Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

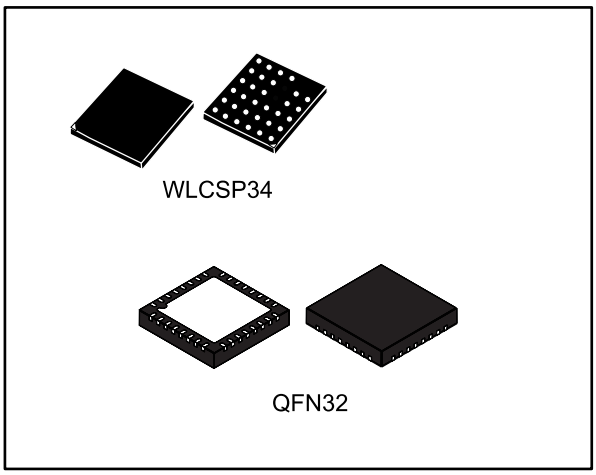
Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

BlueNRG-1

Bluetooth® low energy wireless system-on-chip

Datasheet - production data



Features

- Bluetooth specification compliant master, slave and multiple roles simultaneously, single-mode Bluetooth low energy system-on-chip
- 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 160 KB Flash
- 24 KB RAM with retention (two 12 KB banks)
- 1 x UART interface
- 1 x SPI interface
- 2 x I²C interface
- 14 or 15 GPIO
- 2 x multifunction timer
- 10-bit ADC
- Watchdog & RTC
- DMA controller
- PDM stream processor
- 16 or 32 MHz crystal oscillator
- 32 kHz crystal oscillator
- 32 kHz ring oscillator
- Battery voltage monitor and temperature sensor
- 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)
- 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, WCSP34 package option
- Dedicated wettable flank QFN package for automotive grade qualification

Applications

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

Table 1: Device summary table

Order code	Package	Packing
BLUENRG-132	QFN32 (5 x 5 mm)	Tape and reel
BLUENRG-134	WLCSP34	Tape and reel
BLUENRG-132Y	QFN32 (5 x 5 mm) Automotive grade level	Tape and reel

Contents

List of tables

List of figures

1 Description

The BlueNRG-1 is a very low power Bluetooth low energy (BLE) single-mode system-on-chip, compliant with Bluetooth specification.

