LoRa Studio User Guide

Welcome to LoRa® Studio v1.1.0!

LoRa Studio is an application that demonstrates the capabilities and features of Semtech® radio chips. It allows users to configure, test, and visualize the performances of the Gen 4 series in various scenarios. All this without writing even one line of code!

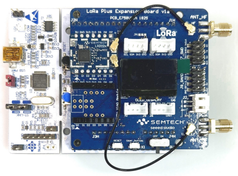
Please make sure you are up to date with the latest version by visiting the LoRa PLUS webpage!

Getting Started

Supported development kits

LoRa Studio supports the following development kits:

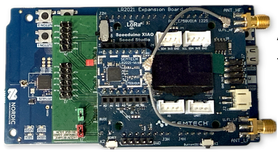
- LR2021 ARDUINO® LEGACY (STM32®)



Based on the STM32L4 platform, this dev kit is composed of a Nucleo® STM32L476RG, an adaptation board to handle the Arduino pinning, and the LR2021 radio expansion board. This kit comes as 3 different part numbers depending on the region:

  - `LR2021EVK1XBS1_ARDUINO_LEGACY` : LoRa Plus™ Development Kit, LR2021, 868MHz for Europe
  - `LR2021EVK1XCS1_ARDUINO_LEGACY` : LoRa Plus™ Development Kit, LR2021, 915MHz for North America
  - `LR2021EVK1XGS1_ARDUINO_LEGACY` : LoRa Plus™ Development Kit, LR2021, 490MHz for China and Asia

- LR2021 ARDUINO nRF54L15 Dev Kit



Based on the nRF54L15 platform, this dev kit is composed of a nRF54L15-DK, an adaptation board to handle the Arduino pinning and the LR2021 radio expansion board: This kit comes as 3 different part numbers depending on the region:

  - `LR2021EVK1XBS1_ARDUINO` : LoRa Plus™ Development Kit, LR2021, 868MHz for Europe
  - `LR2021EVK1XCS1_ARDUINO` : LoRa Plus™ Development Kit, LR2021, 915MHz for North America
  - `LR2021EVK1XGS1_ARDUINO` : LoRa Plus™ Development Kit, LR2021, 490MHz for China and Asia

- LR2021 XIAO Dev Kit

Based on the Seeeduino XIAO platform, this dev kit is composed of the latest integration of the nRF54L15 MCU in the compact XIAO form factor, and the LR2021 radio, either as a module or as an expansion board. This kit comes as 3 different part numbers depending on the region:

  - `LR2021EVK1XBS1_XIAO` : LoRa Plus™ Development Kit, LR2021, 868MHz for Europe
  - `LR2021EVK1XCS1_XIAO` : LoRa Plus™ Development Kit, LR2021, 915MHz for North America
  - `LR2021EVK1XGS1_XIAO` : LoRa Plus™ Development Kit, LR2021, 490MHz for China and Asia

Drivers Installation

- STM32L4 based kits

On Windows, to make sure that you can properly connect to this kit, you need to install the ST-LINK USB driver available here: ST-LINK USB driver.

On Linux, you can install the *stlink tool* suite following the instruction on the GitHub page but make sure that your OS version is supported.

Otherwise, you need to create a udev rule to allow non-root access to the device:

Create a file named `/etc/udev/rules.d/49-stlinkv2-1.rules` with the following content:

Copy

```
SUBSYSTEMS=="usb", ATTRS{idVendor}=="0483", ATTRS{idProduct}=="374a", \
    MODE:="0666", \
    SYMLINK+="stlinkv2-1"
SUBSYSTEMS=="usb", ATTRS{idVendor}=="0483", ATTRS{idProduct}=="374b", \
    MODE:="0666", \
    SYMLINK+="stlinkv2-1"
SUBSYSTEMS=="usb", ATTRS{idVendor}=="0483", ATTRS{idProduct}=="3752", \
    MODE:="0666", \
    SYMLINK+="stlinkv2-1"
```

Then reload the udev rules with the command:

```
sudo udevadm control --reload-rules && sudo udevadm trigger
```

After this, unplug and replug the device. You should now have access to it without needing root privileges.

