UM2211
User manual

BLE-Sub1GHz development kit

Introduction

The BLE-Sub1GHz development kit uses the BlueNRG-1 system-on-chip for Bluetooth® low energy (BLE) functionality and the S2-LP transceiver for sub-1GHz functionality.

This package is for applications aiming to implement the BLE and sub-1Ghz protocols with the corresponding BlueNRG-1 and S2-LP devices. It includes use examples and recommendations on the simultaneous use of the two protocols.

This package is based on specific BlueNRG-1 and S2-LP SDK software component versions available at the time of release. Check release notes to find the version number associated with each SDK.

For further information, refer to STSW-BLUENRG1-DK literature for the BlueNRG-1 hardware and software development kit, and STSW-S2LP-DK and STSW-S2LP-SFX-DK literature regarding S2-LP hardware and software development kit.

Contents

List of tables

List of figures

1 Getting started

This section describes the software and hardware components of the kits.

1.1 Software

Download the BLE-Sub1GHz software package (STSW-BNRG-S2LP-DK) from *www.st.com* and extract BLE-Sub1GHz DK-Setup-x.x.x.zip contents to a temporary directory.

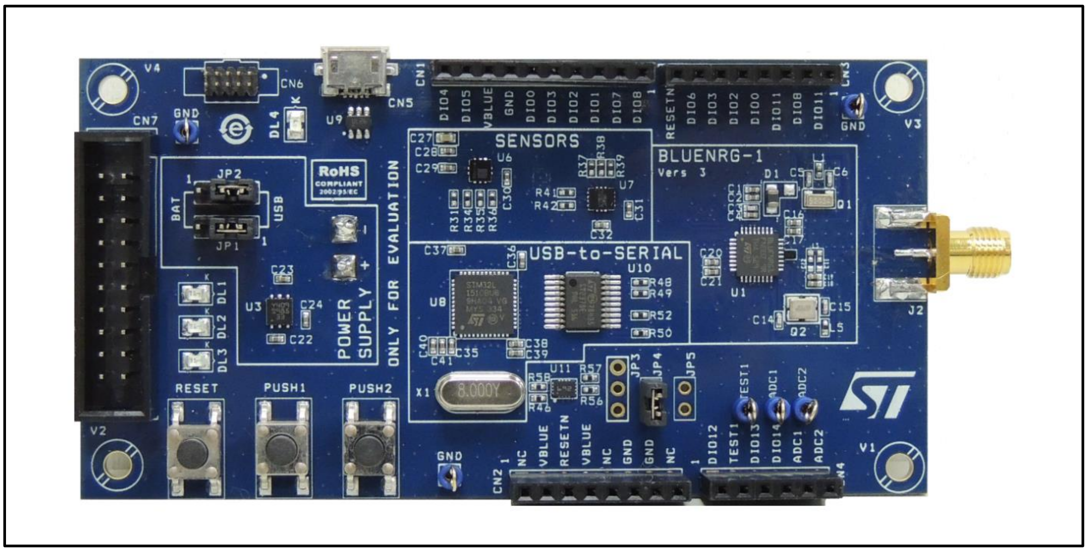
Launch BLE-Sub1GHz DK-Setup-x.x.x.exe and follow the on-screen instructions.

1.2 Hardware

To run the demo of this package, you need:

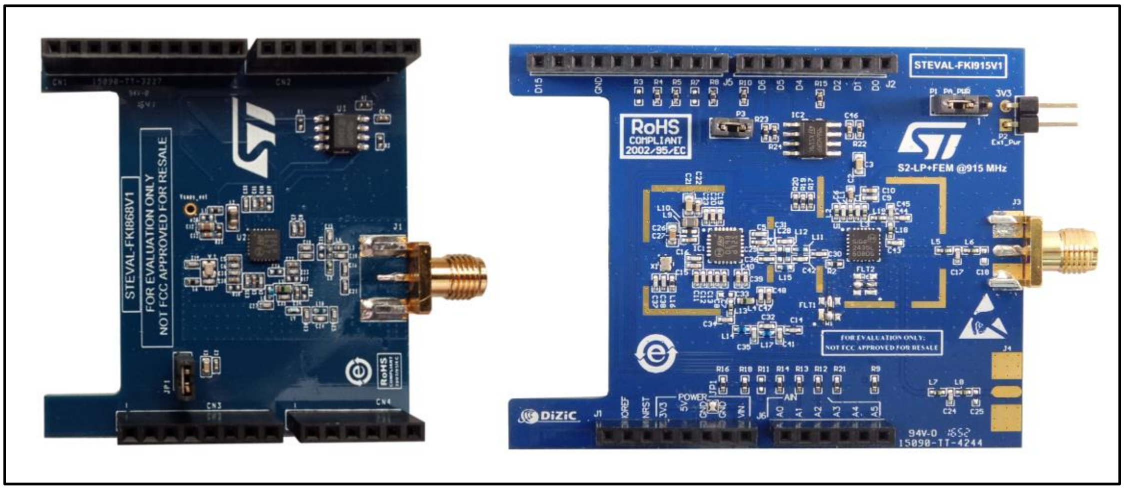
- a BlueNRG-1 STEVAL-IDB007V1 kit (refer to UM2071 on *www.st.com* for details).

Figure 1: STEVAL-IDB007V1 board



- an S2-LP STEVAL-FKI868V1 or STEVAL-FKI915V1 kit, depending on the operating band (see UM2149 on *www.st.com* for details).

Figure 2: STEVAL-FKI868V1 and STEVAL-FKI915V1 boards



For users interested in the Sigfox protocol, the table below shows which boards covers which radio configuration zone (RCZ) or zones. See UM2169 on *www.st.com* for more details.

