

---

## BLE-Sub1 GHz development kit

### Introduction

The BLE-Sub1 GHz development kit uses the BlueNRG-1/2 system-on-chip for Bluetooth® low energy (BLE) functionality and the S2-LP transceiver for sub-1 GHz functionality.

This package is for applications aiming to implement the BLE and sub-1 GHz protocols with the corresponding BlueNRG-1/2 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-1/2 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-1/2 hardware and software development kit, and [STSW-S2LP-DK](#) and [STSW-S2LP-SFX-DK](#) literature regarding the S2-LP hardware and software development kit.

# 1 Getting started

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

## 1.1 Software

Download the BLE-Sub1 GHz software package (STSW-BNRG-S2LP-DK) from [www.st.com](http://www.st.com) and extract BLE-Sub1 GHz DK-Setup-x.x.x.zip contents to a temporary directory.

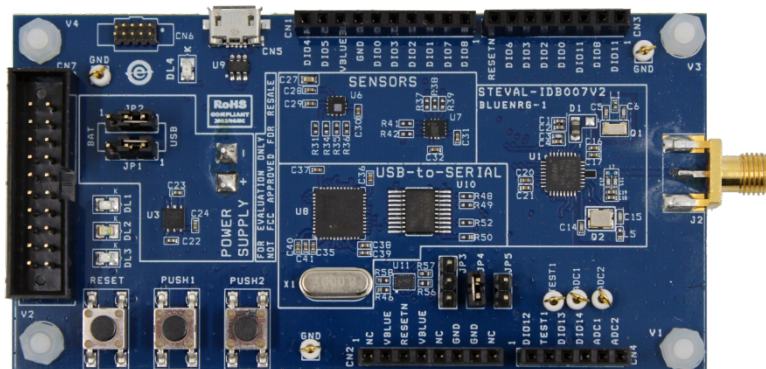
Launch BLE-Sub1 GHz DK-Setup-x.x.x.exe and follow the on-screen instructions.

## 1.2 Hardware

To run the demo of this package, one of the following is necessary:

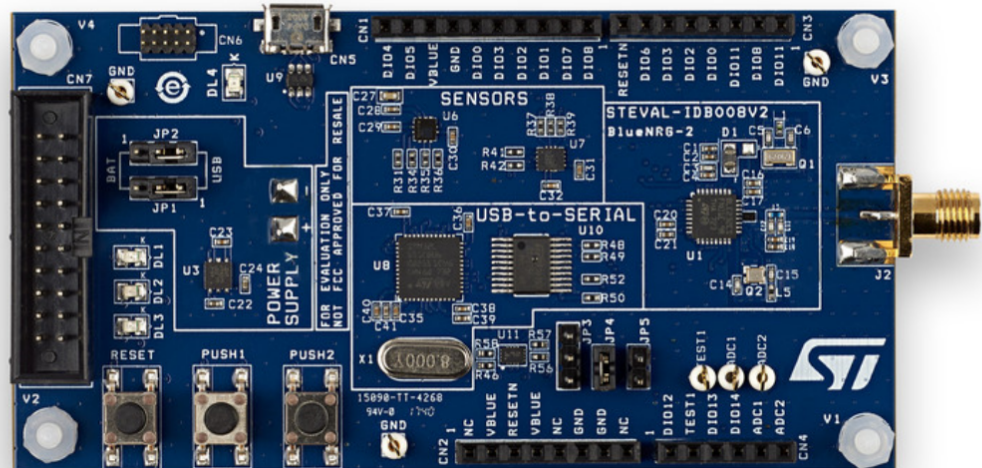
- a BlueNRG-1 STEVAL-IDB007V2 kit (refer to UM2071 on [www.st.com](http://www.st.com) for details).

Figure 1. STEVAL-IDB007V2 board



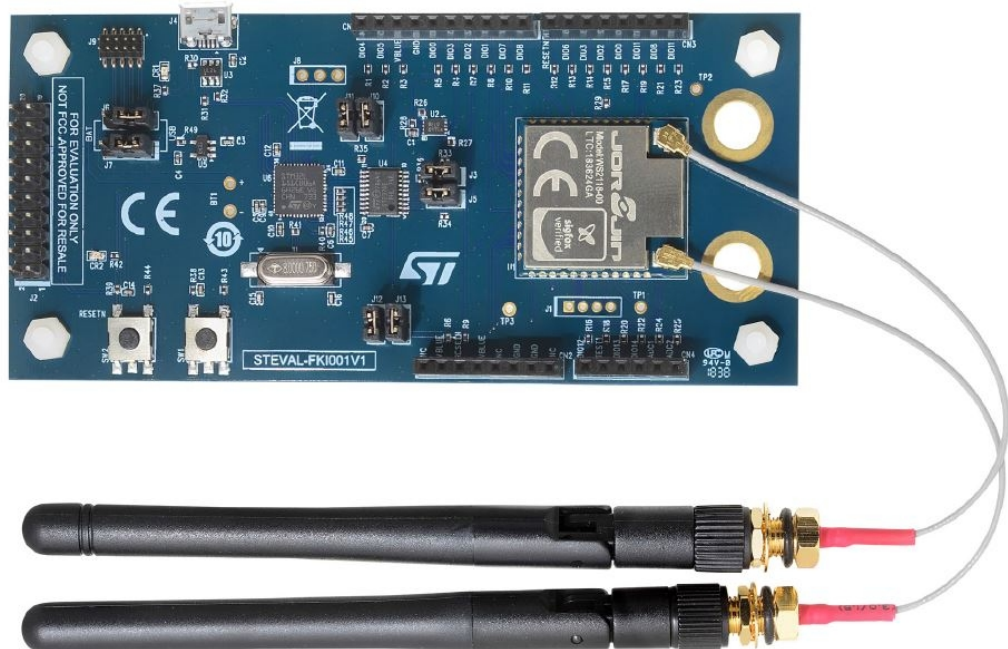
- a BlueNRG-2 STEVAL-IDB008V2 kit (refer to UM2071 on [www.st.com](http://www.st.com) for details).

Figure 2. STEVAL-IDB008V2 board



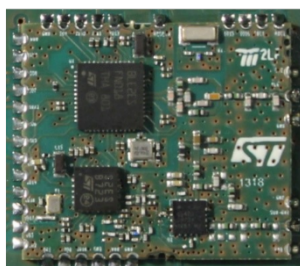
- a BlueNRG-1 STEVAL-FKI001V1 kit (refer to DB3740 on [www.st.com](http://www.st.com) for details).