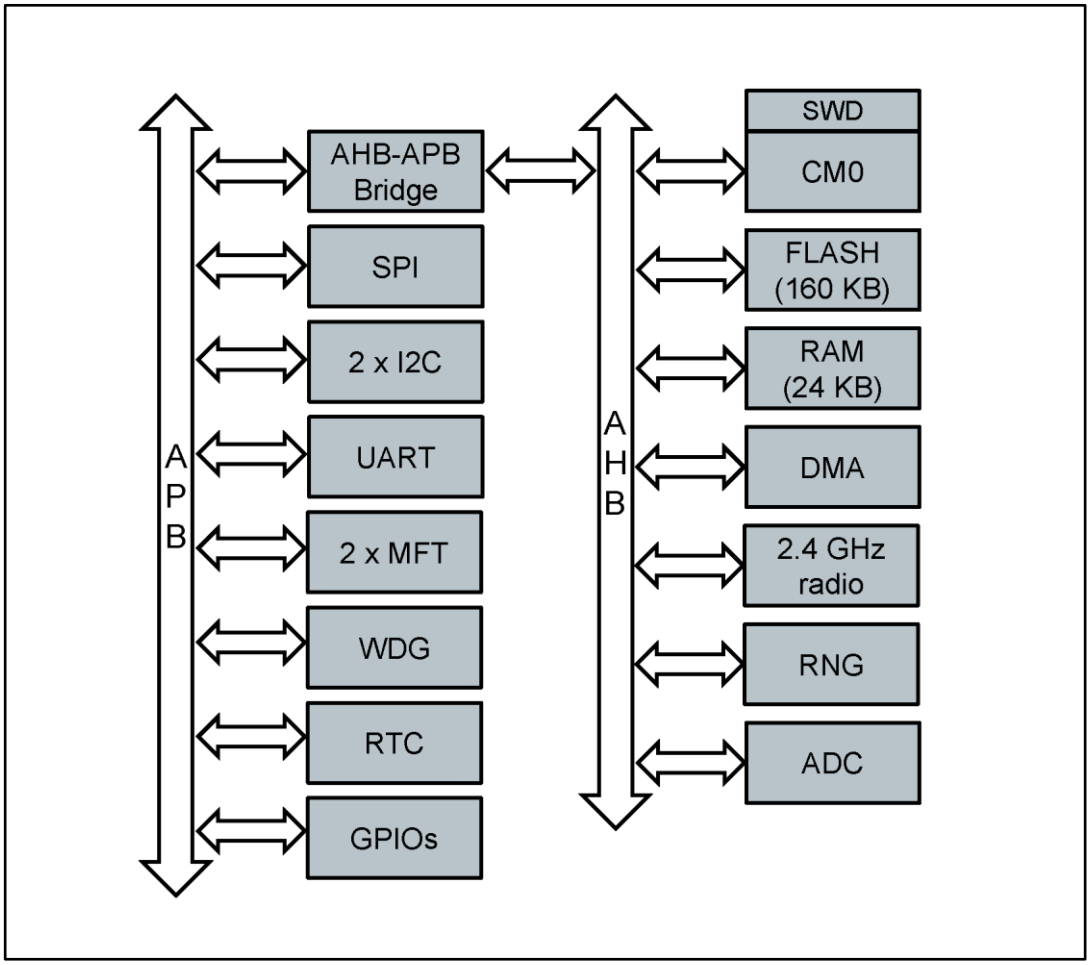
The BlueNRG-1 extends the features of award-winning BlueNRG network processor, enabling the usage of the embedded Cortex M0 for running the user application code.

The BlueNRG-1 includes 160 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 for interfacing with analog sensors, and for reading the measurement of the integrated battery monitor. A digital filter is available for processing a PDM stream.

The BlueNRG-1 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-1 improves the BlueNRG sleep mode current consumption allowing a further increase in the battery lifetime of the applications.

Figure 1: BlueNRG-1 architecture block diagram



2 BlueNRG-1 Bluetooth low energy stack

The BlueNRG-1 is complemented with a Bluetooth low energy stack C library that provides:

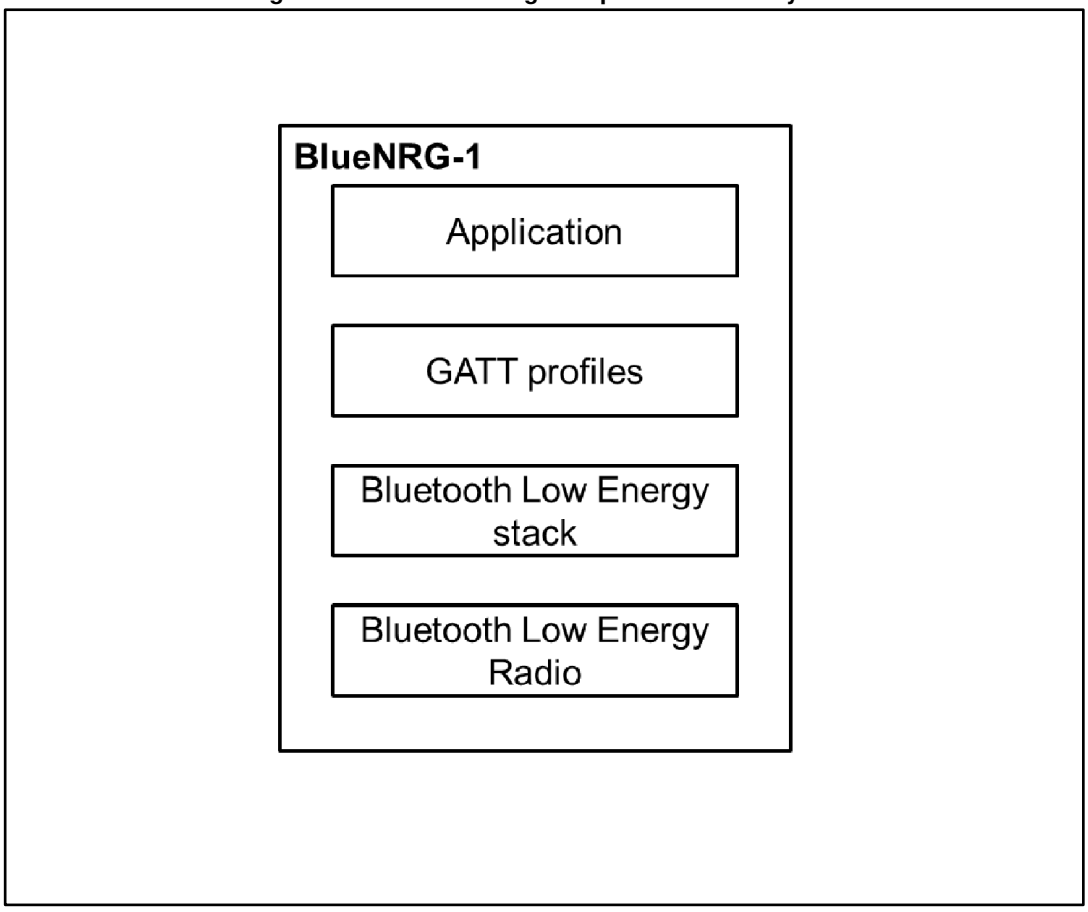
- Master, slave role support
- GAP: central, peripheral, observer or broadcaster roles
- ATT/GATT: client and server
- SM: privacy, authentication and authorization.
- L2CAP
- Link Layer: AES-128 encryption and decryption

The BlueNRG-1 can be configured for supporting Single Chip or Network processor applications.

In the first configuration, the BlueNRG-1 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.

*Figure 2: "BlueNRG-1 single chip RF software layers"*shows the single chip RF software layers.

