LoRaSniffer

User Guide

Rev. 1.0

Contents

1 Introduction

1.1 Features

- Small and light
- LED indicators
- One Touch Power and Control button
- USB and battery powered (Rechargeable)
- Bluetooth® communication
- LoRa® and LoRaWAN® ready

1.2 Overview

The LoRaSniffer is a radio transceiver over LoRa® radio communication technology. It can be handheld or fixed to a structure to help you track a signal.

2 The product

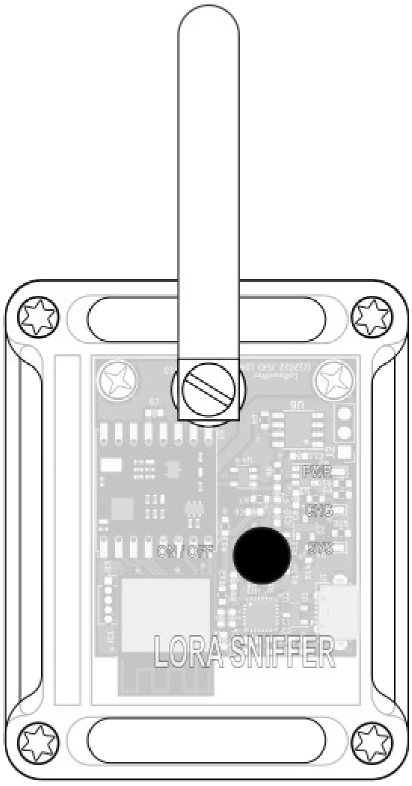


Figure 1 - LoRaSniffer

2.1 Specifications

2.1.1 Operating Conditions

Table 1- LoRaSniffer Operating Conditions.

Parameter	Rating	Units
Operating Ambient Temperature	-40 – +70º	ºC

2.1.2 Electrical Characteristics

Table 2- LoRaSniffer Electrical Characteristics.

Parameter		Rating	Units
Supply Voltage	USB	5.0	V
	Li-Ion, Li-Polymer Battery	3.2 – 4.2	V
Current	Radio Module	120	mA
	Device	500	mA

2.1.3 Mechanical Characteristics

Table 3- LoraSniffer Mechanical Characteristics.

Parameter	Value	Units
Width	54	mm
Height	18	mm
Height (w/ antenna)	34	mm
Length	67	mm
Length (w/ antenna)	96	mm
Ingress Protection	IP40	

2.1.4 Software Requirements

Table 4- LoraSniffer Software Requirements.

Software	Versions
Android ™	12, 13, 14
iOS	14, 15, 16, 17
Windows	7, 8, 10 and 11

2.2 Connectors

2.2.1 USB

There's one Micro USB™ type AB connector in the device that is used to charge the battery and to communicate with the LoRaSniffer.

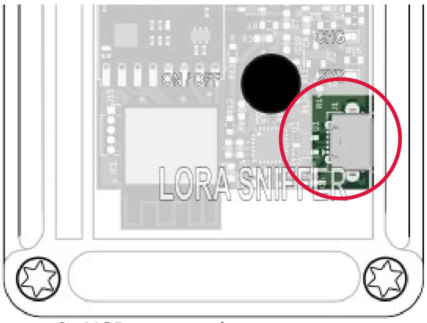


Figure 2- USB connection.

⚠ The Micro USB connector should be inserted with the spring locks facing the acrylic lid.

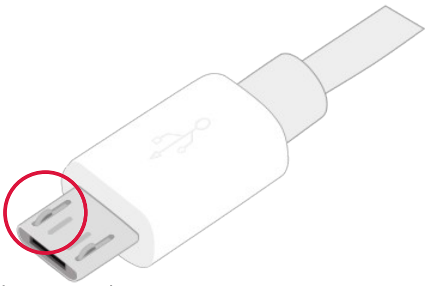


Figure 3- Micro USB connector.

2.2.2 Antenna

The antenna has to be attached to the female SMA Female jack in the LoRaSniffer.

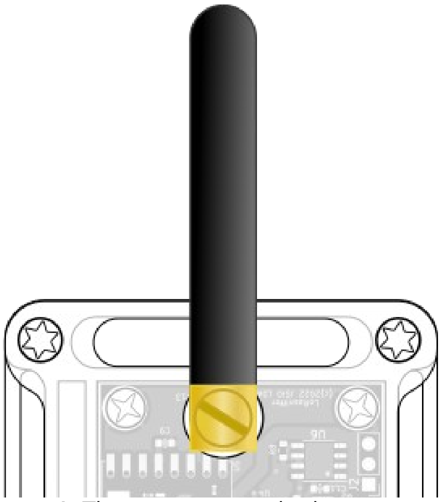


Figure 4- The antenna attached.

Do not attach the antenna if you are communicating at a distance of less than 3 meters between devices!

2.3 User interface

The LoRaSniffer has 3 LEDs indicators and a Button.

2.3.1 LED Indicators

2.3.1.1 PWR (green)

Power Indicator. When lit, the LoRaSniffer is ON.

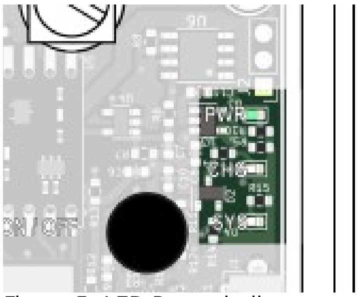


Figure 5- LED Power indicator.

2.3.1.2 CHG (red)

Battery Charging. When lit, the USB is connected and the LoRaSniffer battery is charging from the USB port.

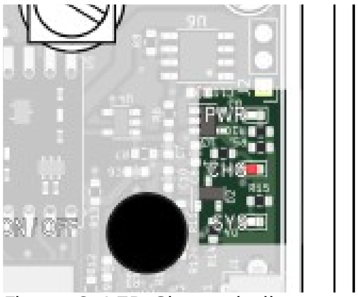


Figure 6- LED Charge indicator.

2.3.1.3 SYS (red)

System state. The LoRaSniffer use blink codes to advertise is state.

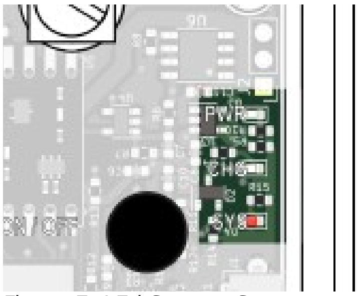


Figure 7- LEd System State indicator.

The following table enumerates the different SYS LED behaviors.

Table 5 - SYS LED behavior.

Process	Behavior	Times	Interval
LoRaSniffer is connected.	off	0	-
The LoRaSniffer is waiting to be connected via Mobile App.	blink	1	500 ms
The LoRaSniffer is in Repeater Mode.	blink	1	2 s
Battery is empty, the LoRaSniffer will soon switch OFF to safeguard the battery.	blink	1	200 ms
Switching ON / OFF	blink	3	1 s
Switching into Repeater Mode	blink	1	2.5 s

2.3.2 System Button (SW1)

The button is used to:

1. **Switch ON** : hold down the button until the SYS LED triple-flashes once, then release. The PWR LED (green) stays lit. If in this step the USB cable is connected, than the LoRaSniffer will enter USB mode and the BLE won't work.
2. **Switch OFF** : hold down the button until the SYS LED triple-flashes once, then release. If the button is hold down for more then 3 seconds, the device switches OFF as well.
3. **Repeater Mode** : Switch ON the LoRaSniffer, next hold down the button until the SYS LED triple-flashes twice, then release.

