

---

## Getting started with the S2-LP development kits

### Introduction

This document describes the development kits for the [S2-LP](#) device and related hardware and software components.

The S2-LP is an ultra low-power, low data rate, sub-1 GHz transceiver.

The following S2-LP kits are available:

1. S2-LP development platform for 413-479 MHz (order code: [STEVAL-FKI433V2](#))
2. S2-LP development platform for 826-958 MHz (order code: [STEVAL-FKI868V2](#))
3. S2-LP development platform for 452-527 MHz (order code: [STEVAL-FKI512V1](#))
4. S2-LP development platform for 902-928 MHz with external FEM (order code: [STEVAL-FKI915V1](#))

## 1 Overview

This section describes all the software and hardware components of the [S2-LP](#) kits.

### 1.1 System requirements

The STSW-S2LP-DK application GUI has the following minimum requirements:

- PC with Intel® or AMD® processor running Windows (XP SP3, Vista, 7 or 10)
- At least 128 Mb of RAM
- USB ports
- 40 MB of available hard disk space
- Adobe Acrobat Reader 6.0 or above.

### 1.2 STSW-S2LP-DK development kit setup

Launch the S2-LP DK-Setup-X.X.X.exe file and follow the onscreen instructions.

*Note:* **EWARM Compiler 7.40.3 or later is required for building the S2LPLibrary\_Examples demonstration applications for STM32L1 and STM32L0 microcontrollers.**

**MDK-ARM Keil V5.17 or later is required for building the S2LPLibrary\_Examples demonstration applications for STM32L0 microcontroller.**

## 2 Hardware description

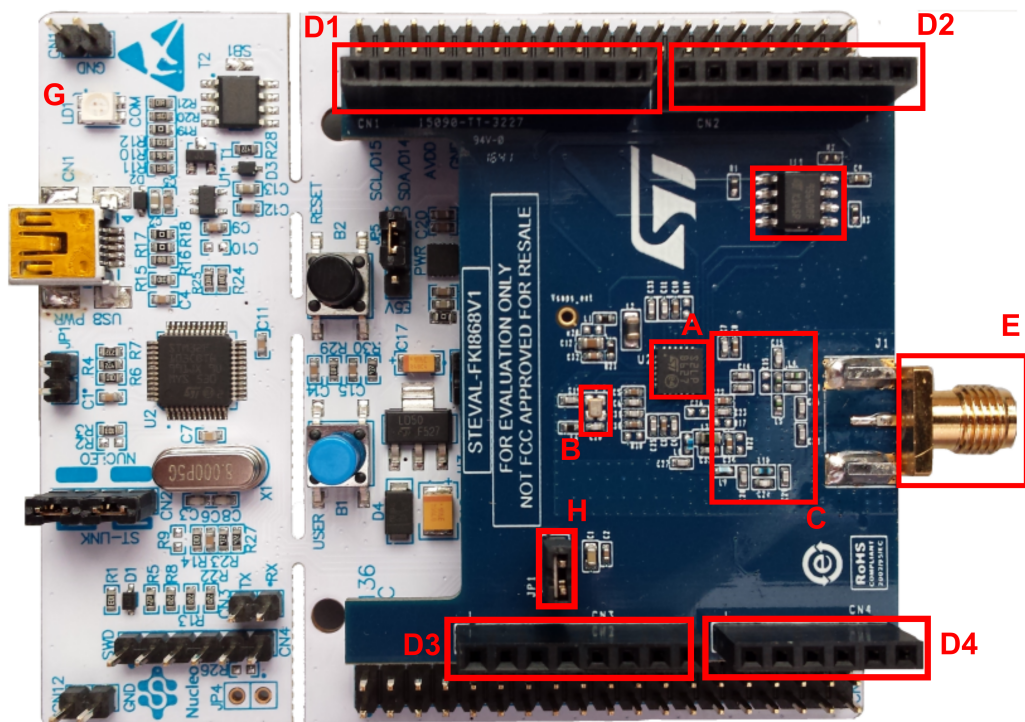
### 2.1 STEVAL-FKI868Vx, STEVAL-FKI433Vx and STEVAL-FKI512V1 evaluation boards

The [STEVAL-FKI868V2/V1](#), the [STEVAL-FKI433V2/V1](#) and [STEVAL-FKI512V1](#) evaluation boards are designed to work in the 826-958 MHz, in the 413-479 MHz and in the 452-527 MHz band, respectively.

Some features on the boards are (see [Figure 1. STEVAL-FKI868Vx evaluation board features](#)):

- S2-LP (A)
- 8 MHz high frequency crystal (B)
- Balun, matching network and harmonic filter (C)
- Two rows with Arduino compliant connectors (D1-4)
- SMA connector (E)
- An EEPROM to store the manufacturing data (F)
- A NUCLEO-L152RE or NUCLEO-L053R8 evaluation board (G)
- A jumper for S2-LP current measurement (H)

Figure 1. STEVAL-FKI868Vx evaluation board features



Pressing the reset button, the STM32 Nucleo development board resets and a hex format string containing ID and PAC is shown.

#### 2.1.1 S2-LP connections

S2-LP signal test points are split across two rows which are Arduino compliant connectors: CN1, CN3 and CN2, CN4.

The S2-LP shield is connected to the Nucleo motherboard via the Arduino compliant connectors.

The connectors and pin names below are used in the STEVAL-FKIxxxVx schematic diagram.

**Table 1. S2-LP: FKI868Vx and FKI433Vx evaluation board pin description**

Pin name	Pin number	Arduino connectors			
		CN1 (D1)	CN2 (D2)	CN3 (D3)	CN4 (D4)
VDD SMPS	1			pin 4	
SMPS1	2				
SMPS2	3				
XOUT	4				
XIN	5				
SDN	6		pin 8(SDN)		
VDDANASYNTH	7			pin 4	
VRSYNTH	8				
VREFVCO	9				
VDDVCOTX	10			pin 4	
TX	11				
VRRF	12				
RXN	13				
RXP	14				
VDDRxDIG	15			pin 4	
SDO	16	pin 5 (MISO)			
SDI	17	pin 4 (MOSI)			
SCLK	18		pin 4(SCK)		
CSN	19				pin 2 (CS)
GPIO0	20				pin 1 (GPIO0)
GPIO1	21				pin 3 (GPIO1)
GPIO2	22				pin 4 (GPIO2)
GPIO3	23				pin 6 (GPIO3)
VSMPS3	24				
GND	25	pin 7		pins 6 and 7	

The S2-LP evaluation board contains a discrete passive circuit for RF matching and balun and other additional components required by the S2-LP for proper operation.

### 2.1.2 STEVAL-FKI868Vx, STEVAL-FKI433Vx and STEVAL-FKI512V1 power

The board can be powered by the Nucleo evaluation board mini USB connector.

When the JP1 jumper is fitted (H in [Figure 1. STEVAL-FKI868Vx evaluation board features](#)), the radio section is supplied.

By removing this jumper and connecting a power meter, you can measure the [S2-LP](#) current consumption.

## 2.2 STEVAL-FKI915V1 evaluation board

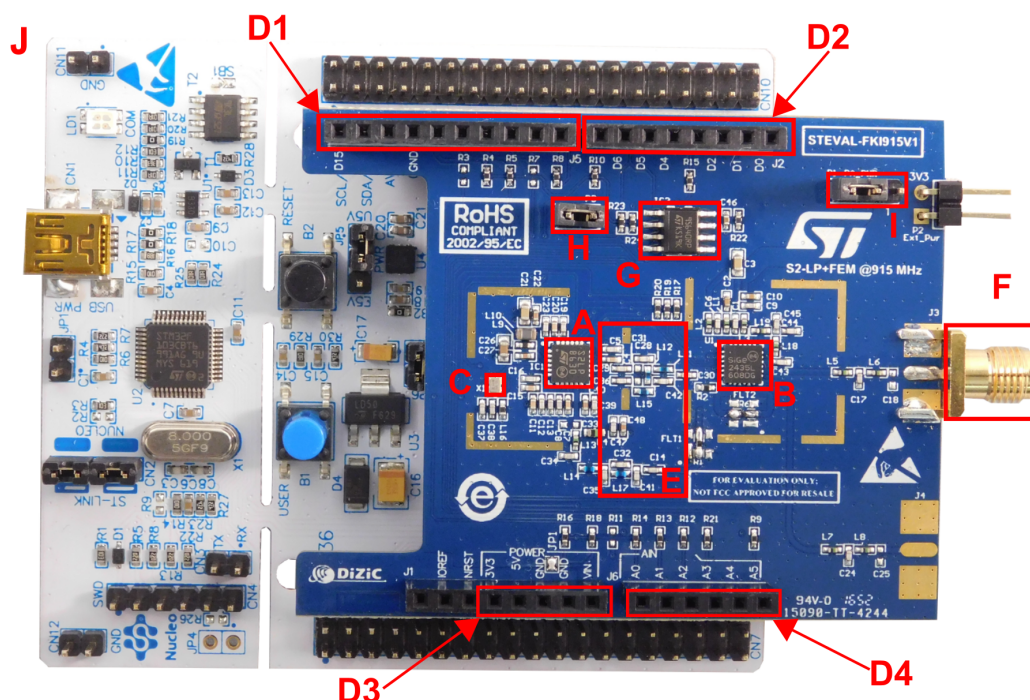
The [STEVAL-FKI915V1](#) evaluation board is tuned to work in the 902 MHz and 928 MHz bands.



