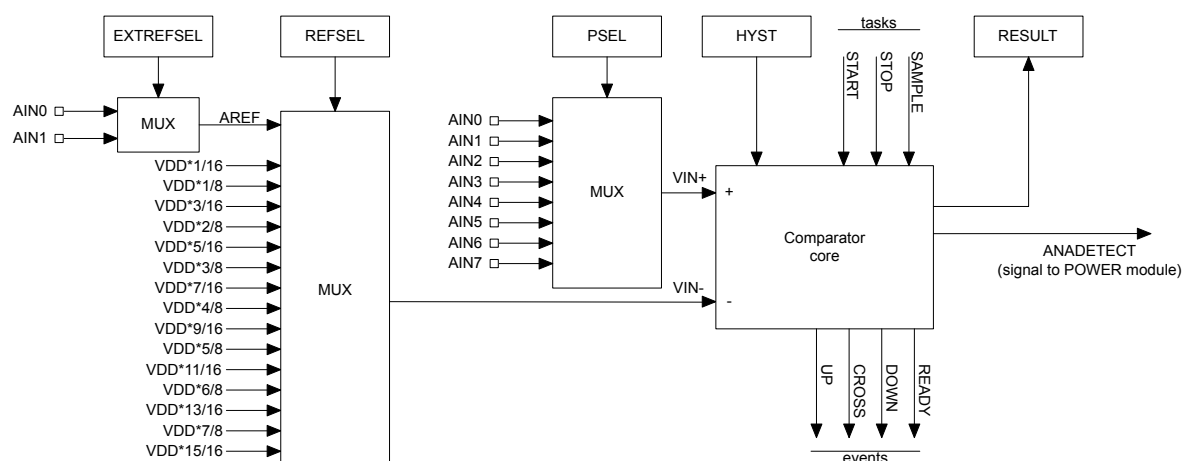


Figure 56: Low-power comparator

The wakeup comparator (LPCOMP) compares an input voltage ($VIN+$), which comes from an analog input pin selected via the **PSEL** register, against a reference voltage ($VIN-$) selected via registers **REFSEL** on page 178 and **EXTREFSEL**.

The **PSEL**, **REFSEL**, and **EXTREFSEL** registers must be configured before the LPCOMP is enabled through the **ENABLE** register.

The **HYST** register allows enabling an optional hysteresis in the comparator core. This hysteresis shall prevent noise on the signal to create unwanted events. Figure below illustrates the effect of an active hysteresis on a noisy input signal. It is disabled by default, and shall be configured before enabling LPCOMP as well.

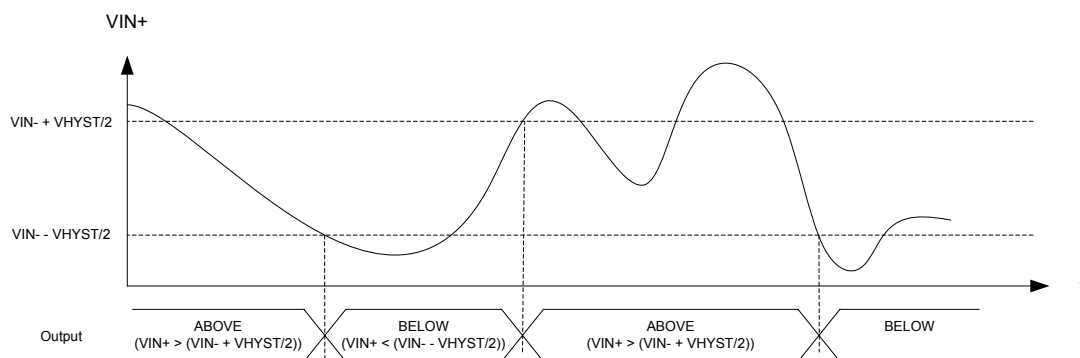


Figure 57: Effect of hysteresis on a noisy input signal

The LPCOMP is started by triggering the START task. After a startup time of $t_{\text{LPCOMP,STARTUP}}$, the LPCOMP will generate a READY event to indicate that the comparator is ready to use and the output of the LPCOMP is correct. The LPCOMP will generate events every time $V_{\text{IN}+}$ crosses $V_{\text{IN}-}$. More specifically, every time $V_{\text{IN}+}$ rises above $V_{\text{IN}-}$ (upward crossing) an UP event is generated along with a CROSS event. Every time $V_{\text{IN}+}$ falls below $V_{\text{IN}-}$ (downward crossing), a DOWN event is generated along with a CROSS event. When hysteresis is enabled, the upward crossing level becomes $(V_{\text{IN}-} + V_{\text{HYST}}/2)$, and the downward crossing level becomes $(V_{\text{IN}-} - V_{\text{HYST}}/2)$.

The LPCOMP is stopped by triggering the STOP task.

LPCOMP will be operational in both System ON and System OFF mode when it is enabled through the ENABLE register. See [POWER — Power supply](#) on page 58 for more information about power modes. Note that it is not allowed to go to System OFF when a READY event is pending to be generated.

All LPCOMP registers, including [ENABLE](#), are classified as retained registers when the LPCOMP is enabled. However, when the device wakes up from System OFF, all LPCOMP registers will be reset.

The LPCOMP can wake up the system from System OFF by asserting the ANADETECT signal. The ANADETECT signal can be derived from any of the event sources that generate the UP, DOWN and CROSS events. In case of wakeup from System OFF, no events will be generated, only the ANADETECT signal. See the ANADETECT register ([ANADETECT](#) on page 179) for more information on how to configure the ANADETECT signal.

The immediate value of the LPCOMP can be sampled to [RESULT](#) on page 177 by triggering the SAMPLE task.

See [RESETREAS](#) on page 73 for more information on how to detect a wakeup from LPCOMP.

6.11.1 Shared resources

The LPCOMP shares analog resources with SAADC. While it is possible to use the SAADC at the same time as the LPCOMP, selecting the same analog input pin for both modules is not supported.

Additionally, LPCOMP shares registers and other resources with other peripherals that have the same ID as the LPCOMP. See [Peripherals with shared ID](#) on page 97 for more information.

The LPCOMP peripheral shall not be disabled (by writing to the ENABLE register) before the peripheral has been stopped. Failing to do so may result in unpredictable behavior.

6.11.2 Pin configuration

You can use the LPCOMP.PSEL register to select one of the analog input pins, **AIN0** through **AIN7**, as the analog input pin for the LPCOMP.

See [GPIO — General purpose input/output](#) on page 138 for more information about the pins. Similarly, you can use [EXTRESEL](#) on page 179 to select one of the analog reference input pins, **AIN0** and **AIN1**,

as input for AREF in case AREF is selected in [EXTREFSEL](#) on page 179. The selected analog pins will be acquired by the LPCOMP when it is enabled through [ENABLE](#) on page 178.

6.11.3 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40013000	LPCOMP	LPCOMP	Low power comparator	

Table 50: Instances

Register	Offset	Description
TASKS_START	0x000	Start comparator
TASKS_STOP	0x004	Stop comparator
TASKS_SAMPLE	0x008	Sample comparator value
EVENTS_READY	0x100	LPCOMP is ready and output is valid
EVENTS_DOWN	0x104	Downward crossing
EVENTS_UP	0x108	Upward crossing
EVENTS_CROSS	0x10C	Downward or upward crossing
SHORTS	0x200	Shortcuts between local events and tasks
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
RESULT	0x400	Compare result
ENABLE	0x500	Enable LPCOMP
PSEL	0x504	Input pin select
REFSEL	0x508	Reference select
EXTREFSEL	0x50C	External reference select
ANADETECT	0x520	Analog detect configuration
HYST	0x538	Comparator hysteresis enable

Table 51: Register overview

6.11.3.1 TASKS_START

Address offset: 0x000

Start comparator

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field		Value ID	Value		Description																													
A	W	TASKS_START				Start comparator																													
		Trigger		1		Trigger task																													

6.11.3.2 TASKS_STOP

Address offset: 0x004

Stop comparator

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID																																								A				
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value		Description																																					
A	W	TASKS_STOP		Trigger		1		Stop comparator																																				
								Trigger task																																				

6.11.3.3 TASKS_SAMPLE

Address offset: 0x008

Sample comparator value

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID																																								A			
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field			Value ID			Value			Description																																	
A	W	TASKS_SAMPLE			Trigger			1			Sample comparator value																																
										Trigger task																																	

6.11.3.4 EVENTS_READY

Address offset: 0x100

LPCOMP is ready and output is valid

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	RW	EVENTS_READY		LPCOMP is ready and output is valid																															
		NotGenerated	0	Event not generated																															
		Generated	1	Event generated																															

6.11.3.5 EVENTS_DOWN

Address offset: 0x104

Downward crossing

Bit number										31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID										A																															
Reset 0x00000000										0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field		Value ID		Value		Description																																		
A	RW		EVENTS_DOWN				Downward crossing																																		
			NotGenerated		0		Event not generated																																		
			Generated		1		Event generated																																		

6.11.3.6 EVENTS_UP

Address offset: 0x108

Upward crossing

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	RW	EVENTS_UP		Upward crossing																															
		NotGenerated	0	Event not generated																															
		Generated	1	Event generated																															

6.11.3.7 EVENTS_CROSS

Address offset: 0x10C

Downward or upward crossing

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	EVENTS_CROSS		Downward or upward crossing																														
		NotGenerated	0	Event not generated																														
		Generated	1	Event generated																														

6.11.3.8 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
ID																																				E	D	C	B	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	Acce	Field	Value	ID	Value	Description																																		
A	RW	READY_SAMPLE				Shortcut between event READY and task SAMPLE																																		
			Disabled	0	Disable shortcut																																			
			Enabled	1	Enable shortcut																																			
B	RW	READY_STOP				Shortcut between event READY and task STOP																																		
			Disabled	0	Disable shortcut																																			
			Enabled	1	Enable shortcut																																			
C	RW	DOWN_STOP				Shortcut between event DOWN and task STOP																																		
			Disabled	0	Disable shortcut																																			
			Enabled	1	Enable shortcut																																			
D	RW	UP_STOP				Shortcut between event UP and task STOP																																		
			Disabled	0	Disable shortcut																																			
			Enabled	1	Enable shortcut																																			
E	RW	CROSS_STOP				Shortcut between event CROSS and task STOP																																		
			Disabled	0	Disable shortcut																																			
			Enabled	1	Enable shortcut																																			

6.11.3.9 INTENSET

Address offset: 0x304

Enable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW READY			Write '1' to enable interrupt for event READY																														
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
B	RW DOWN			Write '1' to enable interrupt for event DOWN																														
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
C	RW UP			Write '1' to enable interrupt for event UP																														
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
D	RW CROSS			Write '1' to enable interrupt for event CROSS																														
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														

6.11.3.10 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW READY			Write '1' to disable interrupt for event READY																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
B	RW DOWN			Write '1' to disable interrupt for event DOWN																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
C	RW UP			Write '1' to disable interrupt for event UP																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
D	RW CROSS			Write '1' to disable interrupt for event CROSS																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														

6.11.3.11 RESULT

Address offset: 0x400

Compare result

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	R	RESULT		Result of last compare. Decision point SAMPLE task.																														
		Below	0	Input voltage is below the reference threshold (VIN+ < VIN-)																														
		Above	1	Input voltage is above the reference threshold (VIN+ > VIN-)																														

6.11.3.12 ENABLE

Address offset: 0x500

Enable LPCOMP

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	ENABLE		Enable or disable LPCOMP																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														

6.11.3.13 PSEL

Address offset: 0x504

Input pin select

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A	RW	PSEL				Analog pin select																												
		AnalogInput0	0		AIN0 selected as analog input																													
		AnalogInput1	1		AIN1 selected as analog input																													
		AnalogInput2	2		AIN2 selected as analog input																													
		AnalogInput3	3		AIN3 selected as analog input																													
		AnalogInput4	4		AIN4 selected as analog input																													
		AnalogInput5	5		AIN5 selected as analog input																													
		AnalogInput6	6		AIN6 selected as analog input																													
		AnalogInput7	7		AIN7 selected as analog input																													

6.11.3.14 REFSEL

Address offset: 0x508

Reference select

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A A A A																															
Reset 0x00000004				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	RW	REFSEL		Reference select																															
		Ref1_8Vdd	0	VDD * 1/8 selected as reference																															
		Ref2_8Vdd	1	VDD * 2/8 selected as reference																															

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A A A A																															
Reset 0x00000004			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0																															
ID	Acce Field	Value ID	Value	Description																														
		Ref3_8Vdd	2	VDD * 3/8 selected as reference																														
		Ref4_8Vdd	3	VDD * 4/8 selected as reference																														
		Ref5_8Vdd	4	VDD * 5/8 selected as reference																														
		Ref6_8Vdd	5	VDD * 6/8 selected as reference																														
		Ref7_8Vdd	6	VDD * 7/8 selected as reference																														
		ARef	7	External analog reference selected																														
		Ref1_16Vdd	8	VDD * 1/16 selected as reference																														
		Ref3_16Vdd	9	VDD * 3/16 selected as reference																														
		Ref5_16Vdd	10	VDD * 5/16 selected as reference																														
		Ref7_16Vdd	11	VDD * 7/16 selected as reference																														
		Ref9_16Vdd	12	VDD * 9/16 selected as reference																														
		Ref11_16Vdd	13	VDD * 11/16 selected as reference																														
		Ref13_16Vdd	14	VDD * 13/16 selected as reference																														
		Ref15_16Vdd	15	VDD * 15/16 selected as reference																														

6.11.3.15 EXTREFSEL

Address offset: 0x50C

External reference select

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A																															
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value				Description																											
A	RW	EXTREFSEL					External analog reference select																											
		AnalogReference0	0				Use AIN0 as external analog reference																											
		AnalogReference1	1				Use AIN1 as external analog reference																											

6.11.3.16 ANADETECT

Address offset: 0x520

Analog detect configuration

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID																																	A	A				
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	Acce Field	Value ID	Value			Description																																
A	RW	ANADETECT				Analog detect configuration																																
		Cross	0			Generate ANADETECT on crossing, both upward crossing and downward crossing																																
		Up	1			Generate ANADETECT on upward crossing only																																
		Down	2			Generate ANADETECT on downward crossing only																																

6.11.3.17 HYST

Address offset: 0x538

Comparator hysteresis enable

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	HYST		Comparator hysteresis enable																														
		Disabled	0	Comparator hysteresis disabled																														
		Enabled	1	Comparator hysteresis enabled																														

6.11.4 Electrical specification

6.11.4.1 LPCOMP Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t _{LPCANADET}	Time from VIN crossing (>=50 mV above threshold) to ANADETECT signal generated		5		µs
V _{INPOFFSET}	Input offset including reference ladder error	-40		40	mV
V _{HYST}	Optional hysteresis		35		mV
t _{STARTUP}	Startup time for LPCOMP		140		µs

6.12 MWU — Memory watch unit

The Memory watch unit (MWU) can be used to generate events when a memory region is accessed by the CPU. The MWU can be configured to trigger events for access to Data RAM and Peripheral memory segments. The MWU allows an application developer to generate memory access events during development for debugging or during production execution for failure detection and recovery.

Listed here are the main features for MWU:

- Six memory regions, four user-configurable and two fixed regions in peripheral address space
- Flexible configuration of regions with START and END addresses
- Generate events on CPU read and/or write to a defined region of Data RAM or peripheral memory address space
- Programmable maskable or non-maskable (NMI) interrupt on events
- Peripheral interfaces can be watched for read and write access using subregions of the two fixed memory regions

Memory region	START address	END address
REGION[0..3]	Configurable	Configurable
PREGION[0]	0x40000000	0x4001FFFF
PREGION[1]	0x40020000	0x4003FFFF

Table 52: Memory regions

Each MWU region is defined by a start address and an end address, configured by the START and END registers respectively. These addresses are byte aligned and inclusive. The END register value has to be greater or equal to the START register value. Each region is associated with a pair of events that indicate that either a write access or a read access from the CPU has been detected inside the region.

For regions containing subregions (see below), a set of status registers PERREGION[0..1].SUBSTATWA and PERREGION[0..1].SUBSTATRA indicate which subregion(s) caused the EVENT_PREGION[0..1].WA and EVENT_PREGION[0..1].RA respectively.

The MWU is only able to detect memory accesses in the Data RAM and Peripheral memory segments from the CPU, see [Memory](#) on page 19 for more information about the different memory segments. EasyDMA

accesses are not monitored by the MWU. The MWU requires two HCLK cycles to detect and generate the event.

The peripheral regions, PREGION[0..1], are divided into 32 equally sized subregions, SR[0..31]. All subregions are excluded in the main region by default, and any can be included by specifying them in the SUBS register. When a subregion is excluded from the main region, the memory watch mechanism will not trigger any events when that subregion is accessed.

Subregions in PREGION[0..1] cannot be individually configured for read or write access watch. Watch configuration is only possible for a region as a whole. The PRGNiRA and PRGNiWA (i=0..1) fields in the REGIONEN register control watching read and write access.

REGION[0..3] can be individually enabled for read and/or write access watching through their respective RGNiRA and RGNiWA (i=0..3) fields in the REGIONEN register.

REGIONENSET and REGIONENCLR allow respectively enabling and disabling one or multiple REGIONS or PREGIONS watching in a single write access.

6.12.1 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40020000	MWU	MWU	Memory watch unit	

Table 53: Instances

Register	Offset	Description
EVENTS_REGION[0].WA	0x100	Write access to region 0 detected
EVENTS_REGION[0].RA	0x104	Read access to region 0 detected
EVENTS_REGION[1].WA	0x108	Write access to region 1 detected
EVENTS_REGION[1].RA	0x10C	Read access to region 1 detected
EVENTS_REGION[2].WA	0x110	Write access to region 2 detected
EVENTS_REGION[2].RA	0x114	Read access to region 2 detected
EVENTS_REGION[3].WA	0x118	Write access to region 3 detected
EVENTS_REGION[3].RA	0x11C	Read access to region 3 detected
EVENTS_PREGION[0].WA	0x160	Write access to peripheral region 0 detected
EVENTS_PREGION[0].RA	0x164	Read access to peripheral region 0 detected
EVENTS_PREGION[1].WA	0x168	Write access to peripheral region 1 detected
EVENTS_PREGION[1].RA	0x16C	Read access to peripheral region 1 detected
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
NMIEN	0x320	Enable or disable interrupt
NMIENSET	0x324	Enable interrupt
NMIENCLR	0x328	Disable interrupt
PERREGION[0].SUBSTATWA	0x400	Source of event/interrupt in region 0, write access detected while corresponding subregion was enabled for watching
PERREGION[0].SUBSTATRA	0x404	Source of event/interrupt in region 0, read access detected while corresponding subregion was enabled for watching
PERREGION[1].SUBSTATWA	0x408	Source of event/interrupt in region 1, write access detected while corresponding subregion was enabled for watching
PERREGION[1].SUBSTATRA	0x40C	Source of event/interrupt in region 1, read access detected while corresponding subregion was enabled for watching
REGIONEN	0x510	Enable/disable regions watch
REGIONENSET	0x514	Enable regions watch
REGIONENCLR	0x518	Disable regions watch

Register	Offset	Description
REGION[0].START	0x600	Start address for region 0
REGION[0].END	0x604	End address of region 0
REGION[1].START	0x610	Start address for region 1
REGION[1].END	0x614	End address of region 1
REGION[2].START	0x620	Start address for region 2
REGION[2].END	0x624	End address of region 2
REGION[3].START	0x630	Start address for region 3
REGION[3].END	0x634	End address of region 3
PREGION[0].START	0x6C0	Reserved for future use
PREGION[0].END	0x6C4	Reserved for future use
PREGION[0].SUBS	0x6C8	Subregions of region 0
PREGION[1].START	0x6D0	Reserved for future use
PREGION[1].END	0x6D4	Reserved for future use
PREGION[1].SUBS	0x6D8	Subregions of region 1

Table 54: Register overview

6.12.1.1 EVENTS_REGION[n].WA (n=0..3)

Address offset: $0x100 + (n \times 0x8)$

Write access to region n detected

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW WA			Write access to region n detected																														
		NotGenerated	0	Event not generated																														
		Generated	1	Event generated																														

6.12.1.2 EVENTS_REGION[n].RA (n=0..3)

Address offset: $0x104 + (n \times 0x8)$

Read access to region n detected

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW RA			Read access to region n detected																														
		NotGenerated	0	Event not generated																														
		Generated	1	Event generated																														

6.12.1.3 EVENTS_PREGION[n].WA (n=0..1)

Address offset: $0x160 + (n \times 0x8)$

Write access to peripheral region n detected

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																														
ID			A																														
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																														
ID	Acce Field	Value ID	Value	Description																													
A	RW WA			Write access to peripheral region n detected																													
		NotGenerated	0	Event not generated																													
		Generated	1	Event generated																													

6.12.1.4 EVENTS_PREGION[n].RA (n=0..1)

Address offset: 0x164 + (n × 0x8)

Read access to peripheral region n detected

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																
ID				A																																
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																
ID	Acce Field		Value ID	Value	Description																															
A	RW	RA			Read access to peripheral region n detected																															
			NotGenerated	0	Event not generated																															
			Generated	1	Event generated																															

6.12.1.5 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID				L										K	J	I	H										G	F	E	D	C	B	A			
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce	Field	Value	ID	Value		Description																													
A	RW	REGION0WA					Enable or disable interrupt for event REGION0WA																													
			Disabled	0	Disable																															
			Enabled	1	Enable																															
B	RW	REGION0RA					Enable or disable interrupt for event REGION0RA																													
			Disabled	0	Disable																															
			Enabled	1	Enable																															
C	RW	REGION1WA					Enable or disable interrupt for event REGION1WA																													
			Disabled	0	Disable																															
			Enabled	1	Enable																															
D	RW	REGION1RA					Enable or disable interrupt for event REGION1RA																													
			Disabled	0	Disable																															
			Enabled	1	Enable																															
E	RW	REGION2WA					Enable or disable interrupt for event REGION2WA																													
			Disabled	0	Disable																															
			Enabled	1	Enable																															
F	RW	REGION2RA					Enable or disable interrupt for event REGION2RA																													
			Disabled	0	Disable																															
			Enabled	1	Enable																															
G	RW	REGION3WA					Enable or disable interrupt for event REGION3WA																													
			Disabled	0	Disable																															
			Enabled	1	Enable																															
H	RW	REGION3RA					Enable or disable interrupt for event REGION3RA																													
			Disabled	0	Disable																															

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			L K J I																H G F E D C B A															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
I	RW PREGION0WA	Enabled	1	Enable																														
		Disabled	0	Enable or disable interrupt for event PREGION0WA																														
		Enabled	1	Disable																														
J	RW PREGION0RA	Enabled	1	Enable																														
		Disabled	0	Enable or disable interrupt for event PREGION0RA																														
		Enabled	1	Disable																														
K	RW PREGION1WA	Enabled	1	Enable																														
		Disabled	0	Enable or disable interrupt for event PREGION1WA																														
		Enabled	1	Disable																														
L	RW PREGION1RA	Enabled	1	Enable																														
		Disabled	0	Enable or disable interrupt for event PREGION1RA																														
		Enabled	1	Disable																														

6.12.1.6 INTENSET

Address offset: 0x304

Enable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			L K J I																H G F E D C B A															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																													
A	RW	REGION0WA			Write '1' to enable interrupt for event REGION0WA																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
B	RW	REGION0RA			Write '1' to enable interrupt for event REGION0RA																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
C	RW	REGION1WA			Write '1' to enable interrupt for event REGION1WA																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
D	RW	REGION1RA			Write '1' to enable interrupt for event REGION1RA																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
E	RW	REGION2WA			Write '1' to enable interrupt for event REGION2WA																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
F	RW	REGION2RA			Write '1' to enable interrupt for event REGION2RA																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
G	RW	REGION3WA			Write '1' to enable interrupt for event REGION3WA																													
			Set	1	Enable																													

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			L K J I																H G F E D C B A															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																													
H	RW	REGION3RA	Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
			Set	1	Write '1' to enable interrupt for event REGION3RA Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
I	RW	PREGION0WA			Write '1' to enable interrupt for event PREGION0WA																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
J	RW	PREGION0RA			Write '1' to enable interrupt for event PREGION0RA																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
K	RW	PREGION1WA			Write '1' to enable interrupt for event PREGION1WA																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
L	RW	PREGION1RA			Write '1' to enable interrupt for event PREGION1RA																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													

6.12.1.7 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			L K J I																H G F E D C B A															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																													
A	RW	REGION0WA			Write '1' to disable interrupt for event REGION0WA																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
B	RW	REGION0RA			Write '1' to disable interrupt for event REGION0RA																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
C	RW	REGION1WA			Write '1' to disable interrupt for event REGION1WA																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
D	RW	REGION1RA			Write '1' to disable interrupt for event REGION1RA																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
E	RW	REGION2WA			Write '1' to disable interrupt for event REGION2WA																													

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																						
ID			L K J I														H G F E D C B A																								
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																						
ID	Acce	Field	Value	ID	Value	Description																																			
			Clear	1		Disable																																			
			Disabled	0		Read: Disabled																																			
			Enabled	1		Read: Enabled																																			
	F	RW	REGION2RA			Write '1' to disable interrupt for event REGION2RA																																			
			Clear	1		Disable																																			
			Disabled	0		Read: Disabled																																			
			Enabled	1		Read: Enabled																																			
	G	RW	REGION3WA			Write '1' to disable interrupt for event REGION3WA																																			
			Clear	1		Disable																																			
			Disabled	0		Read: Disabled																																			
			Enabled	1		Read: Enabled																																			
	H	RW	REGION3RA			Write '1' to disable interrupt for event REGION3RA																																			
			Clear	1		Disable																																			
			Disabled	0		Read: Disabled																																			
			Enabled	1		Read: Enabled																																			
	I	RW	PREGION0WA			Write '1' to disable interrupt for event PREGION0WA																																			
			Clear	1		Disable																																			
			Disabled	0		Read: Disabled																																			
			Enabled	1		Read: Enabled																																			
	J	RW	PREGION0RA			Write '1' to disable interrupt for event PREGION0RA																																			
			Clear	1		Disable																																			
			Disabled	0		Read: Disabled																																			
			Enabled	1		Read: Enabled																																			
	K	RW	PREGION1WA			Write '1' to disable interrupt for event PREGION1WA																																			
			Clear	1		Disable																																			
			Disabled	0		Read: Disabled																																			
			Enabled	1		Read: Enabled																																			
	L	RW	PREGION1RA			Write '1' to disable interrupt for event PREGION1RA																																			
			Clear	1		Disable																																			
			Disabled	0		Read: Disabled																																			
			Enabled	1		Read: Enabled																																			

6.12.1.8 NMIIEN

Address offset: 0x320

Enable or disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			L K J I																H G F E D C B A															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW REGION0WA			Enable or disable interrupt for event REGION0WA																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
B	RW REGION0RA			Enable or disable interrupt for event REGION0RA																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
C	RW REGION1WA			Enable or disable interrupt for event REGION1WA																														
		Disabled	0	Disable																														

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			L K J I																H G F E D C B A															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value		ID	Value		Description																										
D	RW	REGION1RA	Enabled		1	Enable																												
			Disabled		0	Enable or disable interrupt for event REGION1RA																												
			Enabled		1	Disable																												
E	RW	REGION2WA	Enabled		1	Enable																												
			Disabled		0	Enable or disable interrupt for event REGION2WA																												
			Enabled		1	Disable																												
F	RW	REGION2RA	Enabled		1	Enable																												
			Disabled		0	Enable or disable interrupt for event REGION2RA																												
			Enabled		1	Disable																												
G	RW	REGION3WA	Enabled		1	Enable																												
			Disabled		0	Enable or disable interrupt for event REGION3WA																												
			Enabled		1	Disable																												
H	RW	REGION3RA	Enabled		1	Enable																												
			Disabled		0	Enable or disable interrupt for event REGION3RA																												
			Enabled		1	Disable																												
I	RW	PREGION0WA	Enabled		1	Enable																												
			Disabled		0	Enable or disable interrupt for event PREGION0WA																												
			Enabled		1	Disable																												
J	RW	PREGION0RA	Enabled		1	Enable																												
			Disabled		0	Enable or disable interrupt for event PREGION0RA																												
			Enabled		1	Disable																												
K	RW	PREGION1WA	Enabled		1	Enable																												
			Disabled		0	Enable or disable interrupt for event PREGION1WA																												
			Enabled		1	Disable																												
L	RW	PREGION1RA	Enabled		1	Enable																												
			Disabled		0	Enable or disable interrupt for event PREGION1RA																												
			Enabled		1	Disable																												

6.12.1.9 NMIENSET

Address offset: 0x324

Enable interrupt

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID			L K J I																H G F E D C B A																			
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	Acce Field	Value ID	Value		Description																																	
A	RW	REGION0WA		Write '1' to enable interrupt for event REGION0WA																																		
		Set	1	Enable																																		
		Disabled	0	Read: Disabled																																		
		Enabled	1	Read: Enabled																																		
B	RW	REGION0RA		Write '1' to enable interrupt for event REGION0RA																																		
		Set	1	Enable																																		
		Disabled	0	Read: Disabled																																		
		Enabled	1	Read: Enabled																																		
C	RW	REGION1WA		Write '1' to enable interrupt for event REGION1WA																																		
		Set	1	Enable																																		
		Disabled	0	Read: Disabled																																		

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			L K J I																H G F E D C B A															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
D	RW REGION1RA	Enabled	1	Read: Enabled																														
		Set	1	Write '1' to enable interrupt for event REGION1RA																														
		Disabled	0	Enable																														
		Enabled	1	Read: Disabled																														
E	RW REGION2WA	Enabled	1	Read: Enabled																														
		Set	1	Write '1' to enable interrupt for event REGION2WA																														
		Disabled	0	Enable																														
		Enabled	1	Read: Disabled																														
F	RW REGION2RA	Enabled	1	Read: Enabled																														
		Set	1	Write '1' to enable interrupt for event REGION2RA																														
		Disabled	0	Enable																														
		Enabled	1	Read: Disabled																														
G	RW REGION3WA	Enabled	1	Read: Enabled																														
		Set	1	Write '1' to enable interrupt for event REGION3WA																														
		Disabled	0	Enable																														
		Enabled	1	Read: Disabled																														
H	RW REGION3RA	Enabled	1	Read: Enabled																														
		Set	1	Write '1' to enable interrupt for event REGION3RA																														
		Disabled	0	Enable																														
		Enabled	1	Read: Disabled																														
I	RW PREGION0WA	Enabled	1	Read: Enabled																														
		Set	1	Write '1' to enable interrupt for event PREGION0WA																														
		Disabled	0	Enable																														
		Enabled	1	Read: Disabled																														
J	RW PREGION0RA	Enabled	1	Read: Enabled																														
		Set	1	Write '1' to enable interrupt for event PREGION0RA																														
		Disabled	0	Enable																														
		Enabled	1	Read: Disabled																														
K	RW PREGION1WA	Enabled	1	Read: Enabled																														
		Set	1	Write '1' to enable interrupt for event PREGION1WA																														
		Disabled	0	Enable																														
		Enabled	1	Read: Disabled																														
L	RW PREGION1RA	Enabled	1	Read: Enabled																														
		Set	1	Write '1' to enable interrupt for event PREGION1RA																														
		Disabled	0	Enable																														
		Enabled	1	Read: Disabled																														

6.12.1.10 NMIENCLR

Address offset: 0x328

Disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			L K J I																H G F E D C B A															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value																Description															
A	RW REGION0WA		Write '1' to disable interrupt for event REGION0WA																															
		Clear	1																Disable															

6.12.1.11 PERREGION[n].SUBSTATWA (n=0..1)

4452 021 v1.3

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			L K J I																H G F E D C B A															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A	RW	RGN0WA				Enable write access watch in region[0]																												
			Set	1	Enable write access watch in this region																													
			Disabled	0	Write access watch in this region is disabled																													
			Enabled	1	Write access watch in this region is enabled																													
B	RW	RGN0RA				Enable read access watch in region[0]																												
			Set	1	Enable read access watch in this region																													
			Disabled	0	Read access watch in this region is disabled																													
			Enabled	1	Read access watch in this region is enabled																													
C	RW	RGN1WA				Enable write access watch in region[1]																												
			Set	1	Enable write access watch in this region																													
			Disabled	0	Write access watch in this region is disabled																													
			Enabled	1	Write access watch in this region is enabled																													
D	RW	RGN1RA				Enable read access watch in region[1]																												
			Set	1	Enable read access watch in this region																													
			Disabled	0	Read access watch in this region is disabled																													
			Enabled	1	Read access watch in this region is enabled																													
E	RW	RGN2WA				Enable write access watch in region[2]																												
			Set	1	Enable write access watch in this region																													
			Disabled	0	Write access watch in this region is disabled																													
			Enabled	1	Write access watch in this region is enabled																													
F	RW	RGN2RA				Enable read access watch in region[2]																												

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			L K J I																H G F E D C B A															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value		ID	Value		Description																										
A	RW	RGN0WA					Disable write access watch in region[0]																											
			Clear		1	Disable write access watch in this region																												
			Disabled		0	Write access watch in this region is disabled																												
			Enabled		1	Write access watch in this region is enabled																												
B	RW	RGN0RA					Disable read access watch in region[0]																											
			Clear		1	Disable read access watch in this region																												
			Disabled		0	Read access watch in this region is disabled																												
			Enabled		1	Read access watch in this region is enabled																												
C	RW	RGN1WA					Disable write access watch in region[1]																											
			Clear		1	Disable write access watch in this region																												
			Disabled		0	Write access watch in this region is disabled																												
			Enabled		1	Write access watch in this region is enabled																												

Address offset: 0x518

Disable regions watch

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A																															

Address offset: $0x600 + (n \times 0x10)$

Start address for region n

6.12.1.17 REGION[n].END (n=0..3)

Address offset: $0x604 + (n \times 0x10)$

End address of region n

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID	Value				Description																											
A	RW END							End address of region.																											

6.12.1.18 PREGION[n].START (n=0..1)

Address offset: $0x6C0 + (n \times 0x10)$

Reserved for future use

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																					
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A																																					
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																					
ID	Acce	Field	Value	ID	Value																																		Description																																	
A	R	START			Reserved for future use																																																																			

6.12.1.19 PREGION[n].END (n=0..1)

Address offset: $0x6C4 + (n \times 0x10)$

Reserved for future use

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																					
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A																																					
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																					
ID	Acce	Field	Value	ID	Value																																		Description																																	
A	R	END			Reserved for future use																																																																			

6.12.1.20 PREGION[n].SUBS (n=0..1)

Address offset: $0x6C8 + (n \times 0x10)$

Subregions of region n

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce	Field	Value		ID	Value		Description																										
A-f	RW	SR[i] (i=0..31)				Include or exclude subregion i in region																												
			Exclude	0		Exclude																												
			Include	1		Include																												

6.13 NFCT — Near field communication tag

The NFCT peripheral is an implementation of an NFC Forum compliant listening device NFC-A.

With appropriate software, the NFCT peripheral can be used as the listening device NFC-A as specified by the [NFC Forum](#).

