BlueNRG-2

Bluetooth® Low Energy wireless system-on-chip

Product status link
BlueNRG-2

Features

- Bluetooth Low Energy single-mode system-on-chip compliant with Bluetooth v5.0 specifications:
  - master, slave and multiple simultaneous roles
  - LE data packet length extension
- Operating supply voltage: from 1.7 to 3.6 V
- Integrated linear regulator and DC-DC step-down converter
- Operating temperature range: -40 °C to 105 °C
- High performance, ultra-low power Cortex-M0 32-bit based architecture core
- Programmable 256 kB Flash
- 24 kB RAM with retention (two 12 kB banks)
- 1 x UART interface
- 1 x SPI interface
- 2 x I²C interface
- 14, 15 or 26 GPIOs
- 2 x multifunction timer
- 10-bit ADC
- Watchdog and RTC
- DMA controller
- PDM stream processor
- 16 or 32 MHz crystal oscillator
- 32 kHz crystal oscillator
- 32 kHz ring oscillator
- Battery voltage and temperature sensors
- Up to +8 dBm available output power (at antenna connector)
- Excellent RF link budget (up to 96 dB)
- Accurate RSSI to allow power control
- 8.3 mA TX current (@ -2 dBm, 3.0 V)
- Down to 1 µA current consumption with active BLE stack (sleep mode)
- ST companion integrated balun/filter chips are available
- Average advertisement current consumption 15.34 µA (advertisement interval 1000 ms) – 1 year, 8 months, 19 days with 230 mAh battery (CR2032)
- Average connection current consumption 7.059 µA (connection interval 1000 ms) – 3 years, 10 months, 12 days with 230 mAh battery (CR2032)
- Suitable for building applications compliant with the following radio frequency regulations: ETSI EN 300 328, EN 300 440, FCC CFR47 part 15, ARIB STD-T66
- Pre-programmed bootloader via UART
- QFN32, QFN48 and WCSP34 package options

Applications

- Watches

- Fitness, wellness and sports
- Consumer medical
- Security/proximity
- Remote control
- Home and industrial automation
- Assisted living
- Mobile phone peripherals
- Lighting
- PC peripherals

1 Description

The BlueNRG-2 is a very low power Bluetooth Low Energy (BLE) single-mode system-on-chip, compliant with Bluetooth specifications.

The BlueNRG-2 extends the features of award-winning BlueNRG network processor, enabling the usage of the embedded Cortex M0 to run the user application code.

The BlueNRG-2 includes 256 kB of programming Flash memory, 24 kB of static RAM memory with retention (two 12 kB banks) and SPI, UART, I²C standard communication interface peripherals. It also features multifunction timers, watchdog, RTC and DMA controller.

An ADC is available to interface with analog sensors, and to read the measurement of the integrated battery voltage sensor. A digital filter is available to process a PDM stream.

The BlueNRG-2 offers the same excellent RF performance of the BlueNRG radio, and the integrated high efficiency DC-DC converter keeps the same ultra-low power characteristics, but the BlueNRG-2 improves the BlueNRG sleep mode current consumption allowing a further increase in the battery lifetime of the applications.

Figure 1. BlueNRG-2 architecture

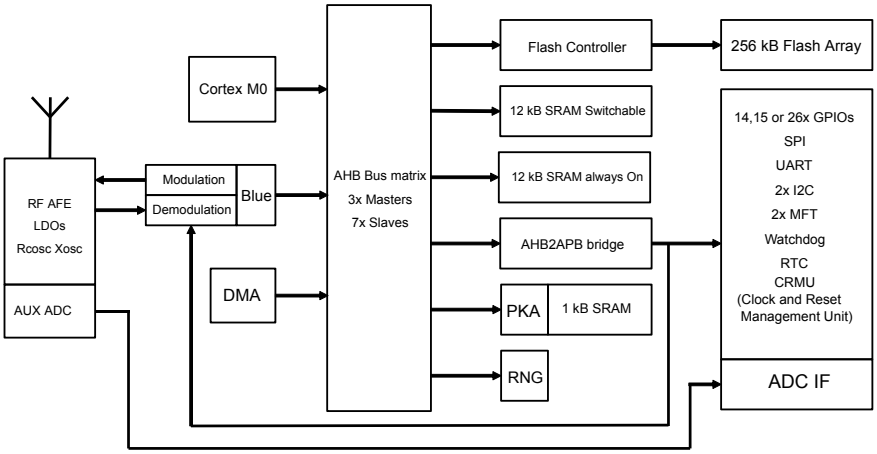
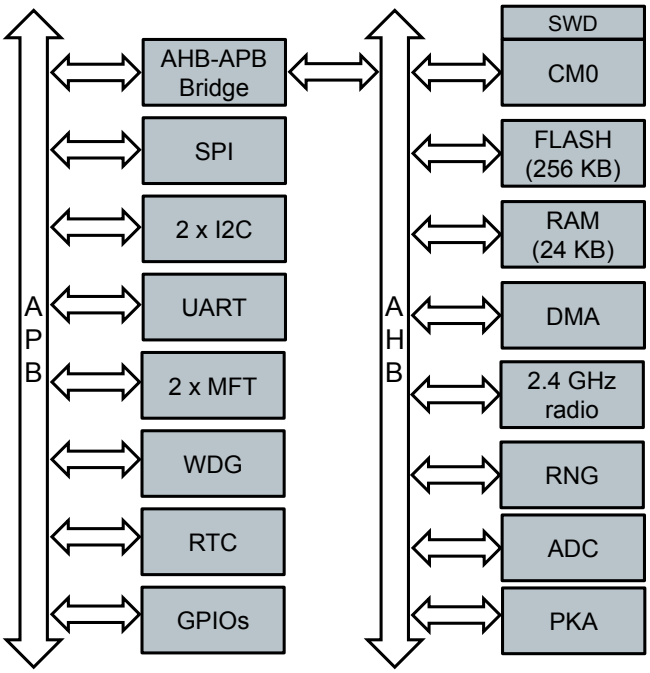


Figure 2. **BlueNRG-2 bus architecture**



2 BlueNRG-2 Bluetooth Low Energy stack

The BlueNRG-2 is complemented with a Bluetooth Low Energy stack C library that provides:

- Master, slave role support
- GAP: central, peripheral, observer or broadcaster roles
- ATT/GATT: client and server
- SM: privacy, authentication and authorization
- L2CAP
- Link layer: AES-128 encryption and decryption

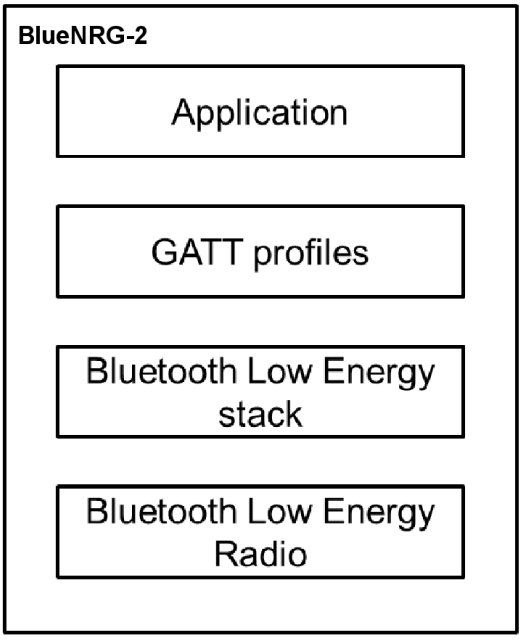
The BlueNRG-2 can be configured to support single chip or network processor applications.

The BlueNRG-2 supports LE data packet length extension, in compliance with Bluetooth smart v5.0.

In the first configuration, the BlueNRG-2 operates as single device in the application for managing both the application code and the Bluetooth Low Energy stack. The whole Bluetooth low energy stack is provided as object code in a single library file whereas the GATT low energy profiles are provided as object codes in separate libraries.

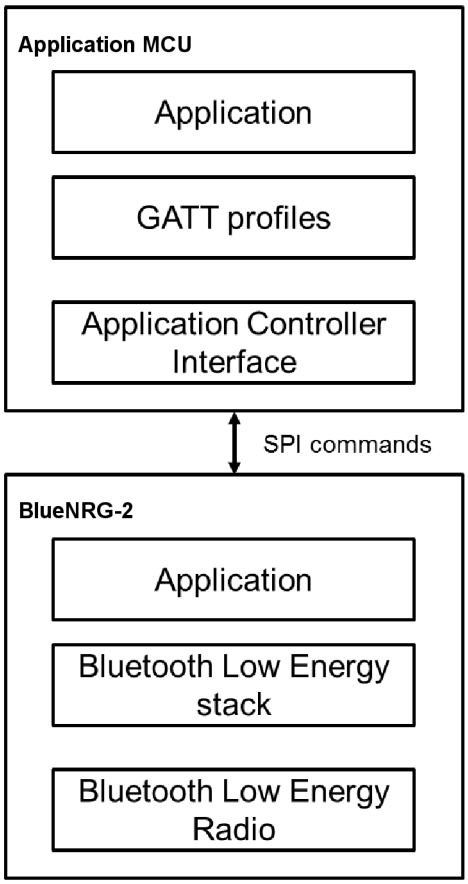
The figure below shows the single chip RF software layers.

Figure 3. BlueNRG-2 single-chip RF software layers



The BlueNRG-2 can be configured to operate as a network coprocessor. In this case, dedicated firmware is provided to support the interface with an external application processor. The whole Bluetooth low energy stack runs in the BlueNRG-2; the GATT profiles are provided to run in the application processor together with the application code. The figure below shows the network processor RF software layers.

Figure 4. **BlueNRG-2 network processor RF software layers**



3 Functional details

The BlueNRG-2 integrates:

- ARM Cortex-M0 core
- Interrupts management
- 256 kB Flash memory
- 24 kB of RAM with two retention options (12 kB or 24 kB)
- Power management
- Clocks
- Bluetooth low energy radio
- Random number generator (RNG) (reserved for Bluetooth low energy protocol stack, but user applications can read it)
- Public key cryptography (PKA) (reserved for Bluetooth low energy protocol stack)
- Peripherals:
  - SPI interface
  - UART interface
  - I²C bus interface
  - GPIO
  - Multifunction timer
  - DMA controller
  - Watchdog
  - RTC
  - ADC with battery voltage sensor and temperature sensor
  - PDM stream processor

3.1 Core

The ARM® Cortex®-M0 processor has been developed to provide a low-cost platform that meets the needs of MCU implementation, with a reduced pin count and low-power consumption, while delivering outstanding computational performance and an advanced system response to interrupts.

The ARM® Cortex®-M0 32-bit RISC processor features exceptional code-efficiency, delivering the high-performance expected from an ARM core in the memory size usually associated with 8-bit and 16-bit devices.

The BlueNRG-2 has an embedded ARM core and is therefore compatible with all ARM tools and software.

3.2 Interrupts

The Cortex-M0 nested vector interrupt controller (NVIC) handles interrupts. The NVIC controls specific Cortex-M0 interrupts (address 0x00 to 0x3C) as well as 32-user interrupts (address 0x40 to 0xBC). In the BlueNRG-2 device, the user interrupts are connected to the interrupt signals of the different peripherals.

Table 1. BlueNRG-2 interrupt vectors

Position	Priority	Priority type	Description	Address
			Initial main SP	0x0000_0000
	-3	Fixed	Reset handler	0x0000_0004
	-2	Fixed	NMI handler	0x0000_0008
	-1	Fixed	HardFault handler	0x0000_000C
			RESERVED	0x0000_000C – 0x0000_0028
	3	Settable	SVC handler	0x0000_002C

Position	Priority	Priority type	Description	Address
			RESERVED	0x0000_0030 - 0x0000_0034
	5	Settable	PendSV handler	0x0000_0038
	6	Settable	SystemTick handler	0x0000_003C
0	Init 0	Settable	GPIO interrupt	0x0000_0040
1	Init 0	Settable	FLASH controller interrupt	0x0000_0044
2	Init 0	Settable	RESERVED	0x0000_0048
3	Init 0	Settable	RESERVED	0x0000_004C
4	Init 0	Settable	UART interrupt	0x0000_0050
5	Init 0	Settable	SPI interrupt	0x0000_0054
6	Init 0	CRITICAL	BLE controller interrupt	0x0000_0058
7	Init 0	Settable	Watchdog interrupt	0x0000_005C
8	Init 0	Settable	RESERVED	0x0000_0060
9	Init 0	Settable	RESERVED	0x0000_0064
10	Init 0	Settable	RESERVED	0x0000_0068
11	Init 0	Settable	RESERVED	0x0000_006C
12	Init 0	Settable	RESERVED	0x0000_0070
13	Init 0	Settable	ADC interrupt	0x0000_0074
14	Init 0	Settable	I2C 2 interrupt	0x0000_0078
15	Init 0	Settable	I2C 1 interrupt	0x0000_007C
16	Init 0	Settable	RESERVED	0x0000_0080
17	Init 0	Settable	MFT1 A interrupt	0x0000_0084
18	Init 0	Settable	MFT1 B interrupt	0x0000_0088
19	Init 0	Settable	MFT2 A interrupt	0x0000_008C
20	Init 0	Settable	MFT2 B interrupt	0x0000_0090
21	Init 0	Settable	RTC interrupt	0x0000_0094
22	Init 0	Settable	PKA interrupt	0x0000_0098
23	Init 0	Settable	DMA interrupt	0x0000_009C
24 – 31	Init 0	Settable	RESERVED	0x0000_00A0 – 0x0000_00BC

3.3 Memories

The memory subsystem consists 256 kB Flash memory and two banks of 12 kB ultra-low leakage static RAM blocks.

The 256 kB Flash memory is available to the user and can be accessed per 32-bit for read access and per 32-bit for write access (with 4x32-bit FIFO).

The access to the static RAM can be bytes, half words (16 bits) or words (32 bits).

The two banks of 12 kB RAM blocks are always in retention mode.

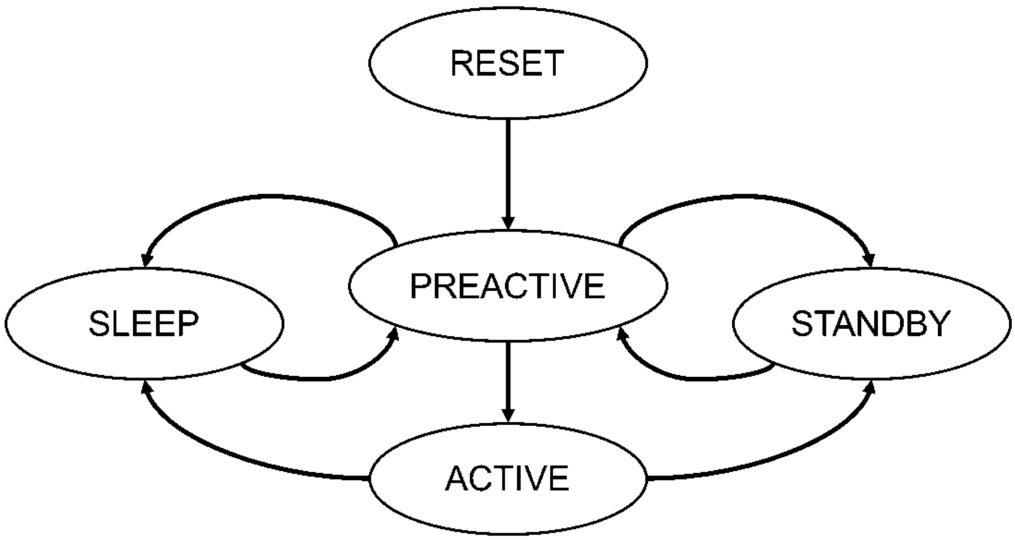
3.4 Power management

The BlueNRG-2 integrates both a low dropout voltage regulator (LDO) and a step-down DC-DC converter to supply the internal BlueNRG-2 circuitry.

The BlueNRG-2 most efficient power management configuration is with DC-DC converter active where best power consumption is obtained without compromising performances. Nevertheless, a configuration based on LDO can also be used, if needed.

A simplified version of the state machine is shown below.

Figure 5. BlueNRG-2 power management state machine



3.4.1 State description

3.4.1.1 *Preactive state*

The preactive state is the default state after a POR event.

In this state:

- All the digital power supplies are stable.
- The high frequency clock runs on internal fast clock RC oscillator (16 MHz).
- The low frequency clock runs on internal RC oscillator (32.768 kHz).

3.4.1.2 *Active state*

In this state:

- The high frequency runs on the accurate clock (16 MHz ±50 ppm) provided by the external XO. The internal fast clock RO oscillator is switched off.

3.4.1.3 *Standby state*

In this state:

- Only the digital power supplies necessary to keep the RAM in retention are used.

The wake-up from this low power state is driven by the following sources:

- IO9
- IO10
- IO11
- IO12
- IO13(1)

1. *Not available on WCSP34.*

If they have been programmed as wake-up source in the system controller registers.

3.4.1.4 *Sleep state*

In this state:

- Only the digital power supplies necessary to keep the RAM in retention are used.
- The low frequency oscillator is switched on.

The wake-up from this low power state is driven by the following sources:

- IO9
- IO10
- IO11
- IO12
- IO13 (1)

1. *Not available on WCSP34.*

If they have been programmed as wake-up source in the system controller registers and from the internal timers of the BLE radio.

3.4.1.5 *GPIO management during low power modes*

3.4.1.5.1 IO wake-up sources

The IOs programmed to be wake-up sources need an external drive according to the selected level sensitivity.

If the wake-up level is high level, a pull-down drive should be used. If the wake-up level is low level, a pull-up drive should be used.

If no external drive is applied, IO9, IO10 and IO11 are only sensitive to low level as they have an internal pull-up (activated by default). IO12 and IO13 do not have an internal pull and therefore require an external drive.

3.4.1.5.2 Wake-up time

The wake-up time is typically 200 µs at 3.0 V and a temperature of 25 °C.

3.4.1.5.3 GPIO special settings in low power modes

The GPIO9, GPIO10 and GPIO11 can be configured as output GPIO during sleep and standby mode. In addition, they can have enabled their internal pull. Their configuration is done in specific registers:

- SLEEPIO_OEN: has the same functionality of the register OEN (GPIO peripheral) and it is used to configure in output mode or input mode (default)
- SLEEPIO_PE: has the same functionality of the register PE (GPIO peripheral) and it is used to enable the internal pull. This register allows the internal pull of these IOs to be enabled or disabled also if they are not configured in output state.
- SLEEPIO_DS: has the same functionality of the register DS (GPIO peripheral) and it is used to configure the drive strength
- SLEEPIO_OUT: has the same functionality of the register DATA (GPIO peripheral) and it is used to set the state of the GPIO (high state or low state)

Note: If the GPIO9, GPIO10 or GPIO11 are used as wake-up source, then the SLEEPIO_OEN has no effect, but it is always possible to enable or disable the internal pull.

3.4.2 Power saving strategy

The application power saving strategy is based on clock stopping, dynamic clock gating,

digital power supply switch off and analog current consumption minimization.

A summary of functional blocks versus the BlueNRG-2 states is provided below.

Table 2. Relationship between the BlueNRG-2 states and functional blocks

Functional blocks	RESET	STANDBY	SLEEP	Preactive	Active	LOCK RX/ LOCK TX	RX	TX
LDO_SOFT_1V2 or LDO_SOFT_0V9	OFF	ON	ON	ON	ON	ON	ON	ON
LDO_STRONG_1V2	OFF	OFF	OFF	ON	ON	ON	ON	ON
LDO_DIG_1V8	OFF	OFF	OFF	ON	ON	ON	ON	ON
SMPS	OFF	OFF	OFF	ON	ON	ON	ON	ON

Functional blocks	RESET	STANDBY	SLEEP	Preactive	Active	LOCK RX/ LOCK TX	RX	TX
LDO_DIG_1V2	OFF	OFF	OFF	ON	ON	ON	ON	ON
BOR	OFF	OFF	OFF	ON	ON	ON	ON	ON
16 MHz RO	OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF
16 MHz XO	OFF	OFF	OFF	OFF	ON	ON	ON	ON
32 kHz RO or XO	OFF	OFF	ON	ON	ON	ON	ON	ON

3.4.3 System controller registers

SYSTEM_CTRL peripheral base address (SYSTEM_CTRL_BASE_ADDR) 0x40200000.

Table 3. SYSTEM_CTRL registers

Address offset	Name	RW	Reset	Description
0x00	WKP_IO_IS	RW	0x00000000	Level selection for wake-up IO (1 bit for IO). 0: The system wakes up when IO is low. 1: The system wakes up when IO is high.
0x04	WKP_IO_IE	RW	0x00000007	Enables the IO that wakes up the device (1 bit for IO). 0: The wake-up feature on the IO is disabled. 1: The wake-up feature on the IO is enabled.
0x08	CTRL	RW	0x00000000	XO frequency indication is provided by the application. Refer to the detailed description below.
0x0C	SLEEPIO_OEN	RW	0x00000007	GPIO output enable register for low power modes.
0x10	SLEEPIO_OUT	RW	0x00000000	GPIO output value register for low power modes.
0x14	SLEEPIO_DS	RW	0x00000000	GPIO drive strength control register for low power modes.
0x18	SLEEPIO_PE	RW	0x00000000	GPIO pull enable register for low power modes.

Table 4. SYSTEM_CTRL - WKP_IO_IS register description: address offset SYSTEM_CTRL_BASE_ADDR+0x00

Bit	Field name	Reset	RW	Description
4:0	LEVEL_SEL	0x00	RW	Selects the active wake-up level for the five IOs. 0: The system wakes up when IO is low. 1: The system wakes up when IO is high. One bit by IO: Bit0: IO9 Bit1: IO10 Bit2: IO11 Bit3: IO12 Bit4: IO13
31:5	RESERVED	0x00	RW	RESERVED

Table 5. SYSTEM_CTRL - WKP_IO_IE register description: address offset SYSTEM_CTRL_BASE_ADDR+0x04

Bit	Field name	Reset	RW	Description
4:0	IO_WAKEUP_EN	0x07	RW	Enables the IOs to be wake-up source. 0: Wake-up on the IO disabled. 1: Wake-up on the IO enabled. One bit by IO: Bit0: IO9 Bit1: IO10 Bit2: IO11 Bit3: IO12 Bit4: IO13
31:5	RESERVED	0x00	RW	RESERVED

Table 6. SYSTEM_CTRL - CTRL register description: address offset SYSTEM_CTRL_BASE_ADDR+0x08

Bit	Field name	Reset	RW	Description
0	MHZ32_SEL	0x0	RW	Indicates the crystal frequency used in the application. 0: The 16 MHz is selected. 1: The 32 MHz is selected.
31:1	RESERVED	0x0	RW	RESERVED

Table 7. SYSTEM_CTRL - SLEEPIO_OEN register description: address offset SYSTEM_CTRL_BASE_ADDR+0x0C

Bit	Field name	Reset	RW	Description
2:0	SLEEPIO_OEN	0x07	RW	Enables the IOs to act as output during low power modes. 0: output mode. 1: input mode. One bit by IO: Bit0: IO9 Bit1: IO10 Bit2: IO11
31:4	RESERVED	0x00	RW	RESERVED

Table 8. SYSTEM_CTRL – SLEEPIO_OUT register description: address offset SYSTEM_CTRL_BASE_ADDR+0x10

Bit	Field name	Reset	RW	Description
2:0	SLEEPIO_OUT	0x00	RW	Writing to a bit drives the written value on the corresponding IO when it is configured in output direction in SLEEPIO_OEN register. Reading a bit in this register returns the last written value. One bit by IO: Bit0: IO9 Bit1: IO10 Bit2: IO11
31:4	RESERVED	0x00	RW	RESERVED

Table 9. SYSTEM_CTRL - SLEEPIO_DS register description: address offset SYSTEM_CTRL_BASE_ADDR+0x14

Bit	Field name	Reset	RW	Description
2:0	SLEEPIO_DS	0x00	RW	Configure the drive strength during low power modes for the IO9, IO10 and IO11. 0: low drive strength. 1: high drive strength One bit by IO: Bit0: IO9 Bit1: IO10 Bit2: IO11
31:4	RESERVED	0x00	RW	RESERVED

Table 10. SYSTEM_CTRL - SLEEPIO_PE register description: address offset SYSTEM_CTRL_BASE_ADDR+0x18

Bit	Field name	Reset	RW	Description
2:0	SLEEPIO_PE	0x00	RW	Enable/disable the internal pull during low power modes for the IO9, IO10 and IO11. 0: pull disabled. 1: pull enabled. One bit by IO: Bit0: IO9 Bit1: IO10 Bit2: IO11
31:4	RESERVED	0x00	RW	RESERVED

AHBUPCONV peripheral base address (AHBUPCONV_BASE_ADDR) 0x40C00000.

Table 11. AHBUPCONV registers

Address offset	Name	RW	Reset	Description
0x00	COMMAND	RW	0x00000000	AHB up/down converter command register

BLUE_CTRL peripheral base address (BLUE_CTRL_BASE_ADDR) 0x48000000.

Table 12. BLUE_CTRL registers

Address offset	Name	RW	Reset	Description
0x04	TIMEOUT	RW	0x00000000	Timeout programming register
0x0C	RADIO_CONFIG	RW	0x00000000	Radio configuration register

Note: All RESERVED fields inside registers must always be written with their default value.

3.5 Clocks and reset management

The BlueNRG-2 embeds an RC low-speed frequency oscillator at 32 kHz and an RO high-speed frequency oscillator at 16 MHz.

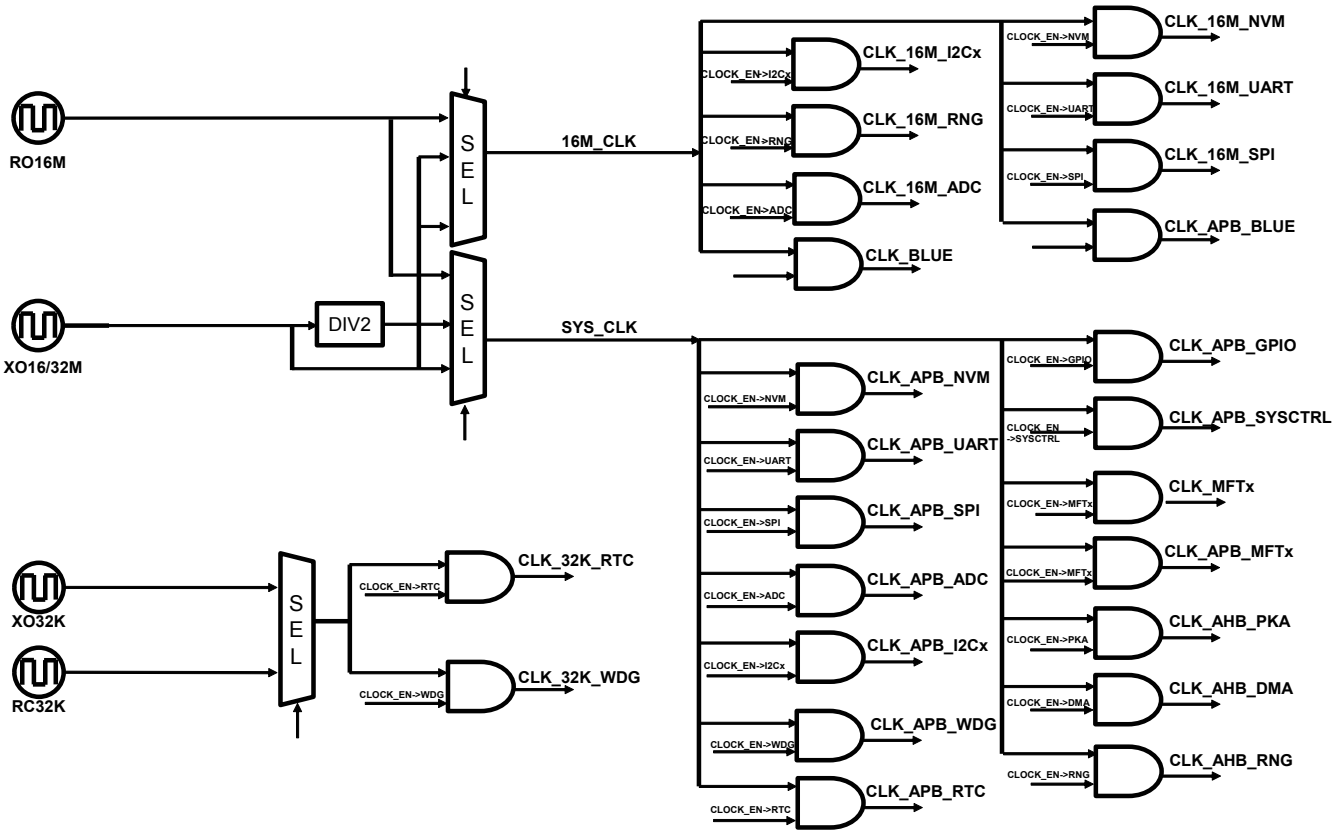
The low-frequency clock is used in low power mode and can be supplied either by a 32.7 kHz oscillator that uses an external crystal and guarantees up to ±50 ppm frequency tolerance, or by a ring oscillator, which does not require any external components.

The primary high-speed frequency clock is a 16 MHz or 32 MHz crystal oscillator. A fast-starting 16 MHz ring oscillator provides the clock while the crystal oscillator is starting up. Frequency tolerance of high-speed crystal oscillator is ±50 ppm.

The usage of the 16 MHz (for constraints related to the 16 MHz high-speed crystal usage, refer to the BlueNRG-1 DK SW release notes) (or 32 MHz) crystal is strictly necessary for RF communications.

The clock tree for the peripherals is as follows:

Figure 6. Clock tree



Note: When 32 MHz XO is used, the Cortex-M0, the DMA and the APB tree (except for BLE radio access) run at 32 MHz. The rest of the clock tree is divided by two and is at 16 MHz.

The following clocks can be enabled/disabled by software to implement optimal power consumption:

- DMA
- BLE controller
- BLE clock generator
- RNG
- Flash controller
- GPIO
- System controller
- UART
- SPI
- I2C1(1)
- I2C2
- ADC
- MFT1
- MFT2

- RTC
- WDG
- PKA

1. *The I²C1 is not available with WLCSP34 package.*

By default, all the peripheral APB and AHB clocks are enabled, except for the PKA peripheral. The following clocks are enabled/disabled automatically:

- Processor clock (disabled in sleep mode)
- RAM clock (disabled if processor clock, SPI clock and BLE clock are all disabled)

Note: It is possible to provide an external 32 kHz signal to the BlueNRG-2 device through the SXTAL0 pin by sourcing a periodic waveform from 0 to 1.2 V.

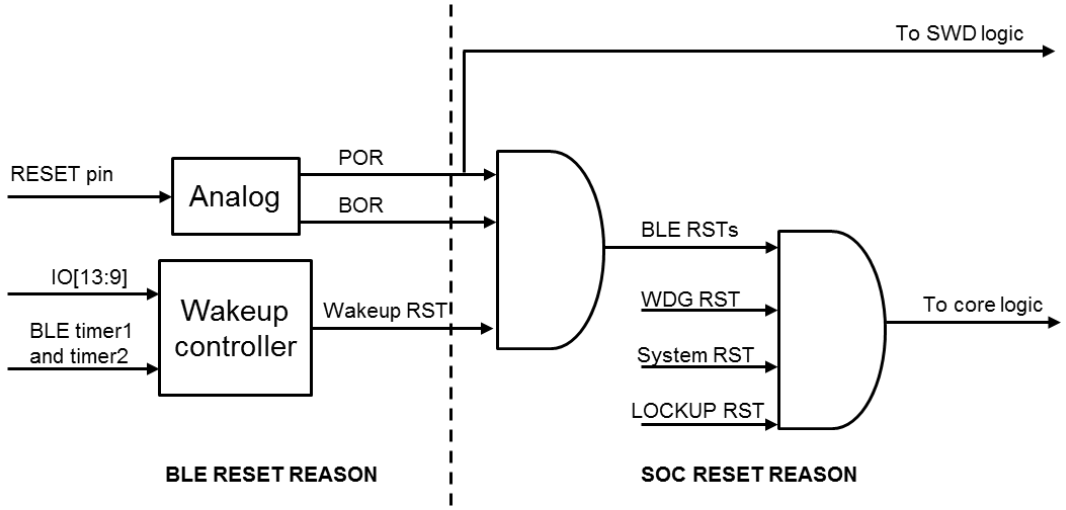
3.5.1 Reset management

Figure 7. Reset and wake-up generation shows the general principle of reset. Releasing the reset pin puts the chip out of shutdown state. The wake-up logic is powered and receives the POR. Each time the wake-up controller decides to exit sleep or standby modes, it will generate a reset for the core logic. The core logic can also be reset by:

- Watchdog
- Reset request from the processor (system reset)
- LOCKUP state of the Cortex-M0

The SWD logic is reset by the POR. It is important to highlight that reset pin actually power down chip, so it is not possible to perform debug access with system under reset.

Figure 7. Reset and wake-up generation



If, for any reason, the users would like to power off the device there are two options:

1. Force RESETN pin to ground, keeping VBAT level
2. To put VBAT pins to ground (e.g. via a transistor)

In the second option, care must be taken to ensure that no voltage is applied to any of the other pins since device can be powered and having an anomalous power consumption. ST recommendation is to use RESETN whenever it is possible.

3.5.1.1 *Power-on-Reset*

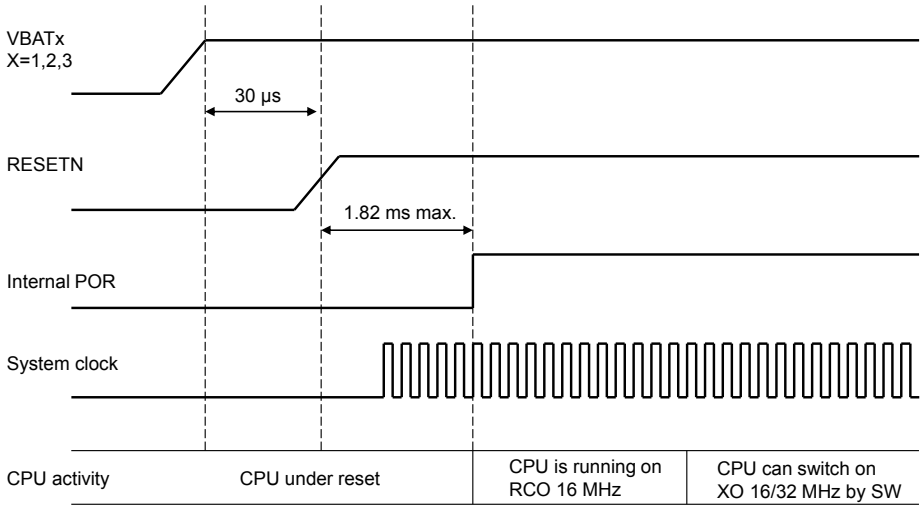
The Power-on-Reset (POR) signal is the combination of the POR signal and the BOR signal generated by the analog circuitry contained in the BlueNRG-2 device. The combination of these signals is used to generate the input to the Cortex-M0, which is used to reset the debug access port (DAP) of the processor. It is also used to generate the signal, which resets the debug logic of the Cortex-M0. The POR signal also resets the TAP controller of the BlueNRG-2 and a part of the Flash controller (managing the Flash memory boot, which does not need to be impacted by system resets).

The BOR reset is enabled by default. At software level, it can be decided to change the default values after reset.

3.5.1.2 *Power-up sequence*

The starting sequence of the BlueNRG-2 supply and reset signal is shown below.

Figure 8. BlueNRG-2 power-up sequence

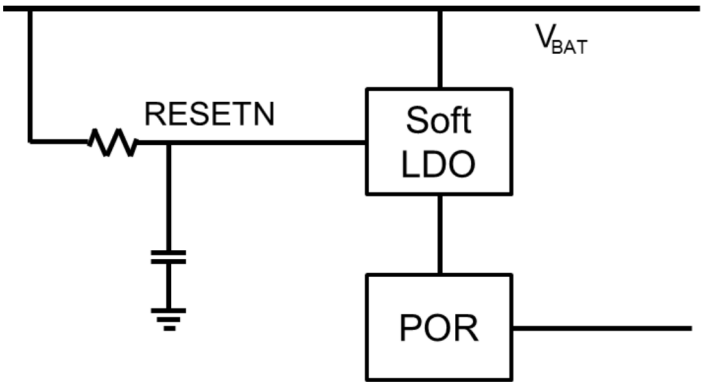


- The VBATx power must only be raised when RESETN pin is low.
- The different VBATx (x=1,2,3) power can be raised separately or together.
- Once the VBATx (x=1, 2, 3) reaches the nominal value, the RESETN pin could be driven high after a 30 us.
- The internal POR is released once internal LDOs are established and RCO clock is ready.
- The system starts on RCO 16 MHz clock system. The software is responsible for configuring the XO 16/32 MHz when necessary.

Note: The minimum negative pulse to reset the system must be at least 30 µs.

The POR circuit is powered by a 1.2 V regulator, which must also be powered up with the correct startup sequence. Before VBAT has reached the nominal value, RESETN line must be kept low. An external RC circuit on RESETN pin adds a delay that can prevent RESETN signal from going high before VBAT has reached the nominal value.

Figure 9. Reset circuit



If the above conditions are not satisfied, ST cannot guarantee the correct operation of the device.

3.5.1.3 *Watchdog reset*

The BlueNRG-2 contains a watchdog timer, which may be used to recover from software crashes. The watchdog contains a 32-bit down counter, which generates an interrupt, if the interrupt is not serviced, the watchdog generates a reset. The watchdog reset resets the Flash controller, the Cortex-M0 and all its peripherals but it does not reset the debug circuitry of the Cortex-M0.

3.5.1.4 *System reset request*

The system reset request is generated by the debug circuitry of the Cortex-M0. The debugger writes to the SYSRESETREQ bit of the “application interrupt and reset control register” (AIRCR). This system reset request through AIRCR register can also be done by embedded software. The system reset request does not affect the debugger, thus allowing the debugger to remain connected during the reset sequence.

3.5.1.5 *LOCKUP reset*

The Cortex-M0 generates an output LOCKUP that indicates that the core is in a deliberate lock-up state resulting from an unrecoverable exception. The LOCKUP signal is used to generate a reset in the BlueNRG-2, which affects the Cortex-M0, the Flash controller and all the peripherals.

The LOCKUP signal does not reset the Cortex-M0 debug circuitry and it is not generated if a debugger is connected.

3.5.2 Reset and wake-up reason decoding

The BlueNRG-2 provides a set of registers to identify the reason behind a reset and wake-up generation. Two registers are used: CKGEN_SOC->REASON_RST and CKGEN_BLE->REASON_RST. The possible reasons are listed below:

1. If the register CKGEN_SOC->REASON_RST = 0, according to the CKGEN_BLE->REASON_RST the possible reasons are:
   a. Wake-up from IO9, IO10, IO11, IO12, IO13
   b. Wake-up from internal timer: BLE timer 1 or BLE timer 2
   c. POR or BOR
2. If the register CKGEN_SOC->REASON_RST is not 0, according to its value the possible reasons are:
   a. System reset
   b. Watchdog reset
   c. Lockup reset

3.5.3 Clock and reset registers

CKGEN_SOC peripheral base address (CKGEN_SOC_BASE_ADDR) 0x40900000.

Table 13. CKGEN_SOC registers

Address offset	Name	RW	Reset	Description
0x08	REASON_RST	R	0x00000000	Indicates the reset reason from Cortex-M0. Refer to the detailed description below.
0x1C	DIE_ID(1)	R	0x00000100	Identification information of the device. Refer to the detailed description below.
0x20	CLOCK_EN	RW	0x0003FFFF	Enable or gates the APB clock of the peripherals. Refer to the detailed description below.
0x24	DMA_CONFIG	RW	0x00000000	DMA config. Refer to the detailed description below.
0x28	JTAG_IDCODE	R	0x0200A041	BlueNRG-2 JTAG IDCODE.

1. *It depends on the cut version.*

Table 14. CKGEN_SOC - REASON_RST register description: address offset CKGEN_SOC_BASE_ADDR+0x08

Bit	Field name	Reset	RW	Description
0	RESERVED	0x0	R	RESERVED.
1	SYSREQ	0x0	R	Reset caused by Cortex-M0 debug asserting SYSRESETREQ.
2	WDG	0x0	R	Reset caused by assertion of watchdog reset.
3	LOCKUP	0x0	R	Reset caused by Cortex-M0 asserting LOCKUP signal.
31:4	RESERVED	0x0	R	RESERVED.

Table 15. CKGEN_SOC - DIE_ID register description: address offset CKGEN_SOC_BASE_ADDR+0x1C

Bit	Field name	Reset	RW	Description
3:0	REV	0x0	R	Cut revision.
7:4	VERSION	0x0	R	Cut version.
11:8	PRODUCT	0x1	R	Product.
31:12	RESERVED	0x0	R	RESERVED.

Table 16. CKGEN_SOC - CLOCK_EN register description: address offset CKGEN_SOC_BASE_ADDR+0x20

Bit	Field name	Reset	RW	Description
0	GPIO	0x1	RW	GPIO clock
1	NVM	0x1	RW	Flash controller clock
2	SYSCTRL	0x1	RW	System controller clock
3	UART	0x1	RW	UART clock
4	SPI	0x1	RW	SPI clock
6:5	RESERVED	0x3	RW	RESERVED
7	WDOG	0x1	RW	Watchdog clock
8	ADC	0x1	RW	ADC clock
9	I2C1	0x1	RW	I2C1 clock
10	I2C2	0x1	RW	I2C2 clock
11	MFT1	0x1	RW	MFT1 clock
12	MFT2	0x1	RW	MFT2 clock
13	RTC	0x1	RW	RTC clock
14	SYSCLK_IO_EN	0x1	RW	System clock output on IO0 enable bit
15	XO_IO_EN	0x1	RW	XO clock output on IO9 enable bit
16	DMA	0x1	RW	DMA AHB clock
17	RNG	0x1	RW	RNG AHB clock
18	PKA	0x0	RW	PKA AHB clock
19	PKA RAM	0x0	RW	PKA RAM clock
31:20	RESERVED	0x0	RW	RESERVED

Table 17. CKGEN_SOC - DMA_CONFIG register description: address offset CKGEN_SOC_BASE_ADDR+0x24

Bit	Field name	Reset	RW	Description
0	ADC_CH0	0x0	RW	Select ADC on DMA channel 0 instead of peripheral.
1	ADC_CH1	0x0	RW	Select ADC on DMA channel 1 instead of peripheral.
2	ADC_CH2	0x0	RW	Select ADC on DMA channel 2 instead of peripheral.
3	ADC_CH3	0x0	RW	Select ADC on DMA channel 3 instead of peripheral
4	ADC_CH4	0x0	RW	Select ADC on DMA channel 4 instead of peripheral.
5	ADC_CH5	0x0	RW	Select ADC on DMA channel 5 instead of peripheral.
6	ADC_CH6	0x0	RW	Select ADC on DMA channel 6 instead of peripheral.
7	ADC_CH7	0x0	RW	Select ADC on DMA channel 7 instead of peripheral.
31:8	RESERVED	0x0	RW	RESERVED

Note: Only one DMA channel for the ADC should be selected at time. Hardware does not prevent selecting more than one DMA channel for ADC.

Table 18. CKGEN_SOC - JTAG_IDCODE register description: address offset CKGEN_SOC_BASE_ADDR+0x28

Bit	Field name	Reset	RW	Description
0	RESERVED	0x1	R	RESERVED

Bit	Field name	Reset	RW	Description
11:1	MANUF_ID	0x020	R	Manufacturer ID
27:12	PART_NUMBER	0x200A	R	Part number
31:28	VERSION_NUM	0x0	R	Version

CKGEN_BLE peripheral base address (CKGEN_BLE_BASE_ADDR) 0x48100000.

Table 19. CKGEN_BLE registers

Address offset	Name	RW	Reset	Description
0x08	REASON_RST	R	0x00000005	Indicates the Reset reason from BLE. Refer to the detailed description below.
0x0C	CLK32K_COUNT	RW	0x0000000F	Counter of 32 kHz clock. Refer to the detailed description below.
0x10	CLK32K_PERIOD	R	0x00000000	Period of 32 kHz clock. Refer to the detailed description below.
0x14	CLK32K_FREQ	R	0x00000000	Measurement of frequency of 32 kHz clock. Refer to the detailed description below.
0x18	CLK32K_IT	RW	0x00000000	Interrupt event for 32 kHz clock measurement. Refer to the detailed description below.

Table 20. CKGEN_BLE - REASON_RST register description: address offset CKGEN_BLE_BASE_ADDR+0x08

Bit	Field name	Reset	RW	Description
0	RESERVED	0x1	R	RESERVED
1	BOR	0x0	R	Reset from BOR
2	POR	0x1	R	Reset from POR
3	WKP_IO9	0x0	R	Wake-up from external IO9
4	WKP_IO10	0x0	R	Wake-up from external IO10
5	WKP_IO11	0x0	R	Wake-up from external IO11
6	WKP_IO12	0x0	R	Wake-up from external IO12
7	WKP_IO13	0x0	R	Wake-up from external IO13
8	WKP_BLUE	0x0	R	Wake-up comes from the timer 1 expiration in the wake-up control block of the BLE radio
10	WKP2_BLUE	0x0	R	Wake-up comes from the timer 2 expiration in the wake-up control block of the BLE radio
31:11	RESERVED	0x0	R	RESERVED

Table 21. CKGEN_BLE - CLK32K_COUNT register description: address offset CKGEN_BLE_BASE_ADDR+0x0C

Bit	Field name	Reset	RW	Description
8:0	SLOW_COUNT	0xF	RW	Program the window length (in slow clock period unit) for slow clock measurement
31:9	RESERVED	0x0	RW	RESERVED

Table 22. CKGEN_BLE - CLK32K_PERIOD register description: address offset CKGEN_BLE_BASE_ADDR+0x10

Bit	Field name	Reset	RW	Description
18:0	SLOW_PERIOD	0x0	R	Indicates slow clock period information. The result provided in this field corresponds to the length of SLOW_COUNT periods of the slow clock (32 kHz) measured in 16 MHz half-period unit. The measurement is done automatically each time the device enters in active2 mode using SLOW_COUNT = 16. A new calculation can be launched by writing zero in CLK32K_PERIOD register. In this case, the time window uses the value programmed in SLOW_COUNT field.
31:19	RESERVED	0x0	R	RESERVED

Table 23. CKGEN_BLE - CLK32K_FREQ register description: address offset CKGEN_BLE_BASE_ADDR+0x14

Bit	Field name	Reset	RW	Description
26:0	SLOW_FREQ	0x0	R	Value equal to 2^{39} / SLOW_PERIOD
31:27	RESERVED	0x0	R	RESERVED

Table 24. CKGEN_BLE - CLK32K_IT register description: address offset CKGEN_BLE_BASE_ADDR+0x18

Bit	Field name	Reset	RW	Description
0	CLK32K_MEAS_IRQ	0x0	RW	When read, provides the status of the interrupt indicating slow clock measurement is finished: 0: No pending interrupt. 1: Pending interrupt. When written, clears the interrupt: 0: No effect. 1: Clear the interrupt.
31:1	RESERVED	0x0	RW	RESERVED

Note: All RESERVED fields inside registers must always be written with their default value.

3.6 ADC

3.6.1 Introduction

The BlueNRG-2 integrates a 10-bit analog-to-digital converter (ADC) for sampling an external signal.