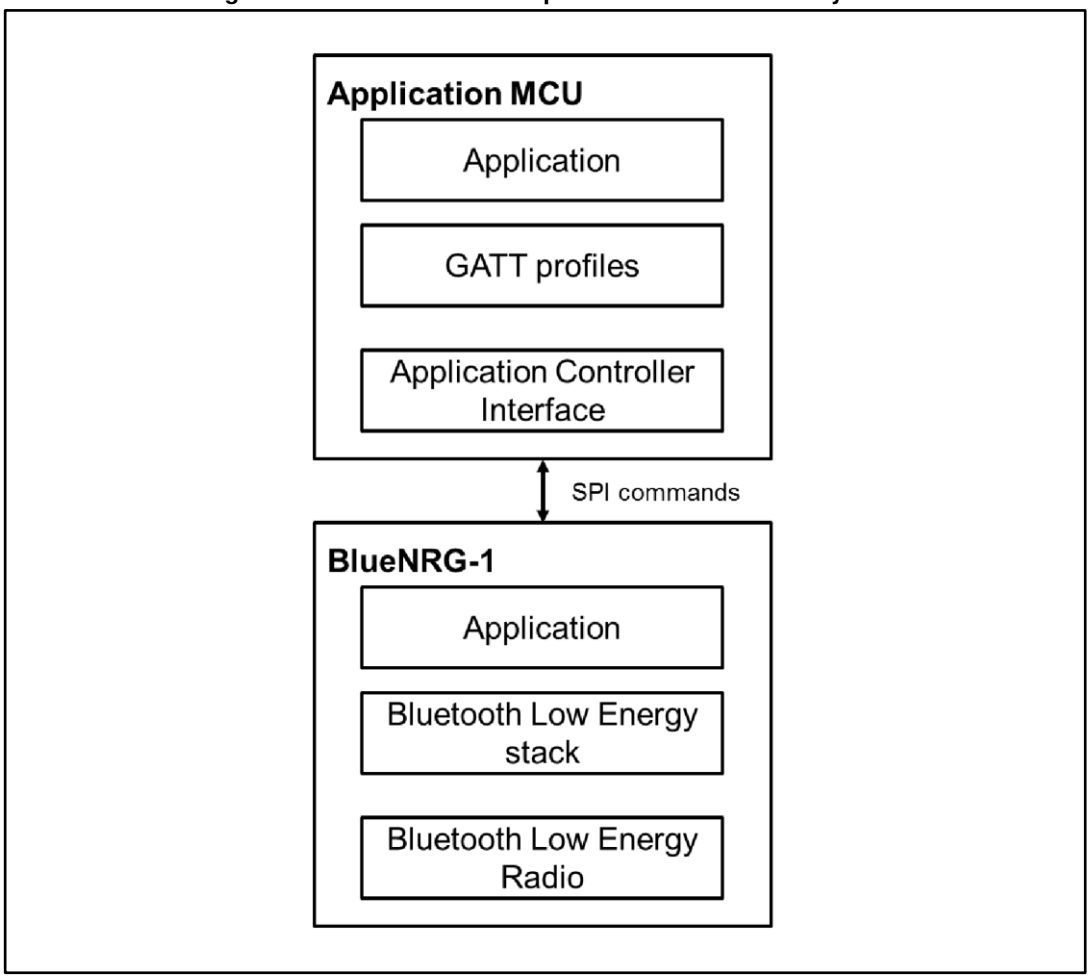
Figure 2: BlueNRG-1 single chip RF software layers



The BlueNRG-1 can be configured for operating as a network coprocessor. In this case a dedicated firmware is provided for supporting the interface with an external application

processor. The whole Bluetooth Low energy stack runs in the BlueNRG-1; the GATT profiles are provided for running in the application processor together with the application code. *Figure 2: "BlueNRG-1 single chip RF software layers"* shows the network processor RF software layers.

Figure 3: BlueNRG-1 network processor RF software layers



3 Functional details

The BlueNRG-1 integrates the blocks listed below:

- Cortex M0 core
- Interrupts management
- 160 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) (it is reserved to Bluetooth low energy protocol stack, but user application can read it)
- Public Key Cryptography (PKA) (reserved to Bluetooth low energy protocol stack)
- Peripherals:
  - SPI interface
  - UART interface
  - I^2C bus interface
  - GPIO
  - Multifunction timer
  - DMA controller
  - Watchdog
  - RTC
  - ADC with battery indicator and temperature sensor
  - PDM stream processor

The main blocks are described in the following subsection.

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-1 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-1 device, the user interrupts are connected to the interrupt signals of the different peripherals.

Table 2: Interrupt vectors

Position	Priority	Priority type	Description	Address
			Initial main SP	0x0000_0000
	-3	Fixed	Reset handler	0x0000_0004

Position	Priority	Priority type	Description	Address
	-2	Fixed	NMI handler	0x0000_0008
	-1	Fixed	HardFault handler	0x0000_000C
			RESERVED	0x0000_000C – 0x0000_0028
	3	Settable	SVC handler	0x0000_002C
			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	RESERVED	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 of 160 KB Flash memory and two banks of 12 KB ultra-low leakage static RAM blocks.

The 160 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 as bytes, half words (16 bits) or words (32 bits).

A 12 KB RAM block is always in retention mode, whereas the second 12 KB RAM block is switchable and can be put in retention mode according to the user needs.