Table 1: S2-LP boards for Sigfox radio configuration zones

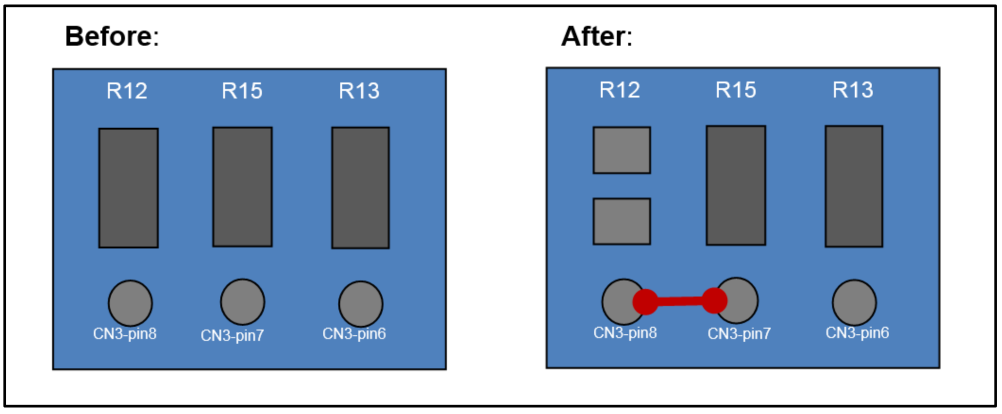
S2-LP board	Radio configuration zone
STEVAL-FKI868V1	RCZ1
STEVAL-FKI915V1	RCZ2
STEVAL-FKI915V1	RCZ4

1.2.1 Modifying the STEVAL-IDB007V1 kit

To render the STEVAL-IDB007V1 compatible with the STEVAL-FKI868V1 and STEVAL-FKI915V1 boards, perform these modifications on the **rear side of the board**.

1. Remove R12
2. Create a short circuit between pin 8 and pin 7 on the CN3 connector

Figure 3: CN3 alterations

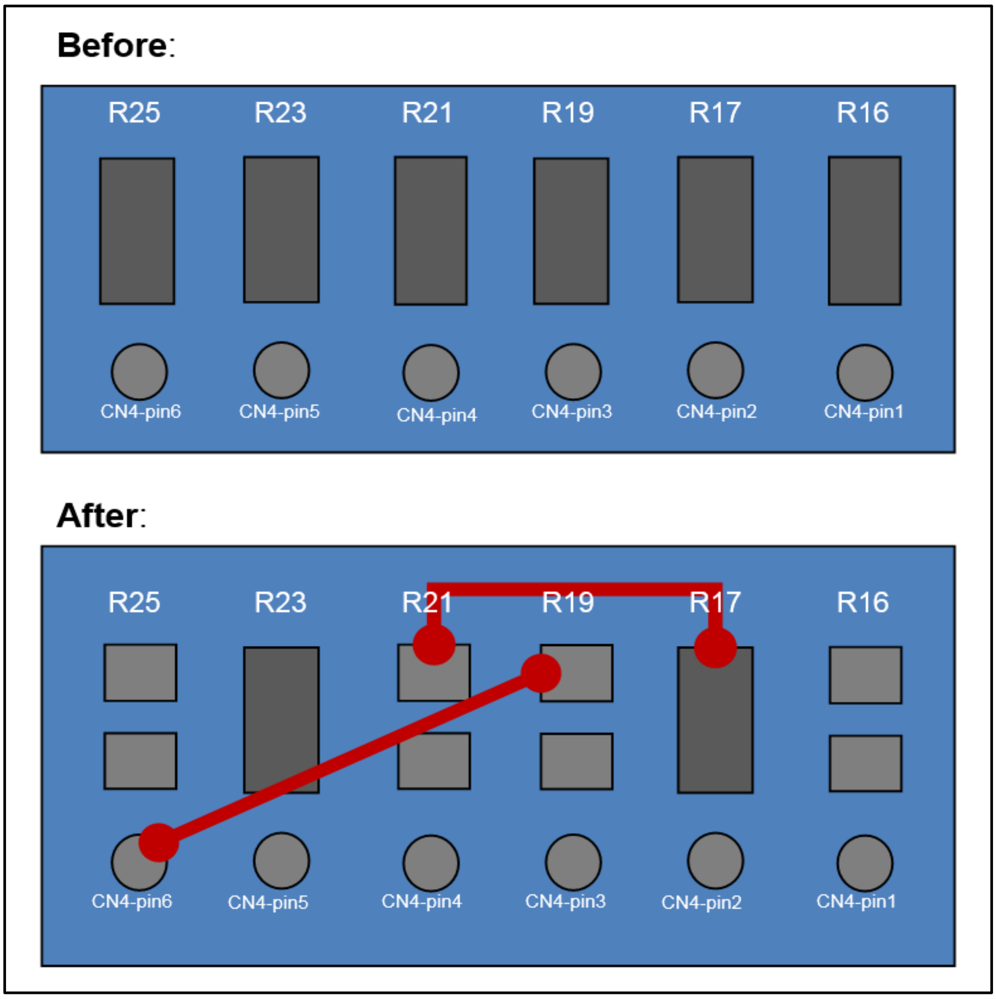


3. Remove R25, R21, R19 and R16

4 Set a short between the internal pad of R21 and R17

5 Set a short between the pin 6 of the CN4 connector and the internal pad of R19

Figure 4: CN4 alterations



1.2.2 BlueNRG-1 SoC connections

The platform pin connection of the STEVAL-IDB007V1 is given in the following table.

The information refers to a STEVAL-IDB007V1 board that has been modified for compatibility with the STEVAL-FKI868V1 and STEVAL-FKI915V1 boards.

Table 2: Platform pin description with board function

Pin name	Pin num	Function							Arduino connectors			
		LEDs	S2-LP	Buttons	FKI_E2PROM	Pressure sensor	3D accelerometer and gyroscope	JTAG	CN1	CN2	CN3	CN4
DIO10	1							JTMS-SWTDIO				
DIO9	2							JTCK-SWTCK				
DIO8	3				SPI_CS				pin 1 (IO8)		pin 2 (TX)	
DIO7	4	DL2							pin 2 (IO9)			pin 6 (SCL)
DIO6	5	DL1	SDN								pin 7 (IO6)	pin 5 (SDA)
DIO5	7			PUSH2		SDA (PUSH2 button)			pin 9 (SDA)			
DIO4	8					SCL			pin 10 (SCL)			
DIO3	9		SPI_SDO		SPI_SDO		SPI_SDO		pin 5 (MISO)		pin 6 (IO5)	
DIO2	10		SPI_SDA		SPI_SDA		SPI_SDA		pin 4 (MOSI)		pin 5 (IO4)	
DIO1	11						SPI_CS	JTAG-TDO	pin 3 (CS)			
DIO0	12		SPI_SCL		SPI_SCL		SPI_SCL	JTAG-TDI	pin 6 (SCK)		pin 4 (IO3)	
DIO14	13	DL3	SPI_CS									pin 4 (AD3)
RESET	25		RESET	RESET				RESET		pin 3 (NRST)	pin 8 (IO7)	
DIO13	29		GPIO3	PUSH1								pin 3 (AD2)
DIO12	30						INT1					pin 1 (AD0)
DIO11	32										pin 1 (RX) pin 3 (IO2)	pin 2 (AD1)

1.2.3 S2-LP

The S2-LP embedded on the STEVAL-FKI868V1 or STEVAL-FKI915V1 board connected to the STEVAL-IDB007V1 can be driven by the BlueNRG-1 via SPI.