Listed here are the main features for the NFCT peripheral:

- NFC-A listen mode operation
 - 13.56 MHz input frequency
 - Bit rate 106 kbps
- Wake-on-field low power field detection (SENSE) mode
- Frame assemble and disassemble for the NFC-A frames specified by the NFC Forum
- Programmable frame timing controller
- Integrated automatic collision resolution, cyclic redundancy check (CRC), and parity functions

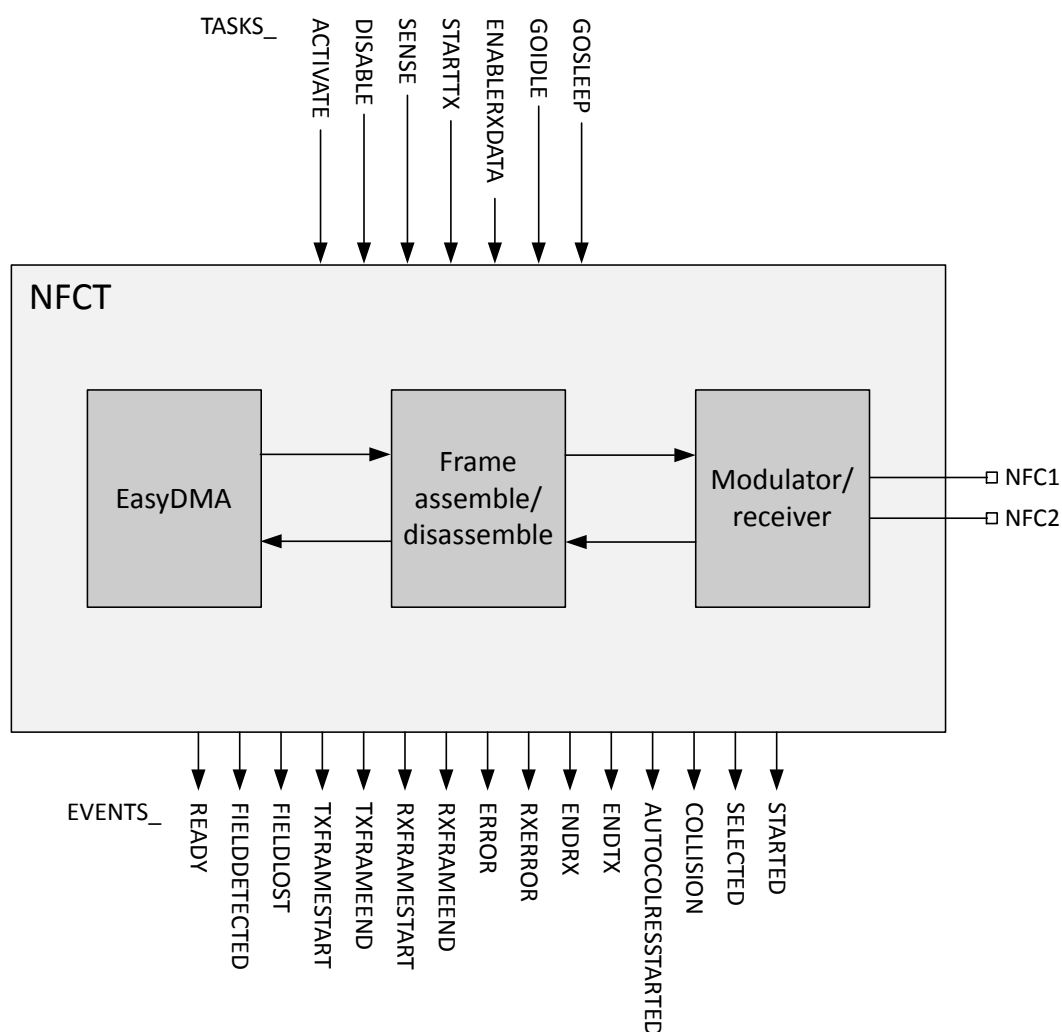


Figure 58: NFCT block diagram

6.13.1 Overview

The NFCT peripheral contains a 13.56 MHz AM receiver and a 13.56 MHz load modulator with 106 kbps data rate as defined by the NFC Forum.

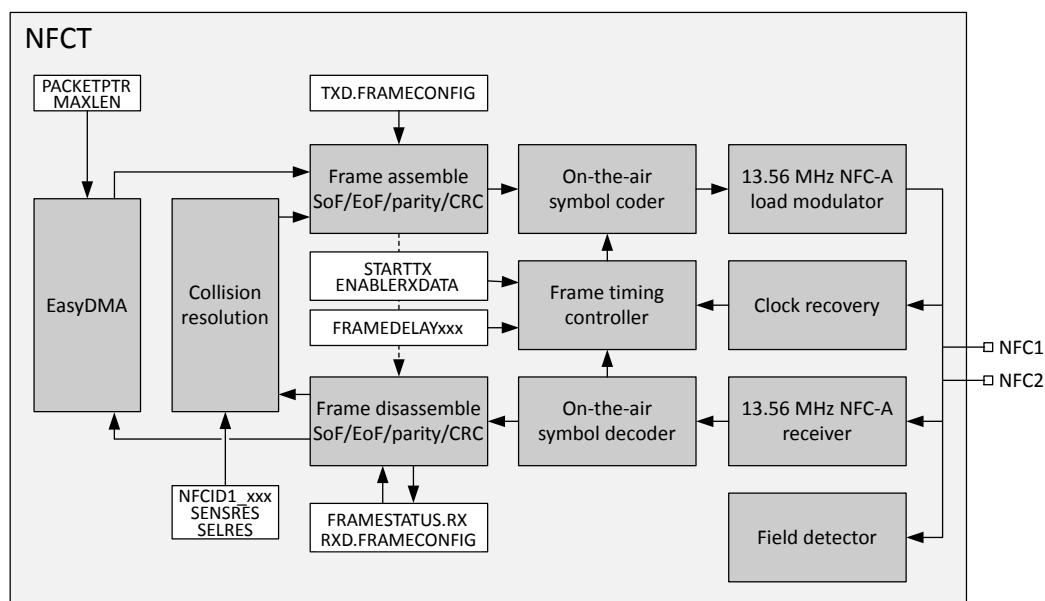


Figure 59: NFCT overview

When transmitting, the frame data will be transferred directly from RAM and transmitted with configurable frame type and delay timing. The system will be notified by an event whenever a complete frame is received or sent. The received frames will be automatically disassembled and the data part of the frame transferred to RAM.

The NFCT peripheral also supports the collision detection and resolution ("anticollision") as defined by the NFC Forum.

Wake-on-field is supported in SENSE mode while the device is either in System OFF or System ON mode. When the antenna enters an NFC field, an event will be triggered notifying the system to activate the NFCT functionality for incoming frames. In System ON, if the energy detected at the antenna increases beyond a threshold value, the module will generate a **FIELDDETECTED** event. When the strength of the field no longer supports NFC communication, the module will generate a **FIELDLOST** event. For the Low Power Field Detect threshold values, refer to **NFCT Electrical Specification** on page 224.

In System OFF, the NFCT Low Power Field Detect function can wake the system up through a reset. See **RESETREAS** on page 73 for more information on how to detect a wakeup from NFCT.

If the system is put into System OFF mode while a field is already present, the NFCT Low Power Field Detect function will wake the system up right away and generate a reset.

Important: As a consequence of a reset, NFCT is disabled, and therefore the reset handler will have to activate NFCT again and set it up properly.

The HFXO must be running before the NFCT peripheral goes into ACTIVATED state. Note that the NFCT peripheral calibration is automatically done on **ACTIVATE** task. The HFXO can be turned off when the NFCT peripheral goes into SENSE mode. The shortcut **FIELDDETECTED_ACTIVATE** can be used when the HFXO is already running while in SENSE mode.

Outgoing data will be collected from RAM with the EasyDMA function and assembled according to the **TXD.FRAMECONFIG** on page 219 register. Incoming data will be disassembled according to the **RXD.FRAMECONFIG** register and the data section in the frame will be written to RAM via the EasyDMA function.

The NFCT peripheral includes a frame timing controller that can be used to accurately control the inter-frame delay between the incoming frame and a corresponding outgoing frame. It also includes optional CRC functionality.

6.13.2 Operating states

Tasks and events are used to control the operating state of the peripheral. The module can change state by triggering a task, or when specific operations are finalized. Events and tasks allow software to keep track of and change the current state.

See [NFCT block diagram](#) on page 195 and [NFCT state diagram, automatic collision resolution enabled](#) on page 197 for more information. See *NFC Forum, NFC Activity Technical Specification* for description on NFCT operating states.

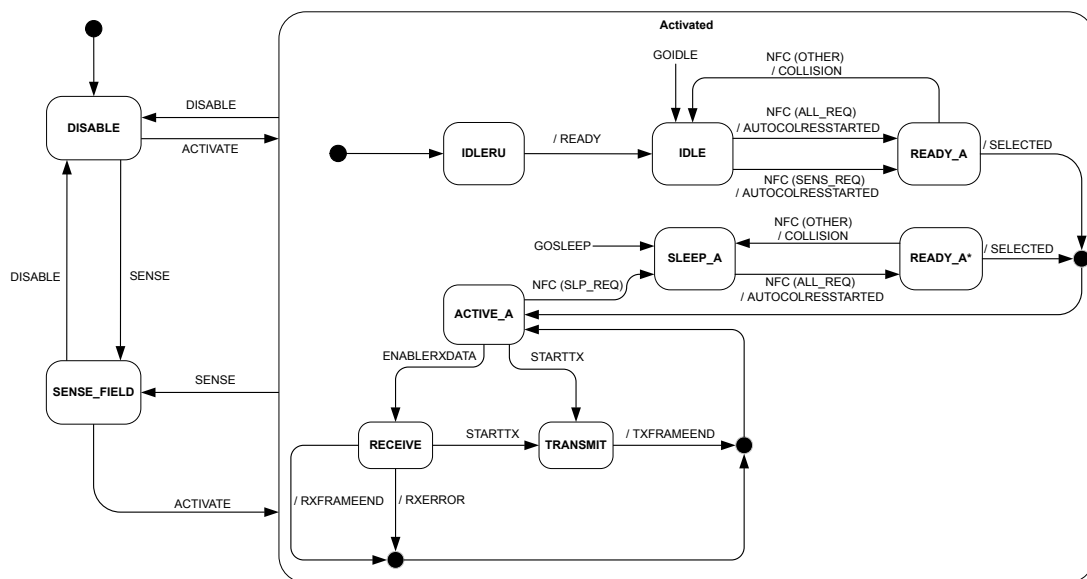


Figure 60: NFCT state diagram, automatic collision resolution enabled

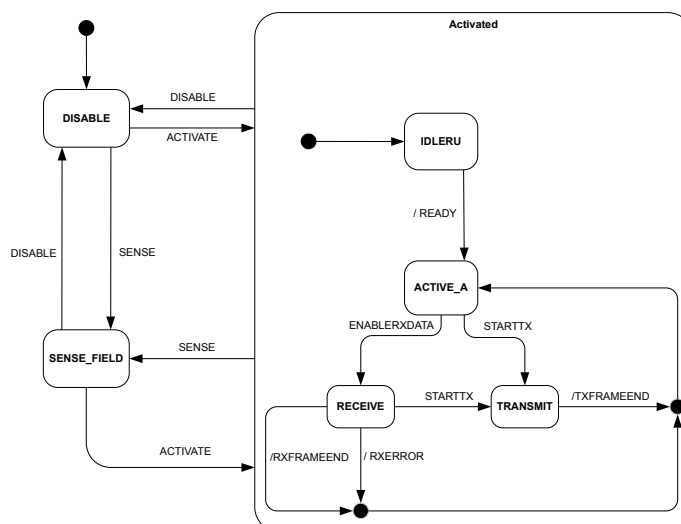


Figure 61: NFCT state diagram, automatic collision resolution disabled

Important:

- FIELDLOST event is not generated in SENSE mode.
- Sending SENSE task while field is still present does not generate **FIELDDETECTED** event.
- If the FIELDDETECTED event is cleared before sending the **ACTIVATE** task, then the FIELDDETECTED event shows up again after sending the ACTIVATE task. The shortcut FIELDDETECTED_ACTIVATE can be used to avoid this condition.

6.13.3 Pin configuration

NFCT uses two pins to connect the antenna and these pins are shared with GPIOs.

The PROTECT field in the NFCPINS register in UICR defines the usage of these pins and their protection level against excessive voltages. The content of the NFCPINS register is reloaded at every reset. See [Pin assignments](#) on page 557 for the pins used by the NFCT peripheral.

When NFCPINS.PROTECT=NFC, a protection circuit will be enabled on the dedicated pins, preventing the chip from being damaged in the presence of a strong NFC field. The protection circuit will short the two pins together if voltage difference exceeds approximately 2V. The GPIO function on those pins will also be disabled.

When NFCPINS.PROTECT=Disabled, the device will not be protected against strong NFC field damages caught by a connected NFCT antenna, and the NFCT peripheral will not operate as expected, as it will never leave the DISABLE state.

The pins dedicated to the NFCT antenna function will have some limitation when the pins are configured for normal GPIO operation. The pin capacitance will be higher on those (refer to C_{PAD_NFC} in the Electrical Specification of [GPIO — General purpose input/output](#) on page 138), and some increased leakage current between the two pins is to be expected if they are used in GPIO mode, and are driven to different logical values. To save power, the two pins should always be set to the same logical value whenever entering one of the device power saving modes. For details, refer to I_{NFC_LEAK} in the Electrical Specification of [GPIO — General purpose input/output](#) on page 138.

6.13.4 EasyDMA

The NFCT peripheral implements EasyDMA for reading and writing of data packets from and to the Data RAM.

The NFCT EasyDMA utilizes a pointer called [PACKETPTR](#) on page 219 for receiving and transmitting packets.

The NFCT peripheral uses EasyDMA to read or write RAM, but not both at the same time. The event [RXFRAMESTART](#) indicates that the EasyDMA has started writing to the RAM for a receive frame and the event [RXFRAMEEND](#) indicates that the EasyDMA has completed writing to the RAM. Similarly, the event [TXFRAMESTART](#) indicates that the EasyDMA has started reading from the RAM for a transmit frame and the event [TXFRAMEEND](#) indicates that the EasyDMA has completed reading from the RAM. If a transmit and a receive operation is issued at the same time, the transmit operation would be prioritized.

Starting a transmit operation while the EasyDMA is writing a receive frame to the RAM will result in unpredictable behavior. Starting an EasyDMA operation when there is an ongoing EasyDMA operation may result in unpredictable behavior. It is recommended to wait for the TXFRAMEEND or RXFRAMEEND event for the ongoing transmit or receive before starting a new receive or transmit operation.

The [MAXLEN](#) on page 219 register determines the maximum number of bytes that can be read from or written to the RAM. This feature can be used to ensure that the NFCT peripheral does not overwrite, or read beyond, the RAM assigned to a packet. Note that if the [RXD.AMOUNT](#) or [TXD.AMOUNT](#) register indicates longer data packets than set in MAXLEN, the frames sent to or received from the physical layer

will be incomplete. In that situation, in RX, the **OVERRUN** bit in the **FRAMESTATUS.RX** register will be set and an **RXERROR** event will be triggered.

Important: The **RXD.AMOUNT** and **TXD.AMOUNT** define a frame length in bytes and bits excluding start of frame (SoF), end of frame (EoF), and parity, but including CRC for **RXD.AMOUNT** only. Make sure to take potential additional bits into account when setting **MAXLEN**.

Only sending task **ENABLERXDATA** ensures that a new value in **PACKETPTR** pointing to the RX buffer in Data RAM is taken into account.

If **PACKETPTR** is not pointing to the Data RAM region, an EasyDMA transfer may result in a hard fault or RAM corruption. For more information about the different memory regions, see Chapter **Memory** on page 19.

The NFCT peripherals normally do alternative receive and transmit frames. Therefore, to prepare for the next frame, the **PACKETPTR**, **MAXLEN**, **TXD.FRAMECONFIG** and **TXD.AMOUNT** can be updated while the receive is in progress, and, similarly, the **PACKETPTR**, **MAXLEN** and **RXD.FRAMECONFIG** can be updated while the transmit is in progress. They can be updated and prepared for the next NFC frame immediately after the **STARTED** event of the current frame has been received. Updating the **TXD.FRAMECONFIG** and **TXD.AMOUNT** during the current transmit frame or updating **RXD.FRAMECONFIG** during current receive frame may cause unpredictable behaviour.

In accordance with *NFC Forum, NFC Digital Protocol Technical Specification*, the least significant bit (LSB) from the least significant byte (LSByte) is sent on air first. The bytes are stored in increasing order, starting at the lowest address in the EasyDMA buffer in RAM.

6.13.5 Frame assembler

The NFCT peripheral implements a frame assembler in hardware.

When the NFCT peripheral is in the **ACTIVE_A** state, the software can decide to enter RX or TX mode. For RX, see **Frame disassembler** on page 200. For TX, the software must indicate the address of the source buffer in Data RAM and its size through programming the **PACKETPTR** and **MAXLEN** registers respectively, then issuing a **STARTTX** task.

MAXLEN must be set so that it matches the size of the frame to be sent.

The **STARTED** event indicates that the **PACKETPTR** and **MAXLEN** registers have been captured by the frame assembler EasyDMA.

When asserting the **STARTTX** task, the frame assembler module will start reading **TXD.AMOUNT.TXDATABYTES** bytes (plus one additional byte if **TXD.AMOUNT.TXDATABITS > 0**) from the RAM position set by the **PACKETPTR**.

The NFCT peripheral transmits the data as read from RAM, adding framing and the CRC calculated on the fly if set in **TXD.FRAMECONFIG**. The NFCT peripheral will take $(8 * \text{TXD.AMOUNT.TXDATABYTES} + \text{TXD.AMOUNT.TXDATABITS})$ bits and assemble a frame according to the settings in **TXD.FRAMECONFIG**. Both short frames, standard frames, and bit-oriented SDD frames as specified in the *NFC Forum, NFC Digital Protocol Technical Specification* can be assembled by the correct setting of the **TXD.FRAMECONFIG** register.

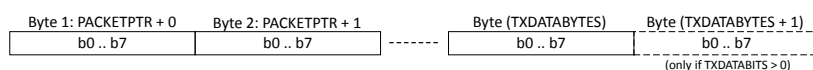
The bytes will be transmitted on air in the same order as they are read from RAM with a rising bit order within each byte, least significant bit (LSB) first. That is, b0 will be transmitted on air before b1, and so on. The bits read from RAM will be coded into symbols as defined in the *NFC Forum, NFC Digital Protocol Technical Specification*.

Important: Some NFC Forum documents, such as *NFC Forum*, *NFC Digital Protocol Technical Specification*, define bit numbering in a byte from b1 (LSB) to b8 (most significant bit (MSB)), while most other technical documents from the NFC Forum, and also the Nordic Semiconductor documentation, traditionally number them from b0 to b7. The present document uses the b0–b7 numbering scheme. Be aware of this when comparing the *NFC Forum*, *NFC Digital Protocol Technical Specification* to others.

The frame assembler can be configured in TXD.FRAMECONFIG to add SoF symbol, calculate and add parity bits, and calculate and add CRC to the data read from RAM when assembling the frame. The total frame will then be longer than what is defined by TXD.AMOUNT.TXDATABYTES. TXDATABITS. DISCARDMODE will select if the first bits in the first byte read from RAM or the last bits in the last byte read from RAM will be discarded if TXD.AMOUNT.TXDATABITS are not equal to zero. Note that if TXD.FRAMECONFIG.PARITY = Parity and TXD.FRAMECONFIG.DISCARDMODE=DiscardStart, a parity bit will be included after the non-complete first byte. No parity will be added after a non-complete last byte.

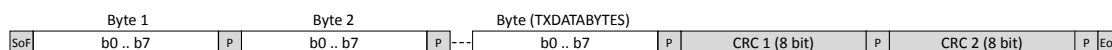
The frame assemble operation is illustrated in [Frame assemble illustration](#) on page 200 for different settings in TXD.FRAMECONFIG. All shaded bit fields are added by the frame assembler. Some of these bits are optional and appearances are configured in TXD.FRAMECONFIG. Note that the frames illustrated do not necessarily comply with the NFC specification. The figure is only to illustrate the behavior of the NFCT peripheral.

Data from RAM

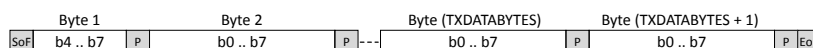


Frame on air

PARITY = Parity
TXDATABITS = 0
CRCMODETX = CRC16TX



PARITY = Parity
TXDATABITS = 4
CRCMODETX = NoCRCTX
DISCARDMODE = DiscardStart



PARITY = Parity
TXDATABITS = 0
CRCMODETX = NoCRCTX



Figure 62: Frame assemble illustration

The accurate timing for transmitting the frame on air is set using the frame timing controller settings.

6.13.6 Frame disassembler

The NFCT peripheral implements a frame disassembler in hardware.

When the NFCT peripheral is in the ACTIVE_A state, the software can decide to enter RX or TX mode. For TX, see [Frame assembler](#) on page 199. For RX, the software must indicate the address and size of the destination buffer in Data RAM through programming the [PACKETPTR](#) and [MAXLEN](#) registers before issuing an [ENABLERXDATA](#) task.

The [STARTED](#) event indicates that the [PACKETPTR](#) and [MAXLEN](#) registers have been captured by the frame disassembler EasyDMA.

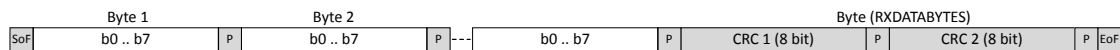
When an incoming frame starts, the [RXFRAMESTART](#) event will get issued and data will be written to the buffer in Data RAM. The frame disassembler will verify and remove any parity bits, start of frame (SoF) and

end of frame (EoF) symbols on the fly based on [RXD.FRAMECONFIG](#) register configuration. It will, however, verify and transfer the CRC bytes into RAM, if the CRC is enabled through [RXD.FRAMECONFIG](#).

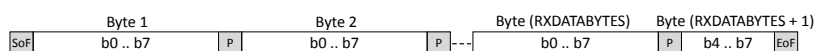
When an EoF symbol is detected, the NFCT peripheral will assert the [RXFRAMEEND](#) event and write the [RXD.AMOUNT](#) register to indicate numbers of received bytes and bits in the data packet. The module does not interpret the content of the data received from the remote NFC device, except for SoF, EoF, parity, and CRC checking, as described above. The frame disassemble operation is illustrated below.

Frame on air

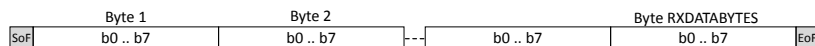
PARITY = Parity
RXDATABITS = 0
CRCMODERX = CRC16RX



PARITY = Parity
CRCMODERX = NoCRCTR
RXDATABITS = 4



PARITY = NoParity
CRCMODERX = NoCRCRX
RXDATABITS = 0



Data to RAM

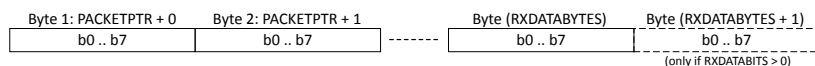


Figure 63: Frame disassemble illustration

Per NFC specification, the time between EoF to the next SoF can be as short as 86 μ s, and therefore care must be taken that PACKETPTR and MAXLEN are ready and ENABLERXDATA is issued on time after the end of previous frame. The use of a PPI shortcut from [TXFRAMEEND](#) to ENABLERXDATA is recommended.

6.13.7 Frame timing controller

The NFCT peripheral includes a frame timing controller that continuously keeps track of the number of the 13.56 MHz RF carrier clock periods since the end of the EoF of the last received frame.

The NFCT peripheral can be programmed to send a responding frame within a time window or at an exact count of RF carrier periods. In case of [FRAMEDELAYMODE](#) = Window, a [STARTTX](#) task triggered before the frame timing controller counter is equal to [FRAMEDELAYMIN](#) will force the transmission to halt until the counter is equal to [FRAMEDELAYMIN](#). If the counter is within [FRAMEDELAYMIN](#) and [FRAMEDELAYMAX](#) when the [STARTTX](#) task is triggered, the NFCT peripheral will start the transmission straight away. In case of [FRAMEDELAYMODE](#) = ExactVal, a [STARTTX](#) task triggered before the frame delay counter is equal to [FRAMEDELAYMAX](#) will halt the actual transmission start until the counter is equal to [FRAMEDELAYMAX](#).

In case of [FRAMEDELAYMODE](#) = WindowGrid, the behaviour is similar to the [FRAMEDELAYMODE](#) = Window, but the actual transmission between [FRAMEDELAYMIN](#) and [FRAMEDELAYMAX](#) starts on a bit grid as defined for NFC-A Listen frames (slot duration of 128 RF carrier periods).

An [ERROR](#) event (with [FRAMEDELAYTIMEOUT](#) cause in [ERRORSTATUS](#)) will be asserted if the frame timing controller counter reaches [FRAMEDELAYMAX](#) without any [STARTTX](#) task triggered. This may happen even when the response is not required as per *NFC Forum, NFC Digital Protocol Technical Specification*. Any commands handled by the automatic collision resolution that don't involve a response being generated may also result in an [ERROR](#) event (with [FRAMEDELAYTIMEOUT](#) cause in [ERRORSTATUS](#)). The [FRAMEDELAYMIN](#) and [FRAMEDELAYMAX](#) values shall only be updated before the [STARTTX](#) task is triggered. Failing to do so may cause unpredictable behaviour.

The frame timing controller operation is illustrated in [Frame timing controller \(FRAMEDELAYMODE=Window\)](#) on page 202. The frame timing controller automatically adjusts the frame timing counter based on the last received data bit according to NFC-A technology in the *NFC Forum, NFC Digital Protocol Technical Specification*.

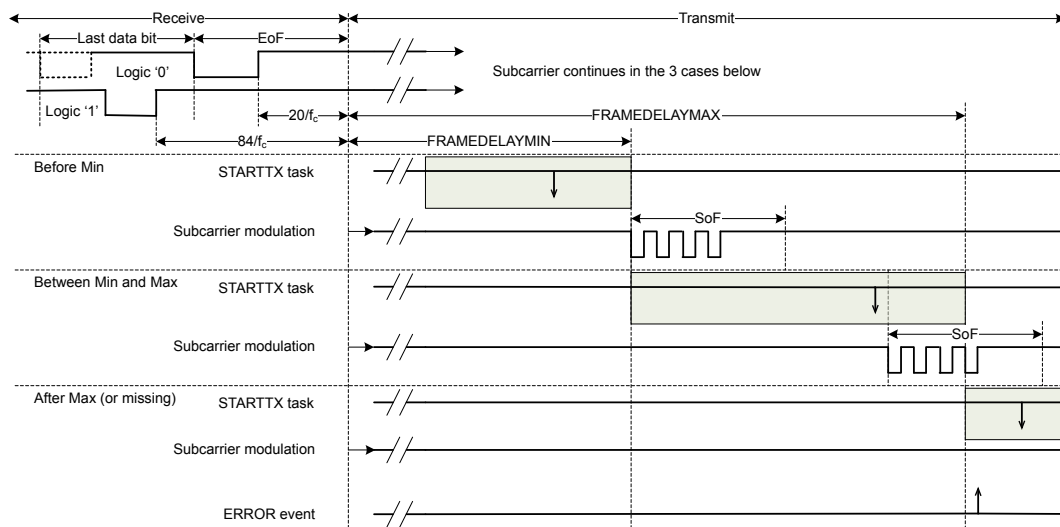


Figure 64: Frame timing controller (FRAMEDELAYMODE=Window)

6.13.8 Collision resolution

The NFCT peripheral implements an automatic collision resolution function as defined by the NFC Forum.

Automatic collision resolution is enabled by default, and it is recommended that the feature is used since it is power efficient and reduces the complexity of software handling the collision resolution sequence. This feature can be disabled through the MODE field in the [AUTOCOLRESCONFIG](#) register. When the automatic collision resolution is disabled, all commands will be sent over EasyDMA as defined in frame disassembler.

The [SENSRES](#) and [SELRES](#) registers need to be programmed upfront in order for the collision resolution to behave correctly. Depending on the NFCIDSIZE field in SENSRES, the following registers also need to be programmed upfront:

- NFCID1_LAST if NFCID1SIZE=NFCID1Single (ID = 4 bytes);
- NFCID1_2ND_LAST and NFCID1_LAST if NFCID1SIZE=NFCID1Double (ID = 7 bytes);
- NFCID1_3RD_LAST, NFCID1_2ND_LAST and NFCID1_LAST if NFCID1SIZE=NFCID1Triple (ID = 10 bytes);

A pre-defined set of registers, NFC.TAGHEADER0..3, containing a valid NFCID1 value, is available in FICR and can be used by software to populate the NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST registers.

[NFCID1 byte allocation \(top sent first on air\)](#) on page 203 explains the position of the ID bytes in NFCID1_3RD_LAST, NFCID1_2ND_LAST, and NFCID1_LAST, depending on the ID size, and as compared to the definition used in the *NFC Forum, NFC Digital Protocol Technical Specification*.

	ID = 4 bytes	ID = 7 bytes	ID = 10 bytes
NFCID1_Q			nfcid1 ₀
NFCID1_R			nfcid1 ₁
NFCID1_S			nfcid1 ₂
NFCID1_T		nfcid1 ₀	nfcid1 ₃
NFCID1_U		nfcid1 ₁	nfcid1 ₄
NFCID1_V		nfcid1 ₂	nfcid1 ₅
NFCID1_W	nfcid1 ₀	nfcid1 ₃	nfcid1 ₆
NFCID1_X	nfcid1 ₁	nfcid1 ₄	nfcid1 ₇
NFCID1_Y	nfcid1 ₂	nfcid1 ₅	nfcid1 ₈
NFCID1_Z	nfcid1 ₃	nfcid1 ₆	nfcid1 ₉

Table 55: NFCID1 byte allocation (top sent first on air)

The hardware implementation can handle the states from IDLE to ACTIVE_A automatically as defined in the *NFC Forum, NFC Activity Technical Specification*, and the other states are to be handled by software. The software keeps track of the state through events. The collision resolution will trigger an **AUTOCOLRESSTARTED** event when it has started. Reaching the ACTIVE_A state is indicated by the **SELECTED** event.

If collision resolution fails, a **COLLISION** event is triggered. Note that errors occurring during automatic collision resolution may also cause **ERROR** and/or **RXERROR** events to be generated. Other events may also get generated. It is recommended that the software ignores any event except COLLISION, SELECTED and FIELDLOST during automatic collision resolution. Software shall also make sure that any unwanted SHORT or PPI shortcut is disabled during automatic collision resolution.

The automatic collision resolution will be restarted, if the packets are received with CRC or parity errors while in ACTIVE_A state. The automatic collision resolution feature can be disabled while in ACTIVE_A state to avoid this.

The SLP_REQ is automatically handled by the NFCT peripheral when the automatic collision resolution is enabled. However, this results in an ERROR event (with FRAMEDELAYTIMEOUT cause in ERRORSTATUS) since the SLP_REQ has no response. This error must be ignored until the SELECTED event is triggered and this error should be cleared by the software when the SELECTED event is triggered.

6.13.9 Antenna interface

In ACTIVATED state, an amplitude regulator will adjust the voltage swing on the antenna pins to a value that is within the V_{swing} limit.

Refer to [NFCT Electrical Specification](#) on page 224.

6.13.10 NFCT antenna recommendations

The NFCT antenna coil must be connected differential between **NFC1** and **NFC2** pins of the device.

Two external capacitors should be used to tune the resonance of the antenna circuit to 13.56 MHz.

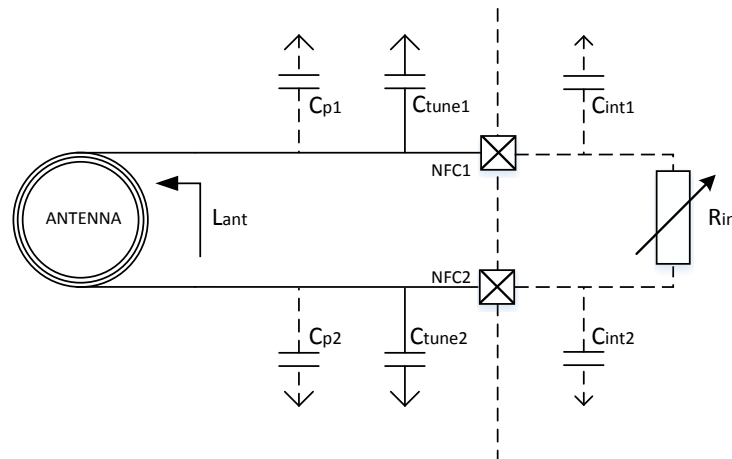


Figure 65: NFCT antenna recommendations

The required tuning capacitor value is given by the below equations:

$$C_{tune} = \frac{1}{(2\pi \cdot 13.56 \text{ MHz})^2 \cdot L_{ant}} \quad \text{where } C_{tune} = \frac{1}{2} \cdot (C_p + C_{int} + C_{tune})$$

$$\text{and } C_{tune1} = C_{tune2} = C_{tune} \quad C_{p1} = C_{p2} = C_p \quad C_{int1} = C_{int2} = C_{int}$$

$$C_{tune} = \frac{2}{(2\pi \cdot 13.56 \text{ MHz})^2 \cdot L_{ant}} - C_p - C_{int}$$

An antenna inductance of $L_{ant} = 2 \mu\text{H}$ will give tuning capacitors in the range of 130 pF on each pin. The total capacitance on **NFC1** and **NFC2** must be matched.

6.13.11 Battery protection

If the antenna is exposed to a strong NFC field, current may flow in the opposite direction on the supply due to parasitic diodes and ESD structures.

If the battery used does not tolerate return current, a series diode must be placed between the battery and the device in order to protect the battery.

6.13.12 Digital Modulation Signal

Support for external analog frontends or antenna architectures is possible by optionally outputting the digital modulation signal to a GPIO.

The NFCT peripheral is designed to connect directly to a loop antenna, receive a modulated signal from an NFC Reader with its internal analog frontend and transmit data back by changing the input resistance that is then seen as modulated load by the NFC Reader.

In addition, the peripheral has an option to output the digital modulation signal to a GPIO. Reception still occurs through the internal analog frontend, whereas transmission can be done by one of the following:

- The internal analog frontend through the loop antenna (default)
- An external frontend using the digital modulation signal
- The combination of both above

There are two registers that allow configuration of the modulation signal (i.e. of the response from NFCT to the NFC Reader), [MODULATIONCTRL](#) and [MODULATIONPSEL](#). The registers need to be programmed before NFCT sends a response to a request from a reader. Ideally, this configuration is performed during startup and whenever the NFCT peripheral is powered up.

The selected GPIO needs to be configured as output in the corresponding GPIO configuration register. It is recommended to set an output value in the corresponding GPIO.OUT register – this value will be driven whenever the NFCT peripheral is disabled.

NFCT drives the pin low when there is no modulation, and drives it with On-Off Keying (OOK) modulation of an 847 kHz subcarrier (derived from the carrier frequency) when it responds to commands from an NFC Reader.