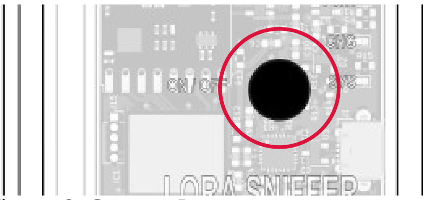


Figure 8- System Button.

3 Getting started

In order to start using the LoraSniffer you need to download the Mobile Application[1] (Mobile App) for the LoRaSniffer and install it in your Tablet or Smartphone.

The Mobile App is the complement for the user interface. It communicate with the LoRaSniffer via Bluetooth®.

Following the installation you run the Mobile App. You will be presented with the Graphic User Interface (GUI) in it's unconnected state.

Figure 9- Mobile App unconnected state.

Make sure that the Bluetooth® is active in the Tablet or Smartphone.

1 Download Mobile App from JSIO website, Google Play™ Store or Apple App Store®.

Switch ON the LoRaSniffer. As stated earlier in this guide, hold down the SW button until the SYS LED triple-flashes once, then release. Check if the Green LED stay lit.

If power on is okay the SYS LED (red) flash short every 500 ms. The Sniffer is waiting to be connected to the Tablet or Smartphone.

In the Mobile App GUI click the green **Connect Button**. The connection takes a few seconds to be established.

If the connection is successful the GUI changes to the following.

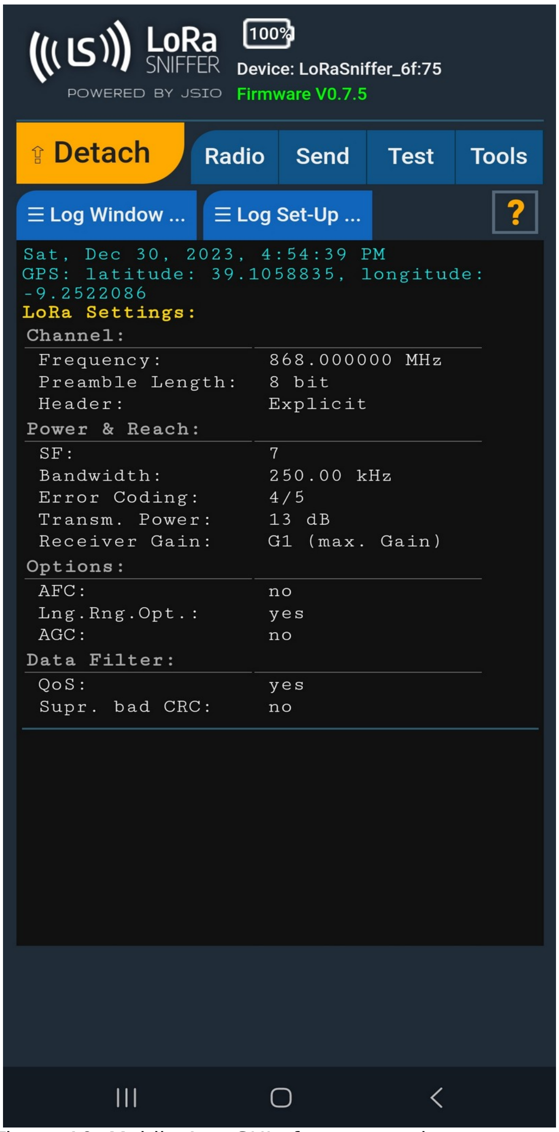


Figure 10- Mobile App GUI after connection success.

4 The Mobile App

The Mobile App is the main interface for the LoRaSniffer and will allow you to monitor and control it.

It is organized in four main elements. Bellow are a description of each.

4.1 The Header

The header (top of the GUI) basically shows information about the LoRaSniffer identification, the firmware version and the battery charge level.

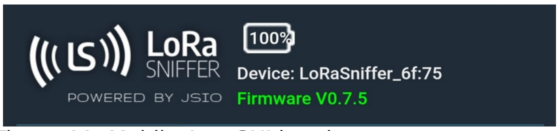


Figure 11- Mobile App GUI header.

4.2 The Connect Button

The **Connect Button** is just bellow the Header on the left and allow you to connect and disconnect the communication with the LoRaSniffer.

It has two states, **Connect** and **Detach** that are illustrated in the following figures.

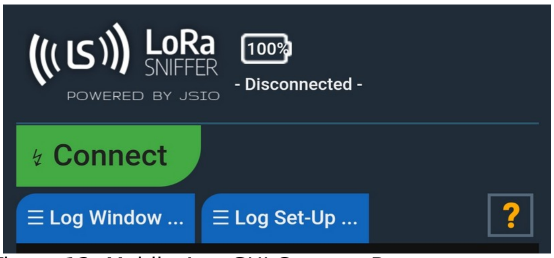


Figure 12- Mobile App GUI Connect Button.

When the communication is established with success the Connect Button changes it's label to **Detach**.

Clicking the Detach Button will disconnect the communication between the Mobile App and the LoRaSniffer.

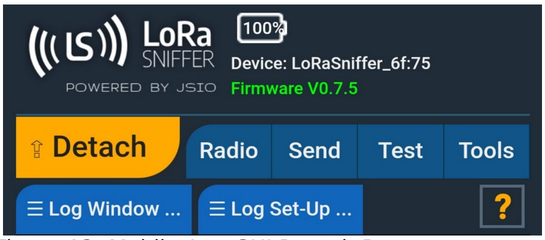


Figure 13- Mobile App GUI Detach Button.

Note that if the LoRaSniffer is powered off while it is connected to the Mobile App, the Mobile App will disconnect automatically.

4.3 The Tabs

The Tabs give you access to hidden panels. Just click on the tabs to hide or display the respective panel.

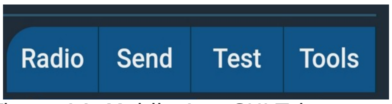


Figure 14- Mobile App GUI Tabs.

4.3.1 Radio Tab

This tab holds the LoRa Radio Settings panel and as the name implies here you can configure everything regarding the radio module.

If you are new to LoRa, you recommend to have some reading about it's concepts, since LoRa is not a trivial technology. A good starting point might be Wikipedia LoRa page.

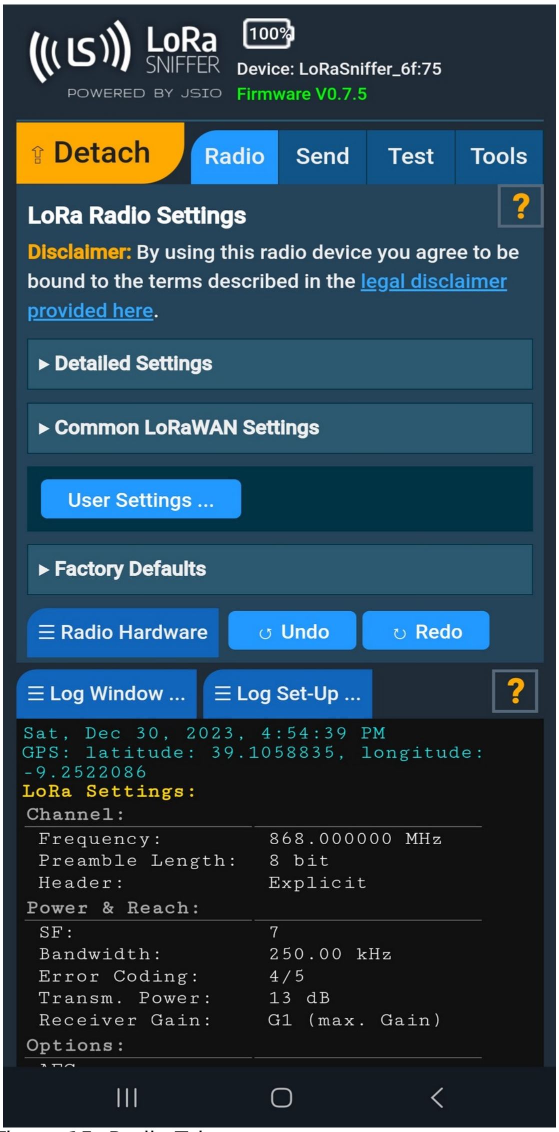


Figure 15- Radio Tab.

