BlueNRG-LP

Current Overconsumption analysis.

EMEA Application

Sept. 2021

#1 Defining application needs

#2 Tips to meet power consumption target on your PCB

#1 Defining application needs

Estimate BlueNRG-LP power consumption

Datasheet

5.3 **Operating conditions**

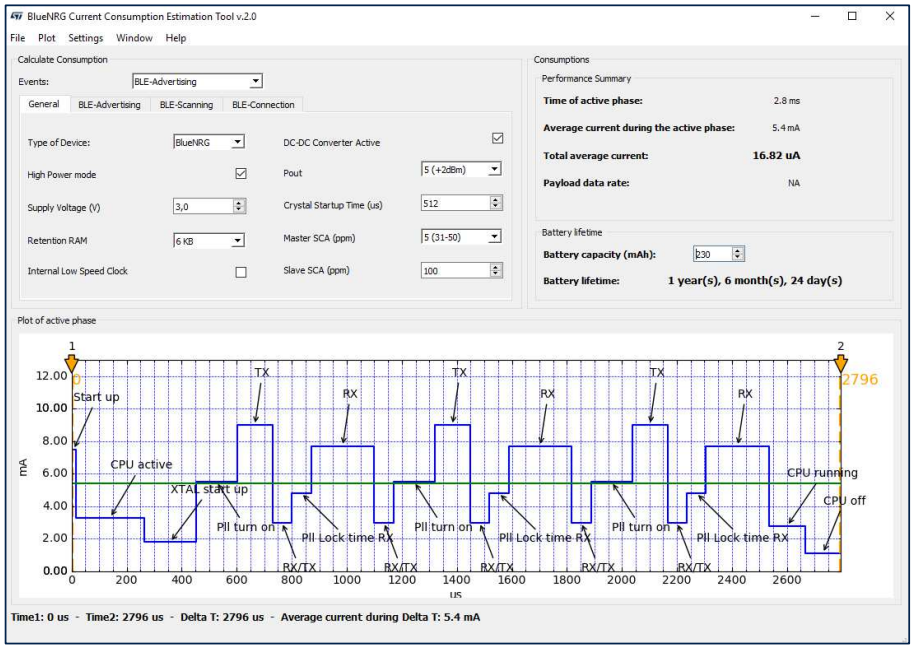
5.3.1 **Summary of main performance**

Table 11. **Main performance SMPS ON**

Symbol	Parameter	Test conditions	Typ. VDD = 1.8 V	Typ. VDD = 3.3 V	Unit
I_{CORE}	Core current consumption	SHUTDOWN	8	19	nA
		DEEPSTOP, no timer, wake-up GPIO, RAM0 retained	0.44	0.46	µA
		DEEPSTOP, no timer, wakeup GPIO, all RAM retained	0.62	0.64	
		DEEPSTOP (32 kHz LSI), RAM0 retained	0.94	1.06	
		DEEPSTOP (32 kHz LSI), all RAMs retained	1.12	1.24	
		DEEPSTOP (32 kHz LSE), RAM0 retained	0.64	0.75	
		DEEPSTOP (32 kHz LSE), all RAM retained	0.83	0.94	
		CPU in RUN (64 MHz). Dhrystone, clock source PLL64		2719	uA
		CPU in RUN (32 MHz). Dhrystone, clock source PLL64		2188	
		CPU in WFI (64 MHz), all peripherals off, clock source PLL64		1708	
		Radio RX at		3350	

typical figures

BNRG Power Consumption Tool (st.com)



estimate a real BLE profile

Application constraints vs. Power consumption

What is the distance expected

Device output power

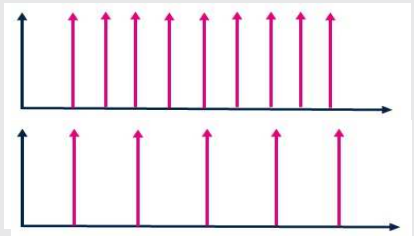
> **100 m** : +8 dBm required

< **10 m** : -2 dBm acceptable

up to -30% moving to -2 dBm

Application latency

BLE advertising & connection intervals



Power consumption vs. application **latency**

Application needs

MIPS requirements

Application running
@ 16 MHz ?
@ 32 MHz ?
@ 64 MHz ?