6.13.13 References

NFC Forum, NFC Analog Specification version 1.0, www.nfc-forum.org

NFC Forum, NFC Digital Protocol Technical Specification version 1.1, www.nfc-forum.org

NFC Forum, NFC Activity Technical Specification version 1.1, www.nfc-forum.org

6.13.14 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40005000	NFCT	NFCT	Near field communication tag	

Table 56: Instances

Register	Offset	Description
TASKS_ACTIVATE	0x000	Activate NFCT peripheral for incoming and outgoing frames, change state to activated
TASKS_DISABLE	0x004	Disable NFCT peripheral
TASKS_SENSE	0x008	Enable NFC sense field mode, change state to sense mode
TASKS_STARTTX	0x00C	Start transmission of an outgoing frame, change state to transmit
TASKS_ENABLERXDATA	0x01C	Initializes the EasyDMA for receive.
TASKS_GOidle	0x024	Force state machine to IDLE state
TASKS_GOSLEEP	0x028	Force state machine to SLEEP_A state
EVENTS_READY	0x100	The NFCT peripheral is ready to receive and send frames
EVENTS_FIELDDETECTED	0x104	Remote NFC field detected
EVENTS_FIELDLOST	0x108	Remote NFC field lost
EVENTS_TXFRAMESTART	0x10C	Marks the start of the first symbol of a transmitted frame
EVENTS_TXFRAMEEND	0x110	Marks the end of the last transmitted on-air symbol of a frame
EVENTS_RXFRAMESTART	0x114	Marks the end of the first symbol of a received frame
EVENTS_RXFRAMEEND	0x118	Received data has been checked (CRC, parity) and transferred to RAM, and EasyDMA has ended accessing the RX buffer
EVENTS_ERROR	0x11C	NFC error reported. The ERRORSTATUS register contains details on the source of the error.
EVENTS_RXERROR	0x128	NFC RX frame error reported. The FRAMSTATUS.RX register contains details on the source of the error.
EVENTS_ENDRX	0x12C	RX buffer (as defined by PACKETPTR and MAXLEN) in Data RAM full.
EVENTS_ENDTX	0x130	Transmission of data in RAM has ended, and EasyDMA has ended accessing the TX buffer
EVENTS_AUTOCOLRESSTARTED	0x138	Auto collision resolution process has started
EVENTS_COLLISION	0x148	NFC auto collision resolution error reported.
EVENTS_SELECTED	0x14C	NFC auto collision resolution successfully completed
EVENTS_STARTED	0x150	EasyDMA is ready to receive or send frames.
SHORTS	0x200	Shortcuts between local events and tasks
INTEN	0x300	Enable or disable interrupt

Register	Offset	Description
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ERRORSTATUS	0x404	NFC Error Status register
FRAMESTATUS.RX	0x40C	Result of last incoming frame
NFCTAGSTATE	0x410	NfcTag state register
SLEEPSTATE	0x420	Sleep state during automatic collision resolution
FIELDPRESENT	0x43C	Indicates the presence or not of a valid field
FRAMEDELAYMIN	0x504	Minimum frame delay
FRAMEDELAYMAX	0x508	Maximum frame delay
FRAMEDELAYMODE	0x50C	Configuration register for the Frame Delay Timer
PACKETPTR	0x510	Packet pointer for TXD and RXD data storage in Data RAM
MAXLEN	0x514	Size of the RAM buffer allocated to TXD and RXD data storage each
TXD.FRAMECONFIG	0x518	Configuration of outgoing frames
TXD.AMOUNT	0x51C	Size of outgoing frame
RXD.FRAMECONFIG	0x520	Configuration of incoming frames
RXD.AMOUNT	0x524	Size of last incoming frame
MODULATIONCTRL	0x52C	Enables the modulation output to a GPIO pin which can be connected to a second external antenna.
MODULATIONPSEL	0x538	Pin select for Modulation control.
NFCID1_LAST	0x590	Last NFCID1 part (4, 7 or 10 bytes ID)
NFCID1_2ND_LAST	0x594	Second last NFCID1 part (7 or 10 bytes ID)
NFCID1_3RD_LAST	0x598	Third last NFCID1 part (10 bytes ID)
AUTOCOLRESCONFIG	0x59C	Controls the auto collision resolution function. This setting must be done before the NFCT peripheral is activated.
SENSRES	0x5A0	NFC-A SENS_RES auto-response settings
SELRES	0x5A4	NFC-A SEL_RES auto-response settings

Table 57: Register overview

6.13.14.1 TASKS_ACTIVATE

Address offset: 0x000

Activate NFCT peripheral for incoming and outgoing frames, change state to activated

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W	TASKS_ACTIVATE		Activate NFCT peripheral for incoming and outgoing frames, change state to activated																															
		Trigger	1	Trigger task																															

6.13.14.2 TASKS_DISABLE

Address offset: 0x004

Disable NFCT peripheral

6.13.14.3 TASKS SENSE

Enable NFC sense field mode, change state to sense mode

6.13.14.4 TASKS STARTTX

Start transmission of an outgoing frame, change state to transmit

6.13.14.5 TASKS_ENABLERXDATA

Initializes the EasyDMA for receive.

6.13.14.6 TASKS GOIDLE

Force state machine to IDLE state

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID																																				A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ID	Acce Field		Value ID	Value				Description																												
A	W	TASKS_GOIDLE						Force state machine to IDLE state																												
			Trigger	1				Trigger task																												

6.13.14.7 TASKS_GOSLEEP

Address offset: 0x028

Force state machine to SLEEP_A state

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field		Value ID	Value				Description																											
A	W	TASKS_GOSLEEP						Force state machine to SLEEP_A state																											
		Trigger		1				Trigger task																											

6.13.14.8 EVENTS_READY

Address offset: 0x100

The NFCT peripheral is ready to receive and send frames

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	EVENTS_READY		The NFCT peripheral is ready to receive and send frames																														
		NotGenerated	0	Event not generated																														
		Generated	1	Event generated																														

6.13.14.9 EVENTS_FIELDDETECTED

Address offset: 0x104

Remote NFC field detected

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
ID			A																																				
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
ID	Acce Field	Value ID	Value		Description																																		
A	RW	EVENTS_FIELDDETECTED			Remote NFC field detected																																		
		NotGenerated	0		Event not generated																																		
		Generated	1		Event generated																																		

6.13.14.10 EVENTS_FIELDLOST

Address offset: 0x108

Remote NFC field lost

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A																															
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																															
A	RW	EVENTS_FIELDLOST		Remote NFC field lost																															
		NotGenerated	0	Event not generated																															
		Generated	1	Event generated																															

6.13.14.11 EVENTS_TXFRAMESTART

Address offset: 0x10C

Marks the start of the first symbol of a transmitted frame

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field		Value ID	Value	Description																														
A	RW	EVENTS_TXFRAMESTART			Marks the start of the first symbol of a transmitted frame																														
		NotGenerated		0	Event not generated																														
		Generated		1	Event generated																														

6.13.14.12 EVENTS_TXFRAMEEND

Address offset: 0x110

Marks the end of the last transmitted on-air symbol of a frame

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	RW	EVENTS_TXFRAMEEND		Marks the end of the last transmitted on-air symbol of a frame																															
		NotGenerated	0	Event not generated																															
		Generated	1	Event generated																															

6.13.14.13 EVENTS_RXFRAMESTART

Address offset: 0x114

Marks the end of the first symbol of a received frame

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID				A																																			
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
ID	Acce	Field	Value	ID	Value	Description																																	
A	RW	EVENTS_RXFRAMESTART				Marks the end of the first symbol of a received frame																																	
			NotGenerated	0	Event not generated																																		
			Generated	1	Event generated																																		

6.13.14.14 EVENTS_RXFRAMEEND

Address offset: 0x118

Received data has been checked (CRC, parity) and transferred to RAM, and EasyDMA has ended accessing the RX buffer

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID										A																																	
Reset 0x00000000										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID		Value		Description																																				
A	RW		EVENTS_RXFRAMEEND				Received data has been checked (CRC, parity) and transferred to RAM, and EasyDMA has ended accessing the RX buffer																																				
			NotGenerated		0		Event not generated																																				
			Generated		1		Event generated																																				

6.13.14.15 EVENTS_ERROR

Address offset: 0x11C

NFC error reported. The ERRORSTATUS register contains details on the source of the error.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID				A																																			
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	Acce	Field	Value	ID	Value	Description																																	
A	RW	EVENTS_ERROR				NFC error reported. The ERRORSTATUS register contains details on the source of the error.																																	
			NotGenerated	0		Event not generated																																	
			Generated	1		Event generated																																	

6.13.14.16 EVENTS_RXERROR

Address offset: 0x128

NFC RX frame error reported. The FRAMESTATUS.RX register contains details on the source of the error.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A																															
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																															
A	RW	EVENTS_RXERROR		NFC RX frame error reported. The FRAMESTATUS.RX register contains details on the source of the error.																															
		NotGenerated	0	Event not generated																															
		Generated	1	Event generated																															

6.13.14.17 EVENTS_ENDRX

Address offset: 0x12C

RX buffer (as defined by PACKETPTR and MAXLEN) in Data RAM full.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID				A																																
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																
ID	Acce	Field	Value ID	Value	Description																															
A	RW	EVENTS_ENDRX			RX buffer (as defined by PACKETPTR and MAXLEN) in Data RAM full.																															
			NotGenerated	0	Event not generated																															
			Generated	1	Event generated																															

6.13.14.18 EVENTS_ENDTX

Address offset: 0x130

Transmission of data in RAM has ended, and EasyDMA has ended accessing the TX buffer

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																													
A	RW	EVENTS_ENDTX				Transmission of data in RAM has ended, and EasyDMA has ended accessing the TX buffer																													
			NotGenerated	0		Event not generated																													
			Generated	1		Event generated																													

6.13.14.19 EVENTS_AUTOCOLRESSTARTED

Address offset: 0x138

Auto collision resolution process has started

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID																																				A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																																
A	RW	EVENTS_AUTOCOLRESSTARTED		Auto collision resolution process has started																																
		NotGenerated	0	Event not generated																																
		Generated	1	Event generated																																

6.13.14.20 EVENTS_COLLISION

Address offset: 0x148

NFC auto collision resolution error reported.

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID																																				A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																																
A	RW	EVENTS_COLLISION		NFC auto collision resolution error reported.																																
		NotGenerated	0	Event not generated																																
		Generated	1	Event generated																																

6.13.14.21 EVENTS_SELECTED

Address offset: 0x14C

NFC auto collision resolution successfully completed

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A																															
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value		Description																													
A	RW	EVENTS_SELECTED			NFC auto collision resolution successfully completed																													
		NotGenerated	0		Event not generated																													
		Generated	1		Event generated																													

6.13.14.22 EVENTS_STARTED

Address offset: 0x150

EasyDMA is ready to receive or send frames.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																	A	
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value		Description																													
A	RW	EVENTS_STARTED			EasyDMA is ready to receive or send frames.																													
		NotGenerated	0		Event not generated																													
		Generated	1		Event generated																													

6.13.14.23 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			F B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	FIELDDETECTED_ACTIVATE		Shortcut between event FIELDDETECTED and task ACTIVATE																														
		Disabled	0	Disable shortcut																														
		Enabled	1	Enable shortcut																														
B	RW	FIELDLOST_SENSE		Shortcut between event FIELDLOST and task SENSE																														
		Disabled	0	Disable shortcut																														
		Enabled	1	Enable shortcut																														
F	RW	TXFRAMEEND_ENABLERXDATA		Shortcut between event TXFRAMEEND and task ENABLERXDATA																														
		Disabled	0	Disable shortcut																														
		Enabled	1	Enable shortcut																														

6.13.14.24 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			T S R N M L K H G F E D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW READY			Enable or disable interrupt for event READY																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
B	RW FIELDDETECTED			Enable or disable interrupt for event FIELDDETECTED																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
C	RW FIELDLOST			Enable or disable interrupt for event FIELDLOST																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
D	RW TXFRAMESTART			Enable or disable interrupt for event TXFRAMESTART																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
E	RW TXFRAMEEND			Enable or disable interrupt for event TXFRAMEEND																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
F	RW RXFRAMESTART			Enable or disable interrupt for event RXFRAMESTART																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
G	RW RXFRAMEEND			Enable or disable interrupt for event RXFRAMEEND																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
H	RW ERROR			Enable or disable interrupt for event ERROR																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
K	RW RXERROR			Enable or disable interrupt for event RXERROR																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
L	RW ENDRX			Enable or disable interrupt for event ENDRX																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
M	RW ENDTX			Enable or disable interrupt for event ENDTX																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
N	RW AUTOCOLRESSTARTED			Enable or disable interrupt for event AUTOCOLRESSTARTED																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
R	RW COLLISION			Enable or disable interrupt for event COLLISION																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
S	RW SELECTED			Enable or disable interrupt for event SELECTED																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														
T	RW STARTED			Enable or disable interrupt for event STARTED																														
		Disabled	0	Disable																														
		Enabled	1	Enable																														

6.13.14.25 INTENSET

Address offset: 0x304

Enable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			T S R																N				M L K				H G F E				D C B A			
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
		S RW SELECTED			Write '1' to enable interrupt for event SELECTED																													
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
		T RW STARTED			Write '1' to enable interrupt for event STARTED																													
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														

6.13.14.26 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			T S R																N				M L K				H G F E				D C B A			
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																													
A	RW	READY			Write '1' to disable interrupt for event READY																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
B	RW	FIELDDETECTED			Write '1' to disable interrupt for event FIELDDETECTED																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
C	RW	FIELDLOST			Write '1' to disable interrupt for event FIELDLOST																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
D	RW	TXFRAMESTART			Write '1' to disable interrupt for event TXFRAMESTART																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
E	RW	TXFRAMEEND			Write '1' to disable interrupt for event TXFRAMEEND																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
F	RW	RXFRAMESTART			Write '1' to disable interrupt for event RXFRAMESTART																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
G	RW	RXFRAMEEND			Write '1' to disable interrupt for event RXFRAMEEND																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			T S R																N				M L K				H G F E				D C B A			
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																													
H	RW	ERROR			Write '1' to disable interrupt for event ERROR																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
K	RW	RXERROR			Write '1' to disable interrupt for event RXERROR																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
L	RW	ENDRX			Write '1' to disable interrupt for event ENDRX																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
M	RW	ENDTX			Write '1' to disable interrupt for event ENDTX																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
N	RW	AUTOCOLRESSTARTED			Write '1' to disable interrupt for event AUTOCOLRESSTARTED																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
R	RW	COLLISION			Write '1' to disable interrupt for event COLLISION																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
S	RW	SELECTED			Write '1' to disable interrupt for event SELECTED																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
T	RW	STARTED			Write '1' to disable interrupt for event STARTED																													
			Clear	1	Disable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													

6.13.14.27 ERRORSTATUS

Address offset: 0x404

NFC Error Status register

Write a bit to '1' to clear it. Writing '0' has no effect.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	FRAMEDELAYTIMEOUT		No STARTTX task triggered before expiration of the time set in FRAMEDELAYMAX																														

6.13.14.28 FRAMESTATUS.RX

Address offset: 0x40C

Result of last incoming frame

Write a bit to '1' to clear it. Writing '0' has no effect.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																			
ID			C B A																																			
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																			
ID	Acce	Field	Value	ID	Value	Description																																
A	RW	CRCERROR				No valid end of frame (EoF) detected																																
		CRCCorrect	0			Valid CRC detected																																
		CRCError	1			CRC received does not match local check																																
B	RW	PARITYSTATUS				Parity status of received frame																																
		ParityOK	0			Frame received with parity OK																																
		ParityError	1			Frame received with parity error																																
C	RW	OVERRUN				Overrun detected																																
		NoOverrun	0			No overrun detected																																
		Overrun	1			Overrun error																																

6.13.14.29 NFCTAGSTATE

Address offset: 0x410

NfcTag state register

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A	R	NFCTAGSTATE				NfcTag state																												
		Disabled	0	Disabled or sense																														
		RampUp	2	RampUp																														
		Idle	3	Idle																														
		Receive	4	Receive																														
		FrameDelay	5	FrameDelay																														
		Transmit	6	Transmit																														

6.13.14.30 SLEEPSTATE

Address offset: 0x420

Sleep state during automatic collision resolution

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	R	SLEEPSTATE		Reflects the sleep state during automatic collision resolution. Set to IDLE by a GOIDLE task. Set to SLEEP_A when a valid SLEEP_REQ frame is received or by a GOSLEEP task.																														
		Idle	0	State is IDLE.																														
		SleepA	1	State is SLEEP_A.																														

6.13.14.31 FIELDPRESENT

Address offset: 0x43C

Indicates the presence or not of a valid field

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				B A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																														
A	R	FIELDPRESENT			Indicates if a valid field is present. Available only in the activated state.																														
			NoField	0	No valid field detected																														
			FieldPresent	1	Valid field detected																														
B	R	LOCKDETECT			Indicates if the low level has locked to the field																														
			NotLocked	0	Not locked to field																														
			Locked	1	Locked to field																														

6.13.14.32 FRAMEDELAYMIN

Address offset: 0x504

Minimum frame delay

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A																															
Reset 0x00000480				0 0																															

6.13.14.33 FRAMEDELAYMAX

Address offset: 0x508

Maximum frame delay

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A A																															

Configuration register for the Frame Delay Timer

6.13.14.35 PACKETPTR

Packet pointer for TXD and RXD data storage in Data RAM

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.13.14.36 MAXLEN

Size of the RAM buffer allocated to TXD and RXD data storage each

6.13.14.37 TXD.FRAMECONFIG

Configuration of outgoing frames

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																							
ID																																								C	B	A
Reset 0x00000015			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1																																							
ID	Acce Field	Value ID	Value	Description																																						
A	RW	PARITY		Indicates if parity expected in RX frame																																						
		NoParity	0	Parity is not expected in RX frames																																						
		Parity	1	Parity is expected in RX frames																																						
B	RW	SOF		SoF expected or not in RX frames																																						
		NoSoF	0	SoF symbol is not expected in RX frames																																						
		SoF	1	SoF symbol is expected in RX frames																																						
C	RW	CRCMODERX		CRC mode for incoming frames																																						
		NoCRCRX	0	CRC is not expected in RX frames																																						
		CRC16RX	1	Last 16 bits in RX frame is CRC, CRC is checked and CRCSTATUS updated																																						

6.13.14.40 RXD.AMOUNT

Address offset: 0x524

Size of last incoming frame

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			B B B B B B B B B B A A A A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																													
A	R	RXDATABITS			Number of bits in the last byte in the frame, if less than 8 (including CRC, but excluding parity and SoF/EoF framing). Frames with 0 data bytes and less than 7 data bits are invalid and are not received properly.																													
B	R	RXDATABYTES			Number of complete bytes received in the frame (including CRC, but excluding parity and SoF/EoF framing)																													

6.13.14.41 MODULATIONCTRL

Address offset: 0x52C

Enables the modulation output to a GPIO pin which can be connected to a second external antenna.

See [MODULATIONPSEL](#) for GPIO configuration.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A																															
Reset 0x00000001			0 0																															

6.13.14.42 MODULATIONPSEL

Address offset: 0x538

Pin select for Modulation control.

Last NFCID1 part (4, 7 or 10 bytes ID)

Second last NFCID1 part (7 or 10 bytes ID)

Third last NFCID1 part (10 bytes ID)

6.13.14.46 AUTOCOLRESCONFIG

Address offset: 0x59C

Controls the auto collision resolution function. This setting must be done before the NFCT peripheral is activated.

When modifying this register bit 1 must be written to '1'.

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A																															
Reset 0x00000002			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
ID	Acce Field	Value ID	Value		Description																													
A	RW	MODE			Enables/disables auto collision resolution																													
		Enabled	0	Auto collision resolution enabled																														
		Disabled	1	Auto collision resolution disabled																														

6.13.14.47 SENSRES

Address offset: 0x5A0

NFC-A SENS_RES auto-response settings

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
ID				E E E E D D D C C B A A A A A																																				
Reset 0x00000001				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1																																				
ID	Acce	Field	Value ID	Value	Description																																			
A	RW	BITFRAMESDD			Bit frame SDD as defined by the b5:b1 of byte 1 in SENS_RES response in the NFC Forum, NFC Digital Protocol Technical Specification																																			
			SDD00000	0	SDD pattern 00000																																			
			SDD00001	1	SDD pattern 00001																																			
			SDD00010	2	SDD pattern 00010																																			
			SDD00100	4	SDD pattern 00100																																			
			SDD01000	8	SDD pattern 01000																																			
			SDD10000	16	SDD pattern 10000																																			
			B	RW	RFU5			Reserved for future use. Shall be 0.																																
C	RW	NFCIDSIZE			NFCID1 size. This value is used by the auto collision resolution engine.																																			
			NFCID1Single	0	NFCID1 size: single (4 bytes)																																			
			NFCID1Double	1	NFCID1 size: double (7 bytes)																																			
			NFCID1Triple	2	NFCID1 size: triple (10 bytes)																																			
D	RW	PLATFCONFIG			Tag platform configuration as defined by the b4:b1 of byte 2 in SENS_RES response in the NFC Forum, NFC Digital Protocol Technical Specification																																			
			E	RW	RFU74			Reserved for future use. Shall be 0.																																

6.13.14.48 SELRES

Address offset: 0x5A4

NFC-A SEL_RES auto-response settings

and supports single-channel or dual-channel (left and right) data input. Data is transferred directly to RAM buffers using EasyDMA.

Listed here are the main features for PDM:

- Up to two PDM microphones configured as a left/right pair using the same data input
- 16 kHz output sample rate, 16-bit samples
- EasyDMA support for sample buffering
- HW decimation filters
- Selectable ratio of 64 or 80 between PDM_CLK and output sample rate

The PDM module illustrated below is interfacing up to two digital microphones with the PDM interface. EasyDMA is implemented to relieve the real-time requirements associated with controlling of the PDM slave from a low priority CPU execution context. It also includes all the necessary digital filter elements to produce pulse code modulation (PCM) samples. The PDM module allows continuous audio streaming.

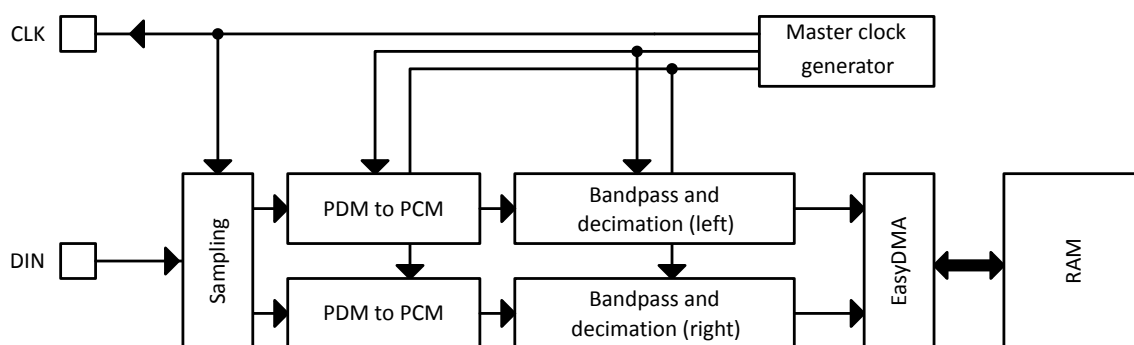


Figure 67: PDM module

6.14.1 Master clock generator

The master clock generator's PDMCLKCTRL register allows adjusting the PDM clock's frequency.

The master clock generator does not add any jitter to the HFCLK source chosen. It is recommended (but not mandatory) to use the Xtal as HFCLK source.

6.14.2 Module operation

By default, bits from the left PDM microphone are sampled on PDM_CLK falling edge, and bits for the right are sampled on the rising edge of PDM_CLK, resulting in two bitstreams. Each bitstream is fed into a digital filter which converts the PDM stream into 16-bit PCM samples, then filters and down-samples them to reach the appropriate sample rate.

The EDGE field in the MODE register allows swapping left and right, so that left will be sampled on rising edge, and right on falling.

The PDM module uses EasyDMA to store the samples coming out from the filters into one buffer in RAM. Depending on the mode chosen in the OPERATION field in the MODE register, memory either contains alternating left and right 16-bit samples (Stereo), or only left 16-bit samples (Mono). To ensure continuous PDM sampling, it is up to the application to update the EasyDMA destination address pointer as the previous buffer is filled.

The continuous transfer can be started or stopped by sending the START and STOP tasks. STOP becomes effective after the current frame has finished transferring, which will generate the STOPPED event. The STOPPED event indicates that all activity in the module is finished, and that the data is available in RAM (EasyDMA has finished transferring as well). Attempting to restart before receiving the STOPPED event may result in unpredictable behavior.

6.14.3 Decimation filter

In order to convert the incoming data stream into PCM audio samples, a decimation filter is included in the PDM interface module.

The input of the filter is the two-channel PDM serial stream (with left channel on clock high, right channel on clock low). Depending on the **RATIO** selected, its output is 2×16 -bit PCM samples at a sample rate either 64 times or 80 times (depending on the **RATIO** register) lower than the PDM clock rate.

The filter stage of each channel is followed by a digital volume control, to attenuate or amplify the output samples in a range of -20 dB to +20 dB around the default (reset) setting, defined by $G_{PDM, default}$. The gain is controlled by the **GAINL** and **GAINR** registers.

As an example, if the goal is to achieve 2500 RMS output samples (16-bit) with a 1 kHz 90 dBA signal into a -26 dBFS sensitivity PDM microphone, do the following:

- Sum the PDM module's default gain ($G_{PDM, default}$) and the gain introduced by the microphone and acoustic path of his implementation (an attenuation would translate into a negative gain)
- Adjust **GAINL** and **GAINR** by the above summed amount. Assuming that only the PDM module influences the gain, **GAINL** and **GAINR** must be set to $-G_{PDM, default}$ dB to achieve the requirement.

With $G_{PDM, default}=3.2$ dB, and as **GAINL** and **GAINR** are expressed in 0.5 dB steps, the closest value to program would be 3.0 dB, which can be calculated as:

```
GAINL = GAINR = (DefaultGain - (2 * 3))
```

Remember to check that the resulting values programmed into **GAINL** and **GAINR** fall within **MinGain** and **MaxGain**.

6.14.4 EasyDMA

Samples will be written directly to RAM, and EasyDMA must be configured accordingly.

The address pointer for the EasyDMA channel is set in **SAMPLE.PTR** register. If the destination address set in **SAMPLE.PTR** is not pointing to the Data RAM region, an EasyDMA transfer may result in a **HardFault** or RAM corruption. See [Memory](#) on page 19 for more information about the different memory regions.

DMA supports Stereo (Left+Right 16-bit samples) and Mono (Left only) data transfer, depending on the setting in the **OPERATION** field in the **MODE** register. The samples are stored little endian.

MODE.OPERATION	Bits per sample	Result stored per RAM word	Physical RAM allocated (32-bit words)	Result boundary indexes in RAM	Note
Stereo	32 (2x16)	L+R	$\text{ceil}(\text{SAMPLE.MAXCNT}/2)$	$R0=[31:16]; L0=[15:0]$	Default
Mono	16	2xL	$\text{ceil}(\text{SAMPLE.MAXCNT}/2)$	$L1=[31:16]; L0=[15:0]$	

Table 58: DMA sample storage

The destination buffer in RAM consists of one block, the size of which is set in **SAMPLE.MAXCNT** register. Format is number of 16-bit samples. The physical RAM allocated is always:

```
(RAM allocation, in bytes) = SAMPLE.MAXCNT * 2;
```

(but the mapping of the samples depends on **MODE.OPERATION**).

If **OPERATION**=Stereo, RAM will contain a succession of left and right samples.

If **OPERATION**=Mono, RAM will contain a succession of left only samples.

For a given value of `SAMPLE.MAXCNT`, the buffer in RAM can contain half the stereo sampling time as compared to the mono sampling time.

The PDM acquisition can be started by the `START` task, after the `SAMPLE.PTR` and `SAMPLE.MAXCNT` registers have been written. When starting the module, it will take some time for the filters to start outputting valid data. Transients from the PDM microphone itself may also occur. The first few samples (typically around 50) might hence contain invalid values or transients. It is therefore advised to discard the first few samples after a PDM start.

As soon as the `STARTED` event is received, the firmware can write the next `SAMPLE.PTR` value (this register is double-buffered), to ensure continuous operation.

When the buffer in RAM is filled with samples, an `END` event is triggered. The firmware can start processing the data in the buffer. Meanwhile, the PDM module starts acquiring data into the new buffer pointed to by `SAMPLE.PTR`, and sends a new `STARTED` event, so that the firmware can update `SAMPLE.PTR` to the next buffer address.

6.14.5 Hardware example

PDM can be configured with a single microphone (mono), or with two microphones.

When a single microphone is used, connect the microphone clock to `CLK`, and data to `DIN`.

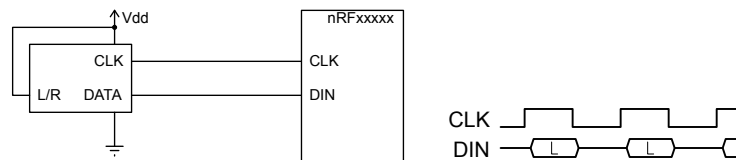


Figure 68: Example of a single PDM microphone, wired as left

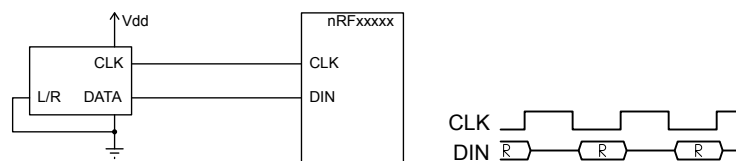


Figure 69: Example of a single PDM microphone, wired as right

Note that in a single-microphone (mono) configuration, depending on the microphone's implementation, either the left or the right channel (sampled at falling or rising CLK edge respectively) will contain reliable data.

If two microphones are used, one of them has to be set as left, the other as right (L/R pin tied high or to GND on the respective microphone). It is strongly recommended to use two microphones of exactly the same brand and type so that their timings in left and right operation match.

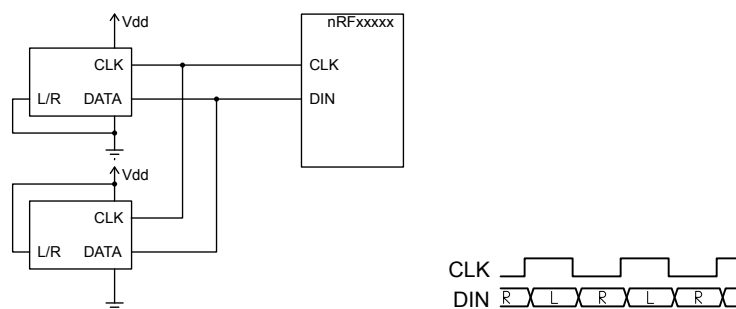


Figure 70: Example of two PDM microphones

6.14.6 Pin configuration

The CLK and DIN signals associated to the PDM module are mapped to physical pins according to the configuration specified in the PSEL.CLK and PSEL.DIN registers respectively. If the CONNECT field in any PSEL register is set to Disconnected, the associated PDM module signal will not be connected to the required physical pins, and will not operate properly.

The PSEL.CLK and PSEL.DIN registers and their configurations are only used as long as the PDM module is enabled, and retained only as long as the device is in System ON mode. See [POWER — Power supply](#) on page 58 for more information about power modes. When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register.

To ensure correct behavior in the PDM module, the pins used by the PDM module must be configured in the GPIO peripheral as described in [GPIO configuration before enabling peripheral](#) on page 228 before enabling the PDM module. This is to ensure that the pins used by the PDM module are driven correctly if the PDM module itself is temporarily disabled or the device temporarily enters System OFF. This configuration must be retained in the GPIO for the selected I/Os as long as the PDM module is supposed to be connected to an external PDM circuit.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

PDM signal	PDM pin	Direction	Output value	Comment
CLK	As specified in PSEL.CLK	Output	0	
DIN	As specified in PSEL.DIN	Input	Not applicable	

Table 59: GPIO configuration before enabling peripheral

6.14.7 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4001D000	PDM	PDM	Pulse Density modulation (digital microphone) interface	

Table 60: Instances

Register	Offset	Description
TASKS_START	0x000	Starts continuous PDM transfer
TASKS_STOP	0x004	Stops PDM transfer
EVENTS_STARTED	0x100	PDM transfer has started
EVENTS_STOPPED	0x104	PDM transfer has finished
EVENTS_END	0x108	The PDM has written the last sample specified by SAMPLE.MAXCNT (or the last sample after a STOP task has been received) to Data RAM
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ENABLE	0x500	PDM module enable register
PDMCLKCTRL	0x504	PDM clock generator control
MODE	0x508	Defines the routing of the connected PDM microphones' signals
GAINL	0x518	Left output gain adjustment
GAINR	0x51C	Right output gain adjustment
RATIO	0x520	Selects the ratio between PDM_CLK and output sample rate. Change PDMCLKCTRL accordingly.
PSEL.CLK	0x540	Pin number configuration for PDM CLK signal
PSEL.DIN	0x544	Pin number configuration for PDM DIN signal
SAMPLE.PTR	0x560	RAM address pointer to write samples to with EasyDMA

Register	Offset	Description
SAMPLE.MAXCNT	0x564	Number of samples to allocate memory for in EasyDMA mode

Table 61: Register overview

6.14.7.1 TASKS_START

Address offset: 0x000

Starts continuous PDM transfer

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	W	TASKS_START		Starts continuous PDM transfer																														
		Trigger	1	Trigger task																														

6.14.7.2 TASKS_STOP

Address offset: 0x004

Stops PDM transfer

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W	TASKS_STOP		Stops PDM transfer																															
		Trigger	1	Trigger task																															

6.14.7.3 EVENTS_STARTED

Address offset: 0x100

PDM transfer has started

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	EVENTS_STARTED		PDM transfer has started																														
		NotGenerated	0	Event not generated																														
		Generated	1	Event generated																														

6.14.7.4 EVENTS_STOPPED

Address offset: 0x104

PDM transfer has finished

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																		A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value		Description																													
A	RW	EVENTS_STOPPED			PDM transfer has finished																													
		NotGenerated	0	Event not generated																														
		Generated	1	Event generated																														

6.14.7.5 EVENTS_END

Address offset: 0x108

The PDM has written the last sample specified by SAMPLE.MAXCNT (or the last sample after a STOP task has been received) to Data RAM

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID																																						A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	Acce Field	Value ID	Value	Description																																		
A	RW	EVENTS_END		The PDM has written the last sample specified by SAMPLE.MAXCNT (or the last sample after a STOP task has been received) to Data RAM																																		
		NotGenerated	0	Event not generated																																		
		Generated	1	Event generated																																		

6.14.7.6 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID																																				C	B	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
ID	Acce	Field	Value	ID	Value	Description																																
A	RW	STARTED				Enable or disable interrupt for event STARTED																																
			Disabled	0	Disable																																	
			Enabled	1	Enable																																	
B	RW	STOPPED				Enable or disable interrupt for event STOPPED																																
			Disabled	0	Disable																																	
			Enabled	1	Enable																																	
C	RW	END				Enable or disable interrupt for event END																																
			Disabled	0	Disable																																	
			Enabled	1	Enable																																	

6.14.7.7 INTENSET

Address offset: 0x304

Enable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	STARTED		Write '1' to enable interrupt for event STARTED																														
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
B	RW	STOPPED		Write '1' to enable interrupt for event STOPPED																														
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
C	RW	END		Write '1' to enable interrupt for event END																														
		Set	1	Enable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														

6.14.7.8 INTENCLR

Address offset: 0x308

Disable interrupt

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																
ID				C B A																																
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																
ID	Acce Field		Value ID	Value	Description																															
A	RW STARTED				Write '1' to disable interrupt for event STARTED																															
			Clear	1	Disable																															
			Disabled	0	Read: Disabled																															
			Enabled	1	Read: Enabled																															
B	RW STOPPED				Write '1' to disable interrupt for event STOPPED																															
			Clear	1	Disable																															
			Disabled	0	Read: Disabled																															
			Enabled	1	Read: Enabled																															
C	RW END				Write '1' to disable interrupt for event END																															
			Clear	1	Disable																															
			Disabled	0	Read: Disabled																															
			Enabled	1	Read: Enabled																															

6.14.7.9 ENABLE

Address offset: 0x500

PDM module enable register

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																														
ID			A																														
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																														
ID	Acce Field	Value ID	Value	Description																													
A	RW	ENABLE		Enable or disable PDM module																													
		Disabled	0	Disable																													
		Enabled	1	Enable																													

6.14.7.10 PDMCLKCTRL

Address offset: 0x504

PDM clock generator control

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x08400000			0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce	Field	Value		ID	Value		Description																										
A	RW	FREQ						PDM_CLK frequency configuration																										
			1000K	0x08000000		PDM_CLK = 32 MHz / 32 = 1.000 MHz																												
			Default	0x08400000		PDM_CLK = 32 MHz / 31 = 1.032 MHz. Nominal clock for RATIO=Ratio64.																												
			1067K	0x08800000		PDM_CLK = 32 MHz / 30 = 1.067 MHz																												
			1231K	0x09800000		PDM_CLK = 32 MHz / 26 = 1.231 MHz																												
			1280K	0x0A000000		PDM_CLK = 32 MHz / 25 = 1.280 MHz. Nominal clock for RATIO=Ratio80.																												
			1333K	0x0A800000		PDM_CLK = 32 MHz / 24 = 1.333 MHz																												

6.14.7.11 MODE

Address offset: 0x508

Defines the routing of the connected PDM microphones' signals

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID																																		B	A			
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
ID	Acce	Field	Value	ID	Value	Description																																
A	RW	OPERATION				Mono or stereo operation																																
			Stereo	0		Sample and store one pair (left + right) of 16-bit samples per RAM word R=[31:16]; L=[15:0]																																
			Mono	1		Sample and store two successive left samples (16 bits each) per RAM word L1=[31:16]; L0=[15:0]																																
B	RW	EDGE				Defines on which PDM_CLK edge left (or mono) is sampled																																
			LeftFalling	0		Left (or mono) is sampled on falling edge of PDM_CLK																																
			LeftRising	1		Left (or mono) is sampled on rising edge of PDM_CLK																																

6.14.7.12 GAINL

Address offset: 0x518

Left output gain adjustment

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A A A A A																															
Reset 0x00000028			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	GAINL		Left output gain adjustment, in 0.5 dB steps, around the default module gain (see electrical parameters)																														
				0x00 -20 dB gain adjust																														
				0x01 -19.5 dB gain adjust																														
				(...)																														
				0x27 -0.5 dB gain adjust																														
				0x28 0 dB gain adjust																														
				0x29 +0.5 dB gain adjust																														
				(...)																														
				0x4F +19.5 dB gain adjust																														
				0x50 +20 dB gain adjust																														
		MinGain	0x00	-20 dB gain adjustment (minimum)																														
		DefaultGain	0x28	0 dB gain adjustment																														
		MaxGain	0x50	+20 dB gain adjustment (maximum)																														

6.14.7.13 GAINR

Address offset: 0x51C

Right output gain adjustment

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A																															

6.14.7.14 RATIO

Address offset: 0x520

Selects the ratio between PDM_CLK and output sample rate. Change PDMCLKCTRL accordingly.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	RATIO		Selects the ratio between PDM_CLK and output sample rate																														
		Ratio64	0	Ratio of 64																														
		Ratio80	1	Ratio of 80																														

6.14.7.15 PSEL.CLK

Address offset: 0x540

6.14.8 Electrical specification

6.14.8.1 PDM Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
$f_{\text{PDM,CLK},64}$	PDM clock speed. PDMCLKCTRL = Default (Setting needed for 16 MHz sample frequency @ $\text{RATIO} = \text{Ratio64}$)		1.032		MHz
$f_{\text{PDM,CLK},80}$	PDM clock speed. PDMCLKCTRL = 1280K (Setting needed for 16 MHz sample frequency @ $\text{RATIO} = \text{Ratio80}$)		1.280		MHz
$t_{\text{PDM,JITTER}}$	Jitter in PDM clock output			20	ns
$T_{\text{dPDM,CLK}}$	PDM clock duty cycle	40	50	60	%
$t_{\text{PDM,DATA}}$	Decimation filter delay			5	ms
$t_{\text{PDM,cv}}$	Allowed clock edge to data valid			125	ns
$t_{\text{PDM,ci}}$	Allowed (other) clock edge to data invalid	0			ns
$t_{\text{PDM,s}}$	Data setup time at $f_{\text{PDM,CLK}}=1.024$ MHz or 1.280 MHz	65			ns
$t_{\text{PDM,h}}$	Data hold time at $f_{\text{PDM,CLK}}=1.024$ MHz or 1.280 MHz	0			ns
$G_{\text{PDM,default}}$	Default (reset) absolute gain of the PDM module		3.2		dB

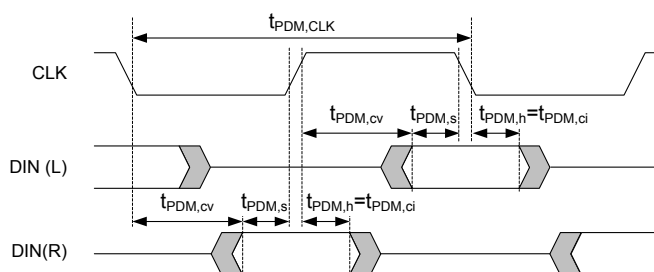


Figure 71: PDM timing diagram

6.15 PPI — Programmable peripheral interconnect

The programmable peripheral interconnect (PPI) enables peripherals to interact autonomously with each other using tasks and events independent of the CPU. The PPI allows precise synchronization between peripherals when real-time application constraints exist and eliminates the need for CPU activity to implement behavior which can be predefined using PPI.