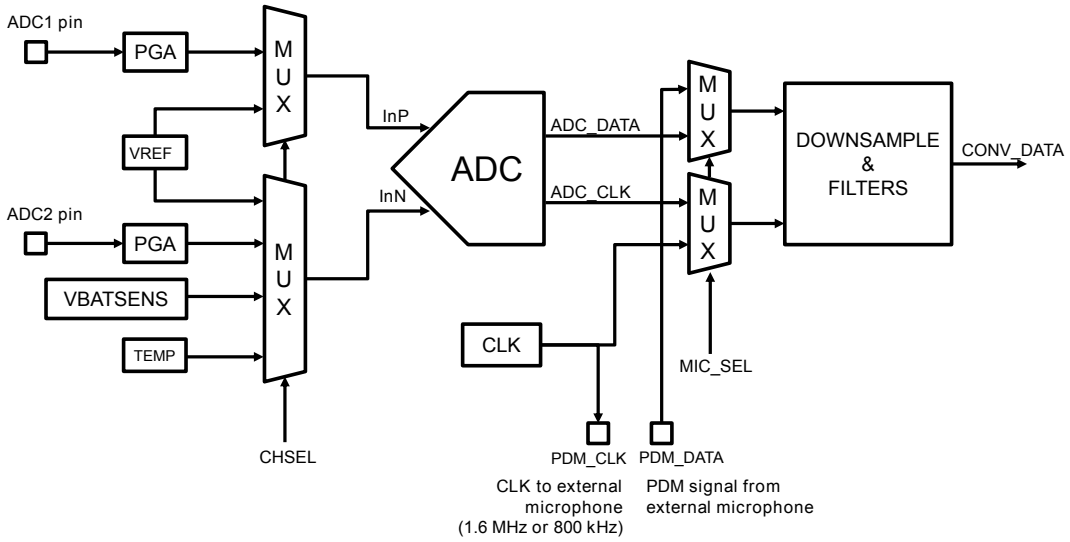
Main features are:

- Sampling frequency 1 MHz
- One channel in single ended or differential input through the pins ADC1 and ADC2
- Temperature and battery voltage sensors
- The conversion are either continuous or single step acquisition
- An integrated digital filter is used to process a PDM data stream from a MEMS microphone

3.6.2 Functional overview

The figure below shows a top diagram of the ADC.

Figure 10. **ADC block diagram**



Several channels are available for the conversion, the CHSEL selects the channel according to Table 25. ADC channels.

Table 25. **ADC channels**

CHSEL	Channels description
0	All switch open. No input
1	Single ended through ADC2 pin. InP = VREF (internal), InN = ADC2 pin
2	Single ended through ADC1 pin. InP = ADC1 pin, InN = VREF (internal)
3	Differential ADC1 pin – ADC2 pin. InP = ADC1 pin, InN = ADC2 pin
4	Temperature sensor. InN=TEMP, InP = 0.6 V (internal)
5	Battery voltage sensor. InN = VBATSENS, InP = 0.6 V (internal)
6	Short. InP = InN = 0.6 V (internal)

The conversion can be single (CONT = 0) or continuous (CONT = 1). In continuous mode, the conversion runs with a pre-programmed sampling rate, the user must discard the first samples that are not valid because generated during the establishment of the internal filter. In particular, it must discards a number of samples as follows:

- 10 samples if the bitfield SKIP is 0 (COMP filter not bypassed)
- 3 samples if the bitfield SKIP is 1 (COMP filter bypassed)

In single step mode the ADC performs a conversion and then stops.

The output data rate depends on the setting of OSR according to the following table.

Table 26. **ADC data rate**

OSR	Output data rate [Ksample/s]
0 (200)	5
1 (100)	10
2 (64)	15.625

OSR	Output data rate [Ksample/s]
3 (32)	31.25

The setting of the oversampling ratio (OSR) must be done according to the frequency of the input signal (AC), while for DC signals, the best performance is with OSR = 200. In order to achieve the best performance within the selected input voltage range, the attenuation value must be set through the corresponding register PGASEL value, as in the following table.

Table 27. ADC parameter settings

Vin range [V]	Attenuation [dB]	REFSEL value	PGASEL value
[0, 1.2]	0	2	0
[0, 2.4]	6.02	2	1
[0, 3.6]	9.54	2	2

The impedance of the ADC input pins is programmable through the register PGASEL as shown in the following table:

Table 28. Impedance of the ADC pin

PGASEL value	Attenuation [dB]	Input impedance[kΩ]
0	0	HiZ
1	6.02	520
2	9.54	585

Note: In order to reduce leakage from the ADC pins, the PGASEL register must be set to 0 if the ADC is not used, regardless the ADC is ON or OFF.

3.6.2.1 *ADC microphone mode*

The system can work in conjunction with an external MEMS microphone. In this mode the user must configure:

- an IO as PDM_CLK (GPIO Serial2 mode) in order to provide the clock signal to an external MEMS microphone (output signal)
- an IO as PDM_DATA (GPIO Serial2 mode) in order to receive and process the PDM data stream from the external MEMS microphone (input signal). See Table 130. IO functional map for more details about how these pins can be used for this mode.
- set the MIC_SEL bitfield of the CONF register, in order to provide a clock to the MEMS microphone. The PDM_CLK signal provides a clock that can be 1.6 MHz (DIG_FILT_CLK = 0) or 0.8 MHz (DIG_FILT_CLK =1)
- set the MIC_ON bitfield of the CTRL register, in order to make the ADC start the conversion from the MEMS microphone

Note: MIC_ON and ON bitfields must be exclusive and must not be set together.

The output data rate changes with the OSR and according to the clock frequency provided as explained in Table 29. Output data rate with microphone.

Table 29. **Output data rate with microphone**

DIG_FILT_CLK	OSR	Output data rate [Ksample/s]
1 (clock = 0.8 MHz)	0 (200)	4
	1 (100)	8
	2 (64)	12.5
	3 (32)	25
0 (clock = 1.6 MHz)	0 (200)	8
	1 (100)	16
	2 (64)	25
	3 (32)	50

3.6.2.2 *ADC start conversion*

The ADC both analog and digital sub-system are switched on by setting ADCON and SWSTART.

The conversion operation consists of four phases.

1. The wake-up phase lasts 6 µs, is present at the beginning of a single acquisition, with the goal to let the analog system to settle before to start the acquisition.
2. When CALEN bit and AUTO_OFFSET are set in ADC_CTRL register, a calibration starts. It permits to compensate the offset in the analog part. The conversion status is tracked by SR status register. At the beginning of the conversion the BUSY bit is set and masks any attempt to change CONF, up to the end of the conversion. At end of this conversion, the ENDCAL flag is generated and the OFFSET register is written with the converted offset voltage.
3. The acquisition phase is regulated by a timeout depending on the resolution. In this phase, digital filter chain processes the data coming from ADC.
4. The elaboration phase is at the end of the timeout, the data obtained at the output of the digital filter is written in the DATA register. The content of the OFFSET register is automatically used to compensate the final result. Furthermore, the ADCEOC flag is generated to warn about the end of conversion. If ENAB_COMP bit is set, the WDOG flag is generated to warn that the result of the conversion is between a high THRESHOLD_HI and low threshold THRESHOLD_LO.

Note: It is always advisable to set the register fields CALEN and AUTO_OFFSET in order to perform automatic calibration for each measurement.

3.6.2.3 *ADC offset*

The ADC automatically corrects a potential offset error by taking into account the content of the register OFFSET.

To enable the automatic offset correction the CALEN and the AUTO_OFFSET must be both set. The result of the calibration is stored in the OFFSET register.

The correction of the offset can be also done manually, for example by performing firstly an automatic offset calibration by making an ADC conversion with both AUTO_OFFSET and CALEN bitfields set. In this way, the OFFSET register is updated with the current offset error. Then, the automatic offset calibration can be disabled by set to 0b the AUTO_OFFSET and the CALEN bitfields. And so, the offset value is applied to all the next ADC conversions.

The calibration value is a 16-bit value in the register OFFSET. It must be placed in the bitfield OFFSET_MSB if the bitfield SKIP is 0 (filter not bypassed). While, if the bitfield SKIP is 1 (filter bypassed), the calibration value must be placed in the bitfield OFFSET_LSB.

3.6.2.4 *ADC conversion*

The relationship between differential input voltage and ADC_{RAW} code (first 16-bit MSB of DATA_CONV register) depends on a limited set of parameters: the digital core power supply VDD1V2, the PGA value, and a scaling factor.

Differential mode

This mode enables the ADC differential conversion from the pins ADC1 and ADC2.

$$V_{ADC12(Volt)} = V_{ADC1} - V_{ADC2} = (1 + PGASEL)^{*}\left(\frac{ADC_{RAW}}{FS\_16(OSR)}\right)2^{*}VDD1V2 \quad (1)$$

Single-ended mode

This mode enables the ADC conversion from the pin ADC1 or from the pin ADC2.

$$V_{ADC1(Volt)} = (1 + PGASEL)^{*}\left(VREF + \left(\frac{ADC_{RAW}}{FS\_16(OSR)}\right)^{*}2^{*}VDD1V2\right) \quad (2)$$

$$V_{ADC2(Volt)} = (1 + PGASEL)^{*}\left(VREF - \left(\frac{ADC_{RAW}}{FS\_16(OSR)}\right)^{*}2^{*}VDD1V2\right) \quad (3)$$

Battery voltage sensor

This mode enables the monitoring of the battery voltage V_{BATT}, through an internal resistive bridge.

$$V_{BATT(Volt)} = K_{BATT}^{*}\left(VREF - \left(\frac{ADC_{RAW}}{FS\_16(OSR)}\right)^{*}2^{*}VDD1V2\right) \quad (4)$$

Temperature sensor

This mode enables the monitoring of the temperature by means of an internal sensor, with the following voltage to temperature conversion:

$$V_{TEMP(°C)} = K_{TC}^{*}\left(VREF - \left(\frac{ADC_{RAW}}{FS\_16(OSR)}\right)^{*}2^{*}VDD1V2\right) + OFFSET_{TC} \quad (5)$$

To ensure an accurate temperature reading, average the value over several readings.

Below the values for the symbols used in the ADC conversion formulas:

- PGASEL is the input attenuation register, values: 0, 1, or 2
- FS_16(OSR) is the full scale factor for ADC_{RAW} and it depends on the oversampling ratio (OSR) as shown below:
- If SKIP is 0 (filter not bypassed), then:
  - ADC_{RAW} is DATA_CONV_MSB
  - FS_16(32) = FS_16(64) = 35442
  - FS_16(100) = FS_16(200) = 41260
- If SKIP is 1 (filter bypassed), then:
  - ADC_{RAW} is DATA_CONV_LSB
  - FS_16(32) = FS_16(64) = 32768
  - FS_16(100) = FS_16(200) = 38147
- VDD1V2 is the digital core power supply = 1.2 V
- VREF is given by the register REFSEL, with a typical value of 0.6 V
- K_{BATT} is 4.36
- K_{TC} is 401
- $OFFSET_{TC}$ is 267 °C

3.6.3 ADC registers

ADC peripheral base address (ADC_BASE_ADDR) 0x40800000.

Table 30. ADC registers

Address offset	Name	RW	Reset	Description
0x00	CTRL	RW	0x00000000	ADC control register. Refer to the detailed description below.
0x04	CONF	RW	0x0000000C	ADC configuration register. Refer to the detailed description below.
0x08	IRQSTAT	R	0x00000000	IRQ masked status register. Refer to the detailed description below.
0x0C	IRQMASK	RW	0x0000000F	It sets the mask for ADC interrupt. Refer to the detailed description below.
0x10	IRQRAW	R	0x00000000	IRQ status register. Refer to the detailed description below.

Address offset	Name	RW	Reset	Description
0x14	DATA_CONV	R	0x00000000	Result of the conversion in two complement format.
0x18	OFFSET	RW	0x00000000	Offset for correction of converted data
0x20	SR_REG	RW	0x00000000	ADC status register. Refer to the detailed description below.
0x24	THRESHOLD_HI	RW	0xFFFFFFFF	High threshold for window comparator.
0x28	THRESHOLD_LO	RW	0x00000000	Low threshold for window comparator.

Table 31. ADC - CTRL register description: address offset ADC_BASE_ADDR+0x00

Bit	Field name	Reset	RW	Description
0	ON	0x0	RW	Starts ADC analog subsystem. This bit must be set before starting a conversion. 0: ADC is OFF. 1: ADC is ON. This bit works for all the mode except the microphone mode.
1	CALEN	0x0	RW	The automatic calibration routine is enabled if both AUTO_OFFSET and CALEN bitfields are set. The result of the calibration is placed in the OFFSET register according to the SKIP bitfield value. 0: disable the automatic calibration 1: enable the automatic calibration This bitfield can be set to 0 only by setting to 1 the bitfield RSTCALEN
2	SWSTART	0x0	RW	Starts the ADC conversion phase when set. This bit works for all the mode except the microphone mode.
3	RESET	0x0	RW	Reset all the ADC APB registers when set (CTRL, CONF, DATA_CONV, THRESHOLD_HI, THRESHOLD_LO). This bit is auto-cleared by the hardware so it is always read 0
4	STOP	0x0	RW	Permits the continuous conversion to be stopped. 1: stop the continuous conversion and switch off the ADC. The bitfields SWSTART, ON, DMA_EN and MIC_ON are auto-cleared if set. This bit is auto-cleared by the hardware so it is always read at 0.
5	ENAB_COMP	0x0	RW	Enables the window comparator when set to 1. WDOG flag is ADC_SR register is set if the converted value is between THRESHOLD_HI and THRESHOLD_LO value.

Bit	Field name	Reset	RW	Description
6	RSTCALEN	0x0	RW	Disable the calibration phase when set to 1. This bit has to be set to disable the calibration each time calibration is enabled. This bit is auto-cleared by the hardware so it is always read at 0.
7	AUTO_OFFSET	0x0	RW	The automatic calibration routine is enabled if both AUTO_OFFSET and CALEN bitfields are set. The result of the calibration is placed in the OFFSET register according to the SKIP bitfield value. 0: disable the automatic calibration. 1: enable the automatic calibration.
8	MIC_ON	0x0	RW	Starts ADC analog subsystem for microphone mode only. 0: ADC is OFF 1: ADC is ON
9	DMA_EN	0x0	RW	Enables the DMA. 0: DMA is disabled. 1: DMA is enabled.
31:10	RESERVED	0x0	RW	RESERVED

Table 32. ADC - CONF register description: address offset ADC_BASE_ADDR+0x04

Bit	Field name	Reset	RW	Description
0	RESERVED	0x0	RW	RESERVED
3:1	CHSEL	0x6	RW	Select the input channel: 000b: All switches open. 001b: Single-ended through ADC2 pin. InP = VREF (internal), InN = ADC2 pin. 010b: Single-ended through ADC1 pin. InP = ADC1 pin, InN = VREF (internal). 011b: Differential ADC1 pin - ADC2 pin, InP = ADC1 pin, InN = ADC2 pin. 100b: Temperature sensor. InP = 0.6 V (internal), InN = TEMP. 101b: Battery voltage sensor. InP = 0.6 V (internal), InN = VBATSENS. 110b: Short InN = InP = 0.6 V (internal).
5:4	REFSEL	0x0	RW	Set the VREF for single ended conversion: 00b: 0.0 V (default value not recommended) 10b: 0.6 V (suggested value)
7:6	OSR	0x0	RW	Set the oversampling ratio. 00b: Set the oversampling ratio to 200 (1) 01b: Set the oversampling ratio to 100 10b: Set the oversampling ratio to 64 11b: Set the oversampling ratio to 32
9:8	PGASEL	0x0	RW	Set the input attenuator value:

Bit	Field name	Reset	RW	Description
				00b: Input attenuator at 0 dB 01b: Input attenuator at 6.02 dB 10b: Input attenuator at 9.54 dB
10	RESERVED	0x0	RW	RESERVED
11	CONT	0x0	RW	Enable the continuous conversion mode: 0: Single conversion 1: Continuous conversion
17:12	RESERVED	0x00	RW	RESERVED
18	SKIP	0x0	RW	It permits the filter COMP to be bypassed to speed up the conversion for signal at low frequency: 0: Filter not bypassed 1: Filter bypassed According to the value of this bitfield, the behavior of the ADC changes as follows: - If SKIP is 0: the first 10 converted samples in ADC mode continuous should be discarded by the user. The converted date is in the bitfield DATA_CONV_MSB. The calibration result is in the bitfield OFFSET_MSB. - If SKIP is 1: the first 3 converted samples in ADC mode continuous should be discarded by the user. The converted date is in the bitfield DATA_CONV_LSB. The calibration result is in the bitfield OFFSET_LSB. So, a calibration must be redone if the SKIP bitfield value changes.
19	RESERVED	0x0	RW	RESERVED
20	DIG_FILT_CLK	0x0	RW	Frequency clock selection value on PDM_CLK when MIC_SEL=1: 0: 1.6 MHz 1: 0.8 MHz
21	RESERVED	0x0	RW	RESERVED
22	MIC_SEL	0x0	RW	Provides the clock on GPIO: 0: Do not provided any external clock source 1: Provide clock source from GPIO
31:23	RESERVED	0x000	RW	RESERVED

1. *Best value for sampling DC signals.*

Table 33. ADC - IRQSTAT register description: address offset ADC_BASE_ADDR+0x08

Bit	Field name	Reset	RW	Description
0	ENDCAL	0x0	R	1: When the calibration is completed. Clear on register read.
1	RESERVED	0x0	R	RESERVED
2	EOC	0x0	R	1: When the conversion is completed. Clear on register read.
3	WDOG	0x0	R	If ENAB_COMP = 1, this bit indicates the result of the conversion is between high and low threshold: 0: DATA_CONV is NOT between THRESHOLD_HI and THRESHOLD_LO values.

Bit	Field name	Reset	RW	Description
				1: DATA_CONV is between THRESHOLD_HI and THRESHOLD_LO values. This field is updated on each new end of conversion event related to the converted data value. Clear on register read.
31:4	RESERVED	0x0	R	RESERVED

Table 34. ADC - IRQMASK register description: address offset ADC_BASE_ADDR+0x0C

Bit	Field name	Reset	RW	Description
0	ENDCAL	0x1	RW	Interrupt mask for the end of calibration event: 0: Interrupt is enabled. 1: Interrupt is disabled.
1	RESERVED	0x1	RW	RESERVED
2	EOC	0x1	RW	Interrupt mask for the end of conversion event: 0: Interrupt is enabled. 1: Interrupt is disabled.
3	WDOG	0x1	RW	Interrupt mask for the within the threshold event: 0: Interrupt is enabled. 1: Interrupt is disabled.
31:4	RESERVED	0x0	RW	RESERVED

Table 35. ADC - IRQRAW register description: address offset ADC_BASE_ADDR+0x10

Bit	Field name	Reset	RW	Description
0	ENDCAL	0x0	R	1: When the calibration is completed. Clear on register read.
1	RESERVED	0x0	R	RESERVED
2	EOC	0x0	R	1: When the conversion is completed. Clear on register read.
3	WDOG	0x0	R	If ENAB_COMP = 1, this bit indicates the result of the conversion is between high and low threshold: 0: DATA_CONV is NOT between THRESHOLD_HI and THRESHOLD_LO values. 1: DATA_CONV is between THRESHOLD_HI and THRESHOLD_LO values. This field is updated on each new end of conversion event related to the converted data value. Clear on register read.
31:4	RESERVED	0x0	R	RESERVED

Table 36. ADC - DATA_CONV register description: address offset ADC_BASE_ADDR+0x14

Bit	Field name	Reset	RW	Description
31:16	DATA_CONV_MSB	0x0000	R	Result of the conversion in two complement format. If the filter is not bypassed, the bitfield SKIP is 0, the DATA_CONV_LSB is negligible and the ADC converted data is the DATA_CONV_MSB.
15:0	DATA_CONV_LSB	0x0000	R	Result of the conversion in two complement format. If the filter is bypassed, the bitfield SKIP is 1, the DATA_CONV_MSB is negligible and the ADC converted data is the value of DATA_CONV_LSB * 1.08 (corrective factor).

Table 37. ADC - OFFSET register description: address offset ADC_BASE_ADDR+0x18

Bit	Field name	Reset	RW	Description
31:16	OFFSET_MSB	0x0000	RW	Offset for correction of converted data. If the bitfield SKIP is 0, the 16-bit offset is in the MSB part of the register.
15:0	OFFSET_LSB	0x0000	RW	Offset for correction of converted data. if the bitfield SKIP is 1, the 16-bit offset is in the LSB part of the register.

Table 38. ADC - SR_REG register description: address offset ADC_BASE_ADDR+0x20

Bit	Field name	Reset	RW	Description
0	RESERVED	0x0	RW	RESERVED
1	BUSY	0x0	RW	1: during conversion.
31:2	RESERVED	0x0	RW	RESERVED

Table 39. ADC - THRESHOLD_HI register description: address offset ADC_BASE_ADDR+0x24

Bit	Field name	Reset	RW	Description
31:0	THRESHOLD_HI	0xFFFFFFFF	RW	High threshold for window comparator.

Table 40. ADC - THRESHOLD_LO register description: address offset ADC_BASE_ADDR+0x28

Bit	Field name	Reset	RW	Description
31:0	THRESHOLD_LO	0x00000000	RW	Low threshold for window comparator.

Note: All RESERVED fields inside registers must always be written with their default value.

3.7 DMA

3.7.1 Introduction

The BlueNRG-2 device embeds a DMA allowing various combination of data transfer between the memory and the peripherals without CPU intervention.

Main features are:

- Eight independently configurable channels connected to dedicated hardware DMA requests; software trigger is also supported.
- Priorities between requests from channels of the DMA are software programmable (four levels consisting of very high, high, medium, low). When two channels with same software priority need attention, channel with lower hardware index will take priority.
- Independent source and destination transfer size (byte, half word, word), emulating packing and unpacking.
- Support for circular buffer management.
- Event flags (DMA half transfer, DMA transfer complete), logically ORed together in a single interrupt request for each channel.
- Memory-to-memory transfer (RAM only), peripheral-to-memory and memory-to-peripheral, and peripheral-to-peripheral transfers.
- Programmable number of data to be transferred up to 65536 bytes.

3.7.2 Functional overview

The DMA controller performs direct memory transfer by sharing the system bus with the other masters of the device. The DMA request may stop the CPU access to the system bus for some bus cycles, when the CPU and DMA are targeting the same destination (memory or peripheral). The bus matrix implements round-robin scheduling, thus ensuring at least half of the system bus bandwidth (both to memory and peripheral) for the CPU.

3.7.2.1 DMA transactions

After an event, the peripheral sends a request signal to the DMA controller. The DMA controller serves the request depending on the channel priorities. As soon as the DMA controller accesses the peripheral, the DMA controller sends an acknowledge to the peripheral. The peripheral releases its request as soon as it gets the acknowledge from the DMA controller. Once the request is deasserted by the peripheral, the DMA controller releases the acknowledge. If there are more requests, the peripheral can initiate the next transaction.

In summary, each DMA transfer consists of three operations:

- The loading of data from the peripheral data register or a location in memory addressed through an internal current peripheral/memory address register. The start address used for the first transfer is the base peripheral/memory address programmed in the DMA_CPARx or DMA_CMARx register.
- The storage of the data loaded to the peripheral data register or a location in memory addressed through an internal current peripheral/memory address register. The start address used for the first transfer is the base peripheral/memory address programmed in the DMA_CPARx or DMA_CMARx register.
- The post-decrementing of the DMA_CNDTRx register, which contains the number of transactions that have still to be performed.

3.7.2.2 Arbiter

The arbiter manages the channel requests based on their priority and launches the peripheral/memory access sequences.

The priorities are managed in two stages:

- Software: each channel priority can be configured in the DMA_CCRx register. There are four levels:
  - Very high priority
  - High priority
  - Medium priority
  - Low priority
- Hardware: if two requests have the same software priority level, the channel with the lowest number has the priority versus the channel with the highest number. For example, channel 2 gets priority over channel 4.

3.7.2.3 DMA channels

Each channel can handle DMA transfer between a peripheral register located at a fixed address and a memory address. The amount of data to be transferred (up to 65535) is programmable. The register, which contains the amount of data items to be transferred, is decremented after each transaction.

Programmable data sizes

Transfer data sizes of the peripheral and memory are fully programmable through the PSIZE and MSIZE bits in the DMA_CCRx register.

Pointer increments

Peripheral and memory pointers can optionally be automatically post-incremented after each transaction depending on the PINC and MINC bits in the DMA_CCRx register. If incremented mode is enabled, the address of the next transfer will be the address of the previous one incremented by 1, 2 or 4 depending on the chosen data size. The first transfer address is the one programmed in the DMA_CPARx/DMA_CMARx registers. During transfer operations, these registers keep the initially programmed value. The current transfer addresses (in the current internal peripheral/memory address register) are not accessible by software. If the channel is configured in non-circular mode, no DMA request is served after the last transfer (that is once the number of data items to be transferred has reached zero). In order to reload a new number of data items to be transferred into the DMA_CNDTRx register, the DMA channel must be disabled.

If a DMA channel is disabled, the DMA registers are not Reset. The DMA channel registers (DMA_CCRx, DMA_CPARx and DMA_CMARx) retain the initial values programmed during the channel configuration phase.

In circular mode, after the last transfer, the DMA_CNDTRx register is automatically reloaded with the initially programmed value. The current internal address registers are reloaded with the base address values from the DMA_CPARx/DMA_CMARx registers.

Channel configuration procedure

The following sequence should be followed to configure a DMA channelx (where x is the channel number).

1. Set the peripheral register address in the DMA_CPARx register. The data are moved from/ to this address to/ from the memory after the peripheral event.
2. Set the memory address in the DMA_CMARx register. The data will be written to or read from this memory after the peripheral event.
3. Configure the total number of data to be transferred in the DMA_CNDTRx register. After each peripheral event, this value is decremented.
4. Configure the channel priority using the PL[1:0] bits in the DMA_CCRx register.
5. Configure data transfer direction, circular mode, peripheral and memory incremented mode, peripheral and memory data size, and interrupt after half and/or full transfer in the DMA_CCRx register.
6. Activate the channel by setting the ENABLE bit in the DMA_CCRx register.

As soon as the channel is enabled, it can serve any DMA request from the peripheral connected on the channel. Once half of the bytes are transferred, the half-transfer flag (HTIF) is set and an interrupt is generated if the half-transfer interrupt enable bit (HTIE) is set. At the end of the transfer, the transfer complete flag (TCIF) is set and an interrupt is generated if the transfer complete interrupt enable bit (TCIE) is set.

Circular mode

Circular mode is available to handle circular buffers and continuous data flows (e.g. ADC scan mode). This feature can be enabled using the CIRC bit in the DMA_CCRx register. When circular mode is activated, the number of data to be transferred is automatically reloaded with the initial value programmed during the channel configuration phase, and the DMA requests continue to be served.

Memory-to-memory mode

The DMA channels can also work without being triggered by a request from a peripheral. This mode is called memory-to-memory mode. If the MEM2MEM bit in the DMA_CCRx register is set, then the channel initiates transfers as soon as it is enabled by software by setting the Enable bit (EN) in the DMA_CCRx register. The transfer stops once the DMA_CNDTRx register reaches zero. Memory-to-Memory mode may not be used at the same time as Circular mode.

3.7.2.4 Programmable data width, data alignment and endianness

When PSIZE and MSIZE are not equal, the DMA performs some data alignments as described in Table 41. Programmable data width & endian behavior (when bits PINC = MINC = 1): Programmable data width & endian behavior (when bits PINC = MINC = 1).

Table 41. Programmable data width and endian behavior (when bits PINC = MINC = 1)

Source port width	Destination port width	Number of data items to transfer (NDT)	Source content: address/data	Transfer operations	Destination content: address/data
8	8	4	@0x0 / B0	1: READ B0[7:0] @0x0 then WRITE B0[7:0] @0x0	@0x0 / B0
			@0x1 / B1	2: READ B1[7:0] @0x1 then WRITE B0[7:0] @0x1	@0x1 / B1
			@0x2 / B2	3: READ B2[7:0] @0x2 then WRITE B0[7:0] @0x2	@0x2 / B2
			@0x3 / B3	4: READ B3[7:0] @0x3 then WRITE B0[7:0] @0x3	@0x3 / B3
8	16	4	@0x0 / B0	1: READ B0[7:0] @0x0 then WRITE 00B0[15:0] @0x0	@0x0 / 00B0
			@0x1 / B1	2: READ B1[7:0] @0x1 then WRITE 00B0[15:0] @0x2	@0x2 / 00B1
			@0x2 / B2	3: READ B2[7:0] @0x2 then WRITE 00B0[15:0] @0x4	@0x4 / 00B2
			@0x3 / B3	4: READ B3[7:0] @0x3 then WRITE 00B0[15:0] @0x6	@0x6 / 00B3
8	32	4	@0x0 / B0	1: READ B0[7:0] @0x0 then WRITE 000000B0[31:0] @0x0	@0x0 / 000000B0
			@0x1 / B1	2: READ B1[7:0] @0x1 then WRITE 000000B0[31:0] @0x4	@0x4 / 000000B1
			@0x2 / B2	3: READ B2[7:0] @0x2 then WRITE 000000B0[31:0] @0x8	@0x8 / 000000B2
			@0x3 / B3	4: READ B3[7:0] @0x3 then WRITE 000000B0[31:0] @0xC	@0xC / 000000B3
16	8	4	@0x0 / B1B0	1: READ B1B0[15:0] @0x0 then WRITE B0[7:0] @0x0	@0x0 / B0
			@0x1 / B3B2	2: READ B3B2[15:0] @0x2 then WRITE B0[7:0] @0x1	@0x1 / B2
			@0x2 / B5B4	3: READ B5B4[15:0] @0x4 then WRITE B0[7:0] @0x2	@0x2 / B4
			@0x3 / B7B6	4: READ B7B6[15:0] @0x6 then WRITE B0[7:0] @0x3	@0x3 / B6
16	16	4	@0x0 / B1B0	1: READ B1B0[15:0] @0x0 then WRITE B1B0[15:0] @0x0	@0x0 / B1B0
			@0x1 / B3B2	2: READ B3B2[15:0] @0x2 then WRITE B3B2[15:0] @0x2	@0x2 / B3B2
			@0x2 / B5B4	3: READ B5B4[15:0] @0x4 then WRITE B5B4[15:0] @0x4	@0x4 / B5B4
			@0x3 / B7B6	4: READ B7B6[15:0] @0x6 then WRITE B7B6[15:0] @0x6	@0x6 / B7B6
16	32	4	@0x0 / B1B0	1: READ B1B0[15:0] @0x0 then WRITE 0000B1B0[31:0] @0x0	@0x0 / 0000B1B0
			@0x1 / B3B2	2: READ B3B2[15:0] @0x2 then WRITE 0000B3B2[31:0] @0x4	@0x4 / 0000B3B2
			@0x2 / B5B4	3: READ B5B4[15:0] @0x4 then WRITE 0000B5B4[31:0] @0x8	@0x8 / 0000B5B4

Source port width	Destination port width	Number of data items to transfer (NDT)	Source content: address/data	Transfer operations	Destination content: address/data
16	32	4	@0x3 / B7B6	4: READ B7B6[15:0] @0x6 then WRITE 0000B7B6[31:0] @0xC	@0xC / 0000B7B6
32	8	4	@0x0 / B3B2B1B0	1: READ B3B2B1B0[31:0] @0x0 then WRITE B0[7:0] @0x0	@0x0 / B0
			@0x4 / B7B6B5B4	2: READ B7B6B5B4[31:0] @0x4 then WRITE B0[7:0] @0x1	@0x1 / B4
			@0x8 / BBBAB9B8	3: READ BBBAB9B8[31:0] @0x8 then WRITE B0[7:0] @0x2	@0x2 / B8
			@0xC / BFBEBDBC	4: READ BFBEBDBC[31:0] @0xC then WRITE B0[7:0] @0x3	@0x3 / BC
32	16	4	@0x0 / B3B2B1B0	1: READ B3B2B1B0[31:0] @0x0 then WRITE B1B0[15:0] @0x0	@0x0 / B1B0
			@0x4 / B7B6B5B4	2: READ B7B6B5B4[31:0] @0x4 then WRITE B3B2[15:0] @0x2	@0x2 / B5B4
			@0x8 / BBBAB9B8	3: READ BBBAB9B8[31:0] @0x8 then WRITE B5B4[15:0] @0x4	@0x4 / B9B8
			@0xC / BFBEBDBC	4: READ BFBEBDBC[31:0] @0xC then WRITE B7B6[15:0] @0x6	@0x6 / BDBC
32	32	4	@0x0 / B3B2B1B0	1: READ B3B2B1B0[31:0] @0x0 then WRITE 0000B1B0[31:0] @0x0	@0x0 / B3B2B1B0
			@0x4 / B7B6B5B4	2: READ B7B6B5B4[31:0] @0x4 then WRITE 0000B3B2[31:0] @0x4	@0x4 / B7B6B5B4
			@0x8 / BBBAB9B8	3: READ BBBAB9B8[31:0] @0x8 then WRITE 0000B5B4[31:0] @0x8	@0x8 / BBBAB9B8
			@0xC / BFBEBDBC	4: READ BFBEBDBC[31:0] @0xC then WRITE 0000B7B6[31:0] @0xC	@0xC / BFBEBDBC

The DMA is addressed through AHB and can be accessed only with 32-bit access. Any 8-bit or 16-bit access will generate a hard fault.

When the DMA initiates an AHB byte or halfword write operation, the data are duplicated on the unused lanes of the HWDATA[31:0] bus. So when the used AHB slave peripheral does not support byte or halfword write operations (when HSIZE is not used by the peripheral) and does not generate any error, the DMA writes the 32 HWDATA bits as shown in the two examples below:

- To write the halfword “0xABCD”, the DMA sets the HWDATA bus to “0xABCDABCD” with HSIZE = HalfWord
- To write the byte “0xAB”, the DMA sets the HWDATA bus to “0xABABABAB” with HSIZE = byte

Assuming that the AHB/APB bridge is an AHB 32-bit slave peripheral that does not take the HSIZE data into account, it will transform any AHB byte or halfword operation into a 32-bit APB operation in the following manner:

- an AHB byte write operation of the data “0xB0” to 0x0 (or to 0x1, 0x2 or 0x3) is converted to an APB word write operation of the data “0xB0B0B0B0” to 0x0
- an AHB half-word write operation of the data “0xB1B0” to 0x0 (or to 0x2) is converted to an APB word write operation of the data “0xB1B0B1B0” to 0x0

3.7.2.5 *Error management*

A DMA transfer error can be generated by reading from or writing to a reserved address space. When a DMA transfer error occurs during a DMA read or a write access, the faulty channel is automatically disabled through a hardware clear of its EN bit in the corresponding channel configuration register (DMA_CCRx). The channel's transfer error interrupt flag (TEIF) in the ISR register is set and an interrupt is generated if the transfer error interrupt enable bit (TEIE) in the DMA_CCRx register is set.

3.7.2.6 *Interrupts*

An interrupt can be produced on a half-transfer, transfer complete or transfer error for each DMA channel. Separate interrupt enable bits are available for flexibility.

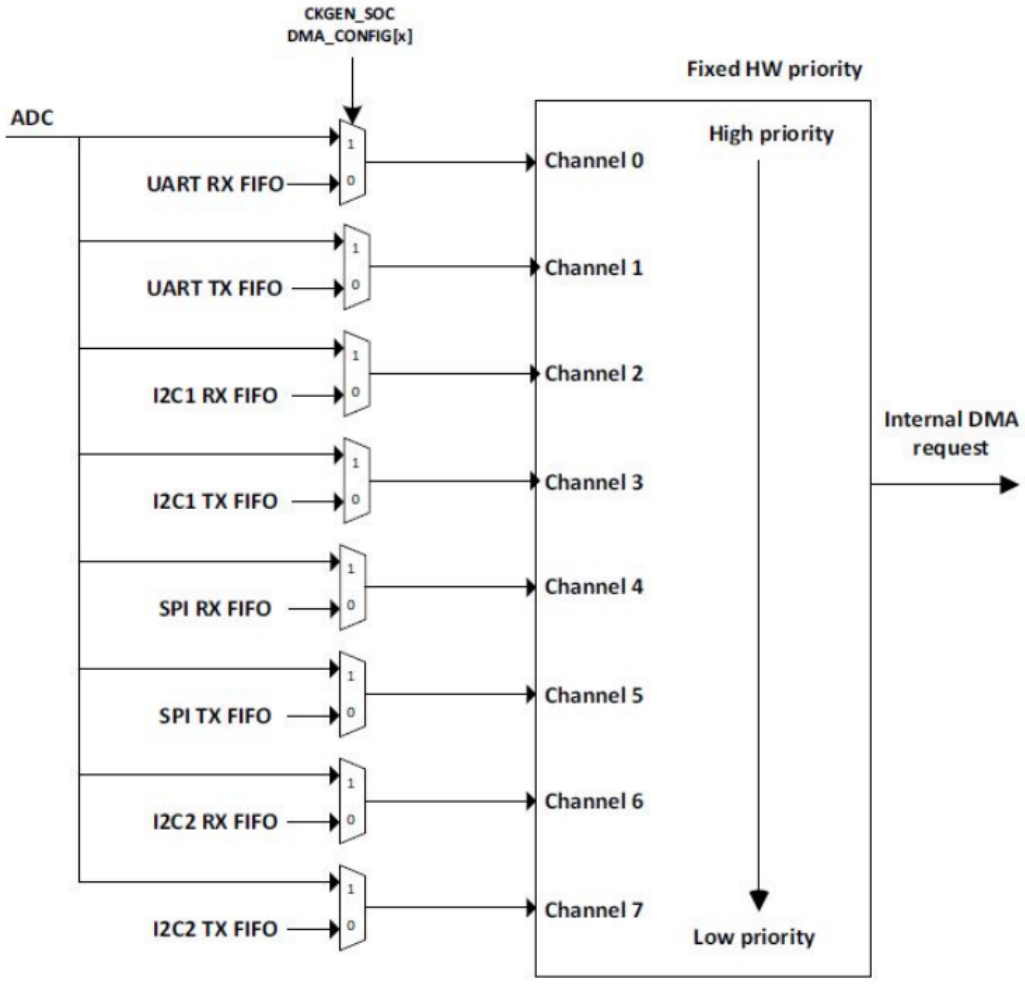
Table 42. DMA interrupt requests

Interrupt event	Event flag	Enable control bit
Half-transfer	HTIF	HTIE
Transfer complete	TCIF	TCIE
Transfer error	TEIF	TEIE

3.7.2.7 *DMA request mapping*

The eight requests from the peripherals (SPI, I2Cx[1,2] and UART) are multiplexed before entering DMA with the ADC request. For each channel, the choice between the peripheral and the ADC is done through the DMA_CONFIG register.

Figure 11. DMA request mapping in BlueNRG-2



3.7.3 DMA registers

DMA peripheral base address (DMA_BASE_ADDR) 0xA0000000.

Table 43. **DMA registers**

Address offset	Name	RW	Reset	Description
0x00	ISR	R	0x00000000	DMA interrupt status register. Refer to the detailed description below.
0x04	IFCR	W	0x00000000	DMA interrupt flag clear register. Refer to the detailed description below.

Table 44. **DMA - ISR register description: address offset DMA_BASE_ADDR+0x00**

Bit	Field name	Reset	RW	Description
0	GIF0	0x0	R	Channel 0 global interrupt flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No TE, HT or TC event on channel 0. 1: A TE, HT or TC event occurred on channel 0.
1	TCIF0	0x0	R	Channel 0 transfer complete flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer complete (TC) on channel 0. 1: A transfer complete (TC) occurred on channel 0.
2	HTIF0	0x0	R	Channel 0 half transfer flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No half transfer (HT) event on channel 0. 1: A half transfer (HT) event occurred on channel 0.
3	TEIF0	0x0	R	Channel 0 transfer error flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer error (TE) event on channel 0. 1: A transfer error (TE) occurred on channel 0.
4	GIF1	0x0	R	Channel 1 global interrupt flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No TE, HT or TC event on channel 1. 1: A TE, HT or TC event occurred on channel 1.
5	TCIF1	0x0	R	Channel 1 transfer complete flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer complete (TC) on channel 1. 1: A transfer complete (TC) occurred on channel 1.
6	HTIF1	0x0	R	Channel 1 half transfer flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No half transfer (HT) event on channel 1. 1: A half transfer (HT) event occurred on channel 1.
7	TEIF1	0x0	R	Channel 1 transfer error flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer error (TE) event on channel 1. 1: A transfer error (TE) occurred on channel 1.

Bit	Field name	Reset	RW	Description
8	GIF2	0x0	R	Channel 2 global interrupt flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No TE, HT or TC event on channel 2. 1: A TE, HT or TC event occurred on channel 2.
9	TCIF2	0x0	R	Channel 2 transfer complete flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer complete (TC) on channel 2. 1: A transfer complete (TC) occurred on channel 2.
10	HTIF2	0x0	R	Channel 2 half transfer flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No half transfer (HT) event on channel 2. 1: A half transfer (HT) event occurred on channel 2.
11	TEIF2	0x0	R	Channel 2 transfer error flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer error (TE) event on channel 2. 1: A transfer error (TE) occurred on channel 2.
12	GIF3	0x0	R	Channel 3 global interrupt flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No TE, HT or TC event on channel 3. 1: A TE, HT or TC event occurred on channel 3.
13	TCIF3	0x0	R	Channel 3 transfer complete flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer complete (TC) on channel 3. 1: A transfer complete (TC) occurred on channel 3.
14	HTIF3	0x0	R	Channel 3 half transfer flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No half transfer (HT) event on channel 3. 1: A half transfer (HT) event occurred on channel 3.
15	TEIF3	0x0	R	Channel 3 transfer error flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer error (TE) event on channel 3. 1: A transfer error (TE) occurred on channel 3.
16	GIF4	0x0	R	Channel 4 global interrupt flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No TE, HT or TC event on channel 4. 1: A TE, HT or TC event occurred on channel 4.
17	TCIF4	0x0	R	Channel 4 transfer complete flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer complete (TC) on channel 4. 1: A transfer complete (TC) occurred on channel 4.

Bit	Field name	Reset	RW	Description
18	HTIF4	0x0	R	Channel 4 half transfer flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No half transfer (HT) event on channel 4. 1: A half transfer (HT) event occurred on channel 4.
19	TEIF4	0x0	R	Channel 4 transfer error flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer error (TE) event on channel 4. 1: A transfer error (TE) occurred on channel 4.
20	GIF5	0x0	R	Channel 5 global interrupt flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No TE, HT or TC event on channel 5. 1: A TE, HT or TC event occurred on channel 5.
21	TCIF5	0x0	R	Channel 5 transfer complete flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer complete (TC) on channel 5. 1: A transfer complete (TC) occurred on channel 5.
22	HTIF5	0x0	R	Channel 5 half transfer flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No half transfer (HT) event on channel 5. 1: A half transfer (HT) event occurred on channel 5.
23	TEIF5	0x0	R	Channel 5 transfer error flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer error (TE) event on channel 5. 1: A transfer error (TE) occurred on channel 5.
24	GIF6	0x0	R	Channel 6 global interrupt flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No TE, HT or TC event on channel 6. 1: A TE, HT or TC event occurred on channel 6.
25	TCIF6	0x0	R	Channel 6 transfer complete flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer complete (TC) on channel 6. 1: A transfer complete (TC) occurred on channel 6.
26	HTIF6	0x0	R	Channel 6 half transfer flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No half transfer (HT) event on channel 6. 1: A half transfer (HT) event occurred on channel 6.
27	TEIF6	0x0	R	Channel 6 transfer error flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer error (TE) event on channel 6. 1: A transfer error (TE) occurred on channel 6.

Bit	Field name	Reset	RW	Description
28	GIF7	0x0	R	Channel 7 global interrupt flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No TE, HT or TC event on channel 7. 1: A TE, HT or TC event occurred on channel 7.
29	TCIF7	0x0	R	Channel 7 transfer complete flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer complete (TC) on channel 7. 1: A transfer complete (TC) occurred on channel 7.
30	HTIF7	0x0	R	Channel 7 half transfer flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No half transfer (HT) event on channel 7. 1: A half transfer (HT) event occurred on channel 7.
31	TEIF7	0x0	R	Channel 7 transfer error flag. This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the IFCR register. 0: No transfer error (TE) event on channel 7. 1: A transfer error (TE) occurred on channel 7.

Table 45. DMA - IFCR register description: address offset DMA_BASE_ADDR+0x04

Bit	Field name	Reset	RW	Description
0	CGIF0	0x0	W	Channel 0 global interrupt flag. This bit is set by software. 0: No effect. 1: Clears the GIF, TEIF, HTIF and TCIF flags in the ISR register.
1	CTCIF0	0x0	W	Channel 0 transfer complete flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TCIF flag in the ISR register.
2	CHTIF0	0x0	W	Channel 0 half transfer flag. This bit is set by software. 0: No effect. 1: Clears the corresponding HTIF flag in the ISR register.
3	CTEIF0	0x0	W	Channel 0 transfer error flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TEIF flag in the ISR register.
4	CGIF1	0x0	W	Channel 1 global interrupt flag. This bit is set by software. 0: No effect. 1: Clears the GIF, TEIF, HTIF and TCIF flags in the ISR register.
5	CTCIF1	0x0	W	Channel 1 transfer complete flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TCIF flag in the ISR register.
6	CHTIF1	0x0	W	Channel 1 half transfer flag. This bit is set by software. 0: No effect. 1: Clears the corresponding HTIF flag in the ISR register.

Bit	Field name	Reset	RW	Description
7	CTEIF1	0x0	W	Channel 1 transfer error flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TEIF flag in the ISR register.
8	CGIF2	0x0	W	Channel 2 global interrupt flag. This bit is set by software. 0: No effect. 1: Clears the GIF, TEIF, HTIF and TCIF flags in the ISR register.
9	CTCIF2	0x0	W	Channel 2 transfer complete flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TCIF flag in the ISR register.
10	CHTIF2	0x0	W	Channel 2 half transfer flag. This bit is set by software. 0: No effect. 1: Clears the corresponding HTIF flag in the ISR register.
11	CTEIF2	0x0	W	Channel 2 transfer error flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TEIF flag in the ISR register.
12	CGIF3	0x0	W	Channel 3 global interrupt flag. This bit is set by software. 0: No effect. 1: Clears the GIF, TEIF, HTIF and TCIF flags in the ISR register.
13	CTCIF3	0x0	W	Channel 3 transfer complete flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TCIF flag in the ISR register.
14	CHTIF3	0x0	W	Channel 3 half transfer flag. This bit is set by software. 0: No effect. 1: Clears the corresponding HTIF flag in the ISR register.
15	CTEIF3	0x0	W	Channel 3 transfer error flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TEIF flag in the ISR register.
16	CGIF4	0x0	W	Channel 4 global interrupt flag. This bit is set by software. 0: No effect. 1: Clears the GIF, TEIF, HTIF and TCIF flags in the ISR register.
17	CTCIF4	0x0	W	Channel 4 transfer complete flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TCIF flag in the ISR register.
18	CHTIF4	0x0	W	Channel 4 half transfer flag. This bit is set by software. 0: No effect. 1: Clears the corresponding HTIF flag in the ISR register.
19	CTEIF4	0x0	W	Channel 4 transfer error flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TEIF flag in the ISR register.
20	CGIF5	0x0	W	Channel 5 global interrupt flag. This bit is set by software. 0: No effect. 1: Clears the GIF, TEIF, HTIF and TCIF flags in the ISR register.
21	CTCIF5	0x0	W	Channel 5 transfer complete flag. This bit is set by software.

Bit	Field name	Reset	RW	Description
				0: No effect. 1: Clears the corresponding TCIF flag in the ISR register.
22	CHTIF5	0x0	W	Channel 5 half transfer flag. This bit is set by software. 0: No effect. 1: Clears the corresponding HTIF flag in the ISR register.
23	CTEIF5	0x0	W	Channel 5 transfer error flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TEIF flag in the ISR register.
24	CGIF6	0x0	W	Channel 6 global interrupt flag. This bit is set by software. 0: No effect. 1: Clears the GIF, TEIF, HTIF and TCIF flags in the ISR register.
25	CTCIF6	0x0	W	Channel 6 transfer complete flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TCIF flag in the ISR register.
26	CHTIF6	0x0	W	Channel 6 half transfer flag. This bit is set by software. 0: No effect. 1: Clears the corresponding HTIF flag in the ISR register.
27	CTEIF6	0x0	W	Channel 6 transfer error flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TEIF flag in the ISR register.
28	CGIF7	0x0	W	Channel 7 global interrupt flag. This bit is set by software. 0: No effect. 1: Clears the GIF, TEIF, HTIF and TCIF flags in the ISR register.
29	CTCIF7	0x0	W	Channel 7 transfer complete flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TCIF flag in the ISR register.
30	CHTIF7	0x0	W	Channel 7 half transfer flag. This bit is set by software. 0: No effect. 1: Clears the corresponding HTIF flag in the ISR register.
31	CTEIF7	0x0	W	Channel 7 transfer error flag. This bit is set by software. 0: No effect. 1: Clears the corresponding TEIF flag in the ISR register.