CLK settings vs. **real needs** ?

life.augmented

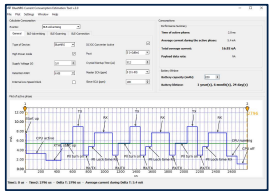
BOM vs. Power consumption

Trade-off between BOM and power consumption

Internal SMPS (extra 10uH coil required) vs. internal LDO (no 10uH coil)	**Rx** 3.35mA (SMPS) vs. 6.7mA (no SMPS) **Tx** (@0dBm) 4.3mA (SMPS) vs. 8.9mA (no SMPS) Thanks to SMPS coil divide consumption / 2
LSE with external 32kHz vs. LSI (no 32kHz XTAL)	Using LSE or LSI is impacting sleep current Deepstop : 0.9uA (LSE) vs. 1.2uA (no SMPS) Using 32kHz ensures best sleep current consumption

LSE : **L**ow **S**peed oscillator **E**xternal 32Khz
LSI : **L**ow **S**peed oscillator **I**nternal 32Khz

Easy simulation impact with BNRG PC power consumption tool



life.augmented

Choose the right components

Choosing right components allows best performance without any extra cost

LSE gain: 32kHz XTAL

32kHz XTAL parameters impact LSE required gain.
LSE gain can bet set from “low” to “high” by SW.

XTAL1

12.5 pF load cap
80 kOhms ESR max
High gain required : **1.25 uA** sleep (typ)

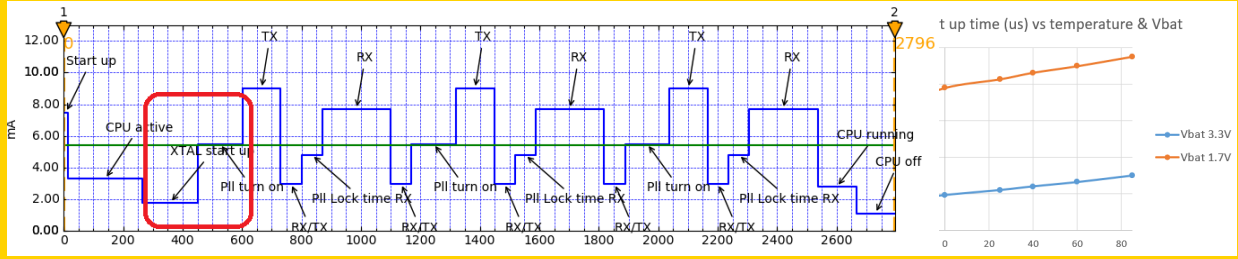
XTAL2

7 pF load cap
55 kOhms ESR max
Low gain required : **0.9 uA** sleep (typ)

XTAL choice impacting sleep by 40%
Better to use low ESR and load cap XTAL

HSE start up time: 32MHz XTAL

32MHz XTAL parameters impact HSE start up time
Depends also on Vbat or temperature



From datasheet 32MHz XTAL must have load caps from 5 to 9.2pF.
Ultra small package size and high load cap can have negative impact