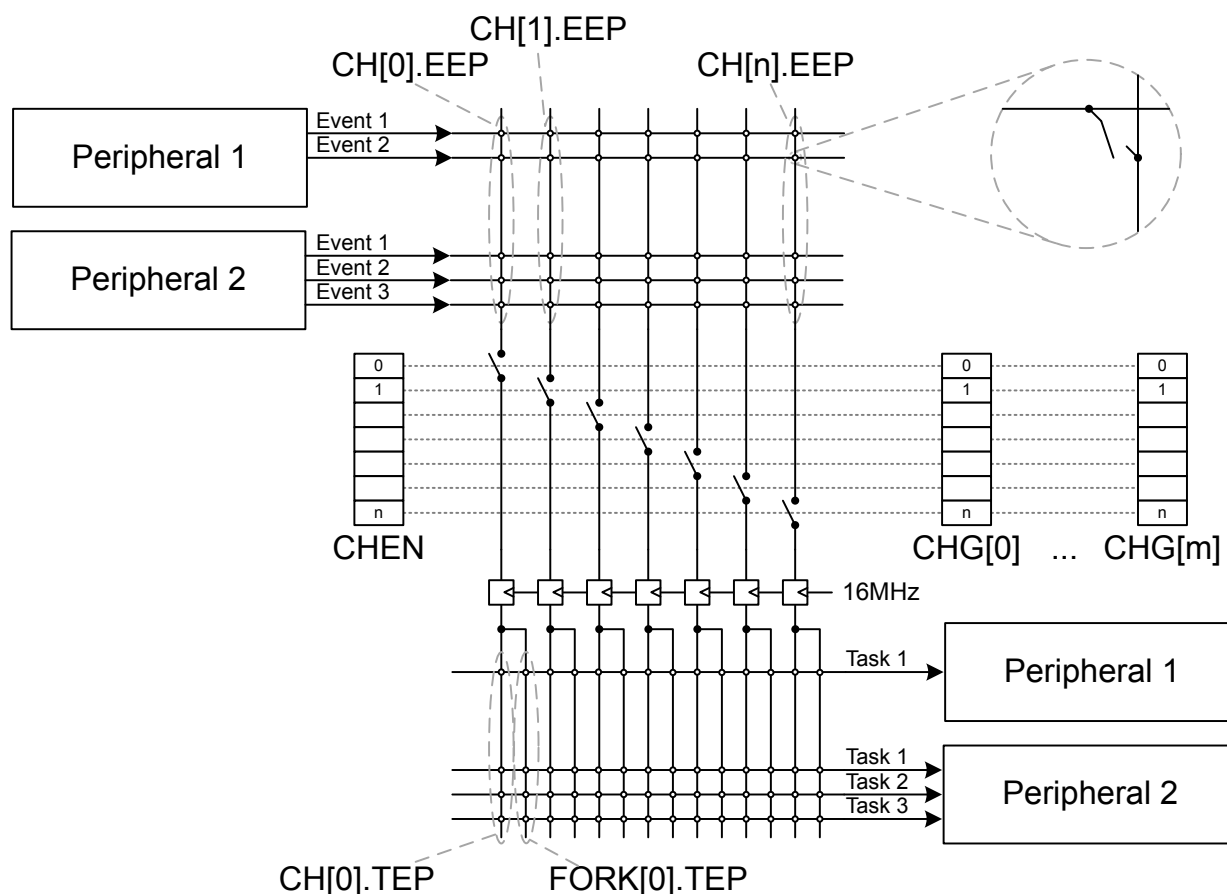


Figure 72: PPI block diagram

The PPI system has, in addition to the fully programmable peripheral interconnections, a set of channels where the event end point (EEP) and task end points (TEP) are fixed in hardware. These fixed channels can be individually enabled, disabled, or added to PPI channel groups (see CHG[n] registers), in the same way as ordinary PPI channels.

Instance	Channel	Number of channels
PPI	0-19	20
PPI (fixed)	20-31	12

Table 62: Configurable and fixed PPI channels

The PPI provides a mechanism to automatically trigger a task in one peripheral as a result of an event occurring in another peripheral. A task is connected to an event through a PPI channel. The PPI channel is composed of three end point registers, one EEP, and two TEPs. A peripheral task is connected to a TEP using the address of the task register associated with the task. Similarly, a peripheral event is connected to an EEP using the address of the event register associated with the event.

On each PPI channel, the signals are synchronized to the 16 MHz clock to avoid any internal violation of setup and hold timings. As a consequence, events that are synchronous to the 16 MHz clock will be delayed by one clock period, while other asynchronous events will be delayed by up to one 16 MHz clock period.

Note: Shortcuts (as defined in the SHORTS register in each peripheral) are not affected by this 16 MHz synchronization, and are therefore not delayed.

Each TEP implements a fork mechanism that enables a second task to be triggered at the same time as the task specified in the TEP is triggered. This second task is configured in the task end point register in the FORK registers groups, e.g. FORK.TEP[0] is associated with PPI channel CH[0].

There are two ways of enabling and disabling PPI channels:

- Enable or disable PPI channels individually using the CHEN, CHENSET, and CHENCLR registers.
- Enable or disable PPI channels in PPI channel groups through the groups' ENABLE and DISABLE tasks. Prior to these tasks being triggered, the PPI channel group must be configured to define which PPI channels belong to which groups.

Note: When a channel belongs to two groups m and n, and the tasks CHG[m].EN and CHG[n].DIS occur simultaneously (m and n can be equal or different), the CHG[m].EN on that channel has priority.

PPI tasks (for example, CHG[0].EN) can be triggered through the PPI like any other task, which means they can be hooked to a PPI channel as a TEP. One event can trigger multiple tasks by using multiple channels and one task can be triggered by multiple events in the same way.

6.15.1 Pre-programmed channels

Some of the PPI channels are pre-programmed. These channels cannot be configured by the CPU, but can be added to groups and enabled and disabled like the general purpose PPI channels. The FORK TEP for these channels are still programmable and can be used by the application.

For a list of pre-programmed PPI channels, see the following table.

Channel	EEP	TEP
20	TIMER0->EVENTS_COMPARE[0]	RADIO->TASKS_TXEN
21	TIMER0->EVENTS_COMPARE[0]	RADIO->TASKS_RXEN
22	TIMER0->EVENTS_COMPARE[1]	RADIO->TASKS_DISABLE
23	RADIO->EVENTS_BCMATCH	AAR->TASKS_START
24	RADIO->EVENTS_READY	CCM->TASKS_KSGEN
25	RADIO->EVENTS_ADDRESS	CCM->TASKS_CRYPT
26	RADIO->EVENTS_ADDRESS	TIMER0->TASKS_CAPTURE[1]
27	RADIO->EVENTS_END	TIMER0->TASKS_CAPTURE[2]
28	RTC0->EVENTS_COMPARE[0]	RADIO->TASKS_TXEN
29	RTC0->EVENTS_COMPARE[0]	RADIO->TASKS_RXEN
30	RTC0->EVENTS_COMPARE[0]	TIMER0->TASKS_CLEAR
31	RTC0->EVENTS_COMPARE[0]	TIMER0->TASKS_START

Table 63: Pre-programmed channels

6.15.2 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4001F000	PPI	PPI	Programmable peripheral interconnect	

Table 64: Instances

Register	Offset	Description
TASKS_CHG[0].EN	0x000	Enable channel group 0
TASKS_CHG[0].DIS	0x004	Disable channel group 0
TASKS_CHG[1].EN	0x008	Enable channel group 1
TASKS_CHG[1].DIS	0x00C	Disable channel group 1

Register	Offset	Description
TASKS_CHG[2].EN	0x010	Enable channel group 2
TASKS_CHG[2].DIS	0x014	Disable channel group 2
TASKS_CHG[3].EN	0x018	Enable channel group 3
TASKS_CHG[3].DIS	0x01C	Disable channel group 3
TASKS_CHG[4].EN	0x020	Enable channel group 4
TASKS_CHG[4].DIS	0x024	Disable channel group 4
TASKS_CHG[5].EN	0x028	Enable channel group 5
TASKS_CHG[5].DIS	0x02C	Disable channel group 5
CHEN	0x500	Channel enable register
CHENSET	0x504	Channel enable set register
CHENCLR	0x508	Channel enable clear register
CH[0].EEP	0x510	Channel 0 event endpoint
CH[0].TEP	0x514	Channel 0 task endpoint
CH[1].EEP	0x518	Channel 1 event endpoint
CH[1].TEP	0x51C	Channel 1 task endpoint
CH[2].EEP	0x520	Channel 2 event endpoint
CH[2].TEP	0x524	Channel 2 task endpoint
CH[3].EEP	0x528	Channel 3 event endpoint
CH[3].TEP	0x52C	Channel 3 task endpoint
CH[4].EEP	0x530	Channel 4 event endpoint
CH[4].TEP	0x534	Channel 4 task endpoint
CH[5].EEP	0x538	Channel 5 event endpoint
CH[5].TEP	0x53C	Channel 5 task endpoint
CH[6].EEP	0x540	Channel 6 event endpoint
CH[6].TEP	0x544	Channel 6 task endpoint
CH[7].EEP	0x548	Channel 7 event endpoint
CH[7].TEP	0x54C	Channel 7 task endpoint
CH[8].EEP	0x550	Channel 8 event endpoint
CH[8].TEP	0x554	Channel 8 task endpoint
CH[9].EEP	0x558	Channel 9 event endpoint
CH[9].TEP	0x55C	Channel 9 task endpoint
CH[10].EEP	0x560	Channel 10 event endpoint
CH[10].TEP	0x564	Channel 10 task endpoint
CH[11].EEP	0x568	Channel 11 event endpoint
CH[11].TEP	0x56C	Channel 11 task endpoint
CH[12].EEP	0x570	Channel 12 event endpoint
CH[12].TEP	0x574	Channel 12 task endpoint
CH[13].EEP	0x578	Channel 13 event endpoint
CH[13].TEP	0x57C	Channel 13 task endpoint
CH[14].EEP	0x580	Channel 14 event endpoint
CH[14].TEP	0x584	Channel 14 task endpoint
CH[15].EEP	0x588	Channel 15 event endpoint
CH[15].TEP	0x58C	Channel 15 task endpoint
CH[16].EEP	0x590	Channel 16 event endpoint
CH[16].TEP	0x594	Channel 16 task endpoint
CH[17].EEP	0x598	Channel 17 event endpoint
CH[17].TEP	0x59C	Channel 17 task endpoint
CH[18].EEP	0x5A0	Channel 18 event endpoint
CH[18].TEP	0x5A4	Channel 18 task endpoint
CH[19].EEP	0x5A8	Channel 19 event endpoint
CH[19].TEP	0x5AC	Channel 19 task endpoint
CHG[0]	0x800	Channel group 0
CHG[1]	0x804	Channel group 1

Register	Offset	Description
CHG[2]	0x808	Channel group 2
CHG[3]	0x80C	Channel group 3
CHG[4]	0x810	Channel group 4
CHG[5]	0x814	Channel group 5
FORK[0].TEP	0x910	Channel 0 task endpoint
FORK[1].TEP	0x914	Channel 1 task endpoint
FORK[2].TEP	0x918	Channel 2 task endpoint
FORK[3].TEP	0x91C	Channel 3 task endpoint
FORK[4].TEP	0x920	Channel 4 task endpoint
FORK[5].TEP	0x924	Channel 5 task endpoint
FORK[6].TEP	0x928	Channel 6 task endpoint
FORK[7].TEP	0x92C	Channel 7 task endpoint
FORK[8].TEP	0x930	Channel 8 task endpoint
FORK[9].TEP	0x934	Channel 9 task endpoint
FORK[10].TEP	0x938	Channel 10 task endpoint
FORK[11].TEP	0x93C	Channel 11 task endpoint
FORK[12].TEP	0x940	Channel 12 task endpoint
FORK[13].TEP	0x944	Channel 13 task endpoint
FORK[14].TEP	0x948	Channel 14 task endpoint
FORK[15].TEP	0x94C	Channel 15 task endpoint
FORK[16].TEP	0x950	Channel 16 task endpoint
FORK[17].TEP	0x954	Channel 17 task endpoint
FORK[18].TEP	0x958	Channel 18 task endpoint
FORK[19].TEP	0x95C	Channel 19 task endpoint
FORK[20].TEP	0x960	Channel 20 task endpoint
FORK[21].TEP	0x964	Channel 21 task endpoint
FORK[22].TEP	0x968	Channel 22 task endpoint
FORK[23].TEP	0x96C	Channel 23 task endpoint
FORK[24].TEP	0x970	Channel 24 task endpoint
FORK[25].TEP	0x974	Channel 25 task endpoint
FORK[26].TEP	0x978	Channel 26 task endpoint
FORK[27].TEP	0x97C	Channel 27 task endpoint
FORK[28].TEP	0x980	Channel 28 task endpoint
FORK[29].TEP	0x984	Channel 29 task endpoint
FORK[30].TEP	0x988	Channel 30 task endpoint
FORK[31].TEP	0x98C	Channel 31 task endpoint

Table 65: Register overview

6.15.2.1 TASKS_CHG[n].EN (n=0..5)

Address offset: 0x000 + (n × 0x8)

Enable channel group n

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																			
ID		A																																			
Reset 0x00000000		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																			
ID	Acce Field	Value ID	Value	Description																																	
A	W	EN		Enable channel group n																																	
		Trigger	1	Trigger task																																	

6.15.2.2 TASKS_CHG[n].DIS (n=0..5)

Address offset: 0x004 + (n × 0x8)

Disable channel group n

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	W DIS			Disable channel group n																														
		Trigger	1	Trigger task																														

6.15.2.3 CHEN

Address offset: 0x500

Channel enable register

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A-T	RW	CH[i] (i=0..19)				Enable or disable channel i																												
			Disabled	0	Disable channel																													
			Enabled	1	Enable channel																													
U-f	RW	CH[i] (i=20..31)				Enable or disable channel i																												
			Disabled	0	Disable channel																													
			Enabled	1	Enable channel																													

6.15.2.4 CHENSET

Address offset: 0x504

Channel enable set register

Read: reads value of CH{i} field in CHEN register.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A-T	RW	CH[i] (i=0..19)				Channel i enable set register. Writing '0' has no effect.																												
			Disabled	0	Read: channel disabled																													
			Enabled	1	Read: channel enabled																													
			Set	1	Write: Enable channel																													
U-f	RW	CH[i] (i=20..31)				Channel i enable set register. Writing '0' has no effect.																												
			Disabled	0	Read: channel disabled																													
			Enabled	1	Read: channel enabled																													
			Set	1	Write: Enable channel																													

6.15.2.5 CHENCLR

Address offset: 0x508

Channel enable clear register

Read: reads value of CH{i} field in CHEN register.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			f e d c b a Z Y X W V U T S R Q P O N M L K J I H G F E D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A-T	RW	CH[i] (i=0..19)		Channel i enable clear register. Writing '0' has no effect.																														
		Disabled	0	Read: channel disabled																														
		Enabled	1	Read: channel enabled																														
		Clear	1	Write: disable channel																														
U-f	RW	CH[i] (i=20..31)		Channel i enable clear register. Writing '0' has no effect.																														
		Disabled	0	Read: channel disabled																														
		Enabled	1	Read: channel enabled																														
		Clear	1	Write: disable channel																														

6.15.2.6 CH[n].EEP (n=0..19)

Address offset: 0x510 + (n × 0x8)

Channel n event endpoint

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	RW	EEP		Pointer to event register. Accepts only addresses to registers from the Event group.																															

6.15.2.7 CH[n].TEP (n=0..19)

Address offset: 0x514 + (n × 0x8)

Channel n task endpoint

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																																			
ID				A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A																																			
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																			
ID	Acce Field	Value ID	Value	Description																																			
A	RW	TEP		Pointer to task register. Accepts only addresses to registers from the Task group.																																			

6.15.2.8 CHG[n] (n=0..5)

Address offset: 0x800 + (n × 0x4)

Channel group n

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			f	e	d	c	b	a	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce	Field	Value		ID	Value		Description																										
A-T	RW	CH[i] (i=0..19)					Include or exclude channel i																											
			Excluded		0		Exclude																											
			Included		1		Include																											
U-f	RW	CH[i] (i=20..31)					Include or exclude channel i																											
			Excluded		0		Exclude																											
			Included		1		Include																											

6.15.2.9 FORK[n].TEP (n=0..19, 20..31)

Address offset: $0x910 + (n \times 0x4)$

Channel n task endpoint

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce	Field	Value			ID	Value			Description																								
A	RW	TEP								Pointer to task register																								

6.16 PWM — Pulse width modulation

The pulse with modulation (PWM) module enables the generation of pulse width modulated signals on GPIO. The module implements an up or up-and-down counter with four PWM channels that drive assigned GPIOs.

The following are the main features of a PWM module:

- Programmable PWM frequency
- Up to four PWM channels with individual polarity and duty cycle values
- Edge or center-aligned pulses across PWM channels
- Multiple duty cycle arrays (sequences) defined in RAM
- Autonomous and glitch-free update of duty cycle values directly from memory through EasyDMA (no CPU involvement)
- Change of polarity, duty cycle, and base frequency possibly on every PWM period
- RAM sequences can be repeated or connected into loops

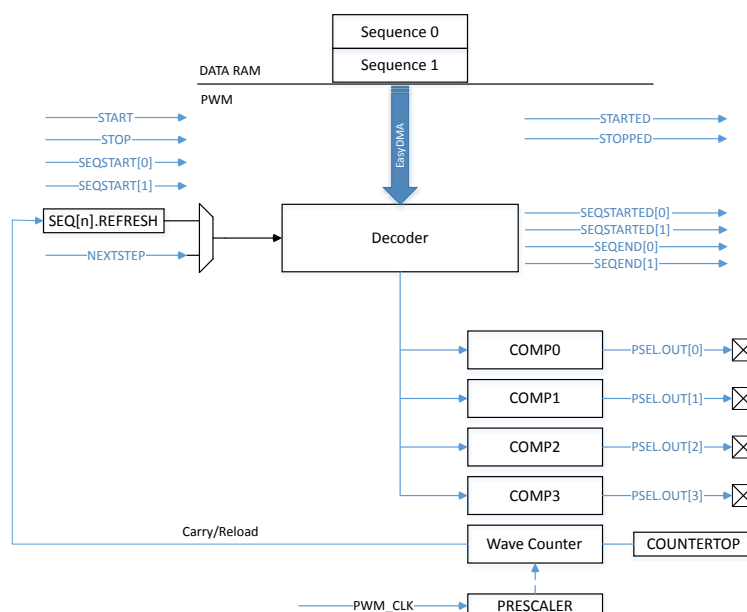


Figure 73: PWM module

6.16.1 Wave counter

The wave counter is responsible for generating the pulses at a duty cycle that depends on the compare values, and at a frequency that depends on COUNTERTOP.

There is one common 15-bit counter with four compare channels. Thus, all four channels will share the same period (PWM frequency), but can have individual duty cycle and polarity. The polarity is set by a value read from RAM (see figure [Decoder memory access modes](#) on page 246). Whether the counter counts up, or up and down, is controlled by the MODE register.

The timer top value is controlled by the COUNTERTOP register. This register value, in conjunction with the selected PRESCALER of the PWM_CLK, will result in a given PWM period. A COUNTERTOP value smaller than the compare setting will result in a state where no PWM edges are generated. OUT[n] is held high, given that the polarity is set to FallingEdge. All compare registers are internal and can only be configured through decoder presented later. COUNTERTOP can be safely written at any time.

Sampling follows the START task. If DECODER.LOAD=WaveForm, the register value is ignored and taken from RAM instead (see section [Decoder with EasyDMA](#) on page 246 for more details). If DECODER.LOAD is anything else than the WaveForm, it is sampled following a STARTSEQ[n] task and when loading a new value from RAM during a sequence playback.

The following figure shows the counter operating in up mode (MODE=PWM_MODE_Up), with three PWM channels with the same frequency but different duty cycle:

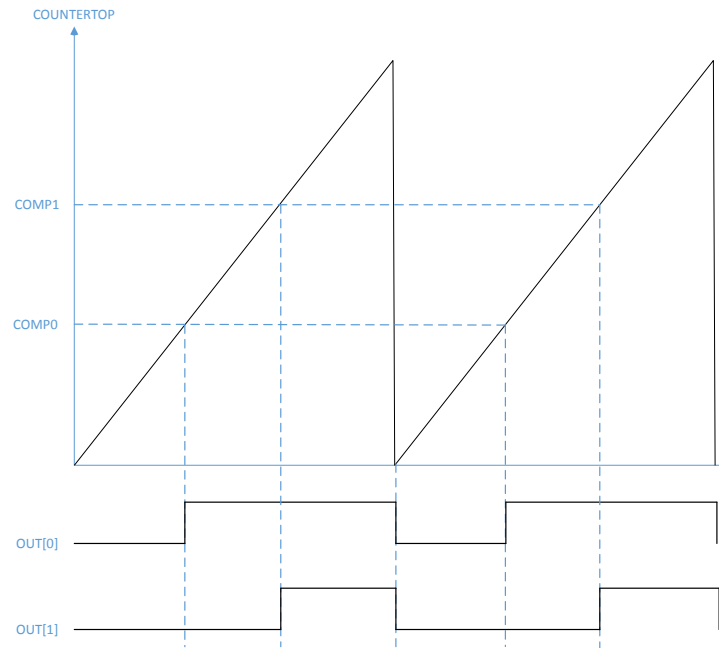


Figure 74: PWM counter in up mode example - FallingEdge polarity

The counter is automatically reset to zero when COUNTERTOP is reached and OUT[n] will invert. OUT[n] is held low if the compare value is 0 and held high if set to COUNTERTOP, given that the polarity is set to FallingEdge. Counter running in up mode results in pulse widths that are edge-aligned. The following is the code for the counter in up mode example:

```
uint16_t pwm_seq[4] = {PWM_CH0_DUTY, PWM_CH1_DUTY, PWM_CH2_DUTY, PWM_CH3_DUTY};
NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
    (PWM_PSEL_OUT_CONNECT_Connected <<
        PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->PSEL.OUT[1] = (second_pin << PWM_PSEL_OUT_PIN_Pos) |
    (PWM_PSEL_OUT_CONNECT_Connected <<
        PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE = (PWM_MODE_UPDOWN_Up << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER = (PWM_PRESCALER_PRESCALER_DIV_1 <<
    PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP = (PWM_LOOP_CNT_Disabled << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER = (PWM_DECODER_LOAD_Individual << PWM_DECODER_LOAD_Pos) |
    (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR = ((uint32_t) (pwm_seq) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT = ((sizeof(pwm_seq) / sizeof(uint16_t)) <<
    PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;
```

When the counter is running in up mode, the following formula can be used to compute the PWM period and the step size:

PWM period: $T_{PWM(Up)} = T_{PWM_CLK} * COUNTERTOP$

Step width/Resolution: $T_{\text{steps}} = T_{\text{PWM_CLK}}$

The following figure shows the counter operating in up-and-down mode (MODE=PWM_MODE_UpAndDown), with two PWM channels with the same frequency but different duty cycle and output polarity:

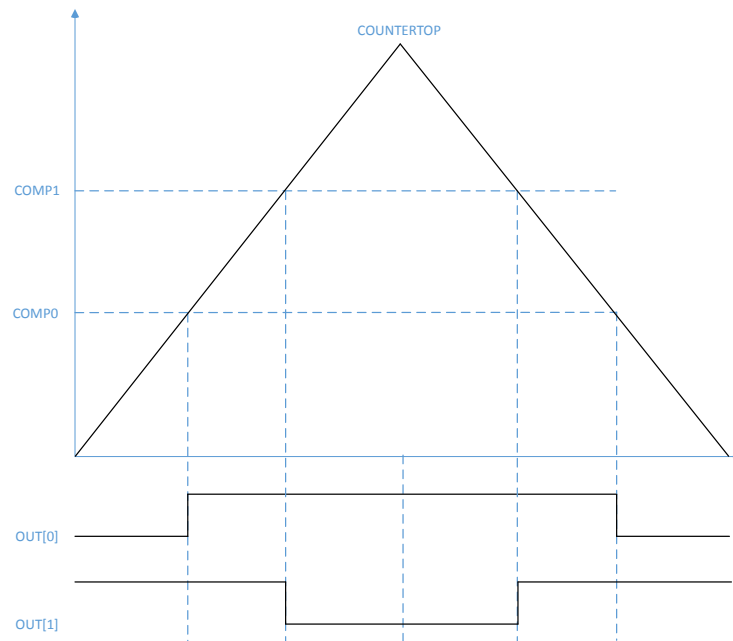


Figure 75: PWM counter in up-and-down mode example

The counter starts decrementing to zero when COUNTERTOP is reached and will invert the OUT[n] when compare value is hit for the second time. This results in a set of pulses that are center-aligned. The following is the code for the counter in up-and-down mode example:

```
uint16_t pwm_seq[4] = {PWM_CH0_DUTY, PWM_CH1_DUTY, PWM_CH2_DUTY, PWM_CH3_DUTY};
NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
    (PWM_PSEL_OUT_CONNECT_Connected <<
        PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->PSEL.OUT[1] = (second_pin << PWM_PSEL_OUT_PIN_Pos) |
    (PWM_PSEL_OUT_CONNECT_Connected <<
        PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE = (PWM_MODE_UPDOWN_UpAndDown << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER = (PWM_PRESCALER_PRESCALER_DIV_1 <<
    PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP = (PWM_LOOP_CNT_Disabled << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER = (PWM_DECODER_LOAD_Individual << PWM_DECODER_LOAD_Pos) |
    (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR = ((uint32_t) (pwm_seq) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT = ((sizeof(pwm_seq) / sizeof(uint16_t)) <<
    PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;
```

When the counter is running in up-and-down mode, the following formula can be used to compute the PWM period and the step size:

$$T_{\text{PWM(Up And Down)}} = T_{\text{PWM_CLK}} * 2 * \text{COUNTERTOP}$$

$$\text{Step width/Resolution: } T_{\text{steps}} = T_{\text{PWM_CLK}} * 2$$

6.16.2 Decoder with EasyDMA

The decoder uses EasyDMA to take PWM parameters stored in RAM and update the internal compare registers of the wave counter, based on the mode of operation.

PWM parameters are organized into a sequence containing at least one half word (16 bit). Its most significant bit[15] denotes the polarity of the OUT[n] while bit[14:0] is the 15-bit compare value.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
Id				B A A A A A A A A A A A A A A A A A A A A A A A A A A A A A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
Id	RW	Field	Value	Id	Value	Description																													
A	RW	COMPARE				Duty cycle setting - value loaded to internal compare register																													
B	RW	POLARITY				Edge polarity of GPIO.																													
		RisingEdge	0			First edge within the PWM period is rising																													
		FallingEdge	1			First edge within the PWM period is falling																													

The DECODER register controls how the RAM content is interpreted and loaded into the internal compare registers. The LOAD field controls if the RAM values are loaded to all compare channels, or to update a group or all channels with individual values. The following figure illustrates how parameters stored in RAM are organized and routed to various compare channels in different modes:

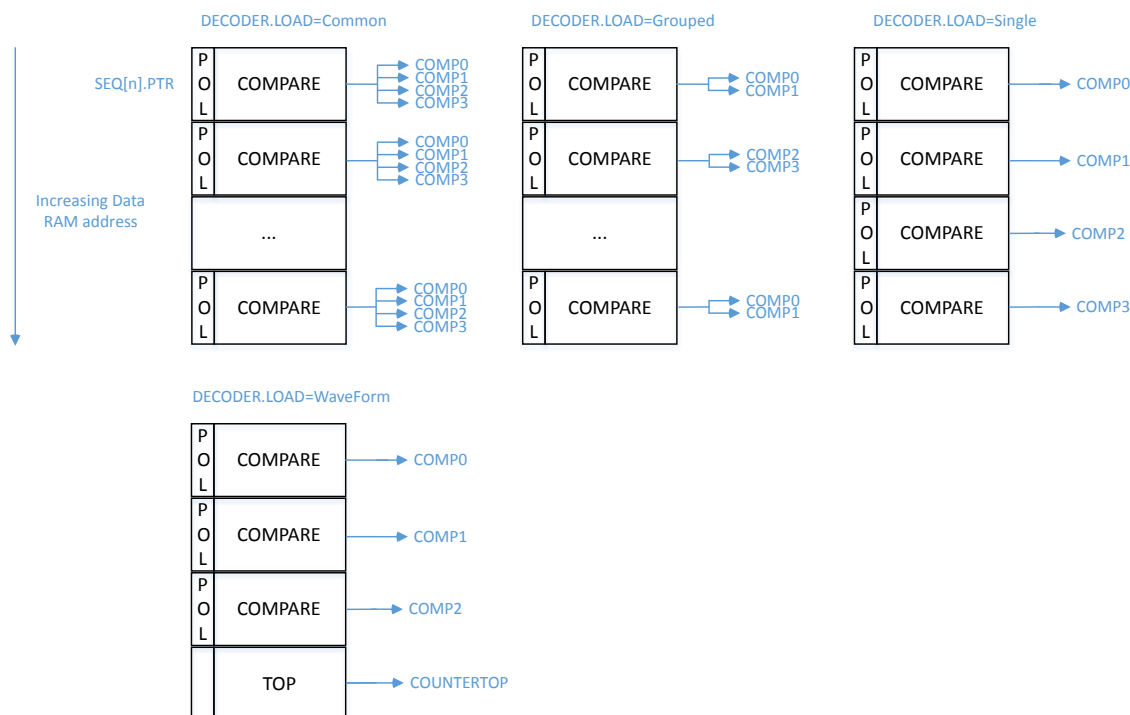


Figure 76: Decoder memory access modes

A special mode of operation is available when DECODER.LOAD is set to WaveForm. In this mode, up to three PWM channels can be enabled - OUT[0] to OUT[2]. In RAM, four values are loaded at a time: the first, second and third location are used to load the values, and the fourth RAM location is used to load

the COUNTERTOP register. This way one can have up to three PWM channels with a frequency base that changes on a per PWM period basis. This mode of operation is useful for arbitrary wave form generation in applications, such as LED lighting.

The register SEQ[n].REFRESH=N (one per sequence n=0 or 1) will instruct a new RAM stored pulse width value on every (N+1)th PWM period. Setting the register to zero will result in a new duty cycle update every PWM period, as long as the minimum PWM period is observed.

Note that registers SEQ[n].REFRESH and SEQ[n].ENDDELAY are ignored when DECODER.MODE=NextStep. The next value is loaded upon every received NEXTSTEP task.

SEQ[n].PTR is the pointer used to fetch COMPARE values from RAM. If the SEQ[n].PTR is not pointing to a RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 19 for more information about the different memory regions. After the SEQ[n].PTR is set to the desired RAM location, the SEQ[n].CNT register must be set to number of 16-bit half words in the sequence. It is important to observe that the Grouped mode requires one half word per group, while the Single mode requires one half word per channel, thus increasing the RAM size occupation. If PWM generation is not running when the SEQSTART[n] task is triggered, the task will load the first value from RAM and then start the PWM generation. A SEQSTARTED[n] event is generated as soon as the EasyDMA has read the first PWM parameter from RAM and the wave counter has started executing it. When LOOP.CNT=0, sequence n=0 or 1 is played back once. After the last value in the sequence has been loaded and started executing, a SEQEND[n] event is generated. The PWM generation will then continue with the last loaded value. The following figure illustrates an example of such simple playback:

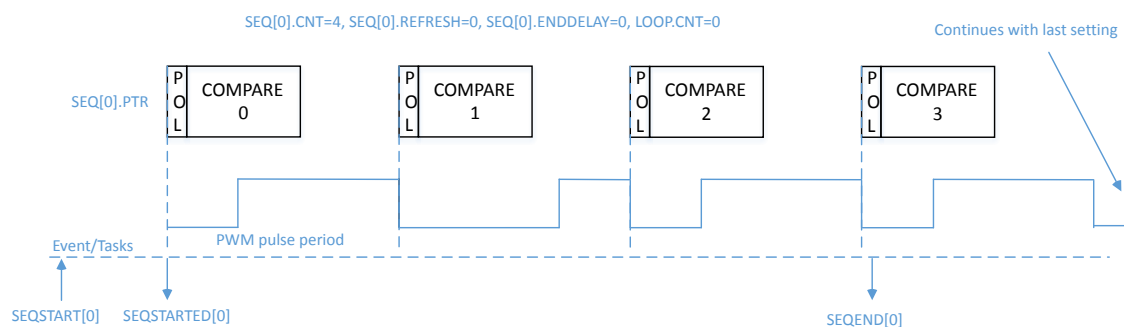


Figure 77: Simple sequence example

Figure depicts the source code used for configuration and timing details in a sequence where only sequence 0 is used and only run once with a new PWM duty cycle for each period.

```

NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                          PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE      = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE        = (PWM_MODE_UPDOWN_Up << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER   = (PWM_PRESCALER_PRESCALER_DIV_1 <<
                          PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP  = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP        = (PWM_LOOP_CNT_Disabled << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER     = (PWM_DECODER_LOAD_Common << PWM_DECODER_LOAD_Pos) |
                        (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR   = ((uint32_t)(seq0_ram) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT   = ((sizeof(seq0_ram) / sizeof(uint16_t)) <<
                          PWM_SEQ_CNT_CNT_Pos);
NRF_PWM0->SEQ[0].REFRESH = 0;
NRF_PWM0->SEQ[0].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;

```

To completely stop the PWM generation and force the associated pins to a defined state, a STOP task can be triggered at any time. A STOPPED event is generated when the PWM generation has stopped at the end of currently running PWM period, and the pins go into their idle state as defined in GPIO OUT register. PWM generation can then only be restarted through a SEQSTART[n] task. SEQSTART[n] will resume PWM generation after having loaded the first value from the RAM buffer defined in the SEQ[n].PTR register.