Figure 3. STEVAL-FKI001V1



- The STDES-MONARCH Sigfox Monarch reference design see [Section 1.2.3 STDES-MONARCH Sigfox Monarch reference design](#).

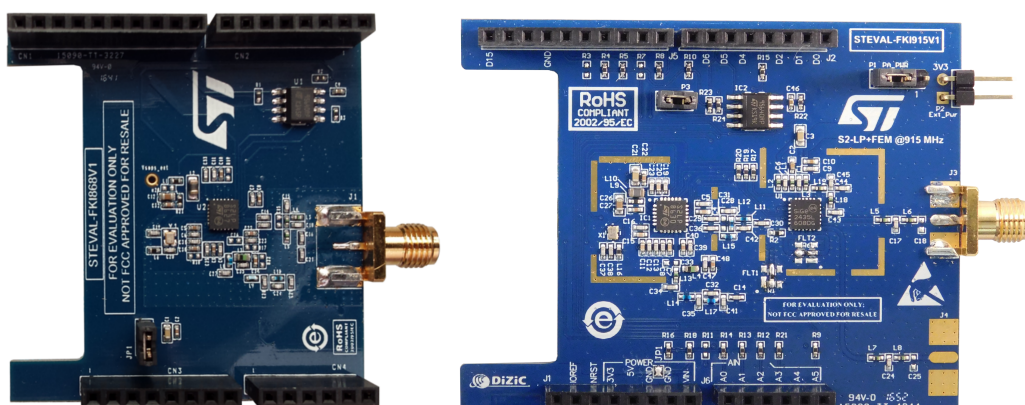
Figure 4. STDES-MONARCH



Together with the STEVAL-IDB007V2 or the STEVAL-IDB008V2, you also need:

- an S2-LP STEVAL-FKI868V2 or STEVAL-FKI915V1 kit, depending on the operating band (see UM2149 on [www.st.com](http://www.st.com) for details).

Figure 5. STEVAL-FKI868V2 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](http://www.st.com) for more details.

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

S2-LP board	Radio configuration zone
STEVAL-FKI868V2	RC1, RC3, RC5, RC6
STEVAL-FKI915V1	RC2, RC4

### 1.2.1 Modifying the STEVAL-IDB007V2 or STEVAL-IDB008V2 kit

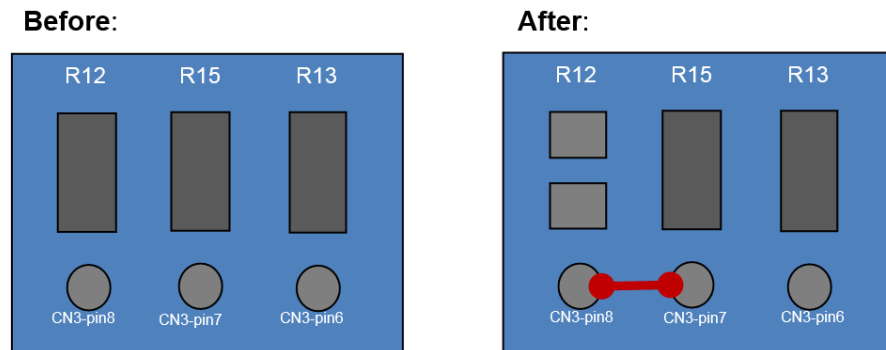
To render the STEVAL-IDB007V2 or the STEVAL-IDB008V2 compatible with the STEVAL-FKI868V2 and STEVAL-FKI915V1 boards, perform these modifications on the **rear side of the board**.

**Step 1.** Remove R12



**Step 2.** Create a short-circuit between pin 8 and pin 7 on the CN3 connector

**Figure 6. CN3 alterations**

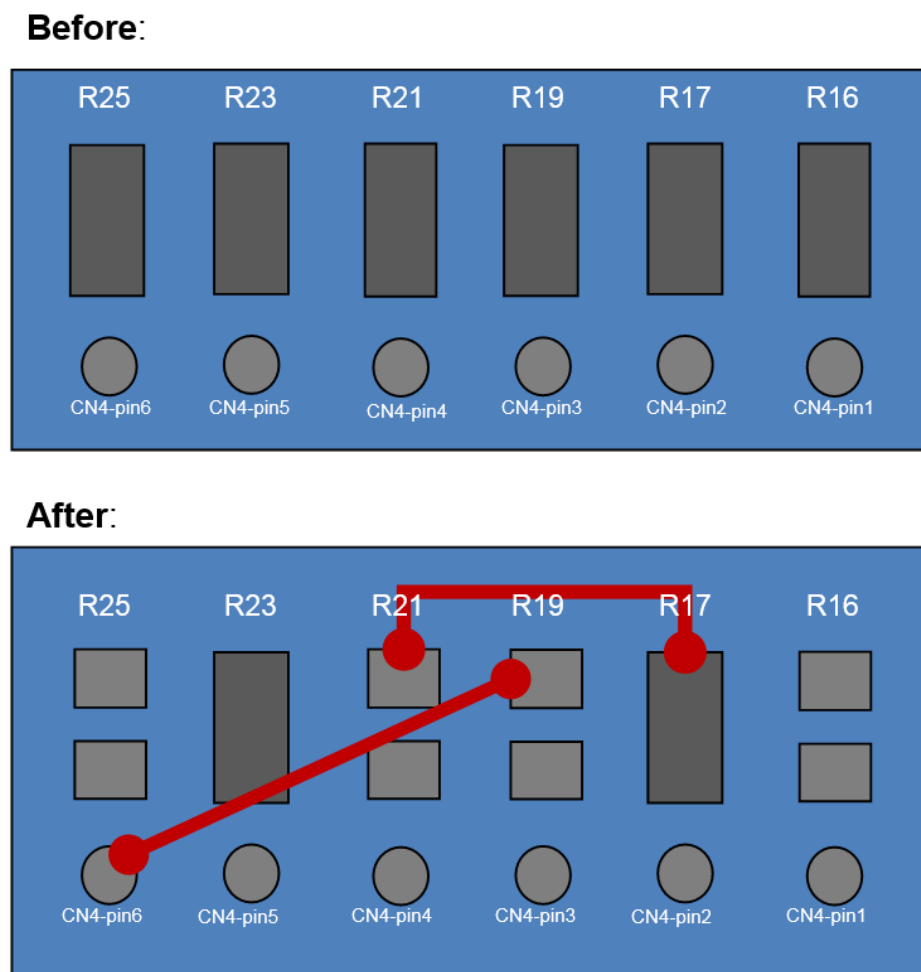


**Step 3.** Remove R25, R21, R19 and R16

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

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

**Figure 7. CN4 alterations**



### 1.2.2 BlueNRG-1/2 SoC connections

The platform pin connection of the STEVAL-IDB007V2 or the STEVAL-IDB008V2 is given in the following table.