HSE : **H**igh **S**peed oscillator **E**xternal 32Mhz

Please refer STEVAL-IDB011V1 for recommended references

#2 Tips to meet power consumption target on your PCB

life.augmented

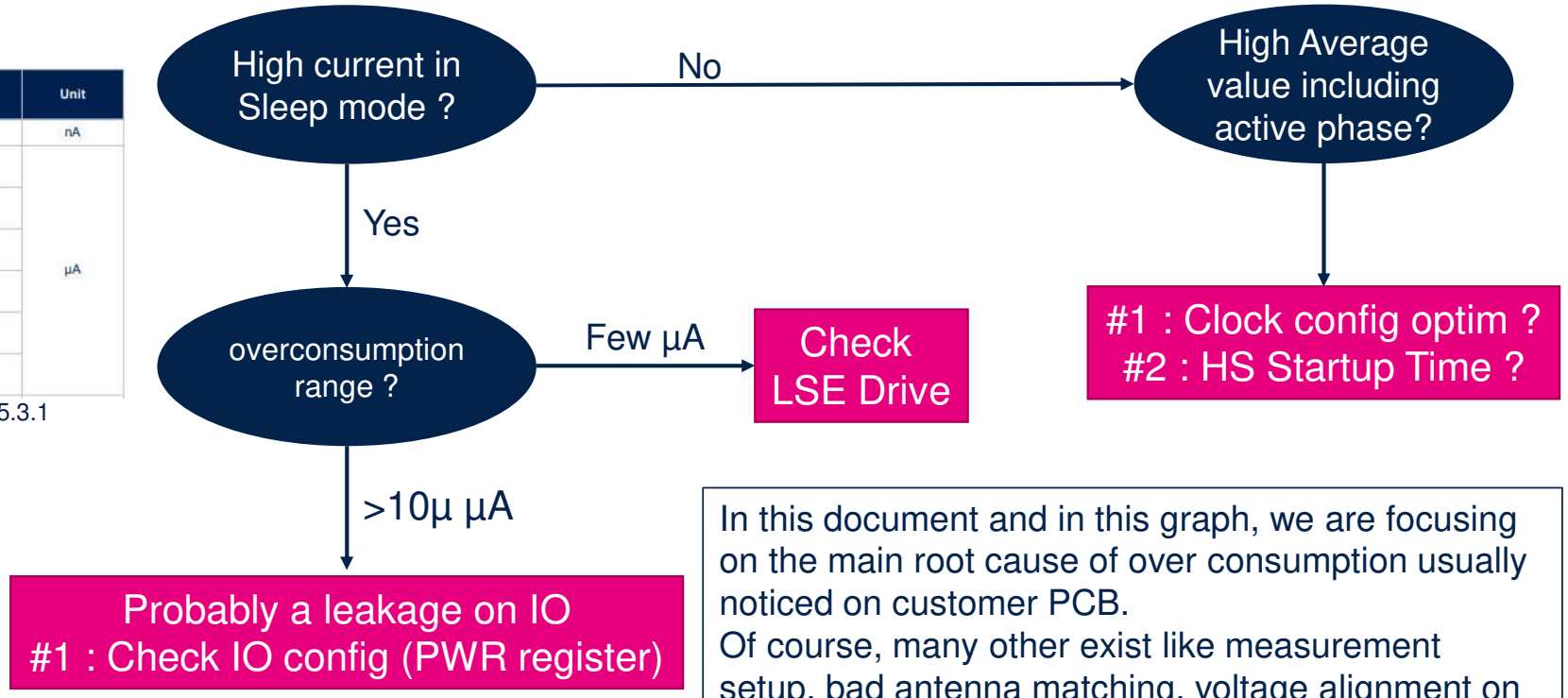
Current Overconsumption analysis Major root cause.

When/Where is the over-consumption ? Few questions to focus on probable root cause

Target value in Sleep mode :

Test conditions	Typ. VDD = 1.8 V	Typ. VDD = 3.3 V	Unit
SHUTDOWN	8	19	nA
DEEPSTOP, no timer, wake-up GPIO, RAM0 retained	0.44	0.46	µA
DEEPSTOP, no timer, wakeup GPIO, all RAM retained	0.62	0.64	
DEEPSTOP (32 kHz LSI), RAM0 retained	0.94	1.06	
DEEPSTOP (32 kHz LSI), all RAMs retained	1.12	1.24	
DEEPSTOP (32 kHz LSE), RAM0 retained	0.64	0.75	
DEEPSTOP (32 kHz LSE), all RAM retained	0.83	0.94	

- Extracted from DS, SMPS ON, section 5.3.1



In this document and in this graph, we are focusing on the main root cause of over consumption usually noticed on customer PCB.
Of course, many other exist like measurement setup, bad antenna matching, voltage alignment on IOs,…

IO configuration : 3 different IO level

In terms of power consumption, each IO can be configured at different level :

- #1 : IO - PAD level : possibility to configure a pull-up or a pull-down for each IO to avoid any leakage. This configuration is PCB dependent and applies to all IOs, even unused IO. This is the “PWR register”
- #2 : Function/Peripheral Level (ex : GPIO) : Each IO can be configured as GPIO, I²C, SPI,… This config is valid only during active phase. In sleep mode, peripherals are switched off.
- #3 : Output driving mode : Force IO level to high or low – This level is maintained whatever active or sleep mode and is having drive capabilities (ex : driving a LED).

The next slides describe behavior of IO during sleep phase according to IO mode.