In both cases, once properly installed, you should see a mass storage device usually named NUCLEO_L476RG when plugging the kit to your computer. You should also see a serial port associated with the device.

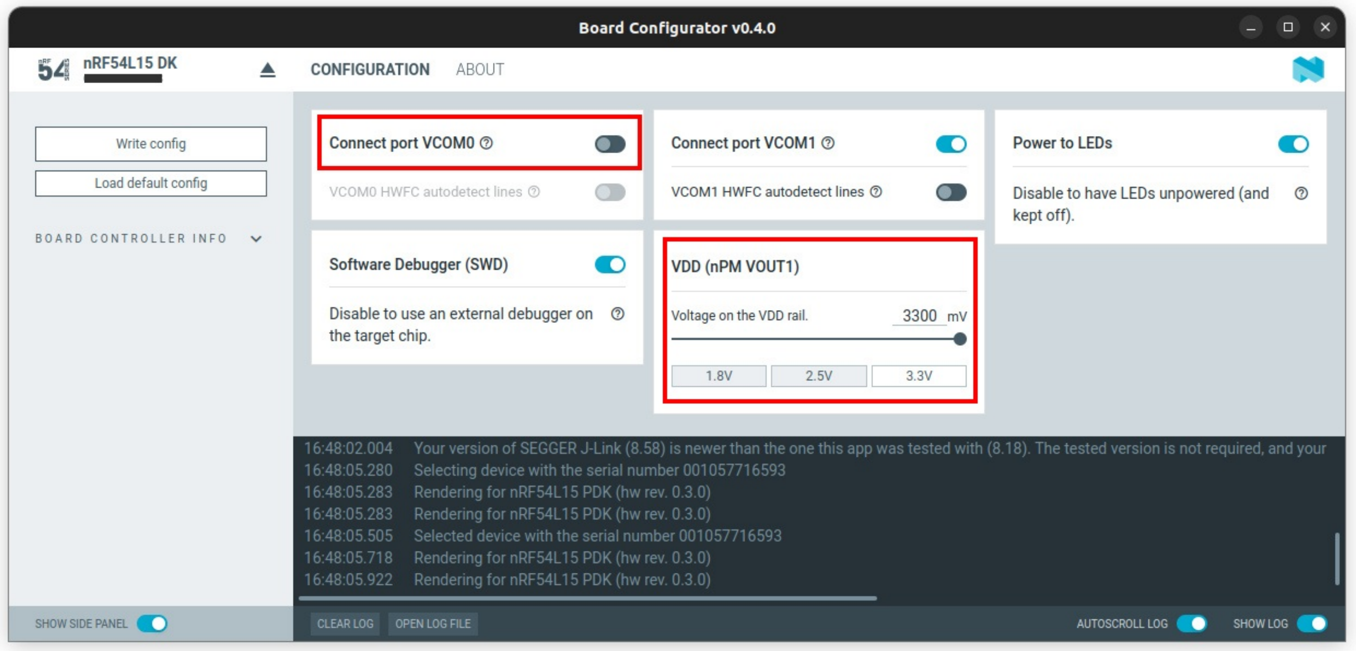
- nRF54L15-DK based kits

To properly handle the Nordic Semiconductor® dev kit, you need to install the nRF Connect for Desktop tool from Nordic website.

Since the nRF54L15-DK is based on a J-Link debugger, you also need to install the J-Link driver from Segger website.

Both software packages are available for both Windows and Linux.