4.3.1.1 Detailed Settings

Channel Specifications. The radio channel-separation in LoRa is mainly defined by the frequency, the preamble length and the header mode.

- Radio Frequency

  If the radio module frequency band is EU863 - from 863 MHz to 870 MHz, you can adjust it from 862 MHz to 1020 MHz.

- Preamble Length

  The Preamble is used to synchronize the receiver with the transmitter. It length can be set from a minimum of 6 to a maximum of 65535.

- Header Mode

  You can choose from an Explicit or Implicit header mode.

Power and Reach. The LoRa parameters **Spreading factor** (SF) and **Bandwidth** (BW) determine how the data is "spread" (repeated). A higher SF value assures better reliability over distance on the cost of lower data-rates, while higher BW increases data-rates.

- Spreading Factor (SF)

  LoRa modulation has a total of 6 spreading factors from SF7 to SF12, however if you choose an **Implicit** Header Mode in the Channel Specifications you get SF6.

  Lowering the SF will make it more sensitive to noise which makes more difficult the communication.

- Bandwidth (BW)

  You can use channels with a bandwidth of either 125 kHz, 250 kHz or 500 kHz, depending on the region or the frequency plan.

- Error Coding Rate (on Send)

  You can choose from 4 code rates. The Error Code Rate expresses the proportion of bits in a data stream that actually carry useful information.

- Transmission Power

  It’s possible to choose from 19 predefined values. Remember that lower Transmission Power saves you battery, but the range of the signal will obviously be shorter.

- Receiver Gain

  This is the Receiver Gain control. You can choose between 6 gain settings.

Options.

- AFC (Automatic Frequency Correction)

  Used to deal with frequency differences between devices at low bandwidths. This has best results when transmitter and receiver are close together.

- No Long Range Optimizing
- AGC (Automatic Gain Control)

  Selecting this option allows the LoRaSniffer to handle a wide Rx input dynamic range.

Data Filter. Allow to suppress the **QoS** info and messages with invalid CRC from the log.

- No QoS (Quality of Signal)
- Suppress Messages with invalid CRC

4.3.1.2 Common LoRaWAN Settings

Here you find a comprehensive list of all Regional Frequency Plans.

By selecting one of the Frequency Plans and pressing button **"Selected Plan ..."** you can choose the desired Frequency and Data Rate available with this plan. More information regarding the plan itself is shown as well.

For a full understanding of the (honestly quite bureaucratic) LoRaWAN regulation and its regional requirements, please download the respective documentation.
An easier read to get a grasp on the basics is the LoRaWAN Regional Parameters page on the **The Things Network** site. For a detailed read about the **LoRaWAN protocol standard** itself, please visit the site of the LoRa Alliance.

4.3.1.3 User Settings

In this dialog you can memorize and organize your own LoRa radio settings for later usage. You can add the current settings to the list by giving it a name, or load settings from the list. Button **Show** will allow you to see the settings details.

You can delete settings from the list, if not useful anymore.

Keep in mind that the LoRaSniffer always power up with a default configuration.

4.3.1.4 Factory Defaults

Show you 2 standard factory settings, for best distance versus best data-rate.

4.3.1.5 Radio Hardware

Allow to reload the radio settings from the hardware, in case the Mobile App is not in sync. More, you can force a hard reset of the LoRa radio hardware, if necessary.

4.3.1.6 Undo & Redo

Allow to navigate the last 32 changes of the Radio Settings.

4.3.2 Send Tab

In this panel you can configure, write and send a data stream with the LoRa Radio Module.

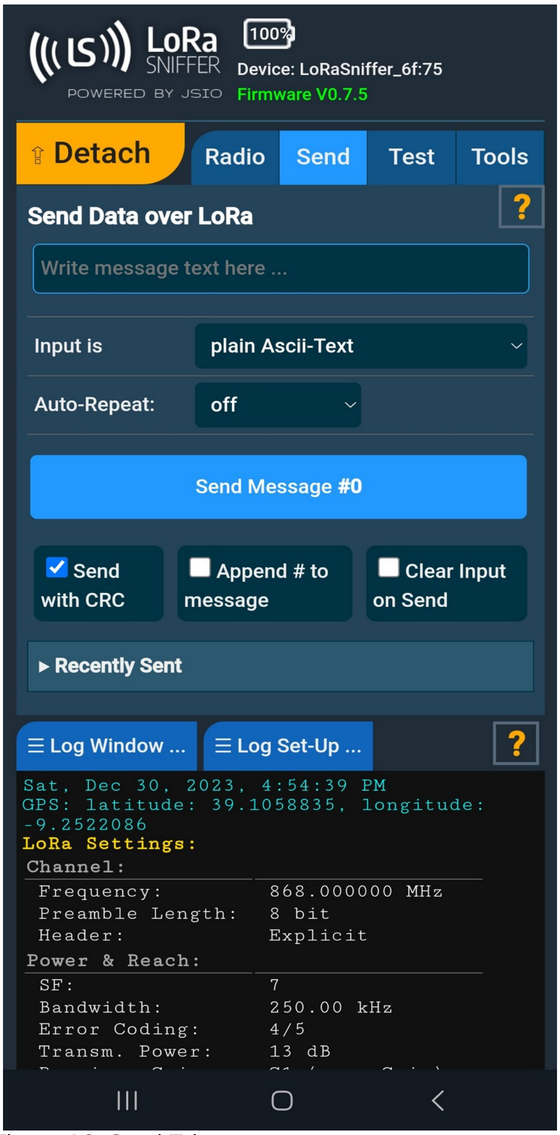


Figure 16- Send Tab.

4.3.2.1 Send Data Over Lora

Here you configure, write and send the data.

- Message text box

  Where you Input the data to be sent.

- Input is

  Here you can choose the type of input.

  - Simple ASCII text;

    - CSV list (E.g. 123,2,55,7,0xFF). The values must fit into 8-bit and are sent as binary data.

- Auto-Repeat (seconds)

  Send the data repeatedly according to the specified period.

- Send Message Button

  Click the button to send the message. The current number of the message is shown.

- Send with CRC

  Select if you want to add Cyclic Redundancy Check to the data payload.

- Append # to message

  Append to the message a running number.

- Clear Input on Send

  Clears the input message text box.

4.3.2.2 Recently Sent List

Lists in a combo the recent sent messages that you can select for later use. Will memorize the last 20 messages.