- DMA_CH0 peripheral base address (DMA_CH0_BASE_ADDR) 0xA0000008
- DMA_CH1 peripheral base address (DMA_CH1_BASE_ADDR) 0xA000001C
- DMA_CH2 peripheral base address (DMA_CH2_BASE_ADDR) 0xA0000030
- DMA_CH3 peripheral base address (DMA_CH3_BASE_ADDR) 0xA0000044
- DMA_CH4 peripheral base address (DMA_CH4_BASE_ADDR) 0xA0000058
- DMA_CH5 peripheral base address (DMA_CH5_BASE_ADDR) 0xA000006C
- DMA_CH6 peripheral base address (DMA_CH6_BASE_ADDR) 0xA0000080
- DMA_CH7 peripheral base address (DMA_CH7_BASE_ADDR) 0xA0000094

Table 46. **DMA_CHx registers**

Address offset	Name	RW	Reset	Description
0x00	CCR	RW	0x00000000	DMA channel configuration register. Refer to the detailed description below.
0x04	CNDTR	RW	0x00000000	DMA channel number of data register. Refer to the detailed description below.
0x08	CPAR	RW	0x00000000	DMA channel peripheral address register. Refer to the detailed description below.
0x0C	CMAR	RW	0x00000000	DMA channel memory address register. Refer to the detailed description below.

Table 47. **DMA_CHx - CCR register description: address offset DMA_CHX_BASE_ADDR+0x00**

Bit	Field name	Reset	RW	Description
0	EN	0x0	RW	DMA channel enable. 0: DMA channel disabled. 1: DMA channel enabled.
1	TCIE	0x0	RW	Transfer complete interrupt enable. 0: TC interrupt disabled. 1: TC interrupt enabled.
2	HTIE	0x0	RW	Half transfer interrupt enable. 0: HT interrupt disabled. 1: HT interrupt enabled.
3	TEIE	0x0	RW	Transfer error interrupt enable. 0: TE interrupt disabled. 1: TE interrupt enabled.
4	DIR	0x0	RW	Data transfer direction. 0: Read from peripheral. 1: Read from memory.
5	CIRC	0x0	RW	Circular mode. 0: Circular mode disabled. 1: Circular mode enabled.
6	PINC	0x0	RW	Peripheral increment mode. 0: Peripheral increment disabled. 1: Peripheral increment enabled.
7	MINC	0x0	RW	Memory increment mode. 0: Memory increment disabled. 1: Memory increment enabled.
9:8	PSIZE	0x0	RW	Peripheral size. 00b: Size 8 bits. 01b: Size 16 bits. 10b: Size 32 bits.
11:10	MSIZE	0x0	RW	Memory size. 00b: Size 8 bits. 01b: Size 16 bits.

Bit	Field name	Reset	RW	Description
				10b: Size 32 bits.
13:12	PL	0x0	RW	Channel priority level. 00b: Low priority. 01b: Medium priority. 10b: High priority. 11b: Very high priority.
14	MEM2MEM	0x0	RW	Memory-to-memory mode. 0: Memory-to-memory mode disabled. 0: Memory-to-memory mode enabled.
31:15	RESERVED	0x0	RW	RESERVED

Table 48. DMA_CHx - CNDTR register description: address offset DMA_CHX_BASE_ADDR+0x04

Bit	Field name	Reset	RW	Description
15:0	NDT	0x0	RW	Number of data to be transferred (0 up to 65535). This register can only be written when the channel is disabled. Once the channel is enabled, this register is read-only, indicating the remaining bytes to be transmitted. This register decrements after each DMA transfer. Once the transfer is completed, this register can either stay at zero or be reloaded automatically by the value previously programmed if the channel is configured in auto-reload mode. If this register is zero, no transaction can be served whether the channel is enabled or not.
31:16	RESERVED	0x0	RW	RESERVED

Table 49. DMA_CHx - CPAR register description: address offset DMA_CHX_BASE_ADDR+0x08

Bit	Field name	Reset	RW	Description
31:0	PA	0x0	RW	Base address of the peripheral data register from/to which the data are read/written. When PSIZE is 01 (16-bit), the PA[0] bit is ignored. Access is automatically aligned to a halfword address. When PSIZE is 10 (32-bit), PA[1:0] are ignored. Access is automatically aligned to a word address.

Table 50. DMA_CHx - CMAR register description: address offset DMA_CHX_BASE_ADDR+0x0C

Bit	Field name	Reset	RW	Description
31:0	MA	0x0	RW	Base address of the memory area from/to which the data are read/written. When MSIZE is 01 (16-bit), the MA[0] bit is ignored. Access is automatically aligned to a halfword address. When MSIZE is 10 (32-bit), MA[1:0] are ignored. Access is automatically aligned to a word address.

Note: All RESERVED fields inside registers must always be written with their default values.

3.8 SPI

3.8.1 Introduction

The BlueNRG-2 integrates a serial peripheral interface compatible with the Motorola and National Semiconductor Microwire standards.

Main features are:

- Maximal supported baud rate is 1 MHz in slave mode and 8 MHz in master mode.
- Parallel-to-serial conversion on data written to an internal 32-bit wide, 16-location deep transmitter FIFO.
- Serial-to-parallel conversion on received data, buffering in a 32-bit wide 16-location deep receive FIFO.
- Programmable data frame size from 4-bit to 32-bit.
- Programmable clock bit rate and prescaler.
- Programmable clock phase and polarity in SPI mode.
- Support for direct memory access (DMA).

3.8.2 Functional overview

The SPI performs serial-to-parallel conversion on data received from a peripheral device on the SPI_IN pin, and parallel-to-serial conversion on data written by CPU for transmission on the SPI_OUT pin.

The role of the SPI pins are defined by the SPI master or SPI slave role.

Table 51. SPI pin assignments

SPI role	SPI_IN pin	SPI_OUT pin
Master	MISO	MOSI
Slave	MOSI	MISO

The transmit and receive paths are buffered with internal FIFO memories allowing up to 16 x 32-bit values to be stored independently in both transmit and receive modes. FIFOs may be burst-loaded or emptied by the system processor or by the DMA, from one to eight words per transfer. Each 32-bit word from the system fills one entry in FIFO.

The SPI includes a programmable bitrate clock divider and prescaler to generate the serial output clock signal from the SPI_CLK pin.

3.8.2.1 SPI clock phase and clock polarity

The SPH control bit selects the clock edge that captures data and allows it to change state. It has the most impact on the first bit transmitted by either allowing or not allowing a clock transition before the first data capture edge. The SPO bit selects the clock polarity (low or high) of the clock signal. SPH in conjunction with the SPO bit allow four possible timing variations listed in the following table.

Table 52. SPI clock phase and clock polarity

SPH	SPO	Timing description
0b	0b	The clock signal is stopped to low inactive level between transfers. The first rising edge occurs in the middle of the first data bit (with delay). The SPI transmits data one-half cycle ahead of the rising edge of clock signal and receives data on the rising edge of clock signal. In case of multi byte transmission, the CS line must be pulsed HIGH between each data word transfer.
0b	1b	The clock signal is stopped to high inactive level between transfers. The first falling edge occurs in the middle of the first data bit (with delay). The SPI transmits data one-half cycle ahead of the falling edge of clock signal and receives data on the falling edge of clock signal. In case of multi byte transmission, the CS line must be pulsed HIGH between each data word transfer.
1b	0b	The clock signal is stopped to low inactive level between transfers. The first rising edge occurs at the start of the first data bit (no delay). The SPI transmits data on the rising edge of clock signal and receives data on the failing edge of clock signal.

SPH	SPO	Timing description
1b	1b	The clock signal is stopped to high inactive level between transfers. The first falling edge occurs at the start of the first data bit (no delay). The SPI transmits data on the falling edge of clock signal and receives data on the rising edge of clock signal.

3.8.2.2 *Procedure for enabling SPI*

The SPI initialization procedure is the following (assuming clocks already enabled):

1. Clear the SSE bit in the CR1 register. This step is not required after a hardware or software reset of the BlueNRG-2.
2. Empty the receive FIFO. This step is not required after a hardware or software reset of the device BlueNRG-2.
3. Program IO_MODE to route SPI port signals on those GPIOs. See section GPIO operating modes.
4. Program the SPI clock prescaler register (CPSR), then program the configuration registers CR0 and CR1.
5. The transmit FIFO can optionally be filled before enabling the SPI.
6. Set the SSE bit to enable SPI operation.

Note: The transmit FIFO and the receive FIFO are not cleared when the SSE bit is cleared.

3.8.2.3 *SPI bit rate generation*

The SPI bitrate is derived by dividing down the peripheral clock (CLK) by an even prescaler value CPSDVSR from 2 to 254, the clock is further divided by a value from 1 to 256, which is 1+SCR. The SPI frequency clock duty cycle is always 0.5.

3.8.2.4 *SPI data endianness*

All transfers can be sent and received with configurable endianness according the setting of the (T/R)ENDN bit in the CR1 registers

The cases "00b" and "11b" of TENDN and RENDN are implemented for data frame size from 4- to 32-bit. The cases "01b" and "10b" of TENDN and RENDN are implemented only for the following data frame sizes: 16-bit, 24-bit and 32-bit. Transmit data endianness: TENDN in CR1:

Table 53. SPI_OUT endianness

TENDN	Endianness
00b	The element is transmitted MSByte-first and MSbit-first.
01b	The element is transmitted LSByte-first and MSbit-first.
10b	The element is transmitted MSByte-first and LSbit-first.
11b	The element is transmitted LSByte-first and LSbit-first.

Table 54. SPI_IN endianness

RENDN	Endianness
00b	The element is received MSByte-first and MSbit-first.
01b	The element is received LSByte-first and MSbit-first.
10b	The element is received MSByte-first and LSbit-first.
11b	The element is received LSByte-first and LSbit-first.

3.8.2.5 *SPI interrupts*

There are six individual maskable interrupt sources generated by the SPI (single interrupt signal that drives the NVIC):

- Receive interrupt
- Transmit interrupt
- Timeout interrupt

- Receive overrun interrupt
- Transmit underrun interrupt
- Transmit empty interrupt

The user can enable or disable the individual interrupt sources by changing the mask bits in the IMSC register. Setting the appropriate mask bit to 1b enables the interrupt. The status of the individual interrupt sources can be read from the RIS register (raw interrupt status) or from the MIS register (masked interrupt status).

3.8.2.6 *Receive interrupt*

The receive interrupt is asserted when the number of data in receive FIFO reaches the programmed trigger watermark level. The receive interrupt is cleared by reading data from the receive FIFO until there are less data than the programmed watermark level.

3.8.2.7 *Transmit interrupt*

The transmit interrupt is asserted when the number of data in the transmit FIFO is less than or equal to the programmed watermark level. It is cleared by performing writes to the transmit FIFO until it holds more elements than the programmed watermark level. The transmitter interrupt is not qualified with the SPI enable bit, which allows operation in one of two ways:

- Data can be written to the transmit FIFO prior to enabling the SPI and the interrupts.
- Or the SPI and the interrupts can be enabled so that data can be written to the transmit FIFO by an interrupt service routine.

3.8.2.8 *Timeout interrupt*

The receive timeout interrupt is asserted when the receive FIFO is not empty, and no further data is received over a 32-bit period of the serial clock. This mechanism ensures that the user is aware that data is still present in the receive FIFO and requires servicing.

The receive timeout interrupt is cleared either when the FIFO becomes empty through reading all the data, or if new data is received, or when a 1b is written to the corresponding bit of the ICR register.

3.8.2.9 *Receive overrun interrupt*

The receive overrun interrupt is asserted when the receive FIFO is already full and an additional data frame is received, causing an overrun of the FIFO. Data is overwritten in the receive shift register, but not in the FIFO. The interrupt is cleared when a 1b is written to the corresponding bit of the ICR register.

3.8.2.10 *Transmit underrun interrupt*

The transmit underrun interrupt is asserted when the transmit FIFO is already empty and an additional frame is transmitted, causing an underrun of the FIFO. Data is over-read in the transmit shift register. This interrupt is cleared when a 1b is written to the corresponding bit of the ICR register.

3.8.2.11 *Transmit empty interrupt*

The transmit interrupt is asserted when the transmit FIFO is empty. It is cleared by performing writes to the transmit FIFO.

3.8.2.12 *SPI master communication mode for Motorola standard*

The SPIM register field selects the SPI transmission mode, these modes are applicable only for SPI master mode:

- Full duplex mode (SPIM = 00b): the master transmits the data available in the TXFIFO and receives the data from the slave.
- Transmit mode (SPIM = 01b): when the data is available in TX FIFO, the SPI_OUT line is run, and no data is written in RX FIFO

- Receive mode (SPIM = 10b): the sequence of receive mode is:
  1. The software sets the mode to receive (SPIM = 10) and writes the dummy character value to the CHN register.
  2. The software writes the value “number of frames to receive from the slave” in the RXFRM register.

  When the receive mode is selected we have two cases:
  1. If the TXFIFO is empty, the master receives data from the slave, transmitting the character from the CHN register in each frame received. The RXFRM register is decremented by one at each transmission/reception. The interface runs until the RXFRM value is dummy AND the written number of frames in RXFRM is received.
  2. If the TXFIFO is not empty, the master first transmits the data available in the TXFIFO and receives the data from the slave (like the full duplex mode). The RXFRM register is not decremented. When all the data available in TXFIFO are transmitted, the TXFIFO becomes empty (case a), then the dummy character from the CHN register are transmitted for each frame received. The RXFRM register is decremented for each transaction. When the value in this register is zero and the written number of frames in RXFRM is received, the interface does not run anymore. The user has to write the RXFRM (with value greater than zero) by software to reactivate the interface.

Note: In all cases, the RXFRM is decremented by one only if the TXFIFO is empty. The RFRM is decremented before the data is sent.

Note: If the software fills the TXFIFO while the SPI is transmitting the dummy character, then the words of TXFIFO should be ignored and we should not send them in this mode, only the dummy -character are transmitted.

- Combined mode (SPIM = 11b): the sequence of combined mode is:
  1. The software sets the mode to combined mode: (SPIM=11).
  2. The software writes to the WDTXF register the "number of frames to be received by the slave (a value greater than zero) from TXFIFO master".

If the number of words written in the WDTXF register are sent (WDTXF is equal to zero) but the RXFRM register is not equal to zero, the master transmits the dummy character (defined in CHM register) and receives the slave data, decrementing the RXFRM register by one. Once the RXFRM register is equal to zero and all the data written in RXFRM are sent, the interface is stopped.

When the RXFRM register is zero and WDTXF is not equal to zero and the TXFIFO of the master and of slave are not empty, the master transmits the data from TXFIFO and receives the data from the slave. Before the data is sent from TXFIFO, the WDTXF is decremented by one. When the WDTXF register is zero and all the numbers written in this register are sent, the interface is stopped.

The interface is stopped when:

- WDTXF and RXFRM registers are equal to zero.
- WDTXF register is not equal to zero and TXFIFO is empty.
- RXFRM register is not equal to zero, the WDTXF register is zero and TXFIFO is not empty.

For each data transmission (TXFIFO data or CHN register data), the data slave is received.

WDTXF is decremented by one only at each data transmission from TXFIFO. The decrementing is done before the words are sent out.

RXFRM is decremented by one only at each dummy character transmission from the CHN register. The decrementing is done before the words are sent out.

When we start decrementing one of the registers (RXFRM or WDTXF), we must decrement until we reach zero and we send the last words before starting the decrement of other registers (RXFRM or WDTXF).

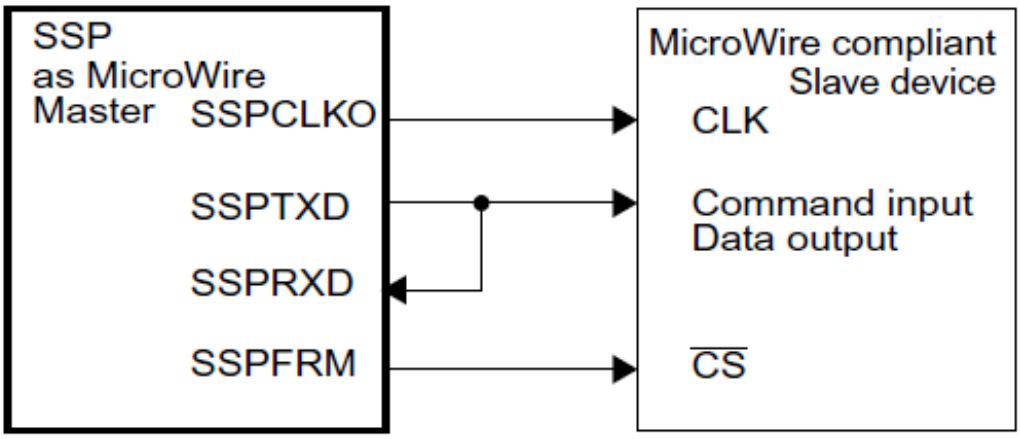
Switching between these different modes when SPI is enabled is possible. If the transmission mode is deselected for another mode during a frame transmission, the new mode becomes active at the start of the next word.

3.8.2.13 *SPI master communication mode for National Semiconductor Microwire standard*

The FRF register field selects the SPI mode interface: the default value 0 indicates the Motorola standard interface, while FRF = 2 selects the Microwire standard interface for the 3-wire SPI mode.

The hardware connection considers the SPI_IN and SPI_OUT pins connected together as shown in the figure below. The two pins are handled internally to send and receive data.

Figure 12. **MicroWire master and slave communication**



The CSS register field specifies the length of the data to send and the DSS register field specifies the length of the data to receive.

In reception mode, the quantity of data to read is specified in the DSS register field. The set of the SSE register field, enables SPI for the communication. The user should wait for data reception to end before attempting any reads. The SSE register field must be reset at the end of the communication.

In transmission mode, the send data are written inside the TX FIFO and the quantity of data to send is specified in the CSS register field. The SSE register field setting enables SPI for the communication. When the busy flag is RESET, SPI communication is terminated. The SSE register field must be reset at the end of the communication.

3.8.3 SPI registers

SPI peripheral base address (SPI_BASE_ADDR) 0x40400000.

Table 55. **SPI registers**

Address offset	Name	RW	Reset	Description
0x00	CR0	RW	0x1C000000	Control register 0. Refer to the detailed description below.
0x04	CR1	RW	0x00000000	Control register 1. Refer to the detailed description below.
0x08	DR	RW	0x00000000	Data register. Refer to the detailed description below.
0x0C	SR	R	0x00000003	Status register. Refer to the detailed description below.
0x10	CPSR	RW	0x00000000	Clock prescale register. Refer to the detailed description below.
0x14	IMSC	RW	0x00000000	Interrupt mask set or clear register. Refer to the detailed description below.
0x18	RIS	R	0x00000000	Raw interrupt status register. Refer to the detailed description below.
0x1C	MIS	R	0x00000000	Masked interrupt status register. Refer to the detailed description below.
0x20	ICR	W	0x00000000	Interrupt clear register. Refer to the detailed description below.
0x24	DMACR	RW	0x00000000	SPI DMA control register. Refer to the detailed description below.
0x28	RXFRM	RW	0x00000000	SPI receive frame register. Indicates the number of frames to receive from the slave.
0x2C	CHN	RW	0x00000000	Dummy character register
0x30	WDTXF	RW	0x00000000	SPI transmit FIFO receive frame number. Indicates the number of frames to receive from the transmit FIFO
0x80	ITCR	RW	0x00000000	Integration test control register. Refer to the detailed description below.
0x8C	TDR	RW	0x00000000	Test data register

Table 56. SPI - CR0 register description: address offset SPI_BASE_ADDR+0x00

Bit	Field name	Reset	RW	Description
4:0	DSS	0x0	RW	Data size select. (DSS+1) defines the number of bits: 0x00: Reserved. 0x01: Reserved. 0x02: Reserved. 0x03: 4-bit data. 0x04: 5-bit data. ... 0x1F: 32-bit data.
5	RESERVED	0x0	RW	RESERVED
6	SPO	0x0	RW	Clock polarity (Motorola SPI format only): 0: Steady-state of clock polarity is low. 1: Steady-state of clock polarity is high.
7	SPH	0x0	RW	Clock phase (Motorola SPI format only): 0: Steady-state of clock phase is low. 1: Steady-state of clock phase is high.
15:8	SCR	0x0	RW	Serial clock rate. The SRC value is used to generate the transmit and receive bit rate of the SPI. The bit rate is: f_SPICLK / (CPSDVR * (1 + SCR)), where CPSDVR is an even value from 2 to 254 and SCR is a value from 0 to 255.
20:16	CSS	0X00	RW	Command size select (CSS+1) defines the number of bits for the command in MicroWire mode only: 0x00: Reserved 0x01: Reserved 0x02: Reserved 0x03: 4-bit data 0x04: 5-bit data ... 0x1F: 32-bit data
21:22	FRF	0x00	RW	Frame format: 0: Motorola SPI frame format 2: National MicroWire frame format
24:23	SPIM	0x0	RW	SPI master transmission mode (in Motorola SPI master mode only): 00b: Full duplex mode. 01b: Transmit mode. 10b: Receive mode. 11b: Combined mode.
25	RESERVED	0x0	RW	RESERVED
26	CS1	0x1	RW	Chip selection for slave one 0: the slave 1 is selected. 1: the slave 1 is not selected.
31:27	RESERVED	0x3	RW	RESERVED

Table 57. **SPI - CR1 register description: address offset SPI_BASE_ADDR+0x04**

Bit	Field name	Reset	RW	Description
0	RESERVED	0x0	RW	RESERVED
1	SSE	0x0	RW	SPI enable. 0: SPI disable. 1: SPI enable.
2	MS	0x0	RW	Master or slave mode select. 0: Master mode. 1: Slave mode.
3	SOD	0x0	RW	Slave mode output disable (slave mode only). 0: SPI can drive the MISO signal in slave mode. 1: SPI must not drive the MISO signal in slave mode. In multiple slave system, it is possible for a SPI master to broadcast a message to all slaves in the system while ensuring only one slave drives data onto the serial output line MISO.
5:4	RENDN	0x0	RW	Receive endian format. 00b: The element is received MSByte-first and MSbit-first. 01b: The element is received LSByte-first and MSbit-first. 10b: The element is received MSByte-first and LSbit-first. 11b: The element is received LSByte-first and LSbit-first. The cases 00b and 11b are set for data frame size from 4 to 32 bits. The cases 01b and 10b are set only for data frame size 16, 24 and 32 bits.
6	MWAIT	0x0	RW	MicroWire wait state bit enable: 0: No wait state 1: One wait state
9:7	RXIFLSEL	0x0	RW	Receive interrupt FIFO level select. This bit field selects the trigger points to receive FIFO interrupt: 000b: RX FIFO contains 1 element or more. 001b: RX FIFO contains 4 elements or more. 010b: RX FIFO contains 8 elements or more. Others: Reserved.
12:10	TXIFLSEL	0x0	RW	Transmit interrupt FIFO level select. This bit field selects the trigger points to transmit FIFO interrupt: 000b: TX FIFO contains 1 element or more. 001b: TX FIFO contains 4 elements or more. 010b: TX FIFO contains 8 elements or more. Others: Reserved.
13	RESERVED	0x0	RW	RESERVED
17:14	MSPIWAIT	0x0	RW	Motorola SPI wait mode. This value is used to insert a wait state between frames.
19:18	TENDN	0x0	RW	Transmit endian format. 00b: The element is transmitted MSByte-first and MSbit-first. 01b: The element is transmitted LSByte-first and MSbit-first. 10b: The element is transmitted MSByte-first and LSbit-first. 11b: The element is transmitted LSByte-first and LSbit-first.

Bit	Field name	Reset	RW	Description
				The cases 00b and 11b are set for data frame size from 4 to 32 bits. The cases 01b and 10b are set only for data frame size 16, 24 and 32 bits.
20	RESERVED	0x0	RW	RESERVED
21	DATAINDEL	0x0	RW	Data input delay. 0: No delay is inserted in data input. 1: A delay of 2-clock cycles is inserted in the data input path.
31:22	RESERVED	0x0	RW	RESERVED

Table 58. SPI - DR register description: address offset SPI_BASE_ADDR+0x08

Bit	Field name	Reset	RW	Description
31:0	DATA	0x0	RW	Transmit/receive data: Read: RX FIFO is read. Write: TX FIFO is written. Data must be right-justified when a data size of less than 32-bit is programmed. Unused bits are ignored by the transmit logic. The receive logic automatically right-justifies data.

Table 59. SPI - SR register description: address offset SPI_BASE_ADDR+0x0C

Bit	Field name	Reset	RW	Description
0	TFE	0x1	R	Transmit FIFO empty: 0: TX FIFO is not empty. 1: TX FIFO is empty.
1	TNF	0x1	R	Transmit FIFO not full: 0: TX FIFO is full. 1: TX FIFO is not full.
2	RNE	0x0	R	Receive FIFO not empty: 0: RX FIFO is empty. 1: RX FIFO is not empty.
3	RFF	0x0	R	Receive FIFO full: 0: RX FIFO is not full. 1: RX FIFO is full.
4	BSY	0x0	R	SPI busy flag: 0: SPI is idle. 1: SPI is currently transmitting and/or receiving a frame or the TX FIFO is not empty.
31:5	RESERVED	0x0	R	RESERVED

Table 60. SPI - CPSR register description: address offset SPI_BASE_ADDR+0x10

Bit	Field name	Reset	RW	Description
7:0	CPSDVSR	0x0	RW	Clock prescaler divisor. It must be an even number from 2 to 254. The value is used to generate the transmit and receive bit rate of the SPI. The bit rate is:

Bit	Field name	Reset	RW	Description
				FSSPCLK / [CPSDVR x (1+SCR)] where SCR is a value from 0 to 255, programmed through the SSP_CR0 register.
31:8	RESERVED	0x0	RW	RESERVED

Table 61. SPI - IMSC register description. Address offset SPI_BASE_ADDR+0x14.

Bit	Field name	Reset	RW	Description
0	RORIM	0x0	RW	Receive overrun interrupt mask: 0: RX FIFO written to while full condition interrupt is masked (irq disabled). 1: RX FIFO written to while full condition interrupt is not masked (irq enabled).
1	RTIM	0x0	RW	Receive timeout interrupt mask: 0: RX FIFO not empty or no read prior to the timeout period interrupt is masked (irq disabled). 1: RX FIFO not empty or no read prior to the timeout period interrupt is not masked (irq enabled).
2	RXIM	0x0	RW	Receive FIFO interrupt mask: 0: Receive interrupt is masked (irq disabled). 1: Receive interrupt is not masked (irq enabled).
3	TXIM	0x0	RW	Transmit FIFO interrupt mask: 0: Transmit interrupt is masked (irq disabled). 1: Transmit interrupt is not masked (irq enabled).
4	TURIM	0x0	RW	Transmit underrun interrupt mask: 0: Transmit underrun interrupt is masked (irq disabled). 1: Transmit underrun interrupt is not masked (irq enabled).
5	TEIM	0x0	RW	Transmit FIFO empty interrupt mask: 0: TX FIFO empty interrupt is masked (irq disabled). 1: TX FIFO empty interrupt is not masked (irq enabled).
31:6	RESERVED	0x0	RW	RESERVED

Table 62. SPI - RIS register description: address offset SPI_BASE_ADDR+0x18

Bit	Field name	Reset	RW	Description
0	RORRIS	0x0	R	Receive overrun raw interrupt status
1	RTRIS	0x0	R	Receive time out raw interrupt status
2	RXRIS	0x0	R	Receive raw interrupt status
3	TXRIS	0x0	R	Transmit raw interrupt status
4	TURRIS	0x0	R	Transmit underrun raw interrupt Status
5	TERIS	0x0	R	Transmit FIFO empty raw interrupt status
31:6	RESERVED	0x0	R	RESERVED

Table 63. SPI - MIS register description: address offset SPI_BASE_ADDR+0x1C

Bit	Field name	Reset	RW	Description
0	RORMIS	0x0	R	Receive overrun masked interrupt status: gives the interrupt status after masking of the receive overrun interrupt.
1	RTMIS	0x0	R	Receive time out masked interrupt status: gives the interrupt status after masking of receive timeout interrupt.
2	RXMIS	0x0	R	Receive masked interrupt status: gives the interrupt status after masking of the receive interrupt.
3	TXMIS	0x0	R	Transmit masked interrupt status: gives the interrupt status after masking of the transmit interrupt.
4	TURMIS	0x0	R	Transmit underrun masked interrupt status: gives the interrupt status after masking of the transmit underrun interrupt.
5	TEMIS	0x0	R	Transmit FIFO empty masked interrupt status: gives the interrupt status after masking of the transmit FIFO empty interrupt.
31:6	RESERVED	0x0	R	RESERVED

Table 64. SPI - ICR register description: address offset SPI_BASE_ADDR+0x20

Bit	Field name	Reset	RW	Description
0	RORIC	0x0	W	Receive overrun clear interrupt: writing 1 clears the receive overrun interrupt.
1	RTIC	0x0	W	Receive time out clear interrupt: writing 1 clears the receive timeout interrupt.
2	TURIC	0x0	W	Transmit underrun clear interrupt: writing 1 clears the transmit overrun interrupt.
31:3	RESERVED	0x0	W	RESERVED

Table 65. SPI - DMACR register description: address offset SPI_BASE_ADDR+0x24

Bit	Field name	Reset	RW	Description
0	RXDMASE	0x0	RW	Single receive DMA request. 0: Single transfer DMA in receive disable. 1: Single transfer DMA in receive enable.
1	RESERVED	0x0	RW	RESERVED
2	TXDMASE	0x0	RW	Single transmit DMA request. 0: Single transfer DMA in transmit disable. 1: Single transfer DMA in transmit enable.
31:3	RESERVED	0x0	RW	RESERVED

Table 66. SPI – RXFRM register description: address offset SPI_BASE_ADDR+0x28

Bit	Field name	Reset	RW	Description
15:0	RXFRM	0x0000	RW	SPI receive frame register. Indicates the number of frames to receive from the slave.
31:16	RESERVED	0x0000	RW	RESERVED

Table 67. SPI – CHN register description: address offset SPI_BASE_ADDR+0x2C

Bit	Field name	Reset	RW	Description
31:0	CHN	0x00000000	RW	Dummy character register.

Table 68. SPI – WDTXF register description: address offset SPI_BASE_ADDR + 0x30

Bit	Field name	Reset	RW	Description
15:0	WDTXF	0x0000	0x0000	SPI transmit FIFO receive frame number. Indicates the number of frames to receive from the TX FIFO.
31:16	RESERVED	0x0000	0x0000	RESERVED

Table 69. SPI - ITCR register description: address offset SPI_BASE_ADDR+0x80

Bit	Field name	Reset	RW	Description
0	RESERVED	0x0	RW	RESERVED
1	SWAPFIFO	0x0	RW	FIFO control mode: 0: FIFO normal mode. Write in TDR register puts data in TX FIFO and read from TDR register read data from RX FIFO. 1: FIFO swapped mode. Write in TDR register puts data in RX FIFO and read from TDR register read data from TX FIFO. The registers SWAPFIFO and TDR allow the TX FIFO to be cleared from unwanted data.
31:2	RESERVED	0x0	RW	RESERVED

Table 70. SPI – TDR register description: address offset SPI_BASE_ADDR+0x8C

Bit	Field name	Reset	RW	Description
31:0	TDR	0x00000000	RW	Allows reading out data from TX FIFO if the SWAPFIFO bitfield is set. The registers SWAPFIFO and TDR allow the TX FIFO to be cleared from unwanted data.

Note: All RESERVED fields inside registers must always be written with their default values.

3.9 UART

3.9.1 Introduction

The BlueNRG-2 integrates a universal asynchronous receiver/transmitter that support much of the functionality of the industry-standard 16C650 UART.

Main features are:

- Programmable baud rates up to 2 Mbps.
- Programmable data frame of 5, 6, 7 or 8 bits of data.
- Even, odd, stick or no-parity bit generation and detection.
- Programmable 1 or 2 stop bit.
- Support of hardware flow control using CTS and RTS pins.
- Support of software flow control using programmable Xon/Xoff characters
- False start bit detection.

- Line break generation and detection.
- Programmable 8-bit wide, 64-deep transmit FIFO and 12-bit wide (8-bit data and 4-bit status) , 64-deep receive FIFO.
- Support for direct memory access (DMA).

3.9.2 Functional description

The UART performs serial-to-parallel conversion on data asynchronously received from a peripheral device on the UART_RX pin, and parallel-to-serial conversion on data written by CPU for transmission on the UART_TX pin. The transmit and receive paths are buffered with internal FIFO memories allowing up to 64 data byte for transmission, and 64 data byte with 4-bit status (break, frame, parity, and overrun) for receive. FIFOs may be burst-loaded or emptied by the system processor from 1 to 16 words per transfer.

3.9.2.1 Data transmission or reception

Data received or transmitted is stored in two 64-byte FIFOs. The receive FIFO has an extra four bits per character for the status information:

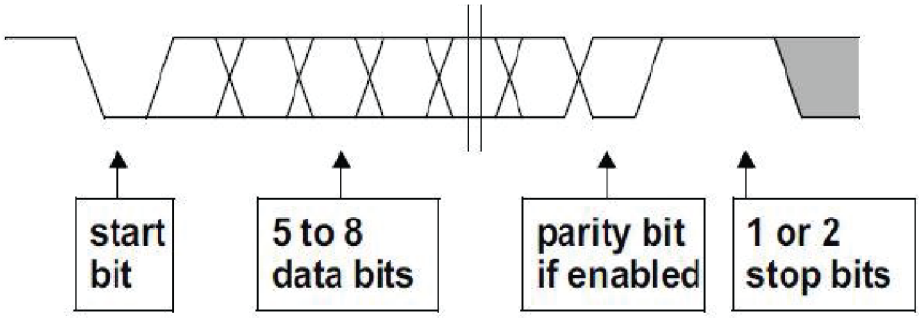
- Error bits 8 to 10 are associated with a particular character: break error, parity error and framing error.
- Overrun indicator bit 11 is set when the FIFO is full, and the next character is completely received in the shift register. The data in the shift register is overwritten, but it is not written into the FIFO. When an empty location is available in the receive FIFO, and another character is received, the state of the overrun bit is copied into the received FIFO along with the received character. The overrun state is then cleared.

Table 71. RX FIFO errors

FIFO bit	Function
11	Overrun indicator
10	Break error
9	Parity error
8	Framing error
7:0	Received data

For transmission, data is written into the transmit FIFO. If the UART is enabled, it causes a data frame to start transmitting with the parameters indicated in LCRH_TX. Data continue to be transmitted until there is no data left in the transmit FIFO. The BUSY flag in the UARTFR register is set as soon as data is written to the transmit FIFO (that is, the FIFO is non-empty) and remains asserted while data is being transmitted. BUSY is cleared only when the transmit FIFO is empty, and the last character has been transmitted from the shift register, including the stop bits. BUSY can be set even though the UART might no longer be enabled. For each sample of data, three readings are taken and the majority value is kept. In the following paragraphs, the middle sampling point is defined, and one sample is taken either side of it. When the receiver detect a start bit, the receive counter runs and data is sampled on the 8th cycle of that counter in normal UART mode. The start bit is valid if UART_RX signal is still low on the eighth cycle of Baud16, otherwise a false start bit is detected and it is ignored. If the start bit is valid, successive data bits are sampled on every 16th cycle of Baud16 (that is 1-bit period later) according to the programmed length of the data characters. The parity bit is then checked if parity mode was enabled. Lastly, a valid stop bit is confirmed if UART_RX signal is high, otherwise a framing error has occurred. When a full word is received, the data is stored in the receive FIFO, with any error bits associated with that. The UART character frame is shown in Figure 13. UART character frame below.

Figure 13. **UART character frame**



The FIFOs can be disabled. In this case, the transmit and receive sides of the UART have 1-byte holding registers (the bottom entry of the FIFOs). The overrun bit is set when a word has been received, and the previous one was not yet read. In this implementation, the FIFOs are not physically disabled, but the flags are manipulated to give the illusion of a 1-byte register.

3.9.2.2 *Baud rate divisor*

The baud rate divisor is a 22-bit number consisting of a 16-bit integer (BRDI) and a 6-bit fractional part (BRDF). The fractional baud rate divider enables the use of any clock to act as UART_CLK, while it is still possible to generate all the standard baud rates.

The 16-bit integer is loaded through the UART_IBRD register and the 6-bit fractional part is loaded into the UART_FBRD register. The baud rate divisor has the following relationship:

When bit OVSFACT = 0b: divisor = UARTCLK/(16 x baud rate) = BRDI + BRDF.

When bit OVSFACT = 1b: divisor = UARTCLK/(8 x baud rate) = BRDI + BRDF.

Calculation of the fractional 6-bit number (DIVFRAC) is done by taking the fractional part of the required baud rate divisor and multiplying it by 64 (that is, 2n, where n is the width of the UART_FBRD register) and adding 0.5 to account for rounding errors:

DIVFRAC = integer(BRDF * 64 + 0.5)

The maximum deviation error using a 6-bit UART_FBRD register is 1/64 * 100 = 1.56%.

This occurs when DIVFRAC = 1, and the error is cumulative over 64 clock ticks.

Example 1: Calculating the divisor value (with OVSFACT = 0b).

If the required baud rate is 460 800 and the UART clock frequency is 16 MHz then:

Baud rate divisor = $(16 \times 10^6) / (16 \times 460\ 800) = 2.170$

BRDI = 2 and BRDF = 0.170

Therefore fractional part DIVFRAC = integer(BRDF * 64 + 0.5) = 11

Generated baud rate divider = 2 + (11 / 64) = 2.171875

Generated baud rate = $(16 \times 10^6) / (16 \times 2.171875) = 460\ 431$

Error = (460 800 - 460 431) / 460 800 x 100 = 0.08%

An internal clock enable signal, BaudI6, is generated, and is a stream of one UARTCLK wide pulses with an average frequency of 16 (OVSFACT = 0b) or 8 (OVSFACT = 1b) times the desired baud rate. This signal is then divided by 16 or 8 to give the transmit clock. A low number in the baud rate divisor gives a short bit period, and a high number in the baud rate divisor gives a long bit period.

Table 72. Typical baud rates with OVSFACT = 0 shows some typical bit rates and their corresponding divisors when OVSFACT = 0b, given the UART clock frequency of 16 MHz.

Table 72. Typical baud rates with OVSFACT = 0

Required bit rate (bps)	Programmed divisor		Generated bit rate (bps)	Error (%)
	Integer (DIVINT)	Fraction (DIVFRAC)		
921 600	1 (16'h0001)	5 (6'h05)	927 557	0.646
460 800	2 (16'h0002)	11 (6'h0B)	460 447	- 0.077
230 400	4 (16'h0004)	22 (6'h16)	230 218	- 0.079
115 200	8 (16'h0008)	44 (6'h2C)	115 107	- 0.081
57 600	17 (16'h0011)	23 (6'h17)	57 606	0.010
38 400	26 (16'h001A)	3 (6'h03)	38 392	- 0.021
28 800	34 (16'h0022)	46 (6'h2E)	28 802	0.007
19 200	52 (16'h0034)	5 (6'h05)	19 201	0.005
9 600	104 (16'h0068)	11 (6'h0B)	9 599	- 0.010
2 400	416 (16'h01A0)	43 (6'h2B)	2 399	- 0.042
1 200	833 (16'h04B0)	21 (6'h15)	1 200	0
300	3333 (16'h0D05)	21 (6'h15)	300	0
110	9090 (16'h2382)	58 (6'h3A)	110	0

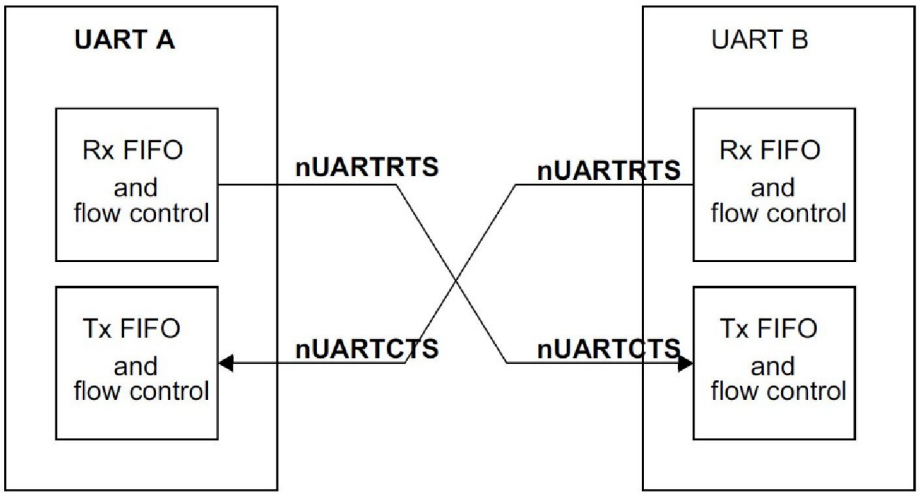
Table 73. Typical baud rates with OVSFACT = 1

Required bit rate (bps)	Programmed divisor		Generated bit rate (bps)	Error (%)
	Integer (DIVINT)	Fraction (DIVFRAC)		
1 843 200	1 (16'h0001)	5 (6'h05)	1 855 115	0.646
921 600	2 (16'h0002)	11 (6'h0B)	920 895	- 0.076
460 800	4 (16'h0004)	22 (6'h16)	461 436	- 0.079
230 400	8 (16'h0008)	44 (6'h2C)	230 215	- 0.080
115 200	17 (16'h0011)	23 (6'h17)	115 212	0.010
57 600	34 (16'h0022)	46 (6'h2E)	57 605	0.009
38 400	52 (16'h0034)	5 (6'h05)	38 403	0.008
28800	69 (16'h0045)	28 (6'h1C)	28 802	0.007
19 200	104 (16'h0068)	11 (6'h0B)	19 199	- 0.005
9 600	208 (16'h00D0)	21 (6'h15)	9 600	0
2 400	833 (16'h0341)	21 (6'h15)	2 400	0
1 200	1666 (16'h0682)	43 (6'h2B)	1 199	-0.083
300	6666 (16'h1A0A)	43 (6'h2B)	299	-0.333
110	18181 (16'h4705)	52 (6'h34)	110	0

3.9.2.3 *Hardware flow control*

The hardware flow controls feature is fully selectable through RTSEN and CTSEN in UARTCR register, and allows to control the serial data flow by using the UART_RTS output and UART_CTS input signals.

Figure 14. **Hardware flow control between two similar devices**



When the RTS flow control is enabled, the UART_RTS signal is asserted until the receive FIFO is filled up to the programmed watermark level. When the CTS flow control is enabled, the transmitter can only transmit data when the UART_CTS signal is asserted.

Table 74. **Control bits to enable and disable hardware flow control**

CTSEN	RTSEN	Function
0b	0b	Both RTS and CTS flow control disabled.
0b	1b	Only RTS flow control enabled.
1b	0b	Only CTS flow control enabled.
1b	1b	Both RTS and CTS flow control enabled.

The RTS flow control logic is linked to the programmable receive FIFO watermark levels. When RTS flow control is enabled, the UART_RTS is asserted until the receive FIFO is filled up to the watermark level. When the receive FIFO watermark level is reached, the UART_RTS signal is de-asserted, indicating that there is no more room to receive any more data. The transmission of data is expected to cease after the current character has been transmitted. The UART_RTS signal is reasserted when data has been read out of the receive FIFO so that it is filled to less than the watermark level. If RTS flow control is disabled and the UART is still enabled, then data is received until the receive FIFO is full, or no more data is transmitted to it.

If CTS flow control is enabled, then the transmitter checks the UART_CTS signal before transmitting the next byte. If the UART_CTS signal is asserted, it transmits the byte otherwise, transmission does not occur. The data continues to be transmitted while UART_CTS is asserted, and the transmit FIFO is not empty. If the transmit FIFO is empty and the UART_CTS signal is asserted no data is transmitted. If the UART_CTS signal is de-asserted and CTS flow control is enabled, then the current character transmission is completed before stopping. If CTS flow control is disabled and the UART is enabled, then the data continues to be transmitted until the transmit FIFO is empty.

3.9.2.4 *Software flow control*

Software flow control is enabled through register UART_XFCR bit SFEN.

Software receive flow control

Once the software receive flow control is enabled, the receiver compares the incoming data with the programmed Xoff values. Different combinations of software receive flow control, which can be selected through SFRMOD, where only 1 character match is needed or in which 2 Xoff characters must be received sequentially.

Table 75. Control bits to enable and program receive software flow control

SFSEN	SFRMOD	Function
0b	xxb	Software flow control disabled.
1b	00b	Software receive flow control disabled.
1b	01b	Use Xon1, Xoff1 for matching.
1b	10b	Use Xon2, Xoff2 for matching.
1b	11b	Use Xon1 & Xon2, Xoff1 & Xoff2 for matching.

If received characters match the programmed Xoff values, the transmission stops as soon as the current character is completely transferred. The interrupt bit XOFFRIS in the raw interrupt register UART_RIS is set. If the corresponding interrupt mask bit is set, the corresponding bit in the UART_MIS register is set and the UART interrupt pin is asserted. Following such a transmission suspension, the receiver will monitor incoming characters for a match with the programmed Xon values. The matching strategy is programmable through SWRFCPROG in register UART_XFCR. Once a match is found, the receiver clears the interrupt bit XOFFRIS in the raw interrupt register UART_RIS and the Xoff interrupt is disabled. The transmission can then resume normally. When the XONANY bit in register UART_XFCR is set, any incoming character is accepted as a valid Xon condition and the transmission can then resume. The received character is written into the received FIFO.

Note: If the software flow control is enabled, the received Xon/Xoff characters are never written into the received FIFO. Exceptions to this occur when the special character detection feature is enabled (Xoff2 is then written into FIFO upon a special character match) and when the Xon-any bit is set.

Note: The received status (parity, framing and break error) of Xon/Xoff characters does not have to be valid for these characters to be accepted as valid matches.

When the software transmit flow control is enabled through the SFTMOD bit field in the UART_XFCR register, the transmitter will automatically insert an Xoff character if the received FIFO has passed the received trigger level (bit field RXIFLSEL in the UART_IFLS register). The RTXDIS (remote transmitter disabled) bit in the UART_FR register is set to signal the remote transfer was stopped. When the receive FIFO falls below the trigger level, an Xon character is automatically inserted in the transmission stream and the RTXDIS bit in the UART_FR register is cleared.

Table 76. Control bits to enable and program transmit software flow control

SFSEN	SFTMOD	Function
0b	xxb	Software flow control disabled.
1b	00b	Software transmit flow control disabled.
1b	01b	Use Xon1, Xoff1 for matching.
1b	10b	Use Xon2, Xoff2 for matching.
1b	11b	Use Xon1 and Xon2, Xoff1 and Xoff2 for matching.

Note: After an Xoff character has been transmitted, if the software flow controlled is turned off, a Xon character is automatically be inserted in the transmission stream and the bit RTXDIS bit in the UART_FR register is cleared.

Note: Transmission of an Xon/Xoff character follows the standard transmission protocol as programmed in the transmitter registers (word length, parity and so on).

Note: When using the software transmit flow control, there are some cautions to take to manage the interrupt handler. The software must react on RX interrupt (the flag is raised once the RX FIFO contains the RXIFSEL trigger level. Then the interrupt handler must first poll the UART_FR.RTXDIS bit until it is set to indicate the Xoff byte transmission is over and only then read the RX FIFO content. Note that hardware and software flow control cannot be enabled simultaneously.

Software transmit flow control

When the special character detection feature is enabled through the SPECHAR bit in the UART_XFCR register, the software flow control is turned off and the receiver compares received characters with the Xoff2 value. When a match is found, the interrupt bit XOFFRIS in the raw interrupt register UART_RIS is set. If the corresponding interrupt mask bit is set, the UART interrupt pin is asserted. The transmission is not halted. The special character is written into the received FIFO. The interrupt bit XOFFIS will be cleared when the corresponding bit in interrupt clear register is written as 1b.

Note: It is assumed that software flow control is turned off when this feature is used. The received status (i.e. parity, framing and break error) of special characters does not have to be valid for these characters to be accepted as valid matches.

