SX1302 LoRa® Corecell Gateway

Reference Design User Guide

Contents

1 Introduction 5
1.1 Purpose of this Document 5
1.2 SX1302 Corecell Gateway Kits 5
2 SX1302 Corecell Gateway Region Versions 6
2.1 EU868 / US915 / AS923 Versions 6
2.2 CN490 Version 7
3 Hardware Overview 8
3.1 Absolute Maximum Ratings 8
3.2 RF Front-End Architecture 8
3.3 LoRa Corecell Reference Design Block Diagrams 9
3.4 Power Consumption 11
3.5 Power Up/Down Sequence 11
4 Software Overview 12
4.1 Use With Raspberry Pi 14
4.2 Corecell Reference Design, Interface Board & Raspberry Pi Connection 14
4.2.1 USB Interface for EU868 / US915 / AS923 Corecell Gateway 14
4.2.2 USB Interface for CN490 Corecell Gateway 15
4.3 Raspberry Pi Installation 16
4.3.1 Raspberry Pi Image Software Installation 16
4.3.2 Setup 17
4.3.3 Update & Configure Raspberry Pi 19
4.3.4 Compile Semtech HAL & Packet Forwarder 20
4.3.5 Update STM32 MCU Firmware 21
4.3.6 Get Gateway's Unique ID 22
4.3.7 Semtech HAL Compilation Check 23
4.3.8 Test HAL TX 24
4.3.9 Configure 'Listen Before Talk' 25
4.3.10 Configure Spectral Scan 25
4.3.11 Run Spectral Scan 26
4.3.12 Activate GPS: Fine Timestamp & Class B Beacon 27
4.3.13 Run Packet Forwarder 28
5 RF Parameters Tuning 29
5.1 Spreading Factors SF5 & SF6 29
5.2 Typical JSON File 29
6 References 34
7 Glossary 35
8 Revision History 36

List of Figures

Figure 1: LoRa® Corecell Reference Design V3 (EU898/US915/AS923) 6
Figure 2: LoRa Corecell Reference Design V1 (CN490) 7
Figure 3: LoRa Corecell Reference Design V3 Block Diagram (EU868/US915/AS923) 9
Figure 4: LoRa Corecell Reference Design V1 Block Diagram (CN490) 9
Figure 5: VCC_CORE and VCC_IO Sequencing 11
Figure 6: GW Software Overview 12
Figure 7: LoRa Corecell Reference Design, Interface Board & Raspberry Pi Connection (EU868/US915/AS923) 14
Figure 8: LoRa Corecell GW, Interface Board & Raspberry Pi Connection (CN490) 15
Figure 9: Win32 Disk Imager 16
Figure 10: enable SSH connection on R-Pi 16
Figure 11: enable SSH connection on R-Pi 17
Figure 12: MobaXterm SSH Client with IP address 17
Figure 13: MobaXterm SSH Client 18
Figure 14: raspi-config “Expand Filesystem” 18
Figure 15: Enable SPI/I2C/UART 19
Figure 16: Git Clone 20
Figure 17: Generate rsa Key 20
Figure 18: Copy rsa Key 20
Figure 19: Executables 21
Figure 20: Util Chip ID 22
Figure 21: HAL Compilation Check Results 23
Figure 22: Listen Before Talk Configuration While Running Packet Forwarder 25
Figure 23: Spectral Scan Configuration While Running Packet Forwarder (Disabled) 25
Figure 24: Spectral Scan Results 26
Figure 25: Packet Forwarder 28

List of Tables

Table 1: List of SX1302 Corecell Gateway Kits 5
Table 2: Absolute Maximum Ratings 8
Table 3: Typical Current Consumption at 5.0V (EU868/US915/AS923) 11
Table 4: Typical Current Consumption at 5.0V (CN490) 11

1 Introduction

1.1 Purpose of this Document

This user guide introduces the Semtech LoRa® SX1302 Corecell reference designs and describes how to set them up with Raspberry Pi 3 or 4.

1.2 SX1302 Corecell Gateway Kits

Here is the list of Part Numbers of the SX1302 Corecell Gateway Kits for different regions that are covered in this document.

Part Number	Corecell Gateway PCB version #	Region	Frequency Band [MHz]
SX1302CSS868GW1	E539V03A	Europe - **EU868**	863 to 870
SX1302CSS915GW1	E539V03A	North America - **US915**	902 to 928
SX1302CSS923GW1	E539V03A	Japan - **AS923**	920 to 928
SX1302CSS490GW1	E616V01A	China / Asia - **CN490**	470 to 510

Table 1: List of SX1302 Corecell Gateway Kits

2 SX1302 Corecell Gateway Region Versions

2.1 EU868 / US915 / AS923 Versions

For EU868, US915, and AS923, the reference design based on the PCB version # E539V03A, consists of a multi-channel SX1302 baseband IC, two SX1250 RF front-end transceivers, an STM32 MCU, one SX1261 RF transceiver, a 27dBm front-end module, and all the necessary filters and power supplies to deliver a high-performance 8-channel LoRa gateway.

The LoRa gateway supports a USB interface to connect to a Raspberry Pi (R-Pi) host computer. It can support an SPI interface, but only with a change of the components BOM for the module assembly, or with the use of the previous PCB version # E539V01A (not available anymore, replaced by USB version).

The LoRa gateway also supports the 'Listen before Talk' (LBT) function, which is required in countries like Japan and Korea. It also has a provision to run a spectral scan.

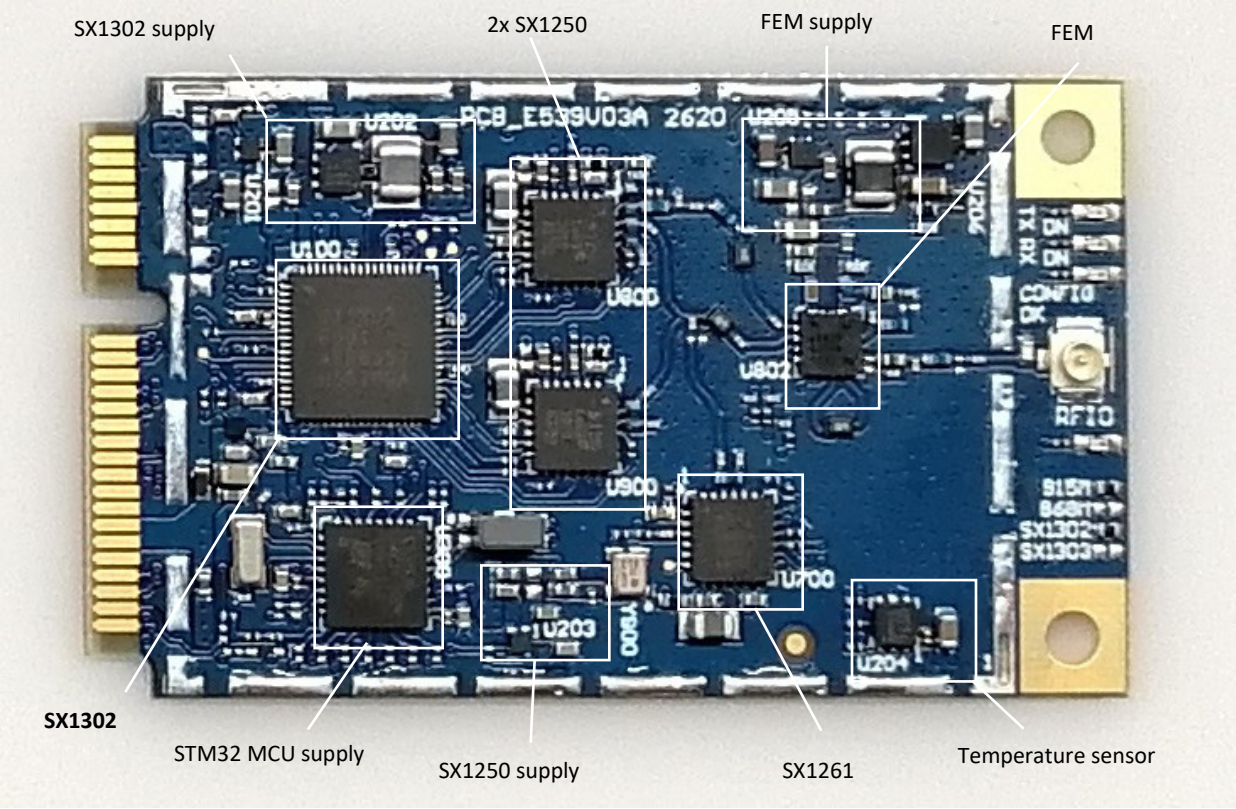


Figure 1: LoRa® Corecell Reference Design V3 (EU898/US915/AS923)

2.2 CN490 Version

For CN490, the reference design based on the PCB version # E616V01A consists of a multi-channel SX1302 baseband IC, two SX1250 RF front-end transceivers, an STM32 MCU, one SX1261 RF transceiver, and all the necessary filters and power supplies to deliver a high-performance 8-channel LoRa gateway.

It supports:

- USB interfaces to connect to a Raspberry Pi host computer,
- 'Listen before Talk' (LBT) feature,
- Running spectral scans.