The table below indicates when specific registers get sampled by the hardware. Care should be taken when updating these registers to avoid that values are applied earlier than expected.

Register	Taken into account by hardware	Recommended (safe) update
SEQ[n].PTR	When sending the SEQSTART[n] task	After having received the SEQSTARTED[n] event
SEQ[n].CNT	When sending the SEQSTART[n] task	After having received the SEQSTARTED[n] event
SEQ[0].ENDDELAY	When sending the SEQSTART[0] task	Before starting sequence [0] through a SEQSTART[0] task
	Every time a new value from sequence [0] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	When no more value from sequence [0] gets loaded from RAM (indicated by the SEQEND[0] event) At any time during sequence [1] (which starts when the SEQSTARTED[1] event is generated)
SEQ[1].ENDDELAY	When sending the SEQSTART[1] task	Before starting sequence [1] through a SEQSTART[1] task
	Every time a new value from sequence [1] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	When no more value from sequence [1] gets loaded from RAM (indicated by the SEQEND[1] event) At any time during sequence [0] (which starts when the SEQSTARTED[0] event is generated)
SEQ[0].REFRESH	When sending the SEQSTART[0] task	Before starting sequence [0] through a SEQSTART[0] task
	Every time a new value from sequence [0] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	At any time during sequence [1] (which starts when the SEQSTARTED[1] event is generated)
SEQ[1].REFRESH	When sending the SEQSTART[1] task	Before starting sequence [1] through a SEQSTART[1] task
	Every time a new value from sequence [1] has been loaded from RAM and gets applied to the Wave Counter (indicated by the PWMPERIODEND event)	At any time during sequence [0] (which starts when the SEQSTARTED[0] event is generated)
COUNTERTOP	In DECODER.LOAD=WaveForm: this register is ignored.	Before starting PWM generation through a SEQSTART[n] task
	In all other LOAD modes: at the end of current PWM period (indicated by the PWMPERIODEND event)	After a STOP task has been triggered, and the STOPPED event has been received.
MODE	Immediately	Before starting PWM generation through a SEQSTART[n] task
		After a STOP task has been triggered, and the STOPPED event has been received.
DECODER	Immediately	Before starting PWM generation through a SEQSTART[n] task
		After a STOP task has been triggered, and the STOPPED event has been received.
PRESCALER	Immediately	Before starting PWM generation through a SEQSTART[n] task
		After a STOP task has been triggered, and the STOPPED event has been received.
LOOP	Immediately	Before starting PWM generation through a SEQSTART[n] task
		After a STOP task has been triggered, and the STOPPED event has been received.
PSEL.OUT[n]	Immediately	Before enabling the PWM instance through the ENABLE register

Table 66: When to safely update PWM registers

Note: SEQ[n].REFRESH and SEQ[n].ENDDELAY are ignored at the end of a complex sequence, indicated by a LOOPSDONE event. The reason for this is that the last value loaded from RAM is maintained until further action from software (restarting a new sequence, or stopping PWM generation).

A more complex example, where LOOP.CNT>0, is shown in the following figure:

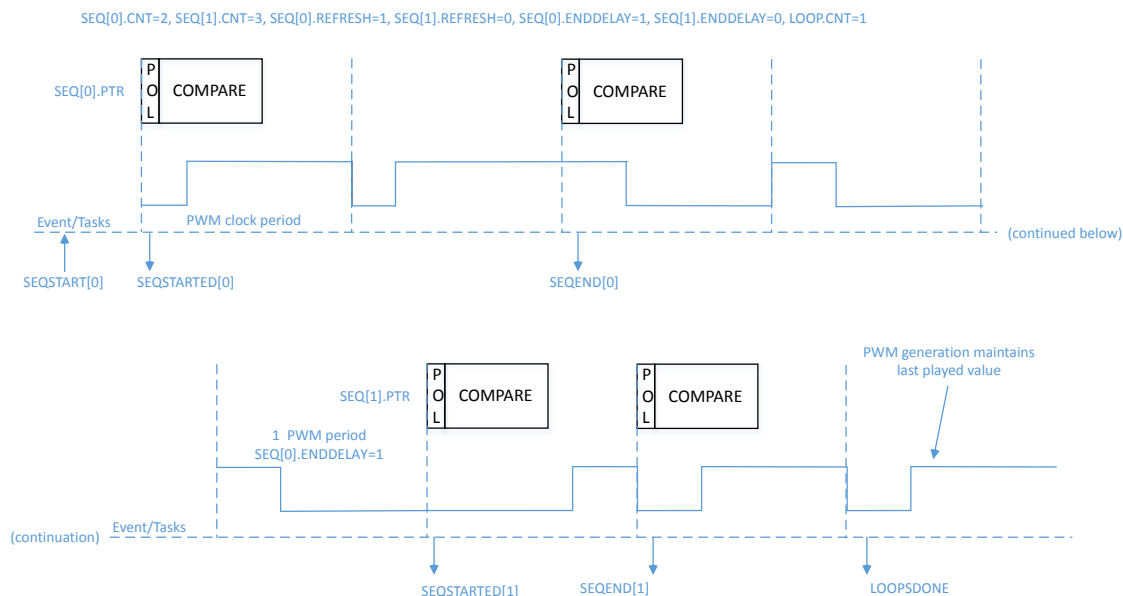


Figure 78: Example using two sequences

In this case, an automated playback takes place, consisting of SEQ[0], delay 0, SEQ[1], delay 1, then again SEQ[0], etc. The user can choose to start a complex playback with SEQ[0] or SEQ[1] through sending the SEQSTART[0] or SEQSTART[1] task. The complex playback always ends with delay 1.

The two sequences 0 and 1 are defined by the addresses of value tables in RAM (pointed to by SEQ[n].PTR) and the buffer size (SEQ[n].CNT). The rate at which a new value is loaded is defined individually for each sequence by SEQ[n].REFRESH. The chaining of sequence 1 following the sequence 0 is implicit, the LOOP.CNT register allows the chaining of sequence 1 to sequence 0 for a determined number of times. In other words, it allows to repeat a complex sequence a number of times in a fully automated way.

In the following code example, sequence 0 is defined with SEQ[0].REFRESH set to 1, meaning that a new PWM duty cycle is pushed every second PWM period. This complex sequence is started with the SEQSTART[0] task, so SEQ[0] is played first. Since SEQ[0].ENDDelay=1 there will be one PWM period delay between last period on sequence 0 and the first period on sequence 1. Since SEQ[1].ENDDelay=0 there is no delay 1, so SEQ[0] would be started immediately after the end of SEQ[1]. However, as LOOP.CNT is

1, the playback stops after having played SEQ[1] only once, and both SEQEND[1] and LOOPSDONE are generated (their order is not guaranteed in this case).

```

NRF_PWM0->PSEL.OUT[0] = (first_pin << PWM_PSEL_OUT_PIN_Pos) |
                        (PWM_PSEL_OUT_CONNECT_Connected <<
                         PWM_PSEL_OUT_CONNECT_Pos);
NRF_PWM0->ENABLE      = (PWM_ENABLE_ENABLE_Enabled << PWM_ENABLE_ENABLE_Pos);
NRF_PWM0->MODE        = (PWM_MODE_UPDOWN_Up << PWM_MODE_UPDOWN_Pos);
NRF_PWM0->PRESCALER   = (PWM_PRESCALER_PRESCALER_DIV_1 <<
                         PWM_PRESCALER_PRESCALER_Pos);
NRF_PWM0->COUNTERTOP  = (16000 << PWM_COUNTERTOP_COUNTERTOP_Pos); //1 msec
NRF_PWM0->LOOP        = (1 << PWM_LOOP_CNT_Pos);
NRF_PWM0->DECODER     = (PWM_DECODER_LOAD_Common << PWM_DECODER_LOAD_Pos) |
                        (PWM_DECODER_MODE_RefreshCount << PWM_DECODER_MODE_Pos);
NRF_PWM0->SEQ[0].PTR   = ((uint32_t)(seq0_ram) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[0].CNT   = ((sizeof(seq0_ram) / sizeof(uint16_t)) <<
                         PWM_SEQ_CNT_CNT_Pos);

NRF_PWM0->SEQ[0].REFRESH = 1;
NRF_PWM0->SEQ[0].ENDDELAY = 1;
NRF_PWM0->SEQ[1].PTR   = ((uint32_t)(seq1_ram) << PWM_SEQ_PTR_PTR_Pos);
NRF_PWM0->SEQ[1].CNT   = ((sizeof(seq1_ram) / sizeof(uint16_t)) <<
                         PWM_SEQ_CNT_CNT_Pos);

NRF_PWM0->SEQ[1].REFRESH = 0;
NRF_PWM0->SEQ[1].ENDDELAY = 0;
NRF_PWM0->TASKS_SEQSTART[0] = 1;

```

The decoder can also be configured to asynchronously load new PWM duty cycle. If the DECODER.MODE register is set to NextStep, then the NEXTSTEP task will cause an update of internal compare registers on the next PWM period.

The following figures provide an overview of each part of an arbitrary sequence, in various modes (LOOP.CNT=0 and LOOP.CNT>0). In particular, the following are represented:

- Initial and final duty cycle on the PWM output(s)
- Chaining of SEQ[0] and SEQ[1] if LOOP.CNT>0
- Influence of registers on the sequence
- Events generated during a sequence
- DMA activity (loading of next value and applying it to the output(s))

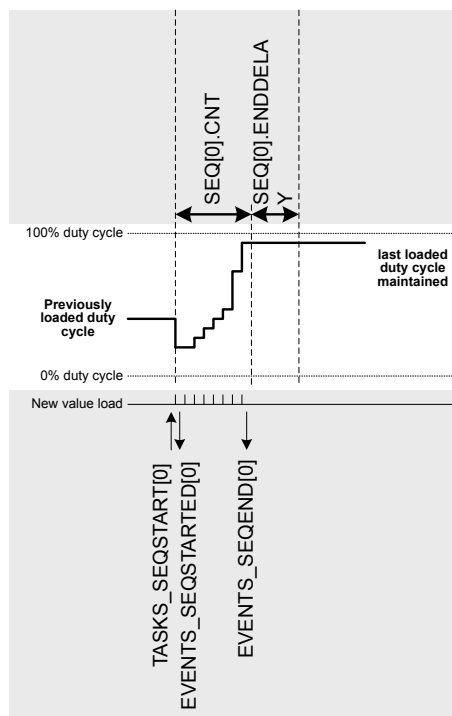


Figure 79: Single shot (LOOP.CNT=0)

Note: The single-shot example also applies to SEQ[1]. Only SEQ[0] is represented for simplicity.

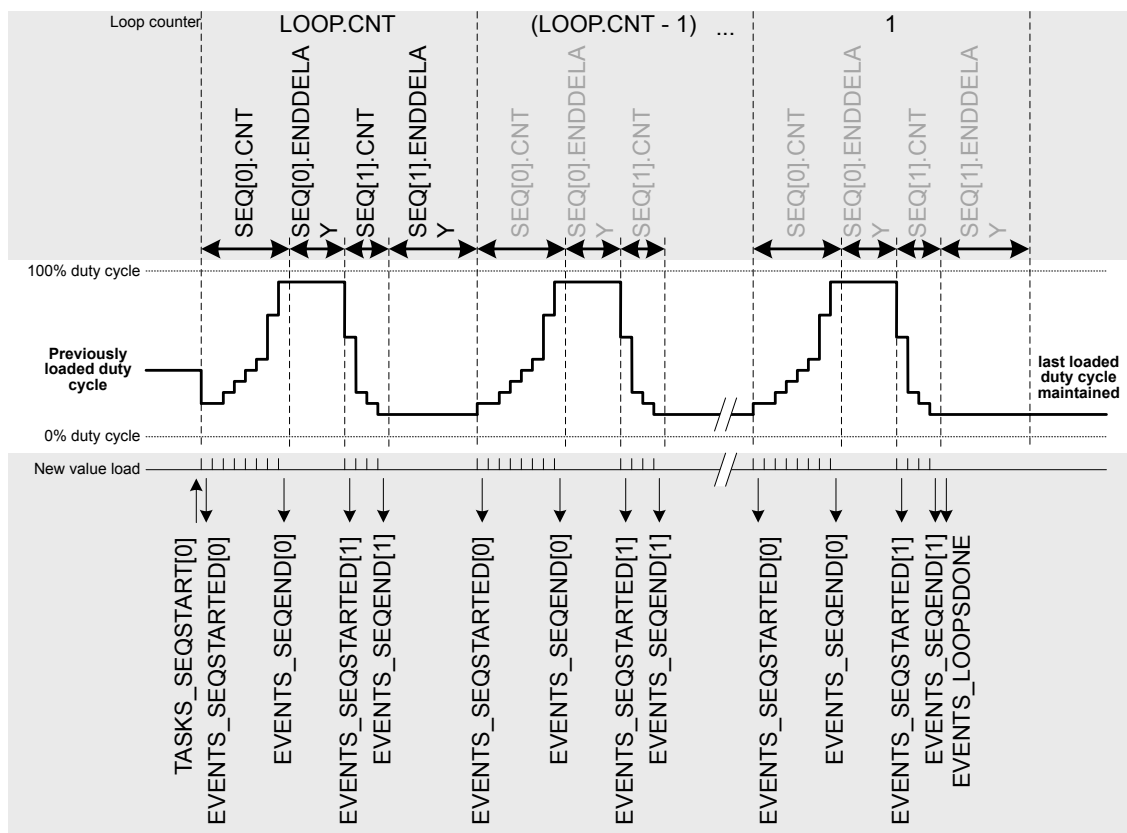


Figure 80: Complex sequence (LOOP.CNT>0) starting with SEQ[0]

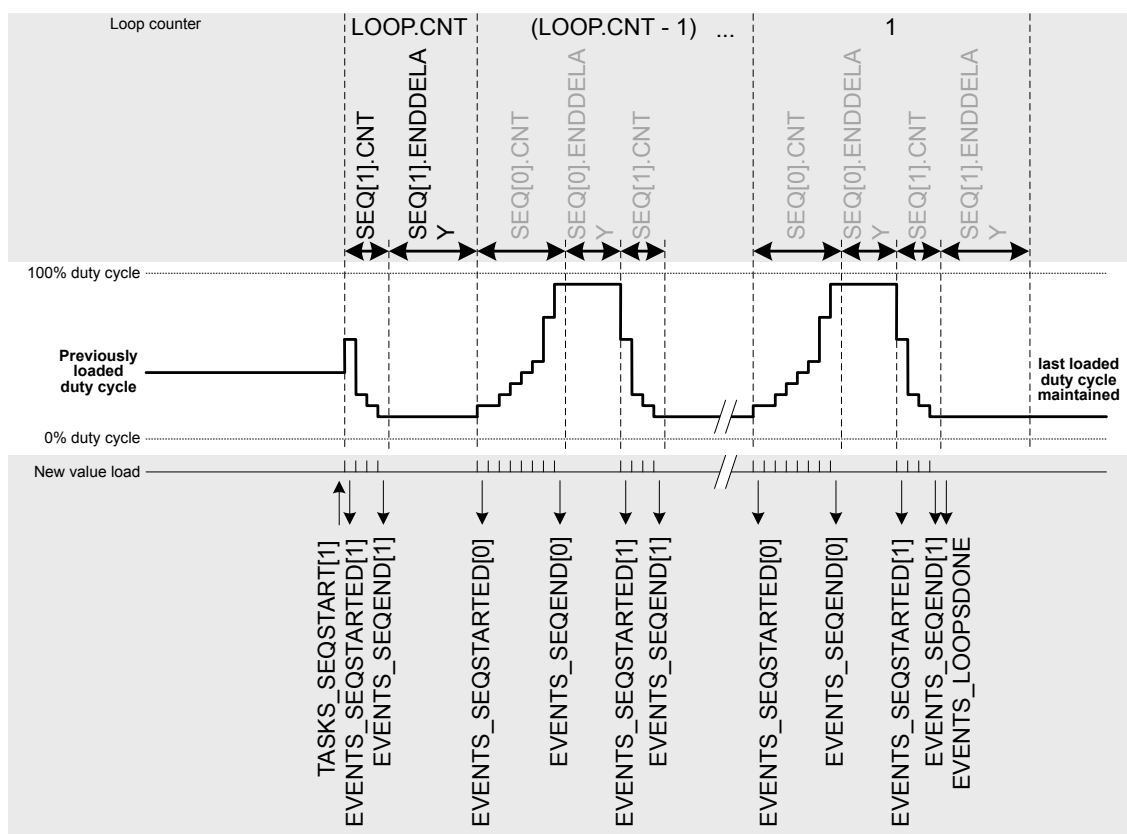


Figure 81: Complex sequence ($LOOP.CNT > 0$) starting with $SEQ[1]$

Note: If a sequence is in use in a simple or complex sequence, it must have a length of $SEQ[n].CNT > 0$.

6.16.3 Limitations

Previous compare value is repeated if the PWM period is shorter than the time it takes for the EasyDMA to retrieve from RAM and update the internal compare registers. This is to ensure a glitch-free operation even for very short PWM periods.

6.16.4 Pin configuration

The $OUT[n]$ ($n=0..3$) signals associated with each PWM channel are mapped to physical pins according to the configuration of $PSEL.OUT[n]$ registers. If $PSEL.OUT[n].CONNECT$ is set to Disconnected, the associated PWM module signal will not be connected to any physical pins.

The $PSEL.OUT[n]$ registers and their configurations are used as long as the PWM module is enabled and the PWM generation active (wave counter started). They are retained only as long as the device is in System ON mode (see section [POWER](#) for more information about power modes).

To ensure correct behavior in the PWM module, the pins that are used must be configured in the GPIO peripheral in the following way before the PWM module is enabled:

PWM signal	PWM pin	Direction	Output value	Comment
$OUT[n]$	As specified in $PSEL.OUT[n]$ ($n=0..3$)	Output	0	Idle state defined in GPIO OUT register

Table 67: Recommended GPIO configuration before starting PWM generation

The idle state of a pin is defined by the OUT register in the GPIO module, to ensure that the pins used by the PWM module are driven correctly. If PWM generation is stopped by triggering a STOP task, the PWM module itself is temporarily disabled or the device temporarily enters System OFF. This configuration must be retained in the GPIO for the selected pins (I/Os) for as long as the PWM module is supposed to be connected to an external PWM circuit.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

6.16.5 Registers

Base address	Peripheral	Instance	Description	Configuration
0x4001C000	PWM	PWM0	Pulse width modulation unit 0	
0x40021000	PWM	PWM1	Pulse width modulation unit 1	
0x40022000	PWM	PWM2	Pulse width modulation unit 2	
0x4002D000	PWM	PWM3	Pulse width modulation unit 3	

Table 68: Instances

Register	Offset	Description
TASKS_STOP	0x004	Stops PWM pulse generation on all channels at the end of current PWM period, and stops sequence playback
TASKS_SEQSTART[0]	0x008	Loads the first PWM value on all enabled channels from sequence 0, and starts playing that sequence at the rate defined in SEQ[0]REFRESH and/or DECODER.MODE. Causes PWM generation to start if not running.
TASKS_SEQSTART[1]	0x00C	Loads the first PWM value on all enabled channels from sequence 1, and starts playing that sequence at the rate defined in SEQ[1]REFRESH and/or DECODER.MODE. Causes PWM generation to start if not running.
TASKS_NEXTSTEP	0x010	Steps by one value in the current sequence on all enabled channels if DECODER.MODE=NextStep. Does not cause PWM generation to start if not running.
EVENTS_STOPPED	0x104	Response to STOP task, emitted when PWM pulses are no longer generated
EVENTS_SEQSTARTED[0]	0x108	First PWM period started on sequence 0
EVENTS_SEQSTARTED[1]	0x10C	First PWM period started on sequence 1
EVENTS_SEQEND[0]	0x110	Emitted at end of every sequence 0, when last value from RAM has been applied to wave counter
EVENTS_SEQEND[1]	0x114	Emitted at end of every sequence 1, when last value from RAM has been applied to wave counter
EVENTS_PWMPERIODEND	0x118	Emitted at the end of each PWM period
EVENTS_LOOPSDONE	0x11C	Concatenated sequences have been played the amount of times defined in LOOP.CNT
SHORTS	0x200	Shortcuts between local events and tasks
INTEN	0x300	Enable or disable interrupt
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ENABLE	0x500	PWM module enable register
MODE	0x504	Selects operating mode of the wave counter
COUNTERTOP	0x508	Value up to which the pulse generator counter counts
PRESCALER	0x50C	Configuration for PWM_CLK
DECODER	0x510	Configuration of the decoder
LOOP	0x514	Number of playbacks of a loop
SEQ[0].PTR	0x520	Beginning address in RAM of this sequence
SEQ[0].CNT	0x524	Number of values (duty cycles) in this sequence
SEQ[0].REFRESH	0x528	Number of additional PWM periods between samples loaded into compare register
SEQ[0].ENDDelay	0x52C	Time added after the sequence
SEQ[1].PTR	0x540	Beginning address in RAM of this sequence

Register	Offset	Description
SEQ[1].CNT	0x544	Number of values (duty cycles) in this sequence
SEQ[1].REFRESH	0x548	Number of additional PWM periods between samples loaded into compare register
SEQ[1].ENDDelay	0x54C	Time added after the sequence
PSEL.OUT[0]	0x560	Output pin select for PWM channel 0
PSEL.OUT[1]	0x564	Output pin select for PWM channel 1
PSEL.OUT[2]	0x568	Output pin select for PWM channel 2
PSEL.OUT[3]	0x56C	Output pin select for PWM channel 3

Table 69: Register overview

6.16.5.1 TASKS_STOP

Address offset: 0x004

Stops PWM pulse generation on all channels at the end of current PWM period, and stops sequence playback

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																													
A	W	TASKS_STOP				Stops PWM pulse generation on all channels at the end of current PWM period, and stops sequence playback																													
			Trigger	1		Trigger task																													

6.16.5.2 TASKS_SEQSTART[n] (n=0..1)

Address offset: 0x008 + (n × 0x4)

Loads the first PWM value on all enabled channels from sequence n, and starts playing that sequence at the rate defined in SEQ[n]REFRESH and/or DECODER.MODE. Causes PWM generation to start if not running.

Bit number					31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID					A																																
Reset 0x00000000					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ID	Acce	Field	Value	ID	Value	Description																															
A	W	TASKS_SEQSTART				Loads the first PWM value on all enabled channels from sequence n, and starts playing that sequence at the rate defined in SEQ[n]REFRESH and/or DECODER.MODE. Causes PWM generation to start if not running.																															
			Trigger	1		Trigger task																															

6.16.5.3 TASKS_NEXTSTEP

Address offset: 0x010

Steps by one value in the current sequence on all enabled channels if DECODER.MODE=NextStep. Does not cause PWM generation to start if not running.

6.16.5.10 INTEN

Address offset: 0x300

Enable or disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			H G F E D C B																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																													
B	RW	STOPPED			Enable or disable interrupt for event STOPPED																													
			Disabled	0	Disable																													
			Enabled	1	Enable																													
C-D	RW	SEQSTARTED[i] (i=0..1)			Enable or disable interrupt for event SEQSTARTED[i]																													
			Disabled	0	Disable																													
			Enabled	1	Enable																													
E-F	RW	SEQEND[i] (i=0..1)			Enable or disable interrupt for event SEQEND[i]																													
			Disabled	0	Disable																													
			Enabled	1	Enable																													
G	RW	PWMPERIODEND			Enable or disable interrupt for event PWMPERIODEND																													
			Disabled	0	Disable																													
			Enabled	1	Enable																													
H	RW	LOOPSDONE			Enable or disable interrupt for event LOOPSDONE																													
			Disabled	0	Disable																													
			Enabled	1	Enable																													

6.16.5.11 INTENSET

Address offset: 0x304

Enable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			H G F E D C B																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID	Value	Description																													
B	RW	STOPPED			Write '1' to enable interrupt for event STOPPED																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
C-D	RW	SEQSTARTED[i] (i=0..1)			Write '1' to enable interrupt for event SEQSTARTED[i]																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
E-F	RW	SEQEND[i] (i=0..1)			Write '1' to enable interrupt for event SEQEND[i]																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
G	RW	PWMPERIODEND			Write '1' to enable interrupt for event PWMPERIODEND																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													
			Enabled	1	Read: Enabled																													
H	RW	LOOPSDONE			Write '1' to enable interrupt for event LOOPSDONE																													
			Set	1	Enable																													
			Disabled	0	Read: Disabled																													

6.16.5.12 INTENCLR

Disable interrupt

6.16.5.13 ENABLE

PWM module enable register

6.16.5.14 MODE

Selects operating mode of the wave counter

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																														
ID			A																														
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																														
ID	Acce Field	Value ID	Value	Description																													
A	RW	UPDOWN		Selects up mode or up-and-down mode for the counter																													
		Up	0	Up counter, edge-aligned PWM duty cycle																													
		UpAndDown	1	Up and down counter, center-aligned PWM duty cycle																													

6.16.5.15 COUNTERTOP

Address offset: 0x508

Value up to which the pulse generator counter counts

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0														
ID																							A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x000003FF			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1												
ID	Acce Field	Value ID	Value		Description																																											
A	RW	COUNTERTOP	[3..32767]		Value up to which the pulse generator counter counts. This register is ignored when DECODER.MODE=WaveForm and only values from RAM are used.																																											

6.16.5.16 PRESCALER

Address offset: 0x50C

Configuration for PWM_CLK

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A A A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																													
A	RW	PRESCALER				Prescaler of PWM_CLK																													
		DIV_1	0	Divide by 1 (16 MHz)																															
		DIV_2	1	Divide by 2 (8 MHz)																															
		DIV_4	2	Divide by 4 (4 MHz)																															
		DIV_8	3	Divide by 8 (2 MHz)																															
		DIV_16	4	Divide by 16 (1 MHz)																															
		DIV_32	5	Divide by 32 (500 kHz)																															
		DIV_64	6	Divide by 64 (250 kHz)																															
		DIV_128	7	Divide by 128 (125 kHz)																															

6.16.5.17 DECODER

Address offset: 0x510

Configuration of the decoder

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			B																								A				A			
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A	RW	LOAD				How a sequence is read from RAM and spread to the compare register																												
			Common	0	1st half word (16-bit) used in all PWM channels 0..3																													
			Grouped	1	1st half word (16-bit) used in channel 0..1; 2nd word in channel 2..3																													
			Individual	2	1st half word (16-bit) in ch.0; 2nd in ch.1; ...; 4th in ch.3																													
			WaveForm	3	1st half word (16-bit) in ch.0; 2nd in ch.1; ...; 4th in COUNTERTOP																													
B	RW	MODE			Selects source for advancing the active sequence																													
			RefreshCount	0	SEQ[n].REFRESH is used to determine loading internal compare registers																													
			NextStep	1	NEXTSTEP task causes a new value to be loaded to internal compare registers																													

6.16.5.18 LOOP

Address offset: 0x514

Number of playbacks of a loop

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID																				A A A A A A A A A A A A A A A A															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field		Value ID	Value				Description																											
A	RW	CNT						Number of playbacks of pattern cycles																											
			Disabled	0				Looping disabled (stop at the end of the sequence)																											

6.16.5.19 SEQ[n].PTR (n=0..1)

Address offset: 0x520 + (n × 0x20)

Beginning address in RAM of this sequence

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value								Description																							
A	RW PTR										Beginning address in RAM of this sequence																							

Note: See the memory chapter for details about which memories are available for EasyDMA.

6.16.5.20 SEQ[n].CNT (n=0..1)

Address offset: 0x524 + (n × 0x20)

Number of values (duty cycles) in this sequence

Bit number		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																									A	A	A	A	A	A	A	A	A
Reset 0x00000000		0																															
ID	Acce Field	Value ID		Value		Description																											
A	RW CNT	Disabled		0		Number of values (duty cycles) in this sequence		Sequence is disabled, and shall not be started as it is empty																									

6.16.5.21 SEQ[n].REFRESH (n=0..1)

Address offset: 0x528 + (n × 0x20)

Number of additional PWM periods between samples loaded into compare register

Bit number		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																										A	A	A	A	A	A	A	A
Reset 0x00000001		0																															1
ID	Acce Field	Value ID		Value		Description																											
A	RW CNT	Continuous		0		Number of additional PWM periods between samples loaded into compare register (load every REFRESH.CNT+1 PWM periods)		Update every PWM period																									

6.16.5.22 SEQ[n].ENDDELAY (n=0..1)

Address offset: 0x52C + (n × 0x20)

Time added after the sequence

Bit number		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																										A	A	A	A	A	A	A	A
Reset 0x00000000		0																															
ID	Acce Field	Value ID		Value		Description																											
A	RW CNT					Time added after the sequence in PWM periods																											

6.16.5.23 PSEL.OUT[n] (n=0..3)

Address offset: 0x560 + (n × 0x4)

Output pin select for PWM channel n

Bit number		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID		C																									B				A		A
Reset 0xFFFFFFFF		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce Field	Value ID		Value		Description																											
A	RW PIN			[0..31]		Pin number																											
B	RW PORT			[0..1]		Port number																											
C	RW CONNECT					Connection																											
		Disconnected		1		Disconnect																											
		Connected		0		Connect																											

6.17 QDEC — Quadrature decoder

The Quadrature decoder (QDEC) provides buffered decoding of quadrature-encoded sensor signals. It is suitable for mechanical and optical sensors.

The sample period and accumulation are configurable to match application requirements. The QDEC provides the following:

- Digital waveform decoding from off-chip quadrature encoder
- Sample accumulation eliminating hard real-time requirements to be enforced on application
- Optional input de-bounce filters.
- Optional LED output signal for optical encoders

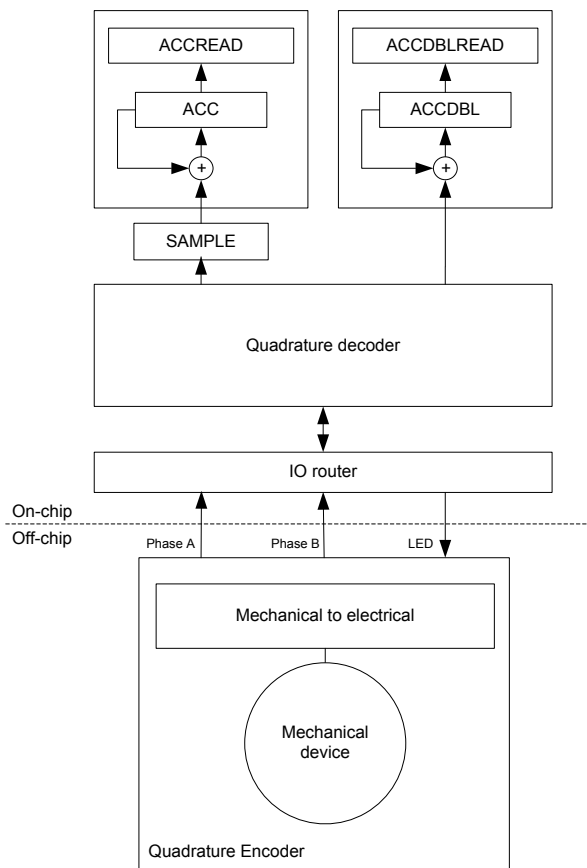


Figure 82: Quadrature decoder configuration

6.17.1 Sampling and decoding

The QDEC decodes the output from an incremental motion encoder by sampling the QDEC phase input pins (A and B).

The off-chip quadrature encoder is an incremental motion encoder outputting two waveforms, phase A and phase B. The two output waveforms are always 90 degrees out of phase, meaning that one always changes level before the other. The direction of movement is indicated by the waveform that changes level first. Invalid transitions may occur, meaning the two waveforms simultaneously switch. This may occur if the wheel rotates too fast relative to the sample rate set for the decoder.

The QDEC decodes the output from the off-chip encoder by sampling the QDEC phase input pins (A and B) at a fixed rate as specified in the **SAMPLEPER** register.

If the **SAMPLEPER** value needs to be changed, the QDEC shall be stopped using the **STOP** task. **SAMPLEPER** can be then changed upon receiving the **STOPPED** event, and QDEC can be restarted using the **START** task. Failing to do so may result in unpredictable behavior.

It is good practice to only change registers LEDPOL, REPORTPER, DBFEN, and LEDPRE when the QDEC is stopped.

When started, the decoder continuously samples the two input waveforms and decodes these by comparing the current sample pair (n) with the previous sample pair (n-1).

The decoding of the sample pairs is described in the table below.

Previous sample pair(n-1)		Current samples pair(n)		SAMPLE register	ACC operation	ACCDBL operation	Description
A	B	A	B				
0	0	0	0	0	No change	No change	No movement
0	0	0	1	1	Increment	No change	Movement in positive direction
0	0	1	0	-1	Decrement	No change	Movement in negative direction
0	0	1	1	2	No change	Increment	Error: Double transition
0	1	0	0	-1	Decrement	No change	Movement in negative direction
0	1	0	1	0	No change	No change	No movement
0	1	1	0	2	No change	Increment	Error: Double transition
0	1	1	1	1	Increment	No change	Movement in positive direction
1	0	0	0	1	Increment	No change	Movement in positive direction
1	0	0	1	2	No change	Increment	Error: Double transition
1	0	1	0	0	No change	No change	No movement
1	0	1	1	-1	Decrement	No change	Movement in negative direction
1	1	0	0	2	No change	Increment	Error: Double transition
1	1	0	1	-1	Decrement	No change	Movement in negative direction
1	1	1	0	1	Increment	No change	Movement in positive direction
1	1	1	1	0	No change	No change	No movement

Table 70: Sampled value encoding

6.17.2 LED output

The LED output follows the sample period. The LED is switched on for a set period before sampling and then switched off immediately after. The period the LED is switched on before sampling is given in the LEDPRE register.

The LED output pin polarity is specified in the LEDPOL register.

When using off-chip mechanical encoders not requiring an LED, the LED output can be disabled by writing value 'Disconnected' to the CONNECT field of the PSEL.LED register. In this case, the QDEC will not acquire access to a pin for the LED output.

6.17.3 Debounce filters

Each of the two-phase inputs have digital debounce filters.

When enabled through the DBFEN register, the filter inputs are sampled at a fixed 1 MHz frequency during the entire sample period (which is specified in the SAMPLEPER register). The filters require all of the samples within this sample period to equal before the input signal is accepted and transferred to the output of the filter.

As a result, only input signal with a steady state longer than twice the period specified in SAMPLEPER are guaranteed to pass through the filter. Any signal with a steady state shorter than SAMPLEPER will always be suppressed by the filter. It is assumed that the frequency during the debounce period never exceeds 500 kHz (as required by the Nyquist theorem when using a 1 MHz sample frequency).

The LED will always be ON when the debounce filters are enabled, as the inputs in this case will be sampled continuously.

When the debounce filters are enabled, displacements reported by the QDEC peripheral are delayed by one SAMPLEPER period.

6.17.4 Accumulators

The quadrature decoder contains two accumulator registers, ACC and ACCDBL. These registers accumulate valid motion sample values and the number of detected invalid samples (double transitions), respectively.

The ACC register accumulates all valid values (1/-1) written to the SAMPLE register. This can be useful for preventing hard real-time requirements from being enforced on the application. When using the ACC register, the application can fetch data when necessary instead of reading all SAMPLE register output. The ACC register holds the relative movement of the external mechanical device from the previous clearing of the ACC register. Sample values indicating a double transition (2) will not be accumulated in the ACC register.