3.9.2.5 *UART interrupts*

There are six individual maskable interrupt sources generated by the UART (single interrupt signal that drives the NVIC):

- TX FIFO empty interrupt
- Xoff/ special character interrupt
- Receive interrupt
- Transmit interrupt
- Timeout interrupt
- Error interrupt

The user can enable or disable the individual interrupt sources by changing the mask bits in the UART_IMSC register. Setting the appropriate mask bit to 1b enables the interrupt. The status of the individual interrupt sources can be read from the UART_RIS register (raw interrupt status) or from the UART_MIS register (masked interrupt status).

3.9.2.6 *TX FIFO empty interrupt*

The TX FIFO empty interrupt is asserted whenever the BUSY status bit goes low to indicate that all DATA has been transmitted. This BUSY bit remains set until the complete byte, including all the stop bits, has been sent from the shift register. So, the new TX FIFO empty interrupt is asserted when the transmit FIFO and the transmit shift register are empty.

3.9.2.7 *Xoff/ special character interrupt*

The Xoff/special character interrupt is asserted whenever an Xoff condition is detected by the receiver (incoming data matches with the programmable Xoff values), or when a special character detection was performed (incoming data matches with the Xoff2 and SPECHAR bit set).

3.9.2.8 *Receive interrupt*

The receive interrupt is asserted HIGH when one of the following conditions occurs:

- If the FIFOs are enabled and the number of characters received reaches the programmed trigger watermark level. The receive interrupt is cleared by reading data from the receive FIFO until it becomes less than the programmed watermark level, or by clearing the interrupt by writing a 1b to the corresponding bit in the UART_ICR register.
- If the FIFOs are disabled (have a depth of one location) and there is a data present in the receiver single location. It is cleared by performing a single read.

3.9.2.9 *Transmit interrupt*

The transmit interrupt is asserted HIGH when one of the following conditions occurs:

- If the FIFOs are enabled and the number of characters in the transmit FIFO is less than the programmed watermark level. It is cleared by performing writes to the transmit FIFO until it holds more characters than the programmed watermark level, or by clearing the interrupt by software.

- If the FIFOs are disabled (have a depth of one location) and there is no data present in the transmitter single location. It is cleared by performing a single write to the transmit FIFO, or by clearing the interrupt by software.

Note: The transmit FIFO service interrupt is based on a transition through a level, rather than on the level itself. When the interrupt and the UART are enabled before any data is written to the transmit FIFO, the interrupt is not set. The interrupt is only set once written data leaves the single location of the transmit FIFO and it becomes empty.

Note: When the TX FIFO is disabled, the DATA can be written on the bottom of the FIFO during the transmission of a previous DATA, or in another words, when the holding register is busy.

Note: The interrupt is de-asserted when we write the next DATA on the bottom of the TX FIFO. If we write DATA only on the holding register and the bottom of the TX FIFO is empty, the only way to clear the interrupt is by the software.

3.9.2.10 *Timeout interrupt*

The receive timeout interrupt is asserted when the receive FIFO is not empty, and no further data is received (or no correct start bit of a frame is detected in the RX line) over a programmable timeout period. This mechanism ensures that the user is aware that data is still present in the receive FIFO and requires servicing. The receive timeout interrupt is cleared either when the FIFO becomes empty through reading all the data (or by reading the holding register), or when a 1b is written to the corresponding bit of the UART_ICR register.

3.9.2.11 *Error interrupt*

The error interrupt is asserted when an error occurs in the reception of data by the UART. The interrupt can be caused by the following error conditions:

- Framing
- Parity
- Break
- Overrun

The cause of the interrupt is available by reading the UART_RIS or UART_MIS registers. The interrupt can be cleared by writing to the relevant bits of the UART_ICR register.

3.9.3 UART registers

UART peripheral base address (UART_BASE_ADDR) 0x40300000.

Table 77. UART registers

Address offset	Name	RW	Reset	Description
0x00	DR	RW	0x00000000	Data register. Refer to the detailed description below.
0x04	RSR	R	0x00000000	Receive status register. Refer to the detailed description below.
0x04	ECR	W	0x00000000	Error clear register. A write to this register clears the framing (FE), parity (PE), break (BE), and overrun (OE) errors.
0x0C	TIMEOUT	RW	0x000001FF	Timeout register. Refer to the detailed description below.
0x18	FR	R	0x00001E90	Flag register. Refer to the detailed description below.
0x1C	LCRH_RX	RW	0x00000000	Receive line control register. Refer to the detailed description below.
0x24	IBRD	RW	0x00000000	Integer baud rate register. Refer to the detailed description below.
0x28	FBRD	RW	0x00000000	Fractional baud rate register. Refer to the detailed description below.
0x2C	LCRH_TX	RW	0x00000000	Transmit line control register. Refer to the detailed description below.
0x30	CR	RW	0x00040300	Control register. Refer to the detailed description below.
0x34	IFLS	RW	0x00000012	Interrupt FIFO level select register. Refer to the detailed description below.
0x38	IMSC	RW	0x00000000	Interrupt mask set/clear register. Refer to the detailed description below.

Address offset	Name	RW	Reset	Description
0x3C	RIS	R	0x00000000	Raw interrupt status register. Refer to the detailed description below.
0x40	MIS	R	0x00000000	Masked interrupt status register. Refer to the detailed description below.
0x44	ICR	W	0x00000000	Interrupt clear register. Refer to the detailed description below.
0x48	DMACR	RW	0x00000000	DMA control register. Refer to the detailed description below.
0x50	XFCR	RW	0x00000000	XON/XOFF control register. Refer to the detailed description below.
0x54	XON1	RW	0x00000000	Register used to store the Xon1 character used for software flow control. Refer to the detailed description below.
0x58	XON2	RW	0x00000000	Register used to store the Xon2 character used for software flow control. Refer to the detailed description below.
0x5C	XOFF1	RW	0x00000000	Register used to store the Xoff1 character used for software flow control. Refer to the detailed description below.
0x60	XOFF2	RW	0x00000000	Register used to store the Xoff2 character used for software flow control. Refer to the detailed description below.

Table 78. UART - DR register description: address offset UART_BASE_ADDR+0x00

Bit	Field name	Reset	RW	Description
7:0	DATA	0x0	RW	UART data register: Receive: read data character. Transmit: write data character.
8	FE	0x0	R	Frame error. This bit is set to 1 if the received character did not have a valid stop bit. In FIFO mode, this error is associated with the character at the top of the FIFO.
9	PE	0x0	R	Parity error. This bit is set to 1 if the parity of the received data character does not match the parity selected as defined by bits 2 and 7 of the LCRH_RX register. In FIFO mode, this error is associated with the character at the top of the FIFO.
10	BE	0x0	R	Break error. This bit is set to 1 if a break condition was detected, indicating that the received data input was held low for longer than a full-word transmission time (defined as start, data, parity and stop bits). In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to HIGH (marking state), and the next valid start bit is received
11	OE	0x0	R	Overrun error. This bit is set to 1 if data is received and the receive FIFO is already full. This is cleared to 0b once there is an empty space in the FIFO and a new character can be written to it. The FIFO content remains valid since no further data is written when the FIFO is full, only the content of the shift register is overwritten.
31:12	RESERVED	0x0	R	RESERVED

Table 79. UART - RSR register description: address offset UART_BASE_ADDR+0x04

Bit	Field name	Reset	RW	Description
0	FE	0x0	R	Frame error. This bit is set to 1 if the received character did not have a valid stop bit (a valid stop bit is 1).This bit is cleared to 0b after a write to ECR. In FIFO mode, this error is associated with the character at the top of the FIFO.

Bit	Field name	Reset	RW	Description
1	PE	0x0	R	Parity error. This bit is set to 1 if the parity of the received data character does not match the parity selected as defined by bits 2 and 7 of the LCRH_RX register. This bit is cleared to 0b after a write to ECR. In FIFO mode, this error is associated with the character at the top of the FIFO.
2	BE	0x0	R	Break error. This bit is set to 1 if a break condition was detected, indicating that the received data input was held low for longer than a full-word transmission time (defined as start, data, parity and stop bits). This bit is cleared to 0b after a write to ECR. In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to HIGH (marking state), and the next valid start bit is received.
3	OE	0x0	R	Overrun error. This bit is set to 1 if data is received and the receive FIFO is already full. This is cleared to 0 by a write to ECR (data value is not important). The FIFO contents remain valid since no further data is written when the FIFO is full, only the content of the shift register are overwritten. The CPU or DMA must now read the data in order to empty the FIFO.
31:4	RESERVED	0x0	R	RESERVED

Table 80. UART - TIMEOUT register description: address offset UART_BASE_ADDR+0x0C

Bit	Field name	Reset	RW	Description
21:0	PERIOD	0x1FF	RW	Timeout period configuration. This bit field contains the timeout period for the UART timeout interrupt assertion. The receive timeout interrupt is asserted when the receive FIFO is not empty and no further data is received over a programmed timeout period. The duration before the timeout interrupt assert is calculated by the following formula: timeout = PERIOD / (OverSamplingFactor * BaudRate) Where OverSamplingFactor is: 16 if OVSFACT is 0 8 if OVSFACT is 1.
31:22	RESERVED	0x0	RW	RESERVED

Table 81. UART - FR register description: address offset UART_BASE_ADDR+0x18

Bit	Field name	Reset	RW	Description
0	CTS	0x0	R	Clear to send.
2:1	RESERVED	0x0	R	RESERVED
3	BUSY	0x0	R	UART Busy. If this bit is set to 1, the UART is busy transmitting data. This bit remains set until the complete byte, including all the stop bits, has been sent from the shift register. However, if the transmit section of the UART is disabled in the middle of a transmission, the BUSY bit gets cleared. This bit is set again once the transmit section is re-enabled to complete the remaining transmission. This bit is set as soon as the transmit FIFO becomes nonempty (regardless of whether the UART is enabled or not).
4	RXFE	0x1	R	Receive FIFO empty. If the FIFO is disabled (bit FEN = 0b), this bit is set when the receive holding register is empty. If the FIFO is enabled (FEN = 1b), the RXFE bit is set when the receive FIFO is empty.
5	TXFF	0x0	R	Transmit FIFO full. If the FIFO is disabled (bit FEN = 0b), this bit is set when the transmit holding register is full. If the FIFO is enabled (FEN = 1b), the TXFF bit is set when the transmit FIFO is full.
6	RXFF	0x0	R	Receive FIFO full. If the FIFO is disabled (bit FEN = 0b), this bit is set when the receive holding register is full. If the FIFO is enabled (FEN = 1b), the RXFF bit is set when the receive FIFO is full.

Bit	Field name	Reset	RW	Description
7	TXFE	0x1	R	Transmit FIFO empty. If the FIFO is disabled (bit FEN = 0b), this bit is set when the transmit holding register is empty. If the FIFO is enabled (FEN = 1b), the TXFE bit is set when the transmit FIFO is empty.
8	RESERVED	0x0	R	RESERVED
9	DCTS	0x1	R	Delta clear to send. This bit is set CTS changes since the last read of the FR register.
12:10	RESERVED	0x7	R	RESERVED
13	RTXDIS	0x0	R	Remote transmitter disabled (software flow control). This bit indicates an Xoff character was sent to the remote transmitter to stop it after the received FIFO has passed over its trigger limit. This bit is cleared when a Xon character is sent to the remote transmitter.
31:14	RESERVED	0x0	R	RESERVED

Table 82. UART - LCRH_RX register description: address offset UART_BASE_ADDR+0x1C

Bit	Field name	Reset	RW	Description
0	RESERVED	0x0	RW	RESERVED
1	PEN_RX	0x0	RW	RX parity enable: 0: Parity disabled. 1: Parity enabled.
2	EPS_RX	0x0	RW	RX even parity selection, when the parity is enabled. 0: Odd parity generation and checking is performed during reception, which check for an odd number of 1s in data and parity bits. 1: Even parity generation and checking is performed during reception, which check for an even number of 1s in data and parity bits.
3	STP2_RX	0x0	RW	RX two stop bits select. This bit enables the check for two stop bits being received: 0: 1 stop bit received. 1: 2 stop bits received.
4	FEN_RX	0x0	RW	RX enable FIFOs. This bit enables/disables the receive RX FIFO buffer: 0: RX FIFO is disabled (character mode). 1: RX FIFO is enabled.
6:5	WLEN_RX	0x0	RW	RX word length. This bit field indicates the number of data bits received in a frame as follows: 00b: 5 bits. 01b: 6 bits. 10b: 7 bits. 11b: 8 bits.
7	SPS_RX	0x0	RW	RX stick parity select: 0: stick parity is disabled. 1: when PEN_RX = 1b (parity enabled) and EPS_RX = 1b (even parity), the parity is checked as a 0. When PEN_RX = 1b and EPS_RX = 0b (odd parity), the parity bit is checked as a 1.
31:8	RESERVED	0x0	RW	RESERVED

Table 83. UART - IBRD register description: address offset UART_BASE_ADDR+0x24

Bit	Field name	Reset	RW	Description
15:0	DIVINT	0x0	RW	Baud rate integer. The baud rate divisor is calculated as follows: When OVSFACT = 0b in the CR register: Baud rate divisor = (frequency (UARTCLK)/(16*Baud rate)) When OVSFACT = 1b in CR register: Baud rate divisor = (frequency (UARTCLK)/(8*Baud rate)) where frequency (UARTCLK) is the UART reference clock frequency. The baud rate divisor comprises the integer value (DIVINT) and the fractional value (DIVFRAC). The contents of the IBRD and FBRD registers are not updated until transmission or reception of the current character has completed.
31:16	RESERVED	0x0	RW	RESERVED

Table 84. UART - FBRD register description: address offset UART_BASE_ADDR+0x28

Bit	Field name	Reset	RW	Description
5:0	DIVFRAC	0x0	RW	Baud rate fraction. Baud rate integer. The baud rate divisor is calculated as follows: When OVSFACT = 0b in the CR register: baud rate divisor = (frequency (UARTCLK)/(16*Baud rate)) When OVSFACT = 1b in CR register: baud rate divisor = (frequency (UARTCLK)/(8*Baud rate)) where frequency (UARTCLK) is the UART reference clock frequency. The baud rate divisor comprises the integer value (DIVINT) and the fractional value (DIVFRAC). The contents of the IBRD and FBRD registers are not updated until transmission or reception of the current character has completed.
31:6	RESERVED	0x0	RW	RESERVED

Table 85. UART - LCRH_TX register description: address offset UART_BASE_ADDR+0x2C

Bit	Field name	Reset	RW	Description
0	BRK	0x0	RW	Send break. This bit allows a continuous low-level to be forced on TX output, after completion of the current character. This bit must be asserted for at least one complete frame transmission time in order to generate a break condition. The transmit FIFO contents remain unaffected during a break condition. 0: Normal transmission. 1: Break condition transmission.
1	PEN_TX	0x0	RW	TX parity enable: 0: Parity disabled. 1: Parity enable.
2	EPS_TX	0x0	RW	TX even parity select. This bit selects the parity generation, when the parity is enabled (PEN_TX =1b). This bit has no effect when parity is disabled (PEN_TX = 0b). 0: Odd parity generation and checking is performed during transmission, which check for an odd number of 1s in data and parity bits. 1: Even parity generation and checking is performed during transmission, which check for an even number of 1s in data and parity bits.

Bit	Field name	Reset	RW	Description
3	STP2_TX	0x0	RW	TX two-stop bits select. This bit enables the check for two stop bits being received: 0: 1 stop bit received. 1: 2 stop bits received.
4	FEN_TX	0x0	RW	TX Enable FIFO. This bit enables/disables the transmit TX FIFO buffer: 0: TX FIFO is disabled (character mode), i.e. the TX FIFO becomes a 1-byte deep holding register. 1: TX FIFO is enabled.
6:5	WLEN_TX	0x0	RW	TX word length. This bit field indicates the number of data bits transmitted in a frame as follows: 00b: 5 bits. 01b: 6 bits. 10b: 7 bits. 11b: 8 bits.
7	SPS_TX	0x0	RW	TX stick parity check: 0: stick parity disable. 1: when PEN_TX = 1b (parity enabled) and EPS_TX = 1b (even parity), the parity is transmitted as a 0. When PEN_TX = 1b and EPS_TX = 0b (odd parity), the parity bit is transmitted as a 1.
31:8	RESERVED	0x0	RW	RESERVED

Table 86. UART - CR register description: address offset UART_BASE_ADDR+0x30

Bit	Field name	Reset	RW	Description
0	EN	0x0	RW	UART enable. This bit enables the UART. 0: UART is disabled. 1: UART is enabled. Data transmission and reception can occur. When the UART is disabled in the middle of transmission or reception, it completes the current character before stopping.
2:1	RESERVED	0x0	RW	RESERVED
3	OVSFACT	0x0	RW	UART oversampling factor. This bit enables the UART oversampling factor. If UARTCLK is 16 MHz thus max. baud-rate is 1 Mbaud when OVSFACT = 0b, and 2 Mbaud when OVSFACT = 1b. 0: UART it is 16 UARTCLK clock cycles. 1: UART it is 8 UARTCLK clock cycles.
7:4	RESERVED	0x0	RW	RESERVED
8	TXE	0x1	RW	Transmit enable. 0b: UART TX disabled. 1b: UART TX enabled.
9	RXE	0x1	RW	Receive enable. 0b: UART RX disabled. 1b: UART RX enabled.
10	RESERVED	0x0	RW	RESERVED
11	RTS	0x0	RW	Request to send. 0: RTS is high. 1: RTS is low.

Bit	Field name	Reset	RW	Description
13:12	RESERVED	0x0	RW	RESERVED
14	RTSEN	0x0	RW	RTS hardware flow control enable. 0b: RTS disabled. 1b: RTS enabled. Data is only requested when there is space in the receive FIFO for it to be received.
15	CTSEN	0x0	RW	CTS hardware flow control enable. 0b: CTS disabled. 1b: CTS enabled. Data is only transmitted when the CTS is asserted.
19:16	STA_B_DURATION	0x4	RW	START bit duration receiver state. These bits can be used to configure the START bit duration (in clock cycles) to get the bit sampled in the middle of the UART receiver. These bits can be used only when using high baud rates (IBRD = 1, FBRD ≥ 0 and OVSFACT = 1). Below the formula to calculate the START bit duration receiver state: STA_B_DURATION = Integer(Fuartclk/(2* BAUD RATE)) - 1 Example: when UARTCLK = 16 MHz and BAUD RATE = 2.0 Mbps then STA_B_DURATION = 4 - 1 = 3. STA_B_DURATION field should be configured with 4'b0011.
31:20	RESERVED	0x0	RW	RESERVED

Table 87. UART - IFLS register description: address offset UART_BASE_ADDR+0x34

Bit	Field name	Reset	RW	Description
2:0	TXIFLSEL	0x2	RW	Transmit interrupt FIFO level select. This bit field selects the trigger points for TX FIFO interrupt: 000b: Interrupt when FIFO ≥ 1/64 empty. 001b: Interrupt when FIFO ≥ 1/32 empty. 010b: Interrupt when FIFO ≥ 1/16 empty. 011b: Interrupt when FIFO ≥ 1/8 empty. 100b: Interrupt when FIFO ≥ 1/4 empty. 101b: Interrupt when FIFO ≥ 1/2 empty. 110b: Interrupt when FIFO ≥ 3/4 empty.
5:3	RXIFLSEL	0x2	RW	Receive interrupt FIFO level select. This bit field selects the trigger points for RX FIFO interrupt: 000b: Interrupt when FIFO ≥ 1/64 full. 001b: Interrupt when FIFO ≥ 1/32 full. 010b: Interrupt when FIFO ≥ 1/16 full. 011b: Interrupt when FIFO ≥ 1/8 full. 100b: Interrupt when FIFO ≥ 1/4 full. 101b: Interrupt when FIFO ≥ 1/2 full. 110b: Interrupt when FIFO ≥ 3/4 full.
31:6	RESERVED	0x0	RW	RESERVED

Table 88. **UART - IMSC register description: address offset UART_BASE_ADDR+0x38**

Bit	Field name	Reset	RW	Description
0	RESERVED	0x0	RW	RESERVED
1	CTSMIM	0x0	RW	Clear to send modem interrupt mask. On a read, the current mask for the CTSMIM interrupt is returned. 0: Clears the mask (interrupt is disabled). 1: Sets the mask (interrupt is enabled).
3:2	RESERVED	0x0	RW	RESERVED
4	RXIM	0x0	RW	Receive interrupt mask. On a read, the current mask for the RXIM interrupt is returned. 0: Clears the mask (interrupt is disabled). 1: Sets the mask (interrupt is enabled).
5	TXIM	0x0	RW	Transmit interrupt mask. On a read, the current mask for the TXIM interrupt is returned. 0: Clears the mask (interrupt is disabled). 1: Sets the mask (interrupt is enabled).
6	RTIM	0x0	RW	Receive timeout interrupt mask. On a read, the current mask for the RTIM interrupt is returned. 0: Clears the mask (interrupt is disabled). 1: Sets the mask (interrupt is enabled).
7	FEIM	0x0	RW	Framing error interrupt mask. On a read, the current mask for the FEIM interrupt is returned. 0: Clears the mask (interrupt is disabled). 1: Sets the mask (interrupt is enabled).
8	PEIM	0x0	RW	Parity error interrupt mask. On a read, the current mask for the PEIM interrupt is returned. 0: Clears the mask (interrupt is disabled). 1: Sets the mask (interrupt is enabled).
9	BEIM	0x0	RW	Break error interrupt mask. On a read, the current mask for the BEIM interrupt is returned. 0: Clears the mask (interrupt is disabled). 1: Sets the mask (interrupt is enabled).
10	OEIM	0x0	RW	Overrun error interrupt mask. On a read, the current mask for the OEIM interrupt is returned. 0: Clears the mask (interrupt is disabled). 1: Sets the mask (interrupt is enabled).
11	XOFFIM	0x0	RW	XOFF interrupt mask. On a read, the current mask for the XOFFIM interrupt is returned. 0: Clears the mask (interrupt is disabled). 1: Sets the mask (interrupt is enabled).
12	TXFEIM	0x0	RW	TX FIFO empty interrupt mask. On a read, the current mask for the TXFEIM interrupt is returned. 0: Clears the mask (interrupt is disabled). 1: Sets the mask (interrupt is enabled).
31:13	RESERVED	0x0	RW	RESERVED

Table 89. UART - RIS register description: address offset UART_BASE_ADDR+0x3C

Bit	Field name	Reset	RW	Description
0	RESERVED	0x0	R	RESERVED
1	CTSMIS	0x0	R	Clear to send interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
3:2	RESERVED	0x0	R	RESERVED
4	RXIS	0x0	R	Receive interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
5	TXIM	0x0	R	Transmit interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
6	RTIS	0x0	R	Receive timeout interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
7	FEIS	0x0	R	Framing error interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
8	PEIS	0x0	R	Parity error interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
9	BEIS	0x0	R	Break error interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
10	OEIS	0x0	R	Overrun error interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
11	XOFFIS	0x0	R	XOFF interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
12	TXFEIS	0x0	R	TX FIFO empty interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
31:13	RESERVED	0x0	R	RESERVED

Table 90. UART - MIS register description: address offset UART_BASE_ADDR+0x40

Bit	Field name	Reset	RW	Description
0	RESERVED	0x0	R	RESERVED
1	CTSMMIS	0x0	R	Clear to send masked interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.

Bit	Field name	Reset	RW	Description
3:2	RESERVED	0x0	R	RESERVED
4	RXMIS	0x0	R	Receive masked interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
5	TXMIS	0x0	R	Transmit masked interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
6	RTMIS	0x0	R	Receive timeout masked interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
7	FEMIS	0x0	R	Framing error masked interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
8	PEMIS	0x0	R	Parity error masked interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
9	BEMIS	0x0	R	Break error masked interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
10	OEMIS	0x0	R	Overrun error masked interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
11	XOFFMIS	0x0	R	XOFF interrupt masked status. 0: The interrupt is not pending. 1: The interrupt is pending.
12	TXFEMIS	0x0	R	TX FIFO empty masked interrupt status. 0: The interrupt is not pending. 1: The interrupt is pending.
31:13	RESERVED	0x0	R	RESERVED

Table 91. UART - ICR register description: address offset UART_BASE_ADDR+0x44

Bit	Field name	Reset	RW	Description
0	RESERVED	0x0	RW	RESERVED
1	CTSMIC	0x0	W	Clear to send modem interrupt clear. 0: No effect. 1: Clears the interrupt.
3:2	RESERVED	0x0	RW	RESERVED
4	RXIC	0x0	W	Receive interrupt clear. 0: No effect. 1: Clears the interrupt.
5	TXIC	0x0	W	Transmit interrupt clear.

Bit	Field name	Reset	RW	Description
				0: No effect. 1: Clears the interrupt.
6	RTIC	0x0	W	Receive timeout interrupt clear. 0: No effect. 1: Clears the interrupt.
7	FEIC	0x0	W	Framing error interrupt clear. 0: No effect. 1: Clears the interrupt.
8	PEIC	0x0	W	Parity error interrupt clear. 0: No effect. 1: Clears the interrupt.
9	BEIC	0x0	W	Break error interrupt clear. 0: No effect. 1: Clears the interrupt.
10	OEIC	0x0	W	Overrun error interrupt clear. 0: No effect. 1: Clears the interrupt.
11	XOFFIC	0x0	W	XOFF interrupt clear. 0: No effect. 1: Clears the interrupt.
12	TXFEIC	0x0	W	TX FIFO empty interrupt clear. 0: No effect. 1: Clears the interrupt.
31:13	RESERVED	0x0	RW	RESERVED

Table 92. UART - DMACR register description: address offset UART_BASE_ADDR+0x48

Bit	Field name	Reset	RW	Description
0	RXDMAE	0x0	RW	Receive DMA enable bit. 0: DMA mode is disabled for reception. 1: DMA mode is enabled for reception.
1	TXDMAE	0x0	RW	Transmit DMA enable bit. 0: DMA mode is disabled for transmit. 1: DMA mode is enabled for transmit.
2	RESERVED	0x0	RW	RESERVED
3	DMAONERR	0x0	RW	DMA on error. 0: UART error interrupt status has no impact in receive DMA mode. 1: DMA receive requests are disabled when the UART error interrupt is asserted.
31:4	RESERVED	0x0	RW	RW RESERVED

Table 93. UART - XFCR register description: address offset UART_BASE_ADDR+0x50

Bit	Field name	Reset	RW	Description
0	SFEN	0x0	RW	Software flow control enable. 0: Software flow control disable. 1: software flow control enable.
2:1	SFRMOD	0x0	RW	Software receive flow control mode: 00b: Receive flow control is disabled. 01b: Xon1, Xoff1 characters are used in receiving software flow control. 10b: Xon2, Xoff2 characters are used in receiving software flow control. 11b: Xon1 and Xon2, Xoff1 and Xoff2 characters are used in receiving software flow control.
4:3	SFTMOD	0x0	RW	Software transmit flow control mode: 00b: Transmit flow control is disabled. 01b: Xon1, Xoff1 characters are used in transmitting software flow control. 10b: Xon2, Xoff2 characters are used in transmitting software flow control. 11b: Xon1 and Xon2, Xoff1 and Xoff2 characters are used in transmitting software flow control.
5	XONANY	0x0	RW	Xon-any bit: 0: Incoming character must match Xon programmed value(s) to be a valid Xon. 1: Any incoming character is considered as a valid Xon.
6	SPECHAR	0x0	RW	Special character detection bit. 0: Special character detection disabled. 1: Special character detection enabled.
31:7	RESERVED	0x0	RW	RESERVED

Table 94. UART - XON1 register description: address offset UART_BASE_ADDR+0x54

Bit	Field name	Reset	RW	Description
7:0	XON1	0x0	RW	Value of Xon1 character used in the software flow control
31:8	RESERVED	0x0	RW	RW RESERVED

Table 95. UART - XON2 register description. Address offset UART_BASE_ADDR+0x58.

Bit	Field name	Reset	RW	Description
7:0	XON2	0x0	RW	Value of Xon2 character used in the software flow control.
31:8	RESERVED	0x0	RW	RESERVED

Table 96. UART - XOFF1 register description. Address offset UART_BASE_ADDR+0x5C.

Bit	Field name	Reset	RW	Description
7:0	XOFF1	0x0	RW	Value of Xoff1 character used in the software flow control.
31:8	RESERVED	0x0	RW	RESERVED

Table 97. **UART - XOFF2 register description. Address offset UART_BASE_ADDR+0x60.**

Bit	Field name	Reset	RW	Description
7:0	XOFF2	0x0	RW	Value of Xoff2 character used in the software flow control.
31:8	RESERVED	0x0	RW	RESERVED

Note: All RESERVED fields inside registers must always be written with their default values.

3.10 I²C

3.10.1 Introduction

The BlueNRG-2 integrates two I²C controllers in the QFN32 package (I2C2 and I2C1), and one in the WCSP34 package (I2C2). The I²C controller interface is designed to support the physical and data link layer according to I²C standard revision 3.0 and provides a low-cost interconnection between ICs.

Main features are:

- Up to 400 Kb/s in fast mode and up to 100 Kb/s in standard mode.
- Operating modes supported are master mode, slave mode, master/slave mode for multi-master system with bus arbitration.
- Programmable 7-bit or 10-bit addressing (also with combined formats).
- Programmable start byte procedure.
- 16-byte depth RX FIFO and 16-byte depth TX FIFO.
- Spike digital filtering on the SDA and SCL lines.
- Control timing constraint defined by the I²C standard.
- Support for direct memory access (DMA).

3.10.2 Functional description

Two wires, serial data (SDA) and serial clock (SCL) carry information between the devices connected to the bus. Each device has a unique address and can operate as either a transmitter or receiver, depending on the function of the device. A master is the device that initiates a data transfer on the bus and generates the clock signal. Any device addressed is considered at that time a slave. The I²C bus is a multi-master bus where more than one device is capable of controlling the bus. This means that more than one master could try to initiate a data transfer at the same time. The arbitration procedure relies on the wired-AND connection of all I²C interfaces to the I²C bus. If two or more masters try to put information onto the bus, the first to produce a ‘one’ when the other produces a ‘zero’ will lose the arbitration. The clock signals during arbitration are a synchronized combination of the clocks generated by the masters using the wired-AND connection to the SCL line. Generation of clock signals on the I²C bus is always the responsibility of master devices; each master generates its own clock signals when transferring data to the bus. Bus clock signals from a master can only be altered when they are stretched by a slow slave device holding down the clock line, or by another master when arbitration occurs.

Two modes:

- Standard mode with bit rate up to 100 Kb/s
- Fast mode with bit rate up to 400 Kb/s

3.10.2.1 I²C FIFO management

The transmit and receive paths are buffered with internal FIFO memory enabling up to 16 bytes to be stored independently in both transmit and receive modes. The FIFOs status can be checked using the I²C interrupts. There is a programmable threshold value for each FIFO. When the number of entries is greater for the receive FIFO or less for the transmit FIFO, an interrupt is set.

3.10.2.2 I²C clock rate calculation

To define the I²C clock rate generation there is one register to configure: BRCR. The clock rate can be calculated using this formula:

$$I2C_{clock} = \begin{cases} \dfrac{f_{I2C}}{(BRCNT2 * 2) + Foncycle} & in\ standard\ mode \\ \dfrac{f_{I2C}}{(BRCNT2 * 3) + Foncycle} & in\ fast\ mode \end{cases}$$

Where:

- f_{I2C} is the I²C peripheral clock, clocked on the system clock divided by 3.
- BRCNT2 is a field of the BRCR register.
- Foncycle depends on a programmable field of the CR register:
  - CR: FON = “00” → Filter the clock spike wide = 0 → Foncycle = 1
  - CR: FON = “01” → Filter the clock spike wide = 1 → Foncycle = 3
  - CR: FON = “10” → Filter the clock spike wide = 2 → Foncycle = 4
  - CR: FON = “11” → Filter the clock spike wide = 4 → Foncycle = 6

The minimum input clock frequency for the I²C is:

- 1.4 MHz if the I²C is in standard mode at 100 kHz.
- 7.2 MHz if the I²C is in fast mode at 400 kHz.

3.10.2.3 I²C configuration

Following a reset, the I²C logic is disabled and must be configured when in this state.

The control register (CR) and baud rate register (BRCR) need to be programmed to configure the following parameters of the peripheral:

- Master or slave.
- 7- or 10-bit addressing mode.
- Speed mode.
- Clock rate.

Note: If in slave mode, the SCR register has to be programmed with the selected slave address.

Then, if in master mode, the MCR register is used to define the transaction:

- Read or write.
- Slave addresses (7- or 10-bit) to communicate with.
- Addressing a 7- or 10-bit slave address.
- Stop condition, to generate a stop or restart condition at the end of the transaction (for consecutive transactions).
- Transaction length.

Note: For a master write, the data to transmit have to be written to the transmit FIFO in the I2C_TFR register. For a master read, when the master transaction is done, data are available in the receive FIFO in I2C_RFR.

3.10.2.4 DMA interface

The I²C controller includes a specific DMA interface. The following section describes the signals interface, data flow and programming model for the RX and the TX paths.

The DMA interfaces are separated for each path and two DMA request channels shall be used for a device.

In RX mode, a DMA transfer based on a single descriptor shall be used and the DMA RX channel must be programmed for a peripheral-to-memory transfer where the flow controller is the DMA. Each descriptor is related to a single I²C transaction (master read or write-to-slave operation) and no linked list item should be used. The transfer length is programmed on the DMA and the termination of the frame transfer is notified by the assertion of the related interrupt I2C_RISR:STD or I2C_RISR:MTD bits. In case of read-from-slave operation, on the DMA (master device) the transfer length shall be programmed according to the I2C_MCR:LENGTH register field. In case of write-to-slave operation, on the DMA (slave device) the maximum length (not the real length because it is unknown) shall be programmed.

In TX mode, a DMA transfer based on a single descriptor shall be used and the DMA TX channel must be programmed for a memory-to-peripheral transfer where the flow controller is the DMA. Each descriptor is related to a single I²C transaction (master write or read-from-slave operation) and no linked list item should be used. The transfer length is programmed on the DMA and the termination of the frame transfer is notified by the assertion of the related interrupt I2C_RISR:STD or I2C_RISR:MTD bits. In case of write-to-slave operation, on the DMA (master device) the transfer length shall be programmed according to the I2C_MCR:LENGTH register field. In case of read-from-slave operation, on the DMA (slave device) the maximum length (not the real length because it is unknown) shall be programmed.

3.10.3 I²C registers

I²C2 peripheral base address (I²C2_BASE_ADDR) 0x40A00000

I²C1 peripheral base address (I²C1_BASE_ADDR) 0x40B00000

Table 98. I²Cx registers

Address offset	Name	RW	Reset	Description
0x00	CR	RW	0x00000000	I²C control register. Refer to the detailed description below.
0x04	SCR	RW	0x000F0055	I²C slave control register. Refer to the detailed description below.
0x0C	MCR	RW	0x00000000	I²C master control register. Refer to the detailed description below.
0x10	TFR	RW	0x00000000	I²C transmit FIFO register. Refer to the detailed description below.
0x14	SR	R	0x00000000	I²C status register. Refer to the detailed description below.
0x18	RFR	R	0x00000000	I²C receive FIFO register. Refer to the detailed description below.
0x1C	TFTR	RW	0x00000000	I²C transmit FIFO threshold register. Refer to the detailed description below.
0x20	RFTR	RW	0x00000000	I²C receive FIFO threshold register. Refer to the detailed description below.
0x24	DMAR	RW	0x00000000	I²C DMA register. Refer to the detailed description below.
0x28	BRCR	RW	0x00000008	I²C baud-rate counter register. Refer to the detailed description below.
0x2C	IMSCR	RW	0x00000000	I²C interrupt mask set/clear register. Refer to the detailed description below.
0x30	RISR	R	0x00000013	I²C raw interrupt status register. Refer to the detailed description below.
0x34	MISR	R	0x00000000	I²C masked interrupt status register. Refer to the detailed description below.
0x38	ICR	RW	0x00000000	I²C interrupt clear register. Refer to the detailed description below.
0x4C	THDDAT	RW	0x00000014	I²C hold time data. Refer to the detailed description below.
0x50	THDSTA_FST_STD	RW	0x003F00E2	I²C hold time start condition F/S. Refer to the detailed description below.
0x54	RESERVED	RW	0x00000019	RESERVED
0x58	TSUSTA_FST_STD	RW	0x001D00E2	I²C setup time start condition F/S. Refer to the detailed description below.

Table 99. I2C - CR register description: address offset I2CX_BASE_ADDR+0x00

Bit	Field name	Reset	RW	Description
0	PE	0x0	RW	I²C enable disable: 0: I²C disable. 1: I²C enable. This bit when deasserted works as software reset for I²C peripheral.
2:1	OM	0x0	RW	Select the operating mode: 00b: Slave mode. The peripheral can only respond (transmit/receive) when addressed by a master device 01b: Master mode. The peripheral works in a multi-master system where itself cannot be addressed by another master device. It can only initiate a new transfer as master device. 10b: Master/slave mode. The peripheral works in a multi-master system where itself can be addressed by another master device, besides to initiate a transfer as master device.
3	SAM	0x0	RW	Slave addressing mode. SAM defines the slave addressing mode when the peripheral works in slave or master/slave mode. The received address is compared with the content of the register SCR. 0: 7-bit addressing mode. 1: 10-bit addressing mode.
5:4	SM	0x0	RW	Speed mode. SM defines the speed mode related to the serial bit rate: 0: Standard mode (up to 100 k/s). 1: Fast mode (up to 400 k/s).
6	SGCM	0x0	RW	Slave general call mode defines the operating mode of the slave controller when a general call is received. This setting does not affect the hardware general call that is always managed in transparent mode. 0: transparent mode, the slave receiver recognizes the general call but any action is taken by the hardware after the decoding of the message included in the Rx FIFO. 1: direct mode, the slave receiver recognizes the general call and executes directly (without software intervention) the related actions. Only the status code word is stored in the I2C_SR register for notification to the application.
7	FTX	0x0	RW	FTX flushes the transmit circuitry (FIFO, fsm). The configuration of the I²C node (register setting) is not affected by the flushing operation. The flushing operation is performed on modules working on different clock domains (system and I²C clocks) and needs several system clock cycles before being completed. Upon completion, the I²C node (internal logic) clears this bit. The application must not access the Tx FIFO during the flushing operation and should poll on this bit waiting for completion. 0: Flush operation is completed (I²C controller clears the bit). 1: Flush operation is started and in progress (set by application).
8	FRX	0x0	RW	FRX flushes the receive circuitry (FIFO, fsm).The configuration of the I²C node (register setting) is not affected by the flushing operation. The flushing operation is performed on modules working on different clock domains (system and I²C clocks) and needs several system clock cycles before to be completed. Upon completion, the I²C node (internal logic) clears this bit. The application must not access the Rx FIFO during the flushing operation and should poll on this bit waiting for the completion. 0: Flush operation is completed (I²C controller clears the bit). 1: Flush operation is started and in progress (set by application).
9	DMA_TX_EN	0x0	RW	Enables the DMA TX interface. 0: Idle state, the DMA TX interface is disabled.

Bit	Field name	Reset	RW	Description
				1: Run state, the DMA TX interface is enabled. On the completion of the DMA transfer, the DMA TX interface is automatically turned off clearing this bit when the end of transfer signal coming from the DMA is raised. DMA_TX_EN and DMA_RX_EN must not enabled at the same time.
10	DMA_RX_EN	0x0	RW	Enables the DMA RX interface. 0: Idle state, the DMA RX interface is disabled. 1: Run state, the DMA RX interface is enabled. On the completion of the DMA transfer, the DMA RX interface is automatically turned off clearing this bit when the end of transfer signal coming from the DMA is raised. DMA_TX_EN and DMA_RX_EN must not enabled at the same time.
12:11	RESERVED	0x0	RW	RESERVED
14:13	FON	0x0	RW	Filtering on sets the digital filters on the SDA, SCL line, according to the I²C bus requirements, when standard open-drain pads are used: 00b: No digital filters are inserted. 01b: Digital filters (filter 1 ck wide spikes) are inserted. 10b: Digital filters (filter 2 ck wide spikes) are inserted. 11b: Digital filters (filter 4 ck wide spikes) are inserted.
15	FS_1	0x0	RW	Force stop enable bit. When set to 1b, the STOP condition is generated. 0: Force stop disabled. 1: Enable force stop.
31:16	RESERVED	0x0	RW	RESERVED

Table 100. I2C - SCR register description: address offset I2CX_BASE_ADDR+0x04

Bit	Field name	Reset	RW	Description
6:0	SA7	0x55	RW	Slave address 7-bit. SA7 includes the slave address 7-bit or the LSB bits of the slave address 10-bit
9:7	ESA10	0x0	RW	Extended slave address 10-bit. ESA10 includes the extension (MSB bits) to the SA7 register field in case of slave addressing mode set to 10-bit
15:10	RESERVED	0x0	RW	RESERVED
31:16	SLSU	0xF	RW	Slave data setup time. SLSU defines the data setup time after SCL clock stretching in terms of i2c_clk cycles. Data setup time is actually equal to SLSU-1 clock cycles. The typical values for i2c_clk of 16 MHz are SLSU = 5 in standard mode and SLSU = 3 in fast modes.

Table 101. I2C2 - MCR register description: address offset I2CX_BASE_ADDR+0x0C

Bit	Field name	Reset	RW	Description
0	OP	0x0	RW	Operation 0: Indicates a master write operation. 1: Indicates a master read operation.
7:1	A7	0x0	RW	Address. Includes the 7-bit address or the LSB bits of the10-bit address used to initiate the current transaction
10:8	EA10	0x0	RW	Extended address. Includes the extension (MSB bits) of the field A7 used to initiate the current transaction

Bit	Field name	Reset	RW	Description
11	SB	0x0	RW	Start byte: 0: The start byte procedure is not applied to the current transaction. 1: The start byte procedure is prefixed to the current transaction.
13:12	AM	0x0	RW	Address type: 00b: The transaction is initiated by a general call command. In this case the fields OP, A7, EA10 are "don't care". 01b: The transaction is initiated by the 7-bit address included in the A7 field. 10b: The transaction is initiated by the 10-bit address included in the EA10 and A7 fields.
14	P	0x0	RW	Stop condition: 0: The current transaction is not terminated by a STOP condition. A repeated START condition is generated on the next operation which is required to avoid to stall the I²C line. 1: The current transaction is terminated by a STOP condition.
25:15	LENGTH	0x0	RW	Transaction length. Defines the length, in terms of the number of bytes to be transmitted (MW) or received (MR). In case of write operation, the payload is stored in the Tx FIFO. A transaction can be larger than the Tx FIFO size. In case of read operation the length refers to the number of bytes to be received before generating a not-acknowledge response. A transaction can be larger than the Rx FIFO size. The I²C clock line is stretched low until the data in Rx FIFO are consumed.
31:26	RESERVED	0x0	RW	RESERVED

Table 102. I2C - TFR register description: address offset I2CX_BASE_ADDR+0x10

Bit	Field name	Reset	RW	Description
7:0	TDATA	0x0	RW	Transmission data. TDATA contains the payload related to a master write or read-from-slave operation to be written in the Tx FIFO. TDATA(0) is the first LSB bit transmitted over the I²C line. In case of master write operation, the Tx FIFO shall be preloaded otherwise the I²C controller cannot start the operation until data are available. In case of read-from-slave operation, when the slave is addressed, the interrupt RISR:RFSR bit is asserted and the CPU shall download the data in the FIFO. If the FIFO is empty and the I²C master is still requiring data, a new request (RISR:RFSE interrupt bit) is asserted to require additional data to the CPU. The slave controller stretches the I²C clock line when no data are available for transmission. Since the Tx FIFO could contain some pending data related to the previous transfer (the transfer length may be unknown to the slave controller), the Tx FIFO is self-flushed before asserting the I2C_RISR:RFSR bit. Upon completion of the read-from-slave operation the interrupt bit I2C_RISR:STD is asserted and the related status of the operation is stored in the I2C_SR register. In CPU mode, the FIFO management is based on the assertion of the interrupt bit RISR:TXFNE, related to the nearly-empty threshold. In DMA mode, the single/burst requests are automatically executed based on the number of entries available in the TX FIFO and the related destination burst size programmed in the I2C_DMAR:DBSIZE_TX register field. The DMA requests are terminated at the end of the I²C read operation (notacknowledge received by the master) by a dummy last single/burst request.
31:8	RESERVED	0x0	RW	RESERVED

Table 103. I2C - SR register description: address offset I2CX_BASE_ADDR+0x14

Bit	Field name	Reset	RW	Description
1:0	OP	0x0	R	Operation: 00b: MW: master write operation. 01b: MR: master read operation. 10b: WTS: write-to-slave operation. 11b: RFS: read-from-slave operation.
3:2	STATUS	0x0	R	Controller status. Valid for the operations MW, MR, WTS RFS: 00b: NOP: No operation is in progress. 01b: ON_GOING: An operation is ongoing. 10b: OK: The operation (OP field) has been completed successfully. 11b: ABORT: The operation (OP field) has been aborted due to the occurrence of the event described in the CAUSE field.
6:4	CAUSE	0x0	R	Abort cause. This field is valid only when the STATUS field contains the ABORT tag. Others: RESERVED. 000b: NACK_ADDR: The master receives a not-acknowledge after the transmission of the address. Valid for the operation MW, MR. 001b: NACK_DATA: The master receives a not-acknowledge during the data phase of a MW operation. Valid for the operation MW. 011b: ARB_LOST: The master loses the arbitration during a MW or MR operation. Valid for the operation MW, MR. 100b: BERR_START: Slave restarts, a START Condition occurs while the byte transfer is not terminated. 101b: BERR_STOP: Slave Reset, a STOP Condition while the byte transfer is not terminated. 110b: OVFL: The slave receives a frame related to the WTS operation longer than the maximum size = 2047 bytes. In this case the slave device returns a NACK to complete the data transfer. Valid for WTS operation
8:7	TYPE	0x0	R	Receive type. Valid only for the operation WTS: 00b: FRAME: The slave has received a normal frame. 01b: GCALL: The slave has received a general call. If the it I2C_CR:SGCM is set to 1, the general call is directly executed without software intervention and only the control code word is reported in FIFO (LENGTH =0). 10b: HW_GCALL: The slave has received a hardware general call.
19:9	LENGTH	0x0	R	Transfer length. For an MR, WTS operation the LENGTH field defines the actual size of the subsequent payload, in terms of number of bytes. For an MW, RFS operation the LENGTH field defines the actual number of bytes transferred by the master/slave device. For a WTS operation if the transfer length exceeds 2047 bytes, the operation is stopped by the slave returning a NACK handshake and the flag OVFL is set. For an RFS operation if the transfer length exceeds 2047 bytes, the operation continues normally but the LENGTH field is reset to 0.
28:20	RESERVED	0x0	R	RESERVED
29	DUALF	0x0	R	Dual flag (slave mode): 0: Received address matched with slave address (SA7). 1: Received address matched with dual slave address (DSA7). Cleared by hardware after a stop condition or repeated Start condition, bus error or when PE=0.
31:30	RESERVED	0x0	R	RESERVED

Table 104. I2C - RFR register description: address offset I2CX_BASE_ADDR+0x18

Bit	Field name	Reset	RW	Description
7:0	RDATA	0x0	R	Receive data. RDATA contains the received payload, related to a master read or write-to-slave operation, to be read from the Rx FIFO. The RDATA(0) is the first LSB bit received over the I2C line. In case the FIFO is full, the I2C controller stretches automatically the I2C clock line until a new entry is available. For a write-to-slave operation, when the slave is addressed, the interrupt I2C_RISR:WTSR bit is asserted for notification to the CPU. In CPU mode the FIFO management shall be based on the assertion of the interrupt bit I2C_RISR:RXFNF, related to the nearly-full threshold. In DMA mode, the single requests are automatically executed based on the number of entries contained in the Rx FIFO.
31:8	RESERVED	0x0	R	RESERVED

Table 105. I2C - TFTR register description: address offset I2CX_BASE_ADDR+0x1C

Bit	Field name	Reset	RW	Description
9:0	THRESH_TX	0x0	RW	Threshold TX, contains the threshold value, in terms of number of bytes, of the Tx FIFO. When the number of entries of the Tx FIFO is less or equal than the threshold value, the interrupt bit I2C_RISR:TXFNE is set in order to request the loading of data to the application.
31:10	RESERVED	0x0	RW	RESERVED

Table 106. I2C - RFTR register description: address offset I2CX_BASE_ADDR+0x20

Bit	Field name	Reset	RW	Description
9:0	THRESH_RX	0x0	RW	Threshold RX, contains the threshold value, in terms of number of bytes, of the Rx FIFO. When the number of entries of the RX FIFO is greater than or equal to the threshold value, the interrupt bit RISR:RXFNF is set in order to request the download of received data to the application. The application shall download the received data based on the threshold. (RISR:RXFNF).
31:10	RESERVED	0x0	RW	RESERVED

Table 107. I2C - DMAR register description: address offset I2CX_BASE_ADDR+0x24

Bit	Field name	Reset	RW	Description
7:0	RESERVED	0x0	RW	RESERVED
10:8	DBSIZE_TX	0x0	RW	Destination burst size. This register field is valid only if the BURST_TX bit is set to '1'. If burst size is smaller than the transaction length, only single request are generated.
11	BURST_TX	0x0	RW	Defines the type of DMA request generated by the DMA TX interface. 0: Single request mode. Transfers a single data (one byte) in the TX FIFO. 1: Burst request mode. Transfers a programmed burst of data according to DBSIZE_TX field. When the burst mode is programmed, the DMA transfer can be completed by one or more single requests as required.
31:12	RESERVED	0x0	RW	RESERVED

Table 108. I2C - BRCR register description: address offset I2CX_BASE_ADDR+0x28

Bit	Field name	Reset	RW	Description
15:0	BRCNT	0x8	RW	Baud rate counter. BRCNT defines the counter value used to set up the I2C baud rate in standard and fast mode, when the peripheral is operating in master mode.
31:16	RESERVED	0x0	RW	RESERVED

Table 109. I2C - IMSCR register description: address offset I2CX_BASE_ADDR+0x2C

Bit	Field name	Reset	RW	Description
0	TXFEM	0x0	RW	TX FIFO empty mask. TXFEM enables the interrupt bit TXFE: 0: TXFE interrupt is disabled. 1: TXFE interrupt is enabled.
1	TXFNEM	0x0	RW	TX FIFO nearly empty mask. TXFNEM enables the interrupt bit TXFNE: 0: TXFNE interrupt is disabled. 1: TXFNE interrupt is enabled.
2	TXFFM	0x0	RW	TX FIFO full mask. TXFFM enables the interrupt bit TXFF: 0: TXFF interrupt is disabled. 1: TXFF interrupt is enabled.
3	TXFOVRM	0x0	RW	TX FIFO overrun mask. TXOVRM enables the interrupt bit TXOVR: 0: TXOVR interrupt is disabled. 1: TXOVR interrupt is enabled.
4	RXFEM	0x0	RW	RX FIFO empty mask. RXFEM enables the interrupt bit RXFE: 0: RXFE interrupt is disabled. 1: RXFE interrupt is enabled.
5	RXFNFM	0x0	RW	RX FIFO nearly full mask. RXNFM enables the interrupt bit RXNF: 0: RXNF interrupt is disabled. 1: RXNF interrupt is enabled
6	RXFFM	0x0	RW	RX FIFO full mask. RXFFM enables the interrupt bit RXFF: 0: RXFF interrupt is disabled. 1: RXFF interrupt is enabled.
15:7	RESERVED	0x0	RW	RESERVED
16	RFSRM	0x0	RW	Read-from-slave request mask. RFSRM enables the interrupt bit RFSR: 0: RFSR interrupt is disabled. 1: RFSR interrupt is enabled.
17	RFSEM	0x0	RW	Read-from-slave empty mask. RFSEM enables the interrupt bit RFSE: 0: RFSE interrupt is disabled. 1: RFSE interrupt is enabled.
18	WTSRM	0x0	RW	Write-to-slave request mask. WTSRM enables the interrupt bit WTSR: 0: WTSR interrupt is disabled. 1: WTSR interrupt is enabled.
19	MTDM	0x0	RW	Master transaction done mask. MTDM enables the interrupt bit MTD: 0: MTD interrupt is disabled. 1: MTD interrupt is enabled.