Once both software packages are installed, plug the nRF54-DK to your computer. You should see 2 serial ports associated with the device. You will need to deactivate the first one (the one with the lowest number) and set the power voltage to 3.3V as shown below:



Connect port VCOM0: off
VCOM0 HWFC autodetect lines: off
Connect port VCOM1: on
VCOM1 HWFC autodetect lines: off
Power to LEDs: on
Disable to have LEDs unpowered (and kept off).
Software Debugger (SWD): on
Disable to use an external debugger on the target chip.
VDD (nPM VOUT1)
Voltage on the VDD rail. 3300 mV
1.8V 2.5V 3.3V

```
16:48:02.004   Your version of SEGGER J-Link (8.58) is newer than the one this app was tested with (8.18). The tested version is not required, and your
16:48:05.280   Selecting device with the serial number 001057716593
16:48:05.283   Rendering for nRF54L15 PDK (hw rev. 0.3.0)
16:48:05.283   Rendering for nRF54L15 PDK (hw rev. 0.3.0)
16:48:05.505   Selected device with the serial number 001057716593
16:48:05.718   Rendering for nRF54L15 PDK (hw rev. 0.3.0)
16:48:05.922   Rendering for nRF54L15 PDK (hw rev. 0.3.0)
```

Make sure to write back the change by clicking on the *Write Config* button. You can now close the nRF Connect application for Desktop.

You should now be able to use LoRa Studio with the nRF54L15-DK based kits.

- Seeeduino XIAO based kits

On Windows, good news : nothing to do! The module is plug and play.

On Linux, you need to create a udev rule to allow non-root access to the device:

Create a file named `/etc/udev/rules.d/99-xiao.rules` with the following content:

```
SUBSYSTEMS=="usb", ATTRS{idVendor}=="2886", ATTRS{idProduct}=="0066", \
    MODE:="0666", \
    SYMLINK+="nrf54"
ATTRS{idVendor}=="2886", ATTRS{idProduct}=="[08]02d", MODE="0666", ENV{ID_MM_DEVICE_IGNORE}="1", \
    ENV{ID_MM_PORT_IGNORE}="1"
```

Then reload the udev rules with the command:

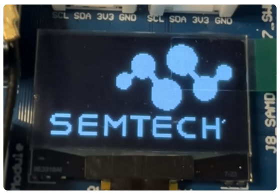
```
sudo udevadm control --reload-rules && sudo udevadm trigger
```

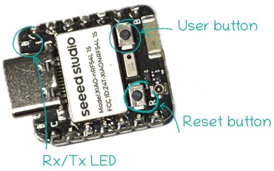
After this, unplug and replug the device. You should now have access to it without needing root privileges.

Using LoRa Studio

Connecting your development kit

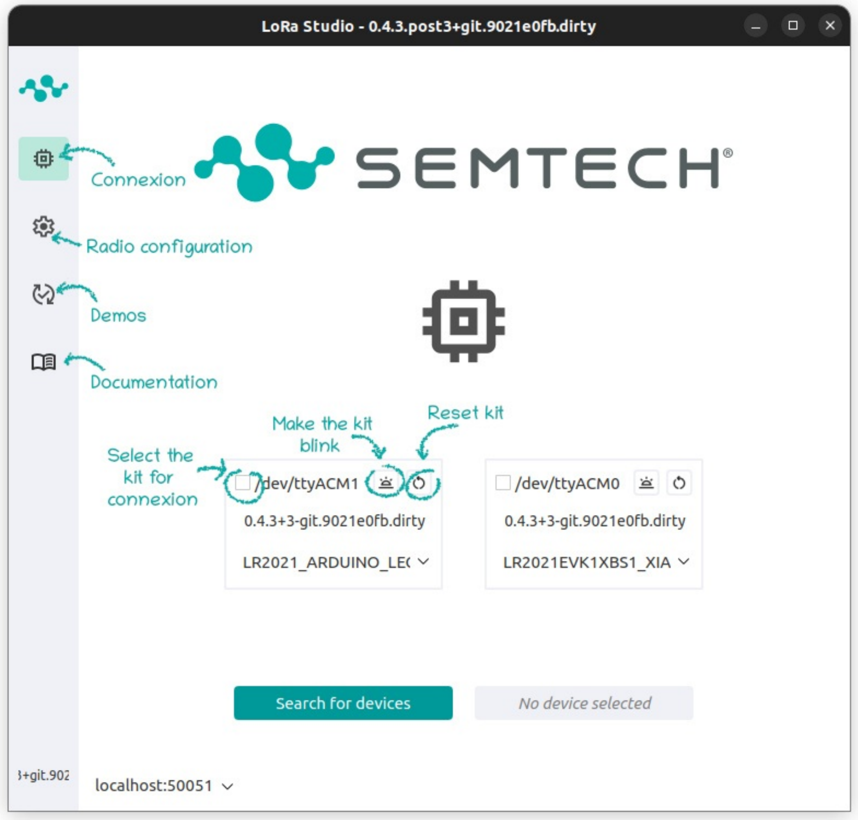
Once the drivers properly installed, you can connect your kits to your computer. If the kit is equipped with the small OLED screen, you shall see a Semtech logo indicating that the kit is powered on. If it does not appear, please reset manually your kit by pressing the reset button.