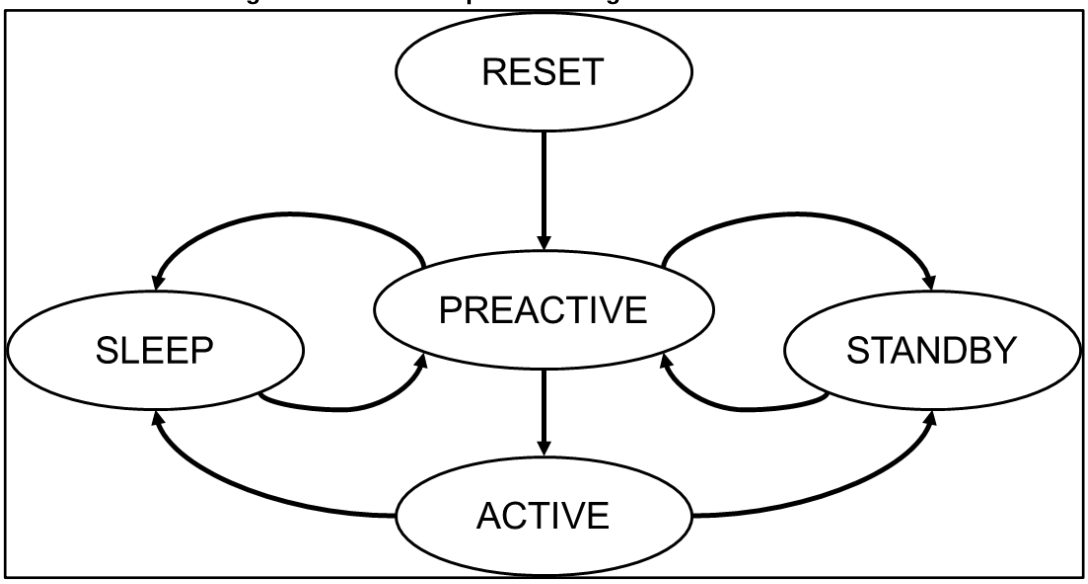
3.4 Power management

The BlueNRG-1 integrates both a low dropout voltage regulator (LDO) and a step-down DC-DC converter to supply the internal BlueNRG-1 circuitry.

The BlueNRG-1 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 4: BlueNRG-1 power management state machine



3.4.1 States 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.

If the external XO is at 32 MHz, some specific programming are needed, see *Section 3.5.3: "Switching to external clock"*.

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

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

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

3.4.2 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-1 states is provided below.

Table 3: Relationship between BlueNRG-1 states and functional blocks

	RESET	STANDBY	SLEEP	Preactive	Active	LOCKRX/ 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
LDO_DIG_1V2	OFF	OFF	OFF	ON	ON	ON	ON	ON
BOR	OFF	ON	ON	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 4: SYSTEM_CTRL registers

Address offset	Name	RW	Reset	Description
0x00	WKP_IO_IS	RW	0x00000000	Level selection for wakeup 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 wakes up feature on the IO is disabled. 1: The wakes 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.

Table 5: 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	R	RESERVED

Table 6: 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	R	RESERVED

Table 7: 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	R	RESERVED

AHBUPCONV peripheral base address (AHBUPCONV_BASE_ADDR) 0x40C00000.

Table 8: 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 9: BLUE_CTRL registers

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

3.5 Clocks and reset management

The BlueNRG-1 embeds an RC low-speed frequency oscillator at 32 kHz and an RO high-speed frequency oscillator at 16 MHz.