4.3.3 Test Tab

In order to perform the testing you will need 2 LoRaSniffers. One has to be set-up as **Repeater** (see below), while the other one is used normally.

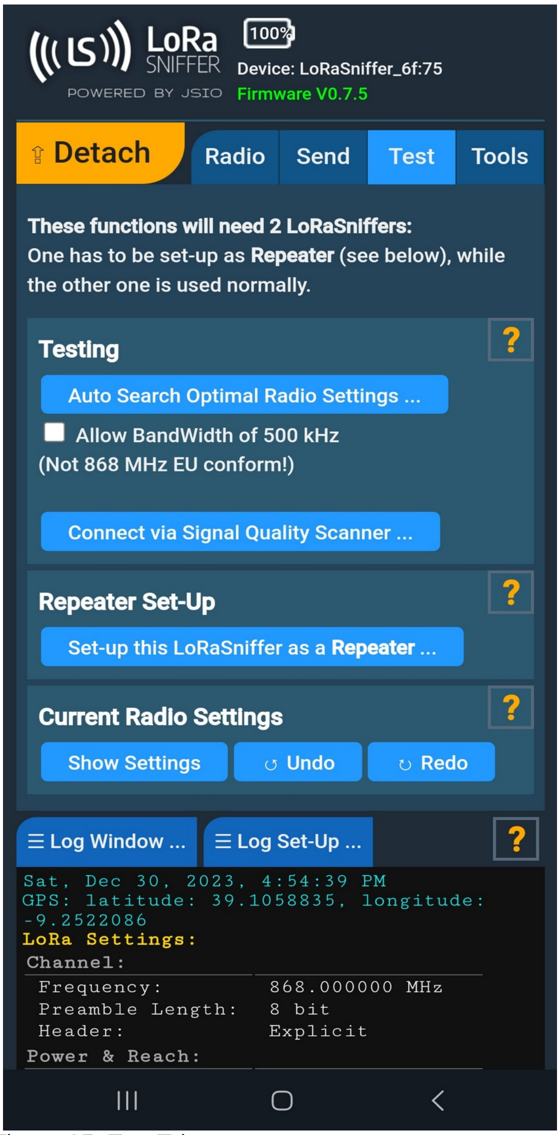


Figure 17- Test Tab.

4.3.3.1 Testing

- Auto Search Optimal Radio Settings

For **Optimal Radio Settings** is based on the use of 2 LoRaSniffers, where one is used as **Repeater**.
The purpose is to quickly provide settings in the field, especially when setting up a LoRa(WAN) network with aggregators and nodes.
You can use the **Repeater** in the location where you plan to have the Aggregator. The other sniffer you can move to the planned position of any

node. This will help to find suitable positions for nodes and aggregators, even in the most adverse environments.

Press the button "Automatic Search for Optimal Radio Settings" to start. The testing algorithm will take a fair amount of time to operate, during which the app will display the progress and the basic settings data.

Between the 2 sniffer, a protocol is established, that allows to interchange radio settings in a way to determine the best settings possible in a given situation.
The algorithm tries to mitigate the "distance versus data-rate" conflict by measuring the current RSSI and SNR settings and providing a solid estimation of reliable connection settings for the positions of the 2 sniffers.

- Connect via Signal Quality Scanner

4.3.3.2 Repeater Set-Up

A Repeater is a LoRaSniffer set into a special mode. In this mode the Sniffer will listen on the LDC (max. distance settings) for incoming radio-setting messages. When it receives one,

- it will switch to this setting,
- send an confirmation message back to the caller,
- and will stay on this setting until a switch-back to LDC is requested.

Set-up this LoRaSniffer as a Repeater

Once you press the button, the LoRaSniffer will disconnect from the app and set itself into **Repeater Mode**. It will stay then in this mode until it is switched OFF. It will not reconnect to the app when in repeater mode!
When in repeater mode, the LoRaSniffer will flash its red LED once every 2.5 seconds.

Manual Set-Up

The LoRaSniffer can as well be set into Repeater Mode without the use of the app. To do so, the LoRaSniffer needs to be switched-ON, but **not connected to the app!** Now, please hold down the button for 2 seconds. The LED will flicker once per second. After the 2nd flicker/second please release the button. The sniffer should now be in Repeater Mode and flash its red LED once every 2.5 seconds.

4.3.3.3 Current Radio Settings

Since on this page, the radio settings will change often, you can see the details of the current settings, and can Undo / Redo the settings as well.

4.3.4 Tools Tab

This panel provide you a set of tools to optimize communications as well firmware updates.

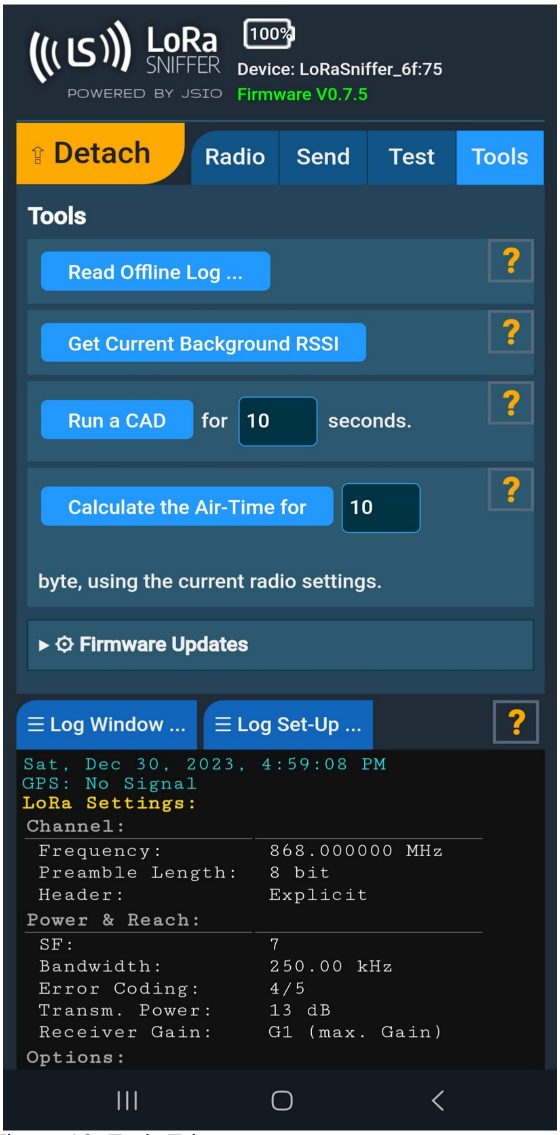


Figure 18- Tools Tab.

4.3.4.1 Read Offline Log

When the LoRaSniffer is ON, but not connected to the Mobile App or via USB, it will log the data received over LoRa (on its current radio settings).
This "Offline Log" is stored in the LoRaSniffer memory, which is lost when the LoRaSniffer is switched OFF! But, once you reconnect the LoRaSniffer without switching OFF, the log can be read out using this function. The data will be written to the Log Window exactly as received, with the current Log Window settings, but without time-stamp and GPS.

The Offline Log has a maximum capacity of 65 Kb. Once it is full, logging will stop, until the log has been read out.

Switching the LoRaSniffer OFF erases the log!

4.3.4.2 Get Current Background RSSI (Received Signal Strength Indicator)

This measures the current overall RSSI of the environment, within the current radio settings. It serves as an indicator for the overall radio noise in the area.