Bit	Field name	Reset	RW	Description
20	STDM	0x0	RW	Slave transaction done mask. STDM enables the interrupt bit STD: 0: STDM interrupt is disabled. 1: STDM interrupt is enabled.
23:21	RESERVED	0x0	RW	RESERVED
24	MALM	0x0	RW	Master arbitration lost mask. MALM enables the interrupt bit MAL: 0: MAL interrupt is disabled. 1: MAL interrupt is enabled.
25	BERRM	0x0	RW	Bus error mask. BERRM enables the interrupt bit BERR: 0: BERR interrupt is disabled. 1: BERR interrupt is enabled.
27:26	RESERVED	0x0	RW	RESERVED
28	MTDWSM	0x0	RW	Master transaction done without stop mask. MTDWSM enables the interrupt bit MTDWS: 0: MTDWS interrupt is disabled. 1: MTDWS interrupt is enabled.
31:29	RESERVED	0x0	RW	RESERVED

Table 110. I2C - RISR register description: address offset I2CX_BASE_ADDR+0x30

Bit	Field name	Reset	RW	Description
0	TXFE	0x1	R	TX FIFO empty. TXFE is set when TX FIFO is empty. This bit is self-cleared by writing in TX FIFO. 0: TX FIFO is not empty. 1: TX FIFO is empty.
1	TXFNE	0x1	R	TX FIFO nearly empty. TXFNE is set when the number of entries in TX FIFO is less than or equal to the threshold value programmed in the I2C_TFTR:THRESHOLD_TX register. It is self-cleared when the threshold level is over the programmed threshold. 0: Number of entries in TX FIFO greater than the TFTR:THRESHOLD_TX register. 1: Number of entries in TX FIFO less than or equal to the TFTR:THRESHOLD_TX register.
2	TXFF	0x0	R	TX FIFO full. TXFF is set when a full condition occurs in TX FIFO. This bit is self-cleared when the TX FIFO is not full: 0: TX FIFO is not full. 1: TX FIFO is full.
3	TXFOVR	0x0	R	TX FIFO overrun. TXFOVR is set when a write operation in TX FIFO is performed and TX FIFO is full. The application must avoid an overflow condition by a proper data flow control. Anyway in case of overrun, the application shall flush the transmitter (CR:FTX bit to set) because the TX FIFO content is corrupted (at least one word has been lost in FIFO). This interrupt is cleared by setting the related bit of the ICR register: 0: No overrun condition occurred in TX FIFO. 1: Overrun condition occurred in TX FIFO.
4	RXFE	0x1	R	RX FIFO empty. RXFE is set when the RX FIFO is empty. This bit is self-cleared when the slave RX FIFO is not empty: 0: RX FIFO is not empty.

Bit	Field name	Reset	RW	Description
				1: RX FIFO is empty.
5	RXFNF	0x0	R	RX FIFO nearly full. RXFNF is set when the number of entries in RX FIFO is greater than or equal to the threshold value programmed in the RFTR:THRESHOLD_RX register. Its self-cleared when the threshold level is under the programmed threshold: 0: Number of entries in the RX FIFO less than the RFTR:THRESHOLD_RX register. 1: Number of entries in the RX FIFO greater than or equal to the RFTR:THRESHOLD_RX register.
6	RXFF	0x0	R	RX FIFO full. RXFF is set when a full condition occurs in RX FIFO. This bit is self-cleared when the data are read from the RX FIFO. 0: RX FIFO is not full. 1: RX FIFO is full.
15:7	RESERVED	0x0	R	RESERVED
16	RFSR	0x0	R	Read-from-slave request. RFSR is set when a read-from-slave "Slavetransmitter" request is received (I²C slave is addressed) from the I²C line. On the assertion of this interrupt the TX FIFO is flushed (pending data are cleared) and the CPU shall put the data in TX FIFO. This bit is self-cleared by writing data in FIFO. In case the FIFO is empty before the completion of the read operation, the RISR:RFSE interrupt bit is set.This interrupt is cleared by setting the related bit of the ICR register. 0: Read-from-slave request has been served. 1: Read-from-slave request is pending.
17	RFSE	0x0	R	Read-from-slave empty. RFSE is set when a read-from-slave operation is in progress and TX FIFO is empty. On the assertion of this interrupt, the CPU shall download in TX FIFO the data required for the slave operation. This bit is self-cleared by writing in TX FIFO. At the end of the read-from-slave operation this bit is cleared although the TX FIFO is empty. 0: TX FIFO is not empty. 1: TX FIFO is empty with the read-from-slave operation in progress.
18	WTSR	0x0	R	Write-to-slave request. WTSR is set when a write-to-slave operation is received (I2C slave is addressed) from the I2C line. This notification can be used by the application to program the DMA descriptor when required. This interrupt is cleared by setting the related bit of the ICR register: 0: No write-to-slave request pending. 1: Write-to-slave request is pending.
19	MTD	0x0	R	Master transaction done. MTD is set when a master operation (master write or master read) has been executed after a stop condition. The application shall read the related transaction status (SR register), the pending data in the RX FIFO (only for a master read operation) and clear this interrupt (transaction acknowledgment). A subsequent master operation can be issued (writing the MCR register) after the clearing of this interrupt. A subsequent slave operation will be notified (RISR:WTSR and RISR:RFSR interrupt bits assertion) after clearing this interrupt, meanwhile the I²C clock line is stretched low. This interrupt is cleared by setting the related bit of the ICR register. 0: Master transaction acknowledged. 1: Master transaction done (ready for acknowledgment).

Bit	Field name	Reset	RW	Description
20	STD	0x0	R	Slave transaction done. STD is set when a slave operation (write-to-slave or read-from-slave) has been executed. The application shall read the related transaction status (SR register), the pending data in the RX FIFO (only for a write-to-slave operation) and clear this interrupt (transaction acknowledgment). A subsequent slave operation will be notified (RISR:WTSR and RISR:RFSR interrupt bits assertion) after clearing this interrupt, meanwhile the I2C clock line will be stretched low. A subsequent master operation can be issued (by writing the MCR register) after clearing this interrupt. This interrupt is cleared by setting the related bit of the ICR register: 0: Slave transaction acknowledged. 1: Slave transaction done (ready for acknowledgment).
23:21	RESERVED	0x0	R	RESERVED
24	MAL	0x0	R	Master arbitration lost. MAL is set when the master loses the arbitration. The status code word in the SR contains a specific error tag (CAUSE field) for this error condition. A collision occurs when 2 stations transmit simultaneously 2 opposite values on the serial line. The station that is pulling up the line, identifies the collision reading a 0 value on the sda_in signal, stops the transmission, leaves the bus and waits for the idle state (STOP condition received) on the bus line before retrying the same transaction. The station which transmits the first unique zero wins the bus arbitration. This interrupt is cleared by setting the related bit of the ICR register. 0: No master arbitration lost. 1: Master arbitration lost.
25	BERR	0x0	R	Bus Error. BERR is set when an unexpected Start/Stop condition occurs during a transaction. The related actions are different, depending on the type of operation in progress.The status code word in the SR contains a specific error tag (CAUSE field) for this error condition. This interrupt is cleared by setting the related bit of the ICR register. 0: No bus error detection. 1: Bus error detection.
27:26	RESERVED	0x0	R	RESERVED
28	MTDWS	0x0	R	Master transaction done without stop. MTDWS is set when a master operation (write or read) has been executed and a stop (MCR:P field) is not programmed. The application shall read the related transaction status (SR register), the pending data in the RX FIFO (only for a master read operation) and clear this interrupt (transaction acknowledgment). A subsequent master operation can be issued (by writing the MCR register) after clearing this interrupt. A subsequent slave operation will be notified (RISR:WTSR and RISR:RFSR interrupt bits assertion) after clearing this interrupt, meanwhile the I²C clock line will be stretched low. This interrupt is cleared by setting the related bit of the ICR register: 0: Master transaction acknowledged. 1: Master transaction done (ready for acknowledgment) and stop is not applied into the I²C bus.
31:29	RESERVED	0x0	R	RESERVED

Table 111. I2C - MISR register description: address offset I2CX_BASE_ADDR+0x34

Bit	Field name	Reset	RW	Description
0	TXFEMIS	0x0	R	TX FIFO empty masked interrupt status. 0: TX FIFO is not empty. 1: TX FIFO is empty.

Bit	Field name	Reset	RW	Description
1	TXFNEMIS	0x0	R	TX FIFO nearly empty masked interrupt status. 0: Number of entries in TX FIFO greater than the TFTR:THRESHOLD_TX register. 1: Number of entries in TX FIFO less than or equal to the TFTR:THRESHOLD_TX register.
2	TXFFMIS	0x0	R	Tx FIFO full masked interrupt status. 0: TX FIFO is not full. 1: TX FIFO is full.
3	TXFOVRMIS	0x0	R	Tx FIFO overrun masked interrupt status. 0: No overrun condition occurred in TX FIFO. 1: Overrun condition occurred in TX FIFO.
4	RXFEMIS	0x0	R	RX FIFO empty masked interrupt status. 0: RX FIFO is not empty. 1: RX FIFO is empty.
5	RXFNFMIS	0x0	R	RX FIFO nearly full masked interrupt status. 0: Number of entries in the RX FIFO less than the RFTR:THRESHOLD_RX register. 1: Number of entries in the RX FIFO greater than or equal to the RFTR:THRESHOLD_RX register.
6	RXFFMIS	0x0	R	RX FIFO full masked interrupt status. 0: RX FIFO is not full. 1: RX FIFO is full.
15:7	RESERVED	0x0	R	RESERVED
16	RFSRMIS	0x0	R	Read-from-slave request masked interrupt status. 0: Read-from-slave request has been served. 1: Read-from-slave request is pending.
17	RFSEMIS	0x0	R	Read-from-slave empty masked interrupt status. 0: TX FIFO is not empty. 1: TX FIFO is empty with the read-from-slave operation in progress.
18	WTSRMIS	0x0	R	Write-to-slave request masked interrupt status. 0: No write-to-slave request pending. 1: Write-to-slave request is pending.
19	MTDMIS	0x0	R	Master transaction done masked interrupt status. 0: Master transaction acknowledged. 1: Master transaction done (ready for acknowledgment).
20	STDMIS	0x0	R	Slave transaction done masked interrupt status. 0: Slave transaction acknowledged. 1: Slave transaction done (ready for acknowledgment).
23:21	RESERVED	0x0	R	RESERVED
24	MALMIS	0x0	R	Master arbitration lost masked interrupt status. 0: No master arbitration lost. 1: Master arbitration lost.
25	BERRMIS	0x0	R	Bus error masked interrupt status. 0: No bus error detection.

Bit	Field name	Reset	RW	Description
				1: Bus error detection.
27:26	RESERVED	0x0	R	RESERVED
28	MTDWSMIS	0x0	R	Master transaction done without stop masked interrupt status. 0: Master transaction acknowledged. 1: Master transaction done (ready for acknowledgment) and stop is not applied into the I²C bus.
31:29	RESERVED	0x0	R	RESERVED

Table 112. I2C - ICR register description: address offset I2CX_BASE_ADDR+0x38

Bit	Field name	Reset	RW	Description
2:0	RESERVED	0x0	RW	RESERVED
3	TXFOVRIC	0x0	RW	Tx FIFO overrun interrupt clear. 0: Has no effect. 1: Clears interrupt pending.
15:4	RESERVED	0x0	RW	RESERVED
16	RFSRIC	0x0	RW	Read-from-Slave request interrupt clear. 0: Has no effect. 1: Clears interrupt pending.
17	RFSEIC	0x0	RW	Read-from-slave empty interrupt clear. 0: Has no effect. 1: Clears interrupt pending.
18	WTSRIC	0x0	RW	Write-to-slave request interrupt clear. 0: Has no effect. 1: Clears interrupt pending.
19	MTDIC	0x0	RW	Master transaction done interrupt clear. 0: Has no effect. 1: Clears interrupt pending.
20	STDIC	0x0	RW	Slave transaction done interrupt clear. 0: Has no effect. 1: Clears interrupt pending.
23:21	RESERVED	0x0	RW	RESERVED
24	MALIC	0x0	RW	Master arbitration lost interrupt clear. 0: Has no effect. 1: Clears interrupt pending.
25	BERRIC	0x0	RW	Bus error interrupt clear. 0: Has no effect. 1: Clears interrupt pending.
27:26	RESERVED	0x0	RW	RESERVED
28	MTDWSIC	0x0	RW	Master transaction done without stop interrupt clear. 0: Has no effect. 1: Clears interrupt pending.
31:29	RESERVED	0x0	RW	RESERVED

Table 113. I2C - THDDAT register description: address offset I2CX_BASE_ADDR+0x4C

Bit	Field name	Reset	RW	Description
8:0	THDDAT	0x14	RW	Hold time data value. In master or slave mode, when the I2C controller detects a falling edge in the SCL line, the counter, which is loaded by the THDDAT, is launched. Once the THDDAT value is reached, the data is transferred.
31:9	RESERVED	0x0	RW	RESERVED

Table 114. I2C - THDSTA_FST_STD register description: address offset I2CX_BASE_ADDR+0x50

Bit	Field name	Reset	RW	Description
8:0	THDSTA_STD	0xE2	RW	Hold time start condition value for standard mode. When the start condition is asserted, the decimeter loads the value of THDSTA_STD for standard mode, once the THDSTA_STD value is reached, the SCL line asserts low.
15:9	RESERVED	0x0	RW	RESERVED
24:16	THDSTA_FST	0x3F	RW	Hold time start condition value for fast mode. When the start condition is asserted, the decimeter loads the value of THDSTA_FST for fast mode, once the THDSTA_FST value is reached, the SCL line assert slow.
31:25	RESERVED	0x0	RW	RESERVED

Table 115. I2C - TSUSTA_FST_STD register description: address offset I2CX_BASE_ADDR+0x58

Bit	Field name	Reset	RW	Description
8:0	TSUSTA_STD	0xE2	RW	Setup time start condition value for standard mode. After a non-stop on the SCL line the decimeter loads the value of TSUSTA_STD according to standard mode. Once the counter is expired, the start condition is generated.
15:9	RESERVED	0x0	RW	RESERVED
24:16	TSUSTA_FST	0x1D	RW	Set-up time start condition value for fast mode. After a non-stop on the SCL line the decimeter loads the value of TSUSTA_FST according to fast mode. Once the counter is expired the start condition is generated.
31:25	RESERVED	0x0	RW	RESERVED

Note: All RESERVED fields inside registers must always be written with their default value.

3.11 Flash controller

3.11.1 Flash controller introduction

The BlueNRG-2 integrates a Flash controller to interface the embedded Flash memory.

Main features are:

- Sector erase and mass erase
- 256 kbyte Flash memory: 128 pages of 8 rows with 64 words each
- Flash programming
- Mass read
- Enable readout protection
- 32-bit read access
- 32-bit write access in single write and 4x32-bit in burst write (reduce programming time by 2)

3.11.2 Flash controller functional description

The BlueNRG-2 embeds up to 256 KB (65536 x 32-bit) of internal Flash memory. A Flash interface implements instruction access and data access based on the AHB protocol. It implements the logic necessary to carry out the Flash memory operations (Program/Erase) controlled through the Flash registers.

Writing to Flash only allows clearing bits from ‘1’ to ‘0’. This means any write from ‘0’ to ‘1’ implies erasing before performing a write.

Flash memory is composed of 128 pages containing 8 rows of 64 words (128 x 8 x 64 = 65536 words). Each word is 32-bit = 4 bytes long which means 256 kB of Flash.

The address inside the ADDRESS register is built as follows:

ADDRESS[15:0] = XADR[9:0] & YADR[5:0] with:

- XADR[9:3] = page address
- XADR[2:0] = row address
- YADR[5:0] = word address (one word = four bytes)

Note: One specific address can be written only twice between two erase actions even if each writing only clears bits.

Note: The Flash data retention is greater than 10 years at 85 °C. The Flash is rated to have a Flash cycling of 10.000 write/erase cycles.

3.11.2.1 *Reading Flash memory*

To read one single word of the flash, just read it as if RAM memory: read the desired flash address and get read data on the bus.

3.11.2.2 *Erasing Flash*

The Flash controller allows erasing one page or the full main Flash.

ERASE sequence (erase one page):

1. Write the page address to be erased by writing in the ADDRESS register the following value:
   a. ADDRESS[15:9] = XADR[9:3] = page address to erase
   b. ADDRESS[8:0] = 9’b0 (row and word addresses at zero).
2. Write the ERASE command (0x11) in the COMMAND register.
3. Wait for the CMDSTART flag in the IRQSTAT register (polling mode or interrupt mode) indicating command is taken into account and under execution.
4. Clear the CMDSTART flag by writing CMDSTART to ‘1’ in the IRQSTAT register.
5. Wait for the CMDDONE flag in the IRQSTAT register (polling mode or interrupt mode) indicating that the command is completed.
6. Clear the CMDDONE flag by writing CMDDONE to ‘1’ in the IRQSTAT register.

After this command, the erased page contains only bits set to ‘1’.

MASSERASE sequence (erase completely main flash):

1. Write the MASSERASE command (0x22) in the COMMAND register.
2. Wait for the CMDSTART flag in the IRQSTAT register (polling mode or interrupt mode) indicating that the command has been taken into account and is under execution.
3. Clear the CMDSTART flag by writing CMDSTART to ‘1’ in the IRQSTAT register.
4. Wait for the CMDDONE flag in the IRQSTAT register (polling mode or interrupt mode) indicating that the command is completed.
5. Clear the CMDDONE flag by writing CMDDONE to ‘1’ in the IRQSTAT register.

After this command, the full main Flash contains only bits set to ‘1’.

3.11.2.3 *Write function examples*

The Flash Controller allows writing one word (WRITE), up to 4 words (BURSTWRITE) or the full main Flash memory (with a single fixed word).

Note: As a write can only program to ‘0’ on bits already set to ‘1’, it is necessary to erase the page and request that the bits be set to ‘1’ (instead of ‘0’) in order to re-write to ‘0’.

WRITE sequence:

1. Indicate the location to write by filling the ADDRESS register with the targeted address (page, row and word number)

2. Write the value to program in the DATA0 register.
3. Write the WRITE command (0x33) in the COMMAND register.
4. Wait for the CMDSTART flag in the IRQSTAT register (polling mode or interrupt mode) indicating that the command has been taken into account and is under execution.
5. Clear the CMDSTART flag by writing CMDSTART to '1' in the IRQSTAT register.
6. Wait for the CMDDONE flag in the IRQSTAT register (polling mode or interrupt mode) indicating that the command is completed.
7. Clear the CMDDONE flag by writing CMDDONE to '1' in the IRQSTAT register.

BURSTWRITE sequence:

1. Indicate the location to write by filling the ADDRESS register with the targeted address of the first data to write (page, row and word number). DATA0 will be written and ADDRESS, DATA1 at ADDRESS+1 and so on. (Write the values to program in the DATA0-3 registers. To write less than four words, write 0xFFFFFFFF in the unused DATA1-3 registers.)
2. Write the BURSTWRITE command (0xCC) in the COMMAND register.
3. Wait for the CMDSTART flag in the IRQSTAT register (polling mode or interrupt mode) indicating that the command has been taken into account and is under execution.
4. Clear the CMDSTART flag by writing CMDSTART to '1' in the IRQSTAT register.
5. Wait for the CMDDONE flag in the IRQSTAT register (polling mode or interrupt mode) indicating that the command is completed.
6. Clear the CMDDONE flag by writing CMDDONE to '1' in the IRQSTAT register.

3.11.2.4 *Flash readout protection*

It is possible to protect flash memory from unwanted access while in debug mode, this is normally used to avoid copy or reverse engineering of a deployed application in the market.

If the readout protection mechanism is activated, as soon as Cortex-M0 is halted, any access to Flash memory will return a fixed 0x0 value and generate an AHB error if a debugger tries to read it.

Note that RAM memory debug accesses are also disabled by this lock protection.

Enabling readout protection:

1. Program a secret 64 bit sequence in the last two word address of the user flash. The secret 64 bit sequence can be anything different from 0xFFFFFFFF, 0xFFFFFFFF.

Disable readout protection:

1. Perform a mass erase of the user flash

3.11.2.5 *Flash command list*

The valid command values list for the COMMAND register is reported in the table below.

Table 116. Flash commands

Command name	Description	Value
ERASE	Erase page defined by register ADDRESS	0x11
MASSERASE	Mass erase (Flash is completely erased)	0x22
WRITE	Program one location (defined by registers DATA and ADDRESS)	0x33
BURSTWRITE	Burst write operation	0xCC

3.11.2.6 *Flash interface timing characteristics*

Table 117. Flash interface timing

Description	Max.	Unit
Page erase time	21.5	ms
Mass erase time	21.5	ms
Program time WRITE	44	µs

Description	Max.	Unit
Program time BURSTWRITE 1 word (1)	44	µs
Program time BURSTWRITE 2 words(1)	65	µs
Program time BURSTWRITE 3 words(1)	86	µs
Program time BURSTWRITE 4 words(1)	107	µs

1. *Burst write procedure skips the 0xFFFF_FFFF word*

3.11.3 Flash controller registers

Flash controller peripheral base address (FLASH_BASE_ADDR) 0x40100000.

Table 118. FLASH controller registers

Address offset	Name	RW	Reset	Description
0x00	COMMAND	RW	0x00000000	Commands for the module
0x04	CONFIG	RW	0x00000000	Configure the wrapper
0x08	IRQSTAT	RW	0x00000000	Flash status interrupt (masked). Refer to the detailed description below.
0x0C	IRQMASK	RW	0x0000003F	Mask for interrupts. Refer to the detailed description below.
0x10	IRQRAW	RW	0x00000000	Status interrupts (unmasked). Refer to the detailed description below.
0x14	SIZE	R	0x0000FFFF	Indicates the size of the Flash
0x18	ADDRESS	RW	0x00000000	Address for programming Flash, will auto-increment
0x24	RESERVED	R	0xFFFFFFFF	RESERVED
0x28	RESERVED	RW	0x0B061515	RESERVED
0x2C	RESERVED	RW	0x0B156506	RESERVED
0x30	RESERVED	RW	0x00000011	RESERVED
0x40	DATA0	RW	0xFFFFFFFF	Program cycle data
0x44	DATA1	RW	0xFFFFFFFF	Program cycle data
0x48	DATA2	RW	0xFFFFFFFF	Program cycle data
0x4C	DATA3	RW	0xFFFFFFFF	Program cycle data

Table 119. FLASH – COMMAND register description: address offset FLASH_BASE_ADDR+0x00

Bit	Field name	Reset	RW	Description
31:0	COMMAND	0x00000000	RW	Command for the module.

Table 120. FLASH – CONFIG register description: address offset FLASH_BASE_ADDR+0x04

Bit	Field name	Reset	RW	Description
0	RESERVED	0	RW	RESERVED
1	REMAP	0	RW	Remaps the interrupt vector table in RAM
2	RESERVED	0	RW	RESERVED
3	PREMAP	0	RW	Remaps the interrupt vector table in FLASH
31:4	RESERVED	0	RW	RESERVED

Table 121. **FLASH - IRQSTAT register description: address offset FLASH_BASE_ADDR+0x08**

Bit	Field name	Reset	RW	Description
0	CMDDONE	0x0	RW	Command is done. 1: clear the interrupt pending bit.
1	CMDSTART	0x0	RW	Command is started.1: clear the interrupt pending bit.
2	CMDERR	0x0	RW	Command written while BUSY. 1: clear the interrupt pending bit.
3	ILLCMD	0x0	RW	Illegal command written. 1: clear the interrupt pending bit.
4	READOK	0x0	RW	Mass read was OK. 1: clear the interrupt pending bit.
5	FLNREADY	0x0	RW	Flash not ready (sleep). 1: clear the interrupt pending bit.
31:6	RESERVED	0x0	RW	RESERVED

Table 122. **FLASH - IRQMASK register description: address offset FLASH_BASE_ADDR+0x0C**

Bit	Field name	Reset	RW	Description
0	CMDDONE	0x1	RW	Command is done.
1	CMDSTART	0x1	RW	Command is started.
2	CMDERR	0x1	RW	Command written while BUSY
3	ILLCMD	0x1	RW	Illegal command written
4	READOK	0x1	RW	Mass read was OK.
5	FLNREADY	0x1	RW	Flash not ready (sleep).
31:6	RESERVED	0x0	RW	RESERVED

Table 123. **FLASH - IRQRAW register description: address offset FLASH_BASE_ADDR+0x10**

Bit	Field name	Reset	RW	Description
0	CMDDONE	0x0	RW	Command is done.
1	CMDSTART	0x0	RW	Command is started.
2	CMDERR	0x0	RW	Command written while BUSY
3	ILLCMD	0x0	RW	Illegal command written
4	READOK	0x0	RW	Mass read was OK.
5	FLNREADY	0x0	RW	Flash not ready (sleep).
31:6	RESERVED	0x0	RW	RESERVED

Table 124. **FLASH – SIZE register description: address offset FLASH_BASE_ADDR+0x14**

Bit	Field name	Reset	RW	Description
15:0	SIZE	0xFFFF	R	Indicates the size of the flash. 0xFFFF: 256 kB of flash
31:16	RESERVED	0x0000	R	RESERVED

Table 125. **FLASH – ADDRESS register description: address offset FLASH_BASE_ADDR+0x18**

Bit	Field name	Reset	RW	Description
31:0	ADDRESS	0x00000000	RW	Address for programming flash, auto-increment.

Table 126. **FLASH – DATA0 register description: address offset FLASH_BASE_ADDR+0x40**

Bit	Field name	Reset	RW	Description
31:0	DATA0	0xFFFFFFFF	RW	Program cycle data.

Table 127. **FLASH – DATA1 register description: address offset FLASH_BASE_ADDR+0x44**

Bit	Field name	Reset	RW	Description
31:0	DATA1	0xFFFFFFFF	RW	Program cycle data.

Table 128. **FLASH – DATA2 register description: address offset FLASH_BASE_ADDR+0x48**

Bit	Field name	Reset	RW	Description
31:0	DATA2	0xFFFFFFFF	RW	Program cycle data.

Table 129. **FLASH – DATA3 register description: address offset FLASH_BASE_ADDR+0x4C**

Bit	Field name	Reset	RW	Description
31:0	DATA3	0xFFFFFFFF	RW	Program cycle data.

Note: All RESERVED fields inside registers must always be written with their default value.

3.12 GPIO

3.12.1 Introduction

The BlueNRG-2 offers 14 GPIOs (WCSP34 package), 15 GPIOs (QFN32 package) or 26 GPIOs (QFN48 package).

The programmable I/O pin can be configured for operating as:

- Programmable GPIOs
- Peripheral input or output line of standard communication interfaces
- PDM processor data/clock
- 2 PWM sources (PWM0 and PWM1 independently configurable) and 4 PWM output pins (IO2, IO3, IO4 and IO5).
- 5 wakeup sources from standby and sleep mode
- Each I/O can generate an interrupt independently to the selected mode. Interrupts are generated depending on a level or edge

3.12.2 Functional description

The table below shows the GPIO configuration table where each IO pin is associated with related functions.

Table 130. **IO functional map**

Pin name(1)	GPIO mode "000"		Serial1 mode '001"		Serial0 mode ‘100’		Serial2 mode '101'	
	Type	Signal	Type	Signal	Type	Signal	Type	Signal
IO0	I/O	GPIO 0	I	UART_CTS	I/O	SPI_CLK	O	CPUCLK(2)
IO1	I/O	GPIO 1	O	UART_RTS	I/O	SPI_CS1	I	PDM_DATA
IO2	I/O	GPIO 2	O	PWM0	O	SPI_OUT	O	PDM_CLK

Pin name(1)	GPIO mode "000"		Serial1 mode '001"		Serial0 mode '100'		Serial2 mode '101'	
	Type	Signal	Type	Signal	Type	Signal	Type	Signal
IO3	I/O	GPIO 3	O	PWM1	I	SPI_IN	-	-
IO4	I/O	GPIO 4	I	UART_RXD	I/O	I2C2_CLK	O	PWM0
IO5	I/O	GPIO 5	O	UART_TXD	I/O	I2C2_DAT	O	PWM1
IO6	I/O	GPIO 6	O	UART_RTS	I/O	I2C2_CLK	I	PDM_DATA
IO7	I/O	GPIO 7	I	UART_CTS	I/O	I2C2_DAT	O	PDM_CLK
IO8	I/O	GPIO 8	O	UART_TXD	I/O	SPI_CLK	I	PDM_DATA
IO9	I/O	GPIO 9	I	SWCLK	I	SPI_IN	O	XO16/32M(3)
IO10	I/O	GPIO 10	I	SWDIO	O	SPI_OUT	O	CLK_32K
IO11	I/O	GPIO 11	I	UART_RXD	I/O	SPI_CS1	O	CLK_32K
IO12	OD	GPI 12 (4)		-	I/O	I2C1_CLK	-	-
IO13	OD	GPI 13(4)	I	UART_CTS	I/O	I2C1_DAT	-	-
IO14	I/O	GPIO 14	I/O	I2C1_CLK	I/O	SPI_CLK	-	-
IO15	I/O	GPIO 15	I/O	I2C1_DAT	I/O	SPI_CS1	-	-
IO16	I/O	GPIO 16	O	PWM0	I	SPI_IN	-	-
IO17	I/O	GPIO 17	O	PWM1	O	SPI_OUT	-	-
IO18	I/O	GPIO 18	O	SPI_CS2	O	UART_RTS	-	-
IO19	I/O	GPIO 19	O	SPI_CS3	I	UART_CTS	-	-
IO20	I/O	GPIO 20	I	UART_CTS	O	SPI_CS2	-	-
IO21	I/O	GPIO 21	O	PWM1	I/O	SPI_CS1	-	-
IO22	I/O	GPIO 22	O	PWM0	O	SPI_CS3	-	-
IO23	I/O	GPIO 23	O	UART_TXD	O	SPI_OUT	O	PDM_CLK
IO24	I/O	GPIO 24	I	UART_RXD	I	SPI_IN	I	PDM_DATA
IO25	I/O	GPIO 25	O	UART_RTS	I/O	SPI_CLK	O	PDM_CLK

1. *All the I^2C pins require external pull-up*
2. *The bitfield SYSCLK_IO_EN of CLOCK_EN must be set in order to enable the CPUCLK signal.*
3. *The bitfield XO_IO_EN of CLOCK_EN must be set in order to enable the CPUCLK signal.*
4. *IO12 and IO13 can only be General Purpose Input pins (not output).*

3.12.2.1 *GPIO interrupts*

Each IO in GPIO mode can be used as interrupt source from external signal. The trigger event is both edge and level sensitive according to configuration. All the configuration are reported in table below.

Table 131. GPIO interrupt modes

Configuration	Interrupt mode				
	Falling edge	Rising edge	Both edges	Low level	High level
IOIS	0	0	0	1	1
IOIBE	0	0	1	NA	NA
IOIEV	0	1	NA	0	1

The interrupt is enabled by writing 1 in the MIS register, in the position with same number of the IO desired. Once the interrupt occurs, it can be cleared by writing 1 in the IC register. All the interrupts drive a single interrupt signal of the NVIC.

Each time the status of an IO matches its interrupt setting expressed by the registers IS, IBE and IEV, then the correspondent bit in the RIS register is set. Before enabling the correspondent interrupt mask (register IE), it is recommended to clear the RIS register by writing the correspondent bit of the register IC.

3.12.2.2 *GPIO characteristics*

By default all the GPIO pins are configured as input with related pull-up or pull-down enabled in order to have a specified voltage level.

Table 132. Pin characteristics

Name	Type	Buffer strength(1)	Pull-up / pull-down availability	Default state after reset	State during low power modes
IO0	I/O	High / very high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO1	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO2	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO3	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO4	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO5	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO6	I/O	High / very high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO7	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO8	I/O	Low /high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO9	I/O	High / very High drive	Pull- up	Serial1 mode (SWCLK), pull-up	GPIO mode, with configurable pull
IO10	I/O	Low / high drive	Pull- up	Serial1 mode (SWDIO), pull-up	GPIO mode, with configurable pull
IO11	I/O	Low / high drive	Pull-up	GPIO input mode, pull-up	GPIO mode, with configurable pull
IO12	I	10 mA(2)	No pull	GPIO input mode	GPIO input mode
IO13	I	10 mA (2)	No pull	GPIO input mode	GPIO input mode
IO14	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO15	I/O	Low /high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO16	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO17	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO18	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state

Name	Type	Buffer strength(1)	Pull-up / pull-down availability	Default state after reset	State during low power modes
IO19	I/O	Low /high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO20	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO21	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO22	I/O	Low /high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO23	I/O	Low /high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO24	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state
IO25	I/O	Low / high drive	Pull-down	GPIO input mode, pull-down	High impedance state

1. See Table 213. Digital I/O specifications
2. Drive strength for I^2C bus

Note: If the user needs pull-up or pull-down capability for driving the GPIO line, this must be done through an external resistor.

All the IOs are in high impedance under reset. In low power modes (sleep and standby) the I/Os are configured as follows:

-IO0 to IO8 and IO14 to IO25 are in high impedance state.

-IO9, IO10, IO11 can be configured as input or output, with or without internal pull by using the SLEEPIO_PE register. Section 3.4.1.5.3 GPIO special settings in low power modes.

-IO12 , IO13 are in input state no pull.

When the device is in low power mode, in order to avoid leakage, IO12 and IO13 need to be driven to logic level high or low. If these pins are no used, it is recommended to pull them down through a pull-down resistor of 10 kΩ.

The GPIO9, GPIO10 and GPIO11 can be configured to be used as output pin during the low power modes such as standby and sleep. For more information see Section 3.4.1.5.3 GPIO special settings in low power modes.

3.12.3 GPIO registers

GPIO peripheral base address (GPIO_BASE_ADDR) 0x40000000.

Table 133. GPIO registers

Address offset	Name	RW	Reset	Description
0x00	DATA	RW	0x00000000	IO0 to IO25 data value. Writing to a bit will drive the written value on the corresponding IO when it is configured in GPIO mode and the output direction. Reading a bit indicates the pin value
0x04	OEN	RW	0x00000000	GPIO output enable register (1 bit per GPIO) 0: input mode 1: output mode
0x08	PE	RW	0x03FFFFFF	Pull enable (1 bit per IO) 0: pull disabled 1: pull enabled
0x0C	DS	RW	0x00000000	IO driver strength (1 bit per IO). 0: low drive

Address offset	Name	RW	Reset	Description
				1: high drive
0x10	IS	RW	0x00000000	Interrupt sense register (1 bit per IO) 0: edge detection 1: level detection
0x14	IBE	RW	0x00000000	Interrupt edge register (1 bit per IO). 0: single edge 1: both edges
0x18	IEV	RW	0x00000000	Interrupt event register (1 bit per IO) 0: falling edge or low level 1: rising edge or high level
0x1C	IE	RW	0x00000000	Interrupt mask register (1 bit per IO) 0: Interrupt disabled 1: Interrupt enabled
0x20	RIS	R	0x00000000	Raw interrupt status register (1 bit per IO)
0x24	MIS	R	0x00000000	Masked interrupt status register (1 bit per IO)
0x28	IC	W	0x00000000	Interrupt clear register (1 bit per IO) 0: no effect. 1: clear interrupt
0x2C	MODE0	RW	0x00000000	Select mode for IO0 to IO7. 000b: GPIO mode 001b: Serial1 mode 100b: Serial0 mode 101b: Serial2 mode Refer to the detailed description below.
0x30	MODE1	RW	0x00000110	Select mode for IO8 to IO15. 000b: GPIO mode. 001b: Serial1 mode. 100b: Serial0 mode. 101b: Serial2 mode. Refer to the detailed description below.
0x34	MODE2	RW	0x00000000	Select mode for IO16 to IO23. 000b: GPIO mode. 001b: Serial1 mode. 100b: Serial0 mode. 101b: Serial2 mode. Refer to the detailed description below.
0x38	MODE3	RW	0x00000000	Select mode for IO24 to IO25. 000b: GPIO mode. 001b: Serial1 mode. 100b: Serial0 mode. 101b: Serial2 mode. Refer to the detailed description below.

Address offset	Name	RW	Reset	Description
0x3C	DATS	RW	0x00000000	Set some bits of DATA when in GPIO mode without affecting the others (1 bit per IO). 0: no effect. 1: set at 1 the bit
0x40	DATC	RW	0x00000000	Clear some bits of DATA when in GPIO mode without affecting the others (1 bit per IO) 0: no effect. 1: clear at 0 the bit
0x44	MFTX	RW	0x00000000	Select the IO to be used as capture input for the MFTX timers. Refer to the detailed description below.

Table 134. GPIO – DATA register description: address offset GPIO_BASE_ADDR+0x00

Bit	Field name	Reset	RW	Description
31:0	DATA	0x00000000	RW	IO0 to IO25 data value. Writing to a bit will drive the written value on the corresponding IO when it is configured in GPIO mode and the output direction. Reading a bit indicates the pin value.

Table 135. GPIO – OEN register description: address offset GPIO_BASE_ADDR+0x04

Bit	Field name	Reset	RW	Description
31:0	OEN	0x00000000	RW	GPIO output enable register (1 bit per GPIO). • 0: Input mode • 1: Output mode

Table 136. GPIO – PE register description: address offset GPIO_BASE_ADDR+0x08

Bit	Field name	Reset	RW	Description
31:0	PE	0x03FFFFFF	RW	Pull enable (1 bit per IO). • 0: Pull disabled. • 1: Pull enabled.

Table 137. GPIO – DS register description: address offset GPIO_BASE_ADDR+0x0C

Bit	Field name	Reset	RW	Description
31:0	DS	0x00000000	RW	IO driver strength (1 bit per IO). • 0: low drive • 1: high drive

Table 138. GPIO – IS register description: address offset GPIO_BASE_ADDR+0x10

Bit	Field name	Reset	RW	Description
31:0	IS	0x00000000	RW	Interrupt sense register (1 bit per IO). • 0: Edge detection.

Bit	Field name	Reset	RW	Description
				• 1: Level detection.

Table 139. GPIO – IBE register description: address offset GPIO_BASE_ADDR+0x14

Bit	Field name	Reset	RW	Description
31:0	IBE	0x00000000	RW	Interrupt edge register (1 bit per IO). • 0: Single edge • 1: Both edges

Table 140. GPIO – IEV register description: address offset GPIO_BASE_ADDR+0x18

Bit	Field name	Reset	RW	Description
31:0	IEV	0x00000000	RW	Interrupt event register (1 bit per IO). • 0: Falling edge or low level. • 1: Rising edge or high level.

Table 141. GPIO – IE register description: address offset GPIO_BASE_ADDR+0x1C

Bit	Field name	Reset	RW	Description
31:0	IE	0x00000000	RW	Interrupt mask register (1 bit per IO). The register MIS is the result of the AND logic between the register IE and the register RIS. • 0: Interrupt mask disable • 1: Interrupt mask enable

Table 142. GPIO – RIS register description: address offset GPIO_BASE_ADDR+0x20

Bit	Field name	Reset	RW	Description
31:0	RIS	0x00000000	R	Raw interrupt status register (1 bit per IO). Each time the status of an IO matches its interrupt setting expressed by the registers IS, IBE and IEV, then the correspondent bit in the RIS register is set. Before enabling the correspondent interrupt mask (register IE), it is recommended to clear the RIS register by writing the correspondent bit of the register IC

Table 143. GPIO – MIS register description: address offset GPIO_BASE_ADDR+0x24

Bit	Field name	Reset	RW	Description
31:0	MIS	0x00000000	R	Masked interrupt status register (1 bit per IO).

Table 144. **GPIO – IC register description: address offset GPIO_BASE_ADDR+0x28**

Bit	Field name	Reset	RW	Description
31:0	IC	0x00000000	W	Interrupt clear register (1 bit per IO). This register clears the bit set in the RIS register. If the same cleared bit is set in the IE register, the clear acts also in the MIS register as consequence. • 0: No effect • 1: Clear interrupt

Table 145. **GPIO - MODE0 register description: address offset GPIO_BASE_ADDR+0x2C**

Bit	Field name	Reset	RW	Description
2:0	IO0	0x0	RW	IO0 mode
3	RESERVED	0x0	RW	RESERVED
6:4	IO1	0x0	RW	IO1 mode
7	RESERVED	0x0	RW	RESERVED
10:8	IO2	0x0	RW	IO2 mode
11	RESERVED	0x0	RW	RESERVED
14:12	IO3	0x0	RW	IO3 mode
15	RESERVED	0x0	RW	RESERVED
18:16	IO4	0x0	RW	IO4 mode
19	RESERVED	0x0	RW	RESERVED
22:20	IO5	0x0	RW	IO5 mode
23	RESERVED	0x0	RW	RESERVED
26:24	IO6	0x0	RW	IO6 mode
27	RESERVED	0x0	RW	RESERVED
30:28	IO7	0x0	RW	IO7 mode
31	RESERVED	0x0	RW	RESERVED

Table 146. **GPIO – MODE1 register description: address offset GPIO_BASE_ADDR+0x30**

Bit	Field name	Reset	RW	Description
2:0	IO8	0x0	RW	IO8 mode
3	RESERVED	0x0	RW	RESERVED
6:4	IO9	0x1	RW	IO9 mode
7	RESERVED	0x0	RW	RESERVED
10:8	IO10	0x1	RW	IO10 mode
11	RESERVED	0x0	RW	RESERVED
14:12	IO11	0x0	RW	IO11 mode
15	RESERVED	0x0	RW	RESERVED
18:16	IO12	0x0	RW	IO12 mode
19	RESERVED	0x0	RW	RESERVED
22:20	IO13	0x0	RW	IO13 mode

Bit	Field name	Reset	RW	Description
23	RESERVED	0x0	RW	RESERVED
26:24	IO14	0x0	RW	IO14 mode
27	RESERVED	0x0	RW	RESERVED
30:28	IO15	0x0	RW	IO15 mode
31	RESERVED	0x0	RW	RESERVED

Table 147. GPIO – MODE2 register description: address offset GPIO_BASE_ADDR+0x34

Bit	Field name	Reset	RW	Description
2:0	IO16	0x0	RW	IO16 mode
3	RESERVED	0x0	RW	RESERVED
6:4	IO17	0x0	RW	IO17 mode
7	RESERVED	0x0	RW	RESERVED
10:8	IO18	0x0	RW	IO18 mode
11	RESERVED	0x0	RW	RESERVED
14:12	IO19	0x0	RW	IO19 mode
15	RESERVED	0x0	RW	RESERVED
18:16	IO20	0x0	RW	IO20 mode
19	RESERVED	0x0	RW	RESERVED
22:20	IO21	0x0	RW	IO21 mode
23	RESERVED	0x0	RW	RESERVED
26:24	IO22	0x0	RW	IO22 mode
27	RESERVED	0x0	RW	RESERVED
30:28	IO23	0x0	RW	IO23 mode
31	RESERVED	0x0	RW	RESERVED

Table 148. GPIO – MODE3 register description: address offset GPIO_BASE_ADDR+0x38

Bit	Field name	Reset	RW	Description
2:0	IO24	0x0	RW	IO24 mode
3	RESERVED	0x0	RW	RESERVED
6:4	IO25	0x0	RW	IO25 mode
31:7	RESERVED	0x0	RW	RESERVED

Table 149. GPIO – DATS register description: address offset GPIO_BASE_ADDR+0x3C

Bit	Field name	Reset	RW	Description
31:0	DATS	0x00000000	RW	Set some bits of DATA when in GPIO mode without affecting the others (1 bit per IO). • 0: No effect • 1: Set at 1 the bit

Table 150. GPIO – DATC register description: address offset GPIO_BASE_ADDR+0x40

Bit	Field name	Reset	RW	Description
31:0	DATC	0x00000000	RW	Clear some bits of DATA when in GPIO mode without affecting the others (1 bit per IO). • 0: No effect • 1: Clear at 0 the bit

Table 151. GPIO - MFTX register description: address offset GPIO_BASE_ADDR+0x44

Bit	Field name	Reset	RW	Description
7:0	MFT1_TIMER_A	0x0	RW	Selects which IO must be used as input pin TnA for the MFT1 peripheral. Mode 2 and mode 4 only. • 0x00: IO0 • 0x01: IO1 • 0x02: IO2 • ... • 0x19: IO25
15:8	MFT1_TIMER_B	0x0	RW	Selects which IO must be used as input pin TnB for the MFT1 peripheral. Mode 2 and mode 4 only. • 0x00: IO0 • 0x01: IO1 • 0x02: IO2 • ... • 0x19: IO25
23:16	MFT2_TIMER_A	0x0	RW	Selects which IO must be used as input pin TnA for the MFT2 peripheral. Mode 2 and mode 4 only. • 0x00: IO0 • 0x01: IO1 • 0x02: IO2 • ... • 0x19: IO25
31:24	MFT2_TIMER_B	0x0	RW	Selects which IO must be used as input pin TnB for the MFT2 peripheral. Mode 2 and mode 4 only. • 0x00: IO0 • 0x01: IO1 • 0x02: IO2 • ... • 0x19: IO25

Note: All RESERVED fields inside registers must always be written with their default value.