GPIO3 is connected to a BlueNRG-1 wake-up pin to signal certain events to the BlueNRG-1.

The BlueNRG-1 SoC acts as an SPI master and can configure the device through registers as well as send and receive data to and from the sub-1GHz channels.

1.2.4 E2PROM

The E2PROM containing the manufacturing data of the S2-LP board can be accessed by the BlueNRG-1 using the SPI bus.

1.2.5 Hardware setup

1. Connect a 2.4 GHz antenna to the STEVAL-IDB007V1 SMA connector.
2. Connect an 868 or 915 MHz antenna to the STEVAL-FKI868V1 or STEVAL-FKI915V1, respectively.
3. Ensure the jumper configuration on the board is as in sec 0.
4. Connect the motherboard to the PC with a USB cable.
5. Verify that PWR LED DL4 is on.

2 BLE-Sub1GHz Navigator overview

The BLE-Sub1GHz Navigator application in the software package is an interactive, simple and user friendly graphical user interface for browsing the following BLE-Sub1GHz DK SW package resources:

- BLE-Sub1GHz examples
- Board description and related changes
- BLE-Sub1GHz Development Kits
- Release notes
- License files

This GUI lets you directly download and run the selected prebuilt application binary image on the BlueNRG-1 platform without a JTAG interface. The user has access to the demo description, board configuration and the source code.

Run the utility by double-clicking the BLE-Sub1GHz Navigator icon in

- Start → STMicroelectronics → BLE-Sub1GHz DK X.X.X → BLE-Sub1GHz Navigator

Figure 5: BLE-Sub1GHz Navigator

The BLE-Sub1GHz DK SW package release notes and license file can be accessed from the menu.

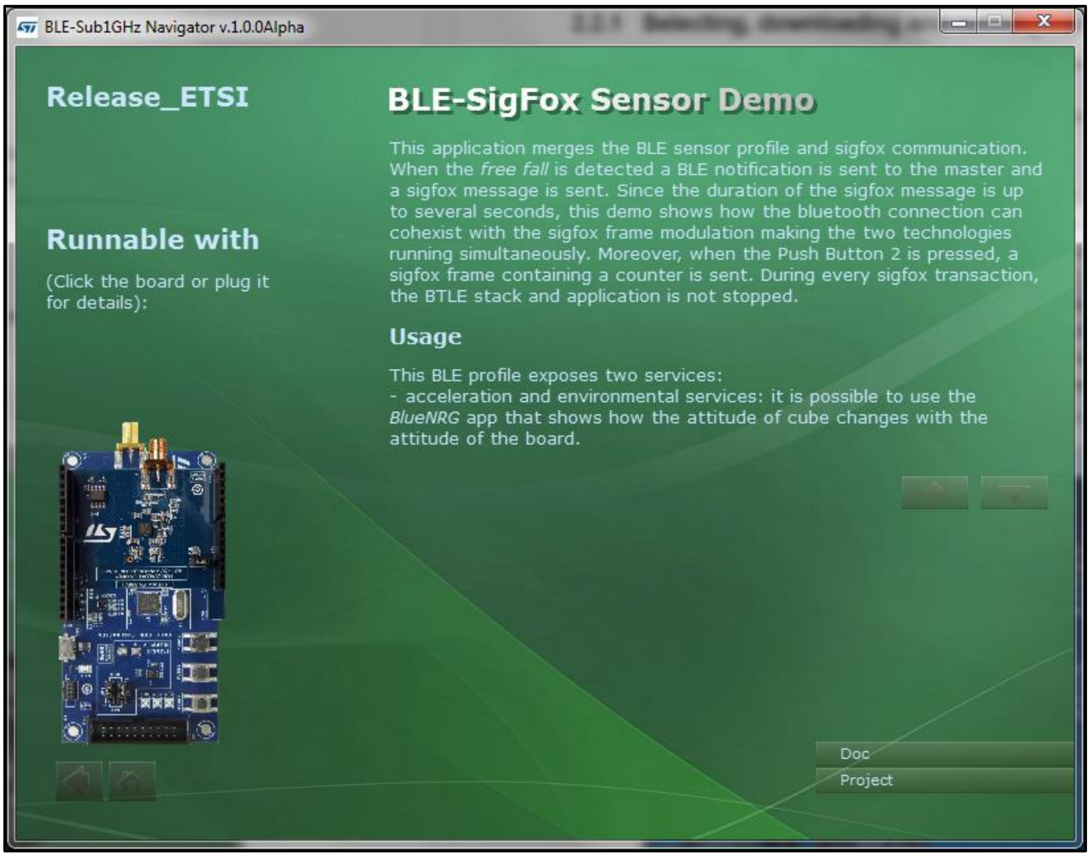
2.1 Demonstration applications

Navigate the menus to find the reference/demo application you wish to run.

The following information is provided for each application:

- Application settings (if applicable)
- Application description
- Hardware-related information (LED signals, jumper settings etc.)

