LoRa® Corecell reference design for LBT Spectral Scan gateway / USB version User Guide

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose Of This Manual

This user guide introduces the Semtech LoRa® Corecell reference design (V3 for EU868 & US915 & AS923, and V3 for CN490) and how to set it up with a Raspberry Pi 3\4.

1.2 Scope Of This Manual

1.2.1 EU868 / US915 / AS923

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

The LoRa gateway supports USB and SPI interfaces to connect to a Raspberry Pi host computer. The gateway also supports 'Listen before Talk' (LBT) function which is a requirement in countries like Japan and Korea. It also has a provision to run a spectral scan.

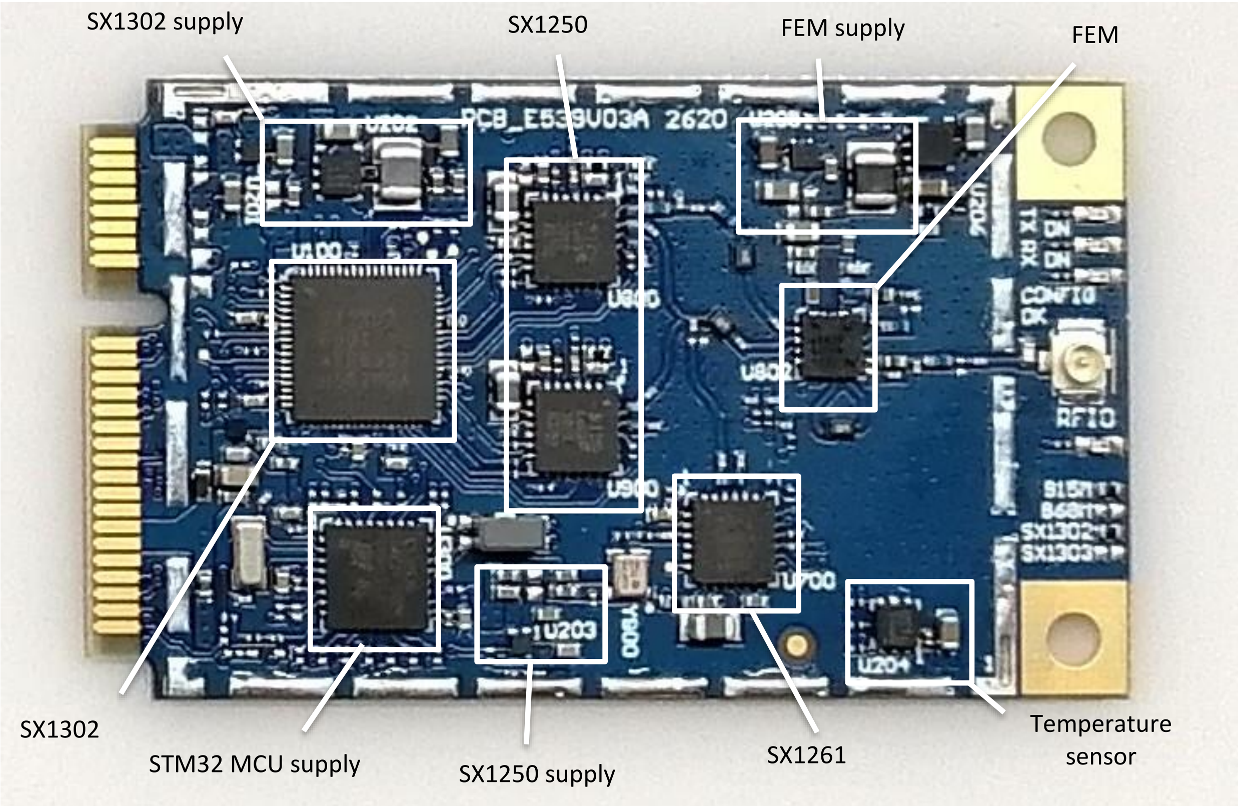

Figure 1: LoRa® Corecell reference design V3 (EU868/US915/AS923)

1.2.2 CN490

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

It also supports the ‘Listen before Talk’ (LBT) feature and is capable of running a spectral scan.