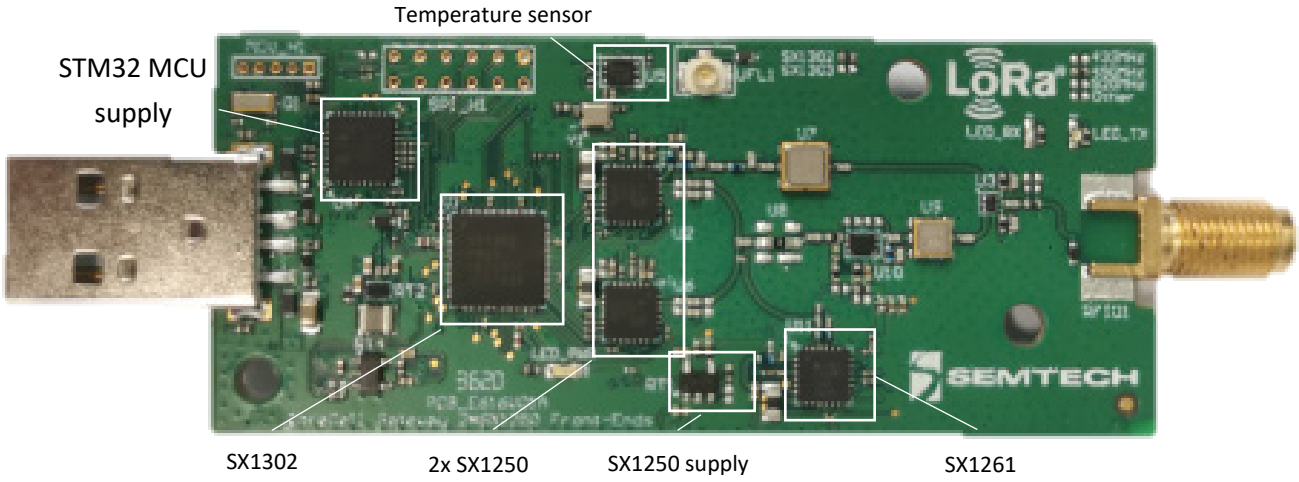


Figure 2: LoRa Corecell Reference Design V1 (CN490)

3 Hardware Overview

3.1 Absolute Maximum Ratings

Item	Minimum	Typical	Maximum	Unit
Maximum Supply Voltage	-0.3	5.0	5.5	V
Operating Temperature	-40	25	85	°C
Maximum RF Input Level	-	-	+10	dBm

Table 2: Absolute Maximum Ratings

3.2 RF Front-End Architecture

The RF front-end architecture of the Corecell Gateway reference design displays the following characteristics:

- Half-duplex mode i.e., can't receive and transmit simultaneously
- Ability to simultaneously receive 8 LoRa channels with multi-data rates (SF5 ~ SF12 / 125kHz) + 2 mono-data rates (LoRa 250 / 500kHz and FSK 50kbps)
- Maximum transmit output power (EU868/US915) = +27dBm
- Maximum transmit output power (CN490) = +17dBm
- Typical sensitivity level (EU868/US915):
    - -140dBm at SF12 BW 125kHz
    - **-125dBm** at SF7 BW 125kHz
    - -110dBm at FSK 50kbps
- Typical sensitivity level (AS923):
    - -141dBm at SF12 BW 125kHz
    - **-126dBm** at SF7 BW 125kHz
    - -110dBm at FSK 50kbps
- Typical sensitivity level (CN490):
    - -141dBm at SF12 BW 125kHz
    - **-126dBm** at SF7 BW 125kHz
    - -109dBm at FSK 50kbps
- Ability to work in hostile RF environments such as close to cellular mobile phones, Wi-Fi routers, and Bluetooth devices

3.3 LoRa Corecell Reference Design Block Diagrams

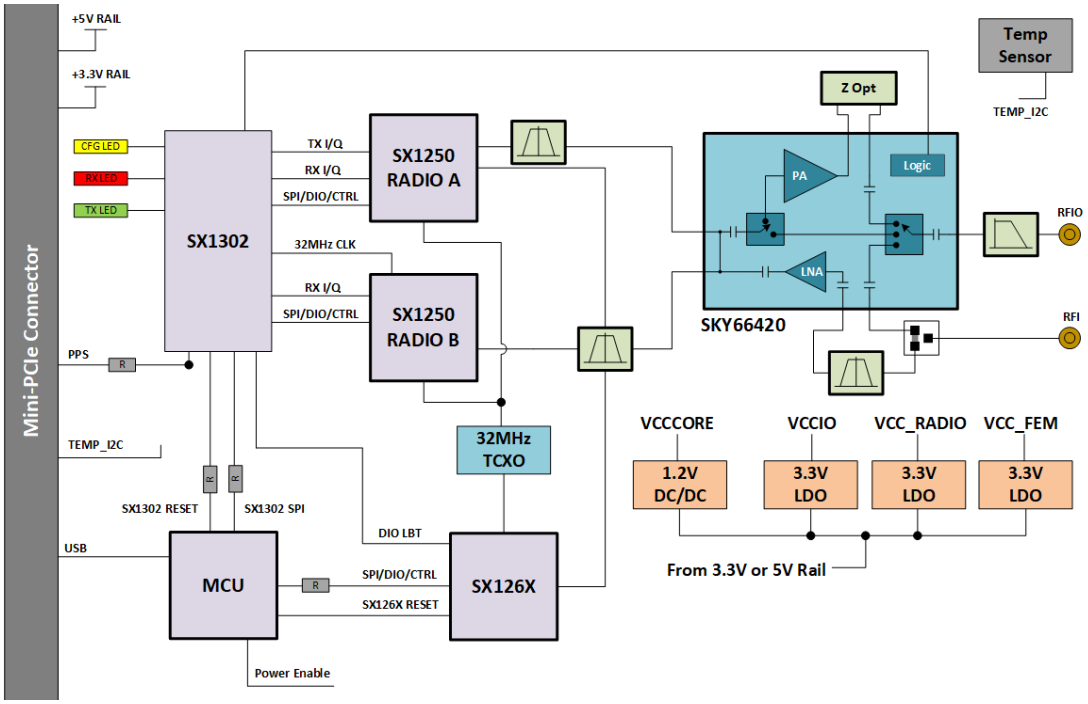


Figure 3: LoRa Corecell Reference Design V3 Block Diagram (EU868/US915/AS923)

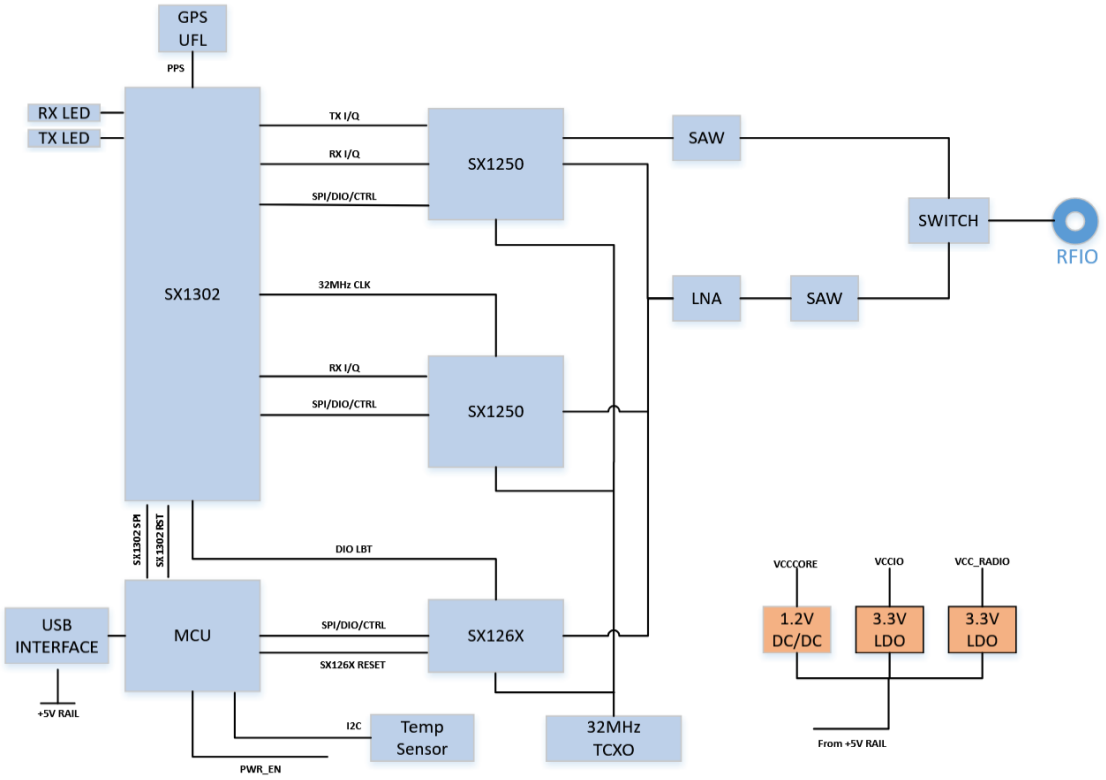


Figure 4: LoRa Corecell Reference Design V1 Block Diagram (CN490)

- One SX1302 and two SX1250 transceivers are used to complete an eight-channel LoRa concentrator:
    - The SX1302 digital baseband chip is a highly innovative digital signal processing engine equipped with 16 modulators, capable of demodulating 64 combinations of LoRa packets, and 2 separate modulators.
    - The SX1250 is a highly integrated RF to IQ transceiver capable of supporting multiple modulation schemes over the 150-960MHz ISM frequency bands, there are two SX1250s in the design to increase the gateway bandwidth
- The STM32 MCU serves as a translator between the SX1302 and Raspberry Pi when using USB mode. The MCU converts the USB data from the Raspberry Pi into SPI while talking to the SX1302 and vice versa.
- For EU868/US915/AS923, the SX1261 works in receiver mode and listens for any interference signal before transmitting data. It is used for the 'Listen before Talk' feature and the spectral scan.
- For EU868/US915/AS923, the onboard Mother board's main requirements to control signals from/to the Mini PCIe and the SX1302 are:
    - 1 SPI: coming from the host to the SX1302 SPI interface (in SPI mode)
    - 1 USB: coming from the host to the MCU (in USB mode)
    - 1 I2C: coming from the host to the temperature sensor I2C interface (in SPI mode)
    - Power Enable line
    - SX1302 reset line
    - PPS (only used for class B beacon and fine timestamp)
- For CN490, the control signals from/to the 2*5 connector (2.00mm pitch) require:
    - 1 SPI: coming from the host to the SX1302 SPI interface
    - 1 USB: coming from the host to the MCU
    - 1 I2C: coming from the host to the temperature sensor I2C interface
    - Power Enable line
    - SX1302 reset line
    - PPS (only used for class B beacon and fine timestamp)
    - Signal input via the Hirose U.FL connector on the board