The STEVAL-FKI915V1 evaluation board features (see [Figure 2. STEVAL-FKI915V1 evaluation board features](#)):

- S2-LP (A)
- Skyworks SE2435L FEM (B)
- High frequency 8 MHz crystal (C)
- Two rows with Arduino compliant connectors (D1-D4)
- Balun, matching network and harmonic filter (E)
- SMA connector (F)
- An EEPROM to store the manufacturing data (G)
- A jumper for S2-LP current measurement (H)
- A jumper for Skyworks SE2435L FEM current measurement (I)
- A NUCLEO-L152RE or NUCLEO-L053R8 board (J)

**Figure 2. STEVAL-FKI915V1 evaluation board features**



### 2.2.1

#### S2-LP connections

S2-LP signal test points are split across two rows which are Arduino compliant connectors: CN1, CN3 and CN2, CN4.

The S2-LP shield is connected to the NUCLEO motherboard using the Arduino compliant connectors.

The connectors and pin names below are used in the [STEVAL-FKI915V1](#) schematic diagram.

**Table 2. S2-LP: FKI915Vx evaluation board pin description**

Pin name	Pin number	Arduino connectors			
		J5 (D1)	J2 (D2)	J1 (D3)	J6 (D4)
VDD SMPS	1			pin 4	
SMPS1	2				
SMPS2	3				
XOUT	4				

Pin name	Pin number	Arduino connectors			
		J5 (D1)	J2 (D2)	J1 (D3)	J6 (D4)
XIN	5				
SDN	6		pin 8 (SDN)		
VDDANASYNTH	7			pin 4	
VRSYNTH	8				
VREFVCO	9				
VDDVCOTX	10			pin 4	
TX	11				
VRRF	12				
RXN	13				
RXP	14				
VDDRXDIG	15			pin 4	
SDO	16	pin 5 (MISO)			
SDI	17	pin 4 (MOSI)			
SCLK	18		pin 4 (SCK)		
CSN	19				pin 2 (CS)
GPIO0	20				pin 1 (GPIO0)
GPIO1	21				pin 3 (GPIO1)
GPIO2	22				pin 4 (GPIO2)
GPIO3	23				pin 6 (GPIO3)
VSMP3	24				
GND	25	pin 7		pins 6 and 7	

### 2.2.2 STEVAL-FKI915V1 power

The radio frontend is supplied via two different jumpers:

- P3 that supplies the S2-LP (H in [Figure 2. STEVAL-FKI915V1 evaluation board features](#))
- P1 that supplies the Skyworks FEM (I in [Figure 2. STEVAL-FKI915V1 evaluation board features](#))

The P1 jumper can be fitted in the following ways:

**Table 3. STEVAL-FKI915V1 expansion board jumper description**

P1 position	Comment
2-3	The FEM power supply is connected to the 3.3 V provided by the Nucleo motherboard
1-2	The FEM is supplied by an external voltage that can be provided by P2 connector pin 2.

To measure the radio part power consumption, add the two currents across P1 and P3.

## 2.3 NUCLEO board

### 2.3.1 Push buttons

The board has one button to reset the microcontroller and another one available for the application.

### 2.3.2 LEDs

The available LEDs are:

- LD1: green (Nucleo embedded ST-LINK power on)
- LD2: green (user LED)
- LD3: red (microcontroller power)

### 2.3.3 Embedded ST-LINK

The ST-LINK/V2-1 programming and debugging tool is integrated in the STM32 Nucleo boards. The ST-LINK/V2-1 makes the STM32 Nucleo boards "mbed" enabled.

### 2.3.4 STM32L microcontroller

The NUCLEO on-board microcontroller is programmed by the S2-LP DK firmware and is used to drive the device through the GUI or through the library examples.

## 2.4 Hardware setup

- Step 1.** Connect an antenna to the SMA connector
- Step 2.** Ensure the jumper configuration on the board is correct (see [Section 2.1.2 STEVAL-FKI868Vx](#), [STEVAL-FKI433Vx](#) and [STEVAL-FKI512V1 power](#) and [Section 2.2.2 STEVAL-FKI915V1 power](#))
- Step 3.** Connect the STM32 Nucleo board to the PC through a USB cable (via CN5 connector)
- Step 4.** Check the power LED DL4 light is on.

## 3 GUI software description

The S2-LP DK GUI included in the software package is a graphical user interface that can be used to interact with and evaluate the capabilities of the S2-LP device.

You can run this utility by clicking on the S2-LP GUI icon on the desktop or under: Start → STMicroelectronics → S2-LP DK X.X.X → S2-LP DK

This version of the GUI for S2-LP exclusively targets RF evaluation performance and only provides the RF test window and the manipulation of the device configuration parameters.

### 3.1 Installation

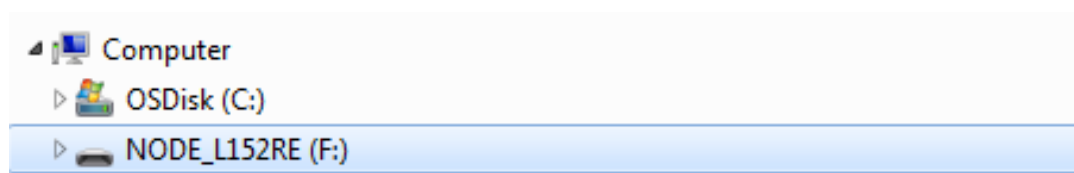
To use the S2-LP GUI, make sure you have correctly set up your hardware and software (S2-LP DK installed). The firmware image to be programmed is the latest S2LP\_CLI\_NUCLEO\_L1/L0.hex available in the S2-LP DK SW package in the Binary folder.

In order to download the S2LP\_CLI\_NUCLEO\_L1/L0.hex binary image into the STM32 internal Flash of the STM32 Nucleo board, follow these steps:

- Connect the NUCLEO board to a PC USB port
- Open the S2-LP DK GUI
- Select the COM port associated to the board
- Open Tools → Firmware Upgrade and Browse to select the file S2LP\_CLI\_NUCLEO\_L1/L0.hex, then press Open and wait for the firmware download.

Alternatively, it is possible to flash the STM32 Nucleo board by dragging and dropping the S2LP\_CLI\_NUCLEO\_L1/L0.bin binary file into the disk drive recognized by Windows (for example, the drive F below).

Figure 3. NODE\_L152RE disk drive



### 3.2 Detailed description

The S2-LP DK GUI can use only one S2-LP DK - MB plus S2-LP RF - DB connected through a USB cable to a PC. So, it is necessary to run one instance of S2-LP DK - GUI for each board connected to the PC. [Figure 4. Connection setup 1: 1 PC with S2-LP-DK GUI](#) and [Figure 5. Connection setup 2: 2 PCs with S2-LP-DK GUI](#) show typical connections with one or two PCs.

Figure 4. Connection setup 1: 1 PC with S2-LP-DK GUI



Figure 5. Connection setup 2: 2 PCs with S2-LP-DK GUI

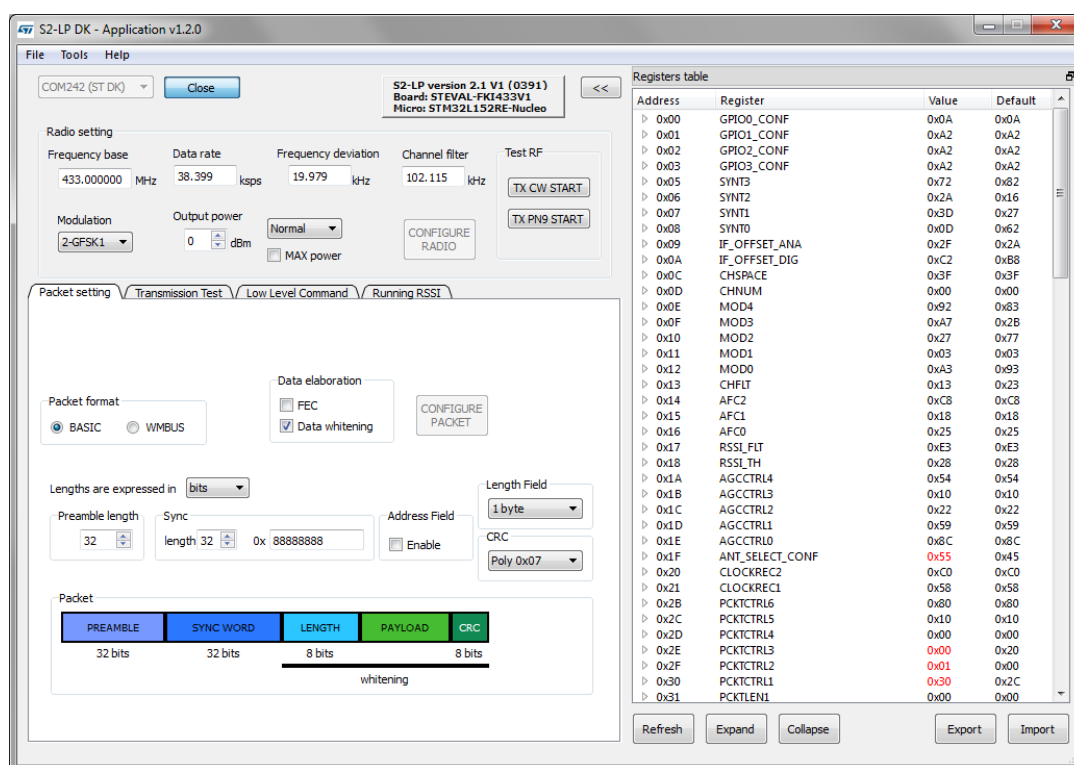


During the tests, each S2-LP DK - DB can work as a transmitter (TX) or a receiver (RX).