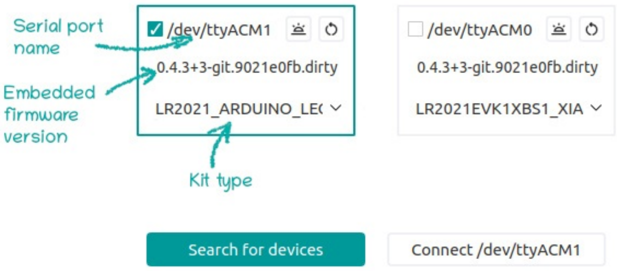
> **Note:** If you have a Seeeduino XIAO based kit, you will have to reset with the button on the XIAO board, not the one on the expansion board.

Launch LoRa Studio, here is what it looks like:



At launch, the three first times you will be redirected to this documentation page. After that you will see the Connexion Page. Until you connect a kit, you can only navigate to the documentation page or stay on the connexion page.

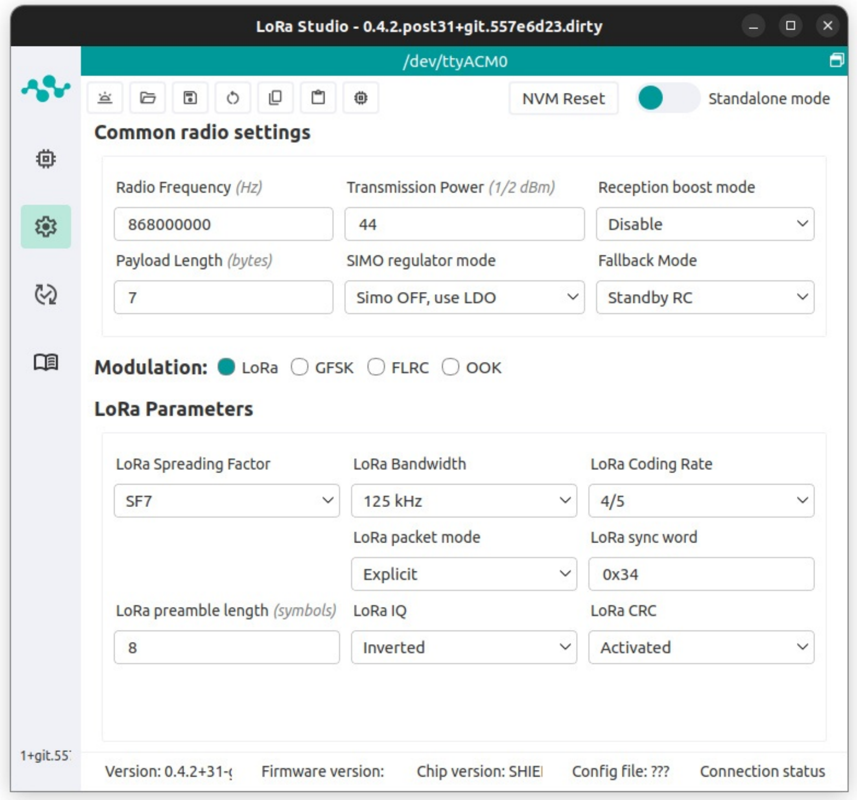
If you have multiple kits connected you can identify which serial port is associated with which kit by clicking on the 🚨 button. This will make LEDs blink on the corresponding kit.
You can also reset your kit using the ↻ button.