An ACCOF event is generated if the ACC receives a SAMPLE value that would cause the register to overflow or underflow. Any SAMPLE value that would cause an ACC overflow or underflow will be discarded, but any samples that do not cause the ACC to overflow or underflow will still be accepted.

The accumulator ACCDBL accumulates the number of detected double transitions since the previous clearing of the ACCDBL register.

The ACC and ACCDBL registers can be cleared by the READCLRACC and subsequently read using the ACCREAD and ACCDBLREAD registers.

The ACC register can be separately cleared by the RDCLRACC and subsequently read using the ACCREAD registers.

The ACCDBL register can be separately cleared by the RDCLRDBL and subsequently read using the ACCDBLREAD registers.

The REPORTPER register allows automated capture of multiple samples before sending an event. When a non-null displacement is captured and accumulated, a REPORTRDY event is sent. When one or more double-displacements are captured and accumulated, a DBLRDY event is sent. The REPORTPER field in this register determines how many samples must be accumulated before the contents are evaluated and a REPORTRDY or DBLRDY event is sent.

Using the RDCLRACC task (manually sent upon receiving the event, or using the DBLRDY_RDCLRACC shortcut), ACCREAD can then be read.

When a double transition has been captured and accumulated, a DBLRDY event is sent. Using the RDCLRDBL task (manually sent upon receiving the event, or using the DBLRDY_RDCLRDBL shortcut), ACCDBLREAD can then be read.

6.17.5 Output/input pins

The QDEC uses a three-pin interface to the off-chip quadrature encoder.

These pins are acquired when the QDEC is enabled in the ENABLE register. The pins acquired by the QDEC cannot be written by the CPU, but they can still be read by the CPU.

The pin numbers used for the QDEC are selected using the PSEL.n registers.

6.17.6 Pin configuration

The Phase A, Phase B, and LED signals are mapped to physical pins according to the configuration specified in the PSEL.A, PSEL.B, and PSEL.LED registers respectively.

If the CONNECT field value 'Disconnected' is specified in any of these registers, the associated signal will not be connected to any physical pin. The PSEL.A, PSEL.B, and PSEL.LED registers and their configurations are only used as long as the QDEC is enabled, and retained only as long as the device is in ON mode.

When the peripheral is disabled, the pins will behave as regular GPIOs, and use the configuration in their respective OUT bit field and PIN_CNF[n] register.

To secure correct behavior in the QDEC, the pins used by the QDEC must be configured in the GPIO peripheral as described in [GPIO configuration before enabling peripheral](#) on page 266 before enabling the QDEC. This configuration must be retained in the GPIO for the selected I/Os as long as the QDEC is enabled.

Only one peripheral can be assigned to drive a particular GPIO pin at a time. Failing to do so may result in unpredictable behavior.

QDEC signal	QDEC pin	Direction	Output value	Comment
Phase A	As specified in PSEL.A	Input	Not applicable	
Phase B	As specified in PSEL.B	Input	Not applicable	
LED	As specified in PSEL.LED	Input	Not applicable	

Table 71: GPIO configuration before enabling peripheral

6.17.7 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40012000	QDEC	QDEC	Quadrature decoder	

Table 72: Instances

Register	Offset	Description
TASKS_START	0x000	Task starting the quadrature decoder
TASKS_STOP	0x004	Task stopping the quadrature decoder
TASKS_READCLRACC	0x008	Read and clear ACC and ACCDBL
TASKS_RDCLRACC	0x00C	Read and clear ACC
TASKS_RDCLRDBL	0x010	Read and clear ACCDBL
EVENTS_SAMPLERDY	0x100	Event being generated for every new sample value written to the SAMPLE register
EVENTS_REPORTRDY	0x104	Non-null report ready
EVENTS_ACCOF	0x108	ACC or ACCDBL register overflow
EVENTS_DBLRDY	0x10C	Double displacement(s) detected
EVENTS_STOPPED	0x110	QDEC has been stopped
SHORTS	0x200	Shortcuts between local events and tasks
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
ENABLE	0x500	Enable the quadrature decoder
LEDPOL	0x504	LED output pin polarity
SAMPLEPER	0x508	Sample period
SAMPLE	0x50C	Motion sample value
REPORTPER	0x510	Number of samples to be taken before REPORTRDY and DBLRDY events can be generated
ACC	0x514	Register accumulating the valid transitions
ACCREAD	0x518	Snapshot of the ACC register, updated by the READCLRACC or RDCLRACC task
PSEL.LED	0x51C	Pin select for LED signal
PSEL.A	0x520	Pin select for A signal
PSEL.B	0x524	Pin select for B signal
DBFEN	0x528	Enable input debounce filters
LEDPRE	0x540	Time period the LED is switched ON prior to sampling
ACCDBL	0x544	Register accumulating the number of detected double transitions

Register	Offset	Description
ACCDLREAD	0x548	Snapshot of the ACCDBL, updated by the READCLRACC or RDCLRDBL task

Table 73: Register overview

6.17.7.1 TASKS_START

Address offset: 0x000

Task starting the quadrature decoder

When started, the SAMPLE register will be continuously updated at the rate given in the SAMPLEPER register.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W	TASKS_START		Task starting the quadrature decoder																															
				When started, the SAMPLE register will be continuously updated at the rate given in the SAMPLEPER register.																															
		Trigger	1	Trigger task																															

6.17.7.2 TASKS_STOP

Address offset: 0x004

Task stopping the quadrature decoder

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W	TASKS_STOP		Task stopping the quadrature decoder																															
		Trigger	1	Trigger task																															

6.17.7.3 TASKS_READCLRACC

Address offset: 0x008

Read and clear ACC and ACCDBL

Task transferring the content of ACC to ACCREAD and the content of ACCDBL to ACCDBLREAD, and then clearing the ACC and ACCDBL registers. These read-and-clear operations will be done atomically.

Bit number		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																														
ID		A																														
Reset 0x00000000		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																														
ID	Acce Field	Value ID		Value		Description																										
A	W	TASKS_READCLRACC				Read and clear ACC and ACCDBL																										
						Task transferring the content of ACC to ACCREAD and the content of ACCDBL to ACCDBLREAD, and then clearing the ACC and ACCDBL registers. These read-and-clear operations will be done atomically.																										
		Trigger		1		Trigger task																										

6.17.7.4 TASKS_RDCLRACC

Address offset: 0x00C

Read and clear ACC

Task transferring the content of ACC to ACCREAD, and then clearing the ACC register. This read-and-clear operation will be done atomically.

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																														
ID				A																														
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																														
ID	Acce	Field	Value ID	Value	Description																													
A	W	TASKS_RDCLRACC			Read and clear ACC																													
					Task transferring the content of ACC to ACCREAD, and then clearing the ACC register. This read-and-clear operation will be done atomically.																													
			Trigger	1	Trigger task																													

6.17.7.5 TASKS_RDCLRDBL

Address offset: 0x010

Read and clear ACCDBL

Task transferring the content of ACCDBL to ACCDBLREAD, and then clearing the ACCDBL register. This read-and-clear operation will be done atomically.

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value ID		Value		Description																											
A	W	TASKS_RDCLRDBL					Read and clear ACCDBL																											
							Task transferring the content of ACCDBL to ACCDBLREAD, and then clearing the ACCDBL register. This read-and-clear operation will be done atomically.																											
			Trigger	1			Trigger task																											

6.17.7.6 EVENTS_SAMPLERDY

Address offset: 0x100

Event being generated for every new sample value written to the SAMPLE register

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	EVENTS_SAMPLERDY		Event being generated for every new sample value written to the SAMPLE register																														
		NotGenerated	0	Event not generated																														
		Generated	1	Event generated																														

6.17.7.7 EVENTS_REPORTRDY

Address offset: 0x104

Non-null report ready

Event generated when REPORTPER number of samples has been accumulated in the ACC register and the content of the ACC register is not equal to 0. (Thus, this event is only generated if a motion is detected since the previous clearing of the ACC register).

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
ID			A																																				
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
ID	Acce Field	Value ID	Value		Description																																		
A	RW	EVENTS_REPORTRDY			Non-null report ready																																		
					Event generated when REPORTPER number of samples has been accumulated in the ACC register and the content of the ACC register is not equal to 0. (Thus, this event is only generated if a motion is detected since the previous clearing of the ACC register).																																		
			NotGenerated	0	Event not generated																																		
			Generated	1	Event generated																																		

6.17.7.8 EVENTS_ACCOF

Address offset: 0x108

ACC or ACCDBL register overflow

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	EVENTS_ACCOF		ACC or ACCDBL register overflow																														
		NotGenerated	0	Event not generated																														
		Generated	1	Event generated																														

6.17.7.9 EVENTS_DBLRDY

Address offset: 0x10C

Double displacement(s) detected

Event generated when REPORTPER number of samples has been accumulated and the content of the ACCDBL register is not equal to 0. (Thus, this event is only generated if a double transition is detected since the previous clearing of the ACCDBL register).

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID				A																																
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																																
A	RW	EVENTS_DBLRDY		Double displacement(s) detected																																
				Event generated when REPORTPER number of samples has been accumulated and the content of the ACCDBL register is not equal to 0. (Thus, this event is only generated if a double transition is detected since the previous clearing of the ACCDBL register).																																
		NotGenerated	0	Event not generated																																
		Generated	1	Event generated																																

6.17.7.10 EVENTS_STOPPED

Address offset: 0x110

QDEC has been stopped

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
ID				A																																		
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
ID	Acce Field	Value ID	Value	Description																																		
A	RW	EVENTS_STOPPED		QDEC has been stopped																																		
		NotGenerated	0	Event not generated																																		
		Generated	1	Event generated																																		

6.17.7.11 SHORTS

Address offset: 0x200

Shortcuts between local events and tasks

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			G F E D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A	RW	REPORTRDY_READCLRACC				Shortcut between event REPORTRDY and task READCLRACC																												
		Disabled	0	Disable shortcut																														
		Enabled	1	Enable shortcut																														
B	RW	SAMPLERDY_STOP				Shortcut between event SAMPLERDY and task STOP																												
		Disabled	0	Disable shortcut																														
		Enabled	1	Enable shortcut																														
C	RW	REPORTRDY_RDCLRACC				Shortcut between event REPORTRDY and task RDCLRACC																												
		Disabled	0	Disable shortcut																														
		Enabled	1	Enable shortcut																														
D	RW	REPORTRDY_STOP				Shortcut between event REPORTRDY and task STOP																												
		Disabled	0	Disable shortcut																														
		Enabled	1	Enable shortcut																														
E	RW	DBLRDY_RDCLRDBL				Shortcut between event DBLRDY and task RDCLRDBL																												
		Disabled	0	Disable shortcut																														
		Enabled	1	Enable shortcut																														
F	RW	DBLRDY_STOP				Shortcut between event DBLRDY and task STOP																												
		Disabled	0	Disable shortcut																														

6.17.7.12 INTENSET

Enable interrupt

6.17.7.13 INTENCLR

Disable interrupt

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			E D C B A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW SAMPLERDY			Write '1' to disable interrupt for event SAMPLERDY																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
B	RW REPORTRDY			Write '1' to disable interrupt for event REPORTRDY																														
				Event generated when REPORTPER number of samples has been accumulated in the ACC register and the content of the ACC register is not equal to 0. (Thus, this event is only generated if a motion is detected since the previous clearing of the ACC register).																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
C	RW ACCOF			Write '1' to disable interrupt for event ACCOF																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														
D	RW DBLRDY			Write '1' to disable interrupt for event DBLRDY																														
				Event generated when REPORTPER number of samples has been accumulated and the content of the ACCDBL register is not equal to 0. (Thus, this event is only generated if a double transition is detected since the previous clearing of the ACCDBL register).																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
E	RW STOPPED			Write '1' to disable interrupt for event STOPPED																														
		Clear	1	Disable																														
		Disabled	0	Read: Disabled																														
		Enabled	1	Read: Enabled																														

6.17.7.14 ENABLE

Address offset: 0x500

Enable the quadrature decoder

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	ENABLE		Enable or disable the quadrature decoder																														
				When enabled the decoder pins will be active. When disabled the quadrature decoder pins are not active and can be used as GPIO .																														
			Disabled	0	Disable																													
			Enabled	1	Enable																													

6.17.7.15 LEDPOL

Address offset: 0x504

LED output pin polarity

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	RW	LEDPOL		LED output pin polarity																														
		ActiveLow	0	Led active on output pin low																														
		ActiveHigh	1	Led active on output pin high																														

6.17.7.16 SAMPLEPER

Address offset: 0x508

Sample period

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
ID																																				A	A	A	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
ID	Acce	Field	Value	ID	Value	Description																																	
A	RW	SAMPLEPER				Sample period. The SAMPLE register will be updated for every new sample																																	
			128us	0		128 μs																																	
			256us	1		256 μs																																	
			512us	2		512 μs																																	
			1024us	3		1024 μs																																	
			2048us	4		2048 μs																																	
			4096us	5		4096 μs																																	
			8192us	6		8192 μs																																	
			16384us	7		16384 μs																																	
			32ms	8		32768 μs																																	
			65ms	9		65536 μs																																	
			131ms	10		131072 μs																																	

6.17.7.17 SAMPLE

Address offset: 0x50C

Motion sample value

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field		Value ID	Value				Description																											
A	R	SAMPLE		[-1..2]				Last motion sample																											
				The value is a 2's complement value, and the sign gives the direction of the motion. The value '2' indicates a double transition.																															

6.17.7.18 REPORTPER

Address offset: 0x510

Number of samples to be taken before REPORTRDY and DBLRDY events can be generated

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A	RW	REPORTPER				Specifies the number of samples to be accumulated in the ACC register before the REPORTRDY and DBLRDY events can be generated.																												
						The report period in [μs] is given as: RPUS = SP * RP Where RPUS is the report period in [μs/report], SP is the sample period in [μs/sample] specified in SAMPLEPER, and RP is the report period in [samples/report] specified in REPORTPER .																												
			10Smpl	0		10 samples/report																												
			40Smpl	1		40 samples/report																												
			80Smpl	2		80 samples/report																												
			120Smpl	3		120 samples/report																												
			160Smpl	4		160 samples/report																												
			200Smpl	5		200 samples/report																												
			240Smpl	6		240 samples/report																												
			280Smpl	7		280 samples/report																												
			1Smpl	8		1 sample/report																												

6.17.7.19 ACC

Address offset: 0x514

Register accumulating the valid transitions

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A A A A A A A A A A A A A A A A A A A A A A A A A A A A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	R	ACC	[-1024..1023]	Register accumulating all valid samples (not double transition) read from the SAMPLE register. Double transitions (SAMPLE = 2) will not be accumulated in this register. The value is a 32 bit 2's complement value. If a sample that would cause this register to overflow or underflow is received, the sample will be ignored and an overflow event (ACCOF) will be generated. The ACC register is cleared by triggering the READCLRACC or the RDCLRACC task.																														

6.17.7.20 ACCREAD

Address offset: 0x518

Snapshot of the ACC register, updated by the READCLRACC or RDCLRACC task

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID		Value				Description																											
A	R	ACCREAD		[-1024..1023]				Snapshot of the ACC register.																											
							The ACCREAD register is updated when the READCLRACC or RDCLRACC task is triggered.																												

6.17.7.21 PSEL.LED

Address offset: 0x51C

Pin select for LED signal

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID				C																								B				A	A	A	A
Reset 0xFFFFFFFF				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ID	Acce	Field	Value	ID	Value		Description																												
A	RW	PIN			[0..31]		Pin number																												
B	RW	PORT			[0..1]		Port number																												
C	RW	CONNECT					Connection																												
			Disconnected		1		Disconnect																												
			Connected		0		Connect																												

6.17.7.22 PSEL.A

Address offset: 0x520

Pin select for A signal

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID				C																												B	A	A	A	A
Reset 0xFFFFFFFF				1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
ID	Acce	Field	Value	ID	Value		Description																													
A	RW	PIN			[0..31]		Pin number																													
B	RW	PORT			[0..1]		Port number																													
C	RW	CONNECT					Connection																													
			Disconnected		1		Disconnect																													
			Connected		0		Connect																													

6.17.7.23 PSEL.B

Address offset: 0x524

Pin select for B signal

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			C																															
Reset 0xFFFFFFFF			1																															
ID	Acce Field	Value ID	Value				Description																											
A	RW PIN		[0..31]				Pin number																											
B	RW PORT		[0..1]				Port number																											
C	RW CONNECT						Connection																											
		Disconnected	1				Disconnect																											
		Connected	0				Connect																											

6.17.7.24 DBFEN

Address offset: 0x528

Enable input debounce filters

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID																																			A	
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
ID	Acce Field	Value ID	Value		Description																															
A	RW	DBFEN			Enable input debounce filters																															
		Disabled	0		Debounce input filters disabled																															
		Enabled	1		Debounce input filters enabled																															

6.17.7.25 LEDPRE

Address offset: 0x540

Time period the LED is switched ON prior to sampling

Bit number										31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
ID																														A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
Reset 0x00000010										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.17.7.26 ACCDBL

Address offset: 0x544

Register accumulating the number of detected double transitions

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A A A A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce	Field	Value	ID	Value	Description																												
A	R	ACCDL	[0..15]			Register accumulating the number of detected double or illegal transitions. (SAMPLE = 2).																												
						When this register has reached its maximum value, the accumulation of double/illegal transitions will stop. An overflow event (ACCOF) will be generated if any double or illegal transitions are detected after the maximum value was reached. This field is cleared by triggering the READCLRACC or RDCLRDL task.																												

6.17.7.27 ACCDBLREAD

Address offset: 0x548

Snapshot of the ACCDBL, updated by the READCLRACC or RDCLRDBL task

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A A A A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acc Field	Value ID	Value	Description																															
A	R	ACCDBLREAD	[0..15]	Snapshot of the ACCDBL register. This field is updated when the READCLRACC or RDCLRDBL task is triggered.																															

6.17.8 Electrical specification

6.17.8.1 QDEC Electrical Specification

Symbol	Description	Min.	Typ.	Max.	Units
t _{SAMPLE}	Time between sampling signals from quadrature decoder	128		131072	μs
t _{LED}	Time from LED is turned on to signals are sampled	0		511	μs

6.18 RADIO — 2.4 GHz radio

The 2.4 GHz radio transceiver is compatible with multiple radio standards such as 1 Mbps and 2 Mbps *Bluetooth*[®] Low Energy modes, Long Range (125 kbps and 500 kbps) *Bluetooth*[®] Low Energy modes, IEEE 802.15.4 250 kbps mode, as well as Nordic's proprietary 1 Mbps and 2 Mbps modes.

Listed here are main features for the RADIO:

- Multidomain 2.4 GHz radio transceiver
 - 1 Mbps and 2 Mbps *Bluetooth*[®] Low Energy modes
 - Long Range (125 kbps and 500 kbps) *Bluetooth*[®] Low Energy modes
 - Angle-of-arrival (AoA) and angle-of-departure (AoD) direction finding using *Bluetooth*[®] Low Energy
 - IEEE 802.15.4 250 kbps mode
 - 1 Mbps and 2 Mbps Nordic proprietary modes
- Best in class link budget and low power operation
- Efficient data interface with EasyDMA support
- Automatic address filtering and pattern matching

EasyDMA, in combination with an automated packet assembler, packet disassembler, automated CRC generator and CRC checker, makes it easy to configure and use the RADIO. See the following figure for details.

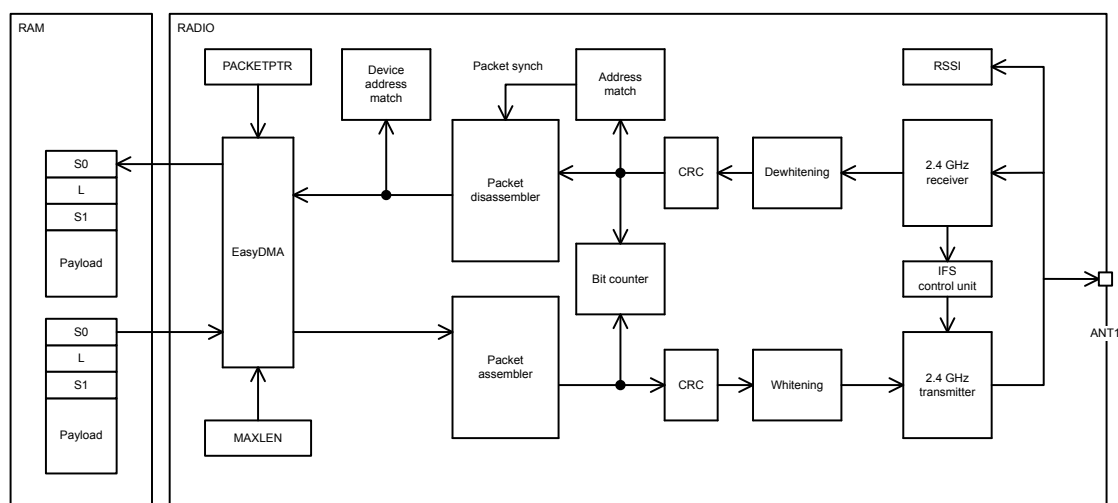


Figure 83: RADIO block diagram

The RADIO includes a device address match unit and an interframe spacing control unit that can be utilized to simplify address whitelisting and interframe spacing respectively in *Bluetooth*® low energy and similar applications.

The RADIO also includes a received signal strength indicator (RSSI) and a bit counter. The bit counter generates events when a preconfigured number of bits are sent or received by the RADIO.

6.18.1 Packet configuration

A RADIO packet contains the fields PREAMBLE, ADDRESS, S0, LENGTH, S1, PAYLOAD, and CRC. For Long Range (125 kbps and 500 kbps) *Bluetooth*® Low Energy modes, fields CI, TERM1 and TERM2 are also included.

The content of a RADIO packet is illustrated in the figures below. The RADIO sends the fields in the packet according to the order illustrated in the figures, starting on the left.

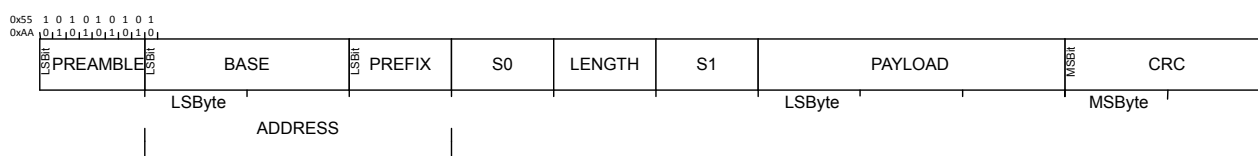
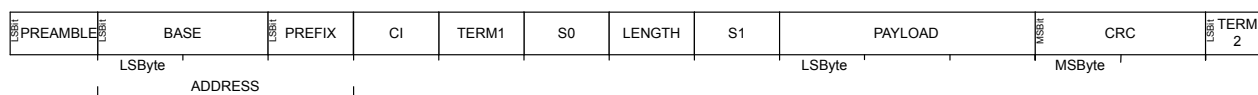


Figure 84: On-air packet layout

Figure 85: On-air packet layout for Long Range (125 kbps and 500 kbps) *Bluetooth*® Low Energy modes

Not shown in the figures is the static payload add-on (the length of which is defined in `PCNF1.STATLEN`, and which is 0 bytes in a standard BLE packet). The static payload add-on is sent between PAYLOAD and CRC fields. The RADIO sends the different fields in the packet in the order they are illustrated above, from left to right.

PREAMBLE is sent with least significant bit first on air. The size of the PREAMBLE depends on the mode selected in the `MODE` register:

- The PREAMBLE is one byte for `MODE = Ble_1Mbit` as well as all Nordic proprietary operating modes (`MODE = Nrf_1Mbit` and `MODE = Nrf_2Mbit`), and `PCNF0.PLEN` has to be set accordingly. If the first bit of the ADDRESS is 0, the preamble will be set to 0xAA. Otherwise the PREAMBLE will be set to 0x55.

- For **MODE** = Ble_2Mbit, the PREAMBLE must be set to 2 byte through **PCNF0.PLEN**. If the first bit of the ADDRESS is 0, the preamble will be set to 0xAAAA. Otherwise the PREAMBLE will be set to 0x5555.
- For **MODE** = Ble_LR125Kbit and **MODE** = Ble_LR500Kbit, the PREAMBLE is 10 repetitions of 0x3C.
- For **MODE** = leee802154_250Kbit, the PREAMBLE is 4 bytes and set to all zeros.

Radio packets are stored in memory inside instances of a RADIO packet data structure as illustrated below. The PREAMBLE, ADDRESS, CI, TERM1, TERM2, and CRC fields are omitted in this data structure. Fields S0, LENGTH, and S1 are optional.

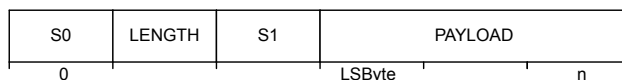


Figure 86: In-RAM representation of RADIO packet

The byte ordering on air is always least significant byte first for the ADDRESS and PAYLOAD fields, and most significant byte first for the CRC field. The ADDRESS fields are always transmitted and received least significant bit first. The CRC field is always transmitted and received most significant bit first. The endianness, i.e. the order in which the bits are sent and received, of the S0, LENGTH, S1, and PAYLOAD fields can be configured via **PCNF1.ENDIAN**.

The sizes of the S0, LENGTH and S1 fields can be individually configured via **SOLEN**, **LFLEN**, and **S1LEN** in **PCNF0** respectively. If any of these fields are configured to be less than 8 bits, the least significant bits of the fields are used.

If S0, LENGTH, or S1 are specified with zero length, their fields will be omitted in memory. Otherwise each field will be represented as a separate byte, regardless of the number of bits in their on-air counterpart.

Independent of the configuration of **PCNF1.MAXLEN**, the combined length of S0, LENGTH, S1, and PAYLOAD cannot exceed 258 bytes.

6.18.2 Address configuration

The on-air radio ADDRESS field is composed of two parts, the base address field and the address prefix field.

The size of the base address field is configurable via **PCNF1.BALEN**. The base address is truncated from the least significant byte if the **PCNF1.BALEN** is less than 4. See [Definition of logical addresses](#) on page 279.

Logical address	Base address	Prefix byte
0	BASE0	PREFIX0.AP0
1	BASE1	PREFIX0.AP1
2	BASE1	PREFIX0.AP2
3	BASE1	PREFIX0.AP3
4	BASE1	PREFIX1.AP4
5	BASE1	PREFIX1.AP5
6	BASE1	PREFIX1.AP6
7	BASE1	PREFIX1.AP7

Table 74: Definition of logical addresses

The on-air addresses are defined in the **BASE0/BASE1** and **PREFIX0/PREFIX1** registers. It is only when writing these registers that the user must relate to the actual on-air addresses. For other radio address registers, such as the **TXADDRESS**, **RXADDRESSES**, and **RXMATCH** registers, logical radio addresses ranging from 0 to 7 are being used. The relationship between the on-air radio addresses and the logical addresses is described in [Definition of logical addresses](#) on page 279.

6.18.3 Data whitening

The RADIO is able to do packet whitening and de-whitening, enabled in [PCNF1.WHITEEN](#). When enabled, whitening and de-whitening will be handled by the RADIO automatically as packets are sent and received.

The whitening word is generated using polynomial $g(D) = D^7 + D^4 + 1$, which then is XORed with the data packet that is to be whitened, or de-whitened. See the figure below.

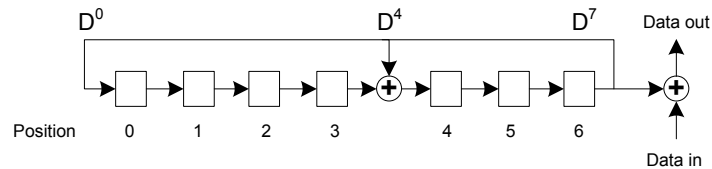


Figure 87: Data whitening and de-whitening

Whitening and de-whitening will be performed over the whole packet except for the preamble and the address fields.

The linear feedback shift register in the figure above is initialized via [DATAWHITEIV](#).

6.18.4 CRC

The CRC generator in the RADIO calculates the CRC over the whole packet excluding the preamble. If desirable, the address field can be excluded from the CRC calculation as well.

See [CRCCNF](#) register for more information.

The CRC polynomial is configurable as illustrated in the following figure, where bit 0 in the [CRCPOLY](#) register corresponds to X^0 and bit 1 corresponds to X^1 etc. See [CRCPOLY](#) on page 325 for more information.

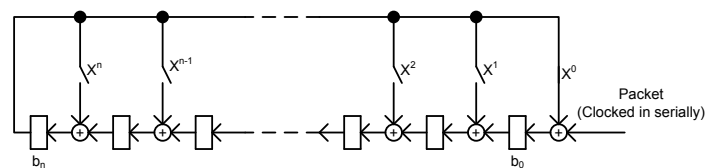


Figure 88: CRC generation of an n bit CRC

The figure shows that the CRC is calculated by feeding the packet serially through the CRC generator. Before the packet is clocked through the CRC generator, the CRC generator's latches b_0 through b_n will be initialized with a predefined value specified in the [CRCINIT](#) register. After the whole packet has been clocked through the CRC generator, b_0 through b_n will hold the resulting CRC. This value will be used by the RADIO during both transmission and reception. Latches b_0 through b_n are not available to be read by the CPU at any time. However, a received CRC can be read by the CPU via the [RXCRC](#) register.

The length (n) of the CRC is configurable, see [CRCCNF](#) for more information.

Once the entire packet, including the CRC, has been received and no errors were detected, the RADIO generates a [CRCOK](#) event. If CRC errors were detected, a [CRCERROR](#) event is generated.

Tasks and events are used to control the operating state of the RADIO.

State	Description
DISABLED	No operations are going on inside the RADIO and the power consumption is at a minimum
RXRU	The RADIO is ramping up and preparing for reception
RXIDLE	The RADIO is ready for reception to start
RX	Reception has been started and the addresses enabled in the RXADDRESSES register are being monitored
TXRU	The RADIO is ramping up and preparing for transmission
TXIDLE	The RADIO is ready for transmission to start
TX	The RADIO is transmitting a packet
RXDISABLE	The RADIO is disabling the receiver
TXDISABLE	The RADIO is disabling the transmitter

A state diagram showing an overview of the RADIO is shown in the following figure.



The END to START shortcut should not be used with IEEE 802.15.4 250 kbps mode. Use the PHYEND to START shortcut instead.

6.18.6 Transmit sequence

4452 021 v1.3

TXEN task is triggered. After the RADIO has successfully ramped up it will generate the **READY** event indicating that a packet transmission can be initiated. A packet transmission is initiated by triggering the **START** task. The **START** task can first be triggered after the RADIO has entered into the TXIDLE state.

The following figure illustrates a single packet transmission where the CPU manually triggers the different tasks needed to control the flow of the RADIO, i.e. no shortcuts are used. If shortcuts are not used, a certain amount of delay caused by CPU execution is expected between **READY** and **START**, and between **END** and **DISABLE**. As illustrated in [Transmit sequence](#) on page 282 the RADIO will by default transmit '1's between **READY** and **START**, and between **END** and **DISABLE**. What is transmitted can be programmed through the DTX field in the **MODECNF0** register.

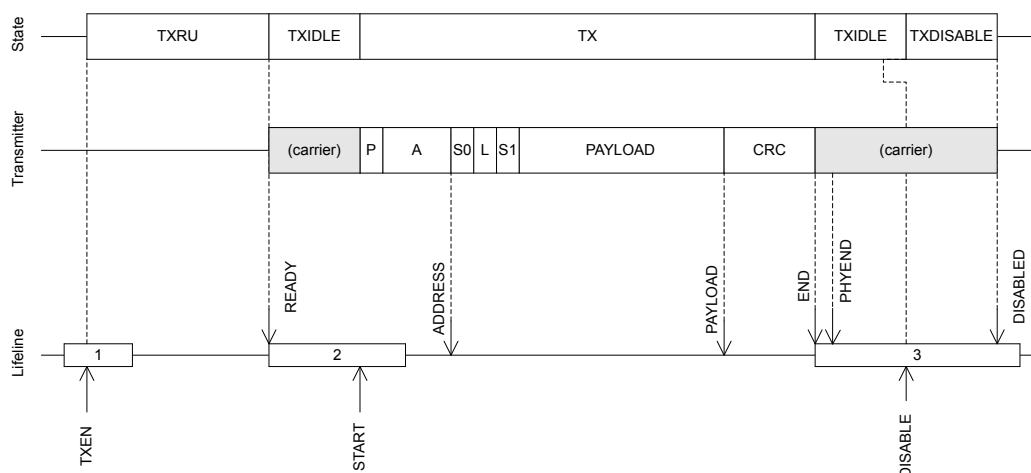


Figure 90: Transmit sequence

The following figure shows a slightly modified version of the transmit sequence where the RADIO is configured to use shortcuts between **READY** and **START**, and between **END** and **DISABLE**, which means that no delay is introduced.

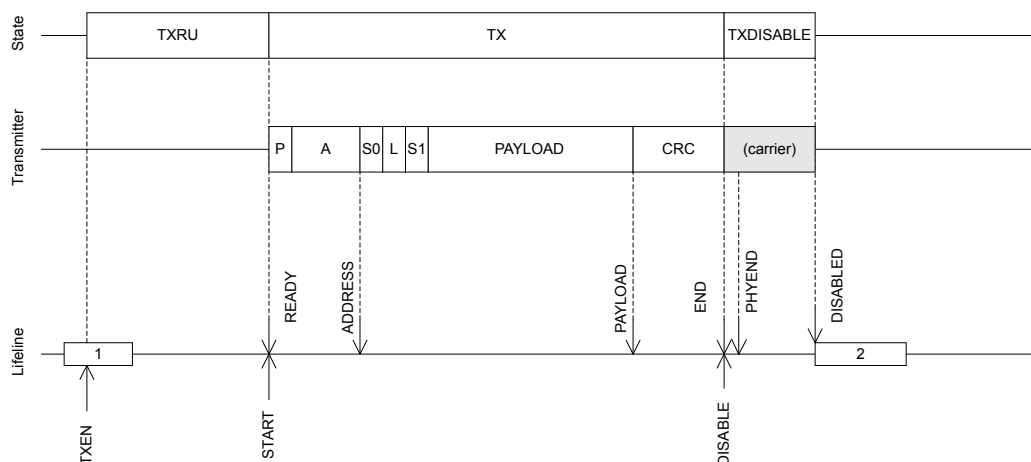


Figure 91: Transmit sequence using shortcuts to avoid delays

The RADIO is able to send multiple packets one after the other without having to disable and re-enable the RADIO between packets, as illustrated in the following figure.

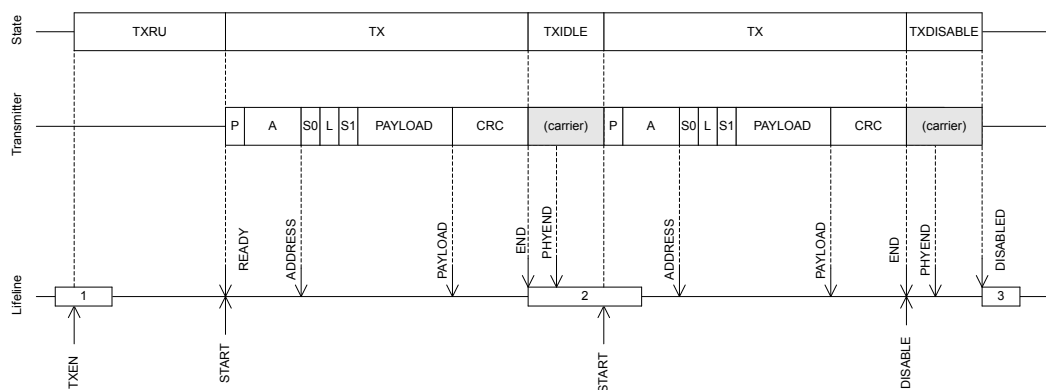


Figure 92: Transmission of multiple packets

6.18.7 Receive sequence

Before the RADIO is able to receive a packet, it must first ramp up in RX mode, see RXRU in [Radio states](#) on page 281 and [Receive sequence](#) on page 283.

An RXRU ramp up sequence is initiated when the [RXEN](#) task is triggered. After the RADIO has successfully ramped up it will generate the [READY](#) event indicating that a packet reception can be initiated. A packet reception is initiated by triggering the [START](#) task. As illustrated in [Radio states](#) on page 281, the [START](#) task can first be triggered after the RADIO has entered into the RXIDLE state.