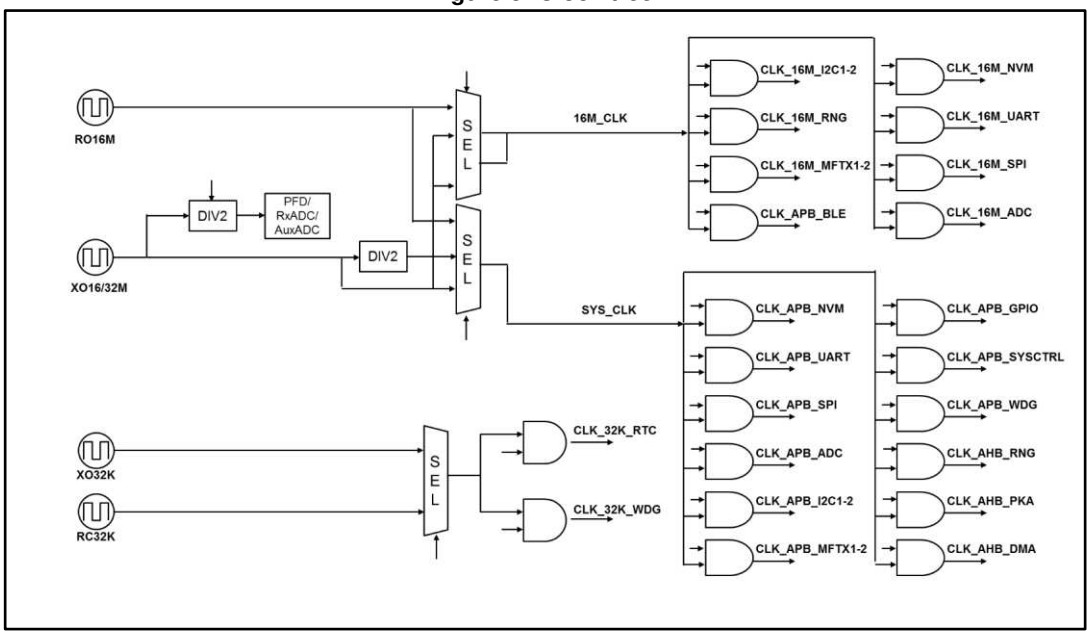
The low-frequency clock is used in low power mode and can be supplied either by a 32.7 kHz oscillator that uses an external crystal and guarantees up to ±50 ppm frequency tolerance, or by a ring oscillator with maximum ±500 ppm frequency tolerance, which does not require any external components.

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 (or 32 MHz) crystal is strictly necessary for RF communications.

The clock tree for the peripherals is as follows:

Figure 5: Clock tree



When 32 MHz XO is used, only the Cortex-M0, the DMA and the APB tree (except for BLE radio access) runs at 32 MHz. The rest of the clock tree is divided by two and stays 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
- I2C2
- ADC
- MFT1
- MFT2
- RTC
- WDG

By default all peripherals APB clock are enabled.

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)