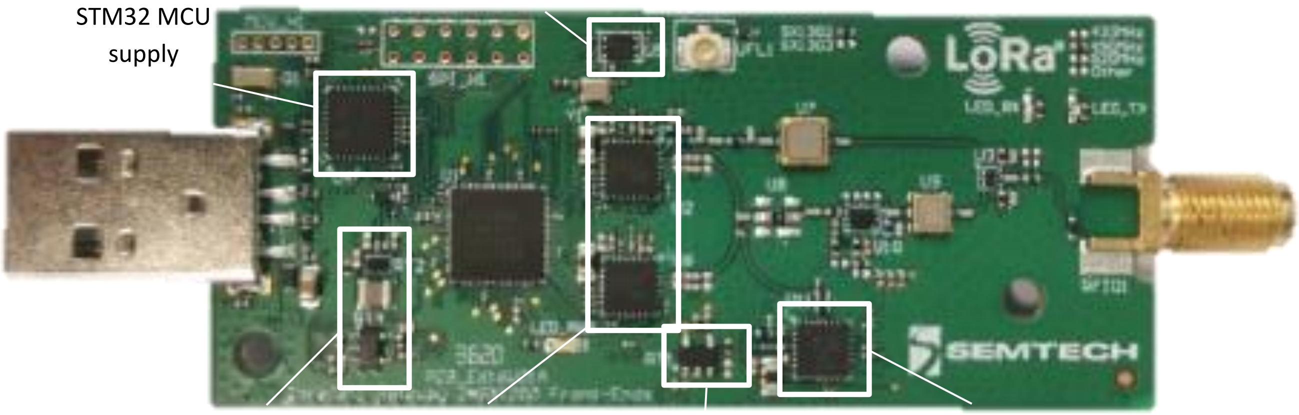

Figure 2: LoRa® Corecell reference design V3 (CN490)

2 Hardware Overview

2.1 Absolute Maximum Ratings

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

Table 1: Absolute Maximum Ratings

2.2 RF Front-End Architecture

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

- Half-duplex mode i.e. can't receive and transmit simultaneously
- Simultaneously receive 8 LoRa® channels multi-data rates (SF5 ~ SF12 / 125 kHz) + 2 mono-data rate (LoRa® 250 / 500 kHz and FSK 50 kbps)
- Maximum transmit output power (EU868/US915) = +27dBm
- Maximum transmit output power (AS923) = +10dBm
- Maximum transmit output power (CN490) = +17dBm

- Typical sensitivity level (EU868/US915):
  - -141 dBm at SF12 BW 125 kHz
  - **-127 dBm** at SF7 BW 125 kHz
  - -111 dBm at FSK 50 kbps
- Typical sensitivity level (AS923) :
  - -141 dBm at SF12 BW 125kHz
  - **-126 dBm at SF7 BW 125kHz**
  - -109 dBm at FSK 50kbps
- Typical sensitivity level (CN490):
  - -141 dBm at SF12 BW 125 kHz
  - **-126 dBm** at SF7 BW 125 kHz
  - -109 dBm at FSK 50 kbps
- Ability to work in hostile RF environments such as close to cellular mobile phones, WiFi routers, Bluetooth devices

2.3 Corecell Reference Design Block Diagram

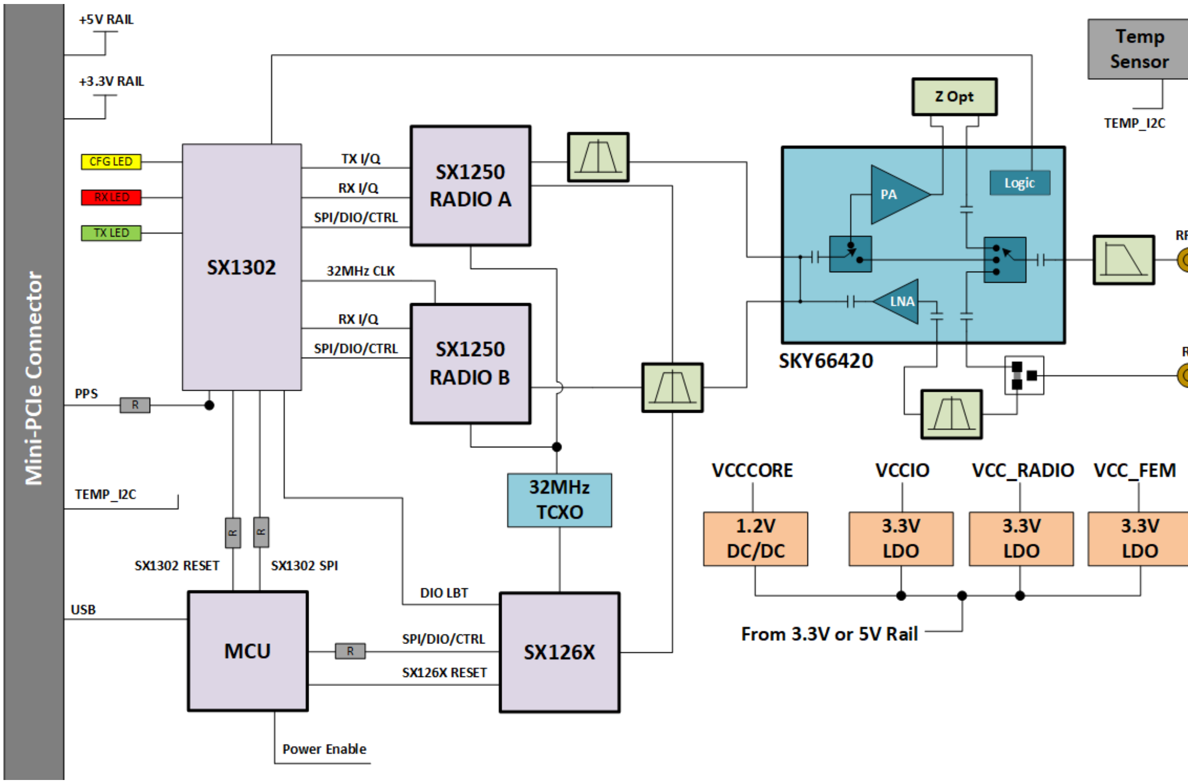

Figure 3: Corecell reference design V3 Block Diagram (EU868/US915/AS923)