The following figure shows a single packet reception where the CPU manually triggers the different tasks needed to control the flow of the RADIO, i.e. no shortcuts are used. If shortcuts are not used, a certain amount of delay caused by CPU execution is expected between [READY](#) and [START](#), and between [END](#) and [DISABLE](#). The RADIO will be listening and possibly receiving undefined data, represented with an 'X', from [START](#) and until a packet with valid preamble (P) is received.

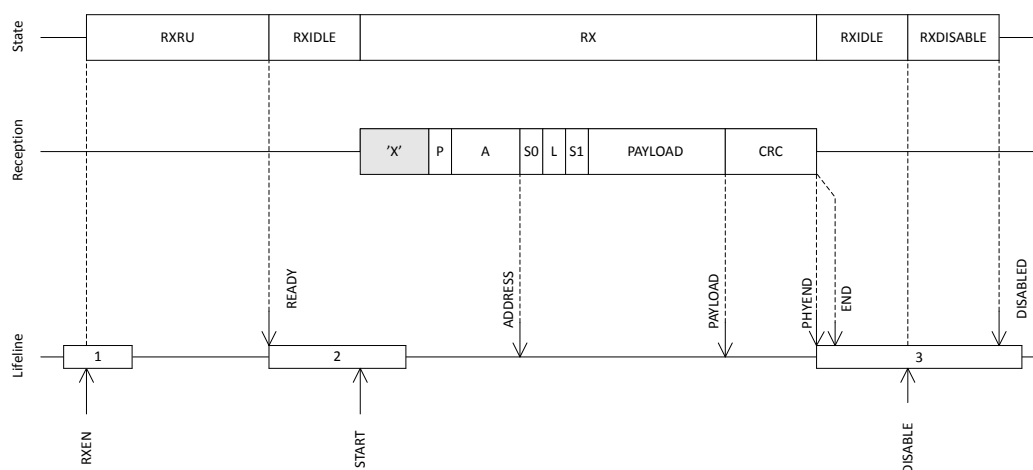


Figure 93: Receive sequence

The following figure shows a slightly modified version of the receive sequence, where the RADIO is configured to use shortcuts between [READY](#) and [START](#), and between [END](#) and [DISABLE](#), which means that no delay is introduced.

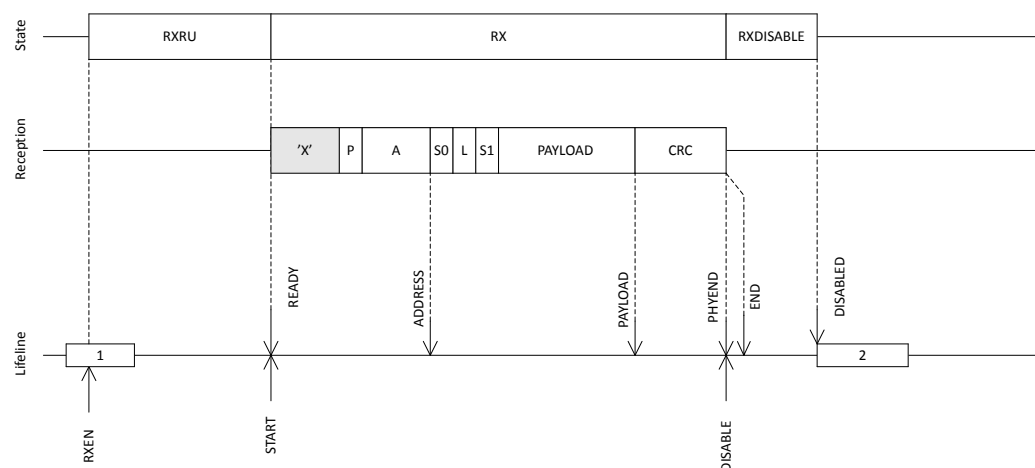


Figure 94: Receive sequence using shortcuts to avoid delays

The RADIO is able to receive consecutive packets without having to disable and re-enable the RADIO between packets, as illustrated in the figure below.

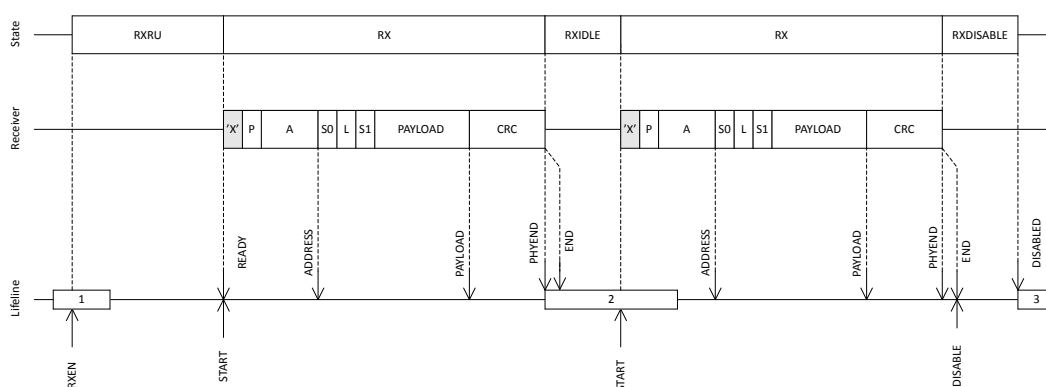


Figure 95: Reception of multiple packets

6.18.8 Received signal strength indicator (RSSI)

The RADIO implements a mechanism for measuring the power in the received signal. This feature is called received signal strength indicator (RSSI).

The RSSI is measured continuously and the value filtered using a single-pole IIR filter. After a signal level change, the RSSI will settle after approximately `RSSISETTLE`.

Sampling of the received signal strength is started by using the `RSSISTART` task. The sample can be read from the `RSSISAMPLE` register.

The sample period of the RSSI is defined by `RSSIPERIOD`. The `RSSISAMPLE` will hold the filtered received signal strength after this sample period.

For the RSSI sample to be valid, the RADIO has to be enabled in receive mode (`RXEN` task) and the reception has to be started (`READY` event followed by `START` task).

6.18.9 Interframe spacing (IFS)

Interframe spacing (IFS) is defined as the time, in microseconds, between two consecutive packets, starting from when the end of the last bit of the previous packet is received, to the beginning of the first bit of the subsequent packet that is transmitted. The RADIO is able to enforce this interval, as specified in the `TIFS` register, as long as the `TIFS` is not specified to be shorter than the RADIO's turnaround time, i.e.

the time needed to switch off the receiver, and then switch the transmitter back on. The **TIFS** register can be written any time before the last bit on air is received.

This timing is illustrated in the figure below.

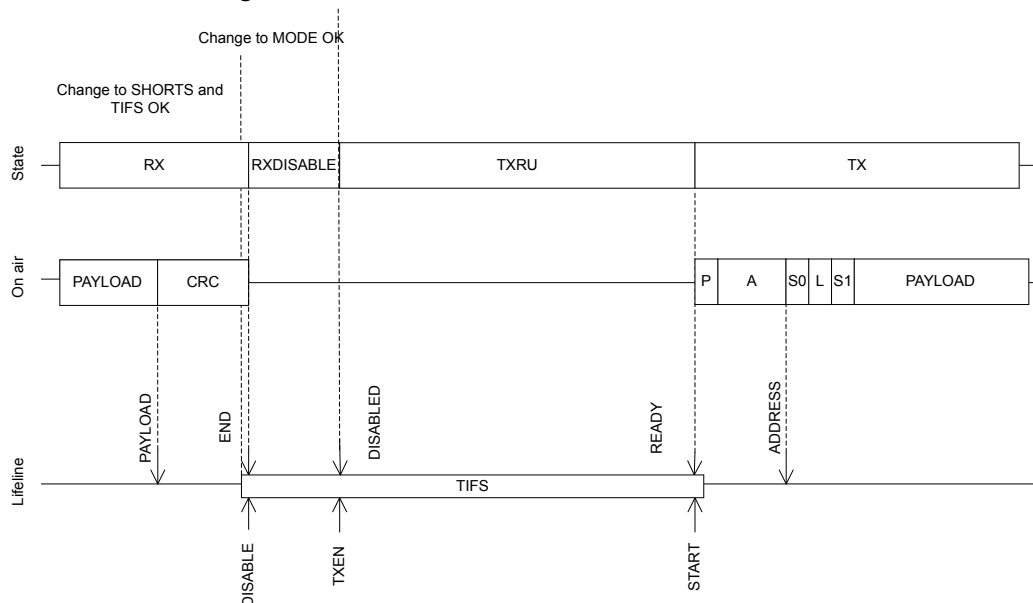


Figure 96: IFS timing detail

The TIFS duration starts after the last bit on air (just before the **END** event), and elapses with first bit being transmitted on air (just after **READY** event).

TIFS is only enforced if the shortcuts **END** to **DISABLE** and **DISABLED** to **TXEN** or **END** to **DISABLE** and **DISABLED** to **RXEN** are enabled.

TIFS is qualified for use in IEEE 802.15.4 250kbps mode, Long Range (125 kbps and 500 kbps) **Bluetooth**® Low Energy modes, 1 Mbps and 2 Mbps **Bluetooth**® Low Energy modes, using the default ramp-up mode.

SHORTS and **TIFS** registers are not double-buffered, and can be updated at any point before the last bit on air is received. The **MODE** register is double-buffered and sampled at the **TXEN** or **RXEN** task.

6.18.10 Device address match

The device address match feature is tailored for address whitelisting in **Bluetooth**® low energy and similar implementations.

This feature enables on-the-fly device address matching while receiving a packet on air. This feature only works in receive mode and when the RADIO is configured for little endian, see **PCNF1.ENDIAN**.

The device address match unit assumes that the first 48 bits of the payload are the device address and that bit number 6 in S0 is the TxAdd bit. See the **Bluetooth**® Core Specification for more information about device addresses, TxAdd, and whitelisting.

The RADIO is able to listen for eight different device addresses at the same time. These addresses are specified in a DAB/DAP register pair, one pair per address, in addition to a TxAdd bit configured in the DACNF register. The DAB register specifies the 32 least significant bits of the device address, while the DAP register specifies the 16 most significant bits of the device address.

Each of the device addresses can be individually included or excluded from the matching mechanism. This is configured in the **DACNF** register.

6.18.11 Bit counter

The RADIO implements a simple counter that can be configured to generate an event after a specific number of bits have been transmitted or received.

By using shortcuts, this counter can be started from different events generated by the RADIO and count relative to these.

The bit counter is started by triggering the **BCSTART** task, and stopped by triggering the **BCSTOP** task. A **BCMATCH** event will be generated when the bit counter has counted the number of bits specified in the **BCC** register. The bit counter will continue to count bits until the **DISABLED** event is generated or until the **BCSTOP** task is triggered. The CPU can therefore, after a **BCMATCH** event, reconfigure the **BCC** value for new **BCMATCH** events within the same packet.

The bit counter can only be started after the RADIO has received the **ADDRESS** event.

The bit counter will stop and reset on either the **BCSTOP**, **STOP**, or **DISABLE** task, or the **END** event.

The figure below illustrates how the bit counter can be used to generate a **BCMATCH** event in the beginning of the packet payload, and again generate a second **BCMATCH** event after sending 2 bytes (16 bits) of the payload.

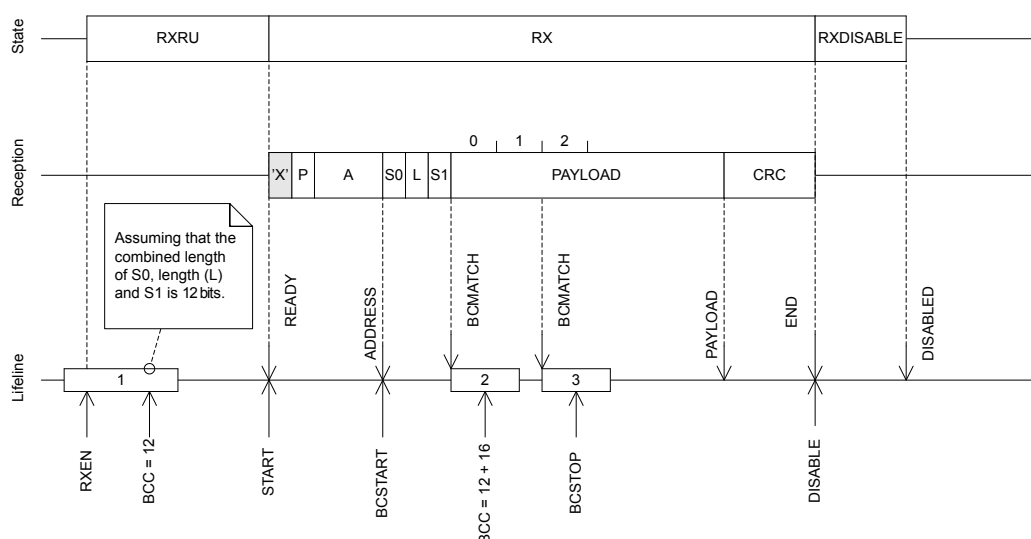


Figure 97: Bit counter example

6.18.12 Direction finding

The RADIO implements the Angle-of-Arrival (AoA) and Angle-of-Departure (AoD) Bluetooth Low Energy feature, which can be used to determine the direction of a peer device. The feature is available for the BLE 1 Mbps and BLE 2 Mbps modes.

When using this feature, the transmitter sends a packet with a continuous tone extension (CTE) appended to the packet, after the CRC. During the CTE, the receiver can take IQ samples of the incoming signal.

An antenna array is employed at the transmitter (AoD) or at the receiver (AoA). The AoD transmitter, or AoA receiver, switches between the antennas, in order to collect IQ samples from the different antenna pairs. The IQ samples can be used to calculate the relative path lengths between the antenna pairs, which can be used to estimate the direction of the transmitter.

6.18.12.1 CTE format

The CTE is from 16 μ s to 160 μ s and consists of an unwhitened sequence of 1's, equivalent to a continuous tone nominally offset from the carrier by +250 kHz for the 1 Mbps PHY and +500 kHz for the 2 Mbps BLE PHYs. The format of the CTE, when switching and/or sampling, is shown below.

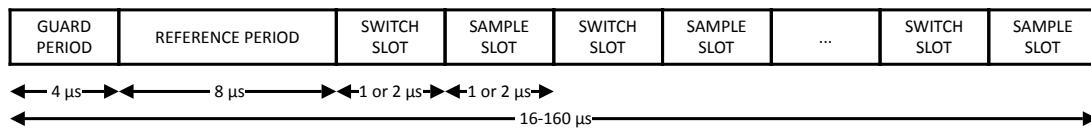


Figure 98: Constant tone extension (CTE) structure

Antenna switching is performed during switch slots and the guard period. The AoA/AoD feature requires that one IQ sample is taken for each microsecond within the reference period, and once for each sample slot. Oversampling is possible by changing the sample spacing as described in [IQ sampling](#) on page 290. The switch slot and sample slot durations are either 1 or 2 μs, but must be equal. The format of the CTE and switching and sampling procedures may be configured prior to, or during, packet transmission and reception. Alternatively, during packet reception, these operations can be configured by reading specific fields of the packet contents.

6.18.12.2 Mode

Depending on the [DFEMODE](#), the device performs the following procedures:

		DFEMODE			
		AOA		AOD	
		TX	RX	TX	RX
AoA/AoD Procedure	Generating and transmitting CTE	x		x	
	Receiving, interpreting, and sampling CTE		x		x
	Antenna switching		x	x	

Table 76: AoA/AoD Procedures performed as a function of DFEMODE and TX/RX mode

6.18.12.3 Inline configuration

When inline configuration is enabled during RX, further configuration of the AoA/AoD procedures is performed based on the values of the CP bit and the CTEInfo octet within the packet. This is enabled by setting [CTEINLINECONF.CTEINLINECTRLIN](#). The CTEInfo octet is present only if the CP bit is set. The position of the CP bit and CTEInfo octet depends on whether the packet has a *Data Channel PDU* ([CTEINLINECONF.CTEINFOINS1=InS1](#)), or an *Advertising Channel PDU* ([CTEINLINECONF.CTEINFOINS1=NotInS1](#)).

Data channel PDU

For Data Channel PDUs, [PCNF0.SOLEN](#) must be 1 byte, and [PCNF0.LFLEN](#) must be 8 bits. To determine if S1 is present, the registers [CTEINLINECONF.SOMASK](#) and [CTEINLINECONF.SOCONF](#) forms a bitwise mask-and-test for the S0 field. If the bitwise AND between S0 and SOMASK equals SOCONF, then S1 is determined to be present. When present, the value of [PCNF0.S1LEN](#) will be ignored, as this is decided by the CP bit in the the following figure.

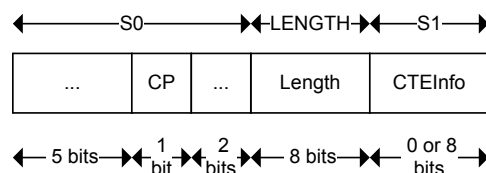


Figure 99: Data channel PDU header

When encrypting and decrypting BLE packets using the [CCM](#) peripheral, it is also required to set [PCNF0.S1INCL=1](#). The CCM mode must be configured to use an 8-bit length field. The value of the CP bit is included in the calculation of the MIC, while the S1 field is ignored by the CCM calculation.

Advertising channel PDU

For advertising channel PDUs, the CTEInfo Flag replaces the CP bit. The CTEInfo Flag is within the extended header flag field in some of the advertising PDUs that employ the common extended advertising payload format (i.e. AUX_SYNC_IND, AUX_CHAIN_IND). The format of such packets is shown in the following figure.

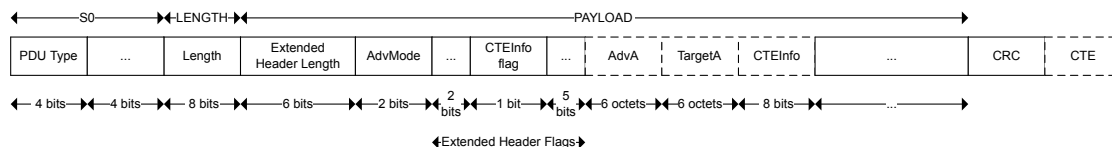


Figure 100: Advertising channel PDU header

The [CTEINLINECONF.SOCONF](#) and [CTEINLINECONF.SOMASK](#) fields can be configured to accept only certain advertising PDU Types. If the extended header length is non-zero, the CTEInfo extended header flag is checked to determine whether CTEInfo is present. If a bit before the CTEInfo flag within the extended header flags is set, then the CTEInfo position is postponed 6 octets.

CTEInfo parsing

The CTEInfo field is shown in the following figure.

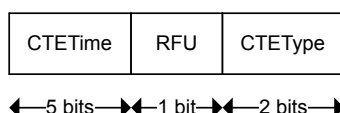


Figure 101: CTEInfo field

The CTETIME field defines the length of the CTE in 8 μ s units. The valid upper bound of values can be adjusted using [CTEINLINECONF.CTETIMEVALIDRANGE](#), including allowing use of the RFU bit within this field. If the CTETIME field is an invalid value of either 0 or 1, the CTE is assumed to be the minimum valid length of 16 μ s. The slot duration is determined by the CTEType field. In RX this determines whether the sample spacing as defined in [CTEINLINECONF.CTEINLINERXMODE1US](#) or [CTEINLINECONF.CTEINLINERXMODE2US](#) is used.

CTEType	Description	TX switch spacing	RX sample spacing during reference period	Sample spacing RX during reference period
0	AoA, no switching	-	TSAMPLESPACING1	TSAMPLESPACING2
1	AoD, 1 μ s slots	2 μ s	TSAMPLESPACING1	CTEINLINERXMODE1US
2	AoD, 2 μ s slots	4 μ s	TSAMPLESPACING1	CTEINLINERXMODE2US
3	Reserved for future use			

Table 77: Switching and sampling spacing based on CTEType

6.18.12.4 Manual configuration

If [CTEINLINECONF.CTEINLINETRLN](#) is not set, then the packet is not parsed to determine the CTE parameters, and the antenna switching and sampling is controlled by other registers, see [Antenna switching](#) on page 289. The length of the CTE is given in 8 μ s units by [DFECTRL1.NUMBEROF8US](#). The start of the antenna switching and/or sampling (denoted as an AoA/AoD procedure), can be configured to start at some trigger with an additional offset. Using [DFECTRL1.DFEINEXTENSION](#), the trigger can be configured to be the end of the CRC, or alternatively, the ADDRESS event. The additional offset for antenna switching is configured using [DFECTRL2.TSWITCHOFFSET](#). Similarly, the additional offset for antenna sampling is configured using [DFECTRL2.TSAMPLEOFFSET](#).

6.18.12.5 Receive- and transmit sequences

The addition of the CTE to the transmitted packet is illustrated in the following figure.

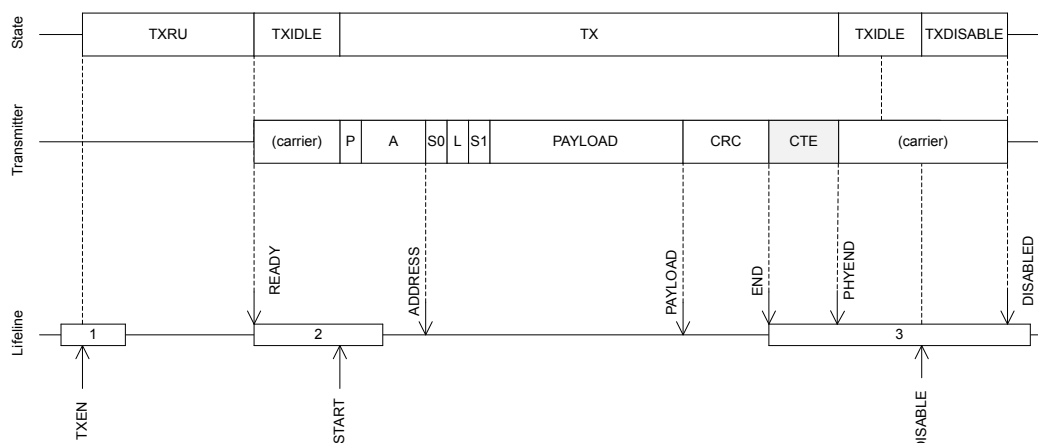


Figure 102: Transmit sequence with DFE

The presence of CTE within a received packet is signalled by the **CTEPRESENT** event illustrated in the figure below.

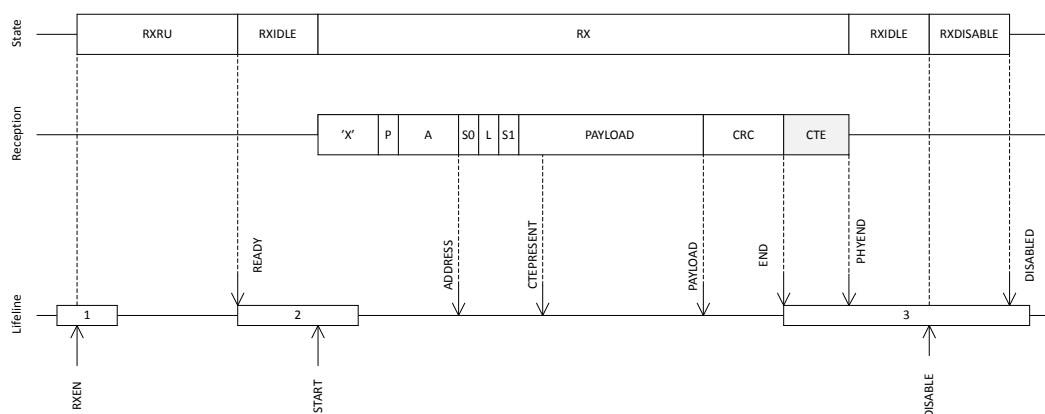


Figure 103: Receive sequence with DFE

6.18.12.6 Antenna switching

The RADIO can control up to 8 GPIO pins in order to control external antenna switches used in direction finding.

Pin configuration

The eight antenna selection signals are mapped to physical pins according to the pin numbers specified in the PSEL.DFEGPIO[n] registers. Only pins that have the PSEL.DFEGPIO[n].CONNECTED field set to *Connected* will be controlled by the RADIO. Pins that are *Disconnected* will be controlled by GPIO.

During transmission in AoD TX mode or reception in AoA RX mode, the RADIO automatically acquires the pins as needed. At times when the RADIO does not use the pin, the pin is released to its default state and controlled by the **GPIO** configuration. Thus, the pin must be configured using the **GPIO** peripheral.

Pin acquired by RADIO	Direction	Value	Comment
Yes	Output	Specified in SWITCHPATTERN	Pin acquired by RADIO, and in use for DFE.
No	Specified by GPIO	Specified by GPIO	DFE not in progress. Pin has not been acquired by RADIO, but is available for DFE use.

Table 78: Pin configuration matrix for a connected and enabled pin [n]

Switch pattern configuration

The values of the GPIOs while switching during the CTE are configured by writing successively to the [SWITCHPATTERN](#) register. The first write to [SWITCHPATTERN](#) is the GPIO pattern applied from the call of [TASKS_TXEN](#) or [TASKS_RXEN](#) until the first antenna switch is triggered. The second write sets the pattern for the reference period and is applied at the start of the guard period. The following writes set the pattern for the remaining switch slots and are applied at the start of each switch slot. If writing beyond the total number of antenna slots, the pattern will wrap to [SWITCHPATTERN\[2\]](#) and start over again. During operation, when the end of the [SWITCHPATTERN](#) buffer is reached, the RADIO cycles back to [SWITCHPATTERN\[2\]](#). At the end of the AoA/AoD procedure, [SWITCHPATTERN\[0\]](#) is applied to [DFECTRL1.TSWITCHSPACING](#) after the previous antenna switch. The [SWITCHPATTERN](#) buffer can be erased/cleared using [CLEARPATTERN](#).

A minimum number of three patterns must be written to the [SWITCHPATTERN](#) register.

If [CTEINLINECONF.CTEINLINECTRLLEN](#) is not set, then the antenna switch spacing is determined by [DFECTRL1.TSWITCHSPACING](#) (otherwise described by [Switching and sampling spacing based on CTEType](#) on page 288). [DFECTRL2.TSWITCHOFFSET](#) determines the position of the first switch compared to the configurable start of CTE (see [DFECTRL1.DFEINEXTENSION](#)).

6.18.12.7 IQ sampling

The RADIO uses DMA to write IQ samples recorded during the CTE to RAM. Alternatively, the magnitude and phase of the samples can be recorded using the [DFECTRL1.SAMPLETYPE](#) field. The samples are written to the location in RAM specified by [DFEPACKET.PTR](#). The maximum number of samples to transfer are specified by [DFEPACKET.MAXCNT](#) and the number of samples transferred are given in [DFEPACKET.AMOUNT](#). The IQ samples are recorded with respect to the RX carrier frequency. The format of the samples is provided in the following table.

SAMPLETYPE	Field	Bits	Description
0: I_Q (default)	Q	31:16	12 bits signed, sign extended to 16 bits
	I	15:0	
1: MagPhase	reserved	31:29	Always zero
	magnitude	28:16	13 bits unsigned. Equals $1.646756 \cdot \sqrt{I^2 + Q^2}$
	phase	15:0	9 bits signed, sign extended to 16 bits. Equals $64 \cdot \text{atan2}(Q, I)$ in the range [-201, 201]

Table 79: Format of samples

Oversampling is configured separately for the reference period and for the time after the reference period. During the reference period, the sample spacing is determined by [DFECTRL1.TSAMPLESPACINGREF](#). [DFECTRL2.TSAMPLEOFFSET](#) determines the position of the first sample relative to the end of the last bit of the CRC.

For the time after the reference period, if [CTEINLINECONF.CTEINLINECTRLLEN](#) is disabled, the sample spacing is set in [DFECTRL1.TSAMPLESPACING](#). However, when [CTEINLINECONF.CTEINLINECTRLLEN](#) is enabled, the sample spacing are determined by two different registers, depending on whether the device is in AoA or AoD RX-mode, as follows.

For AoD RX mode, the sample spacing after the reference period is determined by the CTEType in the packet, as listed in the table below.

CTEType	Sample spacing
AoD 1 μ s slots	CTEINLINECONF.CTEINLINERXMODE1US
AoD 2 μ s slots	CTEINLINECONF.CTEINLINERXMODE2US
Other	DFECTRL1.TSAMPLESPACING

Table 80: Sample spacing when CTEINLINECONF.CTEINLINECTRLLEN is set and the device is in AoD RX mode

For AoA RX mode, the sample spacing after the reference period is determined by DFECTRL1.TSWITCHSPACING, as listed in the table below.

DFECTRL1.TSWITCHSPACING	Sample spacing
2 μ s	CTEINLINECONF.CTEINLINERXMODE1US
4 μ s	CTEINLINECONF.CTEINLINERXMODE2US
Other	DFECTRL1.TSAMPLESPACING

Table 81: Sample spacing when CTEINLINECONF.CTEINLINECTRLLEN is set and the device is in AoA RX mode

For the reference- and switching periods, DFECTRL1.TSAMPLESPACINGREF and DFECTRL1.TSAMPLESPACING can be used to achieve oversampling.

6.18.13 IEEE 802.15.4 operation

With the `MODE=ieee802154_250kbit` the RADIO will comply with the IEEE 802.15.4-2006 standard implementing its 250 kbps, 2450 MHz, O-QPSK PHY.

The IEEE 802.15.4 standard differs from Nordic's proprietary and *Bluetooth*[®] low energy modes. Notable differences include modulation scheme, channel structure, packet structure, security, and medium access control.

The main features of the IEEE 802.15.4 mode are:

- Ultra-low power 250 kbps, 2450 MHz, IEEE 802.15.4-2006 compliant link
- Clear channel assessment
- Energy detection scan
- CRC generation

6.18.13.1 Packet structure

The IEEE 802.15.4 standard defines an on-the-air frame/packet that is different from what is used in BLE mode.

The following figure provides an overview of the physical frame structure and its timing.

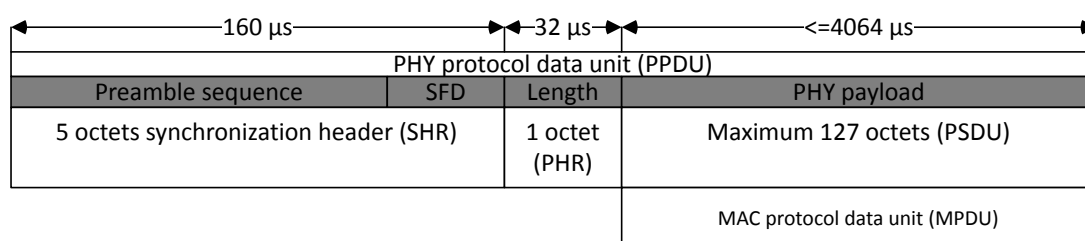


Figure 104: IEEE 802.15.4 frame format (PPDU)

The standard uses the term *octet* for an 8-bit storage unit within the PPDU. For timing, the value *symbol* is used, and it has a duration of 16 μ s.

The total usable payload (PSDU) is 127 octets, but when CRC is in use, this is reduced to 125 octets of usable payload.

The preamble sequence consists of four octets that are all zero, and are used for synchronizing the RADIO's receiver. Following the preamble is the single octet *start of frame delimiter (SFD)*, with a fixed value of 0xA7. An alternate SFD can be programmed through the [SFD](#) register, providing an initial level of frame filtering for those who choose non-standard compliance. It is a valuable feature when operating in a congested or private network. The preamble sequence and the SFD are generated by the RADIO, and are not programmed by the user into the frame buffer.

Following the five octet *synchronization header (SHR)* is the single octet *phy header (PHR)*. The least significant seven bits of PHR denote the frame length of the following PSDU. The most significant bit is reserved and is set to zero for frames that are standard compliant. The RADIO reports all eight bits which can be used to carry additional information. The PHR is the first byte written to the frame data memory pointed to by [PACKETPTR](#). Frames with zero length are discarded, and the [FRAMESTART](#) event is not generated in this case.

The next N octets carry the data of the PHY packet, where N equals the value of the PHR. For an implementation also using the IEEE 802.15.4 MAC layer, the PHY data is a MAC frame of N-2 octets, since two octets occupy a CRC field.

As illustrated in the figure below, an IEEE 802.15.4 MAC layer frame always consists of

- A header:
 - The frame control field (FCF)
 - The sequence number
 - Addressing fields
- A payload
- The 16-bit frame control sequence (FCS)

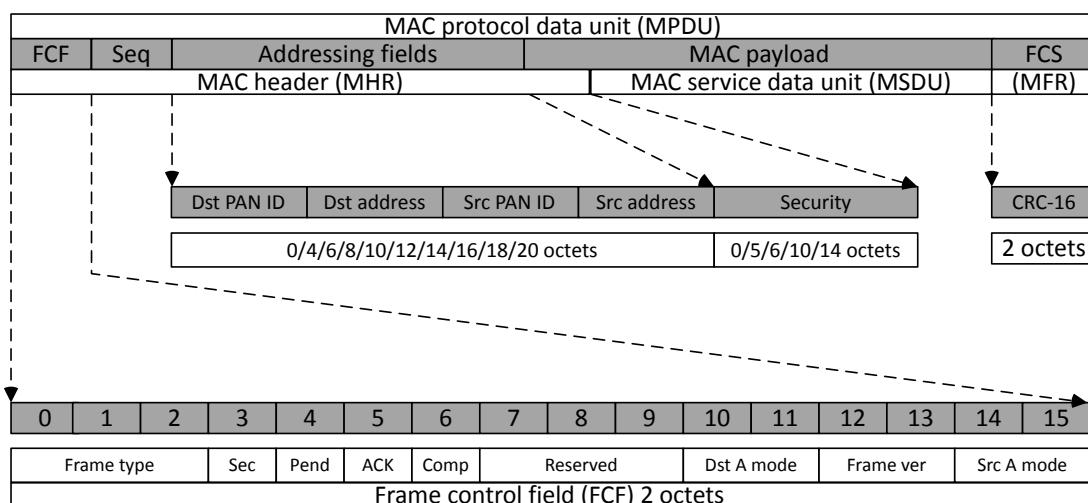


Figure 105: IEEE 802.15.4 frame format (MPDU)

The two FCF octets contain information about the frame type, addressing, and other control flags. This field is decoded when using the assisted operating modes offered by the RADIO.

The sequence number is a single octet in size and is unique for a frame. It is used in the associated acknowledgement frame sent upon successful frame reception.

The addressing field can be zero (acknowledgement frame) or up to 20 octets in size. The field is used to direct packets to the correct recipient and denote its origin. IEEE 802.15.4 bases its addressing on networks being organized in PANs with 16-bit identifier and nodes having a 16-bit or 64-bit address. In the assisted receive mode, these parameters are analyzed for address matching and acknowledgement.

The MAC payload carries the data of the next higher layer, or in the case of a MAC command frame, information used by the MAC layer itself.

The two last octets contain the 16-bit ITU-T CRC. The FCS is calculated over the MAC header (MHR) and MAC payload (MSDU) parts of the frame. This field is calculated automatically when sending a frame, or indicated in the **CRCSTATUS** register when a frame is received. If configured, this feature is taken care of autonomously by the CRC module.

6.18.13.2 Operating frequencies

The IEEE 802.15.4 standard defines 16 channels, 11 - 26, of 5 MHz each, in the 2450 MHz frequency band.

To choose the correct channel center frequency, the **FREQUENCY** register must be programmed according to the table below.

IEEE 802.15.4 channel	Center frequency (MHz)	FREQUENCY setting
Channel 11	2405	5
Channel 12	2410	10
Channel 13	2415	15
Channel 14	2420	20
Channel 15	2425	25
Channel 16	2430	30
Channel 17	2435	35
Channel 18	2440	40
Channel 19	2445	45
Channel 20	2450	50
Channel 21	2455	55
Channel 22	2460	60
Channel 23	2465	65
Channel 24	2470	70
Channel 25	2475	75
Channel 26	2480	80

Table 82: IEEE 802.15.4 center frequency definition

6.18.13.3 Energy detection (ED)

As required by the IEEE 802.15.4 standard, it must be possible to sample the received signal power within the bandwidth of a channel, for the purpose of determining presence of activity.

