Kerlink gateway channel selection

Nicolas sornin

Table of Contents

1. Revisions

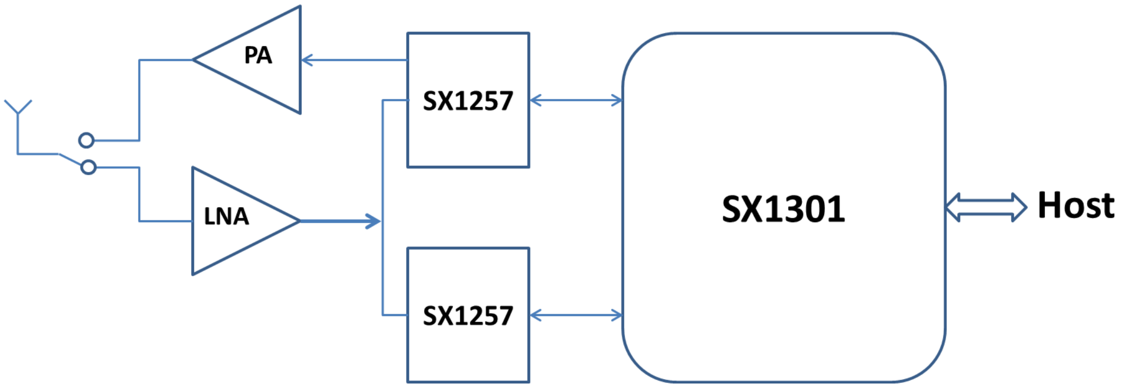
7/10/2014 : NS	Initial release	0.1
18/05/2015 : MC	Minor updates along with forwarder2_0.tar	0.2

2. Introduction

Before setting up a LoRaMAC radio network, it is essential to carefully select the correct frequency channels. A kerlink Wiregrid gateway handles simultaneously 10 frequency channels:

Channels	Data rate range	Channel bandwidth	Comment
CH1 to CH8	Lora 125kHz SF7 to SF12	125kHz	Adaptive Data Rate channels : 300bit/sec to 6kbit/sec used by most of the sensors
CH8	Lora 250kHz / SF7	250kHz	11kbit/sec "high speed" channel used by nodes with good radio link with the gateway
CH9	FSK	150kHz	50kbit/sec high speed channel used by nodes with Line Of Sight connectivity

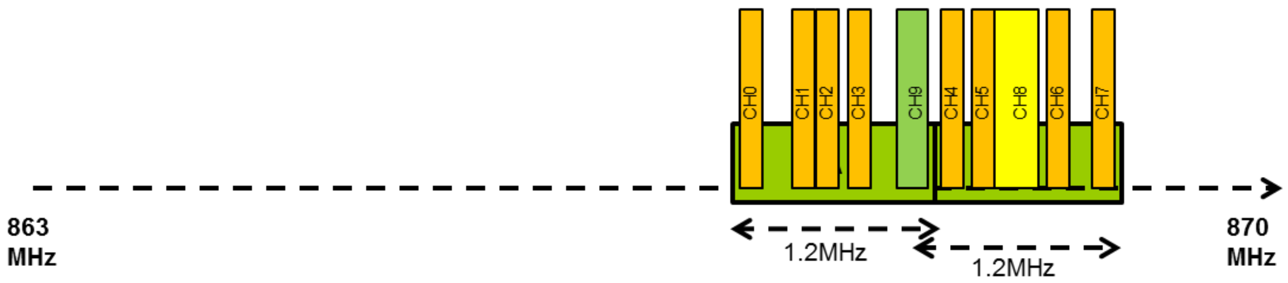
The gateway receive bandwidth extends from 863MHz to 870 MHz. However the 10 different channels cannot be dispatched arbitrarily inside this entire range because of the way the radio front-end receiver is actually implemented. The 10 channels must fall in two 1.2 MHz wide intervals. This is because each SX1257 radio front-end down-converts 1.2 MHz of spectrum. The two front-end central frequency can be set independently therefore the gateway can operate over two independent chunks of 1.2 MHz.