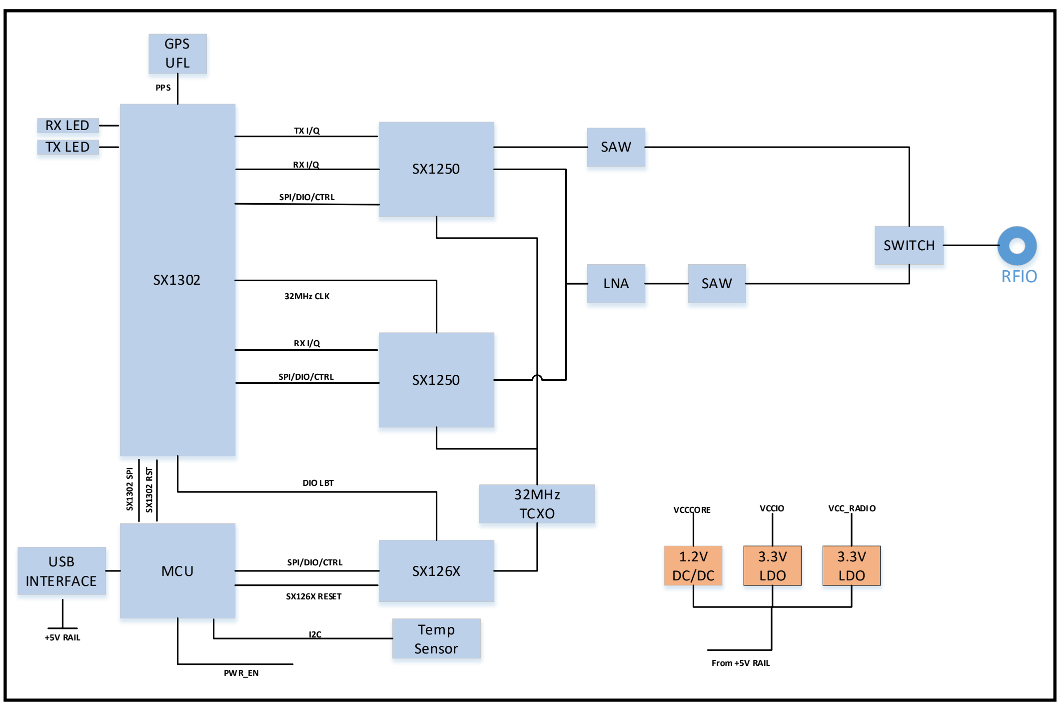

Figure 4: Corecell reference design V3 Block Diagram (CN490)

- 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 150-960 MHz ISM frequency bands.
- The STM32 MCU serves as a translator between the SX1302 and Raspberry Pi when using USB mode. The MCU converts the USB data it receives from the Raspberry Pi into SPI while talking to SX1302 and vice versa.
- For EU868/US915/AS923, the SX1261 works in receiver mode and listens for any interference signal before any data gets transmitted. It is used for the 'Listen before Talk' feature.
- For EU868/US915/AS923, the on-board Mother board main requirements to control signals from/to the Mini PCIe and the SX1302 are:
    - 1 x SPI: coming from host to the SX1302 SPI interface
    - 1 x I2C: coming from host to the temperature sensor I2C interface
    - Power Enable line
    - SX1302 reset line
    - PPS
- For CN490, the control signals from/to the 2*5 connector (2.00mm pitch) require:
    - 1 x SPI : coming from host to the SX1302 SPI interface
    - 1 x I2C : coming from host to the temperature sensor I2C interface
    - Power Enable line
    - SX1302 reset line
    - PPS signal input via UFL connector on the board

2.4 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 27 DBM 868MHZ	HAL util_tx_continuous	421	mA
8 RX channels OFF TX ON at 26 DBM 915MHZ	HAL util_tx_continuous	407	mA
8 RX channels OFF TX ON at 14 DBM 868MHZ	HAL util_tx_continuous	148	mA

Table 2: Typical Current Consumption at 5.0 V (EU868 and US915)

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

Table 3: Typical Current Consumption at 5.0 V (CN490)

3 Software Overview

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

- The **packet forwarder** is a program running 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 to build a Corecell reference design which communicates through SPI or USB interface with a concentrator board based on Semtech SX1302 multi-channel modem and SX1250 RF transceivers.
- The **utils_boot** and **dfu-util** (Firmware Updater) tools run on the host of LoRa® gateway and are used to program the STM32 MCU.
- The **spectral_scan** tool runs on the host gateway and it provides details of the wireless signals present in the surrounding area. The spectrum scan results are stored in a csv file which can be plotted using a python script.