Figure 6: BLE-SigFox Sensor Demo application window



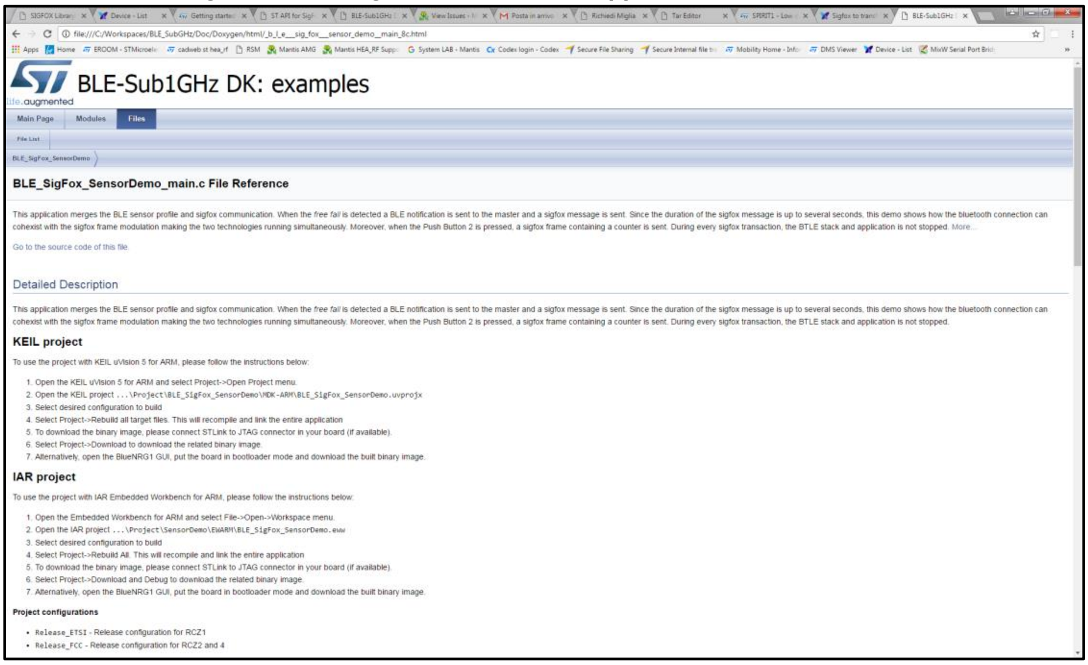
The following functions are available for each application:

- **Flash & Run**: to automatically download and run the available prebuilt binary file on a BlueNRG-1 platform connected to a PC USB port.

Figure 7: BLE-SigFox Sensor Demo application flashing

- **Doc**: to access the application documentation

Figure 8: BLE-SigFox Sensor Demo application documentation



- **Project**: to open the project folder containing the application headers, code and IAR files.

2.2 Development kits

This window displays the available BLE-Sub1GHz DK Kit platforms and resources.

Figure 9: STEVAL-IDB007V1 plus FKI868V1 kit components

Hovering the mouse pointer over a specific item highlights the component on the board image.

3 Programming with BlueNRG-1 System On Chip

The BlueNRG-1 Bluetooth low energy (BLE) stack is provided as a binary library of APIs to control its BLE functions. Some callbacks are included for user applications to handle BLE stack events.

The user must link this binary library to the application and use the relevant APIs for BLE functionality and complete the related stack event callbacks to handle stack events in the desired manner.

A set of software driver APIs is also available to access the BlueNRG-1 SoC peripherals and resources (ADC, GPIO, I2C, MFTX, Micro, RTC, SPI, SysTick, UART and WDG).

3.1 Software directory structure

The BLE-Sub1GHz DK SW package files are organized as follows:

- **Application**: contains BLE-Sub1GHz Navigator PC application.
- **Doc**: contains the BLE-Sub1GHz demo applications doxygen documentation DK release notes and license file.
- **Firmware**: contains prebuilt binary applications.
- **Library**
  – **Bluetooth LE**: Bluetooth low energy stack binary library and all the definitions of stack APIs, stack event callbacks and constants. OTA firmware upgrade source and header file.
  – **BlueNRG1_Periph_Driver**: BlueNRG-1 drivers for device peripherals (adc, clock, dma, flash, gpio, i2c, timers, rtc, spi, uart and watchdog).
  – **CMSIS:** BlueNRG-1 CMSIS files.
  – **SDK_Eval_BlueNRG1**: SDK drivers providing an API interface to the BlueNRG-1 platform hardware resources (LEDs, buttons, sensors, I/O channel).
  – **hal**: Hardware abstraction level APIs for abstracting some BlueNRG-1 hardware features (sleep modes, clock based on SysTick, etc.).
  – **SDK_Eval_S2LP**: SDK drivers providing an API interface to the S2-LP platform hardware resources (S2-LP SPI and GPIOs, E2PROM).
  – **Sigfox**: contains the library for the sigfox protocol on the BlueNRG-1 (ARM® Cortex®-M0 core).
  – **S2LP_Library**: contains the generic library for the S2-LP.
- Projects
  – **SigFox_Applications**: contains demonstration applications using the BLE and SigFox protocols. Headers, source files and Keil and IAR project files are available. These applications link the SigFox library contained in the Library/SigFox folder (not the S2-LP Library).*common/st_lowlevel*: contains the implementation of the ST LowLevel API to run the library on the BlueNRG-1. *common/id_key_retriever*: is used to retrieve the SigFox data as ID and PAC from the board.
  BLE_SigFox_SensorDemo: sources and projects files of the BLE_Sigfox_SensorDemo.
  – **Dual_Radio_Chat**: an application for communication between a mobile device and an S2-LP using BLE-Sub1GHz functionality.
  – **S2LP_Communication**: a simple application that uses the S2-LP to exchange packets with another node; it uses the S2LP Library to configure the S2-LP.

3.2 Before using the kit

You should register your STEVAL-FKI868V1 or STEVAL-FKI915V1 board with a sigfox ID/PAC/KEY before using it with the STEVAL-IDB007V1[a].

3.3 Initialization sequence

This procedure lets you develop an application for sigfox on the STEVAL-FKI868V1 or STEVAL-FKI915V1 platforms.