3.13 MFT

3.13.1 MFT introduction

The BlueNRG-1 integrates two multi functions timers (MFT).

Main features are:

- Two 16-bit programmable timer/counters.
- Two 16-bit reload/capture registers that function either as reload registers or as capture registers, depending on the mode of operation.

- An 8-bit fully programmable clock prescaler.
- Clock source selectors that allow each counter to operate in:
  - Pulse-accumulate mode
  - External-event mode
  - System clock with configurable prescaler
- Two I/O pins (TnA and TnB) with programmable edge detection that operate as:
  - Capture and preset inputs
  - External event (clock) inputs
  - PWM outputs
- Two interrupts, one for each counter, that can be triggered by a:
  - Timer underflow
  - Timer reload
  - Input capture
  - Pulse train for generation of single or multiple PWM pulses.

3.13.2 MFT functional description

The MFT can be configured in five different modes. At each mode is associated a particular function for the two timers both for counter and reload registers as reported in Table 152. MFT modes.

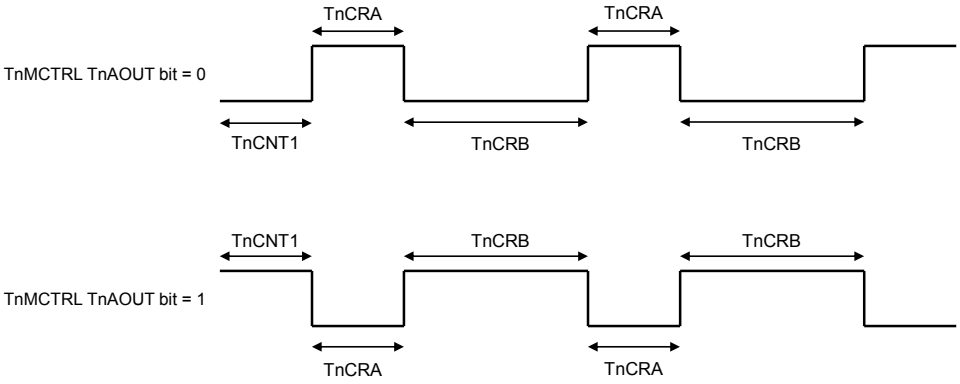
Table 152. MFT modes

Mode	Description	Timer Counter 1 (TnCNT1)	Reload / capture A (TnCRA)	Reload / Capture B (TnCRB)	Timer Counter 2 (TnCNT2)
1	PWM and system timer or external event counter	Counter for PWM	Auto reload A = PWM time 1	Auto reload B = PWM time 2	System time or external event
1a	PWM pulse train	Counter for PWM	Auto reload A = PWM time 1	Auto reload B= PWM time 2	Pulse counter
2	Dual-input capture and system timer	Capture A and B time base	Capture Timer/Counter 1 value upon TnA event	Capture Timer/Counter 1 value upon TnB event	System timer
3	Dual independent timer	Time base for first timer	Reload register for Timer/ Counter 1	Reload register for Timer/ Counter 2	Time base for second timer
4	Single-input capture and single timer	Time base for first timer	Reload register for Timer/ Counter 1	Capture Timer/Counter 1 value upon TnB event	Capture B time base

3.13.2.1 MFT mode 1: processor-independent PWM

In mode 1, the Timer/Counter 1 (register TnCNT1) can be used to generate a PWM signal on an output pin of the device. In this mode, the PWM output can emulate a clock signal with customized duty-cycle. Indeed, the TnCNT1 register is alternatively reloaded with the values in TnCRA and TnCRB registers. The initial value of the PWM output signal can be selected by software to be either high or low thanks to the TnAOUT bit in TnMCTRL register. This bit impact, which reloads value, is used for high level and low level of the PWM signal as shown in Figure 16. PWM signal on TnA pin below.

Figure 15. PWM signal on TnA pin



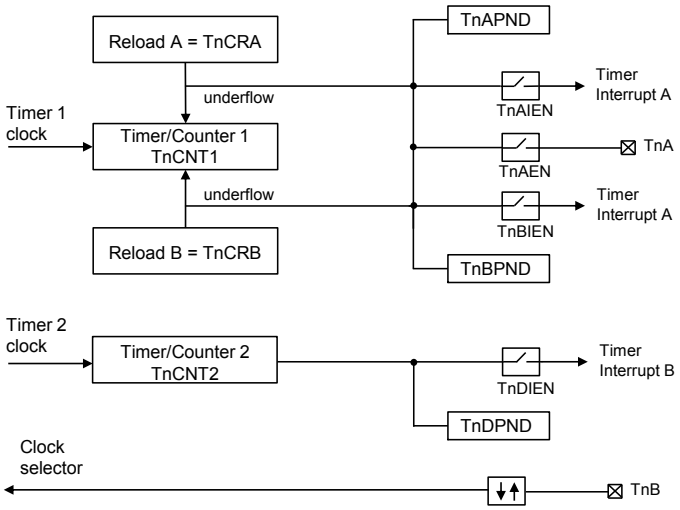
When started, the first time the Timer/Counter 1 starts with preprogrammed value in the register TnCNT1 and count down at the clock rate selected by the Timer/Counter 1 clock selector (TnCKC register). When an underflow occurs, the TnCNT1 register is reloaded alternatively with TnCRA and TnCRB registers (in this order) and counting proceeds downward from the loaded value.

Any time the counter is stopped by choosing "no-clock" in the TnCKC register, it obtains its first reload value after it has been started again from the TnCRA register. Each time the counter is stopped and then restarted, it obtains its first reload value from the TnCRA register. This is true whether the timer is restarted upon reset, after entering mode 1 from another mode, or after stopping and restarting the clock with the TnCKC register.

The mode 1 is selected in the TnMDSEL field of TnMCTRL register

In figure below the block diagram related to the MFT mode 1.

Figure 16. MFT mode 1 block diagram



In this mode, the timer toggles the TnA output upon underflow, which is connected to PWMx pin of the device.

This generates a clock signal on TnA with the width and duty cycle controlled by the values stored in the TnCRA and TnCRB registers.

This is a "processor-independent" PWM clock because once the timer is set up, no more interaction is required from the software and the CPU in order to generate a continuous PWM signal. Refer to Section 3.13.2.6 Timer IO functions for additional details.

The timer can generate separate interrupts upon reload from TnCRA and TnCRB. The TnAPND or TnBPND flags, which are set by the hardware upon occurrence of a timer reload, indicate which interrupt has occurred. Refer to Section 3.13.2.8 Timer interrupts for detailed information.

In this mode, the Timer/Counter 2 can be used as a simple system timer or as an external-event counter.

The Timer/Counter 2 counts down with the clock selected by Timer/Counter 2 clock selector (TnCKC register), and can be configured to generate an interrupt upon underflow if enabled by the TnDIEN bit.

The interrupts can be enabled or disabled by software.

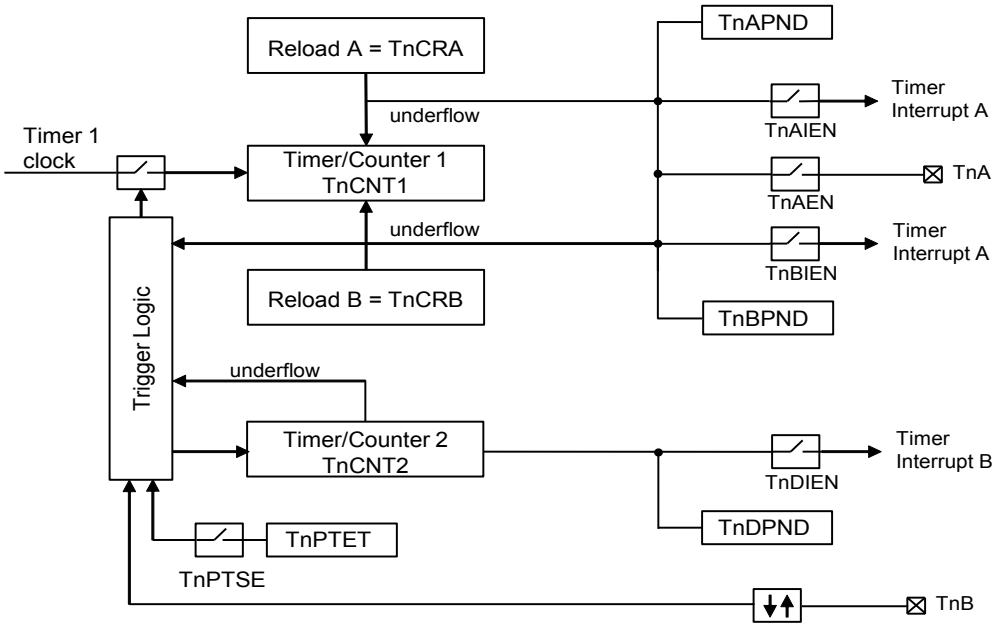
3.13.2.2 *MFT mode 1a: PWM pulse-train mode*

The mode 1a is used to output a PWM signal thanks to the Timer/Counter 1 as for mode 1, but only in a time window defined by the Timer/Counter 2 (TnCNT2). Indeed, the Timer/Counter 2 is used to specify the number of pulses to output on the TnA pin.

The mode 1a corresponds to the mode 1 selected in the TnMDSEL field of TnMCTRL register with in addition the TnPTEN bit set always in the TnMCTRL register. In mode 1a, the Timer/Counter 1 (TnCNT1 register) alternatively is reloaded by TnCRA and TnCRB registers after starting from the value in the TnCNT1 register as for the mode 1 and toggles the TnA output connected to PWMx GPIO each time an underflow occurs. In parallel, a trigger pulse is sent to the Timer/Counter 2 (TnCNT2 register), decrementing it by one. If the TnCNT2 register has reached the underflow condition and the end-of-pulse condition is detected by the trigger logic as well, the clock of the Timer/Counter 1 is disabled immediately.

The figure below shows the block diagram related to the MFT mode 1a.

Figure 17. MFT mode 1a block diagram



In mode 1a, Timer/Counter 2 behaves differently from the way it behaves in the other modes. If an underflow condition occurs, the counter is preset to 0x0000 and not 0xFFFF.

The TnCNT1 register starts to count:

- either on an external event on TnB input,
- or by software if the enable TnPTSE bit has been set by setting the TnPTET bit.

Note: The start of count request through TnPTET bit setting when software trigger option is chosen must be done after the MFT is enabled (TnEN bit in TnMCTRL register).

Any time the counter is stopped by choosing "no-clock" by the Timer/Counter 1 clock selector (TnCKC register), it obtains its first reload value after it has been started again from the TnCRA register. Upon reset, the MFT is disabled. Every time this mode starts, the first reload is from register TnCRA. Once the underflow condition for TnCNT2 has been reached, TnCNT2 must be initialized again by the application. It is not reloaded by any reload register.

Timer/Counter 2 can be configured to generate an interrupt upon underflow if enabled by the TnDIEN bit. Refer to Section 3.13.2.6 Timer IO functions for additional details.

In pulse-train mode, the value of TnCNT2 register specifies the number of pulses to be generated, plus one additional pulse (TnCNT2+1 number of pulses).

In pulse-train mode, the trigger logic uses events on TnB to enable the Timer/Counter 1 clock. This function has to be enabled by setting the TnPTSE bit to 0.

The TnB pin can be configured to sense either rising or falling edges.

The Timer/Counter 1 can be configured to toggle the TnA output bit upon underflow. This results in the generation of a pulse signal on TnA, with the width and duty cycle controlled by the values stored in the TnCRA and TnCRB registers. This is a processor-independent PWM signal because once the timer is set up, no more interactions are required from the software or the CPU in order to generate other PWM pulses. The initial value of the PWM output signal can be selected by software to be either high or low. Refer to Section 3.13.2.6 Timer IO functions for additional details.

The timer can be configured to generate separate interrupts upon reload from TnCRA and TnCRB. The TnAPND or TnBPND flags, which are set by the hardware upon occurrence of a timer reload, indicate which interrupt has occurred. The interrupts can be enabled or disabled under software control. Refer to Section 3.13.2.8 Timer interrupts for detailed information.

3.13.2.3 *MFT mode 2: dual-input capture mode*

The mode 2 is used to capture transitions on two selected input pads of the device. The Timer/Counter1 can be used to manage the dual-capture feature as follows:

- A transition on input pad connected to the TnA pin of the MFT generates a transfer of TnCNT1 register value in TnCRA register.
- A transition on input pad connected to the TnB pin of the MFT generates a transfer of TnCNT1 register value in TnCRB register.

The Timer/Counter2 can be used:

- As a system counter: to count down at the rate of the selected clock.

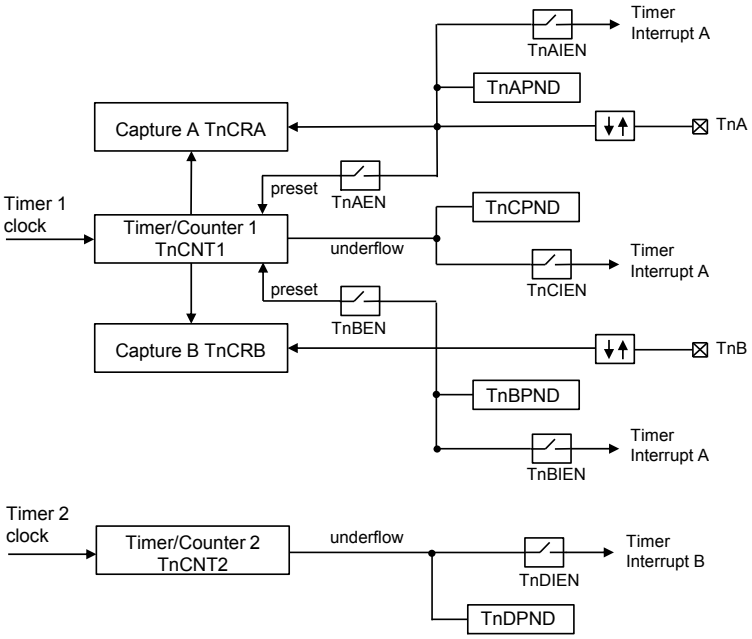
Note: The device input pad selection is done using the register MFTX of the GPIO peripheral.

The transition edge to capture has to be defined in TnAEDG and TnBEDG bits of the TnMCTRL register.

The TnA and TnB inputs can be configured to perform a counter preset to 0xFFFF upon reception of a valid capture event using TnAEN and TnBEN bits in TnMCTRL register. In this case, the current value of the counter is transferred to the corresponding capture register and then the counter is preset to 0xFFFF. Using this approach directly allows the software to determine the on-time, off-time, or period of an external signal, while reducing CPU overhead.

In Figure 19. MFT mode 2 block diagram below the block diagram related to the MFT mode 2.

Figure 18. **MFT mode 2 block diagram**



The input signal on TnA and TnB must have a pulse width equal to or greater than one system clock cycle. The value captured in the TnCRA register at different times reflects the elapsed time between transitions on the TnA pin. The same is true for the TnCRB register and TnB pin. Each input pin can be configured to sense either positive edge or negative edge transitions.

The timer can be configured to generate interrupts on reception of a transition on either TnA or TnB, which can be enabled or disabled separately by the TnAIEN and TnBIEN bits. An underflow of TnCNT1 can generate an interrupt if enabled by the TnCIEN bit. All three interrupts have individual pending flags associated with them. See Section 3.13.2.8 Timer interrupts for further details.

The Timer/Counter 2 can be used as a simple system timer in this mode of operation. The TnCNT2 register counts down with the clock selected by the Timer/Counter 2 clock selector (TnCKC register), and can be configured to generate an interrupt upon underflow if enabled by the TnDIEN bit. See Section 3.13.2.8 Timer interrupts for detailed information.

The Timer/Counter 1 cannot operate in the pulse-accumulate or external-event counter modes, since the TnB pin is used as a capture input. Selecting either of these modes for the Timer/Counter 1 causes the TnCNT1 register to be stopped. However, all available clock source modes may be selected for the Timer/Counter 2. Thus, it is possible to determine the number of capture events on TnB or the elapsed time between capture events on TnB by using the Timer/Counter 2.

3.13.2.4 *MFT mode 3: dual independent timer/counter mode*

This mode 3 allows using the Timer/ Counter 1 and Timer/ Counter 2 separately.

The Timer/ Counter1 can be used:

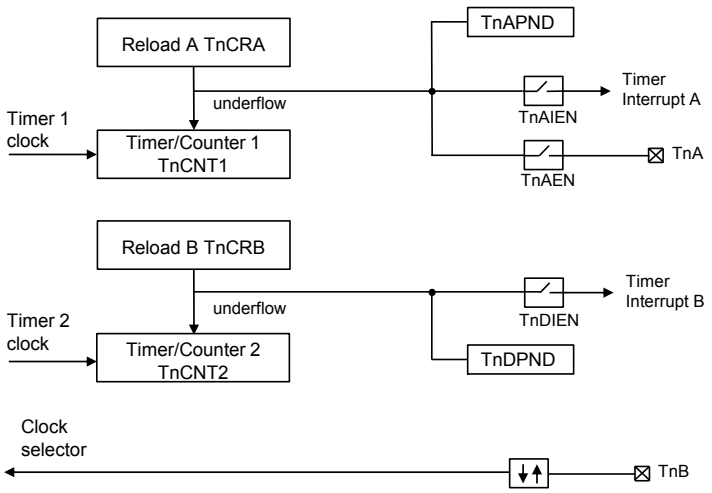
- As a system counter: to count down at the rate of the selected clock.
- To generate a 50% duty-cycle clock signal on TnA pin connected to the PWMx pin of the device (the TnCNT1 register is reloaded with the value of the TnCRA register on underflow event).
- To be an event counter using TnB pin as an external event or pulse-accumulate input.

The Timer/Counter 2 can be used:

- As a system counter: to count down at the rate of the selected clock.
- To be an event counter using TnB pin as an external event or pulse-accumulate input.

In figure below the block diagram related to the MFT mode 3.

Figure 19. MFT mode 3 block diagram



In mode 3, the Timer/Counter is configured to operate as a dual independent system timer or dual external-event counter. In addition, the Timer/Counter 1 can generate a 50% duty cycle PWM signal on the TnA pin. The TnB pin can be used as an external-event input or pulse-accumulate input, and serve as the clock source to either Timer/Counter 1 or Timer/Counter 2. Both counters can also be operated from the prescaled system clock.

The Timer/Counter 1 counts down at the rate of the selected clock. Upon underflow, TnCNT1 register is reloaded from the TnCRA register and counting proceeds. If enabled, the TnA pin toggles upon underflow of the TnCNT1 register. The initial value of the TnA output can be selected by software to be either high or low. See Section 3.13.2.6 Timer IO functions for additional details.

In addition, the TnAPND interrupt-pending flag is set, and a timer interrupt A is generated if the TnAIEN bit is set. See Section 3.13.2.8 Timer interrupts for detailed information.

Since TnA toggles upon every underflow, a 50% duty-cycle PWM signal can be generated on TnA without requiring any interaction of the software or the CPU.

The Timer/Counter 2 counts down at the rate of the selected clock. Upon every underflow of the TnCNT2 register, the value contained in the TnCRB register is loaded into TnCNT2 and counting proceeds downward from that value.

In addition, the TnDPND interrupt-pending flag is set and a timer interrupt B is generated if the TnDIEN bit is set. See Section 3.13.2.8 Timer interrupts for detailed information.

3.13.2.5 *MFT mode 4: input-capture plus timer mode*

This mode 4 is combination of mode 3 and mode 2.

The Timer/Counter1 can be used:

- As a system counter: to count down at the rate of the selected clock.
- To generate a 50% duty-cycle clock signal on TnA pin connected to the PWMx pin of the device (the TnCNT1 register is reloaded with the value of the TnCRA register on underflow event).

The Timer/Counter 2 can be used:

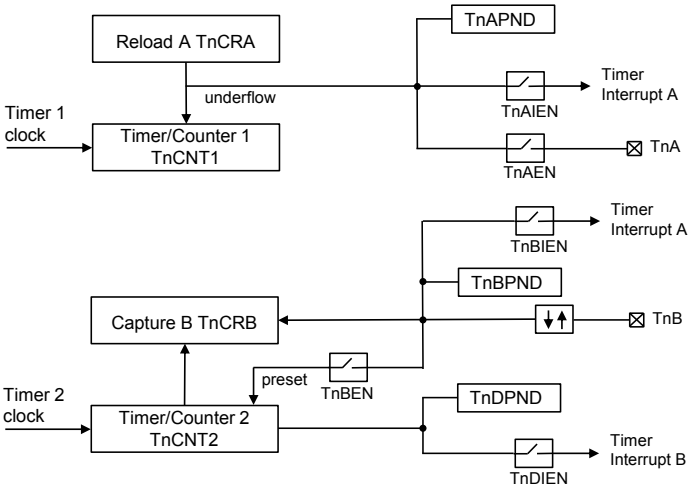
- As a system counter: to count down at the rate of the selected clock.
- A transition on input pad connected to TnB pin of the MFT generates a transfer of TnCNT2 register value in TnCRB register.

Note: The device input pad selection is done using the register MFTX of the GPIO peripheral.

The transition edge to capture has to be defined in TnBEDG bit of the TnMCTRL register. The TnB input can be configured to perform a counter preset to 0xFFFF upon reception of a valid capture event using TnBEN bit in TnMCTRL register.

In figure below the block diagram related to the MFT mode 4.

Figure 20. **MFT mode 4 block diagram**



This mode is a combination of mode 3 and mode 2, and makes it possible to operate Timer/Counter 2 as a single input-capture timer while the Timer/Counter 1 can be used as a system timer as described above.

The Timer/Counter 1 starts counting down once a clock has been enabled. Upon underflow, the TnCNT1 register is reloaded from the TnCRA register, and counting proceeds downward from that value. If enabled, the TnA pin toggles upon every underflow of the TnCNT1 register. The initial value of the TnA output signal can be selected by software to be either high or low. See Section 3.13.2.6 Timer IO functions for additional details.

In addition, the TnAPND interrupt-pending flag is set and a timer interrupt A is generated if the TnAIEN bit is set. See Section 3.13.2.6 Timer IO functions for additional details.

Since TnA toggles upon every underflow, a 50% duty-cycle PWM signal can be generated on TnA without requiring any interaction with the software or the CPU.

The Timer/Counter 2 starts counting down once a clock has been enabled. When a transition is received on TnB, the value contained in the TnCNT2 register is transferred to TnCRB register, and the interrupt-pending flag TnBPND is set. A timer interrupt A is generated if enabled. The software can enable a preset of the counter to 0xFFFF upon detection of a transition on TnB. In this case, the current value of the TnCNT2 register is transferred to TnCRB register, followed by a preset of the counter to 0xFFFF. TnCNT2 starts counting downwards from 0xFFFF until the next transition is received on TnB, which causes the procedure of capture and preset to be repeated. The underflow of the TnCNT2 register causes the TnDPND interrupt-pending flag to be set, and can also generate a timer interrupt B if enabled. See Section 3.13.2.8 Timer interrupts for detailed information.

The input signal on TnB must have a pulse width equal to or greater than one system clock cycle. TnB can be configured to sense either rising or falling edges. The Timer/Counter 2 cannot operate in the pulse-accumulate or external-event counter modes since the TnB input is used as a capture input. Selecting either of these modes for the Timer/Counter 2 causes the TnCNT2 register to be stopped.

However, all available clock source modes may be selected for the Timer/Counter 1. Thus using the TnCNT1 register, it is possible to determine the number of capture events on TnB, or the elapsed time between capture events on TnB.

3.13.2.6 *Timer IO functions*

There are two pins associated with each instance of the MFTX. The pins are called TnA and TnB. The functionality of TnA and TnB depends on the mode of operation and the value of the TnAEN and TnBEN bits. Table 153. MFT IO functions shows the function of TnA and TnB for various modes of operation. Note that if TnA functions as a PWM output, TnAOUT defines the initial and present value of TnA. For example, if the user wishes to start with TnA high, TnAOUT needs to be set before enabling the timer clock.

Table 153. MFT IO functions

Pin	TnAEN TnBEN	Mode 1	Mode 1a	Mode 2	Mode 3	Mode 4
		PWM and counter	PWM pulse train	Dual-input capture and counter	Dual independent counter	Input capture plus timer
TnA	TnAEN = 0 TnBEN = X	No output	No output	Capture TnCNT1 into TnCRA	No output toggle	No output toggle
	TnAEN = 1 TnBEN = X	Toggle output on underflow of TnCNT1	Toggle output on underflow of TnCNT1	Capture TnCNT1 into TnCRA and preset TnCNT1	Toggle output on underflow of TnCNT1	Toggle output on underflow of TnCNT1
TnB	TnAEN = X TnBEN = 0	External event or pulse accumulate input	External event if TnPTSE = 0	Capture TnCNT1 into TnCRB	External event or pulse accumulate input	Capture TnCNT2 into TnCRB
	TnAEN = X TnBEN = 1	External event or pulse accumulate input	External event if TnPTSE = 0	Capture TnCNT1 into TnCRA and preset TnCNT1	External event or pulse accumulate input	Capture TnCNT2 into TnCRB and preset TnCNT2

3.13.2.7 *IO configuration linked to MFT timers*

The MFT timers can be connected to the GPIOs for the following features:

- Input signal used to trigger the timer in capture mode.
- Output signal when a PWM mode is used

In capture mode, the timer waits for an external IO event to start counting. The chosen IO for capture is programmed through the register MFTX of the GPIO peripheral. This register allows configuring input capture IO for Timer/Counter 1 and Timer/Counter 2 of both MFT1 and MFT2, depending on which timer(s) are configured in capture mode.

In PWM mode, the signal is output on PWM0 IO for MFT1 and PWM1 for MFT2. Those IOs are available at different GPIOs thanks to alternate option. So to output the chosen PWM signal, it is necessary to configure the IO with the dedicated mode.

3.13.2.8 *Timer interrupts*

The MFT has four interrupt sources, which are mapped to two different system interrupts. All sources have a pending flag associated with them, and can be enabled or disabled by software. The pending flags are named TnXPND, where n denotes the instance of the module, and X represents a letter from A to D. An interrupt enable flag (TnXIEN) is associated with each interrupt-pending flag. Interrupt sources A, B and C can each generate a timer interrupt MFT1A for MFT1 and MFT2A for MFT2, whereas interrupt source D can generate a timer interrupt MFT1B for MFT1 and MFT2B for MFT2. Not all interrupt sources are available in all modes. Table 154. MFT interrupt functions shows which events can trigger an interrupt in which mode of operation.

Table 154. MFT interrupt functions

MFT interrupt	Interrupt pending flag	Mode 1	Mode 1a	Mode 2	Mode 3	Mode 4
		PWM and counter	PWM pulse train	Dual-input capture and counter	Dual independent counter	Input capture plus timer
Timer interrupt A (MFT1A, MFT2A)	TnAPND	TnCNT1 reload from TnCRA	TnCNT1 reload from TnCRA	Input capture on TnA transition	TnCNT1 reload from TnCRA	TnCNT1 reload from TnCRA
	TnBPND	TnCNT1 reload from TnCRB	TnCNT1 reload from TnCRB	Input capture on TnB transition	N/A	Input capture on TnB transition
	TnCPND	N/A	N/A	TnCNT1 underflow	N/A	N/A
Timer interrupt B (MFT1B, MFT2B)	TnDPND	TnCNT2 underflow	TnCNT2 underflow	TnCNT2 underflow	TnCNT2 reload from TnCRB	TnCNT2 underflow

3.13.3 MFT registers

MFT1 peripheral base address (MFT1_BASE_ADDR) 0x40D00000.

MFT2 peripheral base address (MFT2_BASE_ADDR) 0x40E00000.

Table 155. MFTX registers

Address offset	Name	RW	Reset	Description
0x00	TnCNT1	RW	0x00000000	Timer / Counter1 register. Refer to the detailed description below.
0x04	TnCRA	RW	0x00000000	Capture / Reload A register. Refer to the detailed description below.
0x08	TnCRB	RW	0x00000000	Capture / Reload B register. Refer to the detailed description below.
0x0C	TnCNT2	RW	0x00000000	Timer / Counter 2 register. Refer to the detailed description below.
0x10	TnPRSC	RW	0x00000000	Clock prescaler register. Refer to the detailed description below.
0x14	TnCKC	RW	0x00000000	Clock unit control register. Refer to the detailed description below.
0x18	TnMCTRL	RW	0x00000000	Timer mode control register. Refer to the detailed description below.
0x1C	TnICTRL	RW	0x00000000	Timer interrupt control register. Refer to the detailed description below.
0x20	TnICLR	RW	0x00000000	Timer interrupt clear register. Refer to the detailed description below.

Table 156. MFTX – TnCNT1 register description: address offset MFTX_BASE_ADDR+0x00

Bit	Field name	Reset	RW	Description
15:0	TnCNT1	0x0000	RW	The Timer/Counter 1 register is a 16-bit RW register that is not altered by reset and thus contains random data upon power-up. Reading the register returns the current value of the Timer/Counter 1. TnCNT1 can only be written by the software when MFT is enabled (TnEN = 1). When MFT is disabled (TnEN = 0), write operations on TnCNT1 register are ignored.
31:16	RESERVED	0x0	RW	RESERVED

Table 157. MFTX – TnCRA register description: address offset MFTX_BASE_ADDR+0x04

Bit	Field name	Reset	RW	Description
15:0	TnCRA	0x0000	RW	The Capture/Reload A register is a 16-bit RW register that is not affected by reset and thus contains random data upon power-up. The software may read the register at any time. However, the register can only be written by the software when MFT is enabled (TnEN = 1). When MFT is disabled (TnEN = 0), write operations on TnCRA register are ignored.
31:16	RESERVED	0x0	RW	RESERVED

Table 158. MFTX – TnCRB register description: address offset MFTX_BASE_ADDR+0x08

Bit	Field name	Reset	RW	Description
15:0	TnCRB	0x0000	RW	The Capture/Reload B register is a 16-bit RW register that is not affected by reset and thus contains random data upon power-up. The software may read the register at any time. However, the register can only be written by the software when MFT is enabled (TnEN = 1). When MFT is disabled (TnEN = 0), write operations on TnCRB register are ignored.
31:16	RESERVED	0x0	RW	RESERVED

Table 159. MFTX – TnCNT2 register description: address offset MFTX_BASE_ADDR+0x0C

Bit	Field name	Reset	RW	Description
15:0	TnCNT2	0x0000	RW	The Timer/Counter 2 register is a 16-bit RW register that is not altered by reset and thus contains random data upon power-up. Reading the register returns the current value of the Timer/Counter 2. TnCNT2 can only be written by the software when MFT is enabled (TnEN = 1). When MFT is disabled (TnEN = 0), write operations on TnCNT2 register are ignored.
31:16	RESERVED	0x0	RW	RESERVED

Table 160. MFTX – TnPRSC register description: address offset MFTX_BASE_ADDR+0x10

Bit	Field name	Reset	RW	Description
7:0	TnPRSC	0x00	RW	The clock prescaler register is an 8-bit RW register. It contains the current value of the clock prescaler, which determines the timer clock prescaler ratio. The register value can be changed at any time. In all operating modes except pulse-train (mode1a), a modified value is used upon an underflow of the internal prescaler counter. In mode 1a, the new value is used either upon start of a new pulse train (a write to TnPTSE), or upon an event on TnB (if TnPTET=1). The timer clock is generated by dividing the system clock by TnPRSC + 1. Therefore, the maximum timer clock frequency is equal to the frequency of the system clock (TnPRSC = 0x00), and the minimum timer clock is the frequency of the system clock divided by 256 (TnPRSC = 0xFF).
31:8	RESERVED	0x0	RW	RESERVED

Table 161. MFTX - TnCKC register description: address offset MFTX_BASE_ADDR+0x14

Bit	Field name	Reset	RW	Description
2:0	TnC1CSEL	0x0	RW	Determines the clock mode for the Timer/Counter 1: • 000b: No clock (Timer/Counter 1 stopped). • 001b: System clock with configurable prescaler (register TnPRSC). • 010b: External event on TnB (mode 1 and 3 only). • 011b: Pulse accumulate (mode 1 and 3 only).
5:3	TnC2CSEL	0x0	RW	Determines the clock mode for the Timer/Counter 2: • 000b: No clock (Timer/Counter 2 stopped). • 001b: System clock with configurable prescaler (register TnPRSC). • 010b: External event on TnB (mode 1 and 3 only). • 011b: Pulse accumulate (mode 1 and 3 only).
31:6	RESERVED	0x0	RW	RESERVED

Table 162. MFTX - TnMCTRL register description: address offset MFTX_BASE_ADDR+0x18

Bit	Field name	Reset	RW	Description
1:0	TnMDSEL	0x0	RW	MFT mode select: • 00b: Mode 1 or 1a: PWM mode and system timer or pulse train mode • 01b: Mode 2: Dual-input capture mode and system timer • 10b: Mode 3: Dual independent mode Timer/Counter mode • 11b: Mode 4: Single timer and single input capture mode
2	TnAEDG	0x0	RW	Configure the TnA edge polarity for trigging an action: 0: Input is sensitive to falling edges.

Bit	Field name	Reset	RW	Description
				1: Input is sensitive to rising edges.
3	TnBEDG	0x0	RW	Configure the TnB edge polarity for trigging an action: 0: Input is sensitive to falling edges. 1: Input is sensitive to rising edges.
4	TnAEN	0x0	RW	Enables TnA to either function as a preset input or as a PWM output depending on the mode of operation. If the bit is set (1) while operating in the dual-input capture mode (mode 2), a transition on TnA causes TnCNT1 to be preset to 0xFFFF. In the remaining modes of operation, setting TnAEN enables TnA to function as a PWM output 0: TnA input disable. 1: TnA input enable.
5	TnBEN	0x0	RW	TnB Enable: If set (1) and while operating in dual-input capture mode (mode 2) or input capture and timer mode (mode 4), a transition on TnB causes the corresponding Timer/Counter to be preset to 0xFFFF. In mode 2, TnCNT1 is preset to 0xFFFF, while in mode 4, TnCNT2 is preset to 0xFFFF. The bit has no effect while operating in any other modes than mode 2 or mode 4. 0: TnB input disable. 1: TnB input enable.
6	TnAOUT	0x0	RW	The TnA output data contains the value of the TnA when used as PWM output. The bit will be set and cleared by the hardware and thus reflects the status of TnA. The bit can be read or written by software at any time. If the hardware is attempting to toggle the bit at the same time that software writes to the bit, the software write will take precedence over the hardware update. The bit has no effect when TnA is used as an input or when the module is disabled: 0: TnA pin is low. 1: TnA pin is high.
7	TnEN	0x0	RW	MFT Enable: This bit enables or disables the MFT peripherals. When the bit is set (1), MFT is enabled, and when the bit is cleared (0), MFT is disabled. When MFT is disabled, all clocks to the counter unit are stopped, thus decreasing power consumption to a minimum. For that reason, the Timer/Counter registers (TnCNT1, TnCNT2), the Capture/Reload registers (TnCRA, TnCRB) and the interrupt-pending bits (TnXPND) cannot be written by software. Furthermore, the 8-bit clock prescaler and the interrupt-pending bits are reset and the TnA I/O pin becomes an input: 0: MFT disable 1: MFT enable
8	TnPTEN	0x0	RW	This bitfield enable the mode 1a. If set (1) while TnMDSEL is set to 00b, the Timer/Counter 1 operates in PWM pulse-train mode (mode 1a). The bit has no effect while TnMDSEL is set to any value other than 00b. 0: Mode 1a not selected. 1: Mode 1a selected (if TnMDSEL = 00b).
9	TnPTSE	0x0	RW	Tn Pulse-Train software trigger enable: if set (1) while operating in PWM pulse-train mode (mode 1a), the pulse-train generation can only be triggered by setting the TnPTET to 1. If the TnPTSE bit is reset (0), pulses are generated only if a transition occurs on TnB. The bit has no effect while operating in any other modes than timer mode 1a: 0: No effect 1: Pulse-train generation trigger (in mode 1a)
10	TnPTET	0x0	RW	Tn Pulse-Train event trigger: if set (1) while operating in pulse-train mode (mode 1a) and the TnPTSE bit is set (1), pulse-train generation is triggered. When Timer/Counter 2 (TnCNT2) reaches its underflow condition, this bit is reset (0). If the TnPTSE bit is not set (0) while operating in pulse-train mode (mode 1a), the TnPTET bit cannot be written. Therefore, a 1 in TnPTET indicates that an external event started a pulse-train generation that is not yet finished. When the pulse-train is finished, the bit is reset to 0: 0: No pulse-train event trigger occurred. 1: Pulse-train event trigger occurred (in mode 1a).

Bit	Field name	Reset	RW	Description
31:11	RESERVED	0x0	RW	RESERVED

Table 163. MFTX - TNICTRL register description: address offset MFTX_BASE_ADDR+0x1C

Bit	Field name	Reset	RW	Description
0	TNAPND	0x0	R	Timer interrupt A pending:· 0: No interrupt source pending. 1: Interrupt source pending.
1	TNBPND	0x0	R	Timer interrupt B pending: 0: No interrupt source pending. 1: Interrupt source pending.
2	TNCPND	0x0	R	Timer interrupt C pending: 0: No interrupt source pending. 1: Interrupt source pending.
3	TNDPND	0x0	R	Timer interrupt D pending: 0: No interrupt source pending. 1: Interrupt source pending.
4	TNAIEN	0x0	RW	Timer interrupt A enable: 0: Interrupt disabled. 1: Interrupt enabled.
5	TNBIEN	0x0	RW	Timer interrupt B enable: 0: Interrupt disabled. 1: Interrupt enabled.
6	TNCIEN	0x0	RW	Timer interrupt C enable: 0: Interrupt disabled. 1: Interrupt enabled
7	TNDIEN	0x0	RW	Timer interrupt D enable: 0: Interrupt disabled. 1: Interrupt enabled.
31:8	RESERVED	0x0	RW	RESERVED

Table 164. MFTX - TNICLR register description: address offset MFTX_BASE_ADDR+0x20

Bit	Field name	Reset	RW	Description
0	TNACLR	0x0	W	1: clear the timer pending flag A
1	TNBCLR	0x0	W	1: clear the timer pending flag B.
2	TNCCLR	0x0	W	1: clear the timer pending flag C.
3	TNDCLR	0x0	W	1: clear the timer pending flag D.
31:4	RESERVED	0x0	W	RESERVED

Note: All RESERVED fields inside registers must always be written with their default value.

3.14 Watchdog

3.14.1 Introduction

The watchdog timer provides a way of recovering from software crashes.

The watchdog monitors the interrupt and asserts a reset signal if the interrupt remains unserved for the entire programmed period.

The watchdog clock is used to generate a regular interrupt, depending on a programmed value. It is counting down at a fixed frequency around 32.768 kHz provided either by embedded RCO or by the external XO 32 kHz.

Main features are:

- 32-bit down counter at fixed frequency 32.768 kHz
- Generate an interrupt each time the counter reaches zero
- Generate an internal reset that reboot the system if the generated interrupt is not cleared by software and a second interrupt occurs

3.14.2 Functional description

The watchdog timer is a 32-bit down counter that divides the clock input to produce an interrupt. The divide ratio is fully programmable and controls the interrupt interval, which can be calculated as:

Interrupt interval = (WDT_LOAD + 1) / (clock frequency in Hz).

The table below shows examples of WDT_LOAD values.

Table 165. Watchdog interrupt interval

WDT_LOAD	Interrupt interval (ms)
4294967295	131072000
65535	2000
32767	1000
4095	125
127	3.90625
63	1.953125
1	0.0610

A watchdog interrupt is generated each time the counter reaches zero. The counter is then reloaded with the content of the WDT_LR register. The interrupt status should be cleared by writing to the interrupt clear register. When the interrupt is cleared, the counter is reloaded with the WDT_LOAD value. If the interrupt status is not cleared and a new interrupt is generated, then a watchdog Reset is generated, rebooting the system.

The watchdog interrupt and Reset generation can be enabled or disabled as required by the system using the relevant bits in the control register. When the interrupt generation is disabled the watchdog counter is also stopped, and when the interrupt is enabled the counter will start from the programmed value, not the last-count value.

Write access to the registers within the watchdog timer can be disabled in the watchdog lock register. Writing a value of 0x1ACC_E551 to this WDT_LOCK register allows write access to all other registers; writing any other value disables write access. This feature is included to allow some protection against software that might otherwise disable the watchdog functionality.

3.14.3 Watchdog registers

WDG peripheral base address (WDG_BASE_ADDR) 0x40700000.

Table 166. **WDG registers**

Address offset	Name	RW	Reset	Description
0x00	LR	RW	0xFFFFFFFF	Watchdog load register. Refer to the detailed description below.
0x04	VAL	R	0xFFFFFFFF	Watchdog value register. Refer to the detailed description below.
0x08	CR	RW	0x00000000	Watchdog control register. Refer to the detailed description below.
0x0C	ICR	RW	0x00000000	Watchdog interrupt clear register. Refer to the detailed description below.
0x10	RIS	R	0x00000000	Watchdog raw interrupt status register. Refer to the detailed description below.
0x14	MIS	R	0x00000000	Watchdog masked interrupt status register. Refer to the detailed description below.
0xC00	LOCK	RW	0x00000000	Watchdog Lock register. Refer to the detailed description below.

Table 167. **WDG - LR register description: address offset WDG_BASE_ADDR+0x00**

Bit	Field name	Reset	RW	Description
31:0	LOAD	0xFFFFFFFF	RW	Watchdog load value. Value from which the counter is to decrement. When this register is written to, the count is immediately restarted from the new value.

Table 168. **WDG - VAL register description: address offset WDG_BASE_ADDR+0x04**

Bit	Field name	Reset	RW	Description
31:0	WDTVAL	0xFFFFFFFF	R	Watchdog current value. When read, returns the current value of the decrementing watchdog counter. A write has no effect.

Table 169. **WDG - CR register description: address offset WDG_BASE_ADDR+0x08**

Bit	Field name	Reset	RW	Description
0	INTEN	0x0	RW	Watchdog interrupt enable. Enable the interrupt event: 0: watchdog interrupt is disabled. 1: watchdog interrupt is enabled.
1	RESEN	0x0	RW	Watchdog reset enable. Enable the watchdog reset output: 0: watchdog reset is disabled. 1: watchdog reset is enabled.
31:2	RESERVED	0x0	RW	RESERVED

Table 170. **WDG - ICR register description: address offset WDG_BASE_ADDR+0x0C**

Bit	Field name	Reset	RW	Description
31:0	WDTICLR	0x0	RW	Watchdog interrupt clear: Writing any value will clear the watchdog interrupt and reloads the counter from the LR register. A read returns zero.

Table 171. WDG - RIS register description: address offset WDG_BASE_ADDR+0x10

Bit	Field name	Reset	RW	Description
0	RIS	0x0	R	Watchdog raw interrupt status bit. Reflects the status of the interrupt status from the watchdog: 0: watchdog interrupt is not active. 1: watchdog interrupt is active. Read-only bit. A write has no effect.
31:1	RESERVED	0x0	R	RESERVED

Table 172. WDG - MIS register description: address offset WDG_BASE_ADDR+0x14

Bit	Field name	Reset	RW	Description
0	MIS	0x0	R	Watchdog masked interrupt status bit. Masked value of watchdog interrupt status: 0: watchdog interrupt is not active. 1: watchdog interrupt is active. Read-only bit. A write has no effect.
31:1	RESERVED	0x0	R	RESERVED

Table 173. WDG - LOCK register description: address offset WDG_BASE_ADDR+0xC00

Bit	Field name	Reset	RW	Description
31:0	LOCKVAL	0x0	RW	Watchdog lock value. When read, returns the lock status: 0: Write access to all watchdog other registers is enabled. 1: Write access to all watchdog other registers is disabled. When written, allows enabling or disabling write access to all other watchdog registers: Writing 0x1ACCE551: Write access to all other registers is enabled. Writing any other value: Write access to all other registers is disabled.

Note: All RESERVED fields inside registers must always be written with their default value.

3.15 RTC

3.15.1 Introduction

The RTC timer can be used to provide an interrupt at regular time intervals. It generates an interrupt signal when it reaches zero after decrementing for a programmed number of cycles of the real-time clock input. The RTC timer can restart automatically from a load value when reaching zero if the auto restart mode is enabled, or it can stop when it reaches zero.

The RTC is clocked by the 32 kHz clock and is switched off in low-power modes which prevents this timer to be used for wake-up events.

3.15.2 Functional description

The RTC peripheral can be used either as real-time clock timer or as real-time clock watch.

3.15.2.1 Real-time clock timer

The real-time clock timer (RTC timer) can be used to provide an interrupt at regular time intervals.

The RTC timer can restart automatically from a load value when reaching zero if the auto restart mode is enabled, or it can stop when it reaches zero.

The RTC timer has the following features:

- 32-bit down-counter.
- Interrupt generation when timer reaches zero.
- Start, auto restart (after counts to zero) and stop capability.
- On-the-fly register read and write access.
- 1/32 kHz minimum period.
- Multiple modes: periodic interrupt and single interrupt generation.
- Capability to switch between two load values in periodic mode. The timer reloads alternatively from one load value to the other and the down-counter starts decrementing every 31.25 µs (on average).

The RTC timer is a 32-bit free-running counter, clocked by the 32 kHz clock signal (from an embedded 32 kHz RC), that works in two modes: periodic and one-shot.

Table 174. RTC modes

RTTOS bit	Mode	Description
0b	Periodic	The counter generates an interrupt at a constant interval, reloading a load value after wrapping past zero. There are two load values: RTC_TLR1 for pattern value 0 and RTC_TLR2 for pattern value 1.
1b	One-shot	The counter generates an interrupt once. When the counter reaches zero, it halts until the user restarts it by: setting bit RTTEN in the RTC_TCR register of writing a new value to the load register RTC_TLR1

The RTC timer load registers define the values from which the counter restarts alternatively.

In periodic mode, the timer must be stopped by the software before writing to a load register. The counter loads a value from RTC_TLR1 or from RTC_TLR2, depending on the value of the current pattern value that crosses the pattern register to decide after each interrupt generation, which value to load. The number of pattern bits to be crossed periodically (from the 128 bits) is specified in RTC_TCR [10:4]. This process offers the possibility to have better precision of the average tick period.

In one-shot mode, the timer stops when it reaches zero, but the software can also stop it. Once the counter is halted, the load registers (RTC_TLR1, RTC_TLR2) can be written and the counter considers the new written value. After a write, RTTEN (RTC_TCR) is set if the timer is in self-start mode.

Note: Writing to RTC_TLR1 or RTC_TLR2 has no effect when the counter is running (the registers contents are not changed).

Note: Two consecutive write operations to the RTC_TCR register must be separated by at least 3 times the low speed clock period plus twice the system clock period. This time is about 140 us. If this time is not satisfied, the last written value cannot be guaranteed. The software can read back the RTC_TCR register value after at least 1 period of the low speed clock.

3.15.2.2 *Real-time clock watch*

The RTC clock watch consists of two counters and two alarm registers that have the following features:

- Two counters:
  - Counts seconds, minutes, hours, days of the week, days of the month.
  - Counts years.
- Two alarm registers:
  - To trigger an interrupt at exact date and time.

The clock watch counters are split in two registers:

The RTC_CWDR register that holds:

- Seconds on six bits. Valid values are 0 to 59. (60, 61, 62 and 63 are invalid values, programming the RTC_CWDR with these values leads to unpredictable behavior.) The seconds are incremented on the CLK1HZ clock rate.