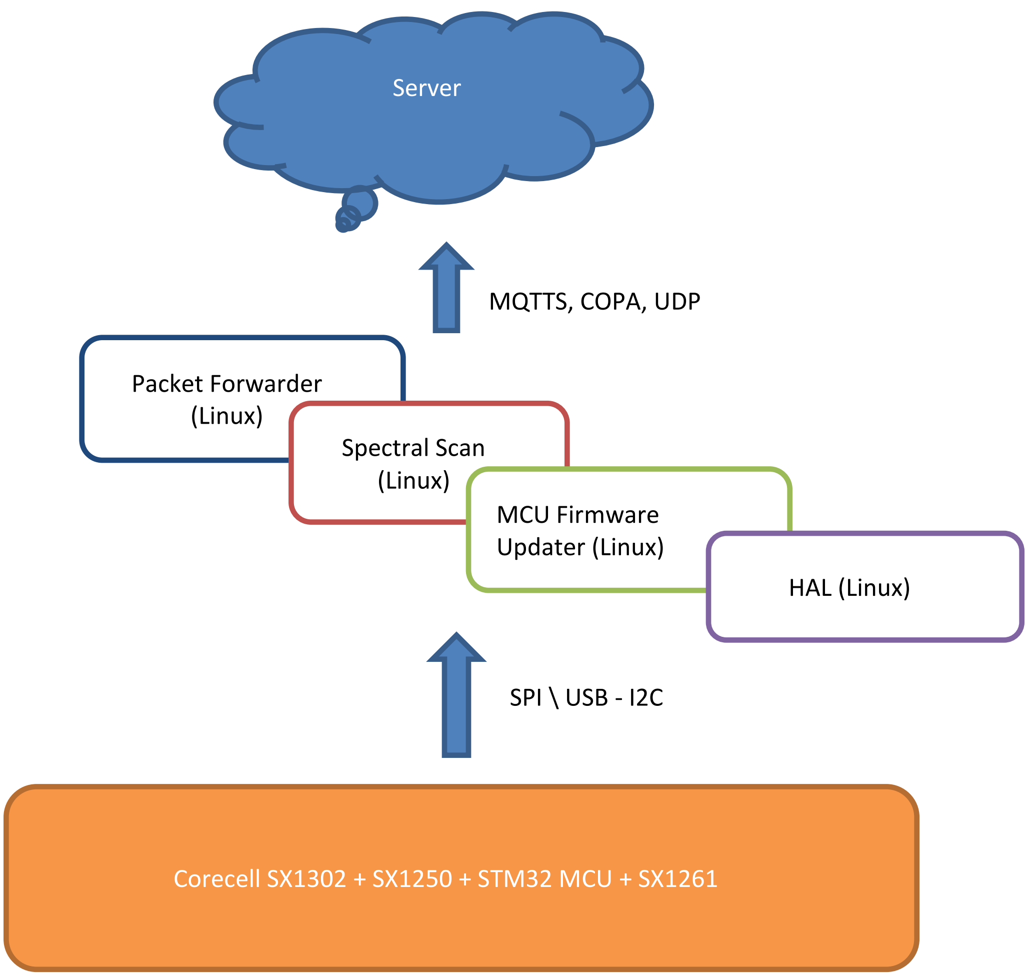

Figure 5: GW Software Overview

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

Notice!

- The default configuration file *global_conf.json.sx1250* is given as an example and may need to 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 Use With Raspberry Pi

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

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

4.1 Corecell Reference Design + Interface Board + Raspberry Pi Connection

The Corecell reference design can be configured to work with a USB interface or an SPI interface.

4.1.1 SPI Interface

For EU868 / US915 / AS923, simply connect the Corecell reference design to the interface board through the mini PCIe and connect the Raspberry Pi on the socket as depicted on the picture below:

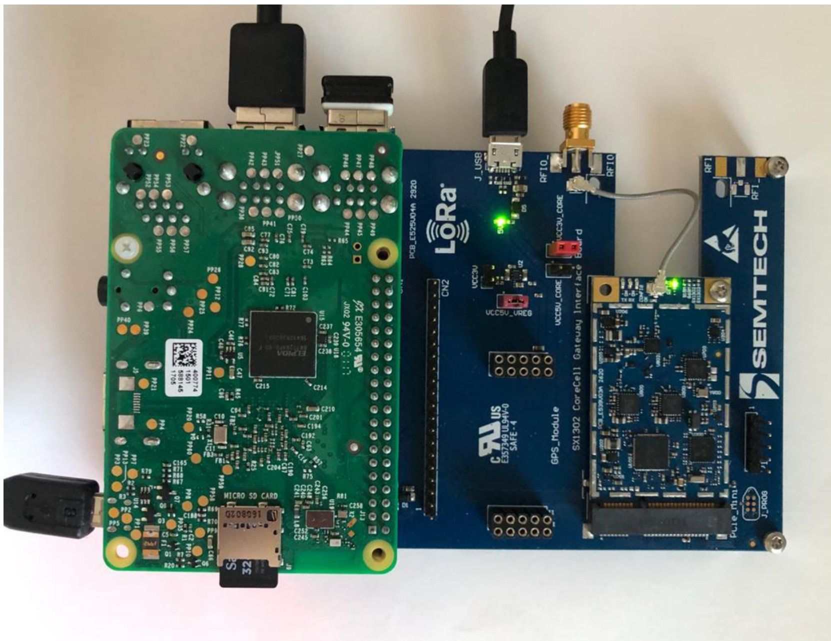

Figure 6: Corecell Reference Design + Interface Board + Raspberry Pi Connection (EU868/US915/AS923)

4.1.2 USB Interface

For EU868 / US915 / AS923, connect a micro-USB to USB-A cable from the micro-USB port on the interface board to the USB-A connector of the Raspberry Pi.