1. Call `SystemInit()` API to initialize the BlueNRG-1 device vector table, interrupt priorities and clock.
2. Call `SdkEvalIdentification()` API to configure selected BlueNRG-1 platform.
3. Call `aci_hal_write_config_data()` API to configure BlueNRG-1 public address (if public address is used)
4. Call `aci_gatt_init()` API to initialize the BLE GATT layer
5. Call `aci_gap_init("role")` API to initialize the BLE GAP layer depending on the selected device role
6. Call `aci_gap_set_io_capability()` and `aci_gap_set_authentication_requirement()` APIs to set the proper security I/O capability and authentication requirement (if BLE security is used).
7. Call `aci_gatt_add_service()`, `aci_gatt_add_char()` and `aci_gatt_add_char_desc()` APIs to define the required Services & Characteristics & Characteristics Descriptors if the device is a GATT server
8. Call `BlueNRG_Stack_Initialization(&BlueNRG_Stack_Init_params)` to initialize the BLE stack.
9. Initialize the S2-LP SPI and S2-LP SDN pin setting it high.
10. Call `ST_MANUF_API_set_xtal_freq(50000000)` API to set the XTAL frequency to the sigfox library.
11. Call function `enc_utils_retrieve_data(…)` to retrieve the SigFox ID, PAC and RCZ.
    This function will give the ID of the board in the variable uint32_t id.
12. For **RCZ1**, use the sequence:
    `ST_SIGFOX_API_open(ST_RCZ1,(uint8_t*)&id);`
13. For **RCZ2**, use the sequence:
```
ST_SIGFOX_API_open(ST_RCZ2,(uint8_t*)&id);
uint32_t config_words[3]={0x1FF,0,0};
ST_SIGFOX_API_set_std_config(config_words,1);
```
14. For **RCZ4**, use the sequence:
```
ST_SIGFOX_API_open(ST_RCZ4,(uint8_t*)&id);
uint32_t config_words[3]={0,0xF0000000,0x1F};
ST_SIGFOX_API_set_std_config(config_words,63);
```

[a] see user manual UM2169 on www.st.com

15 Add the following `while(1)` loop

`while (1){ BTLE_StackTick();` - the BLE stack tick

`APP_Tick();` - a user tick handler where user actions/events are processed

`SigFox_Tick();`

`BlueNRG_Sleep(…);` - enables BlueNRG-1 sleep mode and preserves the BLE radio operating modes

`}`

16 Transmit a sigfox frame with the 4-byte `customer_data` buffer to send:

without downlink request:

– ST_SIGFOX_API_send_frame(customer_data,4,customer_resp,0,0);

with downlink request:

– ST_SIGFOX_API_send_frame(customer_data,4,customer_resp,0,1);

The function returns after approximately 50 s and, if the error code is 0 (ok), the `customer_resp` will contain an 8-byte response.

3.4 ST LowLevel implementation

The *st_lowlevel_bluenrg1.c* module in Projects/SigFox_Applications/common/st_lowlevel interfaces the ST SigFox library with the hardware platform using:

- S2LP_EVAL_SPI, GPIO and E2PROM modules
- BlueNRG1_Periph_Driver

To avoid losing the BLE connection while the sigfox frame is being transmitted, it is necessary to tick the BLE stack while the CPU is not performing operations related to the S2-LP.

The ST-SigFox library continuously calls the `ST_LOWLEVEL_WaitForInterrupt` function whenever it is locked in a wait (for an event) state. Thus this function can be exploited to tick the BLE stack and the application in order to prevent disconnection of the paired BLE devices.

The following function is continuously called at two points of the application:

```
void BLE_Activity(void)
{
  /* BLE Stack Tick */
  BTLE_StackTick();

  /* Application Tick */
  APP_Tick();
}
```

One point is the `while(1)` main loop and the second is the following wait for interrupt function:

```
void ST_LOWLEVEL_WaitForInterrupt(void){
  /* if the MCU is stuck waiting for some IRQ, keep on the BLE activity */
  BLE_Activity();
}
```

The library needs be informed when the S2LP IRQ is raised. In this case, `ST_MANUF_S2LP_Exti_CB()` must be called by the GPIO ISR. This function is implemented by the ST-SigFox library.

A possible implementation of the GPIO ISR could be:

```
void GPIO_Handler(void){
  /* assert that the interrupt comes from the S2-LP pin */
  if(GPIO_GetITPendingBit(M2S_GPIO_3_PIN))
  {
    /* inform the library that the interrupt has been raised by the S2-LP */
    ST_MANUF_S2LP_Exti_CB();


    /* clear the pending bit of the IRQ */
    GPIO_ClearITPendingBit(M2S_GPIO_3_PIN);
  }
}
```

Different actions may be performed when `ST_MANUF_S2LP_Exti_CB()` is called, according to the status of the library:

- **Transmission (uplink)**: FIFO SPI transactions are performed by `ST_MANUF_S2LP_Exti_CB()` to refill the TX FIFO. It is important to ensure that the SPI access is exclusive and that any SPI transactions performed by the main thread or by lower priority interrupts are not interrupted by this ISR. In this case, the ISR should be processed within 1 ms for RCZ2 and RCZ4 and 6 ms for RCZ1.
- **Reception (downlink)**: Register and FIFO readings are not time critical and are not directly performed by `ST_MANUF_S2LP_Exti_CB()`. The callback just sets a flag to signal the running routines in the library that the interrupt has been raised and to perform the appropriate actions.

With this implementation, it is possible to ensure that required timings are satisfied and the modulation is correctly performed.

3.5 Sample applications

3.5.1 BLE-SigFox sensor demo

The project demonstrates how BLE[a] connectivity and sigfox transactions can coexist.

You must first set up the S2-LP FKI board with a sigfox ID,PAC and KEY and register it on the sigfox backend[b]. If the board is not registered, the ID check routine will fail and LED LD1 will blink indefinitely.

This demo merges:

- the BLE sensor demo in the STSW-BLUENRG1-DK
- the push button project in the STSW-S2LP-SFX-DK.

This application provides the same BLE services as the BLE sensor demo, including the free fall characteristic. In addition to sending the BLE notification to the master, the application sends a “fee fall” message to the sigfox network.

When you press PUSH BUTTON 2, the application transmits a sigfox frame over a certain duration (up to a few seconds, depending on the RCZ) while still maintaining the BLE connection.

A BlueNRG application for iOS™ and Android™ is available, that also interacts with the BlueNRG-1 BLE sensor profile demo.

[a] see user manual UM2071 on www.st.com

[b] see user manual UM2169 on www.st.com

This app enables notification of the acceleration characteristic and displays the value on screen. A free fall event causes the app to show a yellow tringle on your mobile device, while the sigfox backend terminal shows a message with the hexadecimal representation of the “free fall” message.