*Note: The information refers to a STEVAL-IDB007V2 or the STEVAL-IDB008V2 board that has been modified for compatibility with the STEVAL-FKI868V2 and STEVAL-FKI915V1 boards.*

**Table 2. Platform pin description with board function**

Pin name	Pin n	Function										
		LEDs	S2-LP	Buttons	FKI_E2PROM	Pressure sensor	3D accelerometer and gyroscope	JTAG	Arduino connectors			
									CN1	CN2	CN3	CN4
DIO10	1							JTMS-SWTDIO				
DIO9	2							JTCK-SWTK				
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 STDES-MONARCH Sigfox Monarch reference design

The reference hardware for Sigfox Monarch uses the BlueNRG-2 for the Bluetooth® low energy (BLE) part and the S2-LP for the Sub-GHz. The aim of this module is to provide the compatibility of Sigfox communication for every RC zone supported from Sigfox. To satisfy this requirement the design mounts a front-end module (FEM) which integrates a power amplifier and a low noise amplifier (LNA). In particular, the power amplifier (PA) can be selectively turned on or off depending on the RC configuration.

**Figure 8. STEVAL-Monarch reference design**



The device shown above should be mounted into a motherboard, which must provide the power supply and two antenna/RF connectors for both BLE and the Sub-GHz part.

The STDES-MONARCH Sigfox reference design is fully certified for Sigfox Monarch operation for all the supported zones currently (RC1 to RC6). Based on Sigfox verified program, a module maker can freely cut and paste the reference design (schematic, BOM and board manufacturing specifications) in order to maintain the certification status.

The reference documents for such design are available in to ST website at the following address: <https://www.st.com/en/evaluation-tools/stdes-monarch.html>

The BLE-Sub1 GHz development kit includes also the software API to enable the Sigfox Monarch features. The latter is demonstrated by the demo BLE-Sigfox Monarch scan sensor demo described into [Section 1.2.4 S2-LP](#).

### 1.2.4 S2-LP

The S2-LP embedded on the STEVAL-FKI868V2 or STEVAL-FKI915V1 board connected to the STEVAL-IDB007V2 or the STEVAL-IDB008V2 can be driven by the BlueNRG-1/2 via SPI.

GPIO3 is connected to a BlueNRG-1/2 wake-up pin to signal certain events to the BlueNRG-1/2.

The BlueNRG-1/2 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-1 GHz channels.

### 1.2.5 E2PROM

The E2PROM containing the manufacturing data of the S2-LP board can be accessed by the BlueNRG-1/2 using the SPI bus.

### 1.2.6 Hardware setup

1. Connect a 2.4 GHz antenna to the SMA connector
2. Connect an 868 or 915 MHz antenna to the STEVAL-FKI868V2 or STEVAL-FKI915V1, or to the Sub-1 GHz SMA connector of the STEVAL-FKI001V1 or STDES-MONARCH
3. Ensure the jumper configuration on the board is as in s 0
4. Connect the motherboard to the PC with a USB cable
5. Verify that PWR LED is on

## 2 BLE-Sub1 GHz navigator overview

The BLE-sub1 GHz navigator application in the software package is an interactive, simple and user friendly graphical user interface for browsing the following BLE-Sub1 GHz DK SW package resources:

- BLE-Sub1 GHz examples
- Board description and related changes
- BLE-Sub1 GHz development kits
- Release notes
- License files

This GUI lets you directly download and run the selected prebuilt application binary image on the BlueNRG-1/2 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-Sub1 GHz navigator icon in

- Start → STMicroelectronics → BLE-Sub1 GHz DK X.X.X → BLE-Sub1 GHz navigator

Figure 9. BLE-Sub1 GHz navigator



The BLE-Sub1 GHz 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 10. 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-1/2 platform connected to a PC USB port.

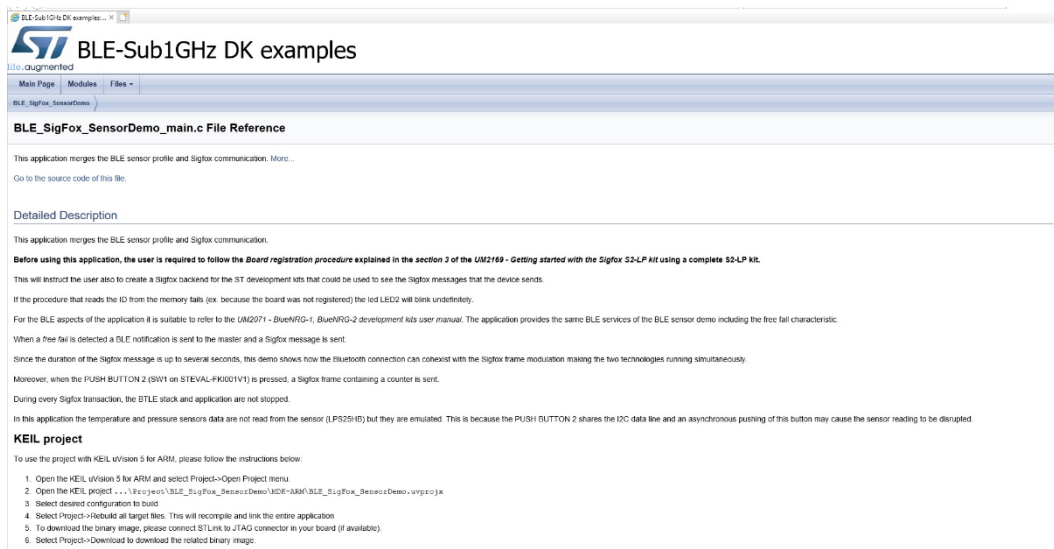
Figure 11. BLE-Sigfox sensor demo application flashing



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



Figure 12. 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-Sub1 GHz DK kit platforms and resources.