3.4 Power Consumption

Mode	Description	Typical Current Consumption	Unit
8 RX channels ON TX OFF	HAL packet_forwarder	39	mA
8 RX channels OFF TX ON at 27dBm 868MHz	HAL util_tx_continuous	421	mA
8 RX channels OFF TX ON at 26dBm 915MHz	HAL util_tx_continuous	407	mA
8 RX channels OFF TX ON at 14dBm 868MHz	HAL util_tx_continuous	148	mA

Table 3: Typical Current Consumption at 5.0V (EU868/US915/AS923)

Mode	Description	Typical Current Consumption	Unit
8 RX channels ON TX OFF	HAL packet_forwarder	49	mA
8 RX channels OFF TX ON at 17dBm 475MHz	HAL util_tx_continuous	145	mA

Table 4: Typical Current Consumption at 5.0V (CN490)

3.5 Power Up/Down Sequence

To ensure proper control of all digital IOs during the power-up sequences of the SX1302:

- VCC_CORE (1.2V) shall be enabled before VCC_IO (3.3V) at start-up
- VCC_CORE (1.2V) shall be disabled after VCC_IO at shutdown.

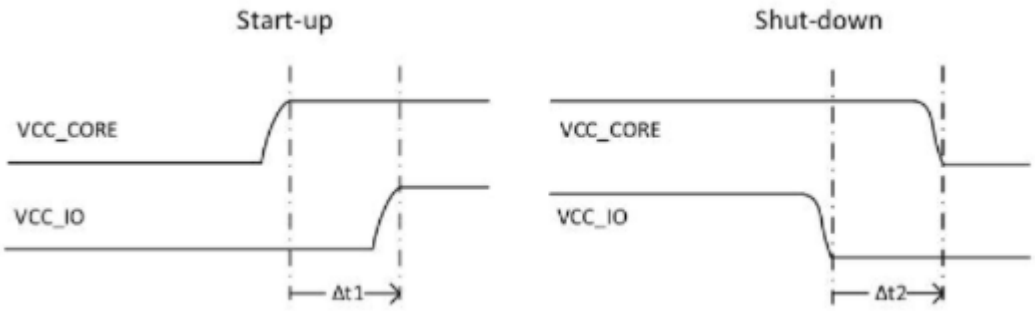


Figure 5: VCC_CORE and VCC_IO Sequencing

Δt1 and Δt2 must be equal to, or greater than 0.

4 Software Overview

The Corecell reference design software can be divided into four main parts:

- The **packet forwarder** program runs on the host of a LoRa gateway that forwards RF packets received by the concentrator to a server through an IP/UDP link, and emits RF packets that are sent by the server.
- The **sx1302_hal** is a host driver/HAL used to build the Corecell reference design which communicates, through USB or SPI interface, with a concentrator board based on Semtech SX1302 multi-channel modem and SX1250 RF transceivers.
- The **utils_boot** and **dfu-util** (MCU Firmware Updater) tools run on the host of the LoRa gateway and are used to program the STM32 MCU.
- The **spectral scan** tool runs on the host gateway and provides details of the wireless signals present in the surrounding area. The spectral scan results are stored in a csv file which can be plotted using a python script.

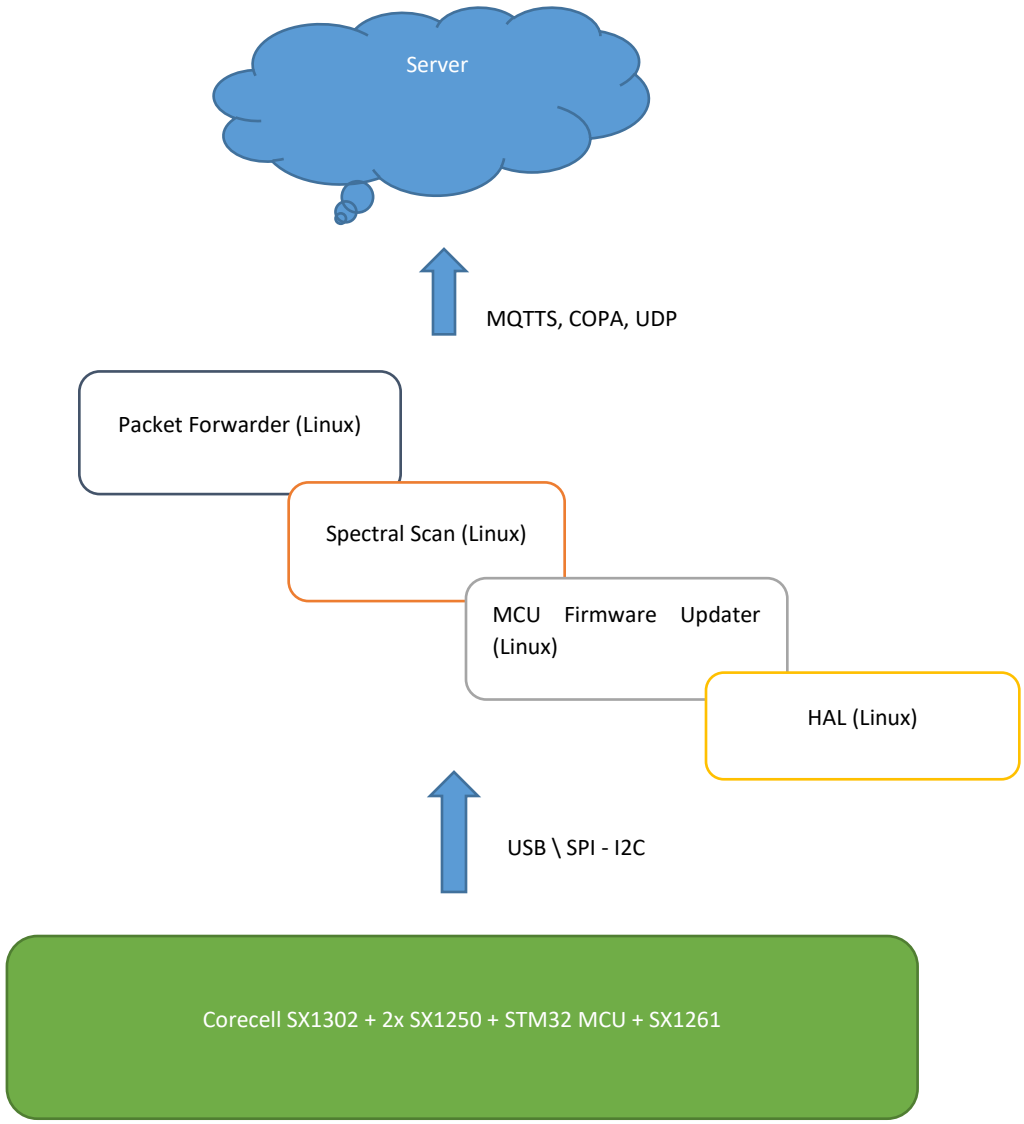


Figure 6: GW Software Overview

The packet_forwarder (gateway application), Spectral Scanner, STM32 MCU firmware updater and sx1302_hal (SX1302 control library) source code can be found in LoRa GitHub: https://github.com/Lora-net/sx1302_hal

For more details see the *readme.md* file in the following directories:

- *sx1302_hal*
- *sx1302_hal/libloragw*
- *sx1302_hal/packet_forwarder*
- *sx1302_hal/util_net_downlink*
- *sx1302_hal/util_chip_id*
- *sx1302_hal/util_boot*
- *sx1302_hal/util_spectral_scan*

For basic testing, utilities such as test_loragw_hal_tx (FSK/LoRa modulation as well as CW), test_loragw_hal_rx, are provided on the LoRa Github repository: https://github.com/Lora-net/sx1302_hal/libloragw

Note:

- The default configuration file *global_conf.json.sx1250* is given as an example and should be adapted to your design.
- Several configuration file examples are in this directory: *[PATH]/sx1302_hal/packet_forwarder*.
- If the Corecell is configured to use the USB interface, be sure to use the files with the extension ".USB".

4.1 Use With Raspberry Pi

The Semtech LoRa Corecell reference design was tested with Raspberry Pi 3 model B and Raspberry Pi 4.

https://www.raspberrypi.org/products/

4.2 Corecell Reference Design, Interface Board & Raspberry Pi Connection

The Corecell reference design V3 can be configured to work with the SX130x Corecell USB interface, PCB version # E525V04A.

4.2.1 USB Interface for EU868 / US915 / AS923 Corecell Gateway

For EU868 / US915 / AS923, simply connect the Corecell reference design to the interface board through the mini PCIe.

Connect the USB-A cable from the Raspberry Pi to the micro-USB port on the interface board, on the left of the RFIO SMA connector for LoRa antenna.