Figure 10: Free fall notification on BlueNRG app

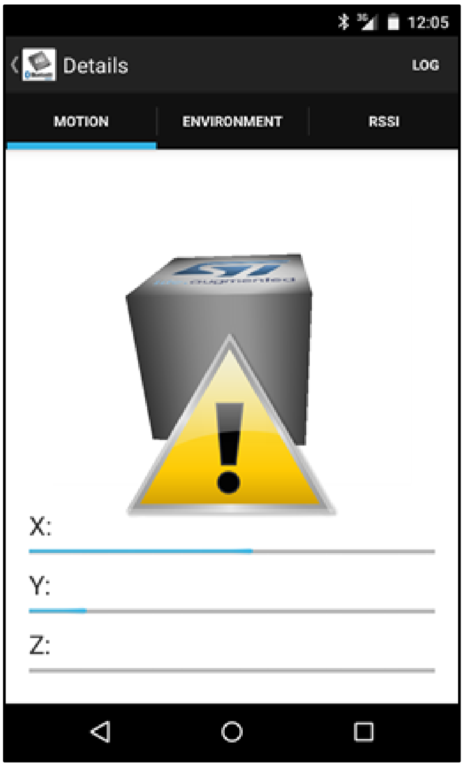


Figure 11: Free fall notification on sigfox backend

Time	Delay (s)	Header	Data / Decoding	Location	Base station	RSSI (dBm)	SNR (dB)	Freq (MHz)	Rep	Callbacks
					2654	-136.00	15.57	868.1059	2	
2017-03-16 12:05:06	1.8	0000	667265652066616c6c		25DF	-127.00	23.94	868.1092	2	
					22AA	-135.00	16.41	868.1099	2	

The *sensor.c* file contains the sensor management (initialization, reading and free fall detection) logic. It is very similar to *BLE_SensorDemo* in the BlueNRG-1 DK, except that the SPI acceleration reading (sensor LSM6DS3) is performed in a critical section to avoid competition on the SPI bus between the S2-LP management (performed in the GPIO ISR) and the sensor itself.

The `s2lp_irq` function is shown below:

```
void s2lp_irq(FunctionalState en)
{
  NVIC_InitType NVIC_InitStructure;

  /* Set the GPIO interrupt priority and enable/disable it */
  NVIC_InitStructure.NVIC_IRQChannel = GPIO_IRQn;
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = LOW_PRIORITY;
  NVIC_InitStructure.NVIC_IRQChannelCmd = en;
  NVIC_Init(&NVIC_InitStructure);
}
```

This function is used to disable (enable) the GPIO interrupt on the core immediately before (after) the accelerometer read:

```
/* Disable the IRQ from S2-LP */
s2lp_irq(DISABLE);
  /* Get Acceleration data */
  if(GetAccAxesRaw(&acc_data) == IMU_6AXES_OK) {
    Acc_Update(&acc_data);
  }
  /* Get free fall status */
  GetFreeFallStatus();
  /* Enable the IRQ from S2-LP */
  s2lp_irq(ENABLE);
```

This ensures that the `ST_MANUF_S2LP_Exti_CB()` callback is called while the SPI bus is free, without disrupting sensor reading and avoiding competition on the SPI bus.

Temperature and pressure sensor data are emulated because PUSH BUTTON 2 shares the I²C data line and asynchronous activation of this button may interfere with real LPS25HB sensor data reading.

You can enable real sensor data reading (*gatt_db.c* file) if you do not plan to use PUSH BUTTON 2.

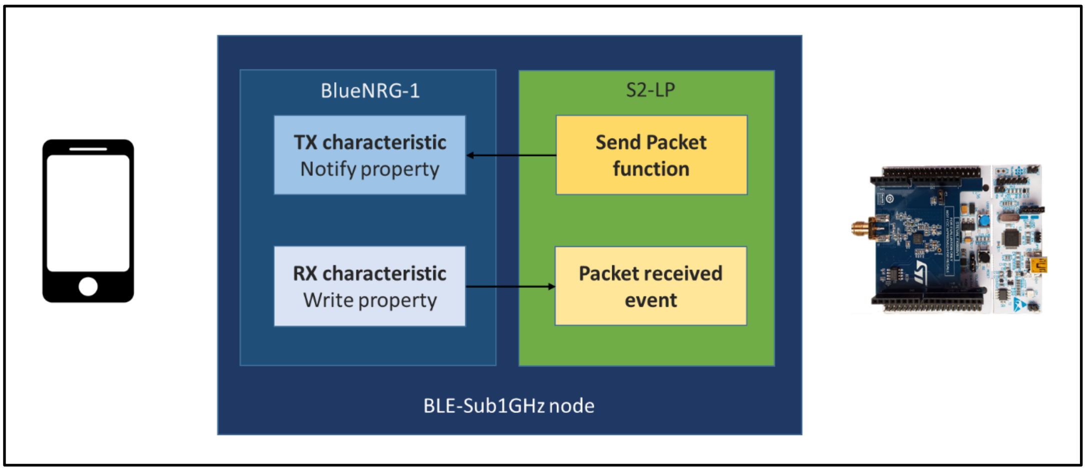
You can force the application to use the public key (for testing purposes or usage with the sigfox SNEK dongle) by:

1. pressing and holding PUSH BUTTON 2
2. pressing the board RESET button
3. releasing PUSH BUTTON 2

3.5.2 Dual Radio Chat

This application allows message exchange between mobile devices and an S2-LP node via the BLE-Sub1GHz bridge.

Figure 12: Dual Radio Chat block diagram



The Bluetooth part exposes a single chat service with the following (20-byte max.) characteristics:

- **TX characteristic** through which the client can enable notifications; when the server has data to be sent, it sends notifications with the value of the TX characteristic. The data coming from the S2-LP is received in a continuous stream.
- **RX characteristic** (writable); when the client (mobile device) has data to be sent to the server, it writes a value in this characteristic, which forwards it on to the S2-LP.

Table 3: S2-LP radio configuration

Parameter	Value
Frequency	868MHz
Modulation	2-FSK
Datarate	38.4 kbps
F. Dev.	20 kHz
Ch. Filter	100 kHz

Table 4: S2-LP packet configuration

Parameter	Value
Packet type	BASIC
Preamble length	4 bytes
SYNC	0x88888888
Whitening	Enabled
FEC	Disabled
CRC	0x07 (1 byte)
Length mode	Variable (2bytes)

This is the same default configuration used by the S2-LP DK GUI, so it is possible to transmit packets through the GUI and receive them with this application.