The TX device is used as a transmitter during the communication tests; the RX device is used as a receiver during the communication tests. The user can configure the S2-LP DK - DB as a TX device or RX device and dynamically change this selection before running a test.

When the user runs the S2-LP DK.exe file, the S2-LP DK - GUI windows appears as shown below:

Figure 6. S2-LP GUI main window



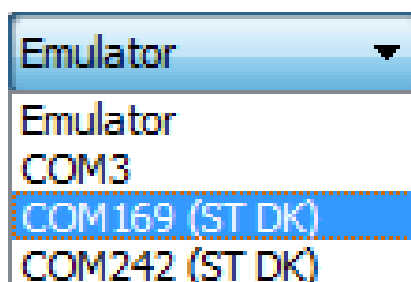
### 3.2.1 Connection panel

At the top of the main window, the user can select the appropriate available COM port from a drop down list. Once the correct S2-LP COM port is selected, and the open button clicked, the default configuration of S2-LP is loaded and displayed on the S2-LP DK - GUI.

Click the COM list also to refresh the available COM port list.

The COM ports associated with the ST development kits are labelled as "(ST DK)".

Figure 7. Available COM ports



### 3.2.2 Radio setting panel

The radio setting panel is always shown, informing the user about:

- frequency base;
- modulation;
- data rate;
- frequency deviation;
- channel filter
- output power.

These fields can be changed according to these limits (the values may change slightly according to the XTAL frequency):

- frequency base
  - Middle band: [413 - 479] MHz - [452 - 527] MHz
  - High band: [826 - 958] MHz - [904 - 1055] MHz
- modulation:
  - 2-FSK
  - 2-GFSK BT 0.5
  - 2-GFSK BT 1
  - 4-FSK
  - 4-GFSK BT 0.5
  - 4-GFSK BT 1
  - ASK
  - OOK
- data rate interval: [0.3 - 250] kbps.
- frequency deviation interval: [0.793 - 761] kHz.
- channel filter interval: [1.1 - 769.3] kHz.
- output power interval: [-30.0 14.0] dBm if the Normal (without external PA) configuration is selected (as for the STEVAL-FKI433V1 or STEVAL-FKI868V1).
- output power interval: [-8.0 27.0] dBm if the PA configuration is selected (as for the STEVAL-FKI915V1).

Clicking the "Configure radio" button, all the values are sent to the device and then read and shown.

Figure 8. S2-LP radio setting

### 3.2.3 RF test mode

TX CW and TX PN9 commands put the S2-LP in test mode.

Figure 9. RF test mode buttons



Both tests require only one device connected to PC.

#### 3.2.3.1 TX CW test

To start this test mode:

1. select the desired radio settings and load them by clicking the "Configure radio" button;
2. Click the "TX CW START" button.

Through this test mode, S2-LP transmits a continuous wave (CW) at the selected frequency and with the selected output power. The user can measure the output signal at the suitable SMA connector or the TX state current consumption.

The S2-LP stays in TX state until the "TX CW STOP" button is clicked.

To change frequency or output power, stop the running test first and then repeat steps 1 and 2 selecting the desired frequency or output power during step 1.

#### 3.2.3.2 TX PN9 test

To start this test mode:

1. select the desired radio settings and load them by clicking the "Configure radio" button;
2. click the "TX PN9 START" button.

Through this test mode, S2-LP transmits a PN9 data stream modulated according to the radio setting.

The user can measure the output signal at the suitable SMA connector or the TX state current consumption.

The S2-LP stays in TX state until the "TX PN9 STOP" button is clicked.

To change frequency, output power or modulation scheme, stop the running test first and then repeat steps 1 and 2 selecting the desired frequency, output power or modulation scheme during step 1.

## 3.3 Packet setting

Select "Panel setting" to view the packet configurations available.

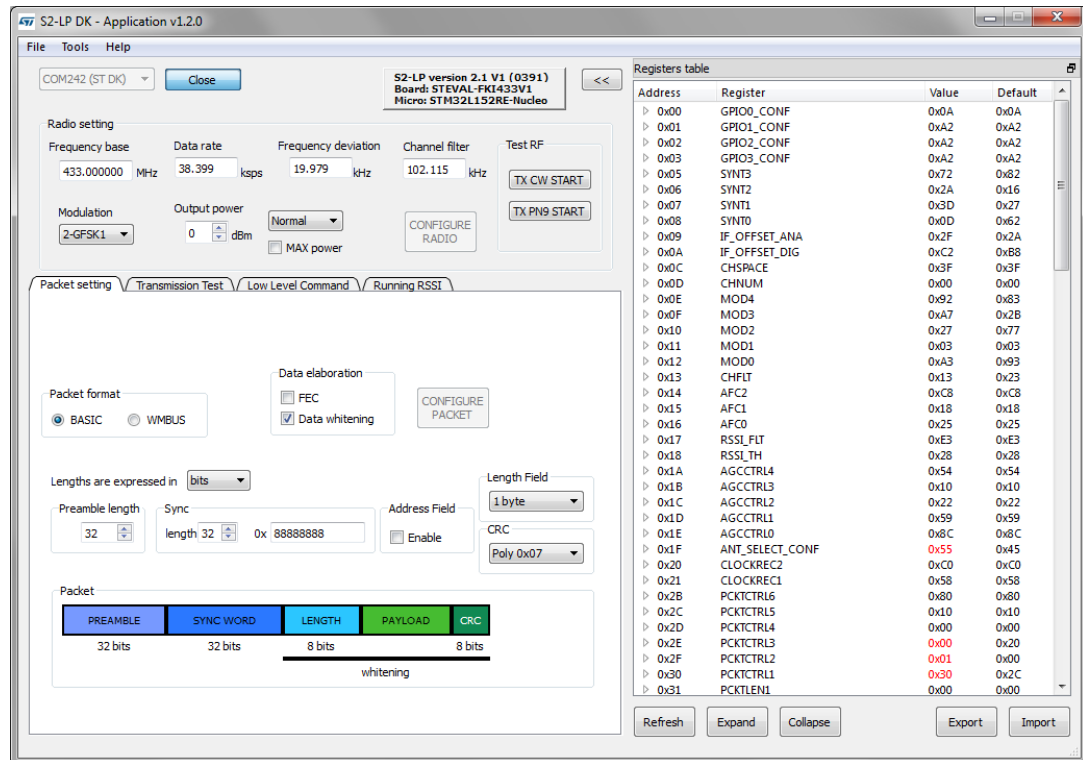
First select the desired packet format by pressing one of the radio buttons in the "Packet format" panel.

The user can choose:

- BASIC
- WMBUS

Each packet format gives different packet setting options.

Figure 10. Basic packet setting panel



### 3.3.1 Packet setting: BASIC

The options for this packet (default configuration) are:

- Preamble length
- Sync length
- Sync value
- CRC
- FEC
- Data whitening

These fields can be changed according to:

- Preamble length interval.
- Sync length interval.
- CRC can be one of the following:
  - NO CRC.
  - Poly 0x07 (1 byte).
  - Poly 0x8005 (2 bytes).
  - Poly 0x1021 (2 bytes).
  - Poly 0x864CFB (3 bytes).
  - Poly 0x04C011BB7 (4 bytes).

The "FEC" and the "Data whitening" can be checked according to the desired setting; if checked, these features are used during the transmission.



### 3.3.2 Packet setting: WMBUS

Figure 11. wM-Bus packet setting panel

As shown, selecting wM-Bus, S2-LP uses certain parameters for the desired wM-Bus submode. These fields can be changed according to:

- Preamble length interval: [0 - 1024] chip sequence (01).
- Postamble length interval: [0 - 64] chip sequence (01).
- wM-Bus submode:
  - S1, S2, long header.
  - S1m, S2, T2 other to meter.
  - T1, T2, meter to other.
  - R2 short header.

## 3.4 Transmission test

Selecting the "Transmission test" view, the user can access all the available packet tests to run the transmission.

Figure 12. Transmission test panel

### Device role panel

In the "Device role" panel in the left corner, you can set the main role of the device during transmission (RX and TX).

### Data to send panel

The Data to send panel has the hexadecimal value or characters sent by a transmitter. The maximum length of this field is 255 bytes (GUI limitation) and represents the effective payload sent. If the HEX check box is selected, the value must be added as 07 08 09 0A and so on; if the ASCII check box is selected, characters are accepted. It is also possible to generate a random set of bytes by clicking "Random". In this case, the random sequence has a length equal to the one set by the payload length field. Since it is not certain that the randomly generated characters can be converted to ASCII, they are always represented in HEX format.

### RX timeout box

In the RX timeout box, the RX timeout in milliseconds should be set to a value large enough to receive the full SYNC word (afterwards, the timer is stopped). It can be set to an approximate value that is larger than the time duration of the preamble and the sync lengths. If the value is 0, the RX timeout is infinite and the S2-LP remains in the RX state until it finds a correct SYNC word.

### HEX or ASCII radio buttons

The data received can be displayed in HEX or ASCII format. If ASCII is set and a non-ASCII character is received, the representation automatically switches to HEX format.