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

Figure 7: Corecell GW + Interface Board + Raspberry Pi Connection (CN490)

4.2 Raspberry Pi Image Software Installation

- Download the Raspbian image:
  - Go to address https://www.raspberrypi.org/downloads/raspbian/
  - Choose “RASPBIAN BUSTER LITE”
- Refer to the following guide to setup your SD card with the downloaded image: https://www.raspberrypi.org/documentation/installation/installing-images/
  - Format the SD card: https://www.sdcard.org/downloads/formatter/eula_windows/

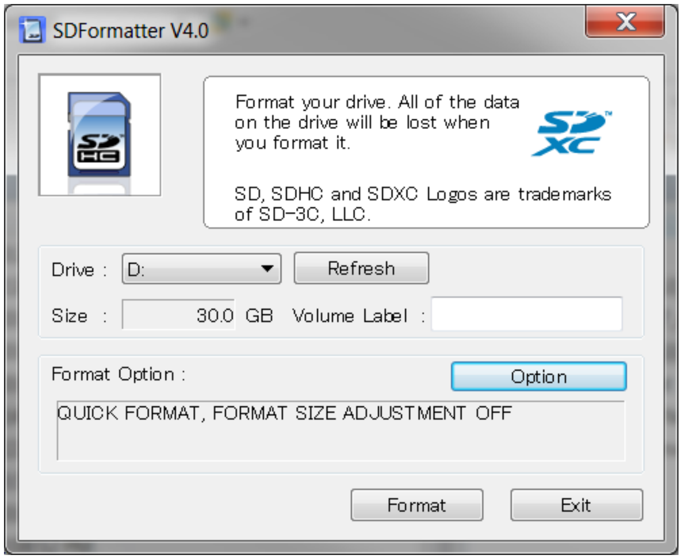

Figure 8: SDFormatter

  - Write the image previously downloaded on the SD card: https://sourceforge.net/projects/win32diskimager/

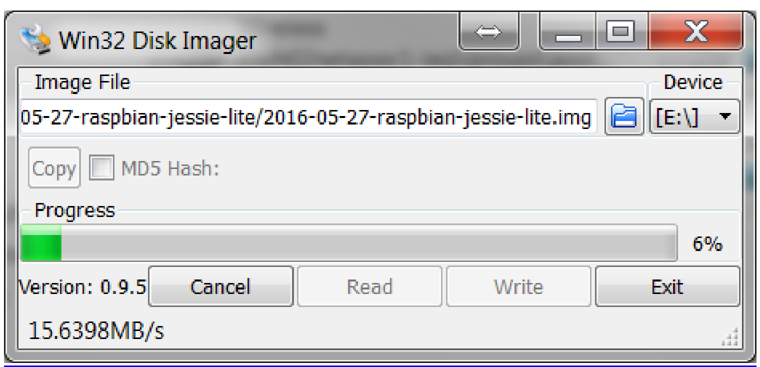

Figure 9: Win32 Disk Imager

4.3 Starting Raspberry Pi

Once the SD card is burned, insert it in the Raspberry Pi and choose a way to login to the Raspberry Pi:

- HDMI monitor and USB keyboard
- SSH connection: Enable SSH by placing a file named *ssh* (without any extension) in the SD card boot partition:

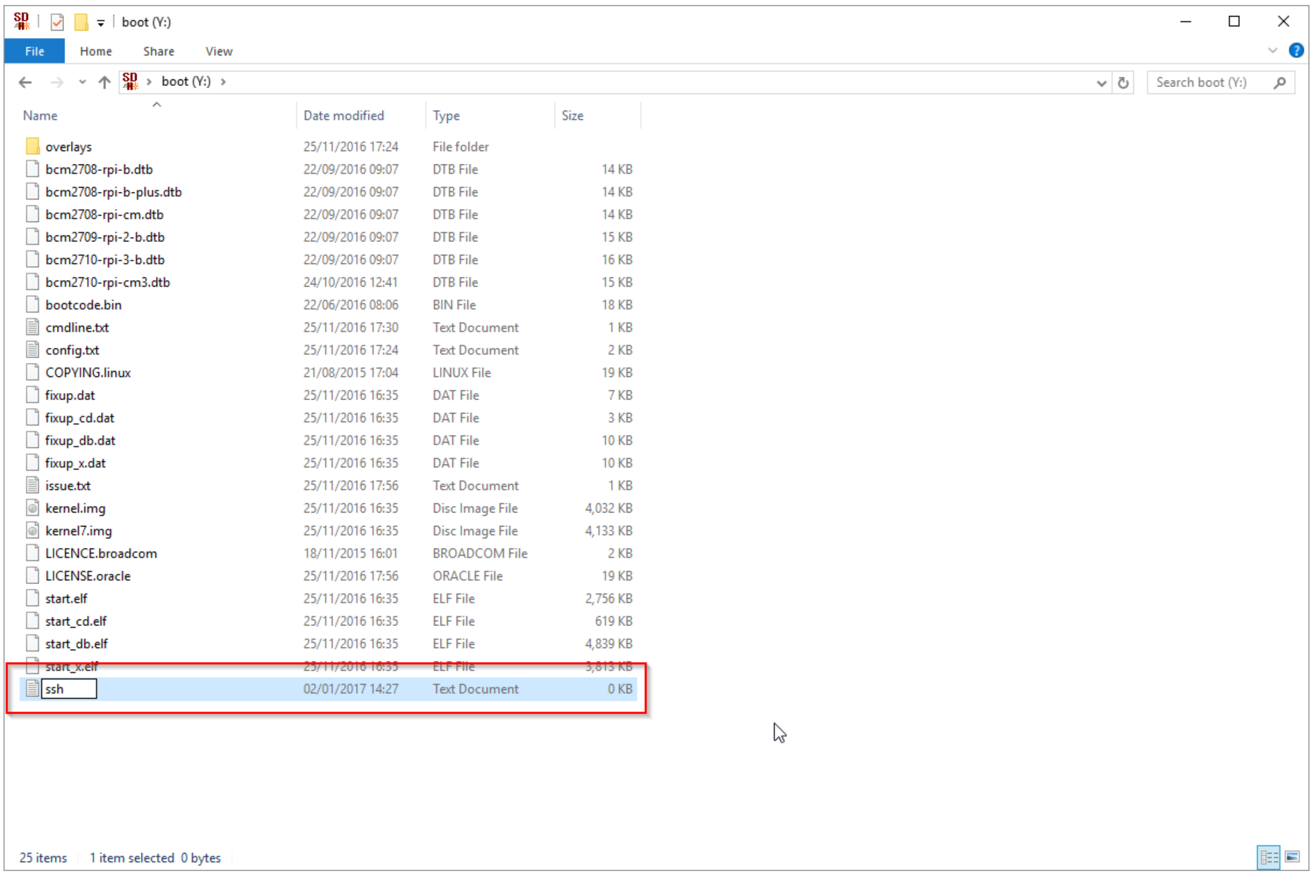

Figure 10: Enable SSH Connection On RPI

Below is the description through an SSH client enabled from raspi-config tool Interfacing Option (activated by HDMI monitor and USB keyboard):