The B-BLE application can run on your mobile client device to send and receive messages; you can use any similar generic app able to discover the characteristic and enable notification writing and reception.

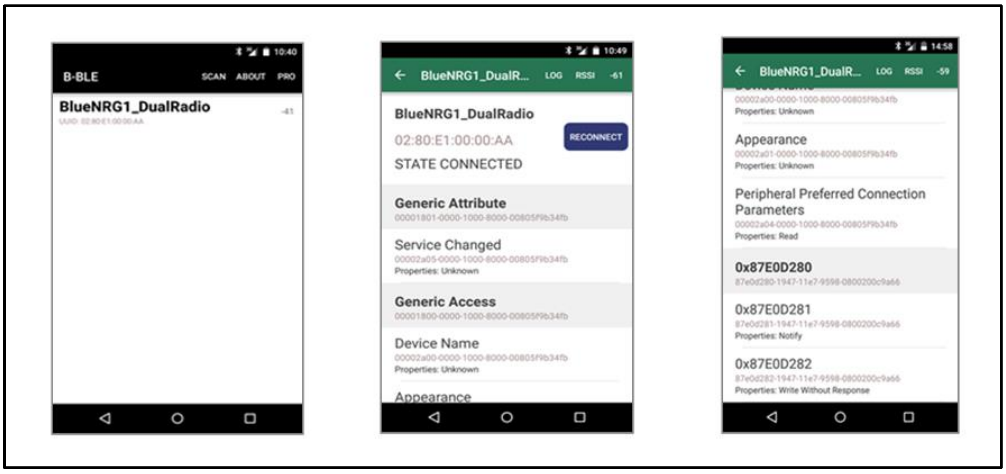
3.5.2.1 Communicating between the BLE-Sub1GHz node and a mobile device

This procedure connects your mobile to the BLE-Sub1GHz node.

1. Run the B-BLE app
2. Select the BlueNRG1_DualRadio device

3 Scroll down to explore the service (0x87E0D280) and the TX (0x87E0D281) and RX (0x87E0D282) characteristics.

Figure 13: BlueNRG1_DualRadio service and characteristics



4 Select the 0x87E0D281 characteristic

Select this characteristic to transmit a message from the S2-LP to the mobile device

A new page for the characteristic opens

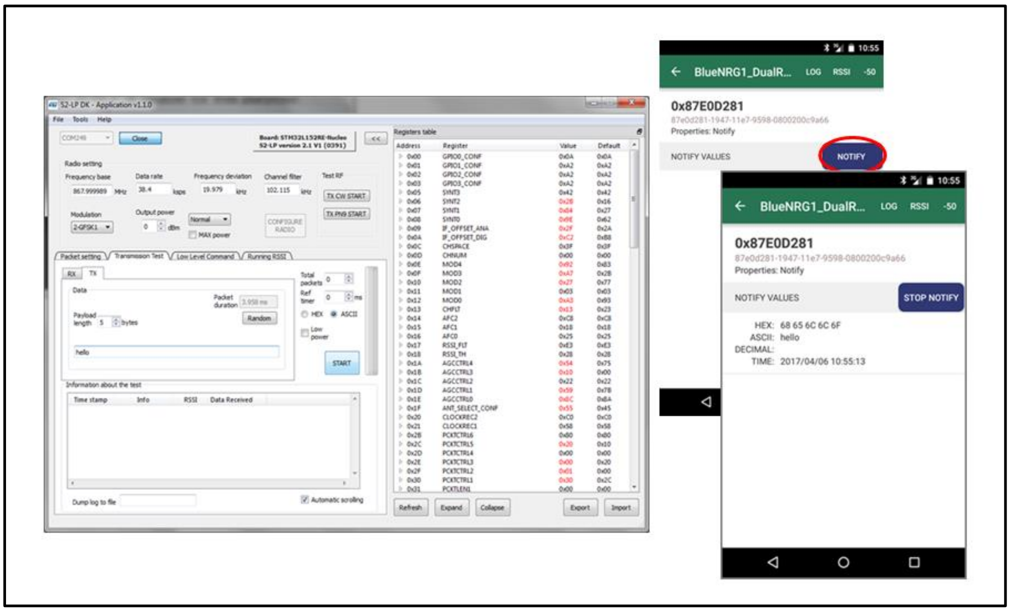
5 Click NOTIFY on the new page.

6 Run the S2-LP GUI and connect to a STEVAL-FKI868V1 or STEVAL-FKI915V1 kit.

7 Go to the TX tab, write a message and click on the START button.

The app displays the message on your phone.

Figure 14: S2-LP GUI to mobile app transmission



8 Return to the B-BLE app page listing the BlueNRG1_DualRadio service and characteristics.

9 Select the 0x87E0D282 characteristic

Select this characteristic to transmit a message from your mobile device to the S2-LP.

10 click WRITE on the new page.

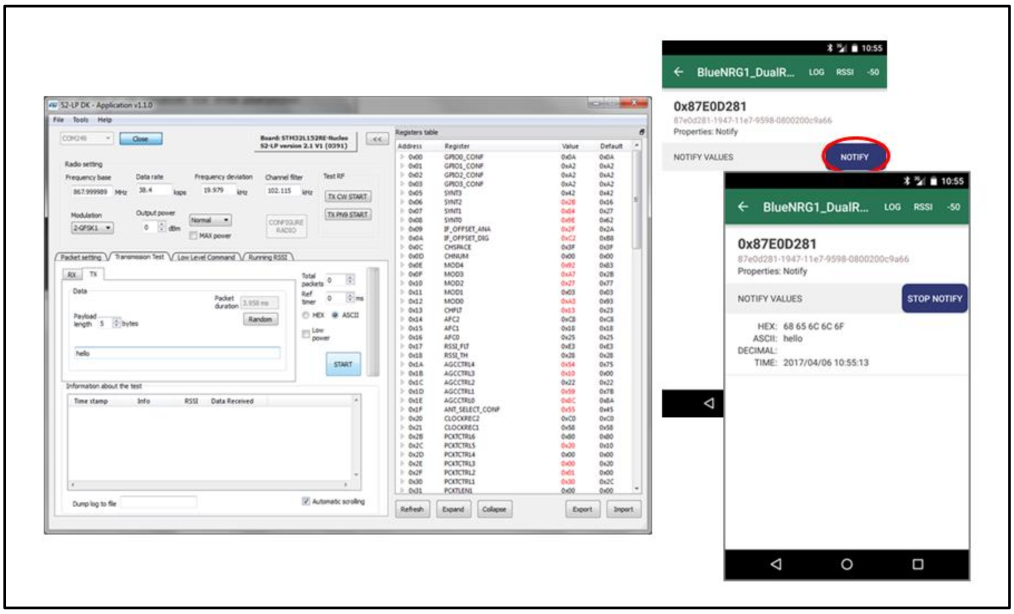
11 Open the S2-LP GUI and go to the RX TAB

12 Zero out the RX timeout and press the START button.

13 Select the Text Format, type a message in the Write Data field and click on the SEND button.

The message will be visible on the S2-LP GUI.