### Packet length modes

The S2-LP has the following packet length modes:

- variable - the receiver retrieves the packet length from the packet itself (information stored by the transmitter in a field). To set the VAR mode (default), the PCKTCTRL2 register bit 0 must be set to 1.
- fixed - needs the receiver to know the length of the expected packet. To set fix mode, the PCKTCTRL2 register bit must be reset and the GUI disables the length field on the RX tab. This operation can be easily done by the register table. If the fix mode is selected, the RX tab in the transmission test will unlock the payload length field, making it settable by the user.

### The Test indicator panel

The Test indicator panel shows all the results about the transmission/reception operations: the packet number correctly received, the packet lost, the RSSI value, the PER since the communication start.

- On the right side, there are these controls:
  - total packets: sets how many packets the transmitter will send or how many packets are expected by the receiver (an infinite number of packets can be set if the value is 0).
  - ref timer: sets the period of time to enter in RX or TX.
- The test defines a cycle in which:
  - if the S2-LP is configured as a transmitter, it sends a packet then the device enters the idle state until the period set in the reference time box expires; then the cycle is repeated. The duration of this operation depends on the data rate and the approximate value is reported in the packet duration box.
  - if the S2-LP is configured as a receiver, the test works in a similar way: the device goes into RX state a couple of milliseconds before the transmitter goes into TX state, then waits for the packet SYNC for the time written in the RX timeout box. If the packet is received or the RX timeout expires, the S2-LP enters the idle state (ready) until the period set in the packet rate box expires. During the first communication, the S2-LP enters the RX state waiting for the first packet (synchronization packet) with infinite RX timeout.
- The RSSI threshold [dBm] sets the RSSI threshold. For good communication, set an RSSI threshold in the receiver greater than the RX noise floor.
- The Sense RSSI button can be used to read the RF power in the air according to the center frequency and the channel filter bandwidth configured. If the Sense RSSI button is clicked when there is no signal in the air, this feature reveals the RF noise in the environment.

### Reference time

It is crucial to set the reference time value greater than the packet duration. Otherwise, the received packet can be truncated or not received at all. Furthermore, the packet rate must be the same for both devices.

### Start/stop button

The Start button runs the test and turns into a Stop button while the test is stopped.

### Low power modes

#### Normal

In this mode, the device is set to RX or TX by the microcontroller and the idle state (when it is not in RX or TX) is READY. The microcontroller timer is used to implement the reference timer. When this timer expires, the MCU sets the S2-LP in active mode (RX or TX).

#### LDC

In this mode, the S2-LP is configured with the embedded Low Duty Cycle mode. The device idle state is SLEEP (in RX it is possible to choose the SLEEP A or B by setting the SLEEP\_MODE\_SEL bit in the PM\_CONF0 register).

The wake up event is generated by the embedded WAKE UP timer clocked by the internal low power RC oscillator. The wake up timer value is set equal to the GUI Ref Timer value.

In RX, the automatic reload on the SYNC function is used.

#### Sniff (RX only)

This configures the device enabling the fast RX termination and the LDC: when the start button is clicked, it starts in the microcontroller (a specific reception routine that manages this mode).

In sniff mode, it is important to set an adequate RSSI threshold to make the receiver work in low-power. The RSSI threshold must be higher than the noise floor seen by the device (otherwise the RSSI threshold is always asserted and thus the fast RX timeout is always stopped).

The wake up timer spin box can be used to configure the time interval between two consecutive RX windows. The fast RX timeout spin box can be used to configure the fast RX timer used to sense the RSSI from the channel. This timer is scaled by the channel filter exponent. Moreover, to set the receiver in the condition of receiving each packet, the wake up timer must be configured to wake up the device, at least twice, inside the preamble. For example, it could be set minor than the preamble duration, which can be computed as the TX (in bits) preamble length divided by the data rate.

For details on the new S2-LP Consumption GUI refer to [Section 4 S2-LP consumption tool](#).

### File name textbox

In the transmission test panel bottom, there is a textbox where you can write the file name in which the GUI saves a log of the current test.

*Note: This operation is performed during the test, so it is important to write the file name before the test starts.*

## 3.5 Low level commands

Selecting the "Low level commands", you can access the [S2-LP](#) test modes, read the status and set the SMPS output voltage.

Figure 13. Low level command panel

The S2-LP status can be read by clicking the S2-LP state read button. The chip version is also shown. The XTAL frequency is available in the XTAL frequency textbox.

### Test modes

Three buttons allow setting some particular test modes:

- RX data in GPIO: it configures the S2-LP GPIO\_0 and GPIO\_1 to send respectively the RX data received and the clock signal. In this way, when the S2-LP goes into RX state, it is possible to see the received packet.
- RX data in GPIO No Packet: it is equal to the RX data in GPIO with the only difference that the packet handler embedded in the S2-LP is by-passed.
- send data from GPIO: it configures the S2-LP GPIO\_0 and GPIO\_1 to send respectively the data to transmit and the clock to sample the data. In this way, when the S2-LP goes into the TX state, it is possible to send data loaded through the GPIO (and not through the FIFO).

### SMPS voltage

You can set the SMPS voltage (in mV) in the SMPS output box.

### SMPS frequency

The SMPS frequency can be set in MHz.

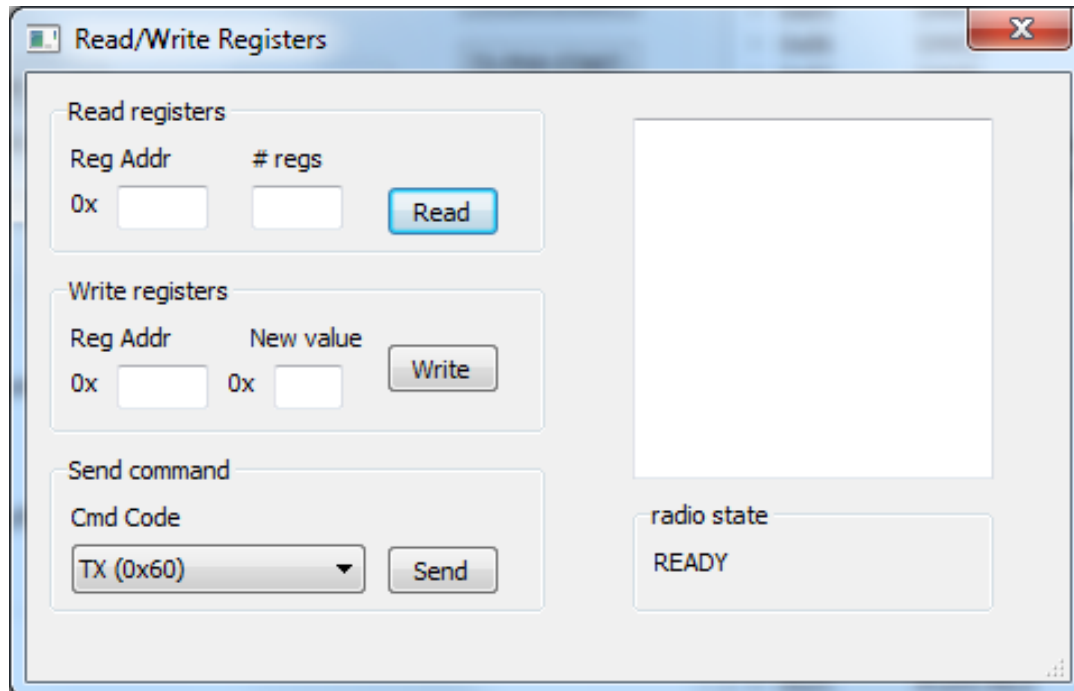
The following options are available:

- auto: the optimal frequency is set according to the state of the device (TX or RX);
- manual: the user can specify the SMPS frequency;
- disable KRM: the SMPS frequency is fixed to the value  $F_{dig}/4$ .

### Read/Write Registers

While you can write the most used registers through the register table, the Read/Write Registers button allows writing all the S2-LP registers. When clicked, the following window is shown:

Figure 14. Read/write register window



- Through the "Read registers" box, you can specify the starting address and the number of registers to be read from there on.
- Through the "Write registers" box, it is possible to specify the address and the value of a single register.