Figure 13. STEVAL-IDB007V2/DB008V2 plus FKI868V1 kit components



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

## 3 Programming with BlueNRG-1/2 system-on-chip

The BlueNRG-1/2 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-1/2 SoC peripherals and resources (ADC, GPIO, I<sup>2</sup>C, 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-Sub1 GHz navigator PC application.
- **Documents:** contain the BLE-Sub1 GHz demo applications doxygen documentation DK release notes and license file.
- **Binaries:** contain prebuilt binary applications
- **Firmware:**
  - **S2LP\_Library:** contains the generic library for the S2-LP
  - **BSP**
    - **SDK\_Eval\_BlueNRG:** SDK drivers providing an API interface to the BlueNRG-1/2 platform hardware resources (LEDs, buttons, sensors, I/O channel)
  - **MCU**
    - **BlueNRG1\_Periph\_Driver:** BlueNRG-1/2 drivers for device peripherals (adc, clock, dma, flash, gpio, I<sup>2</sup>C, timers, rtc, spi, uart and watchdog).
    - **CMSIS:** BlueNRG-1/2 CMSIS files
    - **HAL\_BLUENRG:** Hardware abstraction level APIs for abstracting some BlueNRG-1/2 hardware features (sleep modes, clock based on SysTick, etc.)
  - **S2LP\_Middleware:** SDK drivers providing an API interface to the S2LP platform hardware resources (S2-LP SPI and GPIOs, E2PROM)
- **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
  - **Sigfox:** contains the libraries for the Sigfox protocol on the BlueNRG-1/2 (ARM® Cortex®-M0 core)
- **Projects**
  - **BLE\_SigFox\_SensorDemo:** sources and projects files of the BLE\_Sigfox\_SensorDemo.
  - **BLE\_Sigfox\_MonarchScanSensorDemo:** sources and project files of the BLE\_Sigfox\_MonarchScanSensorDemo
  - **Dual\_Radio\_Chat:** an application for communication between a mobile device and an S2-LP using BLE-Sub1 GHz 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

For the BLE\_Sigfox\_SensorDemo and BLE\_Sigfox\_MonarchScanSensorDemo demo applications, you have to register your STEVAL-FKI868V2 or STEVAL-FKI915V1 board with a Sigfox ID/PAC/KEY before using it with the STEVAL-IDB007V2 or the STEVAL-IDB008V2 (see user manual UM2169 on [www.st.com](http://www.st.com)).

### 3.3 Initialization sequence

This procedure lets you develop an application for Sigfox on the STEVAL-FKI868V2 or STEVAL-FKI915V1 platforms.

- Step 1.** Remap your hardware configuration modifying one of the `Platform_Configuration_x.h` files according to your needs and rename it `USER_Platform_Configuration.h`
- Step 2.** Set the preprocessor define: `USER_DEFINED_PLATFORM=USER_EVAL_PLATFORM`
- Step 3.** Call the `ST_Init()` function
- Step 4.** Call `aci_hal_write_config_data()` API to configure BlueNRG-1/2 public address (if public address is used)
- Step 5.** Call `aci_gatt_init()` API to initialize the BLE GATT layer
- Step 6.** Call `aci_gap_init("role")` API to initialize the BLE GAP layer depending on the selected device role
- Step 7.** 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).
- Step 8.** 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
- Step 9.** Call `BlueNRG_Stack_Initialization(&BlueNRG_Stack_Init_params)` to initialize the BLE stack.
- Step 10.** Initialize the S2-LP SPI and S2-LP SDN pin setting it high.
- Step 11.** Call the `ST_Sigfox_Init()` function
- Step 12.** 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
YourCustomRoutine();
BlueNRG_Sleep(...); - enables BlueNRG-1/2 sleep mode and preserves the BLE radio operating modes
}
```
- Step 13.** 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,2,0);
```
- with downlink request:
- ```
- ST_SIGFOX_API_send_frame(customer_data,4,customer_resp,2,1);
```
- 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 `mcu_api_bluenrg1.c` module in `Projects/Firmware/Libraries/Sigfox/ST_API/src` interfaces the ST Sigfox library with the hardware platform using:

- S2LP\_CORE\_SPI, S2LP\_CORE\_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_MCU_API_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_MCU_API_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_RF_API_S2LP_IRQ_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_IRQ_PIN))
    {
        /* inform the library that the interrupt has been raised by the S2-LP */
        ST_RF_API_S2LP_IRQ_CB();