Figure 15: Mobile app to S2-LP GUI transmission



3.5.3 S2-LP communication

This simple application shows how to perform transmission and reception by driving the S2-LP with the BlueNRG-1 as a core.

It does not use the BLE stack as its only purpose is to demonstrate simple S2-LP management to exchange packets. The S2-LP library is used to configure and to manage the S2-LP.

Table 5: S2-LP radio configuration

Parameter	Value
Frequency	868MHz
Modulation	2-FSK
Datarate	38.4 kbps
F. Dev.	20 kHz
Ch. Filter	100 kHz

Table 6: S2-LP packet configuration

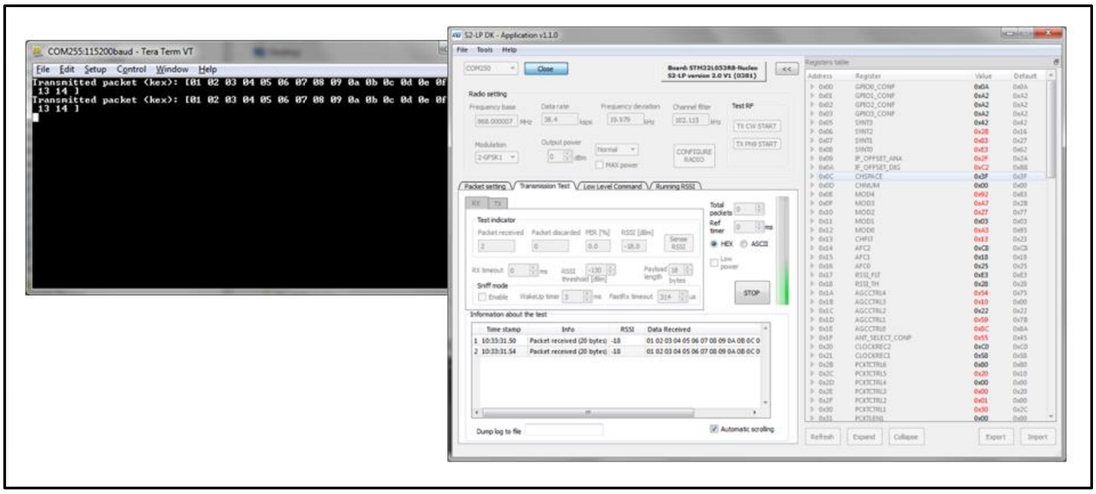
Parameter	Value
Packet type	BASIC
Preamble length	4 bytes
SYNC	0x88888888
Whitening	Enabled
FEC	Disabled
CRC	0x07 (1 byte)
Length mode	Variable (2bytes)

The node is set to continuous RX, with each received packet payload printed directly to the serial terminal.

When the STEVAL-IDB007V1 PUSH BUTTON 2 is pressed, a packet is transmitted by the device and the RX state is recovered to prepare for the next reception.

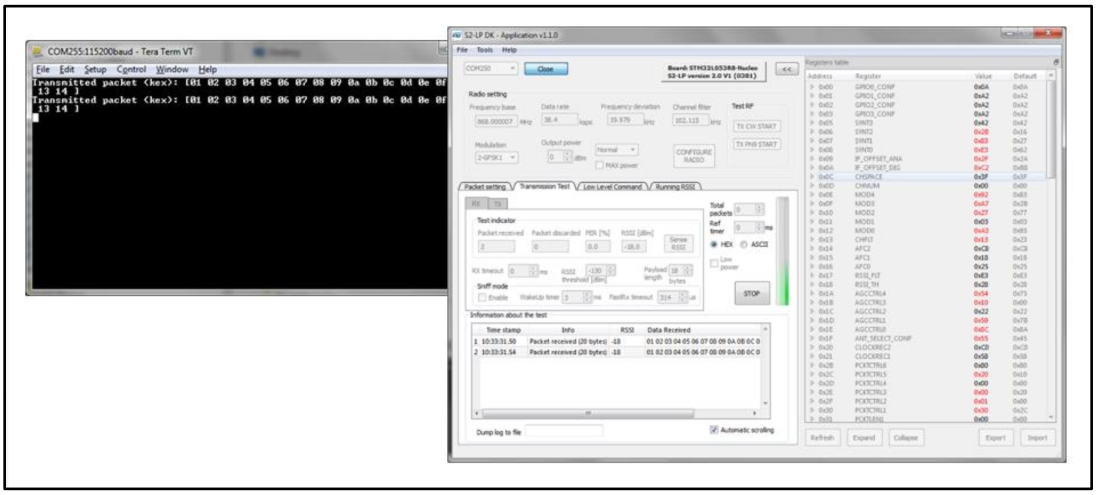
The configuration is the same as the default S2-LP DK GUI configuration, so you can transmit packets via the GUI and receive them in this application.

Figure 16: Receive at the node



Similarly, you can transmit packets from the application and receive them via the GUI.

Figure 17: Transmit to the node



4 References

All of the listed resources are available on www.st.com

1. User Manual UM2071 - *BlueNRG-1 Development Kit*
2. User Manual UM2149 - *Getting started with the S2-LP Kit*
3. User Manual UM2169 - *Getting started with the S2-LP SigFox Kit*
4. User Manual UM2173 - *Getting started with the S2-LP SigFox firmware*

5 Revision history

Table 7: Document revision history

Date	Version	Changes
02-May-2017	1	Initial release.