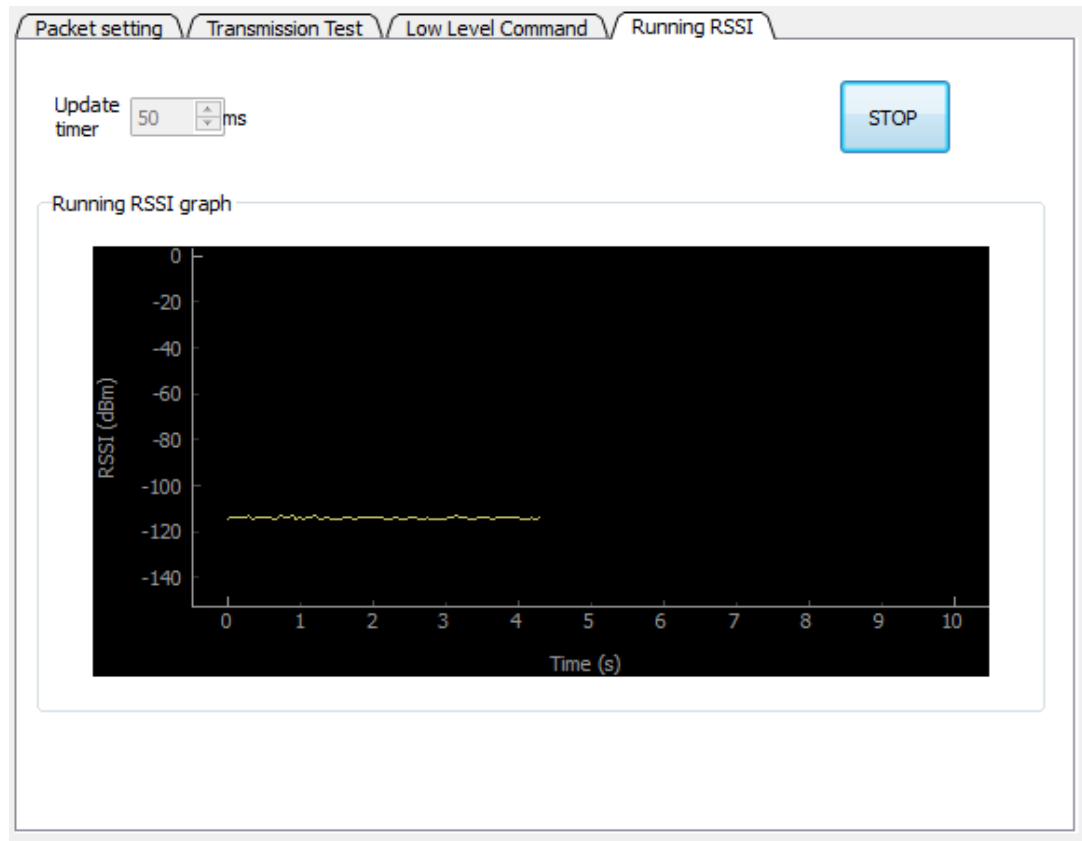
### 3.6 Running RSSI

The running RSSI tab allows the user to measure the power on the channel using the S2-LP running RSSI feature.

The RSSI values are sampled and plotted in a graph (RSSI (dBm) vs time(s)).

The polling interval is settable by the update timer spin-box.

Figure 15. Running RSSI tab



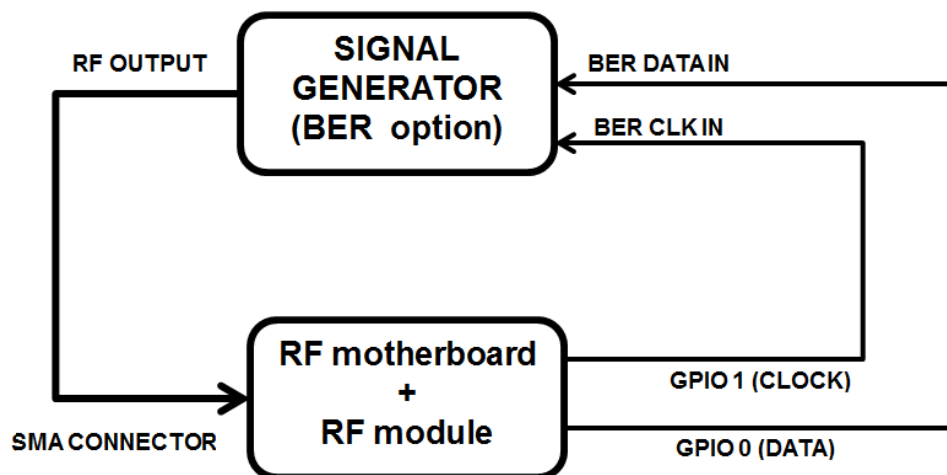
### 3.7 How to run a BER test using a signal generator

Through the low level command tab, you can put the S2-LP directly in RX mode through GPIOs; the packet handler is therefore totally bypassed and the demodulated data plus associated clock signal is available on two GPIOs.

This mode is enabled by the "RX Data on GPIO No Packet" button. The two signals then can be used in a signal generator with BER option to allow measuring the bit error rate according to that particular radio configuration (see Figure 16. BER test bench schema).

The data must be sampled on the clock signal falling edge.

Figure 16. BER test bench schema



### 3.8 Register table

On the right side of the GUI, a register table is shown by default (it can be hidden/shown using the "<<" button). The register table provides a quick and user-friendly way to modify the device registers and bit-fields.

Figure 17. Register table

Registers table			
Address	Register	Value	Default
0x07	SYNT1	0x27	0x27
0x08	SYNT0	0x62	0x62
0x09	IF_OFFSET_ANA	0x29	0x2A
0x0A	IF_OFFSET_DIG	0xB7	0xB8
0x0C	CHSPACE	0x3F	0x3F
0x0D	CHNUM	0x00	0x00
0x0E	MOD4	0x83	0x83
0x0F	MOD3	0x2B	0x2B
0x10	MOD2	0x27	0x77
0x11	MOD1	0x03	0x03
0x12	MOD0	0x93	0x93
0x13	CHFLT	0x23	0x23
0x14	AFC2	0xC8	0xC8
0x15	AFC1	0x18	0x18
0x16	AFC0	0x25	0x25
0x17	RSSI_FLT	0xE3	0xE3
0x18	RSSI_TH	0x28	0x28
0x1A	AGCCTRL4	0x75	0x75
0x1B	AGCCTRL3	0x00	0x00
0x1C	AGCCTRL2	0x22	0x22
0x1D	AGCCTRL1	0x7B	0x7B
0x1E	AGCCTRL0	0x8A	0x8A
0x1F	ANT_SELECT_CONF	0x55	0x45
0x20	CLOCKREC2	0x00	0xC0
0x21	CLOCKREC1	0x58	0x58
0x2B	PCKTCTRL6	0x80	0x80
0x2C	PCKTCTRL5	0x20	0x10
0x2D	PCKTCTRL4	0x00	0x00
0x2E	PCKTCTRL3	0x00	0x20
0x2F	PCKTCTRL2	0x01	0x00
7:6	RESERVED	0x00	0x00
5	FCS_TYPE_4G	0x00	0x00
4	FEC_TYPE_4G/STOP_BIT	0x00	0x00
3	INT_EN_4G/START_BIT	0x00	0x00
2	MBUS_3OF6_EN	0x00	0x00
1	MANCHESTER_EN	0x00	0x00
0	FIX_VAR_LEN	0x01	0x00
0x30	PCKTCTRL1	0x30	0x2C

Refresh Expand Collapse Export Import



The single register can be expanded or compressed to show its logical fields by clicking on the white arrow to the left of each entry.

When a field is modified, the corresponding register is automatically written in S2-LP.

Moreover, if the register modifies a parameter of the radio part or packet, the corresponding tab is updated with the new field value. It is also possible to get a detailed description of a register by double clicking on its entry in the register table.

Five buttons are available on the tab bottom side:

- Refresh: reads all the register value from the device and updates it into the tab.
- Expand: expands all the registers with the bit-fields.
- Collapse: collapses all the bit-fields.
- Export: saves the register configuration to a file selected by the user.
- Import: loads the registers from a file selected by the user. The file can be loaded both in XML and txt.

## 3.9 Menu bar

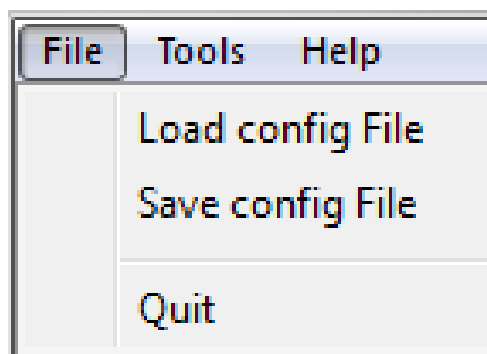
The GUI has a menu bar that exports different functions:

- File
- Tools which includes:
  - firmware upgrade
  - firmware version
  - export code configuration
  - export production info
- Help

### 3.9.1 File

The file menu provides the following list:

Figure 18. File list



#### 3.9.1.1 Save and load high level configurations

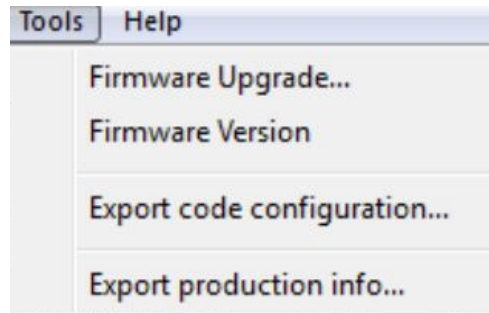
The save option allows saving the current radio and packet configurations in a file, so you can reload it easily. The load option allows loading the stored radio and packet configurations from a file.

#### 3.9.1.2 Save and load register configurations

These features are available through the export and import buttons below the register table.

### 3.9.2 Tools

The Tools menu provides the following list:

**Figure 19. Tool list**


### 3.9.2.1 **Firmware upgrade**

The S2-LP firmware allows performing automatic firmware upgrade via the USB port.

To upgrade the firmware:

1. Launch the S2-LP DK GUI.
2. Select the COM port of the NUCLEO board to upgrade.
3. From Tools→Firmware Upgrade select the firmware image to load (in .bin or .hex format).
4. By clicking OK, the firmware is programmed into the board.

The S2-LP DK binary directory contains the image to run the GUI S2LP\_CLI\_NUCLEO\_L1/L0.hex.

Alternatively, it is possible to directly copy the S2LP\_CLI\_NUCLEO\_L1/L0.bin in the hard drive corresponding to the STM32 Nucleo board to be flashed.

If the GUI finds a firmware that is not coherent, it raises a warning and prompts the user to upgrade the firmware.

### 3.9.2.2 **Firmware version**

The "firmware version" shows the current firmware version running on the microcontroller.

The firmware version format is x.y.z with option BETA to identify beta release and ALPHA to identify alpha release.