        /* clear the pending bit of the IRQ */
        GPIO_ClearITPendingBit(M2S_GPIO_IRQ_PIN);
    }
}
```

Different actions may be performed when `ST_RF_API_S2LP_IRQ_CB()` is called, according to the status of the library:

- **Transmission (uplink):** FIFO SPI transactions are performed by `ST_RF_API_S2LP_IRQ_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 RC2 and RC4 and 6 ms for RC1,3,5,6.
- **Reception (downlink):** Register and FIFO readings are not time critical and are not directly performed by `ST_RF_API_S2LP_IRQ_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 (see user manual UM2071 on [www.st.com](http://www.st.com)) connectivity and Sigfox transactions can coexist.

**Note:** You must first set up the S2-LP FKI board with a sigfox ID, PAC and KEY and register it on the Sigfox backend (see user manual UM2169 on [www.st.com](http://www.st.com)). If the board is not registered, the ID check routine fails and LED DL2 blinks 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 “free fall” message to the Sigfox network.

When you press PUSH BUTTON 2 (SW1 for FKI001V1), 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.

The ST BLE sensor app for iOS™ and Android™ is available, that also interacts with the BlueNRG-1/2 BLE sensor profile demo.

This app enables notification of the acceleration characteristic and displays the value on screen. A free fall event causes the firmware to send a Sigfox message, while the Sigfox backend terminal shows the message with the hexadecimal representation of the “free fall” message.

**Figure 14. Free fall notification on BlueNRG app**

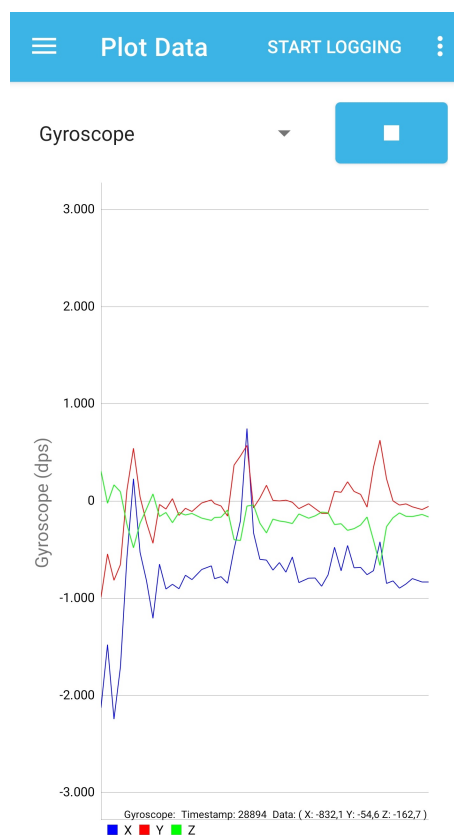




Figure 15. Free fall notification on Sigfox backend

| Time                | Delay (s) | Header | Data / Decoding    | Location | Base station | RSSI (dBm) | SNR (dB) | Freq (MHz) | Rep | Callbacks |
|---------------------|-----------|--------|--------------------|----------|--------------|------------|----------|------------|-----|-----------|
| 2017-03-16 12:05:06 | 1.8       | 0000   | 667265652066616c6c |          | 2654         | -136.00    | 15.57    | 868.1059   | 2   |           |
|                     |           |        |                    |          | 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);
}
```

Temperature and pressure sensor data are emulated because PUSH BUTTON 2 shares the I<sup>2</sup>C 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 Sigfox public key (for testing purposes or usage with the Sigfox network emulator - SNE tool) by:

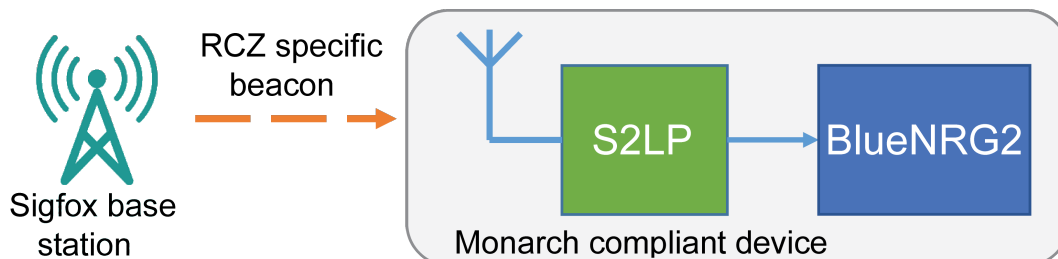
1. pressing and holding PUSH BUTTON 2
2. pressing the board RESET button
3. releasing PUSH BUTTON 2

### 3.5.2

#### BLE-Sigfox Monarch scan sensor demo

The project increases the BLE-Sigfox sensor demo with the Sigfox Monarch feature to be used in conjunction with the reference design described in [Section 1.2.3 STDES-MONARCH Sigfox Monarch reference design](#). The aim of the Monarch feature is to provide a way for Sigfox devices to know what kind of Sigfox radio configuration (RC) to apply wherever they are in the world. These features are enabled by means of specific beacons that base station sends periodically (5 min) with a duration of 400 ms to Sigfox devices. Such beacons are demodulated by the S2-LP and then processed by the BlueNRG-2 to establish the correct RCZ, see figure below.

Figure 16. Beacon demodulation



#### Sigfox Monarch scan demo main flow

If a Sigfox node is moving through the world, it cannot transmit any information while it is not aware about the current zone. For this reason, in this demo there is a start-up phase in which the node starts to scan for Monarch beacons. During this time the board cannot send any information to Sigfox base stations while BLE capability are instead available. The monarch scan session is triggered by means of the following API function.

```
sfx_error_t SIGFOX_MONARCH_API_execute_rc_scan (
sfx_u8 rc_capabilities_bit_mask,
sfx_u16 timer,
sfx_timer_unit_enum_t unit,
sfx_u8 (* app_callback_handler) (sfx_u8 rc_bit_mask, sfx_s16 rssi) ).
```

where **timer** and **unit** specify a time unit in terms of millisecond, seconds, minutes or hours. The function returns the **rc\_bit\_mask** value corresponding to the beacon found and its rssi level by means of the callback **app\_callback\_handler** passed as parameter. To inform the Sigfox library about the device supported RC zones a bitmask is passed through the 8-bit parameter named **rc\_capabilities\_bit\_mask**.

In the demo the function explained above is called into the main function as follows:

```
/* Begin Monarch Scan session for 5 min + 10 sec. */
ret=SIGFOX_MONARCH_API_execute_rc_scan (63, 310, SFX_TIME_S,
monarch_callback_for_found);
```