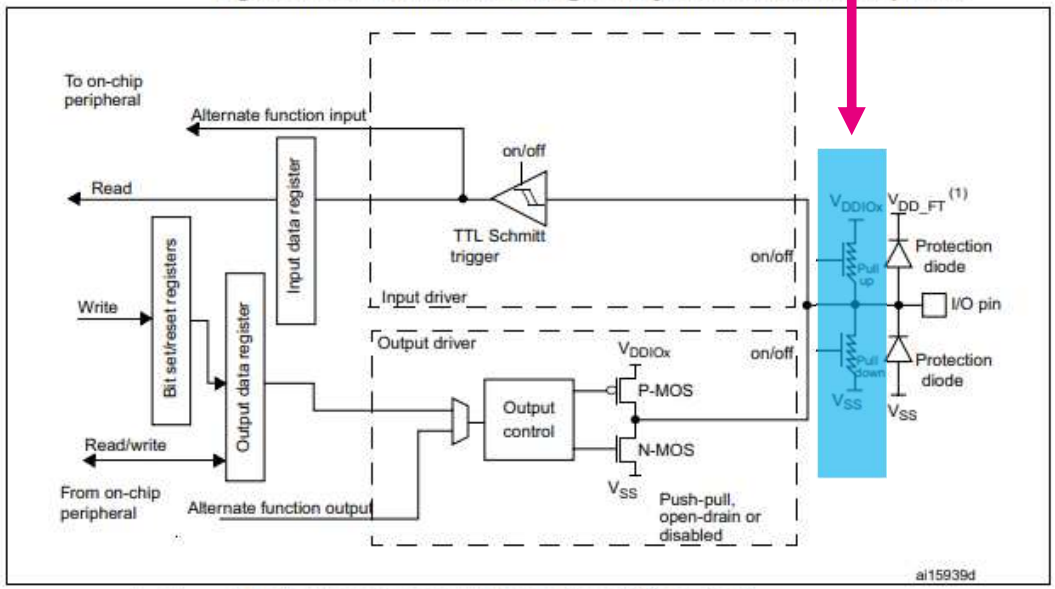
Reminder : IO ≠ GPIO

I/O behavior during Sleep mode #1 : PWR register config

- For All I/Os : programmable Pull-Up/Pull-Down by configuring PWR register
  - Ex : *LL_PWR_EnablePDA(LL_PWR_PUD_IO) //PullDown on PA0*
  - **This level is maintained during Sleep mode**
  - **No drive capabilities.**
  - Purpose is to avoid current leakage.

In BlueNRG-LP SDK code examples, function BSP_IO_Init() is used to configure all IOs for optimal power consumption with STEAVL-IDB01x eval board. It can be reused and adapt for custom PCB.

Figure 16. Basic structure of a digital only five-volt tolerant I/O port bit



1. V_{DD_FT} is a potential specific to five-volt tolerant I/Os and different from V_{DD}.

This PWR register configuration is valid whatever IO usage (GPIO, UART, ADC, SPI, I2C,…)

I/O behavior during Sleep mode #2 GPIO mode

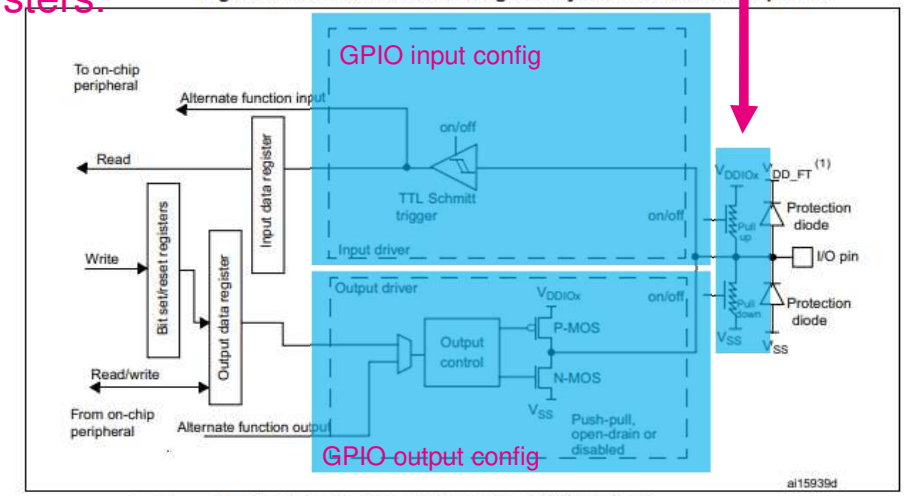
- All I/Os can be configured in GPIO mode :
  - **The GPIO peripheral is switched off during Sleep** like any other peripherals (SPI, I2C,..).
  - **GPIO Output level set by SW is not maintained during Sleep mode.**
    - Hence, GPIO level during sleep phase depends on internal Pull-Down/Pull-Up configuration.
  - Possibility to configurable PullUp/PullDown for each GPIO
    - It affects the same config as the one used by PWR registers.
    - Must be configured to avoid leakage