4.3.4.3 Run a CAD (Channel Activity Detection)

You can check if any preamble is received within the given time. This usually is used to implement "Listen before Transmit".

This function will capture all preambles in the range of the currently set-up frequency plus/minus half of the used bandwidth used for instance with a bandwidth of 500 kHz, it can/will capture as well the preambles of transmissions with 250 kHz and 125 kHz bandwidth.

4.3.4.4 Calculate the Air-Time

Here you can calculate how much airtime a given amount of bytes will need to be transferred using the current radio setting. The time returned in milliseconds is the absolute minimum time and does not account for delays due to message processing.

4.3.4.5 Firmware Updates

When ever you see the necessity to create a new firmware for the LoRaSniffer, here you can upload the new firmware.
For this purpose the LoRaSniffer needs access to a WiFi network, please provide the necessary information.

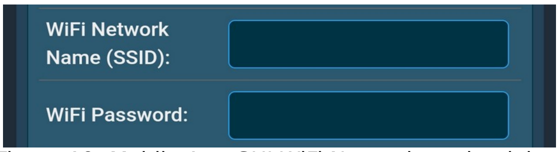


Figure 19- Mobile App GUI WiFi Network credentials.

Firmware updates are without risk - they are non-destructive and can be redone at any time! Once the update starts,there will be information and instructions shown in the Log Window.

Different Firmware Versions

In case you want to install a different version of firmware you can do it here. Follow the link and choose a PIN from the respective firmware version.

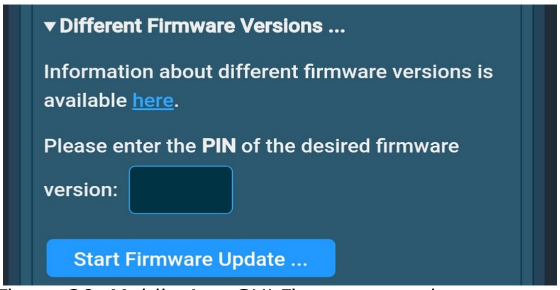


Figure 20- Mobile App GUI Firmware versions.

4.4 The Log Window

The Log Window is active by default and it will show at start the time and date information plus the GPS coordinates, in case your Tablet or Smartphone has that resource. Also the LoRa settings are displayed.

The Log Window shows mainly LoRa communications-events, but can as well log BLE communications.

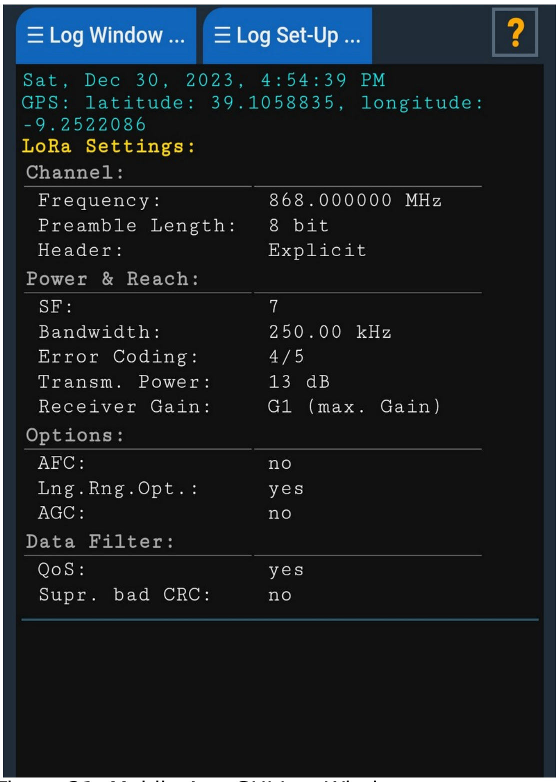


Figure 21- Mobile App GUI Log Window.

The Log window has two tabs on the top left corner that give you access to 2 menus.

4.4.1 Log Window Menu

Here you can clean the log, add a comment to the log, save the log (as HTML file) or copy it to the clipboard.
More, you can switch logging OFF, switch OFF word-wrapping and change from a scrolling log-window, to scroll the entire screen (log-window grows with log).

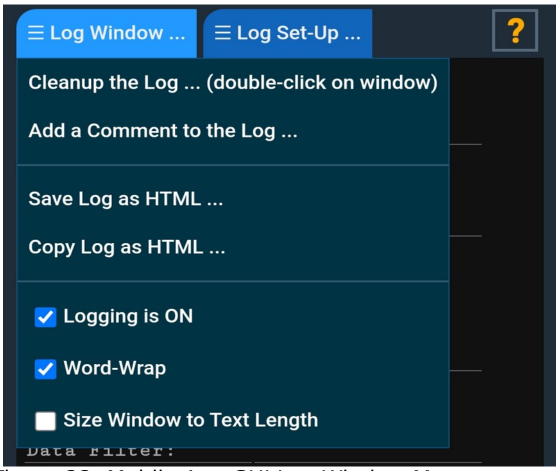


Figure 22- Mobile App GUI Log Window Menu.

4.4.2 Log Set-Up Menu

Here you set what additional data you want to see in the log: **Date & Time, GPS location** and/or a **Separator-line** between log entries.
You can log as well the **BLE communication** between the LoRaSniffer and the app (incl. success "errors" with code = 0).
Log Input as plain JSON allows you to log everything in JSON format, to be able to elaborate the log later.
You can as well **Change the Font-Size** of the log and you can **Save your Setup** as default set-up for the next time.

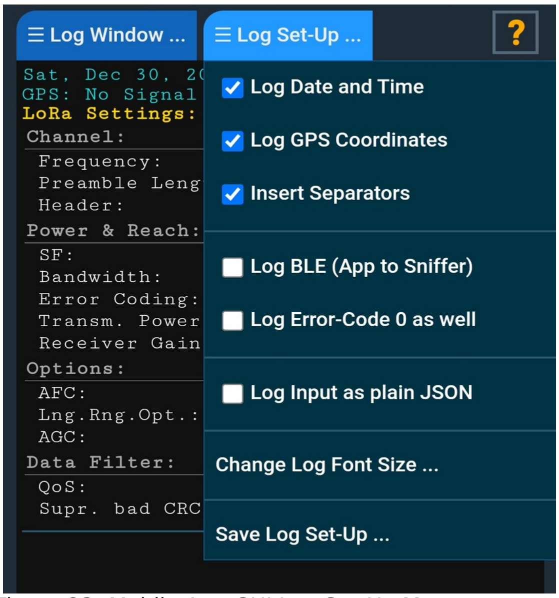


Figure 23- Mobile App GUI Log Set-Up Menu.

5 USB communication

As stated before the USB connector can be used for communication between the LoRaSniffer and a host PC.

In order to activate and use this resource you must proceed as described in the chapter about the System Button (SW1) at 1.

A Virtual Serial Port (CDC) is created and on Windows machines, the CDC will enumerate as COM Port and appear in the Ports section of the Windows Device Manager. The COM port number can also be found there.

On older Windows systems, the CP210x USB to UART driver is required for CDC.

5.1 LoRaSniffer Terminal software (Windows™)

The software can be downloaded from the company website. After the download run the install and the software will be ready for use.

During installation in some machines a Warning message may show up but you don't have to worry as long as you download the Installer directly from the company website.

Before running the software make sure to connect the LoRaSniffer to a host PC, with the USB cable and switch it ON in USB mode.