At startup, LoRa Studio detects the connected development kit(s) and displays the kit info as shown above. If no device is detected, make sure that the kit is properly connected and that the drivers are correctly installed or see the troubleshooting section below.

> **Important Note about Kit Type:** *The Kit type is automatically detected* ***BUT*** *the part number must be selected manually. If the exact reference is not correctly set, LoRa Studio may not be able to operate the kit's radio properly leading to unexpected behavior. Please verify that the Kit type matches the development kit you are using before proceeding with any operations. Please refer to the Supported Development Kits section above for the correct kit names.*

Once connected, LoRa Studio switches automatically to the Radio Configuration page.



Common radio settings

Radio Frequency *(Hz)*: 868000000
Transmission Power *(1/2 dBm)*: 44
Reception boost mode: Disable
Payload Length *(bytes)*: 7
SIMO regulator mode: Simo OFF, use LDO
Fallback Mode: Standby RC

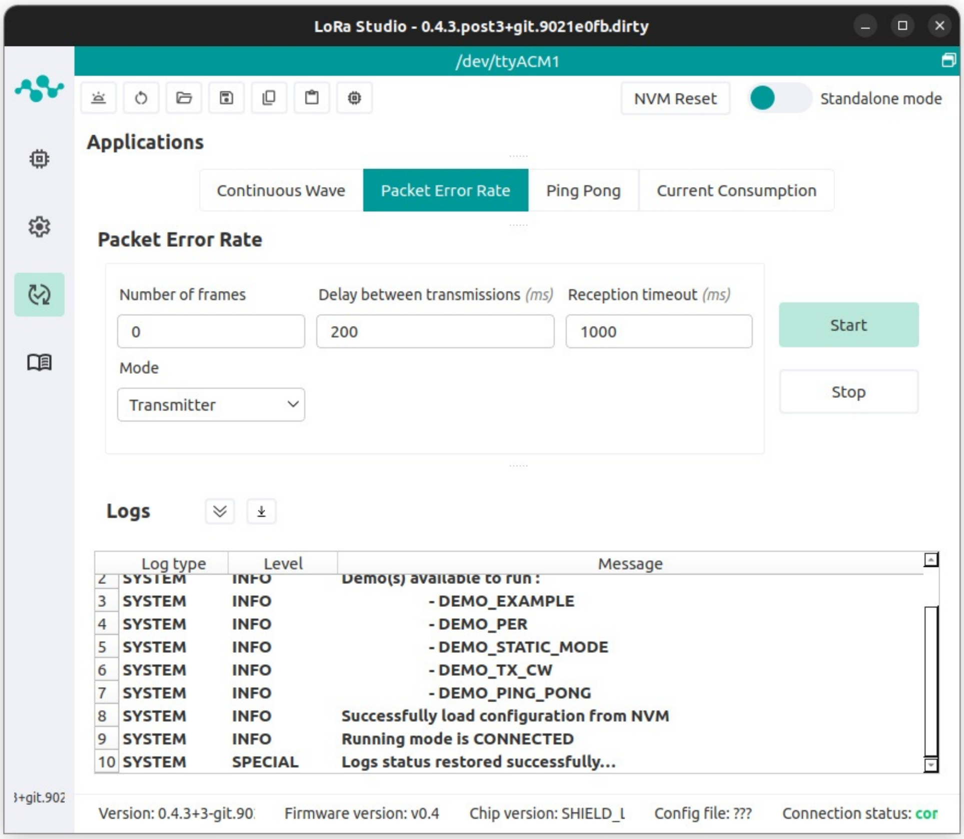
Modulation: ● LoRa ○ GFSK ○ FLRC ○ OOK

LoRa Parameters

LoRa Spreading Factor: SF7
LoRa Bandwidth: 125 kHz
LoRa Coding Rate: 4/5
LoRa packet mode: Explicit
LoRa sync word: 0x34
LoRa preamble length *(symbols)*: 8
LoRa IQ: Inverted
LoRa CRC: Activated

From this page, you can configure the radio parameters of your kit. You can select, among others, the frequency, the output power, the modulation, and its parameters.
You can also, like on the connexion page, identify and reset the connected kit using the 🚨 and ⟳ buttons.