- Minutes on six bits. Valid values are 0 to 59 (60, 61, 62 and 63 are invalid values, programming the RTC_CWDR with these values leads to unpredictable behavior.)
- Hours on five bits. Valid values are 0 to 23 (24 to 31 are invalid values, programming the RTC_CWDR with these values leads to unpredictable behavior.)
- The day of the week on three bits. Valid values are 1 (Sunday) to 7 (Saturday) (0 is an invalid value, programming the RTC_CWDR with this value leads to unpredictable behavior.)
- The day of the month on five bits. Valid values are 1 to 31 for January, March, May, July, August, October and December, 1 to 30 for April, June, September, and November, 1 to 29 for February on leap years, or 1 to 28 for February on non-leap years. All other values are invalid values. Programming the RTC_CWDR with these values leads to unpredictable behavior.)
- The month on four bits. Valid values are 1 (January) to 12 (December). 0, 13, 14 and 15 are invalid values, programming the RTC_CWDR with these values leads to unpredictable behavior.

The RTC_CWYR register holds:

- The year, from 0 to 4096

The clock watch time and date can be changed by writing new settings in the RTC_CWDLR and RTC_CWYLR load registers. The new setting is transferred to the clock watch counters on the next CLK1HZ rising edge after the RTC_CWDLR register has been written.

After each increment of the clock watch counters, the RTC_CWDR and RTC_CWYR registers are compared to the clock watch match registers, RTC_CWDMR and RTC_CWYMR.

If both pairs of registers match, the internal interrupt signal RTCWINTR is raised.

3.15.2.3 *RTC interrupts*

The RTC generates two internal interrupt signals:

- RTCINTR: raised when the two clock watch counter registers (RTC_CWDR and RTC_CWYR) match the two clock watch alarm registers (RTC_CWDMR and RTC_CWYMR). Some bit-fields can be ‘don’t care’ during the comparison if a zero value is used for year, month, day of month and day of week. The software must clear this interrupt by writing 1 in the bit RTCCWIC of RTC_ICR register.
- RTTINTR: raised when the full 32-bit down-counter RTC_TDR reaches zero and is only cleared by writing 1 in the bit RTTIC of the RTC_ICR register. The most significant carry bit of the counter detects the counter reaches zero. The software must clear this interrupt by writing 1 in the bit RTCTIC of the RTC_ICR register.

Each individual interrupt can be masked by writing 0b to its corresponding interrupt mask set/clear bit in the RTC_IMSC register. Both the raw interrupt status (prior to masking) and the final interrupt status (after masking) for each individual interrupt signal can be read from the RTC_RIS and RTC_MIS status registers.

The RTC delivers also a single combined interrupt signal, RTUINTR. This interrupt line is the logical OR of the both internal interrupt signals described above and is the signal connected to the processor interrupt line.

3.15.3 RTC registers

RTC peripheral base address (RTC_BASE_ADDR) 0x40F00000.

Table 175. RTC registers

Address offset	Name	RW	Reset	Description
0x00	CWDR	R	0x02120000	Clockwatch data register. Refer to the detailed description below.
0x04	CWDMR	RW	0x00000000	Clockwatch data match register. Refer to the detailed description below.
0x08	CWDLR	RW	0x00000000	Clockwatch data load register. Refer to the detailed description below.
0x0C	CWYR	R	0x00002000	Clockwatch year register. Refer to the detailed description below.
0x10	CWYMR	RW	0x00002000	Clockwatch year match register. Refer to the detailed description below.
0x14	CWYLR	RW	0x00000000	Clockwatch year load register. Refer to the detailed description below.
0x18	CTCR	RW	0x00007FFF	Control trim and counter register. Refer to the detailed description below.
0x1C	IMSC	RW	0x00000000	RTC interrupt mask register. Refer to the detailed description below.
0x20	RIS	R	0x00000000	RTC raw interrupt status register. Refer to the detailed description below.

Address offset	Name	RW	Reset	Description
0x24	MIS	R	0x00000000	RTC masked interrupt status register. Refer to the detailed description below.
0x28	ICR	W	0x00000000	RTC interrupt clear register. Refer to the detailed description below.
0x2C	TDR	R	0xFFFFFFFF	RTC timer load value
0x30	TCR	RW	0x00000000	RTC timer control register. Refer to the detailed description below.
0x34	TLR1	RW	0x00000000	RTC timer first load register
0x38	TLR2	RW	0x00000000	RTC timer second load Register
0x3C	TPR1	RW	0x00000000	RTC timer pattern register (pattern[31:0])
0x40	TPR2	RW	0x00000000	RTC timer pattern register (pattern[63:32])
0x44	TPR3	RW	0x00000000	RTC timer pattern register (pattern[95:64])
0x48	TPR4	RW	0x00000000	RTC timer pattern register (pattern[127:96])
0x4C	TIN	R	0x00000000	RTC timer interrupt number register

Table 176. RTC - CWDR register description: address offset RTC_BASE_ADDR+0x00

Bit	Field name	Reset	RW	Description
5:0	CWSEC	0x0	R	RTC clockwatch second value. Clockwatch seconds: 0 to 59 (max. 0x3B).
11:6	CWMIN	0x0	R	RTC clockwatch minute value. Clockwatch seconds: 0 to 59 (max. 0x3B).
16:12	CWHOUR	0x0	R	RTC clockwatch hour value. Clockwatch seconds: 0 to 23 (max. 0x17).
19:17	CWDAYW	0x1	R	RTC clockwatch day of week value. Clockwatch day of week: 001b: Sunday. 010b: Monday. 011b: Tuesday. 100b: Wednesday. 101b: Thursday. 110b: Friday. 111b: Saturday.
24:20	CWDAYM	0x1	R	RTC clockwatch day of month value: 1 to 28/29/30 or 31. Range of value to program depends on the month: 1 to 28: February month, non-leap year. 1 to 29: February month, leap year. 1 to 30: April, June, September, November month. 1 to 31: January, March, May, July, August, October, December month.
28:25	CWMONTH	0x1	R	RTC clockwatch month value: 0001b: January. ... 1100: December.
31:29	RESERVED	0x0	R	RESERVED

Table 177. **RTC - CWDMR register description: address offset RTC_BASE_ADDR+0x04**

Bit	Field name	Reset	RW	Description
5:0	CWSECM	0x0	RW	RTC clockwatch second match value: 00 0000 to 11 1011: (0 to 59 or 0x00 to 0x3B) clockwatch seconds. 11 1100 to 11 1111 - (60 to 63 or 0x3C to 0x3F). Non-valid data, match never occurs.
11:6	CWMINM	0x0	RW	RTC clockwatch minute match value: 00 0000 to 11 1011: (0 to 59 or 0x00 to 0x3B) clockwatch minutes. 11 1100 to 11 1111 - (60 to 63 or 0x3C to 0x3F). Non-valid data, match never occurs.
16:12	CWHOURM	0x0	RW	RTC clockwatch hour match value: 00000b to 10111b: (0 to 23 or 0x00 to 0x17) hour match value. 11000b to 11111b - (24 to 31 or 0x18 to 0x1F). Non-valid data, match never occurs.
19:17	CWDAYWM	0x0	RW	RTC clockwatch day of week match value: 000b: day of week does not care in the comparison. (Default value after PORn). 001b to 111b: (1 to 7) day of week match value.
24:20	CWDAYMM	0x0	RW	RTC clockwatch day of month match value: 0000b: (month does not care in the comparison. Default value after PORn). 1 to 31: day of month match value.
28:25	CWMONTHM	0x0	RW	RTC clockwatch month match value: 0000b: (day of month does not in the comparison. Default value after PORn). 0001b to 1100b: (1 to 12) month match value. 1101b (13, 0xD) to 1111b (0xF) non-valid data, match never occurs.
31:29	RESERVED	0x0	RW	RESERVED

Table 178. **RTC - CWDLR register description: address offset RTC_BASE_ADDR+0x08**

Bit	Field name	Reset	RW	Description
5:0	CWSECL	0x0	RW	RTC clockwatch second load value. Clockwatch seconds from 0 to 59 (0x3B). Other values must not be used.
11:6	CWMINL	0x0	RW	RTC clockwatch minute load value. Clockwatch minutes from 0 to 59 (0x3B). Other values must not be used.
16:12	CWHOURL	0x0	RW	RTC clockwatch hour load value. Clockwatch hours from 0 to 23 (0x17). Other values must not be used.
19:17	CWDAYWL	0x0	RW	RTC clockwatch day of week load value. Clockwatch day of week: 000b: Must not be used. 001b: Sunday. 010b: Monday. 011b: Tuesday. 100b: Wednesday. 101b: Thursday. 110b: Friday. 111b: Saturday.

Bit	Field name	Reset	RW	Description
24:20	CWDAYML	0x0	RW	RTC clockwatch day of month load value. 1 to 28/29/30 or 31 depending on month: 1 to 28: February month, non-leap year. 1 to 29: February month, leap year. 1 to 30: April, June, September, November month. 1 to 31: January, March, May, July, August, October, December month. Other values must not be used.
28:25	CWMONTHL	0x0	RW	RTC clockwatch month load value: 0001b: January. ... 1100: December. Other values must not be used.
31:29	RESERVED	0x0	RW	RESERVED

Table 179. RTC - CWYR register description: address offset RTC_BASE_ADDR+0x0C

Bit	Field name	Reset	RW	Description
13:0	CWYEAR	0x2000	R	RTC clockwatch year value. Clockwatch year, in BCD format is from 0 to 4096.
31:14	RESERVED	0x0	R	RESERVED

Table 180. RTC - CWYMR register description: address offset RTC_BASE_ADDR+0x10

Bit	Field name	Reset	RW	Description
13:0	CWYEARM	0x2000	RW	RTC clockwatch year match value. Clockwatch year match value is in BCD format from 0 to 4096.
31:14	RESERVED	0x0	RW	RESERVED

Table 181. RTC - CWYLR register description: address offset RTC_BASE_ADDR+0x14

Bit	Field name	Reset	RW	Description
13:0	CWYEARL	0x0	RW	RTC clockwatch year load value. Clockwatch year load value is in BCD format from 0 to 4096.
31:14	RESERVED	0x0	RW	RW RESERVED

Table 182. RTC - CTCR register description: address offset RTC_BASE_ADDR+0x18

Bit	Field name	Reset	RW	Description
14:0	CKDIV	0x7FFF	RW	Clock divider factor. This value plus one represents the integer part of the CLK32K clock divider used to produce the reference 1 Hz clock. 0x000: CLK1HZ clock is similar to CLK32K for RTC timer and stopped for RTC clockwatch. 0x0001: 2 CLK32K clock cycles per CLK1HZ clock cycle. ...

Bit	Field name	Reset	RW	Description
				0x7FFF: 32768 CLK32K clock cycles per CLK1HZ clock cycle (default value after PORn Reset). ... 0xFFFF: CLK32K clock cycles per CLK1HZ clock cycle. Writing to this bit-field is disregarded if CWEN = 1. A read returns the value of the CKDIV bit-field.
15	RESERVED	0x0	RW	RESERVED
25:16	CKDEL	0x0	RW	Trim delete count. This value represents the number of CLK32K clock pulses to delete every 1023 CLK32K clock cycles to get a better reference 1 Hz clock for incrementing the RTC counter. 0x000: No CLK32K clock cycle is deleted every 1023 CLK1HZ clock cycles (default value after PORn Reset). 0x001: 1 CLK32K clock cycle is deleted every 1023 CLK1HZ clock cycles. ... 0x3FF: 1023 CLK32K clock cycles are deleted every 1023 CLK1HZ clock cycles. Writing to this bit-field is disregarded if CWEN = 1. A read returns the value of the CKDEL bit-field.
26	CWEN	0x0	RW	Clockwatch enable bit. When set to 1, the clockwatch is enabled. Once it is enabled, any write to this register has no effect until a Power-On-Reset. A read returns the value of the CWEN bit value.
31:27	RESERVED	0x0	RW	RESERVED

Table 183. RTC - IMSC register description: address offset RTC_BASE_ADDR+0x1C

Bit	Field name	Reset	RW	Description
0	WIMSC	0x0	RW	RTC clock watch interrupt enable bit: When set to 0, clears the interrupt mask (default after PORn Reset). The interrupt is disabled. When set to 1, the interrupt for RTC clockwatch interrupt is enabled.
1	TIMSC	0x0	RW	RTC timer interrupt enable bit: When set to 0, sets the mask for RTC timer interrupt (default after PORn reset). The interrupt is disabled. When set to 1, clears this mask and enables the interrupt.
31:2	RESERVED	0x0	RW	RESERVED

Table 184. RTC - RIS register description: address offset RTC_BASE_ADDR+0x20

Bit	Field name	Reset	RW	Description
0	WRIS	0x0	R	RTC clock watch raw interrupt status bit. Gives the raw interrupt state (prior to masking) of the RTC clock watch interrupt.
1	TRIS	0x0	R	RTC timer raw interrupt status bit. Gives the raw interrupt state (prior to masking) of the RTC timer interrupt.
31:2	RESERVED	0x0	R	RESERVED

Table 185. RTC - MIS register description: address offset RTC_BASE_ADDR+0x24.

Bit	Field name	Reset	RW	Description
0	WMIS	0x0	R	RTC clock watch interrupt status bit. Gives the masked interrupt status (after masking) of the RTC clock watch interrupt WINTR.
1	TMIS	0x0	R	RTC timer interrupt status bit. Gives the masked interrupt status (after masking) of the RTC timer interrupt TINTR.
31:2	RESERVED	0x0	R	RESERVED

Table 186. RTC - ICR register description: address offset RTC_BASE_ADDR+0x28

Bit	Field name	Reset	RW	Description
0	WIC	0x0	W	RTC clock watch interrupt clear register bit. Clears the RTC clock watch interrupt WINTR. 0: No effect. 1: Clears the interrupt.
1	TIC	0x0	W	RTC timer interrupt clear register bit. Clears the RTC timer interrupt TINTR. 0: No effect. 1: Clears the interrupt.
31:2	RESERVED	0x0	W	RESERVED

Table 187. RTC – TDR register description: address offset RTC_BASE_ADDR+0x2C

Bit	Field name	Reset	RW	Description
31:0	TDR	0xFFFFFFFF	R	RTC time load value.

Table 188. RTC - TCR register description: address offset RTC_BASE_ADDR+0x30

Bit	Field name	Reset	RW	Description
0	OS	0x0	RW	RTC Timer one shot count. 0: Periodic mode (default). When reaching zero, the RTC timer raises its interrupt and is reloaded from the LD content. 1: One-shot mode. When reaching zero, the RTC timer raise its interrupt and stops.
1	EN	0x0	RW	RTC Timer enable bit. 0: The RTC timer is stopped on the next CLK32K cycle. 1: The RTC timer is enabled on the next CLK32K cycle. When the RTC timer is stopped, the content of the counter is frozen. A read returns the value of the EN bit. This bit set by hardware when the TLR register is written to while the counter is stopped. When the device is active, this bit is cleared by hardware when the counter reaches zero in one-shot mode.
2	S	0x0	RW	RTC Timer self start bit. When written to 1b, each write in a load register or a pattern will set EN to 1b, so, start the counter in the next CLK32K cycle.
3	RESERVED	0x0	RW	RESERVED
10:4	SP	0x0	RW	RTC Timer Pattern size. Number of pattern bits crossed by the pointer. It defines the useful pattern size.

Bit	Field name	Reset	RW	Description
11	CLK	0x0	RW	RTC Timer clock. 0: The RTC timer is clocked by CLK32K. 1: The RTC timer is clocked by the trimmed clock.
12	BYPASS_GATED	0x0	RW	Enable or disable the internal clock gating: 0: The internal clock gating is activated. 1: No clock gating, clock is always enabled.
31:13	RESERVED	0x0	RW	RESERVED

Table 189. RTC – TLR1 register description: address offset RTC_BASE_ADDR+0x34

Bit	Field name	Reset	RW	Description
31:0	TLR1	0x00000000	RW	RTC timer first load value.

Table 190. RTC – TLR2 register description: address offset RTC_BASE_ADDR+0x38

Bit	Field name	Reset	RW	Description
31:0	TLR2	0x00000000	RW	RTC timer second load value.

Table 191. RTC – TPR1 register description: address offset RTC_BASE_ADDR+0x3C

Bit	Field name	Reset	RW	Description
31:0	TPR1	0x00000000	RW	RTC timer pattern register (pattern[31:0]).

Table 192. RTC – TPR2 register description: address offset RTC_BASE_ADDR+0x40

Bit	Field name	Reset	RW	Description
31:0	TPR2	0x00000000	RW	RTC timer pattern register (pattern[63:32]).

Table 193. RTC – TPR3 register description: address offset RTC_BASE_ADDR+0x44

Bit	Field name	Reset	RW	Description
31:0	TPR3	0x00000000	RW	RTC timer pattern register (pattern[95:64]).

Table 194. RTC – TPR4 register description: address offset RTC_BASE_ADDR+0x48

Bit	Field name	Reset	RW	Description
31:0	TPR4	0x00000000	RW	RTC timer pattern register (pattern[127:96]).

Table 195. RTC – TIN register description: address offset RTC_BASE_ADDR+0x4C

Bit	Field name	Reset	RW	Description
31:0	TIN	0x00000000	R	RTC timer interrupt number register.

Note: All RESERVED fields inside registers must always be written with their default value.

3.16 RNG

3.16.1 Introduction

The RNG is a real random number generator based on a continuous analog noise that provides a 16-bit value to the host when read.

3.16.2 Functional description

The peripheral is normally used by the Bluetooth Stack, but the user can read the random value at any time by accessing the register VAL. The RNG peripheral is addressed through the AHB, so the access must be at 32-bit, otherwise hard fault is generated on Cortex M0.

The minimum period between two consecutive random numbers is about 1.25 µs.

3.16.3 RNG registers

RNG peripheral base address (RNG_BASE_ADDR) 0xB0000000

Table 196. RNG registers

Address offset	Name	RW	Reset	Description
0x00	CR	RW	0x00000000	RNG configuration register. Refer to the detailed description below.
0x04	SR	R	0x00000000	RNG status register. Refer to the detailed description below.
0x08	VAL	R	0x00000000	RNG 16-bit random value. Refer to the detailed description below.

Table 197. RNG – CR register description: address offset RNG_BASE_ADDR+0x00

Bit	Field name	Reset	RW	Description
1:0	RESERVED	0x0	RW	RESERVED
2	DIS	0x0	RW	Set the state of the random number generator. • 0: RNG is enable. • 1: RNG is disabled. The internal free-running oscillators are put in power-down mode and the RNG clock is stopped at the input of the block.
31:3	RESERVED	0x00000000	RW	RESERVED

Table 198. RNG – SR register description: address offset RNG_BASE_ADDR+0x04

Bit	Field name	Reset	RW	Description
0	RDY	0x0	R	New random value ready. • 0: The VAL register value is not yet valid. If performing a read access to VAL, the host will be put on hold until a random value is available. • 1: The VAL register contains a valid random number. This bit remains at 0 when the RNG is disabled (RNGDIS bit = 1b in CR)
31:1	RESERVED	0x00000000	R	RESERVED

Table 199. RNG – VAL register description: address offset RNG_BASE_ADDR+0x08

Bit	Field name	Reset	RW	Description
15:0	RANDOM_VALUE	0x0000	R	The 16-bit random value.

Bit	Field name	Reset	RW	Description
31:16	RESERVED	0x0000	R	RESERVED

Note: All RESERVED fields inside registers must always be written with their default value.

3.17 PDM stream processor

The BlueNRG-2 integrates a digital filter for processing PDM stream coming from a digital microphone and inputting into a GPIO pin. The BlueNRG-2 outputs a 0.8MHz or 1.6MHz signal into a GPIO pin for providing the digital microphone with a frequency clock.

3.18 System timer (SysTick)

The BlueNRG-2 includes a system timer (SysTick) that can be polled by software or can be configured to generate an interrupt. SysTick interrupt has its own entry in the vector table and therefore can have its own handler.

3.19 Public key accelerator (PKA)

The public key accelerator is for the computation of cryptographic public key primitives through elliptic curve cryptography (ECC) using a predefined prime modulus and a predefined curve.

3.19.1 PKA functional description

The PKA core is clocked by the system clock divided by two and the PKA memory is clocked by system clock. This peripheral is addressed through the AHB, so the access must be at 32-bit or a hard fault is generated on the Cortex M0.

The PKA works on a 1 kB dedicated RAM block located in 0xC0000400.

The main features of the PKA block are:

- elliptic curve Diffie-Hellman (ECDH) public-private key pair calculation accelerator
- based on the Montgomery method for fast modular multiplications
- built-in Montgomery domain inward and outward transformations
- AMBA AHB lite slave interface with a reduced command set
- single port internal memory available for the system when the BlueNRG-2 PKA is not using it.

The PKA and the PKA RAM are clock gated by default after reset,. so the clock must be enabled in CKGEN_SOC before using PKA functionality.

The input data, output data and data verification result have specific locations in the PKA RAM.

Table 200. PKA RAM data location

Parameter description	PKA RAM offset address	Size (words)
INPUT: ECC K value of kP	0x6C	EOS(1)
INPUT: input point P, coordinate X	0x90	EOS
INPUT: input point P, coordinate Y	0xB4	EOS
OUTPUT: output point P, coordinate X	0x90	EOS
OUTPUT: output point P, coordinate Y	0xB4	EOS
OUTPUT: error value	0x00	1

1. *ECC operand size*

An error value of 1 indicates that input point P does not satisfy the curve equation; in this case, the computation is very short. If the calculation returns an error value of 0, the result is valid.

The maximum length of data is calculated by:

Max. EOS = (max_ecc_size / word_size) + 1

If ECC P256 is used, the PKA core needs an operand of 9 (256/32 + 1) words. When loading a 256-bit (8 word) input, an additional word is requested and must be filled with zero.

The starting point P for the computation is:

- **PX** = (0x6B17D1F2, 0xE12C4247, 0xF8BCE6E5, 0x63A440F2, 0x77037D81, 0x2DEB33A0, 0xF4A13945, 0xD898C296)
- **PY** = (0x4FE342E2, 0xFE1A7F9B, 0x8EE7EB4A, 0x7C0F9E16, 0x2BCE3357, 0x6B315ECE, 0xCBB64068, 0x37BF51F5)

3.19.2 PKA registers

PKA peripheral base address (PKA_BASE_ADDR) 0xC0000000

Table 201. PKA registers

Address offset	Name	RW	Reset	Description
0x00	CSR	RW	0x00000002	Command and status register
0x04	ISR	RW	0x00000000	Interrupt status register
0x08	IEN	RW	0x00000000	Interrupt enable register

Table 202. PKA – CSR register description: address offset PKA_BASE_ADDR+0x00

Bit	Field name	Reset	RW	Description
0	GO	0	W	PKA start processing command: • 0: has no effect. • 1: starts the processing. After this bitfield is set to 1, it must be written back to 0 manually.
1	READY	1	R	PKA readiness status: • 0: the PKA is computing. It is not ready. • 1: the PKA is ready to start a new process. The rising edge of the READY bit set the PROC_END flag in the ISR register.
6:2	RESERVED	0x00	RW	RESERVED
7	SFT_RST	0	W	PKA software reset: • 0: has no effect. • 1: reset the PKA peripheral. After this bitfield is set to 1, it must be written back to 0 manually.
31:8	RESERVED	0x00	RW	RESERVED

Table 203. PKA – ISR register description: address offset PKA_BASE_ADDR+0x04

Bit	Field name	Reset	RW	Description
0	PROC_END	0	RW	PKA process ending interrupt. When read: • 0: no event. • 1: PKA process is ended. When written: • 0: no effect. • 1: clears the PKA process ending interrupt. After this bitfield is set to 1, it must be written back to 0 manually.

Bit	Field name	Reset	RW	Description
1	RESERVED	0	RW	RESERVED
2	RAM_ERR	0	RW	RAM read/write access error interrupt. When read: • 0: all AHB read or write access to the PKA RAM occurred while the PKA was stopped. • 1: All the AHB read or write accesses to the PKA RAM occurred while the PKA was operating and using the internal RAM. These reads or writes could not succeed as the PKA internal RAM is disconnected from the AHB bus when the PKA is operating (READ bit low). When written: • 0: no effect. • 1: clears the RAM access error interrupt. After this bitfield is set to 1, it must be written back to 0 manually.
3	ADD_ERR	0	RW	AHB address error interrupt. When read: • 0: All AHB read or write access to the PKA RAM occurred in a mapped address range. • 1: All the AHB read or write access to the PKA RAM occurred in an unmapped address range. When written: • 0: no effect. • 1: clears the AHB address error interrupt. After this bitfield is set to 1, it must be written back to 0 manually.
31:4	RESERVED	0x00000000	RW	RESERVED

Table 204. PKA – IEN register description: address offset PKA_BASE_ADDR+0x08

Bit	Field name	Reset	RW	Description
0	PROCEND_EN	0	RW	Process ended interrupt enable. • 0: interrupt disabled. • 1: interrupt enabled.
1	RESERVED	0	RW	RESERVED
2	RAMERR_EN	0	RW	RAM access error interrupt enable. • 0: interrupt disabled. • 1: interrupt enabled.
3	ADDERR_EN	0	RW	AHB address error interrupt enable. • 0: interrupt disabled. • 1: interrupt enabled.
31:4	RESERVED	0	RW	RESERVED

Note: All RESERVED fields inside registers must always be written with their default value.

3.20 TX/RX event alert

The BlueNRG-2 is provided with the ANATEST1 (pin 14 for QFN32 package, pin 24 for QFN48 package and pin D4 for WCSP34 package) signal which alerts forthcoming transmission or reception event. The ANATEST1 pin switches to high level before transmission and before reception. Then, it switches to low level at the end of the event. The signal can be used for controlling external antenna switching and supporting coexistence with other wireless technologies.

Note: In this mode, the DIO14 cannot be used and it must be set as default (pull-down input). This is valid for the package QFN32 and WCSP34 only.

3.21 SWD debug feature

The BlueNRG-2 embeds the ARM serial wire debug (SWD) port. It is two pins (clock and single bi-directional data) debug interface, providing all the debug functionality plus real-time access to system memory without halting the processor or requiring any target resident code.

Table 205. SWD port

Pin functionality	Pin name	Pin description
SWCLK	IO9	SWD clock signal
SWDIO	IO10	SWD data signal

The Cortex-M0 subsystem of the BlueNRG-2 embeds four breakpoints and two watchpoints.

3.21.1 Debugging tips

There are certain situations where debug access is disabled and the chip cannot be accessed, including:

- application that disables debug pins
- application that set the device in sleep or standby state, in which the debug port is not powered.

These cases are common during application development and device can end up in a state where debug access is no longer possible. To recover this situation, it is recommended to force IO7 pin high and hardware reset the device in order to force execution of the updater code (see Section 3.23 Pre-programmed bootloader). The user can then connect with SWD interface and erase the device Flash memory.

3.22 Bluetooth low energy radio

The BlueNRG-2 integrates an RF transceiver compliant to the Bluetooth specification and to the standard national regulations in the unlicensed 2.4 GHz ISM band.

The RF transceiver requires very few external discrete components. It provides 96 dB link budgets with excellent link reliability, keeping the maximum peak current below 15 mA.

In transmit mode, the power amplifier (PA) drives the signal generated by the frequency synthesizer out to the antenna terminal through a very simple external network. The power delivered as well as the harmonic content depends on the external impedance seen by the PA.

3.22.1 Radio operating modes

Several operating modes are defined for the BlueNRG-2 radio:

- Reset mode
- Sleep mode
- Active mode
- Radio mode
  - RX mode
  - TX mode

In Reset mode, the BlueNRG-2 is in ultra-low power consumption: all voltage regulators, clocks and the RF interface are not powered. The BlueNRG-2 enters Reset mode by asserting the external Reset signal. As soon as it is de-asserted, the device follows the normal activation sequence to transit to active mode.

In sleep mode either the low speed crystal oscillator or the low speed ring oscillator are running, whereas the high speed oscillators are powered down as well as the RF interface. The state of the BlueNRG-2 is retained and the content of the RAM is preserved.

While in sleep mode, the BlueNRG-2 waits until an internal timer expires and then it goes into active mode.

In active mode the BlueNRG-2 is fully operational: all interfaces, including RF, are active as well as all internal power supplies together with the high speed frequency oscillator. The MCU core is also running.

Radio mode differs from active mode as also the RF transceiver is active and it is capable of either transmitting or receiving.

3.23 Pre-programmed bootloader

BlueNRG-2 device has a pre-programmed bootloader supporting UART protocol with automatic baudrate detection. Main features of the embedded bootloader are:

- Auto baudrate detection up to 460 kbps
- Flash mass erase, section erase
- Flash programming
- Flash readout protection enable/disable

The pre-programmed bootloader is an application which is stored on the BlueNRG-2 internal ROM at manufacturing time by STMicroelectronics. This application allows upgrading the device Flash with a user application using a serial communication channel (UART).

Bootloader is activated by hardware by forcing IO7 high during power-up or hardware Reset, otherwise, application residing in Flash will be launched.

Note: The customer application must ensure that IO7 is forced low during power up. Bootloader protocol is described in a separate application note.

3.24 Unique device serial number

The BlueNRG-2 device has a unique six-byte serial number stored at address 0x100007F4: it is stored as two words (8 bytes) at addresses 0x100007F4 and 0x100007F8 with unique serial number padded with 0xAA55.

4 Pin description

The BlueNRG-2 comes in three package versions: WCSP34 offering 14 GPIOs, QFN32 offering 15 GPIOs and QFN48 offering 26 GPIOs. Figure 22. BlueNRG-2 pin out top view (QFN32) shows the QFN32 pin out, Figure 23. BlueNRG-2 pin out top view (QFN48) shows the QFN48 pin out and Figure 24. BlueNRG-2 ball out top view (WCSP34) shows the WCSP34 ball out.

Figure 21. BlueNRG-2 pin out top view (QFN32)

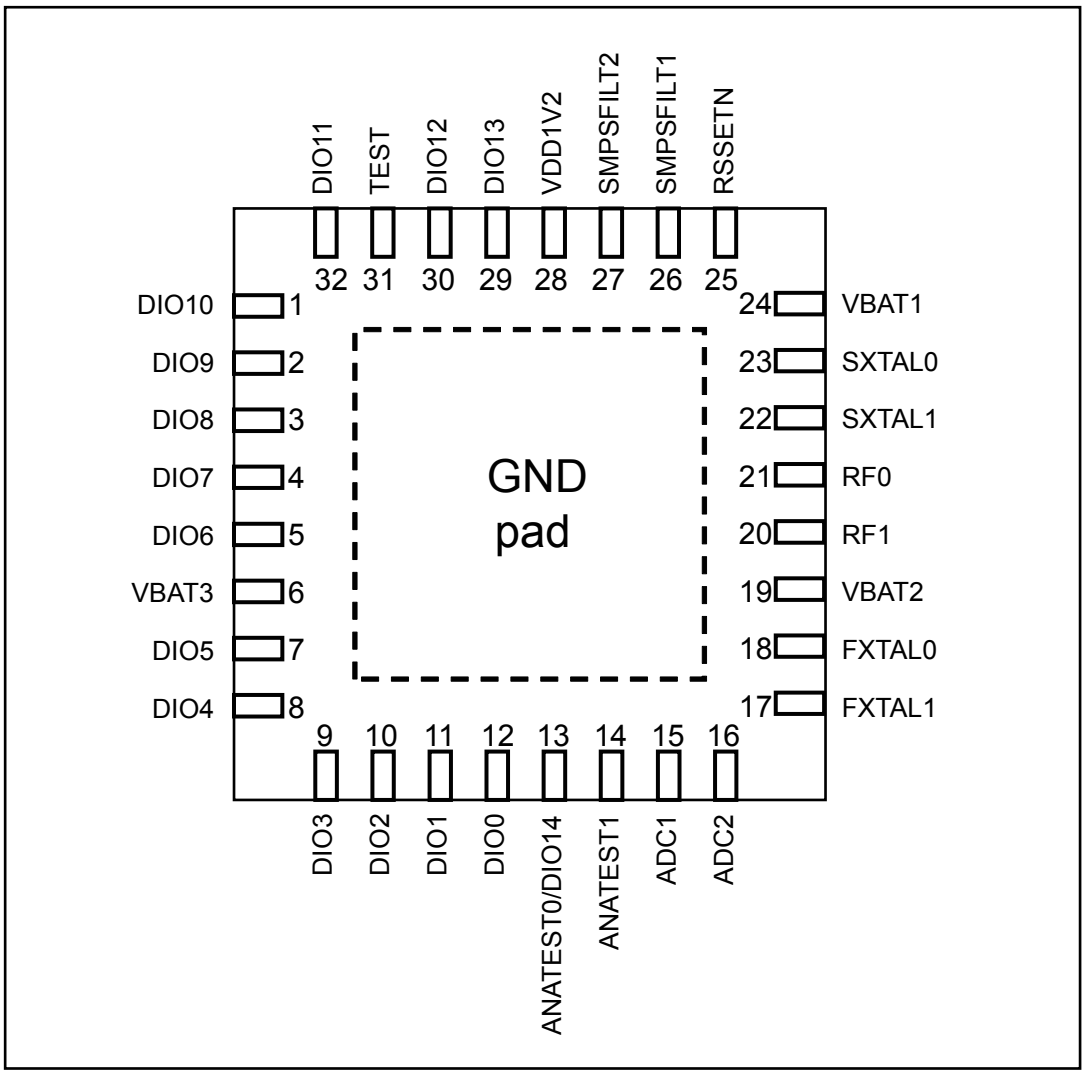


Figure 22. **BlueNRG-2 pin out top view (QFN48)**

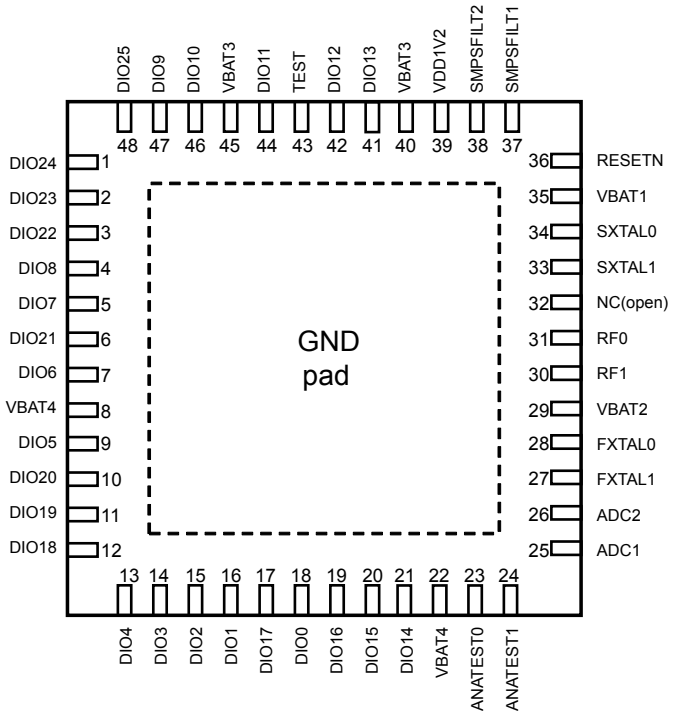


Figure 23. **BlueNRG-2 ball out top view (WCSP34)**

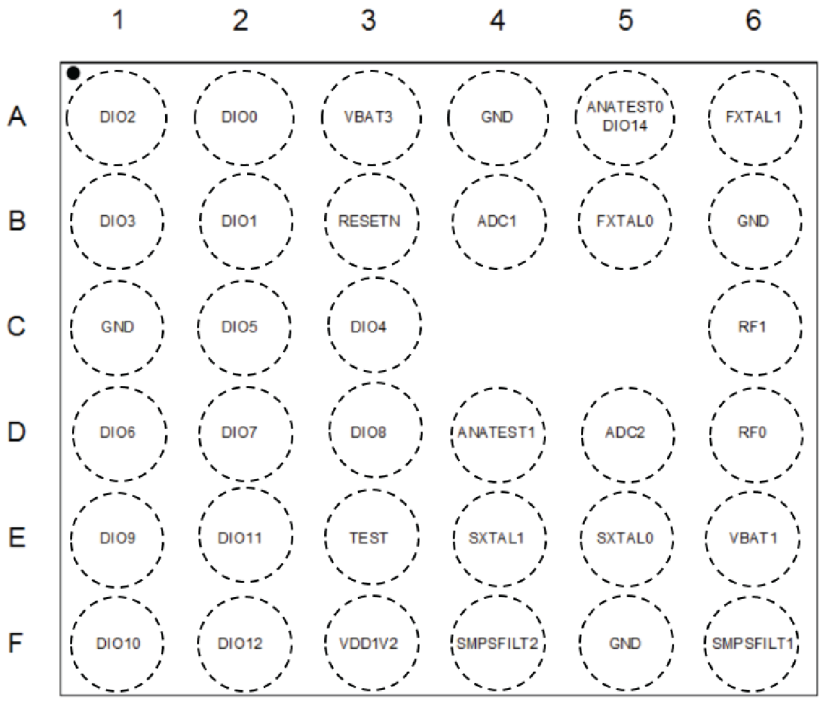


Figure 24. **BlueNRG-2 ball out bottom view (WCSP34)**

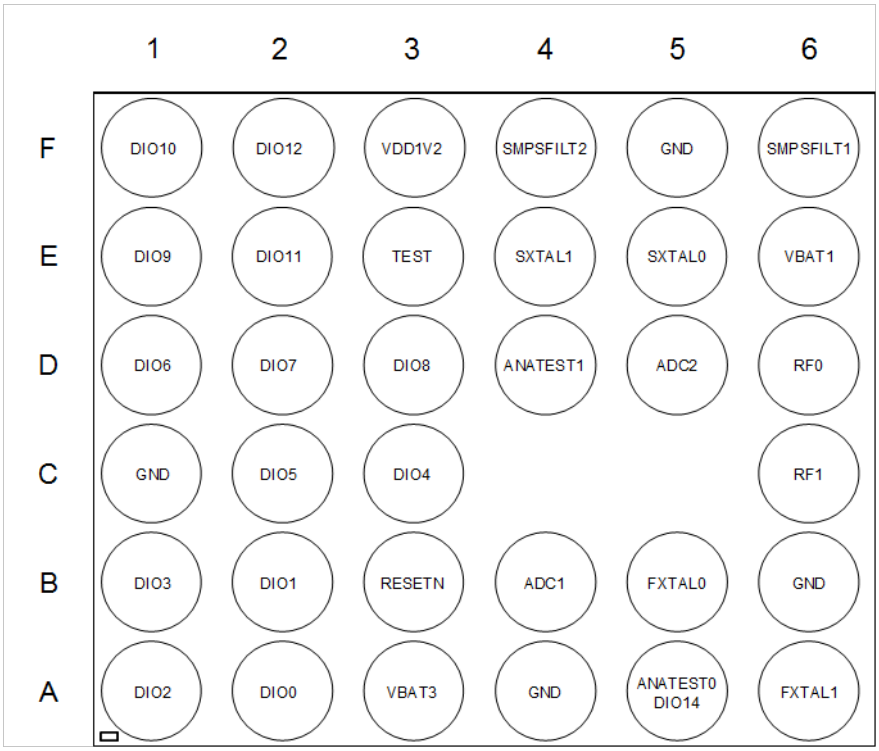


Table 206. **Pinout description**

Pins			Name	I/O	Description
QFN32	QFN48	WCSP34			
1	46	F1	DIO10	I/O	General purpose digital I/O
2	47	E1	DIO9	I/O	General purpose digital I/O
3	4	D3	DIO8	I/O	General purpose digital I/O
4	5	D2	DIO7/BOOT(1)	I/O	Bootloader pin/
					General purpose digital I/O
5	7	D1	DIO6	I/O	General purpose digital I/O
6	40	A3	VBAT3	VDD	Battery voltage input
-	45	-			
7	9	C2	DIO5	I/O	General purpose digital I/O
8	13	C3	DIO4	I/O	General purpose digital I/O
9	14	B1	DIO3	I/O	General purpose digital I/O
10	15	A1	DIO2	I/O	General purpose digital I/O
11	16	B2	DIO1	I/O	General purpose digital I/O
12	18	A2	DIO0	I/O	General purpose digital I/O
13	21	A5	DIO14	I/O	General purpose digital I/O
	23		ANATEST0	O	Analog output

Pins			Name	I/O	Description
QFN32	QFN48	WCSP34			
14	24	D4	ANATEST1	O	Analog output
15	25	B4	ADC1	I	ADC input 1
16	26	D5	ADC2	I	ADC input 2
17	27	A6	FXTAL1	I	16/32 MHz crystal
18	28	B5	FXTAL0	I	16/32 MHz crystal
19	29	-	VBAT2	VDD	Battery voltage input
20	30	C6	RF1	I/O	Antenna + matching circuit connection
21	31	D6	RF0	I/O	Antenna + matching circuit connection
22	33	E4	SXTAL1	I	32 kHz crystal
23	34	E5	SXTAL0	I	32 kHz crystal
24	35	E6	VBAT1	VDD	Battery voltage input
25	36	B3	RESETN	I	System reset
26	37	F6	SMPSFILT1	I	SMPS output to external filter
27	38	F4	SMPSFILT2	I/O	SMPS output to external filter/battery voltage input
28	39	F3	VDD1V2	O	1.2V digital core output
29	41	-	DIO13	I/O	General purpose digital I/O
30	42	F2	DIO12	I/O	General purpose digital I/O
31	43	E3	TEST	I	Test pin put to GND
32	44	E2	DIO11	I/O	General purpose digital I/O
-	-	A4	GND	GND	Ground
-	-	B6			
-	-	C1			
-	-	F5			
-	20	-	DIO15	I/O	General purpose digital I/O
-	19	-	DIO16	I/O	General purpose digital I/O
-	17	-	DIO17	I/O	General purpose digital I/O
-	12	-	DIO18	I/O	General purpose digital I/O
-	11	-	DIO19	I/O	General Purpose Digital I/O
-	10	-	DIO20	I/O	General purpose digital I/O
-	6	-	DIO21	I/O	General Purpose Digital I/O
-	3	-	DIO22	I/O	General purpose digital I/O
-	2	-	DIO23	I/O	General purpose digital I/O
-	1	-	DIO24	I/O	General purpose digital I/O
-	8	-	VBAT4	VDD	Battery voltage input
-	22	-			
-	32	-	Not connected	-	-
-	48	-	DIO25	I/O	General purpose digital I/O

1. *The pin IO7/BOOT is monitored by bootloader after power-up or hardware reset and it should be low to prevent unwanted bootloader activation.*

5 Memory mapping

Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbyte address space.

The bytes are coded in memory in little Endian format. The lowest numbered byte in a word is considered the word's least significant byte and highest numbered byte the most significant.

The addressable memory space is divided into 16 main blocks, each 256 MB. All the memory areas that are not allocated to on-chip memories and peripherals are considered "RESERVED".

For the detailed mapping of an available memory and register areas, please refer to the memory map in table below and to the register lists detailed in each of the peripheral sections.

Table 207. Memory mapping

Address	Cortex-M0 address map	Size	Description
0x0000_0000 – 0x0000_07FF	Code	2 kB	ROM
0x1000_0000 – 0x1000_07FF	Code	2 kB	ROM
0x1004_0000 – 0x1007_FFFF	Code	256 kB	Flash
0x2000_0000 – 0x2000_2FFF(1)	SRAM0 always on	12 kB	SRAM
0x2000_3000 – 0x2000_5FFF	SRAM1 switchable	12 kB	SRAM
0x2000_6000 – 0x3FFF_FFFF			RESERVED
0x4000_0000	APB peripheral	4 kB	GPIO
0x4010_0000		4 kB	Flash controller
0x4020_0000		4 kB	System controller
0x4030_0000		4 kB	UART
0x4040_0000		4 kB	SPI
0x4050_0000		4 kB	RESERVED
0x4060_0000		4 KB	RESERVED
0x4070_0000		4 kB	Watchdog
0x4080_0000		4 kB	ADC
0x4090_0000		4 kB	Clock generator
0x40A0_0000		4 kB	I2C2
0x40B0_0000		4 kB	I2C1 (2)
0x40C0_0000		4 kB	AHB up converter
0x40D0_0000		4 kB	MFT1
0x40E0_0000		4 kB	MFT2
0x40F0_0000		4 kB	RTC
0x4100_0000		4 kB	RESERVED
0x4800_0000		4 kB	BLE controller
0x4810_0000		4 kB	BLE clock generator
0x5000_0000	AHB peripheral	4 kB	RESERVED
0xA000_0000		4 kB	DMA controller
0xB000_0000		4 kB	RNG
0xC000_0000		4 kB	PKA

Address	Cortex-M0 address map	Size	Description
0xC000_0400	AHB peripheral	1 kB	PKA RAM
0xE000_0000 – 0xE00F_FFFF	Private peripheral bus	1 MB	Cortex-M0 registers
0xE010_0000 – 0xEFFF_FFFF	RESERVED	256 MB	RESERVED
0xF000_0000 – 0xFFFF_FFFF		256 MB	RESERVED

1. *0x200000C0-0x200002CB reserved for radio controller.*
2. *The I²C 1 is not available in WLCSP34 package.*

All the peripherals are addressed by APB, except DMA, RNG and PKA peripherals that are addressed by AHB. The peripherals DMA, RNG and PKA that are addressed through the AHB, must be accessed only with 32-bit accesses. Any 8-bit or 16-bit access generates an AHB error leading to a hard fault on Cortex-M0.

6 Application circuit

The schematics below are purely indicative.