**Note:** *A beta release is prior to a final release with the same version number, that is: 2.0.0\_BETA is less recent than 2.0.0.*

### 3.9.2.3 **Export code configuration**

This option generates a C-language list of instructions to write new values into the S2-LP registers.

For example, the user can quickly find the device desired configuration using the GUI and then use this tool to obtain a C snippet that can be easily included in the program running on the microcontroller.

Below is an example of the C file obtained through the default configuration:

```
#include <stdint.h>
/**
 * The SPI interface is platform dependent, this means that it should be implemented
 * according to the used hardware.
 * The function S2LPSpiWriteRegisters(uint8_t address, uint8_t n_regs, uint8_t* buffer)
 * must be implemented.
 * An example of implementation (for the SDK_EVAL motherboards of the S2-LP kit) can
 * be found into the file: Projects/Drivers/BSP/SDK_Eval_STM32L/Src/SDK_EVAL_Spi_Driver.c
 * It is advisable to implement also the read registers, the command strobe and the 2
 * FIFO functions to read and write (no reference in this template code but needed in
 * applications using the FIFO of the S2-LP).
 */
/* list of the command codes of S2-LP */
#define COMMAND_TX ((uint8_t)(0x60)) /*!< Start to transmit; valid only from READY */
#define COMMAND_RX ((uint8_t)(0x61)) /*!< Start to receive; valid only from READY */
```

```
#define COMMAND_READY ((uint8_t)(0x62)) /*!< Go to READY; valid only from STANDBY
or SLEEP or LOCK */
#define COMMAND_STANDBY ((uint8_t)(0x63)) /*!< Go to STANDBY; valid only from READ
Y */
#define COMMAND_SLEEP ((uint8_t)(0x64)) /*!< Go to SLEEP; valid only from READY */
#define COMMAND_LOCKRX ((uint8_t)(0x65)) /*!< Go to LOCK state by using the RX con
figuration of the synth; valid only from READY */
#define COMMAND_LOCKTX ((uint8_t)(0x66)) /*!< Go to LOCK state by using the TX con
figuration of the synth; valid only from READY */
#define COMMAND_SABORT ((uint8_t)(0x67)) /*!< Force exit form TX or RX states and
go to READY state; valid only from TX or RX */
#define COMMAND_SRES ((uint8_t)(0x70)) /*!< Reset of all digital part, except SPI
registers */
#define COMMAND_FLUSHRXFIFO ((uint8_t)(0x71)) /*!< Clean the RX FIFO; valid from a
ll states */
#define COMMAND_FLUSHTXFIFO ((uint8_t)(0x72)) /*!< Clean the TX FIFO; valid from a
ll states */
/* This is the function that initializes the S2-LP with the configuration that the
user has exported using the GUI */
void SpiritBaseConfiguration(void)
{
    uint8_t tmp[5];
    tmp[0]= 0x92; /* reg. GPIO0_CONF (0x00) */
    tmp[1]= 0x52; /* reg. GPIO1_CONF (0x01) */
    tmp[2]= 0x2A; /* reg. GPIO2_CONF (0x02) */
    S2LPSpiWriteRegisters(0x00, 3, tmp);
    tmp[0]= 0x2B; /* reg. SYNT2 (0x06) */
    tmp[1]= 0x85; /* reg. SYNT1 (0x07) */
    tmp[2]= 0x1F; /* reg. SYNT0 (0x08) */
    tmp[3]= 0x2F; /* reg. IF_OFFSET_ANA (0x09) */
    tmp[4]= 0xC2; /* reg. IF_OFFSET_DIG (0x0A) */
    S2LPSpiWriteRegisters(0x06, 5, tmp);
    tmp[0]= 0x92; /* reg. MOD4 (0x0E) */
    tmp[1]= 0xA7; /* reg. MOD3 (0x0F) */
    tmp[2]= 0x27; /* reg. MOD2 (0x10) */
    S2LPSpiWriteRegisters(0x0E, 3, tmp);
    tmp[0]= 0xA3; /* reg. MOD0 (0x12) */
    tmp[1]= 0x13; /* reg. CHFLT (0x13) */
    S2LPSpiWriteRegisters(0x12, 2, tmp);
    tmp[0]= 0x55; /* reg. ANT_SELECT_CONF (0x1F) */
    tmp[1]= 0x00; /* reg. CLOCKREC2 (0x20) */
    S2LPSpiWriteRegisters(0x1F, 2, tmp);
    tmp[0]= 0x20; /* reg. PKTCTRL5 (0x2C) */
    S2LPSpiWriteRegisters(0x2C, 1, tmp);
    tmp[0]= 0x00; /* reg. PKTCTRL3 (0x2E) */
    tmp[1]= 0x01; /* reg. PKTCTRL2 (0x2F) */
    tmp[2]= 0x30; /* reg. PKTCTRL1 (0x30) */
    S2LPSpiWriteRegisters(0x2E, 3, tmp);
    tmp[0]= 0x01; /* reg. PROTOCOL1 (0x3A) */
    S2LPSpiWriteRegisters(0x3A, 1, tmp);
    tmp[0]= 0x41; /* reg. PKT_FLT_OPTIONS (0x40) */
    S2LPSpiWriteRegisters(0x40, 1, tmp);
    tmp[0]= 0x00; /* reg. FAST_RX_TIMER (0x54) */
    S2LPSpiWriteRegisters(0x54, 1, tmp);
    tmp[0]= 0x1D; /* reg. PA_POWER8 (0x5A) */
    S2LPSpiWriteRegisters(0x5A, 1, tmp);
    tmp[0]= 0x07; /* reg. PA_POWER0 (0x62) */
    tmp[1]= 0x01; /* reg. PA_CONFIG1 (0x63) */
    S2LPSpiWriteRegisters(0x62, 2, tmp);
}
```

According to the application, it can be manually modified to become a macro or a simple instruction block.

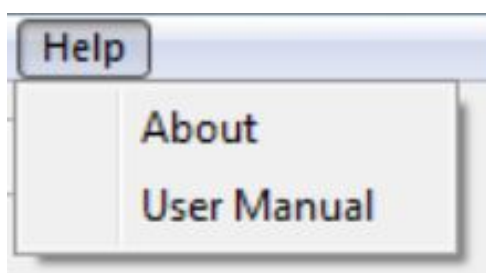
#### 3.9.2.4 **Export production info**

Each STEVAL-FKI board is equipped with a E2PROM with some data stored during the manufacturing phase that can be exported to a file by clicking on "Export production info".

#### 3.9.3 **Help**

The Help button gives you access to the user manual.

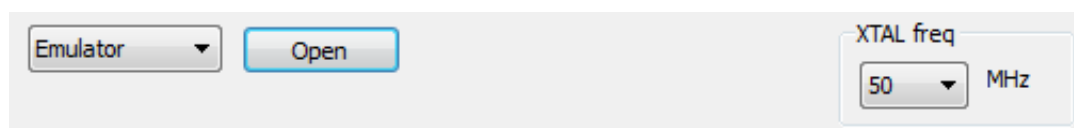
Figure 20. Help menu



#### 3.9.4 **Device emulator**

It is possible to simulate the device without any board connected to the PC through the "Emulator" button: when selected, the user can use the GUI as if a board were connected to the PC.

Figure 21. S2-LP emulator



Since there is no actual board, the user should manually specify an XTAL frequency (otherwise automatically computed by the microcontroller) using the tab which only becomes active in this case.

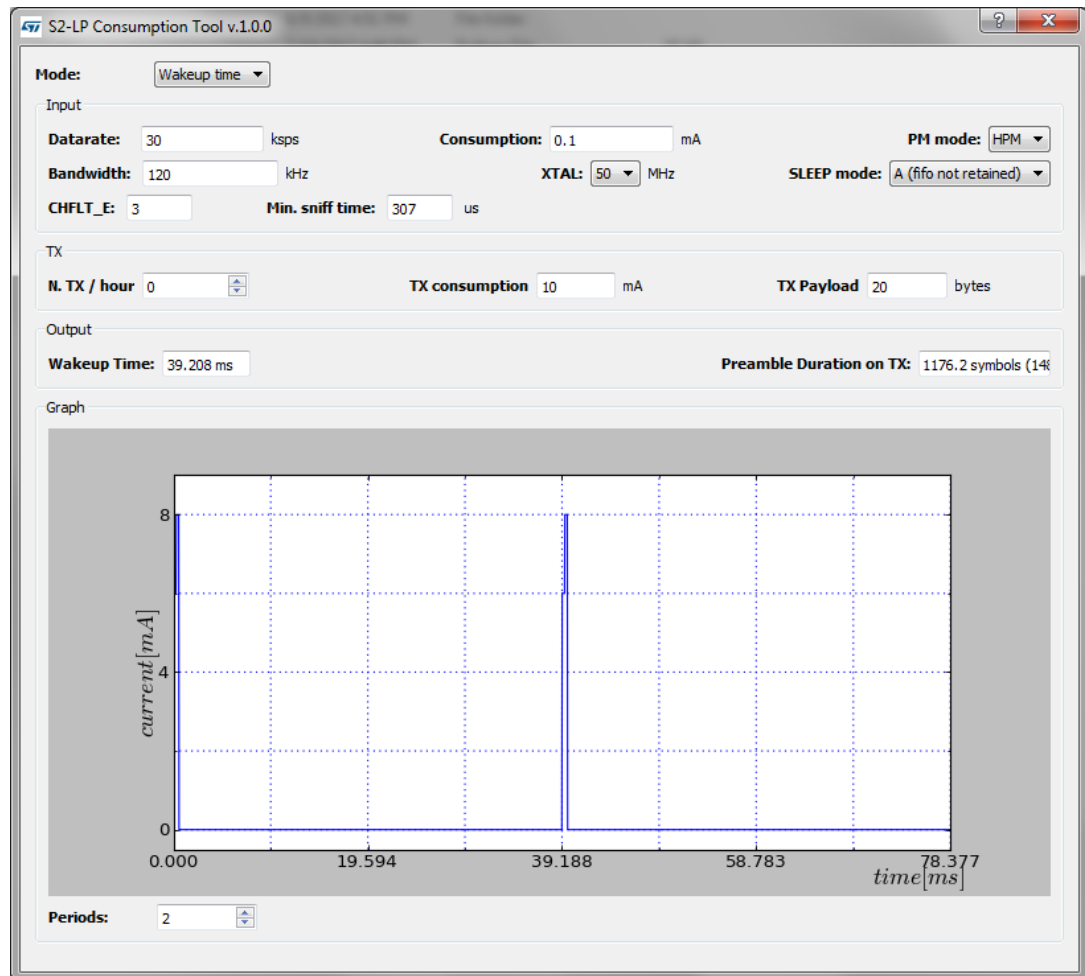
Subsequently, clicking the "Open" button, everything related to the device configuration should run exactly as if a device were connected.