4.3.1 Login: Pi And Password: raspberry

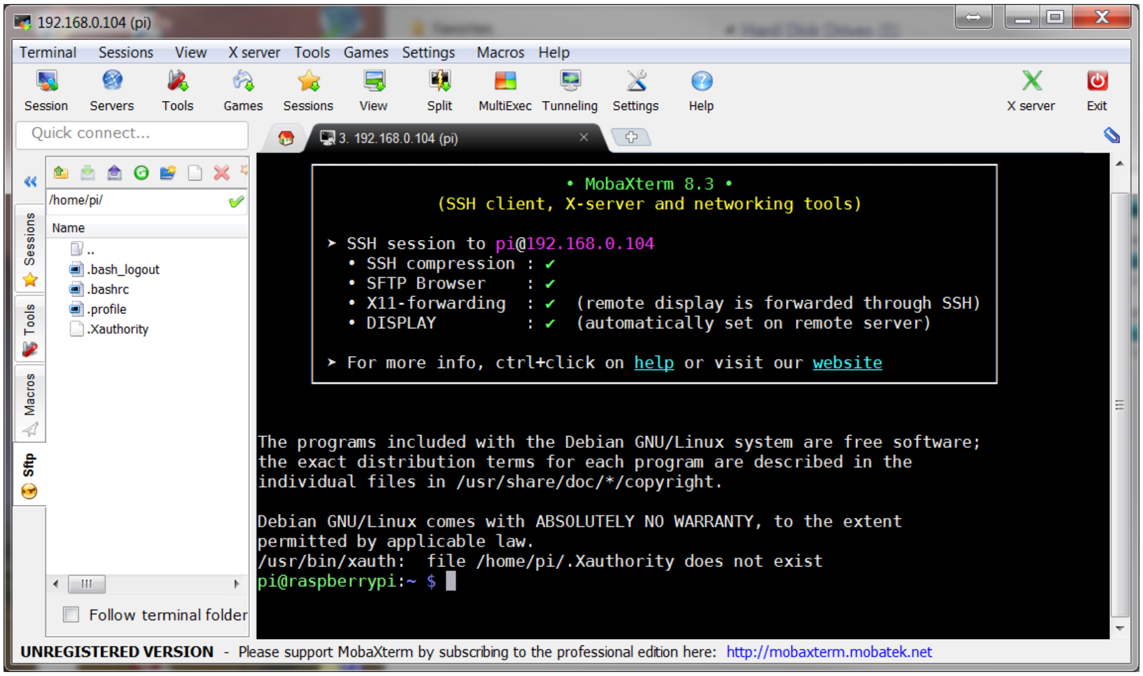

Figure 11: MobaXterm SSH Client

4.3.2 Update And Configure The 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`

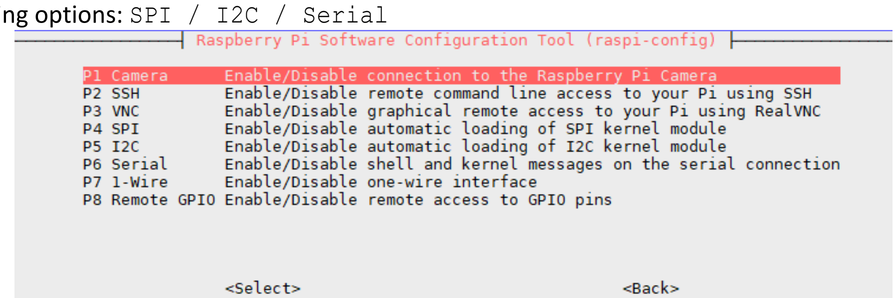

Figure 12: Enable SPI/I2C/UART

4. The system must be then rebooted: `$ sudo reboot`

4.3.3 Compile Semtech HAL + Packet Forwarder

1. Get the latest Semtech software package from LoRa® Github (requires a connection to internet):

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

```
Cloning into 'sx1302_hal'...
Username for 'https://ch02git1.semtech.com': bboulet
Password for 'https://bboulet@ch02git1.semtech.com':
remote: Enumerating objects: 3425, done.
remote: Counting objects: 100% (3425/3425), done.
remote: Compressing objects: 100% (1080/1080), done.
remote: Total 3425 (delta 2455), reused 3279 (delta 2333)
Receiving objects: 100% (3425/3425), 1.05 MiB | 0 bytes/s, done.
Resolving deltas: 100% (2455/2455), done.
```

Figure 13: Git Clone

   $ cd ~/sx1302_hal/

   $ make clean all

2. The *target.cfg* file located in the project's root directory configures where the executables must be installed.

- `TARGET_IP`: sets the IP address of the host of the gateway. In case the project is compiled on the gateway host itself (Raspberry Pi), this can be left set to `localhost`.
- `TARGET_DIR`: sets the directory on the gateway host file system in which the executables must be copied. The directory MUST exist when invoking the install command. By default it is: */home/pi/sx1302_hal/bin*
- `TARGET_USR`: sets the Linux user to be used to perform the SSH/SCP command for copying the executables. By default it is: pi

3. Execute the next two commands to avoid entering the user password when installing the files:

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

4. Now install:

- $ make install
- $ make install_conf

5. 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 14: Executables

4.3.4 Updating The STM32 MCU Firmware

When running the Corecell in USB mode, the STM32 MCU serves as a translator where 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 in the following way:

1. Navigate to the 'bin' directory where all the executables are.
   `$ cd ~/sx1302_hal/bin`
2. Issue following command:
   `./boot -d /dev/ttyACMx`
3. Download and install the DFU tool on the Raspberry pi. This can be done using following command:
   `sudo apt-get install dfu-util`
4. Now 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/xxx_CoreCell_USB.bin`
5. At this point the Corecell is ready to run.

4.3.5 Get The Unique ID Of The Gateway

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 Corecell running in USB configuration:

```
./chip_id -u -d /dev/ttyACM0
```

For 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 15: Util Chip ID

The gateway ID should be then replaced (in order have an 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,
```

4.3.6 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 16: HAL Compilation Check Results

4.3.7 Test The 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
```

For USB configuration:

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

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).

For more information, enter:

```
$ ./test_loragw_hal_tx -h
```

4.3.8 Configuring the 'Listen Before Talk' feature in Packet Forwarder

The Corecell supports 'Listen before Talk' (LBT) feature for 125 kHz and 250 kHz bandwidths only. This feature can be enabled/disabled in the packet forwarder from the *global_conf.json.sx1250.xxxx* file by setting the following parameters to true\false:

```
'SX130x_conf.sx1261_conf.lbt.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_conf": {
            "rssi_offset": 0, /* dB */
            "spectral_scan": {
                "enable": false,
                "freq_start": 922000000,
                "nb_chan": 8,
                "nb_scan": 2000,
                "pace_s": 10
            },
            "lbt": {
                "enable": true,
                "rssi_target": -80, /* dBm */
                "channels":[ /* 16 channels maximum */
```

Figure 17: Listen Before Talk configuration while running Packet Forwarder

4.3.9 Configuring the background Spectral Scan feature in Packet Forwarder

The Corecell supports spectral scan feature. It can either be run within the packet forwarder, as a background spectral scan, or with a standalone tool described in the next chapter. The example provided with the packet forwarder will just run a dedicated thread, regularly scanning a channel, and print the results of the scan on the console. It is left to the user to decide what to do with the data. (send it to the server, or analyze it in place). Please refer to the `thread_spectral_scan()` function of the packet forwarder source code, and extend it as required. The purpose of the example provided is to show how to use the spectral scan API exported by the HAL, in a way that it doesn't interfere with the main purpose of the gateway which is to receive and send packets.

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

```
'SX130x_conf.sx1261_conf.spectral_scan.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_conf": {
            "rssi_offset": 0, /* dB */
            "spectral_scan": {
                "enable": true,
                "freq_start": 922000000,
                "nb_chan": 8,
                "nb_scan": 2000,
                "pace_s": 10
            },
```

Figure 18: Spectral Scan configuration while running Packet Forwarder

The parameters available are :

- “freq_start” : the frequency of the first channel to be scanned, in Hz
- “nb_chan”: the number of channels to be scan (200Khz channels)
- “nb_scan”: the number of measures for each scan [65535 max]
- “pace_s”: the number of seconds between each scan.