5.1.1 Terminal Window

Every communication are echoed in this window. There's an option to prefix the messages – In and Out, with the actual Date and Clock time of the host PC.

The button **Clear** erase all messages in the window without prompting any warning. If you want to save the data for later access is recommended to enable the **Log to File** option in Logging.

5.1.2 Serial Port Settings

The communication Baud Rate, Data Bits, Parity and Stop Bits are hardware defined and can't be changed. These are the settings to configure the host PC Serial Port.

If you don't see your communication port in the list please make sure to:

- Switch ON the LoRaSniffer with the USB cable already plugged in;
- Restart the software.

5.1.3 Logging

As stated before if you want to save the data you have to enable the **Log to File** option. You need to browse to the file where you want to save the data in.

Like in the Terminal Window there's an option to prefix the messages – In and Out, with the actual Date and Clock time of the host PC.

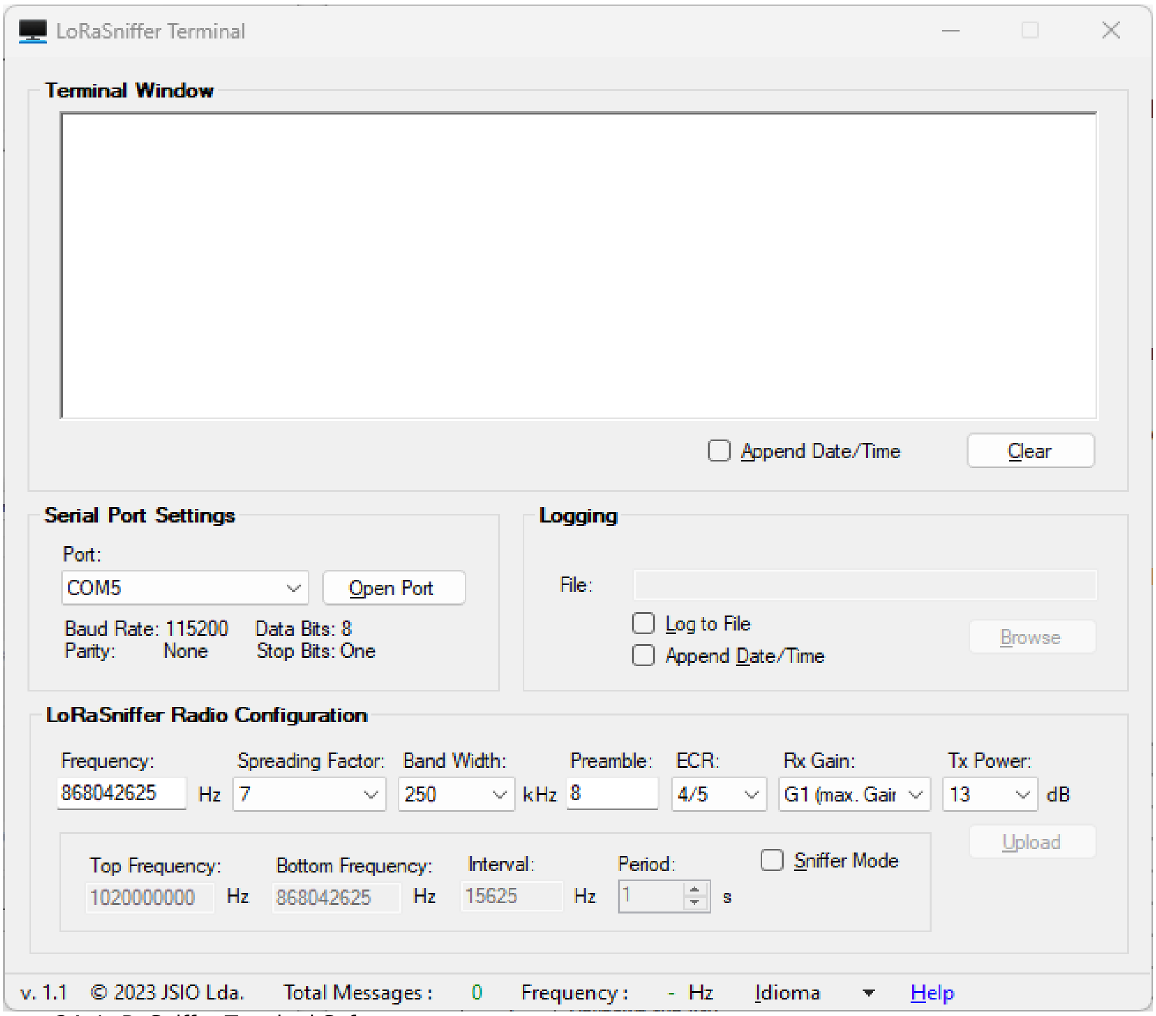


Figure 24- LoRaSniffer Terminal Software.

5.1.4 LoRaSniffer Radio Configuration

The LoRaSniffer connected to the USB can have the following fields configured.

- Frequency
- Spreading Factor
- Band Width
- Preamble Length
- ECR
- Rx Gain
- Tx Power

These settings are the same of the Mobile App described earlier in the sub-chapter **Radio Tab**.

5.1.5 Sniffer Mode

When this option is selected the LoRaSniffer will listen for communications starting at the Bottom Frequency till the Top Frequency. It will increment the Bottom Frequency in intervals specified in **Interval** and will listen at each frequency during the period specified in **Period.**

5.1.6 Status Bar

The Status Bar inform about the Software Version, the Total Input Messages and the current frequency set in the LoRaSniffer. The software language can also be set in the Status Bar.

6 Developer

6.1 JSON Protocol

For all data traffic between the LoRaSniffer and the BLE app, or the Python script used, the JSON protocol will be used.

For LoRa communications text or binary data can be used. Here we treat the communications between the LoRaSniffer and the BLE app or the Python script.

6.1.1 Error Messages

JSON error messages transport a single value: the error code. Error codes are defined in the file ***main.h***, enum **`error_code_t`**.

Here the list of available error codes and their meaning:

Table 6 - Error Codes

Error #	Error Code Description
0	Stands for success, all is OK!
1	Unknown JSON key
2	Incorrect JSON format
3	Value is out of range or wrong (if specific value is asked for)
4	Sending over LoRa failed
5	more items in the JSON than max_nr_results
6	Unreadable number (decimal or 0x-prefixed hex)
7	Unknown sub-key
8	Unknown key
9	JSON contains command-character unknown to JSON
10	JSON contains an array with a wrong number of cells
11	message is corrupt
12	Not enough memory!
13	LoRa: Missing or invalid CRC or message-length is incorrect!
14	Battery is too low, shutting OFF.

6.1.2 LoRa Radio Setup Message

6.1.2.1 Get the Setup

```
{"lora":0}
```

This message will cause the reception of a JSON message containing all radio setup information combined, as shown here. The received JSON message is used as well to setup the LoRa® radio of the sniffer. Here it is:

```
{"lora":
 {
 "flg":N,  // varies bit flags
 "frq":N,  // radio frequency
 "sf":N,       // Spreading Factor
 "bw":N,       // Bandwidth
```

```
 "pal":N,  // Preamble Length
 "ecr":N,  // Error Coding Rate
 "rxg":N,  // Receive Amplifier Sensibility
 "txp":N       // Transmission Amplification
 }
}
```

This JSON message is usually send as 1 single message, but might be send as well containing only part of the keys.