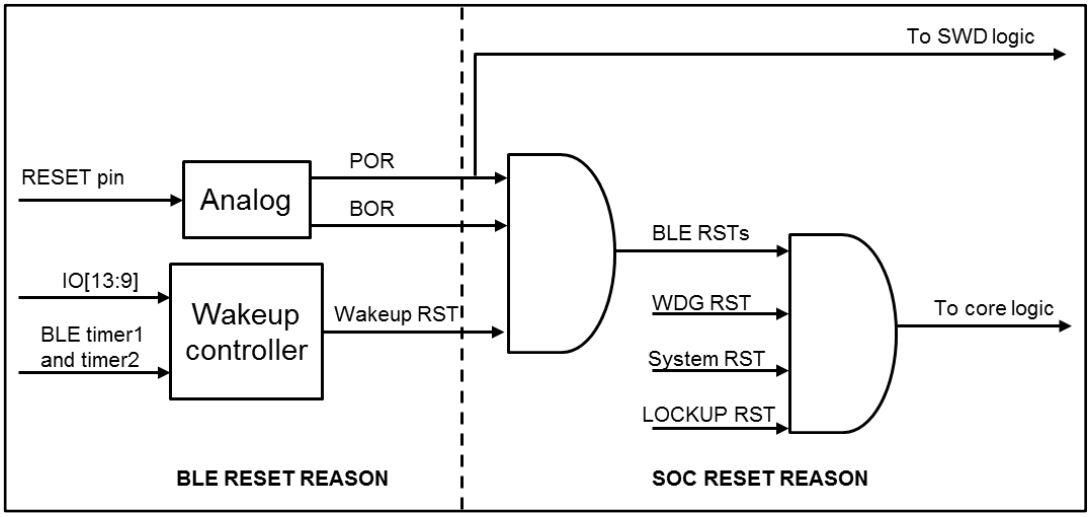
3.5.1 Reset management

Figure 6: "Reset and wakeup generation" shows the general principle of reset. Releasing the Reset pin puts the chip out of shutdown state. The wakeup 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 6: Reset and wakeup 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.

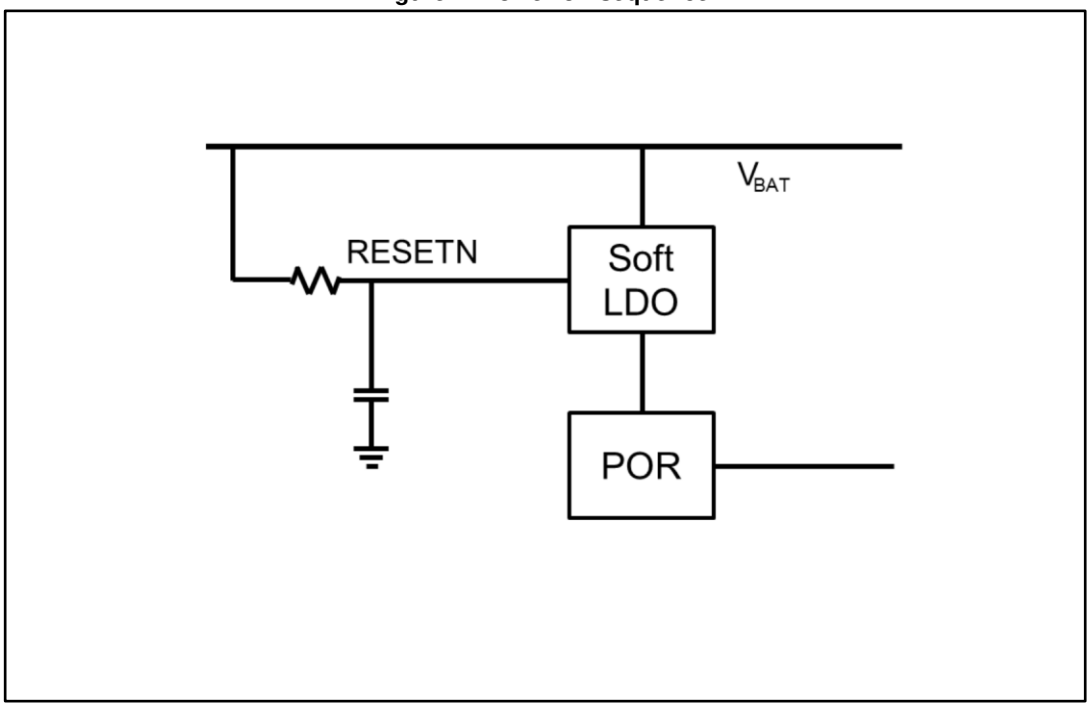
3.5.1.1 Power-on-Reset

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-1 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-1 and a part of the Flash controller (managing the Flash memory boot, which does not need to be impacted by system resets).

3.5.1.2 Power-on-sequence

The POR circuit is powered by a 1.2 V regulator. The regulator must also be powered on 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 will add a delay that can prevent RESETN signal to go high before VBAT has reached the nominal value (*Figure 7: "Power-on-sequence"*).