See USB cable and connections from depicted on the image below:

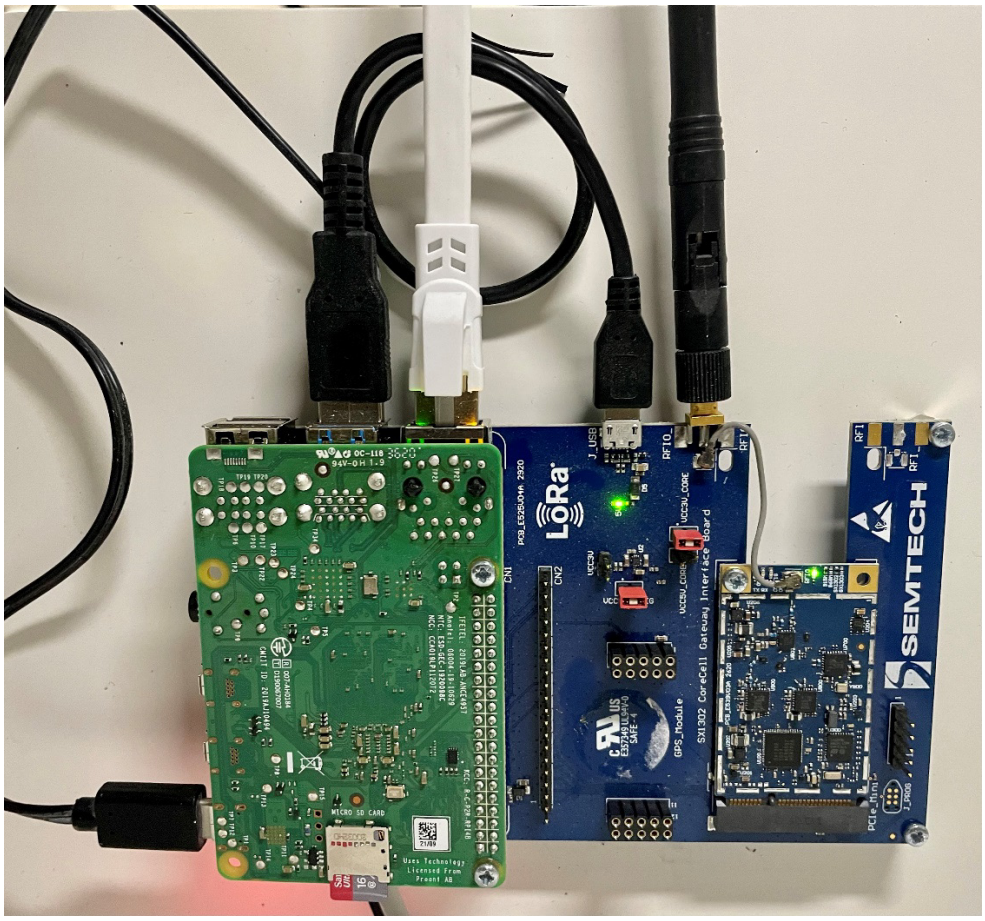


Figure 7: LoRa Corecell Reference Design, Interface Board & Raspberry Pi Connection (EU868/US915/AS923)

4.2.2 USB Interface for CN490 Corecell Gateway

For CN490, simply connect the Corecell gateway via the USB port directly to the Raspberry Pi, as depicted in the picture below:

Figure 8: LoRa Corecell GW, Interface Board & Raspberry Pi Connection (CN490)

4.3 Raspberry Pi Installation

4.3.1 Raspberry Pi Image Software Installation

Download the Raspberry Pi OS imager:

- Go to address: https://www.raspberrypi.com/software/operating-systems/#raspberry-pi-os-64-bit
- Choose and download "*Raspberry Pi OS (Legacy, 64-bit) Lite*" for R-Pi 3 or 4 version 64 bits
- Choose "Write the image previously downloaded on the SD card" by using *Win32 Disk Imager* software
- Download *Win32 Disk Imager* software for installation: https://sourceforge.net/projects/win32diskimager/
- Write the "*Raspberry Pi OS (Legacy, 64-bit) Lite*" image on the selected microSD card for R-Pi

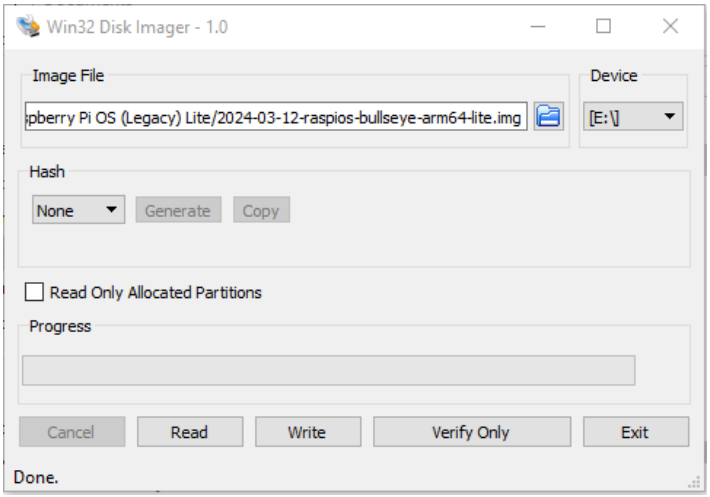


Figure 9: Win32 Disk Imager

Enable SSH:

- Enable SSH by placing a file named "ssh" (without any extension) onto the boot partition of the microSD card:

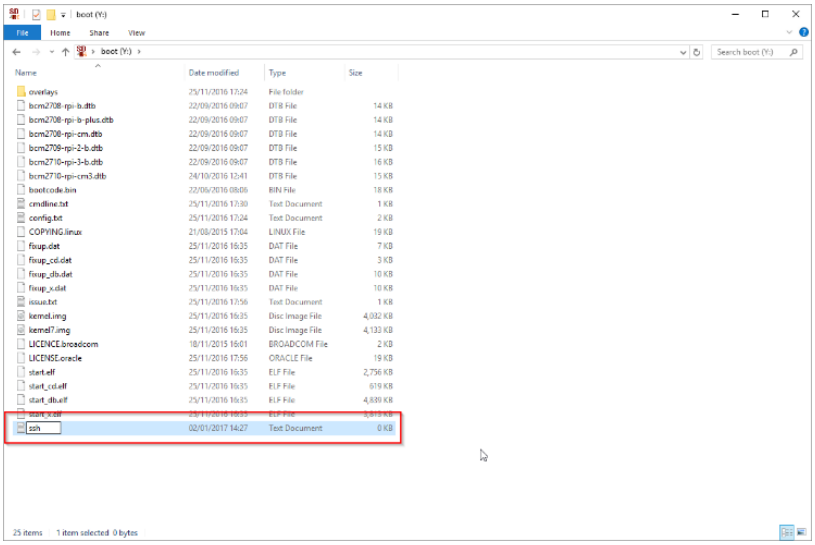


Figure 10: enable SSH connection on R-Pi

4.3.2 Setup

Before starting this procedure setup, check the connection with USB cable from the R-Pi to the micro-USB port located on the interface board for EU868 / US915 / AS923 Corecell Gateway versions, as shown on Figure 7. For CN490 version, USB Corecell Gateway must be directly plugged to the R-Pi, as shown on Figure 8.

It's now mandatory to define a new username and password for first time setup, so you will need to connect a monitor to the R-Pi with an HDMI cable and keyboard for this.

R-Pi 4 requires a micro-HMDI cable, like P/N: T7732AX or equivalent. HDMI cable is not provided with the Corecell Gateway Kit.

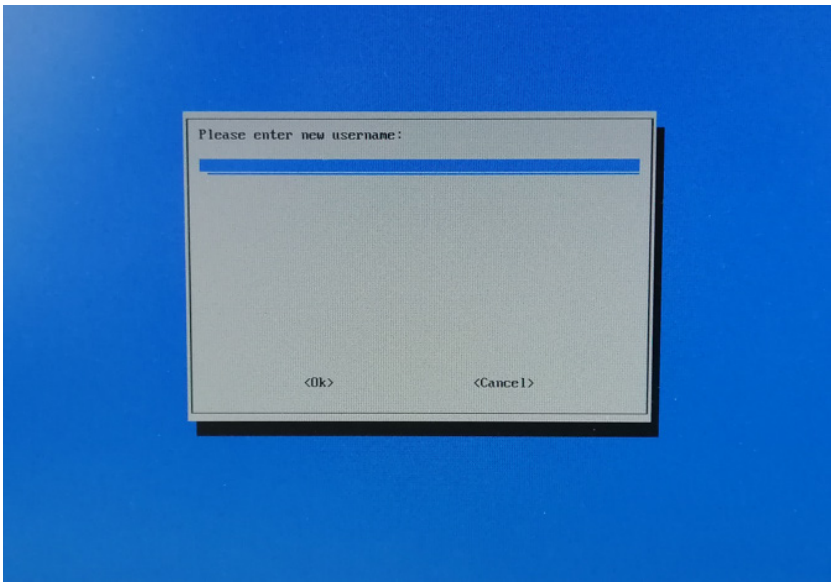


Figure 11: enable SSH connection on R-Pi

Check the IP address with command: *ifconfig*

Then you connect to R-Pi with MobaXterm SSH client. Free MobaXterm Home Edition here: https://mobaxterm.mobatek.net/

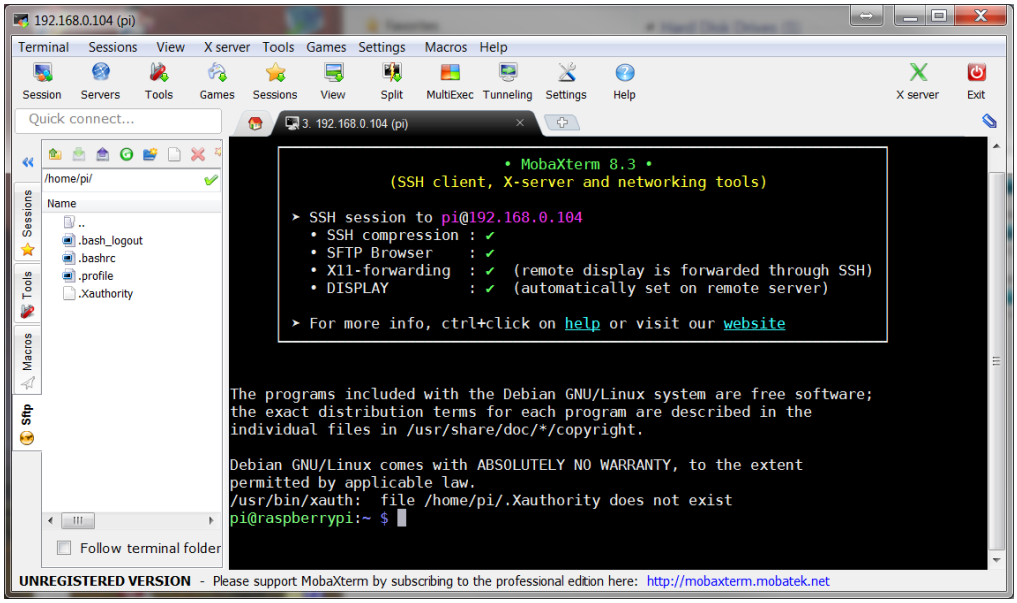


Figure 12: MobaXterm SSH Client with IP address

1.1.3 Resize Partition / FS

On larger microSD cards, the root partition can be resized to use extra space, using the *Expand Filesystem* option from *raspi-config* menu:

Command: *$ sudo raspi-config*

Select: *Advanced Options*

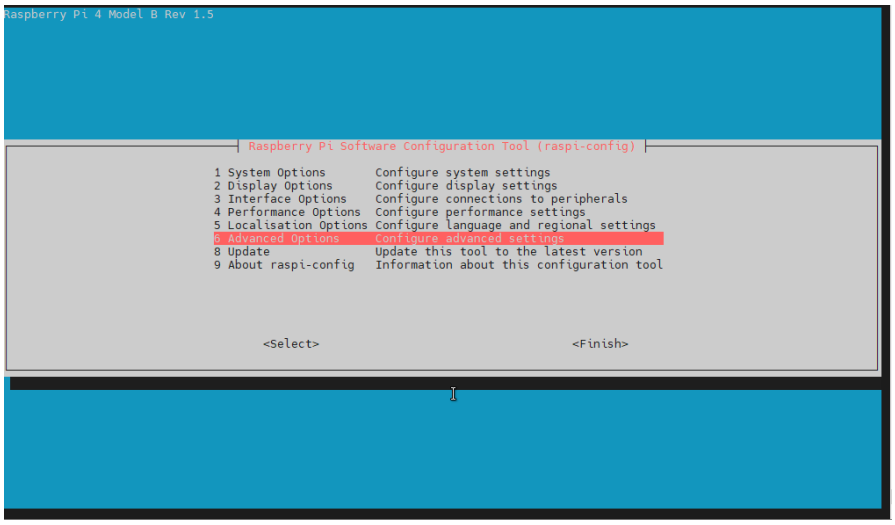


Figure 13: MobaXterm SSH Client

Select **1***: Expand Filesystem* from raspi-config menu and press *Enter*

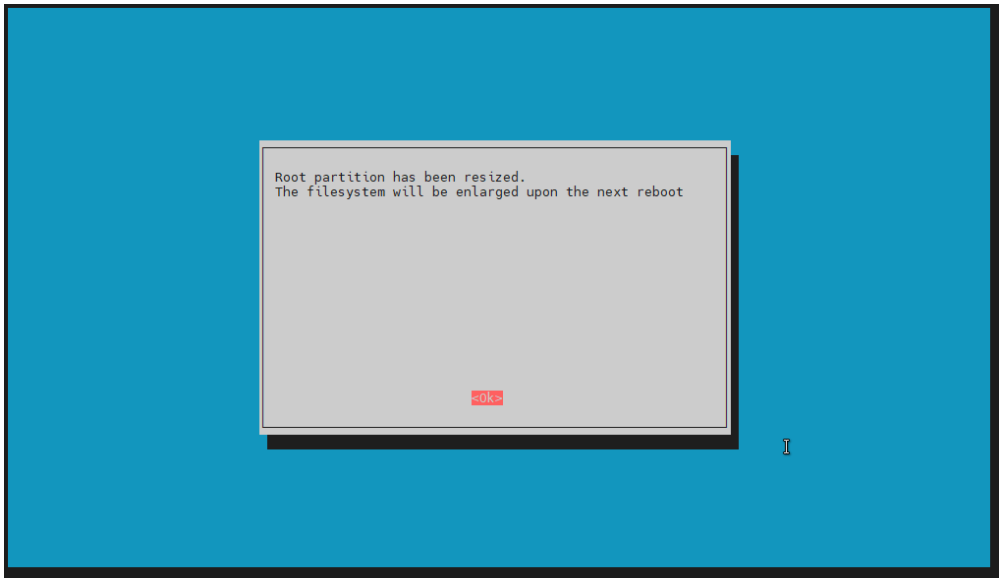


Figure 14: raspi-config "Expand Filesystem"

The R-Pi must be rebooted at the end of the procedure: *$ sudo reboot*

4.3.3 Update & Configure Raspberry Pi

1. **Update** - enter the following commands:

   $ sudo apt-get update

   $ sudo apt-get upgrade

   $ sudo apt-get dist-upgrade

   $ sudo rpi-update

2. **Install Git** - enter the following command: *$ sudo apt install git*

3. **Enable SPI/I2C/UART** - enter the following command: *$ sudo raspi-config*

   Interfacing options: SPI / I2C / Serial

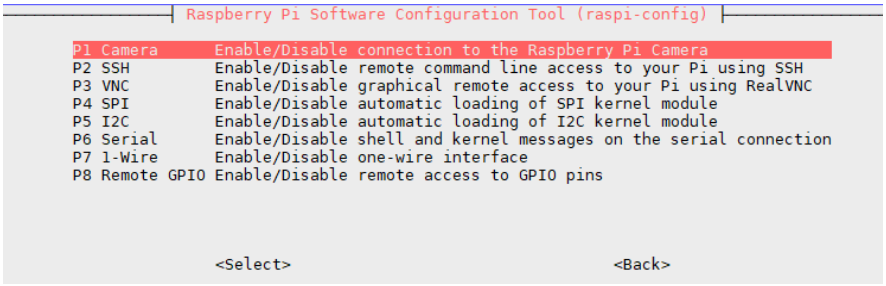


Figure 15: Enable SPI/I2C/UART

4. **Reboot** the system at the end of the procedure: *$ sudo reboot*

4.3.4 Compile Semtech HAL & Packet Forwarder

1. Get the latest Semtech software package from LoRa GitHub (requires an internet connection):

   $ git clone https://github.com/Lora-net/sx1302_hal.git

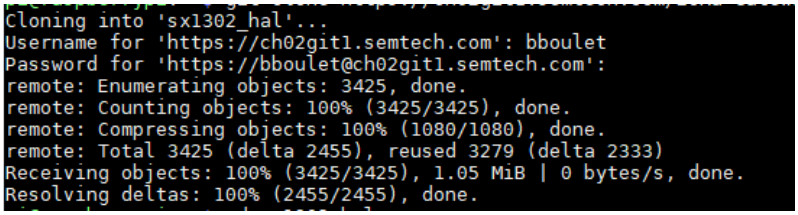


Figure 16: Git Clone

   $ cd ~/sx1302_hal/

   $ make clean all

2. Execute the next two commands to avoid entering the user password when installing the files (recommended but optional):