Figure 25. Application circuit: active DC-DC converter QFN32 package

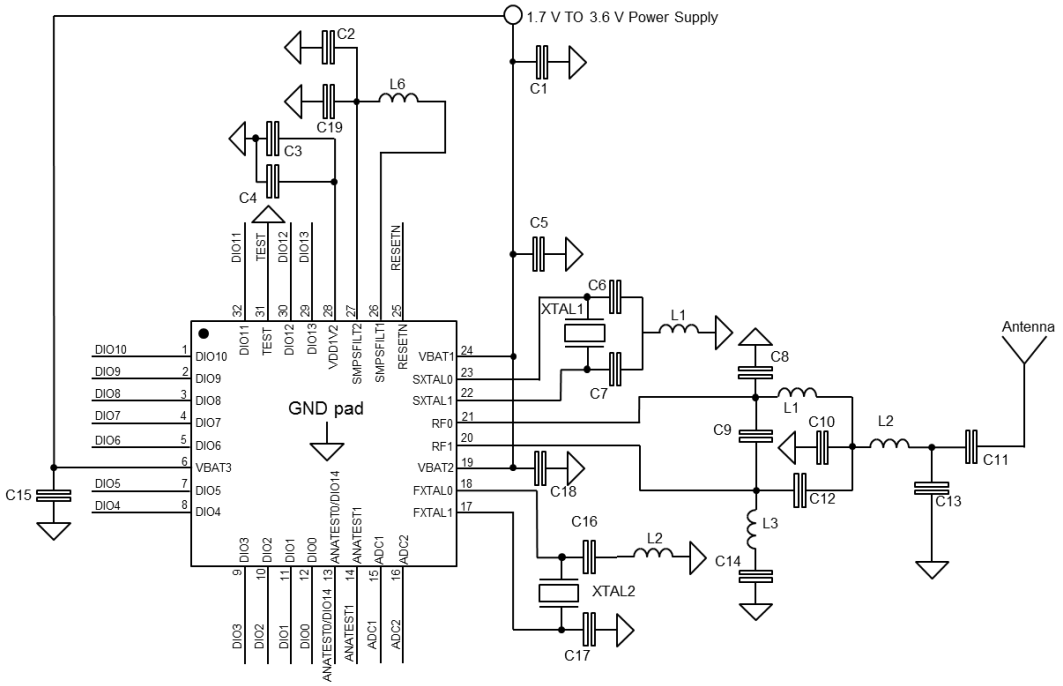


Figure 26. Application circuit: non-active DC-DC converter QFN32 package

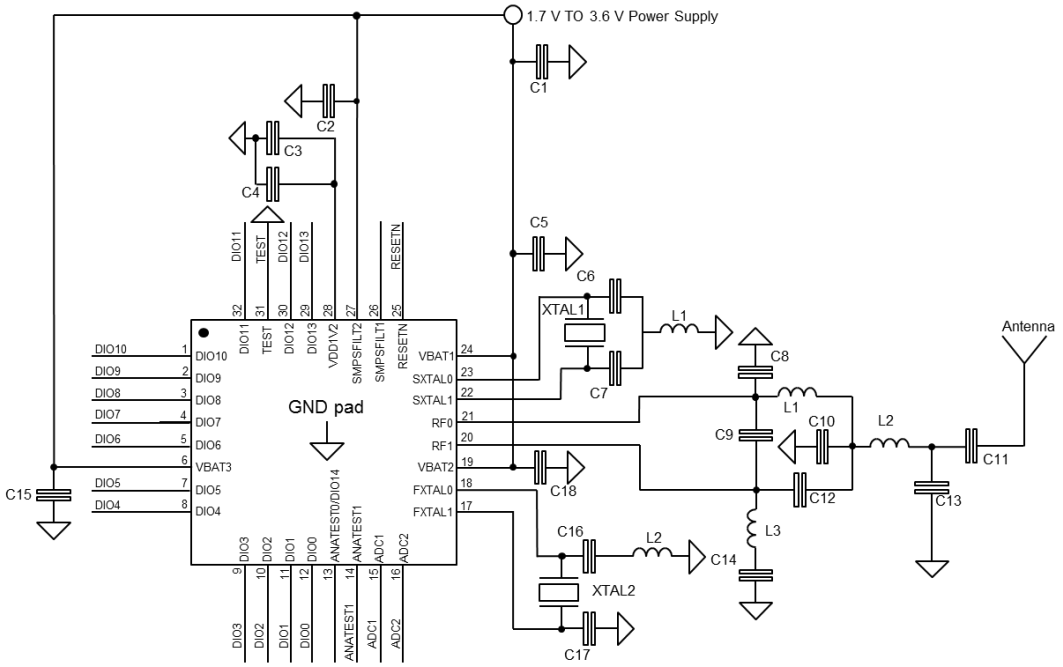


Figure 27. **Application circuit: active DC-DC converter WCSP34 package**

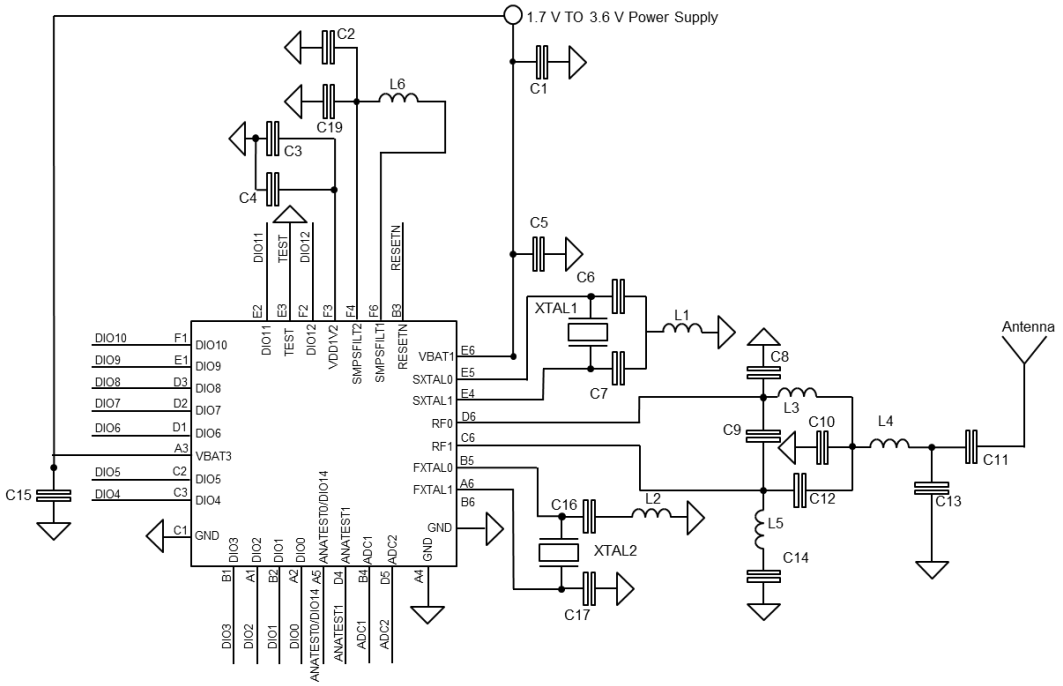


Figure 28. **Application circuit: non active DC-DC converter WCSP34 package**

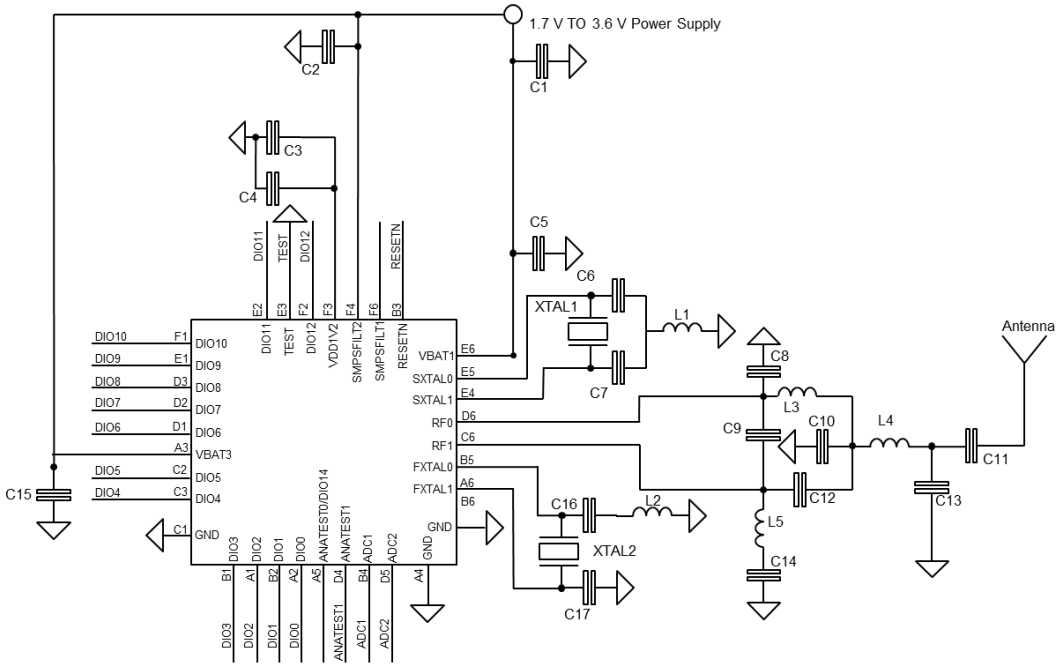


Figure 29. Application circuit: active DC-DC converter QFN32 package with BALF-NRG-02D3 balun

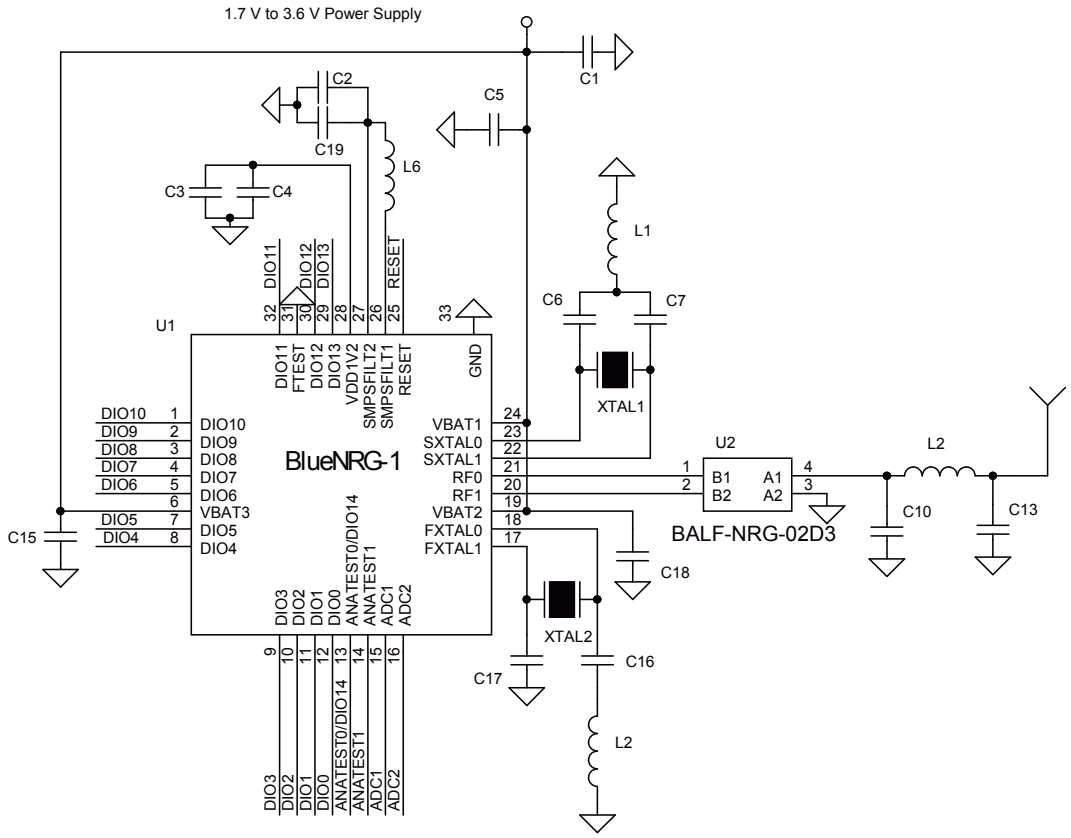


Table 208. External component list

Component	Description
C1	Decoupling capacitor
C2	DC-DC converter output capacitor
C3	Decoupling capacitor for 1.2 V digital regulator
C4	Decoupling capacitor for 1.2 V digital regulator
C5	Decoupling capacitor
C6	32 kHz crystal loading capacitor
C7	32 kHz crystal loading capacitor
C8	RF balun/matching network capacitor
C9	RF balun/matching network capacitor
C10	RF balun/matching network capacitor
C11	RF balun/matching network capacitor
C12	RF balun/matching network capacitor
C13	RF balun/matching network capacitor
C14	RF balun/matching network capacitor
C15	Decoupling capacitor
C16	16/32 MHz crystal loading capacitor
C17	16/32 MHz crystal loading capacitor

Component	Description
C18	Decoupling capacitor
C19	DC-DC converter output capacitor
L1	32 kHz crystal filter inductor
L2	16/32 MHz crystal filter inductor
L3	RF balun/matching network inductor
L4	RF balun/matching network inductor
L5	RF balun/matching network inductor
XTAL1	32 kHz crystal (optional)
XTAL2	16/32 MHz crystal

7 Absolute maximum ratings and thermal data

Table 209. Absolute maximum ratings

Pin	Parameter	Value	Unit
VBAT3, VBAT2, VBAT1, RESETN, SMPSFILT1, SMPSFILT2	DC-DC converter supply voltage input and output	-0.3 to +3.9	V
VDD1V2	DC voltage on linear voltage regulator	-0.3 to +1.3	V
DIO0 to DIO25, TEST	DC voltage on digital input/output pins	-0.3 to +3.9	V
ANATEST0, ANATEST1, ADC1, ADC2	DC voltage on analog pins	-0.3 to +3.9	V
FXTAL0, FXTAL1, SXTAL0, SXTAL1	DC voltage on XTAL pins	-0.3 to +1.4	V
RF0, RF1	DC voltage on RF pins	-0.3 to +1.4	V
TSTG	Storage temperature range	-40 to +125	°C
VESD-HBM	Electrostatic discharge voltage	±2.0	kV

Note: Absolute maximum ratings are those values above which damage to the device may occur. Functional operation under these conditions is not implied. All voltages are referred to GND.

Table 210. Thermal data

Symbol	Parameter	Value	Unit
Rthj-amb	Thermal resistance junction-ambient	34 (QFN32) 50 (WLCSP34)	°C/W
Rthj-c	Thermal resistance junction-case	2.5 (QFN32) 25 (WLCSP34)	°C/W

8 General characteristics

Table 211. Operating conditions

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{BAT}	Operating battery supply voltage	1.7		3.6	V
T_A	Operating Ambient temperature range	-40		+105	°C

9 Electrical specifications

9.1 Electrical characteristics

Characteristics measured over recommended operating conditions unless otherwise specified. Typical value are referred to T_A = 25 °C, V_{BAT} = 3.0 V. All performance data are referred to a 50 Ω antenna connector, via reference design, QFN32 package version.

Table 212. Electrical characteristics

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
Power consumption when DC-DC converter active						
I_{BAT}	Supply current	Reset	–	5	–	nA
		Standby	–	500	–	nA
		Sleep mode: 32 kHz XO ON (24 KB retention RAM)	–	0.9	–	µA
		Sleep mode: 32 kHZ RO ON (24 KB retention RAM)		2.1		
		Active mode: CPU, Flash and RAM on	–	1.9	–	mA
		RX	–	7.7	–	mA
		TX +8 dBm	–	15.1	–	mA
		TX +4 dBm		10.9		
		TX +2 dBm		9		
		TX -2 dBm		8.3		
		TX -5 dBm		7.7		
		TX -8 dBm		7.1		
		TX -11 dBm		6.8		
		TX -14 dBm		6.6		
Power consumption when DC-DC converter not active						
I_{BAT}	Supply current	Reset	–	5	–	nA
		Standby	–	500	–	nA
		Sleep mode: 32 kHz XO ON (24 KB retention RAM)	–	0.9	–	µA
		Sleep mode: 32 kHZ RO ON (24 KB retention RAM)	–	2.1	–	
		Active mode: CPU, Flash and RAM on	–	3.3	–	mA
I_{BAT}	Supply current	RX	–	14.5	–	mA
		TX +8 dBm	–	28.8	–	mA
		TX +4 dBm		20.5		
		TX +2 dBm		17.2		
		TX -2 dBm		15.3		
		TX -5 dBm		14		
		TX -8 dBm		13		
		TX -11 dBm		12.3		
		TX -14 dBm		12		

Table 213. Digital I/O specifications

Symbol	Test conditions	Min.	Typ.	Max.	Unit
T(RST)L			1.5		ms
TC			3.3		V
TC1			2.5		V
TC2			1.8		V
VIL				0.3*VDD	V
VIH		0.65*VDD			V
VOL	IOL = 3 mA			0.4	V
VOH	IOH = 3 mA	0.7*VDD			V
IOL (low drive strength)	TC (VOL = 0.4 V)		5.6		mA
	TC1 (VOL= 0.42 V)		6.6		mA
	TC2 (VOL =0.45 V)		3		mA
IOL (high drive strength)	TC (VOL = 0.4 V)		11.2		mA
	TC1 (VOL= 0.42 V)		13.2		mA
	TC2 (VOL =0.45 V)		6		mA
IOL (Very high drive strength)	TC (VOL = 0.4 V)		16.9		mA
	TC1 (VOL= 0.42 V)		19.9		mA
	TC2 (VOL =0.45 V)		9.2		mA
IOH (low drive strength)	TC (VOH =2.4 V)		10.6		mA
	TC1 (VOH = 1.72 V)		7.2		mA
	TC2 (VOH = 1.35 V)		3		mA
IOH (high drive strength)	TC (VOH = 2.4 V)		19.2		mA
	TC1 (VOH = 1.72 V)		12.9		mA
	TC2 (VOH = 1.35 V)		5.5		mA
IOH (very high drive strength)	TC (VOH = 2.4 V)		29.4		mA
	TC1 (VOH = 1.72 V)		19.8		mA
	TC2 (VOH = 1.35 V)		8.4		mA
IPUD (current sourced/sinked from IOs with pull enabled)	Static supply 1.7 V	5		10	µA
	Static supply 3.6 V	40		60	µA

9.1.1 Peripheral current consumption

Table 214. Peripheral current consumption

Peripheral	Typical consumption V_{DD} = 3.0 V, T_A = 25 °C	Unit
GPIO	11.0	µA
Flash controller	6.0	
System controller	0.75	
UART	77.0	
SPI	41.0	
Watchdog	4.0	

Peripheral	Typical consumption V_{DD} = 3.0 V, T_A = 25 °C	Unit
ADC	5.0	µA
I2C1	92.0	
I2C2	92.0	
MFT1	7.5	
MFT2	7.5	
RTC	7.5	
DMA	16.5	
RNG	25.0	
PKA	26.0	

Note: The values are calculated as the increment to the current consumption when the peripheral is activated. The peripheral is activated if the related clock is provided.

9.2 RF general characteristics

Characteristics measured over recommended operating conditions unless otherwise specified. Typical value are referred to T_A= 25 °C, V_{BAT} =3.0 V. All performance data are referred to a 50 Ω antenna connector, via reference design, QFN32 package version.

Table 215. RF general characteristics

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
FREQ	Frequency range		2400	–	2483.5	MHz
FCH	Channel spacing		–	2	–	MHz
RFch	RF channel center frequency		2402	–	2480	MHz

9.3 RF transmitter characteristics

Characteristics measured over recommended operating conditions unless otherwise specified. Typical value are referred to T_A = 25 °C, V_{BAT} = 3.0 V. All performance data are referred to a 50 Ω antenna connector, via reference design, QFN32 package version.

Table 216. RF transmitter characteristics

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
MOD	Modulation scheme		GFSK			
BT	Bandwidth-bit period product		–	0.5	–	
Mindex	Modulation index		–	0.5	–	
DR	Air data rate		–	1	–	Mbps
PMAX	Maximum output power	At antenna connector	–	+8	+10	dBm
PRFC	Minimum output power		–	-16.5	–	dBm
PBW1M	6 dB bandwidth for modulated carrier (1 Mbps)	Using resolution bandwidth of 100 kHz	500	–	–	kHz
PRF1	1st adjacent channel transmit power 2 MHz	Using resolution bandwidth of 100 kHz and average detector	–	-35	–	dBm

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
PRF2	2nd Adjacent channel transmit power >3 MHz	Using resolution bandwidth of 100 kHz and average detector	–	-40	–	dBm
ZLOAD	Optimum differential load	@ 2440 MHz	–	25.4 + j20.8 (1)	–	Ω

1. *Simulated value.*

9.4 RF receiver characteristics

Characteristics measured over recommended operating conditions unless otherwise specified. Typical value are referred to T_A = 25 °C, V_{BAT} =3.0 V. All performance data are referred to a 50 Ω antenna connector, via reference design, QFN32 package version.

Table 217. RF receiver characteristics

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
RXSENS	Sensitivity	BER <0.1%		-88		dBm
PSAT	Saturation	BER <0.1%		11		dBm
zIN	Input differential impedance	@ 2440 MHz		25.5-j14.2		Ω
RF selectivity with BLE equal modulation on interfering signal						
C/ICO-channel	Co-channel interference	Wanted signal = -67 dBm, BER ≤ 0.1%		6		dBc
C/I1 MHz	Adjacent (+1 MHz) interference	Wanted signal = -67 dBm, BER≤0.1%		0		dBc
C/I2 MHz	Adjacent (+2 MHz) interference	Wanted signal = -67 dBm, BER ≤ 0.1%		-40		dBc
C/I3 MHz	Adjacent (+3 MHz) interference	Wanted signal = -67 dBm, BER ≤ 0.1%		-47		dBc
C/I≥4 MHz	Adjacent (≥ ± 4 MHz) interference	Wanted signal = -67 dBm, BER ≤ 0.1%		-46		dBc
C/I≥6 MHz	Adjacent (≥ ± 6 MHz) interference	Wanted signal= -67 dBm BER ≤ 0.1%		-48		dBc
C/I≥25 MHz	Adjacent (≥ ±25 MHz) interference	Wanted signal= -67 dBm, BER ≤ 0.1%		-70		dBc
C/IImage	Image frequency interference -2 MHz	Wanted signal = -67 dBm, BER ≤ 0.1%		-16		dBc
C/IImage±1 MHz	Adjacent (±1 MHz) interference to in-band image frequency -1 MHz -3 MHz	Wanted signal = -67 dBm, BER ≤ 0.1%		0 -23		dBc
Intermodulation characteristics (CW signal at f_1, BLE interfering signal at f_2)						
P_IM(3)	Input power of IM interferes at 3 and 6 MHz distance from wanted signal	Wanted signal = -64 dBm, BER ≤ 0.1%		-34		dBm
P_IM(-3)	Input power of IM interferes at -3 and -6 MHz distance from wanted signal	Wanted signal = -64 dBm, BER ≤ 0.1%		-48		dBm
P_IM(4)	Input power of IM interferes at ±4 and ±8 MHz distance from wanted signal	Wanted signal = -64 dBm, BER ≤ 0.1%		-34		dBm
P_IM(5)	Input power of IM interferes at ±5 and ±10 MHz distance from wanted signal	Wanted signal = -64 dBm, BER ≤ 0.1%		-34		dBm

9.5 High speed crystal oscillator characteristics

Characteristics measured over recommended operating conditions unless otherwise specified. Typical value are referred to T_A = 25 °C, V_{BAT} = 3.0 V.

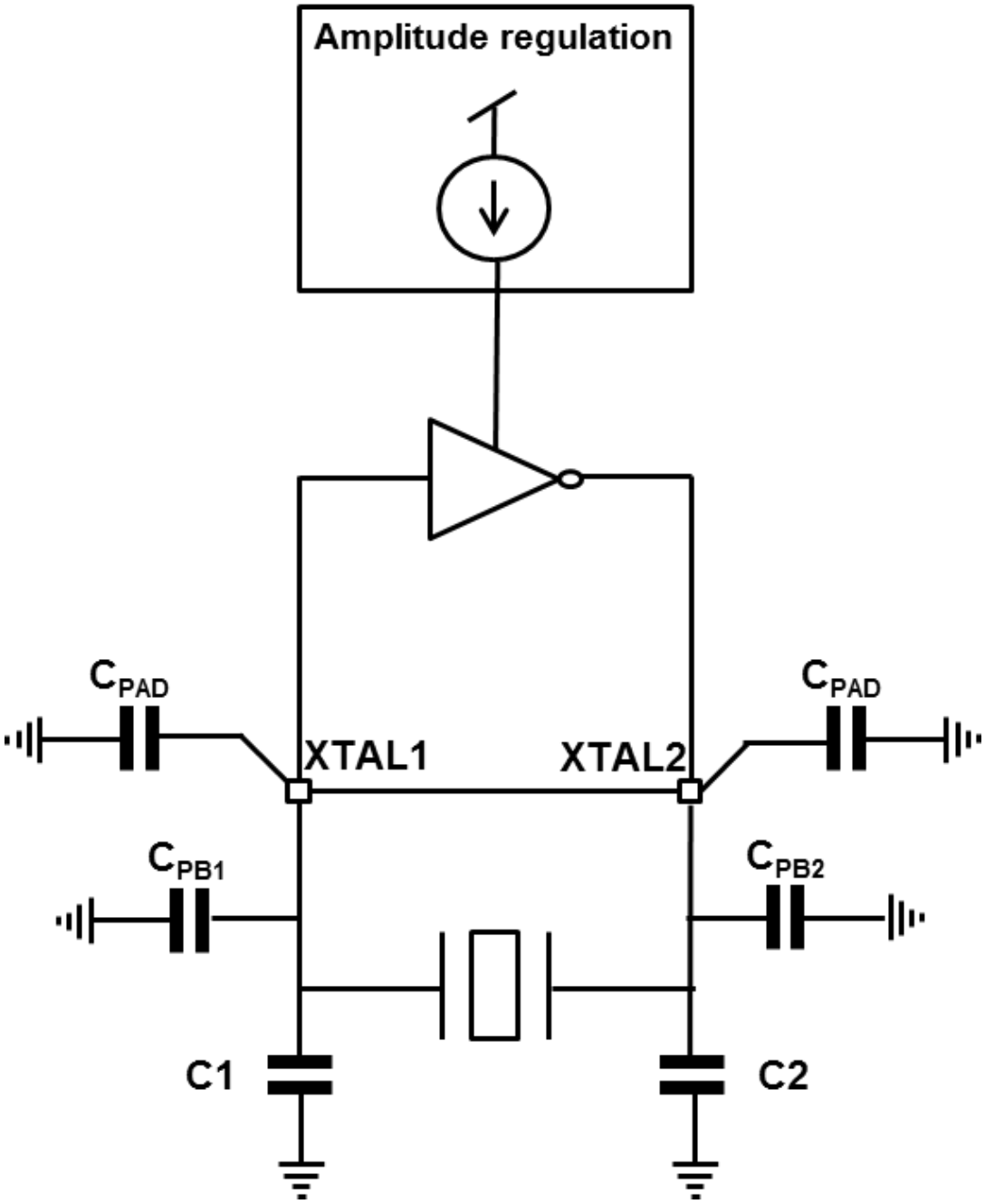
Table 218. High speed crystal oscillator characteristics

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
fNOM	Nominal frequency		–	16/32	–	MHz
fTOL	Frequency tolerance	Includes initial accuracy, stability over temperature, aging and frequency pulling due to incorrect load capacitance	–	–	±50	ppm
ESR	Equivalent series resistance		–	–	100	Ω
PD	Drive level		–	–	100	µW

9.5.1 High speed crystal oscillator

The BlueNRG-2 includes a fully integrated low power 16/32 MHz Xtal oscillator with an embedded amplitude regulation loop. In order to achieve low power operation and good frequency stability of the XTAL oscillator, certain considerations with respect to the quartz load capacitance C0 need to be taken into account.
Figure 31. High speed oscillator block diagram shows a simplified block diagram of the amplitude regulated oscillator used on the BlueNRG-2.

Figure 30. **High speed oscillator block diagram**



Low power consumption and fast startup time is achieved by choosing a quartz crystal with a low load capacitance C0. A reasonable choice for capacitor C0 is 12 pF. To achieve good frequency stability, the following equation needs to be satisfied:

$$C_0 = \frac{C_1' * C_2'}{C_1 + C_2} \quad (6)$$

Where C_1'=C_1+C_{PCB1}+C_{PAD}, C_2'= C_2+C_{PCB2}+C_{PAD}, where C_1 and C_2 are external (SMD) components, C_{PCB1} and C_{PCB2} are PCB routing parasites and C_{PAD} is the equivalent small-signal pad-capacitance. The value of C_{PAD} is around 0.5 pF for each pad. The routing parasites should be minimized by placing quartz and C_1/C_2 capacitors close to the chip, not only for an easier matching of the load capacitance C_0, but also to ensure robustness against noise injection. Connect each capacitor of the Xtal oscillator to ground by a separate vias.

9.6 Low speed crystal oscillator characteristics

Characteristics measured over recommended operating conditions unless otherwise specified. Typical value are referred to T_A = 25 °C, V_{BAT} =3.0 V.

Table 219. Low speed crystal oscillator characteristics

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
fNOM	Nominal frequency		–	32.768	–	kHz
fTOL	Frequency tolerance	Includes initial accuracy, stability over temperature, aging and frequency pulling due to incorrect load capacitance.	–	–	±50	ppm
ESR	Equivalent series resistance		–	–	90	kΩ
PD	Drive level		–	–	0.1	µW

Note: These values are the correct ones for NX3215SA-32.768 kHz-EXS00A-MU00003.

9.7 High speed ring oscillator characteristics

Characteristics measured over recommended operating conditions unless otherwise specified. Typical value are referred to T_A= 25 °C, V_{BAT} =3.0 V.

Table 220. High speed ring oscillator characteristics

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
f_{NOM}	Nominal frequency		–	14	–	MHz

9.8 Low speed ring oscillator characteristics

Characteristics measured over recommended operating conditions unless otherwise specified. Typical value are referred to T_A = 25 °C, V_{BAT} =3.0 V, QFN32 package version.

Table 221. Low speed ring oscillator characteristics

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
32 kHz ring oscillator (LSROSC)						
f_{NOM}	Nominal frequency		–	32	–	kHz

9.9 N-fractional frequency synthesizer characteristics

Characteristics measured over recommended operating conditions unless otherwise specified. Typical value are referred to T_A = 25 °C, V_{BAT} =3.0 V, f_c = 2440 MHz.

Table 222. N-Fractional frequency synthesizer characteristics

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
PNSYNTH	RF carrier phase noise	At ±1 MHz offset from carrier	–	-113	–	dBc/Hz
		At ±3 MHz offset from carrier	–	-119	–	dBc/Hz
LOCKTIME	PLL lock time		–	–	40	µs
TOTIME	PLL turn-on / hop time	Including calibration	–	–	150	µs

9.10 Auxiliary block characteristics

Characteristics measured over recommended operating conditions unless otherwise specified. Typical values are referenced to T_A = 25 °C, V_{BAT} =3.0 V, $f_{\_ADCclk}$ =1 MHz. QFN32 package version.

Table 223. Auxiliary block characteristics

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
Analog-to-digital converter (ADC)						
V_{DDA}	Analog supply voltage		1.7	3.0	3.6	V
$I_{DDA, AVG}$	Analog supply current	Average current during conversion	–	–	0.55	mA
$V_{INP, iNN}$	Input pin voltage	With input attenuator	-50 mV	–	(V_{BAT}+50 mV) / input attenuation	V
SNR Diff	Signal-to-noise ratio	Differential input, with OSR = 200, PGA=0 dB. Sinewave with VinDC=0.6 V, Vpeak diff = 0.85 V, Fin = 1 kHz	–	74	–	dB
SNR SE 1	Signal-to-noise ratio	Single-ended input, with VREF = 0.6 V, OSR = 200, PGA=0 dB. Sinewave with VinDC=0.6 V, Vpeak = 0.425 V, Fin = 1 kHz	–	70	–	dB
ENOB Diff	Effective number of bits	Differential input, OSR = 200, PGA=0 dB. Sinewave with VinDC=0.6 V, Vpeak diff = 0.85 V, Fin = 1 kHz	–	12	–	bit
ENOB SE 1a	Effective number of bits	Single-ended input, with VREF = 0.6 V, with OSR = 200, PGA=0 dB. Sinewave with VinDC=0.6 V, Vpeak = 0.425 V, Fin = 1 kHz	–	8.5	–	bit
ENOB SE 1b	Effective number of bits	Single-ended input, with VREF = 0.6 V, OSR = 200, PGA=0 dB. Sinewave with VinDC=0.6 V, Vpeak = 0.15 V, Fin = 1 kHz	–	9.5	–	bit
Analog temperature sensor						
TRANGE	Temperature range		-40	–	+105	°C
TERR	Error in temperature		-4	0	+4	°C
Battery sensor						
VBLTRANGE	Battery level indicator range		1.8		3.6	V
VBLTERR	Battery level indicator error	After calibration	-150		150	mV
Brown-out reset (BOR)						
VABOR	Ascending brown-out threshold		–	1.68	1.7	V
VDBOR	Descending brown-out threshold		1.62	1.645	–	V

10 Package information

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK packages, depending on their level of environmental compliance. ECOPACK specifications, grade definitions and product status are available at: www.st.com. ECOPACK is an ST trademark.

10.1 QFN32 package information

Figure 31. **QFN32 (5 x 5 x 1 pitch 0.5 mm) package outline**

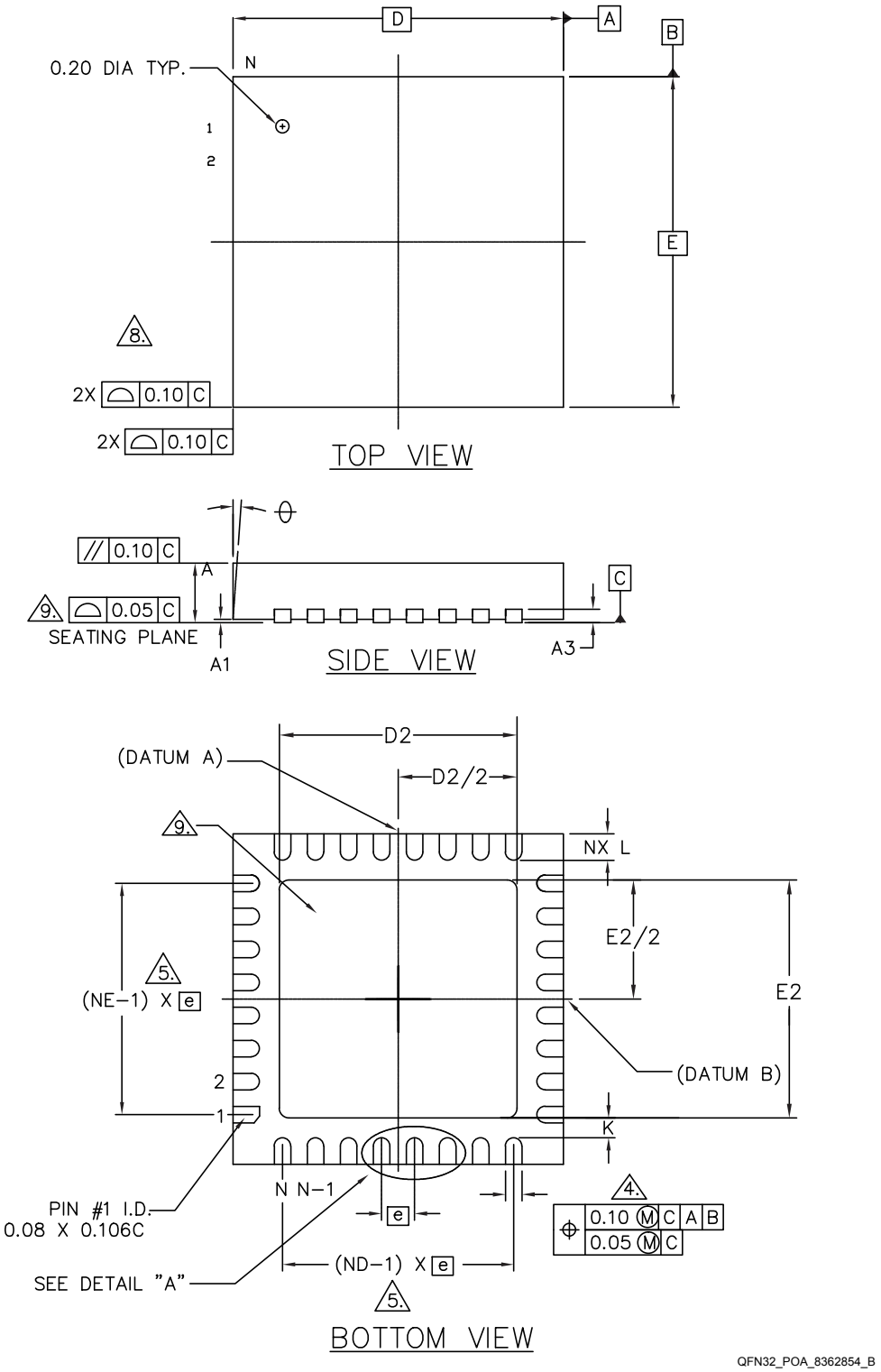
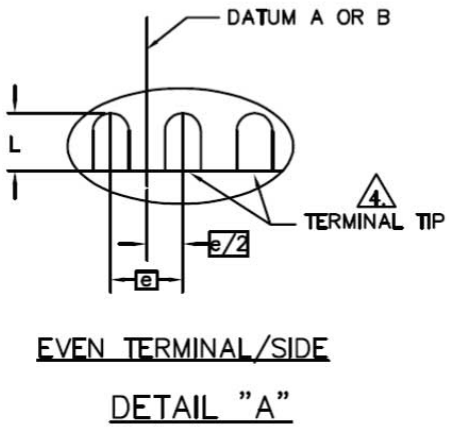


Table 224. **QFN32 (5 x 5 x 1 pitch 0.5 mm) mechanical data**

Dim.	mm		
	Min.	Typ.	Max.
A	0.80	0.85	1.00
A1	0	0.02	0.05
A3		0.20 REF	
b	0.18	0.25	0.30
D		5.00 BSC	
E		5.00 BSC	
D2	3.2		3.70
E2	3.2		3.70
e		0.5 BSC	
L	0.30	0.40	0.50
Φ	0°		14°
K	0.20		

Figure 32. **QFN32 (5 x 5 x 1 pitch 0.5 mm) package detail "A"**



10.2 QFN48 package information

Figure 33. **QFN48 (|6 x 6 x 0.85 pitch 0.4 mm) package outline**

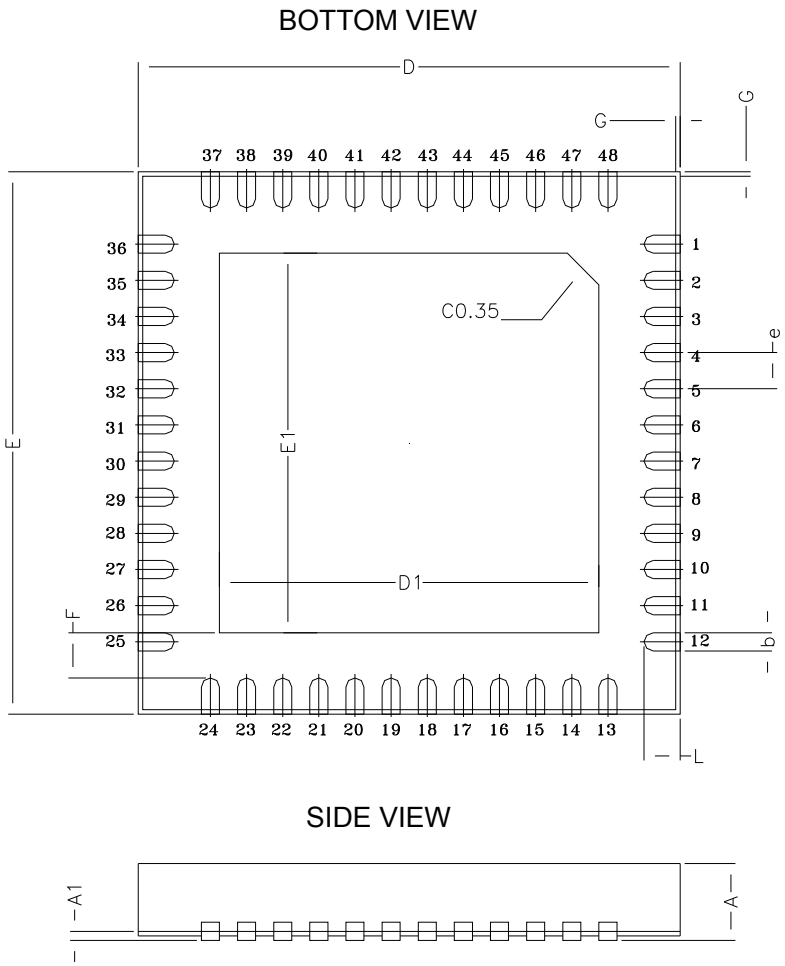


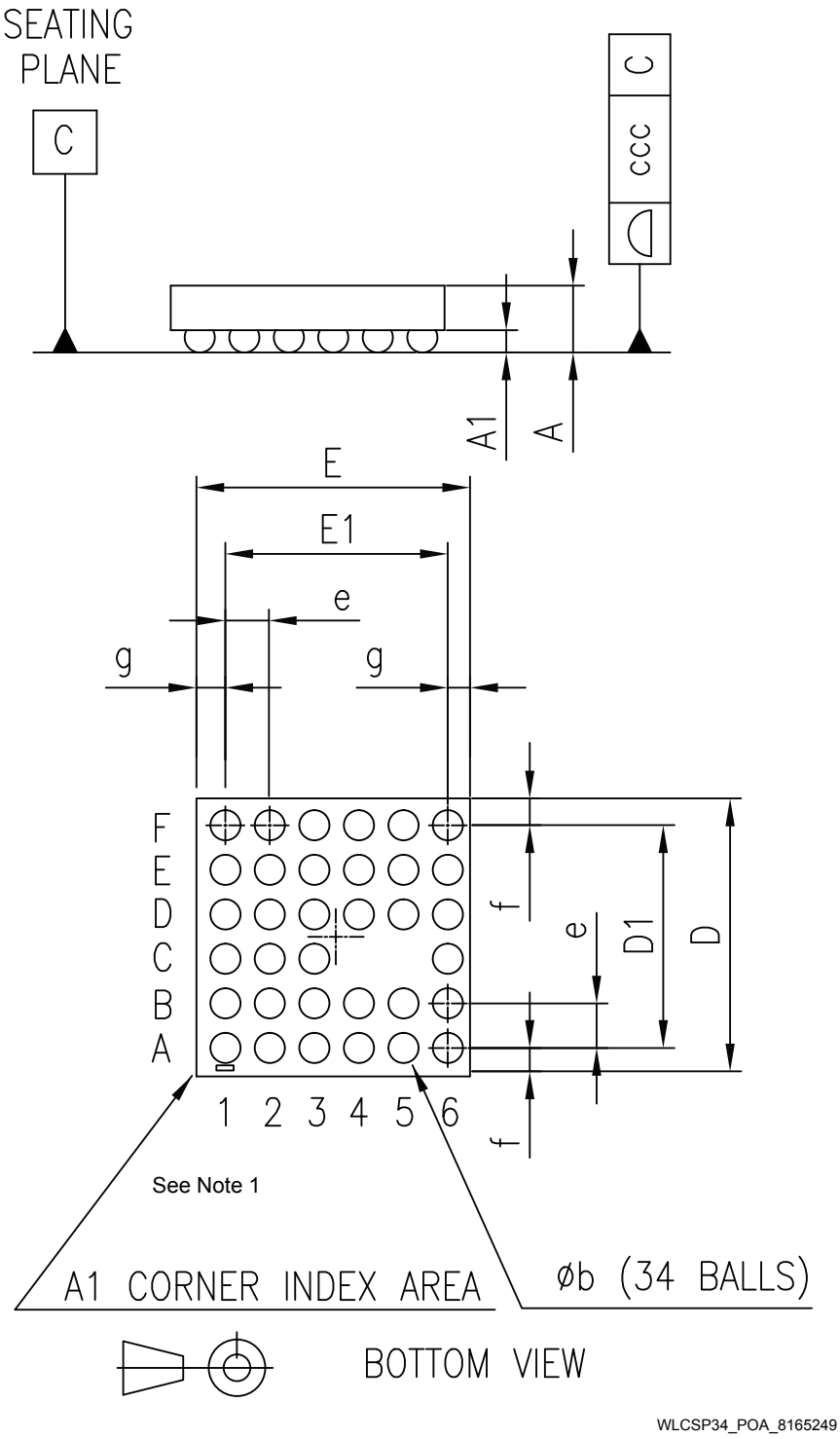
Table 225. **QFN48 (|6 x 6 x 0.85 pitch 0.4 mm) mechanical data**

Dim.	mm		
	Min.	Typ.	Max.
A	0.8	0.85	0.9
A1		0.1	
D	5.9	6	6.1
D1	4.1	4.2	4.3
E	5.9	6	6.1
E1	4.1	4.2	4.3
e		0.4	
F		0.5	
G		0.05	

Dim.	mm		
	Min.	Typ.	Max.
b	0.15	0.2	0.25
L	0.3	0.4	0.5

10.3 WLCSP34 package information

Figure 34. **WLCSP34 (2.66 x 2.56 x 0.5 pitch 0.4 mm) package outline**



1. The corner of terminal A1 must be identified on the top surface by using a laser marking dot.

Table 226. WLCSP34 (2.66 x 2.56 x 0.5 pitch 0.4 mm) mechanical data

Dim.	mm.			Notes
	Min.	Typ.	Max.	
A			0.50	
A1		0.20		
b		0.27		(1)
D	2.50	2.56	2.58	(2)
D1		2.00		
E	2.60	2.66	2.68	(3)
E1		2.00		
e		0.40		
f		0.28		
g		0.33		
ccc			0.05	

1. *The typical ball diameter before mounting is 0.25 mm.*
2. *D = f + D1 + f.*
3. *E = g + E1 + g.*

11 PCB assembly guidelines

For Flip Chip mounting on the PCB, STMicroelectronics recommends the use of a solder stencil aperture of 330 x 330 µm maximum and a typical stencil thickness of 125 µm.

Flip Chips are fully compatible with the use of near eutectic 95.8% Sn, 3.5% Ag, 0.7% Cu solder paste with no-clean flux. ST's recommendations for Flip-Chip board mounting are illustrated on the soldering reflow profile shown in Figure 36. Flip Chip CSP (2.71 x 2.58 x 0.5 pitch 0.4 mm) package reflow profile recommendation.

Figure 35. Flip Chip CSP (2.71 x 2.58 x 0.5 pitch 0.4 mm) package reflow profile recommendation

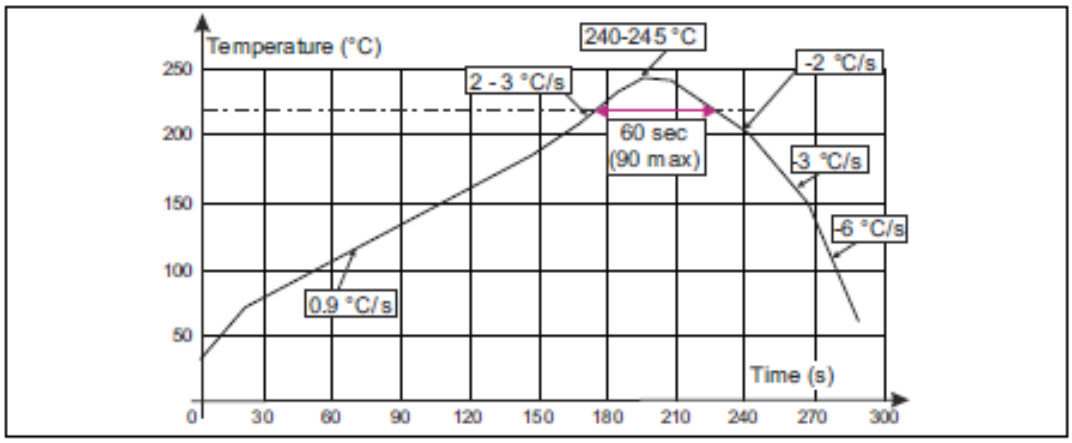


Table 227. Flip Chip CSP (2.71 x 2.58 x 0.5 pitch 0.4 mm) package reflow profile recommendation

Profile	Value	
	Typ.	Max.
Temp. gradient in preheat (T = 70 - 180 °C/s	0.9 °C/s	3 °C/s
Temp. gradient (T = 200 - 225 °C)	2 °C/s	3 °C/s
Peak temp. in reflow	240 - 245 °C	260 °C
Time above 200 °C	60 s	90 s
Temp. gradient in cooling	-2 to -3 °C	-6 °C/s
Time from 50 to 220 °C	160 to 220 °C	

Dwell time in the soldering zone (with temperature higher than 220 °C) has to be kept as short as possible to prevent component and substrate damage. Peak temperature must not exceed 260 °C. Controlled atmosphere (N2 or N2H2) is recommended during the whole reflow, especially above 150 °C.

Flip Chips are able to withstand three times the previous recommended reflow profile to be compatible with a double reflow when SMDs are mounted on both sides of the PCB plus one additional repair.

A maximum of three soldering reflows are allowed for these lead-free packages (with repair step included).

The use of a no-clean paste is highly recommended to avoid any cleaning operation. To prevent any bump cracks, ultrasonic cleaning methods are not recommended.

12 Ordering information

Table 228. **Ordering information**

Order code	Package	Packing
BlueNRG-232	QFN32 (5x5 mm)	Tape and reel
BlueNRG-248	QFN48 (6x6 mm)	
BlueNRG-234	WLCSP34	

Revision history

Table 229. **Document revision history**

Date	Version	Changes
07-Jun-2017	1	Initial release.
16-Nov-2017	2	Updated features in cover page and Section 2 BlueNRG-2 Bluetooth low energy stack.
14-Feb-2018	3	Minor text changes throughout the document.
26-Feb-2018	4	Updated Figure 31. Application circuit: active DC-DC converter QFN32 package with BALFNRG-02D3 balun.
26-Jun-2018	5	Minor text changes throughout the document.
29-Apr-2020	6	Updated Table 207. Memory mapping, Table 16. CKGEN_SOC - CLOCK_EN register description: address offset CKGEN_SOC_BASE_ADDR+0x20, Table 28. Impedance of the ADC pin, Table 31. ADC - CTRL register description: address offset ADC_BASE_ADDR+0x00, Table 130. IO functional map, Table 132. Pin characteristics, Table 153. MFT IO functions, Table 209. Absolute maximum ratings, Table 213. Digital I/O specifications, Table 214. Peripheral current consumption, Table 223. Auxiliary block characteristics. Updated Section 3.3 Memories, Section 3.4.1.2 Active state, Section 3.6.2.4 ADC conversion, Section 3.12.2.2 GPIO characteristics, Section 3.14.2 Functional description, Section 3.15.3 RTC registers Updated Figure 10. ADC block diagram, Figure 21. MFT mode 4 block diagram, Figure 27. Application circuit: non-active DC-DC converter QFN32 package.

Contents

List of figures

List of tables