For example the following frequency plan would be valid. The 10 channels can be grouped in 2x 1.2 MHz intervals.

The following plan is not valid because CH0 and CH9 cannot fit inside the same 1.2 MHz interval.

The channels can be grouped arbitrarily, for example the following plan is totally valid:



The two 1.2MHz intervals can overlap without problem like in the previous example.
The number of channels in each interval is also arbitrary. The 10 channels may be grouped into a single 1.2MHz interval, or the intervals might contain any (N , 10-N) combination of channels.
For example this plan is valid:

The channels do not need to follow any precise order.
Two channels can overlap although this is strongly discouraged as it will most certainly reduce the network capacity by creating unnecessary collisions.

3. Programming channels

The complete gateway radio configuration is coded in a JSON file “global_conf.json” contained in the same directory than the packet_forwarder executable.
The file follows the JSON syntax rules as defined in http://www.json.org/

The gateway radio configuration is contained in the "SX1301_conf" field of the file.

We first define the center frequency of the two SX1257 radio front-ends. This corresponds to the center of the green boxes in the previous diagrams.

```
"radio_0": {
	"enable": true,
	"freq": 868400000
},
"radio_1": {
	"enable": true,
	"freq": 869100000
}
```

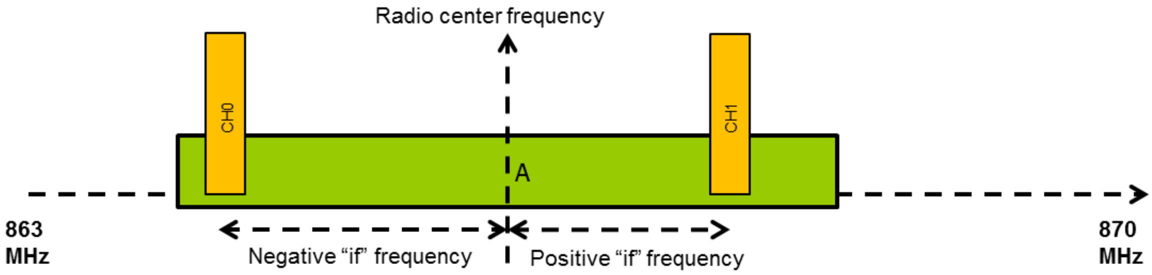
The frequency is expressed in Hz.

We then define the 8x adaptive data rate channels CH0 to CH7

```
"chan_multiSF_0": {
	/* Lora MAC channel, 125kHz, all SF, 868.1 MHz */
	"enable": true,
	"radio": 0,
	"if": -300000
}
```

The “radio” field identifies which radio front end that channel is related to, the “if” field sets the offset in Hz from this radio front-end center frequency.

In this example, the radio 0 operates at 868.4MHz and the CH0 uses an Intermediate Frequency "if offset" equal to -300kHz, therefore the actual CH0 frequency is 868.4 MHz – 300kHz = 868.1Mhz



The field name "chan_multiSF_0" cannot be changed else the JSON object will not be parsed adequately.
You can insert comments using /* */ at any place in the configuration file.
The maximum "if" frequency for the 125kHz multi-data rate channel is 1.2MHz/2 – 125kHz/2 = 537.5kHz

The definition of the high spped SF7/250kHz channel is done by:

```
"chan_Lora_std": {
        /* Lora MAC channel, 250kHz, SF7, 868.4 MHz */
        "enable": true,
        "radio": 0,
        "if": 0,
        "bandwidth": 250000,
        "spread_factor": 7
}
```

This channel uses a fixed spreading factor . For LoRaMAC network the spreading factor should be 7. This corresponds to 11kBit/sec.
The maximum "if" frequency is 1.2MHz/2 – 250kHz = 475kHz

Finally the FSK demodulation path configuration is set by:

```
"chan_FSK": {
        /* FSK 50kbps channel, 868.6 MHz */
        "enable": true,
        "radio": 0,
        "if": 200000,
        "bandwidth": 250000,
        "datarate": 50000
}
```

For LoRaMAC networks the FSK path is configured for 50kbit/sec. The maximum "if" frequency is also 1.2MHz/2 – 250kHz = 475kHz

4. Spectrum analysis

To maximize performances the channels frequencies should be selected such that the noise floor is locally as low as possible.
Semtech provides a spectrum analysis specific firmware that can be run on the gateway and provides a visual representation of the spectrum content.