The user can therefore easily select his own configuration and see or save the register values needed to keep the same configuration on his firmware. For this purpose, this feature can be used in cooperation with the "Export code configuration" tool.

## 4 S2-LP consumption tool

The **S2-LP Consumption Tool** is a graphical user interface to simulate the S2-LP current consumption in sniff mode. As it constitutes a simulation, it has not to be considered as a replacement of measurement on the actual silicon which gives the actual figures.

**Figure 22. S2-LP consumption tool - GUI overview**



The GUI can be set in two modes:

- Wakeup
- Consumption

### 4.1 Wakeup mode

In **Wakeup** mode, the GUI role is computing the wakeup time needed to reach a desired consumption.

The input parameters are:

- the desired communication **Datarate**
- the target **Consumption**
- the channel filter **Bandwidth**
- the **XTAL** frequency
- the **PM Mode** (HPM/LPM)

- the **SLEEP mode** (A or B)

Figure 23. Wakeup mode input

Input			
Datarate:	<input type="text"/>	kps	Consumption:
Bandwidth:	<input type="text"/>	kHz	XTAL:
CHFLT_E:	<input type="text"/>	Min. sniff time:	<input type="text"/>
		us	
			PM mode:
			SLEEP mode:

The output is the wakeup time and the preamble minimum length on the TX to ensure each packet is correctly received.

Figure 24. Wakeup mode output

Output	
Wakeup Time:	<input type="text"/>
Preamble Duration on TX:	<input type="text"/>

## 4.2 Consumption mode

In **Consumption** mode, the GUI computes the consumption given the wakeup time as input data.

The input parameters are:

- the desired communication **Datarate**
- the **Wakeup time**
- the channel filter **Bandwidth**
- the **XTAL** frequency
- the **PM Mode** (HPM/LPM)
- the **SLEEP mode** (A or B)

Figure 25. Consumption mode input

Input			
Datarate:	<input type="text"/>	kps	Wakeup Time:
Bandwidth:	<input type="text"/>	kHz	XTAL:
CHFLT_E:	<input type="text"/>	Min. sniff time:	<input type="text"/>
		us	
			PM mode:
			SLEEP mode:

The output is the current consumption and battery duration indication.

Figure 26. Consumption mode output

Output	
Wakeup Time:	<input type="text"/>
Preamble Duration on TX:	<input type="text"/>

For both modes the GUI performs the computation once all the input fields are filled in.

## 4.3 TX

The GUI also takes into account the consumption due to transmission of packets.

From the TX section, it is possible to specify the number of transmissions per hour, the TX current and the TX payload in bytes.

Figure 27. TX section

TX

N. TX / hour

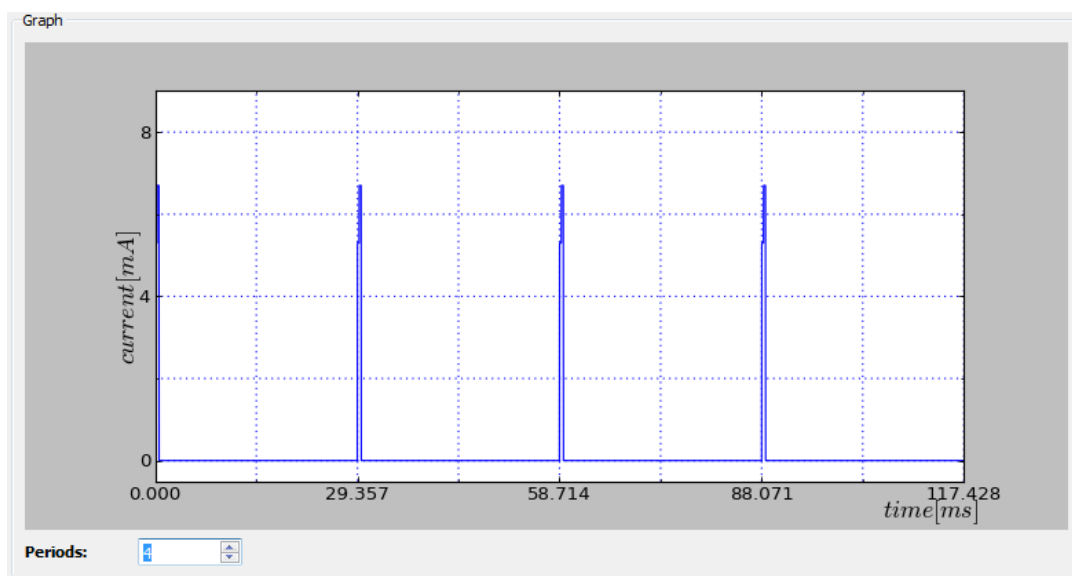
TX consumption  mA

TX Payload  bytes

#### 4.4 Current profile graph

The following picture shows the wakeup RX cycles.

Figure 28. Current graph



## 5 Firmware examples

The S2-LP development kit contains a set of some simple examples showing how to use some S2-LP features (such as the packet handler, low-power modes, auto-retransmissions, CSMA engine, etc).

Each example consists of at least two programs called A and B acting as transmitter and receiver, respectively:

- Basic generic: exchange of S2-LP BASIC packets with a payload length below the FIFO size (128 bytes).
- FIFO handler: exchange of S2-LP BASIC packets with a payload length bigger than the FIFO size.
- wM-Bus STD: exchange of wM-Bus like formatted packets.
- Sniff: exchange of packets using the fast RX termination feature for the receiver.
- LDCR: exchange of packets using the low duty cycle mode.
- Stack LLP: communication using the embedded link layer features of the S2-LP Sack packet (auto-ack and auto-retransmissions).
- CSMA: exchange of packets between two nodes when an interferer (implemented as a third node set in continuous transmission) disturbs the communication.
- Chat: exchange of strings between two nodes. For this example, the role is symmetric, so it is not necessary to have an A and a B program.

The examples are provided in source format and as an IAR (required toolchain is IAR Embedded Workbench for ARM (EWARM) toolchain (V7.40.3 or higher) and an MDK-ARM Keil project (required toolchain is V5.17 or higher).

### 5.1 IAR project

The workspace file is called S2LPLibrary\_Examples.eww and is placed in the directory Projects/Projects\_Cube/S2LPLibrary\_Examples/EWARM.

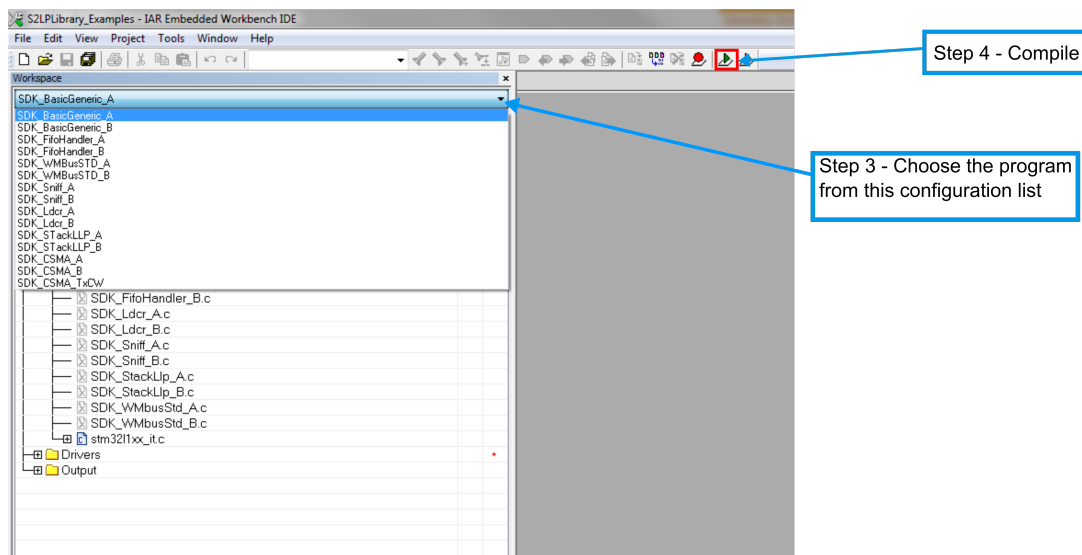
Each program is an IAR configuration and can be compiled and flashed on the STM32 Nucleo board using the embedded ST-LINK.