Once you are satisfied with your configuration, you can go to the Application page by clicking on the 🔄 icon in the left menu.



Applications

Packet Error Rate

Number of frames: 0

Delay between transmissions *(ms)*: 200

Reception timeout *(ms)*: 1000

Mode: Transmitter

Logs

```
   Log type   Level     Message
2  SYSTEM     INFO      Demo(s) available to run :
3  SYSTEM     INFO              - DEMO_EXAMPLE
4  SYSTEM     INFO              - DEMO_PER
5  SYSTEM     INFO              - DEMO_STATIC_MODE
6  SYSTEM     INFO              - DEMO_TX_CW
7  SYSTEM     INFO              - DEMO_PING_PONG
8  SYSTEM     INFO      Successfully load configuration from NVM
9  SYSTEM     INFO      Running mode is CONNECTED
10 SYSTEM     SPECIAL   Logs status restored successfully...
```

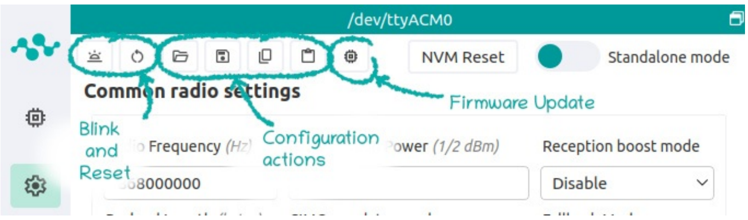
3+git.902 Version: 0.4.3+3-git.90 Firmware version: v0.4 Chip version: SHIELD_L Config file: ??? Connection status: cor

From this page, you can select, set parameters and run demonstration applications.

You can also, like on the previous pages, identify and reset the connected kit using the and buttons.

Finally, on both the Radio Configuration and Application pages, a tool bar allows you to :

- Save the configuration to a JSON file. This will open a dialog to select the location and name of the file.
- Load a configuration from a file.
- Copy the configuration to the clipboard: JSON content can be pasted to wherever you would like or...
- ... paste it on the page of another connected kit to ensure they both use the exact same parameters.
- Update the firmware of the connected kit. This will open a dialog to select the firmware file. The firmware file must be a .bin file and correspond to your kit.



Running demonstration application

From the Application page, you can select and run different demonstration applications. The available applications are:

- Continuous Wave
- Ping Pong
- PER (Packet Error Rate) Test
- Static modes for current consumption measurement

Each application has its own parameters that are set while starting it. You can run the application while connected to the computer or in standalone mode powering the kit(s) by power banks or batteries.

Connected mode

In connected mode, the kit is powered by the USB connection to your computer and the application running on your computer controls the kit. The logs of the application are displayed in the log area of the application and some basic info is shown on the OLED screen (if available).

To run an application in connected mode, you need to:

1. Select your kit(s) on the connection page and click on the Connect button, the application loads and opens automatically the Radio Configuration page.
2. Configure the radio parameters as desired
3. Go to the Application page
4. Configure the application parameters as desired
5. Click on the `Start` button to start the application
6. Stop the application, by clicking on the `Stop` button

You can also save the logs to a text file by clicking on the ⤓ button above the log area.

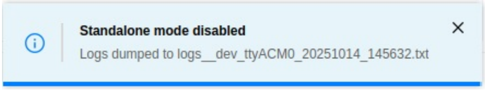
Standalone mode

In standalone mode, the kit(s) is/are powered by an external power source (power bank, batteries, etc.) and the application running on the kit controls the demo. Info about the status of the application is displayed on the OLED screen of the kit(s) (if available) and the logs are stored in the NVM of the kit(s).