The sniffer will return an error code, as defined in file ***main.h***, enum `error_code_t`.

Here the single keys and their meanings:

6.1.2.2 Flags

`{"flg":bits}`

where `bits` is a number, with each bit acting as a switch (0=OFF, 1=ON):

bit	Meaning: 1= ON, 0 = OFF
0 LR	Implicit Header. Else, if = 0, it is an Explicit Header
1 LR	No long range optimisation
2 LR	AGC (Automatic Gain Control)
3 LR	Send all messages with a CRC (Cyclic Redundancy Check)
4 LR	*currently not used*
5 LR	*currently not used*
6 LR	*currently not used*
7 LR	*currently not used*
8	This sniffer is in Repeater Mode
9	Message bit for Repeater: Stay on this setting!
10	Suppress all received messages with an invalid CRC
11	Do not send Quality of Service (QoS) info
12	USB Spy: If USB is connected, repeat all data traffic from/to BLE on USB.
13	AFC On (Automatic Frequency Correction)

All bits signed with a ^LR^ are direct settings of the LoRa® Radio hardware. The other bits control run-time behavior of the app.

6.1.2.3 Frequency

`{"frq":number}`

where `number` has to be in the range between 862000000 and 1020000000 Hz (868 MHz to 1.02 GHz). ==All frequency values are internally quantized to become a multiple of 15625 Hz== and might therefore not return exactly as input by the user. In other words: The frequency can only be set in steps of 15625 Hz.

The single LoRa® frequency bands/channels can have a width of >= 0.3 MHz in the 868 MHz range, and a width of >= 0.6 MHz in the 915 MHZ range.

6.1.2.4 Spreading Factor (SF)

`{"sf":number}`

where `number` is the factor in the range between 6 and 12. **Attention**: a Spreading Factor of 6 is allowed only if the Header is **Implicit** (see **Flags**).

6.1.2.5 Bandwidth

`{"bw":number}`

where `number` can be 125, 250 or 500 (kHz). Transmissions on bandwidth 500 kHz are not allowed in the EU.

6.1.2.6 Preamble Length

`{"pal":number}`

where `number` is the radio's preamble length in the range of min. 6 and max 65535.

6.1.2.7 Error Coding Rate

`{"ecr":number}`

where `number` can be 0, 1, 2 or 3. These numbers correspond to the error coding rates 4/5, 4/6, 4/7 or 4/8.

6.1.2.8 RX Amplifier Sensibility

`{"lna":number}`

where `number` is a value between 0 and 6. These numbers correspond to increasing gain values **G6** to **G1** (= 0-5) and **G1+Booster** (=6)

6.1.2.9 Transmission Amplification

`{"txp":number}`

where `number` is the power value for the transmission amplifier. The value has to be in the range from 2 to 20 (dB).

6.1.3 LoRa® Radio Data Messages

LoRa Data can be moved as **binary** data between LoRa radios or as **JSON message** between the sniffer and the app/Python IO. Between the radios the data is send in binary form. To the App or Python-IO data is transmitted as JSON message in the following form:

```
{"ldt":
 {
  "tp":number, // the type of data
  "pl":...     // the payload as string
 }
}
```

The type of data **`tp`** is:

- 0: **`pl`** is an ascii string, non 0-terminated as e.g. `"pl":"ascii-string"`
- 1: **`pl`** is a string of 8-bit CSVs (Comma-Separated-Values) as e.g. `"pl":"12,56,123"`

- 2: **`pl`** is a string of a LoRa setup message in JSON (`"pl":{"lora":{...}}`)
- 3: **`pl`** is a error message (`"pl":{"error":N}}`)

In binary transmissions, "tp" is the 1st byte, followed by the data. The type of data `"tp"` can be:

"tp" or binary[0]	binary[1 to N] with N = message_length
0	(N-1) * ASCII characters, not 0-terminated
1	(N-1) * 8-bit values
2	7-byte structure of radio settings
3	1-byte error code

6.1.4 Quality of service (QoS)

The **`"ldt"`** LoRa Radio Data Message by default is completed with **Quality of Service** information. The LoRa QoS is send from the device only, it cannot be received by the device. If not disabled, this information is appended to the **`"ldt"`** message and send to the app or PythonIO. It contains the following additional information about the quality of the latest received LoRa message:

```
{"ldt":
 {
  "tp":number, // the type of data
  "pl":...         // the payload as string, number or array
 },
 "crc":N,       // LoRa CRC status
 "snr":N,       // Signal to Noise Ratio
 "rssi":N,  // Packet RSSI
 "fof":N        // Frequency offset
}
```

- **`crc`** status is:
  - 0= CRC is OK (message is correct)
  - 1= CRC is wrong (message has errors or is incomplete)
  - 2= no CRC in message (message might or might not be correct or complete)
  - 3= the LoRa header is invalid (message is definitely incorrect and/or incomplete)
- **`snr`** is given in dB. It is the distance between the overall noise and the useful signal (should be >= -13 dB).
- **`rssi`** is given in dBm. It is the signal strength of the last received message (should be >= -115 dBm)
- **`fof`** is given in Hz. It is the offset between the internal generated radio frequency and the frequency on which the latest message has been received (should be smaller than ±15625 / 2 Hz)

6.1.5 Miscellaneous

6.1.5.1 Read Blind Buffer

While the LoRaSniffer is ON, but not connected to the BLE app or via USB, everything it receives via LoRa is buffered internally into the "Blind Buffer". This buffer can be read out message by message using request command **`{"buff":0}`**.

On reception of this command, the LoRaSniffer will send the first/next LoRa message from the buffer to the BLE app or USB host. The message is in the same format as when normally connected.

Once the buffer is empty, it will reply with **`{"error":0}`** (All OK) instead.

The Blind Buffer can store up to 65 kB of LoRa messages. Once it is full, it will stop storing until it is emptied again.

6.1.5.2 Status

The status message can be used by the app or Python-IO to check the battery status and firmware version of the device:

Request is: `{"st":0}`

Answer is: `{"st":{"ver":N,"bat":N}}` where:

- `"ver"` is the firmware version as V.S.B (Version.Subversion.Batch is thousands.hundreds.rest. E.g.: 1013 is V1.0.13)
- `"bat"` is the battery voltage in millivolt. The sniffer will monitor the battery and switch OFF when the battery drops below 3100 mV. At 3300 mV it will start to blink the red LED as a warning.

RSSI background levels

When receiving `{"bkg":0}` the sniffer will check the currently measured background RSSI levels and send the data back. The signed value reported is in dBm and may range from -30 dBm (very strong) to -130 dBm (very weak).

6.1.5.3 CAD Channel Activity Detection

When receiving `{"cad":SEC}` the sniffer will perform a CAD with the current radio settings for SEC seconds. CAD just scans for any preamble it can detect within the current SF and BW settings.

If the BW is set to 500 kHz, it can detect any LoRa preamble using

- the same SF
- any BW between 125 kHz and 500 kHz
- with a frequency offset up to about ±240 kHz.

This allows to detect traffic e.g. for Listen-Before-Talk implementation.

If any activity is detected, instantly a message `{"cad":1}` is received from the sniffer. If after the given time nothing has been detected a `{"cad":0}` is received (timeout). If a `{"cad":-1}` is received, the sniffer is busy.