   $ ssh-keygen -t rsa → press enter without type anything at each request

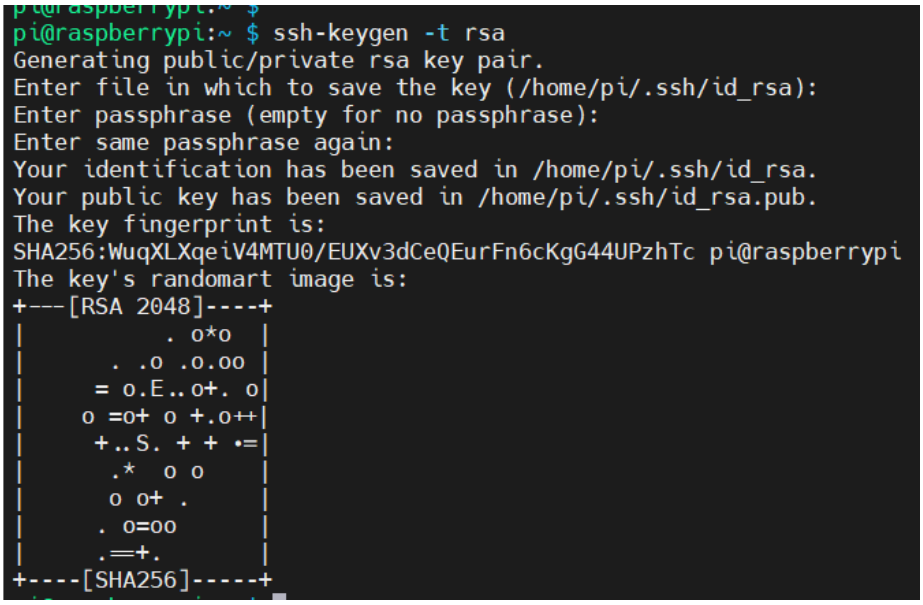


Figure 17: Generate rsa Key

   $ ssh-copy-id -i ~/.ssh/id_rsa.pub pi@localhost

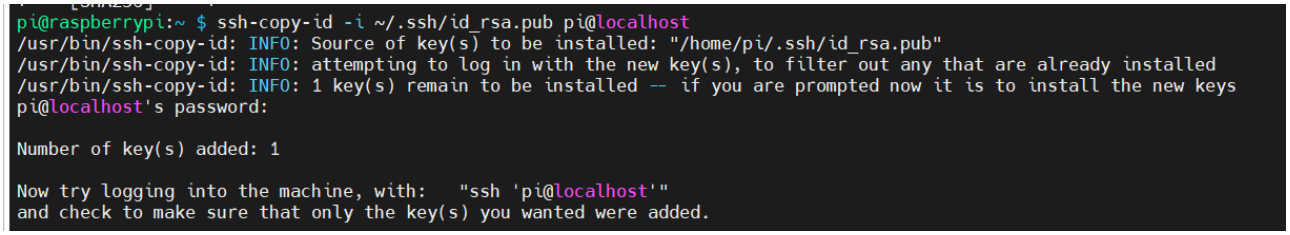


Figure 18: Copy rsa Key

3. Now install the package:

   $ make install

   $ make install_conf

4. The executables are now in the *bin* folder (which was set in TARGET_DIR in the *target.cfg* file).

```
pi@raspberrypi:~/corecellV3_testing/sx1302_hal-develop $ cd bin/
pi@raspberrypi:~/corecellV3_testing/sx1302_hal-develop/bin $ ls
boot                                    spectral_scan
chip_id                                 test_loragw_cal_sx125x
global_conf.json.sx1250.AS923.USB       test_loragw_capture_ram
global_conf.json.sx1250.CN490           test_loragw_com
global_conf.json.sx1250.CN490.USB       test_loragw_com_sx1250
global_conf.json.sx1250.EU868           test_loragw_com_sx1261
global_conf.json.sx1250.EU868.USB       test_loragw_counter
global_conf.json.sx1250.US915           test_loragw_gps
global_conf.json.sx1250.US915.USB       test_loragw_hal_rx
global_conf.json.sx1255.CN490.full-duplex  test_loragw_hal_tx
global_conf.json.sx1257.EU868           test_loragw_i2c
lora_pkt_fwd                            test_loragw_reg
net_downlink                            test_loragw_sx1261_rssi
reset_lgw.sh                            test_loragw_toa
pi@raspberrypi:~/corecellV3_testing/sx1302_hal-develop/bin $
```

Figure 19: Executables

4.3.5 Update STM32 MCU Firmware

When running the Corecell in USB mode, the STM32 MCU serves as a translator, it converts the SPI data it receives from SX1302 and sends it over USB to Raspberry Pi and vice versa.

The STM32 MCU firmware can be updated as follows (by default the MCU is already programmed in the case of a Semtech Corecell order):

1. Navigate to the 'bin' directory where all the executables are:
   $ cd ~/sx1302_hal/bin
2. Issue following command:
   ./boot -d /dev/ttyACM0 ➔ by default it's ACM0 but it could be something else if several USB ports are used.
3. Download and install the DFU tool on the Raspberry Pi using the following command:
   sudo apt-get install dfu-util
4. Flash the firmware binary in the *mcu_bin* directory on the STM32 MCU using the following command:

   sudo dfu-util -a 0 -s 0x08000000:leave -t 0 -D

   ../mcu_bin/rlz_010000_CoreCell_USB.bin
5. At this point the Corecell is ready to run.

4.3.6 Get Gateway's Unique ID

The Corecell reference design has a unique ID assigned at the time of production. This ID can be used as a 64-bit MAC address for the Corecell reference design.

To get the ID:

$ cd ~/sx1302_hal/bin

For a Corecell running in USB configuration:

./chip_id -u -d /dev/ttyACM0

For a Corecell running in SPI configuration:

./chip_id -d /dev/spidev0.0

Returns an Extended Unique ID (EUI) like the following:

```
pi@raspberrypi:~/corecellV3_testing/sx1302_hal-develop $ ./chip_id -u -d /dev/ttyACM0
Opening USB communication interface
INFO: Configuring TTY
INFO: Flushing TTY
INFO: Setting TTY in blocking mode
INFO: Connect to MCU
INFO: Concentrator MCU version is V00.02.06
INFO: MCU status: sys_time:8833838 temperature:28.2oC
Note: chip version is 0x12 (v1.2)
INFO: using legacy timestamp
ARB: dual demodulation disabled for all SF

INFO: concentrator EUI: 0x0016c001ff193579

Closing USB communication interface
```

Figure 20: Util Chip ID

The gateway ID should then be replaced (to have a unique ID) in *~/sx1302_hal/bin/global_conf.json.sx1250*

as seen in

```
"gateway_conf": {
    "gateway_ID": "AA555A0000000000",
    /* change with default server address/ports */
    "server_address": "localhost",
    "serv_port_up": 1730,
    "serv_port_down": 1730,
    /* adjust the following parameters for your network */
    "keepalive_interval": 10,
```

4.3.7 Semtech HAL Compilation Check

The programs test_loragw_com_sx1250 and test_loragw_com_sx1261 check the reliability of the link between the host platform (on which the program is run) and the LoRa concentrator register file (that is the interface through which all interactions with the LoRa concentrator happen). The tests run endlessly or until an error is detected.

To stop the programs: press **Ctrl+C**.

To start the programs:

$ cd ~/sx1302_hal/bin

For USB configuration:

$./test_loragw_com_sx1261 -u -d /dev/ttyACM0

$./test_loragw_com_sx1250 -u -d /dev/ttyACM0

For SPI configuration:

$./test_loragw_com_sx1261

$./test_loragw_com_sx1250

The output looks like this:

```
pi@raspberrypi:~/corecellV3_testing/sx1302_hal-develop $ ./test_loragw_com_sx1250 -u -d /dev/
ttyACM0
Opening USB communication interface
INFO: Configuring TTY
INFO: Flushing TTY
INFO: Setting TTY in blocking mode
INFO: Connect to MCU
INFO: Concentrator MCU version is V00.02.06
INFO: MCU status: sys_time:19424195 temperature:27.1oC
Note: chip version is 0x12 (v1.2)
Radio0: get_status: 0x32
Radio1: get_status: 0x32
Cycle 0 > did a 4-byte R/W on a register with no error
Cycle 1 > did a 4-byte R/W on a register with no error
Cycle 2 > did a 4-byte R/W on a register with no error
Cycle 3 > did a 4-byte R/W on a register with no error
Cycle 4 > did a 4-byte R/W on a register with no error
Cycle 5 > did a 4-byte R/W on a register with no error
Cycle 6 > did a 4-byte R/W on a register with no error
Cycle 7 > did a 4-byte R/W on a register with no error
```

Figure 21: HAL Compilation Check Results

4.3.8 Test HAL TX

The program test_loragw_hal_tx tests the emission of the Corecell reference design while running in SPI or USB configuration, using settings specified by the start commands (described below). The tests run endlessly or until an error is detected.

To stop the program:

press Ctrl+C

To start the program:

$ cd ~/sx1302_hal/bin

Example for USB configuration (be careful if you copy/paste the command from window to Linux, the "-" can be modified during the paste):

$ ./test_loragw_hal_tx –u –d /dev/ttyACM0 -k0 -c0 -r1250 -f868.1 -mLORA --pa 1 -l12 --pwid 14 -s7 -b125 -z16 -n10000 -t 100

Example for SPI configuration:

$ ./test_loragw_hal_tx -k0 -c0 -r1250 -f868.1 -mLORA --pa 1 -l12 --pwid 14 -s7 -b125 -z16 -n10000 -t 100

The commands above send a LoRa frame at 868.1MHz (-m) with the front-end module enabled (--pa) and the pwid from the SX1250 set to 14dBm (--pwid).

The following command sends a TX CW signal at 868.1MHz (-m) with the front-end module enabled (--pa) and the pwid from the SX1250 set to 14dBm (--pwid).

./test_loragw_hal_tx –u –d /dev/ttyACMx -k0 -c0 -r1250 -f868.1 -mCW - -pa 1 -l12 --pwid 14

For more information, enter:

$ ./test_loragw_hal_tx -h

4.3.9 Configure 'Listen Before Talk'

The Corecell supports the 'Listen before Talk' (LBT) feature only for 125kHz and 250kHz bandwidths. If this feature is to be used while the packet forwarder is running, then it can be enabled and disabled from the *global_conf.json.sx1250.xxxx* file by setting the following parameters to true\false:

'SX130x_conf.sx1261_cfg.lbt_cfg.enable' = true\false

```
{
    "SX130x_conf": {
        "com_type": "USB",
        "com_path": "/dev/ttyACM0",
        "lorawan_public": true,
        "clksrc": 0,
        "antenna_gain": 0, /* antenna gain, in dBi */
        "full_duplex": false,
        "fine_timestamp": {
            "enable": false,
            "mode": "all_sf" /* high_capacity or all_sf */
        },
        "sx1261_cfg": {
            "enable": true,
            "rssi_offset": 0, /* dB */
            "lbt_cfg": {
                "enable": true,
                "rssi_target": -70, /* dBm */
                "chan_cfg":[ /* 16 channels maximum */
```

Figure 22: Listen Before Talk Configuration While Running Packet Forwarder

4.3.10 Configure Spectral Scan

The Corecell supports the spectral scan feature only if no other tool is running. (In the future, it will be able to run at the same time as the packet forwarder by adding a dedicated thread for it, and configuring the *global_conf.json.sx1250.xxxx* file to have SX1261 radio enabled.)

To enable\disable the spectral scan feature, set the following parameters to true\false:

'SX130x_conf.sx1261_cfg' = true\false

```
{
    "SX130x_conf": {
        "com_type": "USB",
        "com_path": "/dev/ttyACM0",
        "lorawan_public": true,
        "clksrc": 0,
        "antenna_gain": 0, /* antenna gain, in dBi */
        "full_duplex": false,
        "fine_timestamp": {
            "enable": false,
            "mode": "all_sf" /* high_capacity or all_sf */
        },
        "sx1261_cfg": {
            "enable": false,
            "rssi_offset": 0, /* dB */
```

Figure 23: Spectral Scan Configuration While Running Packet Forwarder (Disabled)

4.3.11 Run Spectral Scan

The Corecell supports a spectral scan feature via the standalone utility util_spectral_scan.

Note: This cannot be run while any other tool is running.

To start the utility: *$ cd ~/sx1302_hal/bin/*

For Corecell with USB configuration: *$./spectral_scan –u –d /dev/ttyACMx*

For Corecell with SPI configuration: *$./spectral_scan –d /dev/spidev0.0*

The output of the spectral scan is a csv file: *rssi_histogram.csv*

A visual representation of the spectral scan results can be obtained by using the python script in the *util_spectral_scan* directory. The *plot_rssi_histogram.py* script requires that the *numpy* and *pylab* packages are installed before use.