To run an application in standalone mode, you need to:

1. Plug the kit to your computer
2. Select your kit on the connection page and click on the Connect button, the application loads and opens automatically the Radio Configuration page.
3. Configure the radio parameters and the application parameters as desired.
4. Click on the `Start` button to start the application.
5. Stop the application, by clicking on the `Stop` button. This step is mandatory to properly save the configuration and application parameters in the NVM of the kit.
6. Set the device in standalone mode by sliding the standalone button to the right. The inner button will turn green and the configurations fields will be disabled (greyed) to indicate that the kit is now in standalone mode:

   The kit is now ready to run in standalone mode and will stay in this mode until you deactivate it.
7. Unplug the kit from the computer and power it with an external power source and close LoRaStudio
8. To start the application in standalone mode, press the user button on the kit
9. Make sure to stop the application in standalone mode by pressing the user button on the kit again
10. Connect back the kit to your computer.
11. Open again LoRa Studio, go back to the connection page, search for devices, select and connect your kit.
12. To retrieve the logs of the application, go to the Application page and deactivate the standalone mode. The inner button will turn grey and the configuration fields will be enabled again to indicate that the kit is now back in connected mode. Also a toast will appear to indicate that the logs have been retrieved and saved to a text file with its name in the LoRa Studio folder:

    Standalone mode disabled

    Logs dumped to logs__dev_ttyACM0_20251014_145632.txt

    It also opens it for you automatically using your default text editor.

Application details

Continuous Wave Demo

The Continuous Wave demo allows you to transmit a continuous wave signal for testing purposes.

This application can be used to test the range of the radio, the sensitivity of a receiver or to perform regulatory tests.

It does not have any application specific parameters. You only need to configure the radio parameters. Just use the `Start` button to start the transmission and the `Stop` button to stop it.

Ping Pong Demo

The Ping Pong demo demonstrates bi-directional communication between two devices. This application obviously needs two kits to work with one another. *TIP: Configure only one kit properly then use the copy/paste* ⎘ 📋 *configuration functionality to make sure you are running with the same parameters on both kits.*

The Ping Pong demo requires the following application parameters:

- Delay before transmission: set the delay before the first packet is sent in ms
- Delay between packets: set the delay in ms between the reception and the retransmission of the packet
- Sync packet threshold: Number of packets to transmit without a response to give enough margin for synchronization
- Prefix size: Number of bytes used as prefix in the packet
- Reception timeout: Time in ms to wait for a response before considering that the packet is lost

When starting the application, the kit will start transmitting packets and wait for a response. If a response is received, it will take its content (a counter) and increase its value before retransmitting the packet after the configured delay. If no response is received within the timeout, it will consider the packet lost and transmit a new one after the configured delay and put itself in reception mode again.

Since no actual synchronization is done between the two kits, it is possible that both kits transmit at the same time and therefore miss each other's packets. The synchronization threshold allows to mitigate this issue by transmitting a certain number of packets without waiting for a response. This increases the chances that one of the packets will be received by the other kit and that the communication can start.

Packet Error Rate

The Packet Error Rate evaluates link quality and performance metrics. This application needs two kits to work with one set as transmitter and the other as receiver. *TIP: Configure only one kit properly then use the copy/paste* ⎘ 📋 *configuration functionality and change the mode on the second kit afterwards.*

Select Packet Error Rate and setup the application parameters:

- Number of frames: Number of frames to receive (as goal) for the reception mode, on which the PER will be calculated. If set to 0, the kit will receive indefinitely and PER will be calculated on the cumulated packets
- Delay between transmissions: Delay in ms between each frame the transmitter sends

- Reception timeout: Time in ms to wait before considering a frame lost
- Mode: Select if the kit will be a transmitter or receiver

A live calculated PER will be shown in the logs and on the OLED screen for you to see along with the number of received packets.

Static modes for current consumption measurement

This application allows you to set the kit in different static modes to measure the current consumption.