==**DO NOT** send a CAD while another CAD is still running! They will be queued and will execute in sequence - this can block the sniffer for a prolonged time!==

6.1.5.4 Airtime Calculation

When receiving `{"air":N}` the sniffer will calculate the airtime in milliseconds needed to send N byte of data with the current radio settings. The response will come back as {"air":[N,T]} where T is the air-time in milliseconds for N byte with the current radio setup.

6.1.5.5 Radio Reset

When receiving `{"rst":3203398350}` the LoRaSniffer will reset the LoRa radio module to bring it into a known state. The former LoRa settings are lost.

This is useful in situations where abnormal radio-behavior is encountered or the state of the radio is unknown. The number `3203398350` (or 0xDEADBEEF in hexadecimal) has the purpose to make sure that the command is intended and not send by error.

6.1.5.6 BLE Connection Check

When the BLE app connects to the sniffer, it will send `{"app":"LrSnfFW"}`. Only on receiving this message, the BLE connection is confirmed and valid. Else the app will close the connection. The device will respond with the LoRa Radio Setup Message, so that the app or Python script can setup its GUI and/or inner variables accordingly.

6.1.5.7 Firmware Update

Any firmware update is handled by the BLE mobile app. In case a new firmware is available, go to Tab **Tools** > **Firmware Update** and allow the LoRaSniffer access to your local WiFi network.

Once you start the update by pressing button **Start Firmware Update...**, the LoRaSniffer device will connect to your WiFi network and automatically download and install the new firmware. If anything goes wrong, the former firmware will stay active. Please follow the instructions shown.

6.2 Repeater

A Repeater is mainly used to be able to check LoRa connection and quality over long distances. The idea is to use 2 LoRaSniffers to be able to test different radio setups over a longer distance.

One LoRaSniffer, setup as a Repeater, is listening on LDC[2] for different radio setups (as described in LoRa Radio Data Messages). When receiving one, it will switch to it and use it to send a message back.

2 LDC stands for Long Distance Channel, usually a channel operating on SF12 with a bandwidth of 125 kHz, ECR 4/8, CRC and maximum RX/TX power to assure save transmissions. It is used as a fall-back whenever transmissions fail due to unfavorable circumstances.

The other LoRaSniffer will evaluate the quality of this message (QoS). The Repeater usually will then switch back again to LDC, to be ready for the next radio-setup to come. If instead bit 9 in key "`flg`" of the setup is set, it will stay on the received setup.

The BLE mobile app, used with the other LoRaSniffer, provides in tab **Test**, **Test this Radio-Setup ...** a simple way to test against a Repeater.

To use any LoRaSniffer as Repeater:

- send `{"repeater":0xDEADCAFE}` from the app or via USB
- or hold down the button for 2 seconds (2 flickers) and release then.

Once the Sniffer is setup to work as Repeater, it will disconnect from BLE or USB (if connected).

To be recognizable, Repeaters blink the red LED once every 2.5 seconds very shortly.

To exit the Repeater mode, just switch the Sniffer OFF.

6.2.1 PING-PONG

When a LoRaSniffer receives via LoRa a bare text 7 characters "`LS_PING`", it will always, automatically and instantly, reply by transmitting a bare message with the 7 characters "`LS_PONG`". This can be used with a Repeater for measuring purposes.

6.2.2 Automatic Setup Search

With a Repeater available, an automatic search for the optimal LoRa setup between a LoRaSniffer and the Repeater can be done as described here. On the mobile app this feature is found in Tab **Test**.

The idea is to find the optimal LoRa setup of SF, BW, ECR and TX-Power using the SNR- and RSSI-values received during the former transmissions as reference for the transmission quality.

If both would have the same effect, the algorithm generally prefers to modify SF, BW and/or ECR instead of simply increasing the transmission power. The goal is to achieve a good transmission reliability, while keeping the power consumption of the radio as low as possible.

During the run time, the log will be updated with all ongoing changes. The search can take some time and can therefore be interrupted in any moment by the user. At the end, the found LoRa setup will be active.

Errors will be reported as well.

6.3 Using Python & USB

The same JSON data traffic as between the LoRaSniffer and the BLE mobile app, can also be run between the device and Python software on a computer. The connection to the computer is done over the inbuilt USB port of the sniffer. The USB CDC class is used (USB to UART/serial bridge) and might need installation of a specific driver on Windows and Mac. Linux supports it natively.

The idea is to allow engineers to use Python to write own programs for an unlimited free use of the LoRaSniffer radio's capabilities.

6.3.1 USB versus BLE

The use of USB and BLE is mutually exclusive - USB **or** BLE can be used, they cannot be used in combination!

The BLE mobile app provides a graphical user-interface for direct usage. Using USB instead, you can create your own Python programs to take fully advantage of all features of the LoRaSniffer.

6.3.2 Using Python

6.3.2.1 USB module

The file **az_usb.py** contains the Python module used for communication with the LoRaSniffer.

Its function **`open_usb(port)`** opens the USB port. It will need the as parameter **`port`** a string with the USB port name used. In Linux and MAC OS this is usually something like `"/dev/ttyUSB0"`, while in Windows it will be a COM port with number, as e.g. **`"COM12"`** where the number stands for the COM port.

Please check which USB port your system is using!

The function **`send(string)`** will send a JSON string from your computer to the LoRaSniffer, which will respond/execute accordingly. Please see the description of JSON Protocol for the details.

The function **`waitIn()`** will block and wait for any input to arrive from the LoRaSniffer. Received data will be returned as a string.

The function checkIn() will check if any data has arrived and return it as a string. If no data has arrived, it will return an empty string. Unlike **`waitIn()`**, function **`checkIn()`** is not blocking.

With **`close_usb()`** the connection can be closed.

6.3.2.2 Example Scripts

The file **minimum_example.py** shows a simple example of how to connect and communicate with the LoRaSniffer via USB using Python.

File **cad_scan.py** shows a way to execute a CAD scan over a range of frequencies and spreading factors. (See CAD Channel Activity Detection for details on CAD).

7 Troubleshooting

Table 7- Troubleshooting.

Issue	Check
Battery doesn't charge.	Confirm that USB cable is plugged in and the CHG indicator is lit.
The Mobile App doesn't connect with the LoRaSniffer.	Confirm that the LoRaSniffer is waiting for a connection (SYS LED is blinking accordingly).
	Make sure that the Bluetooth is activated in your Tablet or Smartphone.
The LoraTerminal Software (Windows®) doesn't connect with the LoRaSniffer.	Confirm that the USB cable is plugged in.
	Make sure that the USB cable is connected, before switching the LoRaSniffer power on.

8 Document revision history

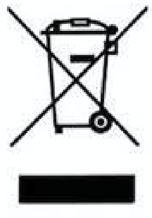
Table 8- Document revisions.

Rev.	Author	Date	Comment
1.0	Paulo Martins	29-09-2022	Initial document release.

9 Disposal and recycling

Dispose the LoRaSniffer to a recycling point for electronic equipment.

Please contact your local authority for further details of your nearest designated collection point.



10 Disclaimer and Copyright Notice

The information in this document is subject to change without notice.

Use of this radio is regulated by regional laws! The user is the sole responsible for the lawful use of this radio.
JSIO Lda. declines any responsibility for any damage resulting from miss-use of this radio, accidentally or intentionally.

This device is intended for use for engineering development, demonstration or evaluation purposes only.

This device does not fall within the scope of the European Union and FCC directives regarding electromagnetic compatibility and therefore may not meet the technical requirements of these directives.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.