$ cd ~/ util_spectral_scan

$ python3 plot_rssi_histogram.py rssi_histogram.csv

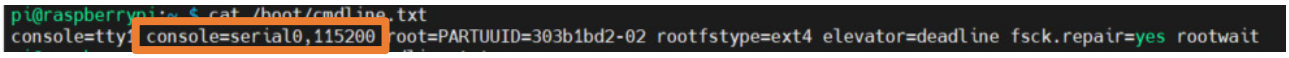
The output of the scan result looks like this:

```
pi@raspberrypi:~/corecellV3_testing/sx1302_hal-develop $ ./spectral_scan -u -d /dev/ttyACM0
==
== Spectral Scan: freq_hz=863100000Hz, nb_channels=35, nb_scan=2000, rssi_offset=-11dB
==
Opening USB communication interface
INFO: Configuring TTY
INFO: Flushing TTY
INFO: Setting TTY in blocking mode
INFO: Connect to MCU
INFO: Concentrator MCU version is V00.02.06
INFO: MCU status: sys_time:17694872 temperature:26.8oC
Note: chip version is 0x12 (v1.2)
INFO: using legacy timestamp
ARB: dual demodulation disabled for all SF
SX1261: PRAM version: SX1261 V2D 2D02
SX1261: PRAM version: SX1261 V2D 2D06
863100000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 766 1234 0 0 0 0
863300000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 623 1377 0 0 0 0
863500000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 562 1438 0 0 0 0
863700000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 486 1514 0 0 0 0
863900000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2000 0 0 0 0 0 0
864100000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2000 0 0 0 0 0 0
864300000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 903 1097 0 0 0 0
864500000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 660 1340 0 0 0 0
864700000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 657 1343 0 0 0 0
864900000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 895 1105 0 0 0 0
865100000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 659 1341 0 0 0 0
865300000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 639 1361 0 0 0 0
865500000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 502 1498 0 0 0 0
865700000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 794 1206 0 0 0 0
865900000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 452 1530 18 0 0 0
866100000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 682 1318 0 0 0 0
866300000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 666 1334 0 0 0 0
866500000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 613 1387 0 0 0 0
866700000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 538 1462 0 0 0 0
866900000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 638 1362 0 0 0 0
867100000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 884 1116 0 0 0 0
867300000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 719 1264 17 0 0 0
867500000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 640 1360 0 0 0 0
867700000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 626 1374 0 0 0 0
867900000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 811 1189 0 0 0 0
868100000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 650 1350 0 0 0 0
868300000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 623 1377 0 0 0 0
868500000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 17 1948 35 0 0 0
868700000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 573 1427 0 0 0 0
868900000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 674 1326 0 0 0 0
869100000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 957 1043 0 0 0 0
869300000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1238 762 0 0 0 0
869500000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 874 1126 0 0 0 0
869700000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1041 959 0 0 0 0
869900000: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1065 935 0 0 0 0
Closing USB communication interface
pi@raspberrypi:~/corecellV3_testing/sx1302_hal-develop $
```

Figure 24: Spectral Scan Results

4.3.12 Activate GPS: Fine Timestamp & Class B Beacon

If you have a GPS module, connect it to the socket on the interface board, then:

1. **Enable** the serial port on the Raspberry using: *"sudo raspi-config", Interface Options, Serial Port, Yes* .
2. **Reboot** the Raspberry.
3. **Modify** */boot/cmdline.tx* file: nano /boot/cmdline.tx ==> ***remove*** *"console=serial0,115200"*
4. 

5. **Reboot** the Raspberry.
6. Check if you received NMEA traces (the GPS antenna must be outdoor) ==> *cat /dev/ttyS0*

To activate the class B beacon, open *global_conf.json* and modify the Beaconing parameters fields:

```
/* Beaconing parameters */
"beacon_period": 128,
"beacon_freq_hz": 869525000,
"beacon_datarate": 9,
"beacon_bw_hz": 125000,
"beacon_power": 14,
"beacon_infodesc": 0
```

To activate the Fine Timestamp (only on SX1303) open *global_conf.json* and modify the fine_timestamp fields:

```
"fine_timestamp": {
        "enable": true,
        "mode": "all_sf" /* high_capacity or all_sf */
    },
```

4.3.13 Run Packet Forwarder

The Packet Forwarder program runs on the host of a LoRa Gateway. It forwards RF packets received by the concentrator to a server, through an IP/UDP link, and emits RF packets that are sent by the server.
The Corecell GW supports SPI and USB modes to communicate with SX1302.

Run Packet Forwarder for a functional check:

$ cd ~/sx1302_hal/bin/

USB mode:

$ ./lora_pkt_fwd –c global_conf.json.sx1250.USB

SPI mode:

$ cd ~/sx1302_hal/bin/

$ ./lora_pkt_fwd –c global_conf.json.sx1250

The output looks like this:

```
pi@raspberrypi:~/sx1302/sx1302_hal/packet_forwarder $ ./lora_pkt_fwd -c global_conf.json.sx1250
*** Packet Forwarder ***
Version: 1.0.0
*** SX1302 HAL library version info ***
Version: 1.0.0;
***
INFO: Little endian host
INFO: found configuration file global_conf.json.sx1250, parsing it
INFO: global_conf.json.sx1250 does contain a JSON object named SX130x_conf, parsing SX1302 parameters
INFO: spidev_path /dev/spidev0.0, lorawan_public 1, clksrc 0, full_duplex 0
INFO: antenna_gain 0 dBi
INFO: Configuring legacy timestamp
INFO: Configuring Tx Gain LUT for rf_chain 0 with 16 indexes for sx1250
INFO: radio 0 enabled (type SX1250), center frequency 867500000, RSSI offset -215.399994, tx enabled 1
INFO: radio 1 enabled (type SX1250), center frequency 868500000, RSSI offset -215.399994, tx enabled 0
INFO: Lora multi-SF channel 0>  radio 1, IF -400000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 1>  radio 1, IF -200000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 2>  radio 1, IF 0 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 3>  radio 0, IF -400000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 4>  radio 0, IF -200000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 5>  radio 0, IF 0 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 6>  radio 0, IF 200000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora multi-SF channel 7>  radio 0, IF 400000 Hz, 125 kHz bw, SF 5 to 12
INFO: Lora std channel> radio 1, IF -200000 Hz, 250000 Hz bw, SF 7, Explicit header
INFO: FSK channel> radio 1, IF 300000 Hz, 125000 Hz bw, 50000 bps datarate
INFO: global_conf.json.sx1250 does contain a JSON object named gateway_conf, parsing gateway parameters
INFO: gateway MAC address is configured to AA555A0000000000
INFO: server hostname or IP address is configured to "localhost"
INFO: upstream port is configured to "1730"
INFO: downstream port is configured to "1730"
INFO: downstream keep-alive interval is configured to 10 seconds
INFO: statistics display interval is configured to 30 seconds
INFO: upstream PUSH_DATA time-out is configured to 100 ms
INFO: packets received with a valid CRC will be forwarded
INFO: packets received with a CRC error will NOT be forwarded
INFO: packets received with no CRC will NOT be forwarded
INFO: GPS serial port path is configured to "/dev/ttyS0"
INFO: Reference latitude is configured to 0.000000 deg
INFO: Reference longitude is configured to 0.000000 deg
INFO: Reference altitude is configured to 0 meters
INFO: Beaconing period is configured to 128 seconds
INFO: Beaconing signal will be emitted at 869525000 Hz
INFO: Beaconing datarate is set to SF9
INFO: Beaconing modulation bandwidth is set to 125000Hz
INFO: Beaconing TX power is set to 14dBm
INFO: Beaconing information descriptor is set to 0
INFO: global_conf.json.sx1250 does contain a JSON object named debug_conf, parsing debug parameters
INFO: got 2 debug reference payload
INFO: reference payload ID 0 is 0xCAFE1234
INFO: reference payload ID 1 is 0xCAFE2345
INFO: setting debug log file name to loragw_hal.log
INFO: [main] TTY port /dev/ttyS0 open for GPS synchronization
Accessing CoreCellSX1302 reset pin through GPIO23...
Accessing CoreCellSX1302 power enable pin through GPIO18...
INFO: [main] concentrator started, packet can now be received
INFO: concentrator EUI: 0x0016C00100001419

INFO: Received pkt from mote: 260114D1 (fcnt=57252)
```

Figure 25: Packet Forwarder

5 RF Parameters Tuning

You can edit the following RF parameters in the file *~/sx1302_hal/bin/global_conf.json.sx1250*:

- freq, radio, and if: to set frequency channels (frequency channels = [*freq* of selected *radio* + *if*] in Hz)
- rssi_offset: to tune SX1250 + SX1302 RSSI
- 16 gain tables (tx_lut_12 to tx_lut_27): to tune Tx output power using these parameters:
    - pa_gain[0 1]: PA Enable Corecell reference design V1.3, 0 = PA bypassed, 1 = PA ON
    - pwr_idx[0 22]: possible gain settings from 0 (min. gain) to 22 (max. gain)
    - rf_power*:* RF output power target in dBm

  Within a Tx gain table index, the setting {pa_gain, pwr_idx} must correspond to the RF output power target defined in the parameter rf_power.

5.1 Spreading Factors SF5 & SF6

The SX1302 supports SF5 and SF6 spreading factors, and the HAL also. But it is important to note that the only syncword supported for SF5 and SF6 is 0x12 (also known as "private").

This is true whatever the setting of the lorawan_public field of lgw_conf_board_s.