To run this spectrum analysis tool on the gateway, you must first stop any packet forwarder process currently running on the gateway. Use the kill-pkt-fwd.sh script to do that:

```
[root@Wirgrid_0802001e forwarder_network_demo]# ./kill-pkt-fwd.sh
stopping the packet forwarder process through KNET agent ..
use restart-pkt-fwd.sh to restart it in its original configuration
Application is connected
SEND COMMAND: <?xml version="1.0"?><kms><appstop><app name="pkt-fwd"/></appstop></kms>
the radio modem card has been switched off.
```

Then go into the spectrum_scan directory of the installed package.

```
[root@Wirgrid_0802001e forwarder_network_demo]# cd spectrum_scan/
[root@Wirgrid_0802001e spectrum_scan]#
```

The README file provides a user guide for the util_rssi_histogram executable.
You can set the min and max frequency of the spectral scan, as well as the frequency step and time spent on each step.
The default scan (without any parameter) covers 863MHz to 870MHz with 50kHz step

```
[root@Wirgrid_0802001e spectrum_scan]# modem_on.sh
[root@Wirgrid_0802001e spectrum_scan]# ./util_rssi_histogram
Capture config: 32.77 ms, 125.00 kHz
Writing to file: rssi_histogram.csv
Channel: 863.000 MHz, Capturing... 10 20 30 40 50 60 70 80 90 done
RSSI 20%: -130, 50%: -130, 80%: -129
Channel: 863.050 MHz, Capturing... 10 20 30 40 50 60 70 80 90 done
RSSI 20%: -130, 50%: -130, 80%: -129
Channel: 863.100 MHz, Capturing... 10 20 30 40 50 60 70 80 90 done
RSSI 20%: -130, 50%: -130, 80%: -129
Channel: 863.150 MHz, Capturing... 10 20 30 40 50 60 70 80 90 done
```

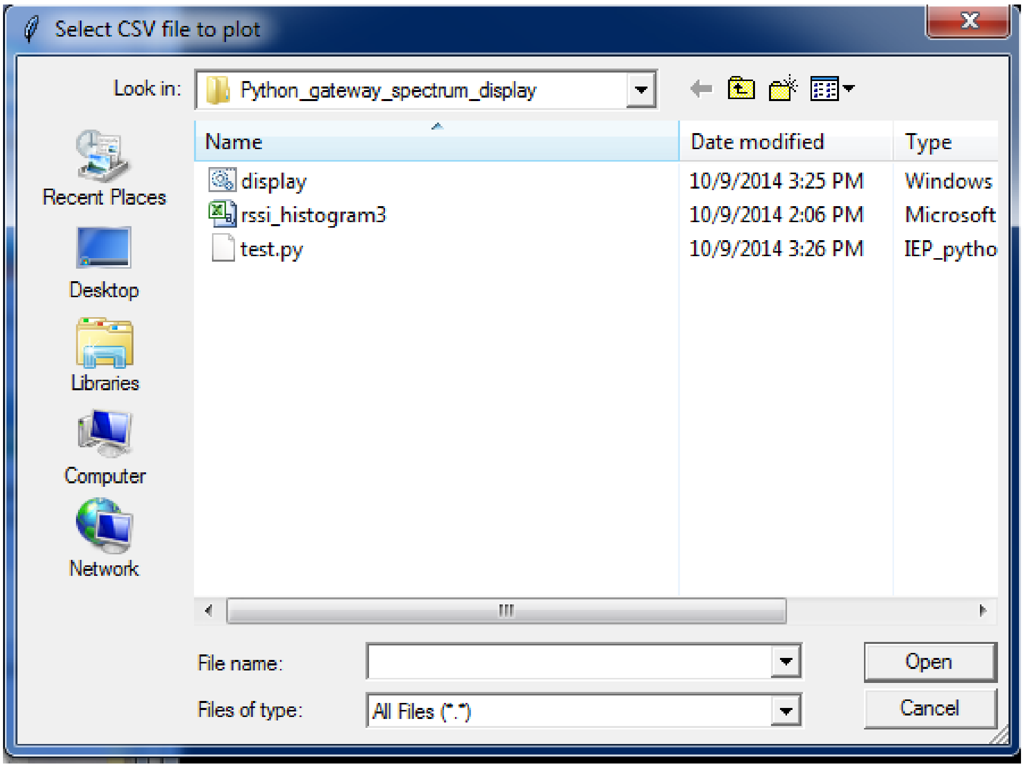
Once the capture is finished the executable dumps an rssi_histogram.csv file.