Example of GPIO configuration : (PA4 - Output – Pull-Down)

```
LL_AHB_EnableClock(LL_AHB_PERIPH_GPIOA);
LL_GPIO_ResetOutputPin(GPIOA, LL_GPIO_PIN_4);

LL_GPIO_SetPinMode(GPIOA, LL_GPIO_PIN_4, LL_GPIO_MODE_OUTPUT);
LL_GPIO_SetPinSpeed(GPIOA, LL_GPIO_PIN_4, LL_GPIO_SPEED_FREQ_HIGH);
LL_GPIO_SetPinOutputType(GPIOA, LL_GPIO_PIN_4, LL_GPIO_OUTPUT_PUSHPULL);
LL_GPIO_SetPinPull(GPIOA, LL_GPIO_PIN_4, LL_GPIO_PULL_DOWN);
```

Figure 16. Basic structure of a digital only five-volt tolerant I/O port bit



1. V_{DD_FT} is a potential specific to five-volt tolerant I/Os and different from V_{DD}.

Same philosophy applies while using alternate function of IO (SPI, I2C, UART,…)

I/O behavior during Sleep mode #3 Output Driving mode

- Eight I/Os (PA4/ PA5/ PA6/ PA7/ PA8/ PA9/ PA10/ PA11) can be : output driving
- **This output level is maintained during both Active and Sleep mode.**
  - **Ability to drive and maintain level during Sleep mode.**
  - Configurable PullUp/PullDown for each I/O

- API in SDK : LL_PWR_SetPA4OutputinDEEPSTOP(LL_PWR_IOCFG_HIGH);
  - This call set PA4 drive to High – PA4 will be always HIGH
  - Similar APIs exist for the 8 eligible I/Os

Typical use case for this mode : “Driving a LED while BlueNRG-LP in sleep mode”

life.augmented

LSE drive config

Low speed oscillator drive is tunable in BlueNRG-LP SDK :

- LL_RCC_LSEDRIVE_LOW
- LL_RCC_LSEDRIVE_MEDIUMLOW
- LL_RCC_LSEDRIVE_MEDIUMHIGH
- LL_RCC_LSEDRIVE_HIGH

Lowest current conso when LS xTAl active

Impact : +0.35µA from LOW to HIGH Drive

Highest current conso when LS xTAl active

LSE drive highly **depends on selected Low Speed Xtal**. A resonator with a relatively high load-capacitance (such as 12.5 pF) will require more power for the oscillator to drive the oscillation loop at the resonator nominal frequency

From tests conducted with NX2012SA-32.768kHz-EXS00A-MU00389 associated with 8.2pF load caps, ST has confirmed that LSE drive can be set to low.

API in SDK : ***LL_RCC_LSE_SetDriveCapability(LL_RCC_LSEDRIVE_LOW)***

HS oscillator start up time

High Speed Oscillator start up time is tunable in BlueNRG-LP SDK :

#define HS_STARTUP_TIME 210 // 512 us

This parameter is defined in 625/256 µs step so ~2.44 us.

HSE start up time highly **depends on selected High Speed Xtal, temperature and voltage**.
BlueNRG-LP documentation is providing below recommendation for start up time measurement

*HS_STARTUP_TIME = XTAL_startup_measured(Vmin)*1.1*1.3*

- Vmin is the minimal operative voltage on user application.
- 10%: to take in account the effect of temperature variations within the related operating range
- 30%: to take in account the tolerance of crystal motional inductance and capacitance.

From tests conducted on STEVAL-IDB011V1 and considering an application running @ 3.3V we measured XTAL_startup_measured @ 358us (507us @1.7V).

In such condition, HS_STARTUP_TIME can be defined at 512us on STEVAL-IDB01x board.

Optimal value must be computed for custom PCB.
Refer to AN5503 for measurement procedure

life.augmented

HS clock configuration

High speed Clock if fully configurable according to application needs. Both sytem clock and BLE clock can be configured thanks to API SystemInit(uint8_t SysClk, uint8_t BleSysClk)

SysClock possible value :

- SYSCLK_DIRECT_HSE /* 32 MHz DIRECT HSE with HSI OFF */ ← Optimal power conso (HSI off)
- SYSCLK_64M /* system clock frequency is 64 MHz from PLL */
- SYSCLK_32M /* system clock frequency is 32 MHz from PLL */
- SYSCLK_16M /* system clock frequency is 16 MHz from PLL */

BLEClock possible value :

- BLE_SYSCLK_NONE /* BLE system clock disabled */
- BLE_SYSCLK_32M /* BLE system clock frequency is 32 MHz */
- BLE_SYSCLK_16M /* BLE system clock frequency is 16 MHz */ ← Optimal power conso

A typical and optimal clock setting in terms of power consumption is the below one :

SystemInit(SYSCLK_DIRECT_HSE, BLE_SYSCLK_16M)

life.augmented

Thank you

life.augmented