5.2 Typical JSON File

A typical LoRa Corecell reference design *global_conf.json* file looks like the following:

```
{
    "SX130x_conf": {
        "com_type": "USB",
        "com_path": "/dev/ttyACM0",
        "lorawan_public": true,
        "clksrc": 0,
        "antenna_gain": 0, /* antenna gain, in dBi */
        "full_duplex": false,
        "fine_timestamp": {
            "enable": false,
            "mode": "all_sf" /* high_capacity or all_sf */
        },
        "sx1261_cfg": {
            "enable": false,
            "rssi_offset": 0, /* dB */
            "lbt_cfg": {
                "enable": false,
                "rssi_target": -70, /* dBm */
                "chan_cfg":[ /* 16 channels maximum */
                    { "freq_hz": 867100000, "bandwidth": 125000, "scan_time_us": 128,  "transmit_time_ms": 400 },
                    { "freq_hz": 867300000, "bandwidth": 125000, "scan_time_us": 128,  "transmit_time_ms": 400 },
                    { "freq_hz": 867500000, "bandwidth": 125000, "scan_time_us": 128,  "transmit_time_ms": 400 },
                    { "freq_hz": 867700000, "bandwidth": 125000, "scan_time_us": 128,  "transmit_time_ms": 400 },
                    { "freq_hz": 867900000, "bandwidth": 125000, "scan_time_us": 128,  "transmit_time_ms": 400 },
                    { "freq_hz": 868100000, "bandwidth": 125000, "scan_time_us": 128,  "transmit_time_ms": 400 },
                    { "freq_hz": 868300000, "bandwidth": 125000, "scan_time_us": 128,  "transmit_time_ms": 400 },
                    { "freq_hz": 868500000, "bandwidth": 125000, "scan_time_us": 128,  "transmit_time_ms": 400 },
                    { "freq_hz": 869525000, "bandwidth": 125000, "scan_time_us": 5000, "transmit_time_ms": 4000 },
                    { "freq_hz": 868300000, "bandwidth": 250000, "scan_time_us": 128,  "transmit_time_ms": 400 }
                ]
            }
        },
```

```
"radio_0": {
    "enable": true,
    "type": "SX1250",
    "freq": 867500000,
    "rssi_offset": -215.4,
    "rssi_tcomp": {"coeff_a": 0, "coeff_b": 0, "coeff_c": 20.41, "coeff_d": 2162.56, "coeff_e": 0},
    "tx_enable": true,
    "tx_freq_min": 863000000,
    "tx_freq_max": 870000000,
    "tx_gain_lut":[
        {"rf_power": 12, "pa_gain": 0, "pwr_idx": 15},
        {"rf_power": 13, "pa_gain": 0, "pwr_idx": 16},
        {"rf_power": 14, "pa_gain": 0, "pwr_idx": 17},
        {"rf_power": 15, "pa_gain": 0, "pwr_idx": 19},
        {"rf_power": 16, "pa_gain": 0, "pwr_idx": 20},
        {"rf_power": 17, "pa_gain": 0, "pwr_idx": 22},
        {"rf_power": 18, "pa_gain": 1, "pwr_idx": 1},
        {"rf_power": 19, "pa_gain": 1, "pwr_idx": 2},
        {"rf_power": 20, "pa_gain": 1, "pwr_idx": 3},
        {"rf_power": 21, "pa_gain": 1, "pwr_idx": 4},
        {"rf_power": 22, "pa_gain": 1, "pwr_idx": 5},
        {"rf_power": 23, "pa_gain": 1, "pwr_idx": 6},
        {"rf_power": 24, "pa_gain": 1, "pwr_idx": 7},
        {"rf_power": 25, "pa_gain": 1, "pwr_idx": 9},
        {"rf_power": 26, "pa_gain": 1, "pwr_idx": 11},
        {"rf_power": 27, "pa_gain": 1, "pwr_idx": 14}
    ]
},
"radio_1": {
    "enable": true,
    "type": "SX1250",
    "freq": 868500000,
```

```
        "rssi_offset": -215.4,
        "rssi_tcomp": {"coeff_a": 0, "coeff_b": 0, "coeff_c": 20.41, "coeff_d": 2162.56, "coeff_e": 0},
        "tx_enable": false
    },
    "chan_multiSF_All": {"spreading_factor_enable": [ 5, 6, 7, 8, 9, 10, 11, 12 ]},
    "chan_multiSF_0": {"enable": true, "radio": 1, "if": -400000},
    "chan_multiSF_1": {"enable": true, "radio": 1, "if": -200000},
    "chan_multiSF_2": {"enable": true, "radio": 1, "if":  0},
    "chan_multiSF_3": {"enable": true, "radio": 0, "if": -400000},
    "chan_multiSF_4": {"enable": true, "radio": 0, "if": -200000},
    "chan_multiSF_5": {"enable": true, "radio": 0, "if":  0},
    "chan_multiSF_6": {"enable": true, "radio": 0, "if":  200000},
    "chan_multiSF_7": {"enable": true, "radio": 0, "if":  400000},
    "chan_Lora_std":  {"enable": true, "radio": 1, "if": -200000, "bandwidth": 250000, "spread_factor": 7,
                       "implicit_hdr": false, "implicit_payload_length": 17, "implicit_crc_en": false, "implicit_coderate": 1},
    "chan_FSK":       {"enable": true, "radio": 1, "if":  300000, "bandwidth": 125000, "datarate": 50000}
},

"gateway_conf": {
    "gateway_ID": "AA555A0000000000",
    /* change with default server address/ports */
    "server_address": "localhost",
    "serv_port_up": 1730,
    "serv_port_down": 1730,
    /* adjust the following parameters for your network */
    "keepalive_interval": 10,
    "stat_interval": 30,
    "push_timeout_ms": 100,
    /* forward only valid packets */
    "forward_crc_valid": true,
    "forward_crc_error": false,
    "forward_crc_disabled": false,
```

```
    /* GPS configuration */
    "gps_tty_path": "/dev/ttyS0",
    /* GPS reference coordinates */
    "ref_latitude": 0.0,
    "ref_longitude": 0.0,
    "ref_altitude": 0,
    /* Beaconing parameters */
    "beacon_period": 0,
    "beacon_freq_hz": 869525000,
    "beacon_datarate": 9,
    "beacon_bw_hz": 125000,
    "beacon_power": 14,
    "beacon_infodesc": 0
  },

  "debug_conf": {
    "ref_payload":[
      {"id": "0xCAFE1234"},
      {"id": "0xCAFE2345"}
    ],
    "log_file": "loragw_hal.log"
  }
}
```

6 References

[1] SX1302 information: https://www.semtech.com/products/wireless-rf/lora-gateways/sx1302

[2] SX1250 information: https://www.semtech.com/products/wireless-rf/lora-gateways/sx1250

[3] SX1302/03 Corecell GW information: https://www.semtech.com/products/wireless-rf/lora-core/sx1302cssxxxgw1

[4] SX1302_hal Github: https://github.com/Lora-net/sx1302_hal

[5] Test LoRa Github repository: https://github.com/Lora-net/sx1302_hal/libloragw

7 Glossary

BB	BaseBand
BoM	Bill Of Materials
BW	BandWidth
CLK	Clock
CW	Continuous Wave
ETSI	European Telecommunications Standard Institute
DFU	Device Firmware Update
EU	Europe
EUI	Extended Unique Identifier
GB	GigaByte
GPS	Global Positioning System
GW	GateWay
HAL	Hardware Abstraction Layer
HDMI	High-Definition Multimedia Interface
HW	HardWare
IP	Intellectual Property
ISM	Industrial, Scientific and Medical applications
LAN	Local Area Network
LBT	Listen Before Talk
LO	Local Oscillator
LoRa®	Long Range modulation technique
LoRaWAN®	LoRa® low power Wide Area Network standard
LPF	Low Pass Filter
LSB	Least Significant Bit
LUT	Look Up Table
MAC	Media Access Control address
MCU	Micro-Controller Unit
MPU	Micro-Processing Unit
PA	Power Amplifier
RSSI	Received Signal Strength Indication
RF	Radio-Frequency
RX	Receiver
R-Pi	Raspberry Pi
SAW	Surface Acoustic Wave filter
SD Card	Secure Digital Card
SF	Spreading Factor
SPI	Serial Peripheral Interface
SPDT	Single-Pole, Double-Throw switch
SSH	Secure SHell
SW	SoftWare
TX	Transmitter
UART	Universal Asynchronous Receiver/Transmitter
UDP	User Datagram Protocol
USB	Universal Serial Bus

8 Revision History

Version	Date	Modifications
1.0	July 2019	First Release
1.1	September 2019	Minor corrections
1.2	November 2019	Added CN490 version
1.3	May 2024	Added supporting documentation for Spectrum scanner, LBT feature, and USB support.

Important Notice

Information relating to this product and the application or design described herein is believed to be reliable, however such information is provided as a guide only and Semtech assumes no liability for any errors in this document, or for the application or design described herein.

Semtech reserves the right to make changes to the product or this document at any time without notice. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. Semtech warrants performance of its products to the specifications applicable at the time of sale, and all sales are made in accordance with Semtech's standard terms and conditions of sale.

SEMTECH PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS, OR IN NUCLEAR APPLICATIONS IN WHICH THE FAILURE COULD BE REASONABLY EXPECTED TO RESULT IN PERSONAL INJURY, LOSS OF LIFE OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. INCLUSION OF SEMTECH PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE UNDERTAKEN SOLELY AT THE CUSTOMER'S OWN RISK.

Should a customer purchase or use Semtech products for any such unauthorized application, the customer shall indemnify and hold Semtech and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs damages and attorney fees which could arise.

The Semtech name and logo are registered trademarks of the Semtech Corporation. All other trademarks and trade names mentioned may be marks and names of Semtech or their respective companies. Semtech reserves the right to make changes to, or discontinue any products described in this document without further notice. Semtech makes no warranty, representation or guarantee, express or implied, regarding the suitability of its products for any particular purpose.