In the latter the monarch scan has been set for about 5 minutes, considering the periodicity of Monarch beacons. The **rc\_capabilities\_bit\_mask** is set to 63 (111111) in order to inform the library that the board can manage all RCs (from 1 to 6). Once the 5 minute timeout expiration, or after that the board recognizes the RCZ, the following callback is called.

```

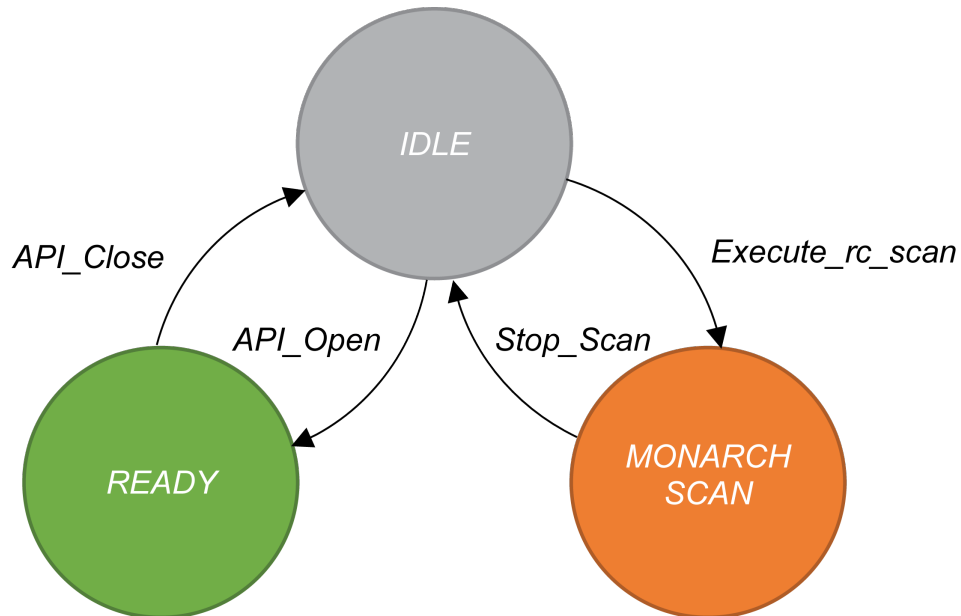
sfx_u8 monarch_callback_for_found(sfx_u8 rc_bit_mask, sfx_s16 rssi )
{
    printf("return rc_bit_mask %d\r\n", rc_bit_mask);
    printf("return rssi %d\r\n", rssi);
    switch (rc_bit_mask)
    {
        case 0x00:
        {
            {
            }
            break;
        case 0x01: //RC1
        {
            printf("Detected RC1!!!:\r\n");
            /* Turn Power Amplifier Off - 14 dBm out */
            ST_RF_API_set_pa(0);
            detected_zone=1;
        }
        break;
        case 0x02: //RC2
        {
            printf("Detected RC2!!!:\r\n");
            /* Turn Power Amplifier On - 24 dBm out */
            ST_RF_API_set_pa(1);
            detected_zone=2;
        }
        break;
        case 0x04: //RC3a
        {
            printf("Detected RC3!!!:\r\n");
            /* Turn Power Amplifier Off - 14 dBm out */
            ST_RF_API_set_pa(0);
            detected_zone=3;
        }
        break;
        case 0x08: //RC4
        {
            printf("Detected RC4!!!:\r\n");
            /* Turn Power Amplifier On - 24 dBm out */
            ST_RF_API_set_pa(1);
            detected_zone=4;
        }
        break;
        case 0x10: //RC5
        {
            printf("Detected RC5!!!:\r\n");
            /* IN RC5 maximum ouput power is 12 dBm */
            //ST_RF_API_reduce_output_power(2);
            /* Turn Power Amplifier Off - 14 dBm out */
            ST_RF_API_set_pa(0);
            detected_zone=5;
        }
        break;
        case 0x20: //RC6
        { printf("Detected RC6!!!:\r\n");
            /* Turn Power Amplifier Off - 14 dBm out */
            ST_RF_API_set_pa(0);
            detected_zone=6;
        }
        break;
    }
    return 1;
}

```

As it should be pointed out, depending on the found configuration, the callback passes the proper value to the `ST_RF_API_set_pa` in order to activate or not the power amplifier. This decision is based on the local laws in terms of admissible output power. Further, a message will appear in the UART terminal and the devices will send a frame in the proper RCZ.

It should be pointed out that, during a Monarch scan it is not possible to send any message into the Sigfox cloud, in fact, when a Monarch session is ongoing, the device will be in the MONARCH SCAN state (see figure below) and it cannot reach the READY state (the state for transmitting any Sigfox message).

Figure 17. Sigfox state machine (partial)



To summarize, for running the BLE-SigFox Monarch scan sensor demo we assume that the monarch signal is present. After the board reset the following procedure is executed by the firmware:

1. The Monarch scan session starts for a period of 5 minutes. During this time is not possible to send Sigfox messages but BLE connectivity still working as well as for the BLE-Sigfox sensor demo
2. After a beacon detection the proper RCZ is configured (e.g. RC1)
3. The device sends a frame to a base station which reports the detected zone
4. Starting from here is possible to send message by means of button or the free fall event

#### Expected results

If the device is in the zone 1, for example, when a Monarch beacon has been received and the boards is connected to a UART terminal (via an USB to serial adapter) the 01 message (RC1) is shown in the Sigfox back-end while **Detected RC1!** message appears into the UART terminal.

Figure 18. Message in the back-end indicating a beacon in RC1

| Sigfox Network Emulator |                          |                 |                 |       |                         |
|-------------------------|--------------------------|-----------------|-----------------|-------|-------------------------|
|                         |                          | CONFIGURATION   | MESSAGES        | ABOUT | Authentication disabled |
| Device ID               | Time                     | Sequence number | Data / Decoding | LQI   | Callbacks               |
| FEDCBA98                | Jun 20, 2018 12:17:48 PM | 1460            | 01              |       |                         |

If, instead, the device is in the zone 2, in the UART terminal reports the message **Detected RC2!** At the same time into the backend the following appears (02 for RC2)-

Figure 19. Message in the back-end indicating a beacon in RC2

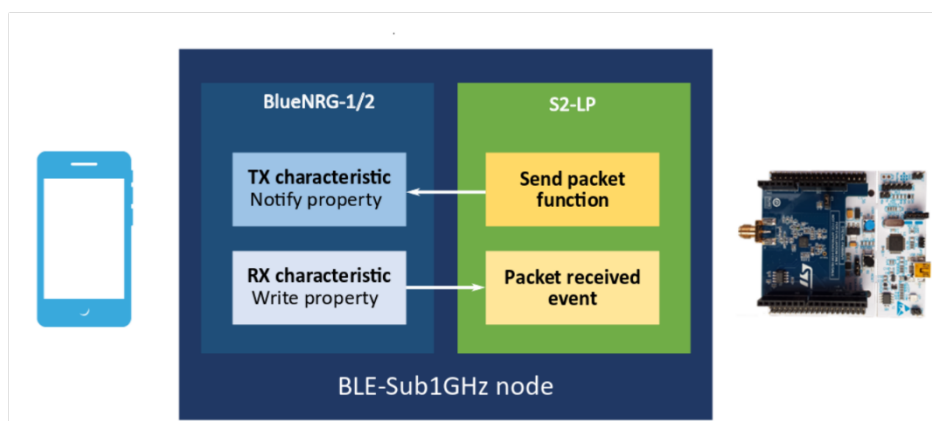
| Device ID | Time                     | Sequence number | Data / Decoding | LQI | Callbacks |
|-----------|--------------------------|-----------------|-----------------|-----|-----------|
| FEDCB998  | Jun 20, 2018 12:25:58 PM | 1406            | 02              |     |           |

If during the Monarch session no beacons have been found, the following message appears in the terminal: **Monarch Scan Timeout**. In this case, the user should reset the board to start a new scan.