The results will be printed on the console like:

```
SPECTRAL SCAN - 868500000 Hz: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 855 1145 0 0 0 0 0 0 0 0
```

The values being the number of points measured at a certain power, from the highest to the lowest (left to right). (the same format as the util_spectral_scan utility uses).

4.3.10 Run the Spectral Scanner

The Corecell supports a spectral scan feature via the standalone utility `util_spectral_scan`. (Note: it cannot be run while the packet forwarder or another 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 prior to use.

```
$ python3 plot_rssi_histogram.py rssi_histogram.csv
```

The output of the scan result:

```
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 0 766 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 0 623 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 0 562 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 0 486 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 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 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 0 903 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 0 660 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 0 657 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 0 895 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 0 659 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 0 639 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 0 502 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 0 794 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 0 452 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 0 682 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 0 666 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 0 613 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 0 538 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 0 638 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 0 884 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 0 719 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 0 640 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 0 626 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 0 811 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 0 650 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 0 623 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 17 1948 35 0 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 0 573 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 0 674 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 0 957 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 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 0 874 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 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 1065 935 0 0 0 0
Closing USB communication interface
pi@raspberrypi:~/corecellV3_testing/sx1302_hal-develop $
```

Figure 19: Spectral Scan Results

4.3.11 Run Packet Forwarder

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 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 20: Packet Forwarder

5 JSON File For RF Parameter 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` until `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 Factor 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 `lorawan_public` field of `lgw_conf_board_s`.

5.2 Typical JSON File

A typical Corecell reference design *global_conf.json* file for AS923 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_conf": {
            "rssi_offset": 0, /* dB */
            "spectral_scan": {
                "enable": false,
                "freq_start": 922000000,
                "nb_chan": 8,
                "nb_scan": 2000,
                "pace_s": 10
            },
            "lbt": {
                "enable": true,
                "rssi_target": -80, /* dBm */
                "channels":[ /* 16 channels maximum */
                    { "freq_hz": 920600000, "bandwidth": 125000, "scan_time_us": 5000, "transmit_time_ms": 4000 },
                    { "freq_hz": 920800000, "bandwidth": 125000, "scan_time_us": 5000, "transmit_time_ms": 4000 },
                    { "freq_hz": 921000000, "bandwidth": 125000, "scan_time_us": 5000, "transmit_time_ms": 4000 },
                    { "freq_hz": 921200000, "bandwidth": 125000, "scan_time_us": 5000, "transmit_time_ms": 4000 },
                    { "freq_hz": 921400000, "bandwidth": 125000, "scan_time_us": 5000, "transmit_time_ms": 4000 },
                    { "freq_hz": 921600000, "bandwidth": 125000, "scan_time_us": 5000, "transmit_time_ms": 4000 },
                    { "freq_hz": 921800000, "bandwidth": 125000, "scan_time_us": 5000, "transmit_time_ms": 4000 },
                    { "freq_hz": 923200000, "bandwidth": 125000, "scan_time_us": 5000, "transmit_time_ms": 4000 },
                    { "freq_hz": 923400000, "bandwidth": 125000, "scan_time_us": 5000, "transmit_time_ms": 4000 }
                ]
            }
        },
        "radio_0": {
            "enable": true,
            "type": "SX1250",
            "freq": 922300000,
            "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": 920000000,
            "tx_freq_max": 924000000,
            "tx_gain_lut":[
                {"rf_power": 0, "pa_gain": 0, "pwr_idx": 0},
                {"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": 9}
            ]
        },
        "radio_1": {
            "enable": true,
            "type": "SX1250",
            "freq": 923100000,
            "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": 0, "if": -300000},
        "chan_multiSF_1": {"enable": true, "radio": 0, "if": -100000},
        "chan_multiSF_2": {"enable": true, "radio": 0, "if":  100000},
        "chan_multiSF_3": {"enable": true, "radio": 0, "if":  300000},
        "chan_multiSF_4": {"enable": true, "radio": 1, "if": -300000},
        "chan_multiSF_5": {"enable": true, "radio": 1, "if": -100000},
        "chan_multiSF_6": {"enable": true, "radio": 1, "if":  100000},
        "chan_multiSF_7": {"enable": true, "radio": 1, "if":  300000},
        "chan_Lora_std":  {"enable": true, "radio": 0, "if":  300000, "bandwidth": 500000, "spread_factor": 8,
                           "implicit_hdr": false, "implicit_payload_length": 17, "implicit_crc_en": false, "implicit_coderate": 1},
        "chan_FSK":       {"enable": false, "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, /* disable class B beacon */
        "beacon_freq_hz": 923400000,
        "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

7 Revision History

Version	Date	Modifications
1.0	December 2020	First Release

8 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 protocol
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
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

Important Notice

Contact Information

Semtech Corporation
Wireless & Sensing Products
200 Flynn Road, Camarillo, CA 93012
E-mail: sales@semtech.com
Phone: (805) 498-2111, Fax: (805) 498-3804