Figure 7: Power-on-sequence



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

3.5.1.3 Watchdog reset

The BlueNRG-1 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 will generate a reset. The watchdog Reset will Reset the Flash controller, the Cortex-M0 and all its peripherals but it will 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 the architected lock-up state resulting from an unrecoverable exception. The LOCKUP signal is used to generate reset in the BlueNRG-1. This reset will affect the Cortex-M0, the Flash controller, and all the peripherals. The LOCKUP signal does not Reset the Cortex-M0 debug circuitry.

3.5.2 Reset and wakeup reason decoding

The BlueNRG-1 provides a set of registers to identify the reason behind a reset and wakeup 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. Wakeup from IO9, IO10, IO11, IO12, IO13.
   b. Wakeup 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 Switching to external clock

When the system is in preactive state (run from internal RO 16 MHz), the software sequence to switch to active should include switch to external XO.

When the system uses an external XO at 32 MHz instead of 16 MHz, the following steps need to be done:

1. In preactive state, operate as follow routine:

```
uint8_t config[] = { 0x02, 0x3A, 0x44, 0x00};

BLUE_CTRL->RADIO_CONFIG = 0x10000 | (uint16_t) ( (uint32_t)config &
0x0000FFFF );

while((BLUE_CTRL->RADIO_CONFIG & 0x10000) != 0);
```

1. Set the bitfield MHZ32_SEL of register SYSTEM_CTRL, for the digital part.
2. Wait the state machine is in active state by using the follow routine:

```
uint32_t fsm_status;
```

```
uint16_t cnt_time = 0;
do {
 cnt_time++;
 fsm_status = 0x00004382;
 BLUE_CTRL->RADIO_CONFIG = 0x10000 | (uint16_t) ( (
(uint32_t)(&fsm_status) ) & 0x0000FFFF);
 while((BLUE_CTRL->RADIO_CONFIG & 0x10000) != 0);
} while(((fsm_status>>16) != 0x05) && (cnt_time<1000));
```

1. After switching to active state (and not before), write 0x15 in the COMMAND register of AHBUPCONV peripheral.

Around 10 system clock cycles after this last write, the system will run at 32 MHz except for specific blocks requiring a fixed 16 MHz.

The application can make the bus and the core run to 16 MHz by writing 0x14 in the COMMAND register of AHBUPCONV peripheral.

3.5.4 Clock and reset registers

CKGEN_SOC peripheral base address (CKGEN_SOC_BASE_ADDR) 0x40900000.

Table 10: CKGEN_SOC registers

Address offset	Name	RW	Reset	Description
0x00	CONTROL	RW	0x01FA03F0	Control clock and Reset of SOC. Refer to the detailed description below.
0x08	REASON_RST	R	0x00000000	Indicates the Reset reason from Cortex-M0. Refer to the detailed description below.
0x1C	DIE_ID	R	0x00000110	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.

Table 11: CKGEN_SOC - CONTROL register description: address offset CKGEN_SOC_BASE_ADDR+0x00

Bit	Field name	Reset	RW	Description
9:0	RESERVED	0x3F0	R	RESERVED
13:10	UART_CKDIV	0x0	RW	UART baud rate clock setting from 1 to 16 MHz according to the formula 16 / (n + 1) MHz.
31:14	RESERVED	0x007E8	R	RESERVED

Table 12: 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 13: CKGEN_SOC - DIE_ID register description: address offset CKGEN_SOC_BASE_ADDR+0x1C

Bit	Field name	Reset	RW	Description
3:0	REV	0x1	R	Cut revision
7:4	VERSION	0x01	R	Cut version
11:8	PRODUCT	0x01	R	Product
31:12	RESERVED	0x0	R	RESERVED

Table 14: 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	R	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
15:14	RESERVED	0x3	R	RESERVED
16	DMA	0x1	RW	DMA AHB clock
17	RNG	0x1	RW	RNG AHB clock
31:18	RESERVED	0x0	R	RESERVED