To prevent the channel signal from being decoded, the shortcut between the **READY** event and the **START** task should be disabled before putting the RADIO in receive mode. The energy detection (ED) measurement time, where RSSI samples are averaged, is 8 symbol periods, corresponding to 128 μ s. The standard further specifies the measurement to be a number between 0 and 255, where 0 shall indicate received power less than 10 dB above the selected receiver sensitivity. The power range of the ED values must be at least a 40 dB linear mapping with accuracy of ± 6 dB. See section 6.9.7 *Receiver ED* in the IEEE 802.15.4 standard for further details.

The following example shows how to perform a single energy detection measurement and convert to IEEE 802.15.4 scale.

IEEE 802.15.4 ED measurement example

```

#define ED_RSSISCALE 4 // From electrical specifications
uint8_t sample_ed(void)
{
    int val;
    NRF_RADIO->TASKS_EDSTART = 1; // Start
    while (NRF_RADIO->EVENTS_EDEND != 1) {
        // CPU can sleep here or do something else
        // Use of interrupts are encouraged
    }
    val = NRF_RADIO->EDSAMPLE; // Read level
    return (uint8_t)(val>63 ? 255 : val*ED_RSSISCALE); // Convert to IEEE
    802.15.4 scale
}

```

For scaling between hardware value and dBm, see equation [Conversion between hardware value and dBm](#) on page 296.

The mlme-scan.req primitive of the MAC layer uses the ED measurement to detect channels where there might be wireless activity. To assist this primitive, a tailored mode of operation is available where the ED measurement runs for a defined number of iterations keeping track of the maximum ED level. This is engaged by writing the **EDCNT** register to a value different from 0, where it will run the specified number of iterations and report the maximum energy measurement in the **EDSAMPLE** register. The scan is started with **EDSTART** task and its end indicated with the **EDEND** event. This significantly reduces the interrupt frequency and therefore power consumption. The following figure shows how the ED measurement will operate depending on the **EDCNT** register.

EDCNT = 0



EDCNT = N-1

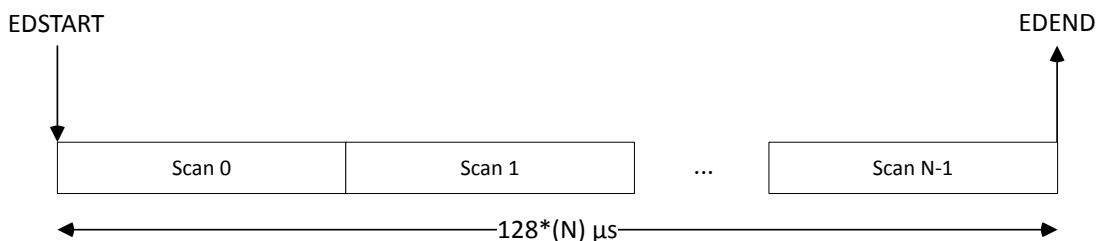


Figure 106: Energy detection measurement examples

The scan is stopped by writing the **EDSTOP** task. It will be followed by the **EDSTOPPED** event when the module has terminated.

6.18.13.4 Clear channel assessment (CCA)

IEEE 802.15.4 implements a listen-before-talk channel access method to avoid collisions when transmitting, known as *carrier sense multiple access with collision avoidance (CSMA-CA)*. The key part of this is measuring if the wireless medium is busy or not.

The following clear channel assesment modes are supported:

- **CCA Mode 1** (energy above threshold): The medium is reported busy upon detecting any energy above the ED threshold.
- **CCA Mode 2** (carrier sense only): The medium is reported busy upon detection of a signal compliant with the IEEE 802.15.4 standard with the same modulation and spreading characteristics.
- **CCA Mode 3** (carrier sense with energy above threshold): The medium is reported busy using a logical combination (AND/OR) between the results from CCA Mode 1 and CCA Mode 2.

The clear channel assessment should survey a period equal to 8 symbols or 128 μ s.

The RADIO must be in receive mode and be able to receive correct packets when performing the CCA. The shortcut between **READY** and **START** must be disabled if baseband processing is not to be performed while the measurement is running.

CCA Mode 1

CCA Mode 1 is enabled by first configuring the field **CCACTRL.CCAMODE**=EdMode and writing the **CCACTRL.CCAEDTHRES** field to a chosen value. Once the **CCASTART** task is written, the RADIO will perform a ED measurement for 8 symbols and compare the measured level with that found in the **CCACTRL.CCAEDTHRES** field. If the measured value is higher than or equal to this threshold, the **CCABUSY** event is generated. If the measured level is less than the threshold, the **CCAIDLE** event is generated.

CCA Mode 2

CCA Mode 2 is enabled by configuring **CCACTRL.CCAMODE**=CarrierMode. The RADIO will sample to see if a valid SFD is found during the 8 symbols. If a valid SFD is detected, the **CCABUSY** event is generated and the device should not send any data. The **CCABUSY** event is also generated if the scan was performed during an ongoing frame reception. In the case where the measurement period completes with no SFD detection, the **CCAIDLE** event is generated. With **CCACTRL.CCACORRCNT** not being zero, the algorithm will look at the correlator output in addition to the SFD detection signal. If a SFD is reported during the scan period, it will terminate immediately indicating busy medium. Similarly, if the number of peaks above **CCACTRL.CCACORRTHRES** crosses the **CCACTRL.CCACORRCNT**, the **CCACTRL.CCABUSY** event is generated. If less than **CCACORRCOUNT** crossings are found and no SFD is reported, the **CCAIDLE** event will be generated and the device can send data.

CCA Mode 3

CCA Mode 3 is enabled by configuring **CCACTRL.CCAMODE**=CarrierAndEdMode or **CCACTRL.CCAMODE**=CarrierOrEdMode, performing the required logical combination of the result from CCA Mode 1 and 2. The **CCABUSY** or **CCAIDLE** events are generated by ANDing or ORing the *energy above threshold* and *carrier detection* scans.

Shortcuts

An ongoing CCA can always be stopped by issuing the **CCASTOP** task. This will trigger the associated **CCASTOPPED** event.

For CCA mode automation, a number of shortcuts are available.

- To automatically switch between RX (when performing the CCA) and to TX where the packet is sent, the shortcut between **CCAIDLE** and **TXEN**, in conjunction with the short between **CCAIDLE** and **STOP** muse be used.

- To automatically disable the RADIO whenever the CCA reports a busy medium, the shortcut between **CCABUSY** and **DISABLE** can be used.
- To immediately start a CCA after ramping up into RX mode, the shortcut between **RXREADY** and **CCASTART** can be used.

Conversion

The conversion from a CCAEDTHRES, CCA, or EDLEVEL value to dBm can be done with the following equation, where $VAL_{HARDWARE}$ is the hardware-reported values, being either CCAEDTHRES, CCA or EDLEVEL, and constants ED_RSSISCALE and ED_RSSIOFFS are from electrical specifications.

$$P_{RF}[dBm] = ED_RSSIOFFS + ED_RSSISCALE \times VAL_{HARDWARE}$$

Figure 107: Conversion between hardware value and dBm

6.18.13.5 Cyclic redundancy check (CRC)

IEEE 802.15.4 uses a 16-bit ITU-T cyclic redundancy check (CRC) calculated over the MAC header (MHR) and MAC service data unit (MSDU).

The standard defines the following generator polynomial:

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

In receive mode the RADIO will trigger the CRC module when the first octet after the frame length (PHR) is received. The CRC will then update on each consecutive octet received. When a complete frame is received the **CRCSTATUS** register will be updated accordingly and the **CRCOK** or **CRCERROR** events generated. When the CRC module is enabled it will not write the two last octets (CRC) to the frame Data RAM. When transmitting, the CRC will be computed on the fly, starting with the first octet after PHR, and inserted as the two last octets in the frame. The EasyDMA will fetch frame length minus 2 octets from RAM and insert the CRC octets insitu.

Below is a code snippet for configuring the CRC module for correct operation when in IEEE 802.15.4 mode. The **CRCNCF** is written to 16-bit CRC and the **CRCPOLY** is written to 0x11021. The start value used by IEEE 802.15.4 is zero and **CRCINIT** is configured to reflect this.

```
/* 16-bit CRC with ITU-T polynomial with 0 as start condition*/
NRF_RADIO->CRCCNF = ((RADIO_CRCCNF_SKIPADDR_Ieee802154 << RADIO_CRCCNF_SKIPADDR_Pos) |
                    (RADIO_CRCCNF_LEN_Two << RADIO_CRCCNF_LEN_Pos));
NRF_RADIO->CRCPOLY = 0x11021;
NRF_RADIO->CRCINIT = 0;
```

The ENDIANESS subregister must be set to little-endian since the FCS field is transmitted from left bit to right.

6.18.13.6 Transmit sequence

The transmission is started by first putting the RADIO in receive mode and triggering the **RXEN** task.

An outline of the IEEE 802.15.4 transmission is illustrated in the figure below.

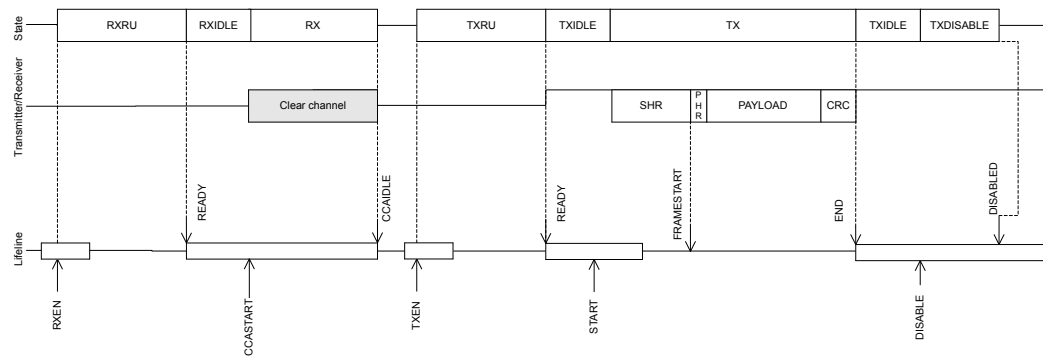


Figure 108: IEEE 802.15.4 transmit sequence

The receiver will ramp up and enter the RXIDLE state where the **READY** event is generated. Upon receiving the ready event, the CCA is started by triggering the **CCASTART** task. The chosen mode of assessment (**CCACTRL.CCAMODE** register) will be performed and signal the **CCAIDLE** or **CCABUSY** event 128 µs later. If the **CCABUSY** event is received, the RADIO will have to retry the CCA after a specific back-off period. This is outlined in the *IEEE 802.15.4 standard, Figure 69 in section 7.5.1.4 The CSMA-CA algorithm*.

If the **CCAIDLE** event is generated, a write to the **TXEN** task register enters the RADIO in TXRU state. The **READY** event will be generated when the RADIO is in TXIDLE state and ready to transmit. With the **PACKETPTR** pointing to the length (PHR) field of the frame, the **START** task can be written. The RADIO will send the four octet preamble sequence followed by the start of frame delimiter (**SFD** register). The first byte read from the Data RAM is the length field (PHR) followed by the transmission of the number of bytes indicated as the frame length. If the CRC module is configured it will run for PHR-2 octets. The last two octets will be substituted with the results from running the CRC. The necessary CRC parameters are sampled on the **START** task. The FCS field of the frame is little endian.

In addition to the already available shortcuts, one is provided between **READY** event and **CCASTART** task so that a CCA can automatically start when the receiver is ready. A second shortcut has been added between **CCAIDLE** event and the **TXEN** task, so that upon detecting a clear channel the RADIO can immediately enter transmit mode.

6.18.13.7 Receive sequence

The reception is started by first putting the RADIO in receive mode. After writing to the **RXEN** task, the RADIO will start ramping up and enter the RXRU state.

When the **READY** event is generated, the RADIO enters the RXIDLE mode. For the baseband processing to be enabled, the **START** task must be written. An outline of the IEEE 802.15.4 reception can be found in the figure below.

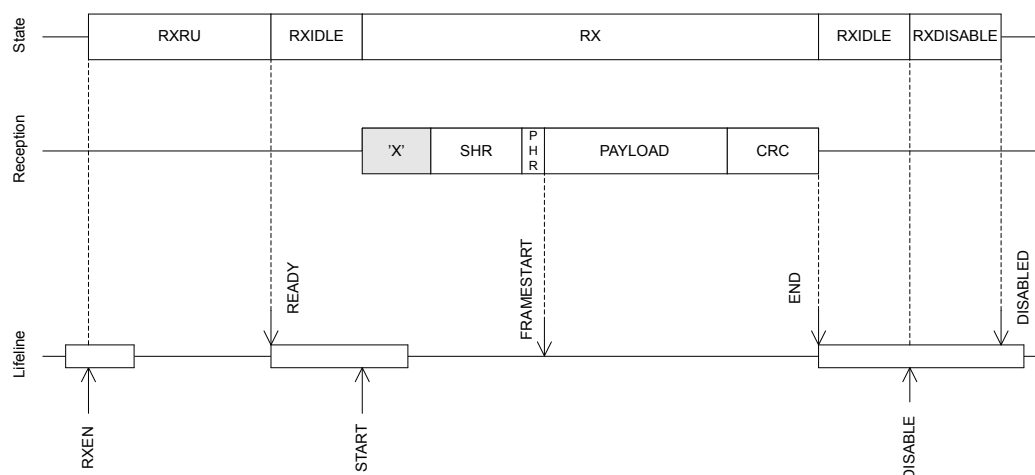


Figure 109: IEEE 802.15.4 receive sequence

When a valid SHR is received the RADIO will start storing future octets (starting with PHR) to the data memory pointed to by [PACKETPTR](#). After the SFD octet is received the [FRAMESTART](#) event is generated. If the CRC module is enabled it will start updating with the second byte received (first byte in payload) and run for the full frame length. The two last bytes in the frame are not written to RAM when CRC is configured. However, if the result of the CRC after running the full frame is zero, the [CRCOK](#) event will be generated. The [END](#) event is generated when the last octet has been received and is available in data memory.

When a packet is received a link quality indicator (LQI) is also generated and appended immediately after the last received octet. When using an IEEE 802.15.4 compliant frame, this will be just after the MSDU since the FCS is not reported. In the case of a non-compliant frame it will be appended after the full frame. The LQI reported by hardware must be converted to IEEE 802.15.4 range by an 8-bit saturating multiplication by 4, as shown in [IEEE 802.15.4 ED measurement example](#) on page 294. The LQI is only valid for frames equal to or longer than three octets. When receiving a frame the RSSI (reported as negative dB) will be measured at three points during the reception. These three values will be sorted and the middle one selected (median 3) to be remapped within the LQI range. The following figure illustrates the LQI measurement and how the data is arranged in data memory.

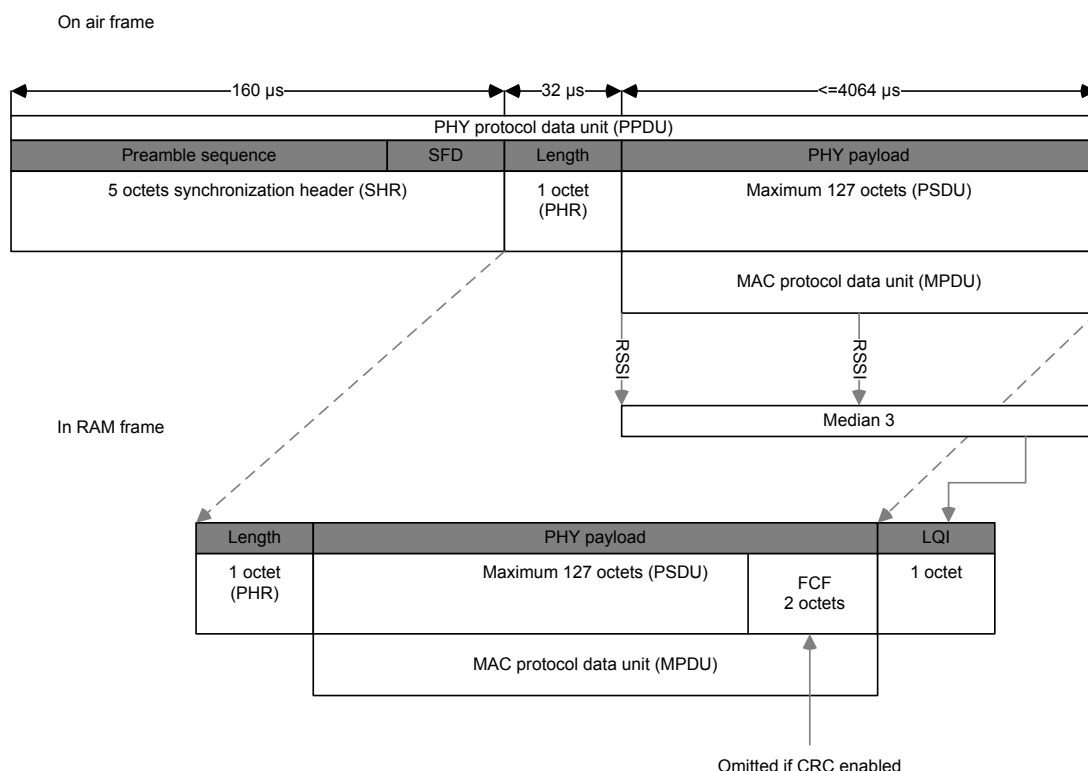


Figure 110: IEEE 802.15.4 frame in data memory

A shortcut has been added between the **FRAMESTART** event and the **BCSTART** task. This can be used to trigger a **BCMATCH** event after N bits, such as when inspecting the MAC addressing fields.

6.18.13.8 Interframe spacing (IFS)

The IEEE 802.15.4 standard defines a specific time that is allotted for the MAC sublayer to process received data. Interframe spacing (IFS) is used to prevent that two frames are transmitted too close together. If the transmission is requesting an acknowledgement, the space before the second frame shall be at least one IFS period.

The IFS is determined to be one of the following:

- IFS equals `macMinSIFSPeriod` (12 symbols) if the MPDU is less than or equal to `aMaxSIFSFrameSize` (18 octets) octets
- IFS equals `macMinLIFSPeriod` (40 symbols) if the MPDU is larger than `aMaxSIFSFrameSize`

Using the efficient assisted modes in the RADIO, the **TIFS** will be programmed with the correct value based on the frame being transmitted. If the assisted modes are not being used the user must update the **TIFS** register manually. The figure below provides details on what IFS period is valid in both acknowledged and unacknowledged transmissions.

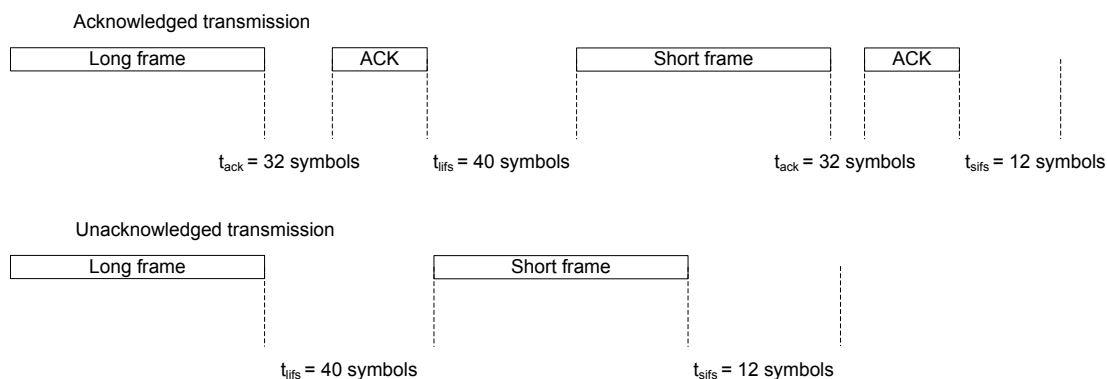


Figure 111: Interframe spacing examples

6.18.14 EasyDMA

The RADIO uses EasyDMA to read and write packets to RAM without CPU involvement.

As illustrated in [RADIO block diagram](#) on page 278, the RADIO's EasyDMA utilizes the same [PACKETPTR](#) for receiving and transmitting packets. This pointer should be reconfigured by the CPU each time before RADIO is started by the [START](#) task. The [PACKETPTR](#) register is double-buffered, meaning that it can be updated and prepared for the next transmission.

The [END](#) event indicates that the last bit has been processed by the RADIO. The [DISABLED](#) event is issued to acknowledge that a [DISABLE](#) task is done.

The structure of a packet is described in detail in [Packet configuration](#) on page 278. The data that is stored in Data RAM and transported by EasyDMA consists of the following fields:

- S0
- LENGTH
- S1
- PAYLOAD

In addition, a static add-on is sent immediately after the payload.

The size of each of the above fields in the frame is configurable (see [Packet configuration](#) on page 278), and the space occupied in RAM depends on these settings. The size of the field can be zero, as long as the resulting frame complies with the chosen RF protocol.

All fields are extended in size to align with a byte boundary in RAM. For instance, a 3-bit long field on air will occupy 1 byte in RAM while a 9-bit long field will be extended to 2 bytes.

The packet's elements can be configured as follows:

- CI, TERM1, and TERM2 fields are only present in *Bluetooth*[®] Low Energy Long Range mode
- S0 is configured through the [PCNF0.S0LEN](#) field
- LENGTH is configured through the [PCNF0.LFLEN](#) field
- S1 is configured through the [PCNF0.S1LEN](#) field
- Payload size is configured through the value in RAM corresponding to the LENGTH field
- Static add-on size is configured through the [PCNF1.STATLEN](#) field

The [PCNF1.MAXLEN](#) field configures the maximum packet payload plus add-on size in number of bytes that can be transmitted or received by the RADIO. This feature can be used to ensure that the RADIO does not overwrite, or read beyond, the RAM assigned to the packet payload. This means that if the LENGTH field of the packet payload exceeds [PCNF1.STATLEN](#), and the LENGTH field in the packet specifies a packet larger than configured in [PCNF1.MAXLEN](#), the payload will be truncated to the length specified in [PCNF1.MAXLEN](#).

Note: The `PCNF1.MAXLEN` field includes the payload and the add-on, but excludes the size occupied by the `S0`, `LENGTH`, and `S1` fields. This has to be taken into account when allocating RAM.

If the payload and add-on length is specified larger than `PCNF1.MAXLEN`, the RADIO will still transmit or receive in the same way as before, except the payload is now truncated to `PCNF1.MAXLEN`. The packet's `LENGTH` field will not be altered when the payload is truncated. The RADIO will calculate CRC as if the packet length is equal to `PCNF1.MAXLEN`.

Note: If `PACKETPTR` is not pointing to the Data RAM region, an EasyDMA transfer may result in a HardFault or RAM corruption. See [Memory](#) on page 19 for more information about the different memory regions.

The `END` event indicates that the last bit has been processed by the RADIO. The `DISABLED` event is issued to acknowledge that an `DISABLE` task is done.

6.18.15 Registers

Base address	Peripheral	Instance	Description	Configuration
0x40001000	RADIO	RADIO	2.4 GHz radio	

Table 83: Instances

Register	Offset	Description
<code>TASKS_TXEN</code>	0x000	Enable RADIO in TX mode
<code>TASKS_RXEN</code>	0x004	Enable RADIO in RX mode
<code>TASKS_START</code>	0x008	Start RADIO
<code>TASKS_STOP</code>	0x00C	Stop RADIO
<code>TASKS_DISABLE</code>	0x010	Disable RADIO
<code>TASKS_RSSISTART</code>	0x014	Start the RSSI and take one single sample of the receive signal strength
<code>TASKS_RSSISTOP</code>	0x018	Stop the RSSI measurement
<code>TASKS_BCSTART</code>	0x01C	Start the bit counter
<code>TASKS_BCSTOP</code>	0x020	Stop the bit counter
<code>TASKS_EDSTART</code>	0x024	Start the energy detect measurement used in IEEE 802.15.4 mode
<code>TASKS_EDSTOP</code>	0x028	Stop the energy detect measurement
<code>TASKS_CCASTART</code>	0x02C	Start the clear channel assessment used in IEEE 802.15.4 mode
<code>TASKS_CCASTOP</code>	0x030	Stop the clear channel assessment
<code>EVENTS_READY</code>	0x100	RADIO has ramped up and is ready to be started
<code>EVENTS_ADDRESS</code>	0x104	Address sent or received
<code>EVENTS_PAYLOAD</code>	0x108	Packet payload sent or received
<code>EVENTS_END</code>	0x10C	Packet sent or received
<code>EVENTS_DISABLED</code>	0x110	RADIO has been disabled
<code>EVENTS_DEVMATCH</code>	0x114	A device address match occurred on the last received packet
<code>EVENTS_DEVMISS</code>	0x118	No device address match occurred on the last received packet
<code>EVENTS_RSSIEND</code>	0x11C	Sampling of receive signal strength complete
<code>EVENTS_BCMATCH</code>	0x128	Bit counter reached bit count value
<code>EVENTS_CRCOK</code>	0x130	Packet received with CRC ok
<code>EVENTS_CRCERROR</code>	0x134	Packet received with CRC error
<code>EVENTS_FRAMESTART</code>	0x138	IEEE 802.15.4 length field received
<code>EVENTS_EDEND</code>	0x13C	Sampling of energy detection complete. A new ED sample is ready for readout from the <code>RADIO.EDSAMPLE</code> register
<code>EVENTS_EDSTOPPED</code>	0x140	The sampling of energy detection has stopped
<code>EVENTS_CCAIDLE</code>	0x144	Wireless medium in idle - clear to send
<code>EVENTS_CCABUSY</code>	0x148	Wireless medium busy - do not send

Register	Offset	Description
EVENTS_CCASTOPPED	0x14C	The CCA has stopped
EVENTS_RATEBOOST	0x150	Ble_LR CI field received, receive mode is changed from Ble_LR125Kbit to Ble_LR500Kbit.
EVENTS_TXREADY	0x154	RADIO has ramped up and is ready to be started TX path
EVENTS_RXREADY	0x158	RADIO has ramped up and is ready to be started RX path
EVENTS_MHRMATCH	0x15C	MAC header match found
EVENTS_SYNC	0x168	Preamble indicator
EVENTS_PHYEND	0x16C	Generated when last bit is sent on air, or received from air
EVENTS_CTEPRESENT	0x170	CTE is present (early warning right after receiving CTEInfo byte)
SHORTS	0x200	Shortcuts between local events and tasks
INTENSET	0x304	Enable interrupt
INTENCLR	0x308	Disable interrupt
CRCSTATUS	0x400	CRC status
RXMATCH	0x408	Received address
RXCRC	0x40C	CRC field of previously received packet
DAI	0x410	Device address match index
PDUSTAT	0x414	Payload status
CTESTATUS	0x44C	CTEInfo parsed from received packet
DFESTATUS	0x458	DFE status information
PACKETPTR	0x504	Packet pointer
FREQUENCY	0x508	Frequency
TXPOWER	0x50C	Output power
MODE	0x510	Data rate and modulation
PCNF0	0x514	Packet configuration register 0
PCNF1	0x518	Packet configuration register 1
BASE0	0x51C	Base address 0
BASE1	0x520	Base address 1
PREFIX0	0x524	Prefixes bytes for logical addresses 0-3
PREFIX1	0x528	Prefixes bytes for logical addresses 4-7
TXADDRESS	0x52C	Transmit address select
RXADDRESSES	0x530	Receive address select
CRC CNF	0x534	CRC configuration
CRC POLY	0x538	CRC polynomial
CRC INIT	0x53C	CRC initial value
TIFS	0x544	Interframe spacing in μ s
RSSISAMPLE	0x548	RSSI sample
STATE	0x550	Current radio state
DATAWHITEIV	0x554	Data whitening initial value
BCC	0x560	Bit counter compare
DAB[n]	0x600	Device address base segment n
DAP[n]	0x620	Device address prefix n
DACNF	0x640	Device address match configuration
MHRMATCHCONF	0x644	Search pattern configuration
MHRMATCHMAS	0x648	Pattern mask
MODECNF0	0x650	Radio mode configuration register 0
SFD	0x660	IEEE 802.15.4 start of frame delimiter
EDCNT	0x664	IEEE 802.15.4 energy detect loop count
EDSAMPLE	0x668	IEEE 802.15.4 energy detect level
CCACTRL	0x66C	IEEE 802.15.4 clear channel assessment control
DFEMODE	0x900	Whether to use Angle-of-Arrival (AOA) or Angle-of-Departure (AOD)
CTEINLINECONF	0x904	Configuration for CTE inline mode
DFECTRL1	0x910	Various configuration for Direction finding
DFECTRL2	0x914	Start offset for Direction finding
SWITCHPATTERN	0x928	GPIO patterns to be used for each antenna

Register	Offset	Description
CLEARPATTERN	0x92C	Clear the GPIO pattern array for antenna control
PSEL.DFEGPIO[0]	0x930	Pin select for DFE pin 0
PSEL.DFEGPIO[1]	0x934	Pin select for DFE pin 1
PSEL.DFEGPIO[2]	0x938	Pin select for DFE pin 2
PSEL.DFEGPIO[3]	0x93C	Pin select for DFE pin 3
PSEL.DFEGPIO[4]	0x940	Pin select for DFE pin 4
PSEL.DFEGPIO[5]	0x944	Pin select for DFE pin 5
PSEL.DFEGPIO[6]	0x948	Pin select for DFE pin 6
PSEL.DFEGPIO[7]	0x94C	Pin select for DFE pin 7
DFEPACKET.PTR	0x950	Data pointer
DFEPACKET.MAXCNT	0x954	Maximum number of buffer words to transfer
DFEPACKET.AMOUNT	0x958	Number of samples transferred in the last transaction
POWER	0xFFC	Peripheral power control

Table 84: Register overview

6.18.15.1 TASKS_TXEN

Address offset: 0x000

Enable RADIO in TX mode

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W TASKS_TXEN			Enable RADIO in TX mode																															
		Trigger	1	Trigger task																															

6.18.15.2 TASKS_RXEN

Address offset: 0x004

Enable RADIO in RX mode

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field		Value ID	Value	Description																														
A	W	TASKS_RXEN			Enable RADIO in RX mode																														
			Trigger	1	Trigger task																														

6.18.15.3 TASKS_START

Address offset: 0x008

Start RADIO

Bit number			31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID			A																															
Reset 0x00000000			0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																														
A	W	TASKS_START		Start RADIO																														
		Trigger	1	Trigger task																														

6.18.15.4 TASKS_STOP

Address offset: 0x00C

Stop RADIO

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W TASKS_STOP			Stop RADIO																															
		Trigger	1	Trigger task																															

6.18.15.5 TASKS_DISABLE

Address offset: 0x010

Disable RADIO

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W	TASKS_DISABLE		Disable RADIO																															
		Trigger	1	Trigger task																															

6.18.15.6 TASKS_RSSISTART

Address offset: 0x014

Start the RSSI and take one single sample of the receive signal strength

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field	Value ID	Value	Description																															
A	W	TASKS_RSSISTART		Start the RSSI and take one single sample of the receive signal strength																															
		Trigger	1	Trigger task																															

6.18.15.7 TASKS_RSSISTOP

Address offset: 0x018

Stop the RSSI measurement

Bit number				31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0																															
ID				A																															
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																															
ID	Acce Field		Value ID	Value	Description																														
A	W	TASKS_RSSISTOP			Stop the RSSI measurement																														
		Trigger	1	Trigger task																															

6.18.15.8 TASKS_BCSTART

Address offset: 0x01C

Start the bit counter

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID																																			A	
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
ID	Acce Field	Value ID	Value		Description																															
A	W	TASKS_BCSTART			Start the bit counter																															
		Trigger	1		Trigger task																															

6.18.15.9 TASKS_BCSTOP

Address offset: 0x020

Stop the bit counter

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																		A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value		Description																													
A	W	TASKS_BCSTOP			Stop the bit counter																													
		Trigger	1		Trigger task																													

6.18.15.10 TASKS_EDSTART

Address offset: 0x024

Start the energy detect measurement used in IEEE 802.15.4 mode

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID																																				A	
Reset 0x00000000				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
ID	Acce Field		Value ID	Value		Description																															
A	W	TASKS_EDSTART				Start the energy detect measurement used in IEEE 802.15.4 mode																															
		Trigger		1		Trigger task																															

6.18.15.11 TASKS_EDSTOP

Address offset: 0x028

Stop the energy detect measurement

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID																																		A	
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ID	Acce Field	Value ID	Value		Description																														
A	W	TASKS_EDSTOP			Stop the energy detect measurement																														
		Trigger	1		Trigger task																														

6.18.15.12 TASKS_CCSTART

Address offset: 0x02C

Start the clear channel assessment used in IEEE 802.15.4 mode

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																																		A
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value		Description																													
A	W	TASKS_CCSTART			Start the clear channel assessment used in IEEE 802.15.4 mode																													
		Trigger	1		Trigger task																													

6.18.15.13 TASKS_CCSTOP

Address offset: 0x030

Stop the clear channel assessment

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
ID																																			A	
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
ID	Acce Field	Value ID	Value		Description																															
A	W	TASKS_CCSTOP			Stop the clear channel assessment																															
		Trigger	1		Trigger task																															

6.18.15.14 EVENTS_READY

Address offset: 0x100

RADIO has ramped up and is ready to be started

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
ID																																						A	
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
ID	Acce Field	Value ID	Value		Description																																		
A	RW	EVENTS_READY			RADIO has ramped up and is ready to be started																																		
		NotGenerated	0		Event not generated																																		
		Generated	1		Event generated																																		

6.18.15.15 EVENTS_ADDRESS

Address offset: 0x104

Address sent or received

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
ID			A																																				
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
ID	Acce Field	Value ID	Value		Description																																		
A	RW	EVENTS_ADDRESS			Address sent or received																																		
		NotGenerated	0		Event not generated																																		
		Generated	1		Event generated																																		

6.18.15.16 EVENTS_PAYLOAD

Address offset: 0x108

Packet payload sent or received

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A																															
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value		Description																													
A	RW	EVENTS_PAYLOAD			Packet payload sent or received																													
		NotGenerated	0		Event not generated																													
		Generated	1		Event generated																													

6.18.15.17 EVENTS_END

Address offset: 0x10C

Packet sent or received

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
ID																																						A	
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
ID	Acce Field	Value ID	Value		Description																																		
A	RW	EVENTS_END			Packet sent or received																																		
		NotGenerated	0		Event not generated																																		
		Generated	1		Event generated																																		

6.18.15.18 EVENTS_DISABLED

Address offset: 0x110

RADIO has been disabled

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID			A																																
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ID	Acce Field	Value ID	Value	Description																															
A	RW	EVENTS_DISABLED		RADIO has been disabled																															
		NotGenerated	0	Event not generated																															
		Generated	1	Event generated																															

6.18.15.19 EVENTS_DEVMATCH

Address offset: 0x114

A device address match occurred on the last received packet

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID			A																															
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ID	Acce Field	Value ID	Value	Description																														
A	RW	EVENTS_DEVMATCH		A device address match occurred on the last received packet																														
		NotGenerated	0	Event not generated																														
		Generated	1	Event generated																														

6.18.15.20 EVENTS_DEVMISS

Address offset: 0x118

No device address match occurred on the last received packet

Bit number				31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID				A																																
Reset 0x00000000				0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																																
ID	Acce	Field	Value ID	Value	Description																															
A	RW	EVENTS_DEVMISS			No device address match occurred on the last received packet																															
			NotGenerated	0	Event not generated																															
			Generated	1	Event generated																															

6.18.15.21 EVENTS_RSSIEND

Address offset: 0x11C

Sampling of receive signal strength complete

A new RSSI sample is ready for readout from the RADIO.RSSISAMPLE register

Bit number			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
ID			A																																
Reset 0x00000000			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
ID	Acce Field	Value ID	Value		Description																														
A	RW	EVENTS_RSSIEND			Sampling of receive signal strength complete																														
					A new RSSI sample is ready for readout from the RADIO.RSSISAMPLE register																														
		NotGenerated	0		Event not generated																														
		Generated	1		Event generated																														

6.18.15.22 EVENTS_BCMATCH

Address offset: 0x128

Bit counter reached bit count value

Bit counter value is specified in the RADIO.BCC register