1. Once connected to your kit, the application loads and opens automatically the Radio Configuration page.
2. Configure the radio parameters (frequency, power, modulation...)
3. Go the the Application page
4. Select Static modes and setup the application parameters:
    - Mode: Select the static mode you want to set the kit in
5. Click on the `Start` button to set the kit in the selected static mode
6. The kit will enter the selected static mode
7. Click on the `Stop` button to exit the static mode and return to normal operation

Firmware Update

To update the firmware of your kit, you need to connect it to your computer and open LoRa Studio. Then, on either the radio configuration page or the application page, click on the button in the toolbar. This will open a file dialog to select the firmware file. The application will automatically open the folder where the firmwares can be found. Choose the file that corresponds to your kit.

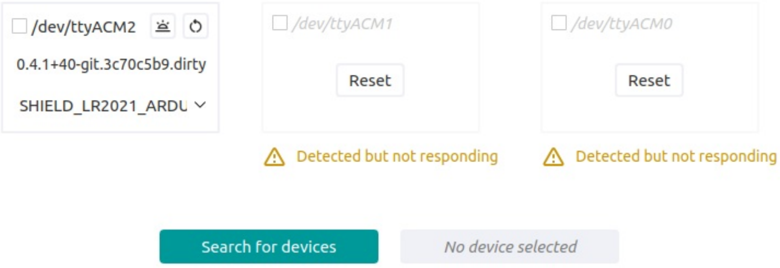
Once the firmware file is selected, the update process will start automatically. A progress bar will show the progress of the update. Do not disconnect or turn off the kit during the update process.

Once the update is complete, a message will indicate that the update was successful. You can now use the new firmware.

Troubleshooting

Common connection issues

Main page shows a device is "Detected but not responding":



□ /dev/ttyACM2
0.4.1+40-git.3c70c5b9.dirty
SHIELD_LR2021_ARDU

□ /dev/ttyACM1
Reset
⚠ Detected but not responding

□ /dev/ttyACM0
Reset
⚠ Detected but not responding

This usually means that the application can see a serial port to communicate with the device, but the device is not responding to commands. It can be that this device is not a supported development kit or the firmware on the kit is not compatible with LoRa Studio. It can also be that the kit has some UART issues. You can try the following steps to resolve the issue:

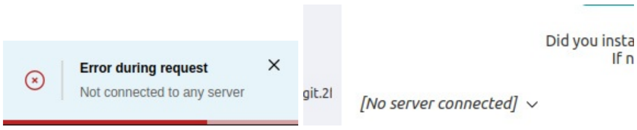
- Click on the `Search for devices` button.
- Click on the "Reset" button associated to the serial port you believe your board is connected to (you can use the identify button to be sure) and then on the "Search" button again.
- Physically reset the kit by pressing the reset button and then click on the "Search for devices" button again.

> **Note:** *For nRF54L15-DK based kits, even though you made sure that you have deactivated the first serial port in nRF Connect, two serial ports will be detected, and one of them will always be not responsive. This does not affect the functionality of LoRa Studio and can be ignored.*

No device shows at all

- Verify driver installation: Double-check that all necessary drivers are installed and up to date.
- Restart the application: Sometimes, simply restarting LoRa Studio can resolve connection issues.
- Try another USB port or cable: Faulty USB ports or cables can cause connection problems. Switch to a different port or use a different cable to see if that resolves the issue.
- Check device power: Ensure that the development kit is powered on and functioning correctly.
- Check for conflicts: Ensure that no other applications are using the same serial port that LoRa Studio is trying to access.

No server connection



Error during request

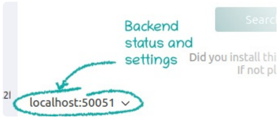
Not connected to any server

Did you insta
If n

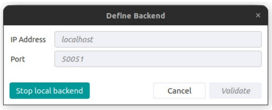
git.2l

[No server connected]

It can happen that the backend does not start properly and hence no device can be connected. You can verify and fix this issue by clicking on the server info at the bottom left of the connexion page.



A dialog box will open, click on the `Stop local backend` button. Some toast will indicate the stopping of the backend, the button shall change into `Start local backend`. Click on this button again, this shall restart the backend properly.



Define Backend

IP Address: localhost

Port: 50051