To use the project with IAR Embedded Workbench for ARM:

- Step 1.** Open the Embedded Workbench for ARM and select File→Open→Workspace menu.
- Step 2.** Open the IAR project .../ Projects/Projects\_Cube/S2LPLibrary\_Examples/EWARM/STM32Lxxxxx-Nucleo/S2LPLibrary\_Examples.eww
- Step 3.** Select the desired configuration to build
- Step 4.** Select the download and debug button to recompile and link the entire application, download the related binary image and go in debug mode.



Figure 29. IAR project

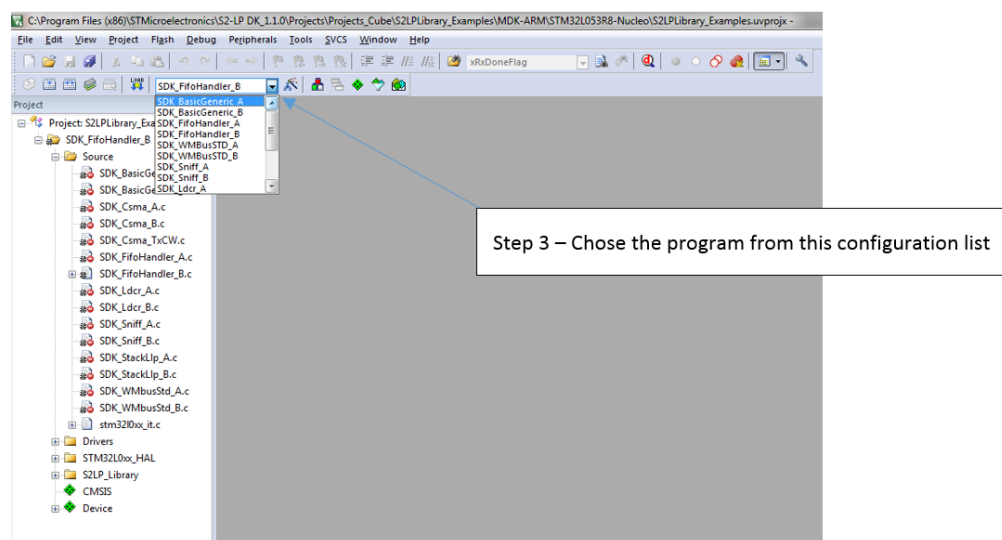


## 5.2 MDK-ARM Keil project

To use the project with KEIL uVision 5 for ARM:

- Step 1.** Open the KEIL uVision 5 for ARM and select Project->Open Project menu
- Step 2.** Open the KEIL project ../Projects/Projects\_Cube/S2LPLibrary\_Examples/EWARM/ STM32xxxxx-Nucleo/S2LPLibrary\_Examples.uvprojx
- Step 3.** Select the desired configuration to build

Figure 30. Keil project



- Step 4.** Select **Project->Rebuild all target files** to recompile and link the entire application.
- Step 5.** Select **Project->Download** to download the related binary image.

## **6 Release notes**

---

The S2-LP DK SW package release notes are contained in Documents/S2LP\_DK\_release\_notes/Release\_Notes.html (html format).

Open the file Documents/index.html for a global documentation index.

---

## 7 License

---

The S2-LP DK software package license file is accessible through the Documents/index.html file.

## Revision history

**Table 4. Document revision history**

Date	Version	Changes
19-Dec-2016	1	Initial release.
01-Feb-2017	2	Updated Section 1.2: "STSW-S2LP-DK development kit setup", Section 2.1: "STEVAL-FKI868V1 and STEVAL-FKI433V1 evaluation boards", Section 2.2: "STEVAL-FKI915V1 evaluation board", Figure 2: "STEVAL-FKI915V1 evaluation board features", Section 2.3.4: "STM32L microcontroller", Section 3.1: "Installation", Section 3.9.2.1: "Firmware upgrade", Section 4: "Firmware examples" Added Section 4.1: "IAR project" and Section 4.2: "MDK-ARM Keil project"
26-Jul-2017	3	Updated Figure 6: "S2-LP GUI main window", Section 3.2.1: "Connection panel", Figure 10: "Basic packet setting panel", Section, 3.3.2: "Packet setting: WMBUS", Section 5: "Firmware examples" and Section 3.4: "Transmission test". Added Section 4: "S2-LP Consumption Tool", Section 4.1: "Wakeup mode", Section 4.2: "Consumption mode", Section 4.3: "TX" and Section 4.4: "Current profile graph".
23-May-2018	4	Updated frequency bands and supported HW Added new device: STEVAL-FKI512V1

## Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
1.1	System requirements	2
1.2	STSW-S2LP-DK development kit setup	2
<b>2</b>	<b>Hardware description</b>	<b>3</b>
2.1	STEVAL-FKI868Vx, STEVAL-FKI433Vx and STEVAL-FKI512V1 evaluation boards	3
2.1.1	S2-LP connections	3
2.1.2	STEVAL-FKI868Vx, STEVAL-FKI433Vx and STEVAL-FKI512V1 power	4
2.2	STEVAL-FKI915V1 evaluation board	4
2.2.1	S2-LP connections	5
2.2.2	STEVAL-FKI915V1 power	6
2.3	NUCLEO board	6
2.3.1	Push buttons	6
2.3.2	LEDs	6
2.3.3	Embedded ST-LINK	7
2.3.4	STM32L microcontroller	7
2.4	Hardware setup	7
<b>3</b>	<b>GUI software description</b>	<b>8</b>
3.1	Installation	8
3.2	Detailed description	8
3.2.1	Connection panel	9
3.2.2	Radio setting panel	10
3.2.3	RF test mode	10
3.3	Packet setting	11
3.3.1	Packet setting: BASIC	12
3.3.2	Packet setting: WMBUS	12
3.4	Transmission test	13
3.5	Low level commands	16
3.6	Running RSSI	17
3.7	How to run a BER test using a signal generator	18
3.8	Register table	19

3.9	Menu bar .....	21
3.9.1	File .....	21
3.9.2	Tools .....	21
3.9.3	Help .....	24
3.9.4	Device emulator .....	24
4	<b>S2-LP consumption tool .....</b>	<b>25</b>
4.1	Wakeup mode .....	25
4.2	Consumption mode .....	26
4.3	TX .....	26
4.4	Current profile graph .....	27
5	<b>Firmware examples .....</b>	<b>28</b>
5.1	IAR project .....	28
5.2	MDK-ARM Keil project .....	29
6	<b>Release notes .....</b>	<b>30</b>
7	<b>License .....</b>	<b>31</b>
	<b>Revision history .....</b>	<b>32</b>

## List of tables

<b>Table 1.</b>	S2-LP: FKI868Vx and FKI433Vx evaluation board pin description . . . . .	4
<b>Table 2.</b>	S2-LP: FKI915Vx evaluation board pin description . . . . .	5
<b>Table 3.</b>	STEVAL-FKI915V1 expansion board jumper description . . . . .	6
<b>Table 4.</b>	Document revision history . . . . .	32

## List of figures

<b>Figure 1.</b>	STEVAL-FKI868Vx evaluation board features. . . . .	3
<b>Figure 2.</b>	STEVAL-FKI915V1 evaluation board features. . . . .	5
<b>Figure 3.</b>	NODE_L152RE disk drive . . . . .	8
<b>Figure 4.</b>	Connection setup 1: 1 PC with S2-LP-DK GUI . . . . .	8
<b>Figure 5.</b>	Connection setup 2: 2 PCs with S2-LP-DK GUI. . . . .	9
<b>Figure 6.</b>	S2-LP GUI main window . . . . .	9
<b>Figure 7.</b>	Available COM ports . . . . .	10
<b>Figure 8.</b>	S2-LP radio setting . . . . .	10
<b>Figure 9.</b>	RF test mode buttons . . . . .	11
<b>Figure 10.</b>	Basic packet setting panel . . . . .	12
<b>Figure 11.</b>	wM-Bus packet setting panel . . . . .	13
<b>Figure 12.</b>	Transmission test panel . . . . .	14
<b>Figure 13.</b>	Low level command panel . . . . .	16
<b>Figure 14.</b>	Read/write register window . . . . .	17
<b>Figure 15.</b>	Running RSSI tab . . . . .	18
<b>Figure 16.</b>	BER test bench schema . . . . .	19
<b>Figure 17.</b>	Register table. . . . .	20
<b>Figure 18.</b>	File list. . . . .	21
<b>Figure 19.</b>	Tool list . . . . .	22
<b>Figure 20.</b>	Help menu . . . . .	24
<b>Figure 21.</b>	S2-LP emulator . . . . .	24
<b>Figure 22.</b>	S2-LP consumption tool - GUI overview . . . . .	25
<b>Figure 23.</b>	Wakeup mode input . . . . .	26
<b>Figure 24.</b>	Wakeup mode output . . . . .	26
<b>Figure 25.</b>	Consumption mode input . . . . .	26
<b>Figure 26.</b>	Consumption mode output . . . . .	26
<b>Figure 27.</b>	TX section . . . . .	27
<b>Figure 28.</b>	Current graph. . . . .	27
<b>Figure 29.</b>	IAR project. . . . .	29
<b>Figure 30.</b>	Keil project. . . . .	29



**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. 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.

© 2018 STMicroelectronics – All rights reserved