### 3.5.3 Dual radio chat

This application allows message exchange between mobile devices and an S2-LP node via the BLE-Sub1 GHz bridge.

Figure 20. 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  | 868 MHz   |
| 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 (2 bytes) |

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.

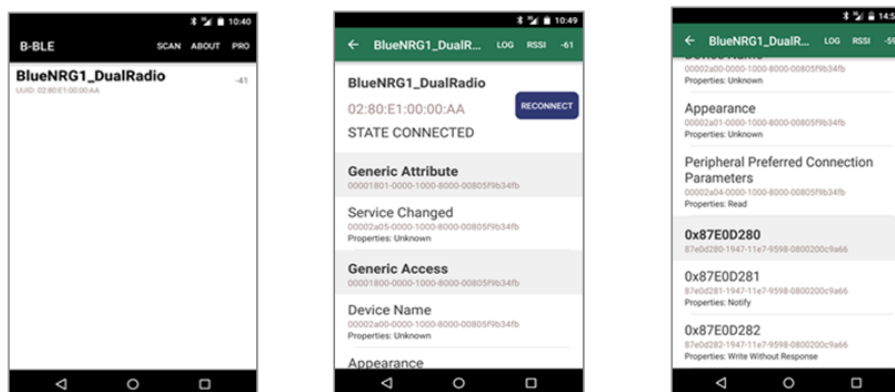
The B-BLE application can run on your mobile client device to send and receive messages; you can use any similar generic app able to discover the characteristic and enable notification writing and reception.

### 3.5.3.1

#### **Communicating between the BLE-Sub1 GHz node and a mobile device**

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

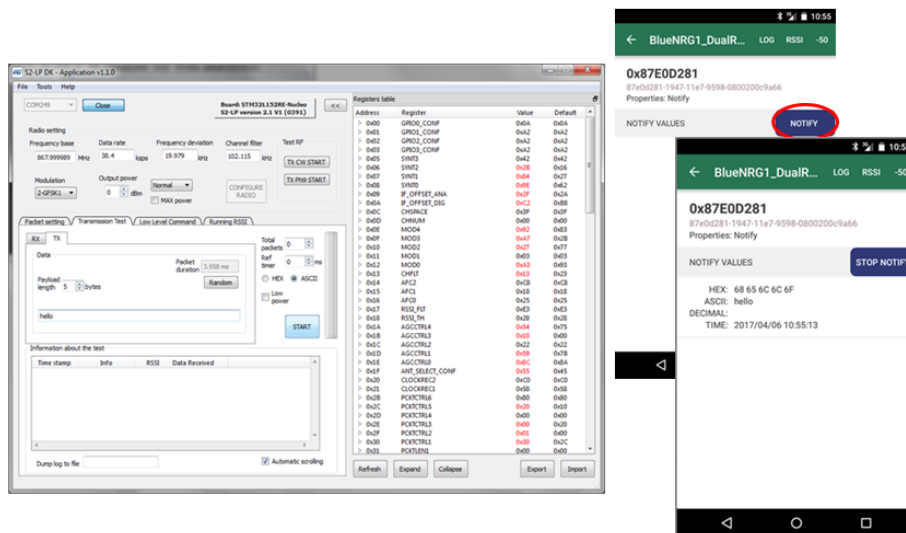
- Step 1.** Run the B-BLE app
- Step 2.** Select the BlueNRG1\_DualRadio device
- Step 3.** Scroll down to explore the service (0x87E0D280) and the TX (0x87E0D281) and RX (0x87E0D282) characteristics.

**Figure 21. BlueNRG1\_DualRadio service and characteristics**


- Step 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
- Step 5.** Click NOTIFY on the new page
- Step 6.** Run the S2-LP GUI and connect to a STEVAL-FKI868V2 or STEVAL-FKI915V1 kit

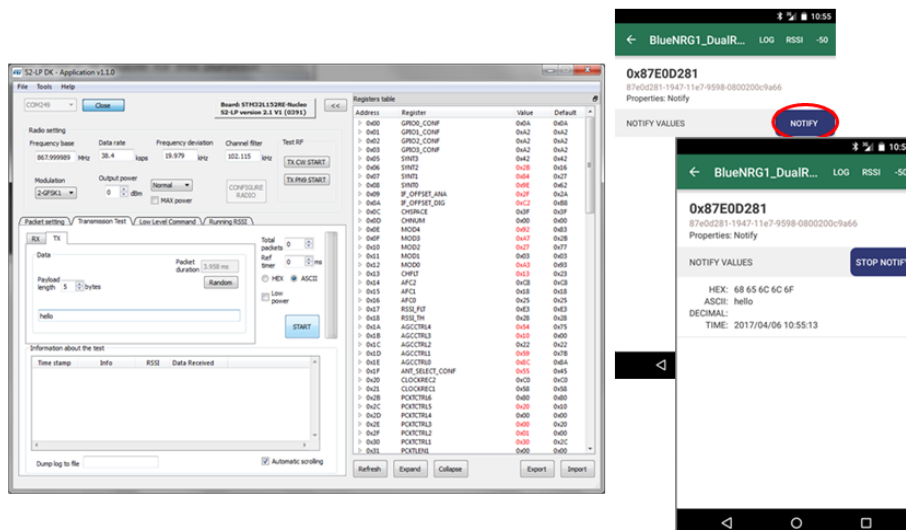
- Step 7.** Go to the TX tab, write a message and click on the START button  
The app displays the message on your phone.

**Figure 22. S2-LP GUI to mobile app transmission**



- Step 8.** Return to the B-BLE app page listing the BlueNRG1\_DualRadio service and characteristics  
**Step 9.** Select the 0x87E0D282 characteristic  
Select this characteristic to transmit a message from your mobile device to the S2-LP.  
**Step 10.** click WRITE on the new page  
**Step 11.** Open the S2-LP GUI and go to the RX TAB  
**Step 12.** Zero out the RX timeout and press the START button  
**Step 13.** Select the Text Format, type a message in the Write Data field and click on the SEND button  
The message IS visible on the S2-LP GUI.

**Figure 23. Mobile app to S2-LP GUI transmission**



### 3.5.4

#### S2-LP communication

This simple application shows how to perform transmission and reception by driving the S2-LP with the BlueNRG-1/2 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  | 868 MHz   |
| Modulation | 2-FSK     |
| Datarate   | 38.4 kbps |
| F. Dev.    | 20 kHz    |
| Ch. Filter | 100 kHz   |

**Table 6. S2-LP packet configuration**

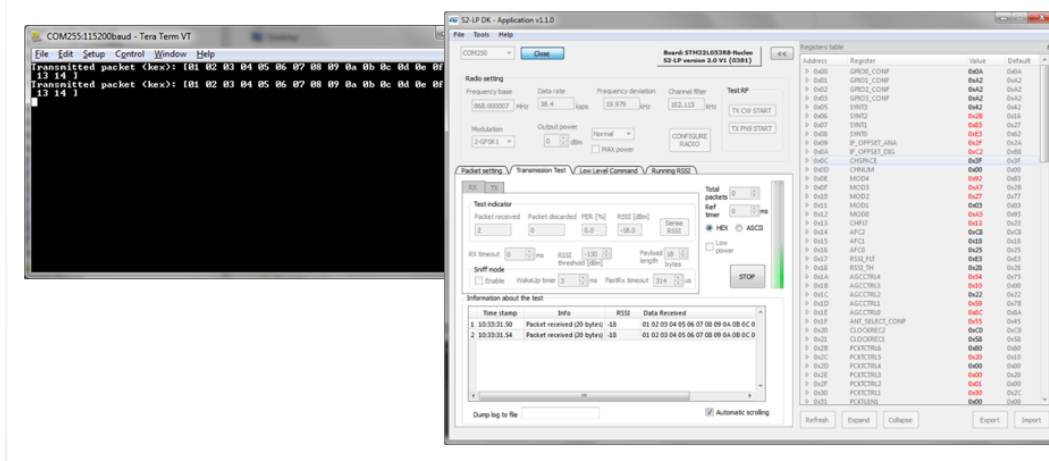
| Parameter       | Value             |
|-----------------|-------------------|
| Packet type     | BASIC             |
| Preamble length | 4 bytes           |
| SYNC            | 0x88888888        |
| Whitening       | Enabled           |
| FEC             | Disabled          |
| CRC             | 0x07 (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 PUSH BUTTON 2 (SW1 for the FK1001V1) is pressed, a packet is transmitted by the device and the RX state is recovered to prepare for the next reception.

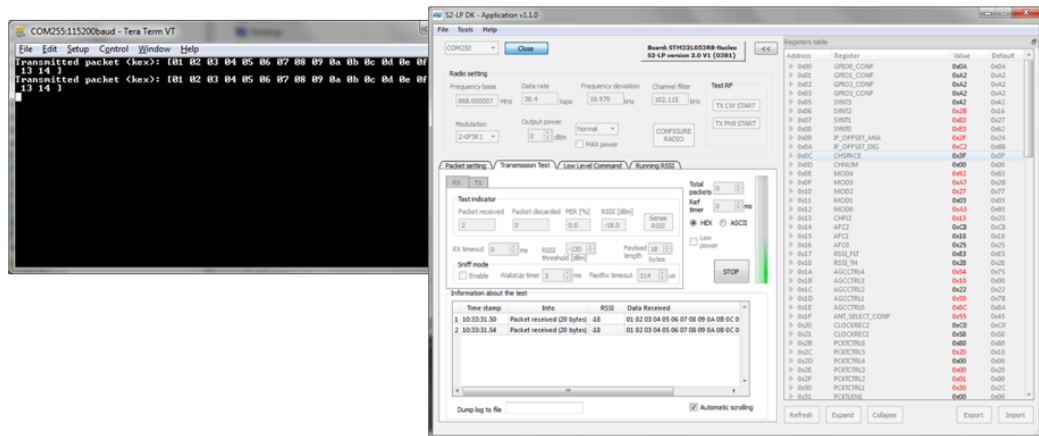
The configuration is the same as the default S2-LP DK GUI configuration, so you can transmit packets via the GUI and receive them in this application.

**Figure 24. Receive at the node**



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

Figure 25. Transmit to the node



---

## 4 References

---

All of the listed resources are available on [www.st.com](http://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*



## Revision history

**Table 7. Document revision history**

| Date        | Version | Changes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 02-May-2017 | 1       | Initial release.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 12-Nov-2018 | 2       | <p>Updated references to BlueNRG-1/2 throughout the document.</p> <p>Added <a href="#">Figure 2. STEVAL-IDB008V2 board</a>.</p> <p>Added <a href="#">Section 1.2.3 STDES-MONARCH Sigfox Monarch reference design</a>.</p> <p>Updated <a href="#">Figure 9. BLE-Sub1 GHz navigator</a>, <a href="#">Figure 10. BLE-Sigfox sensor demo application window</a>, <a href="#">Figure 11. BLE-Sigfox sensor demo application flashing</a>, <a href="#">Figure 13. STEVAL-IDB007V2/DB008V2 plus FK1868V1 kit components</a>, <a href="#">Figure 20. Dual radio chat block diagram</a>.</p> <p>Minor text changes.</p> |
| 14-Sep-2020 | 3       | <p>Updated <a href="#">Section Introduction</a>, <a href="#">Table 1. S2-LP boards for Sigfox radio configuration zones</a>, <a href="#">Section 1.2.3 STDES-MONARCH Sigfox Monarch reference design</a>, <a href="#">Section 1.2.6 Hardware setup</a>, <a href="#">Section 3.1 Software directory structure</a>, <a href="#">Section 3.4 ST LowLevel implementation</a> and <a href="#">Section 3.5.1 BLE-Sigfox sensor demo</a>.</p>                                                                                                                                                                         |

**IMPORTANT NOTICE – PLEASE READ CAREFULLY**

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to [www.st.com/trademarks](http://www.st.com/trademarks). All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2020 STMicroelectronics – All rights reserved