This file contains for each frequency the histogram of the channel power in dBm. That is the statistical repartition of the channel power vs time. The power is integrated over a 125kHz channel.
Example: The channel pent 2.3% of the time at -117dBm, then 3% at -116dBM, etc..

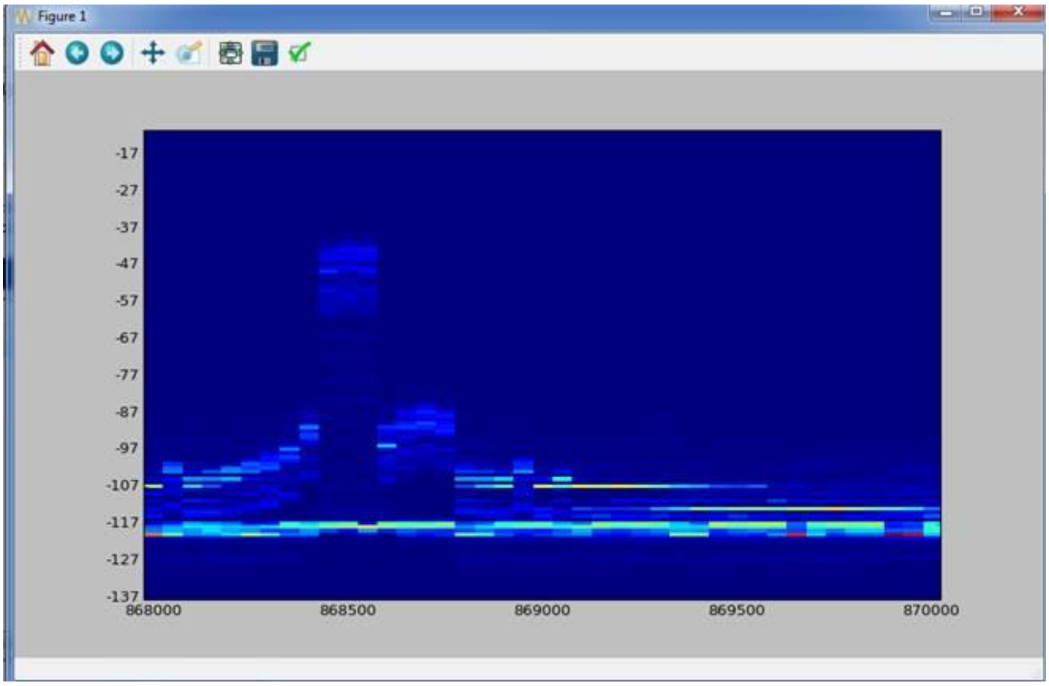
The package includes a python script to visualize the content of this file. *This is based on a "pyzo" installation, see annex.*

Copy the rssi_historgam.csv file to the python script directory on your computer using WinScp as described in the network_demo_guide document. You may rename the csv file if you want to capture several and compare them.

In the python script directory double-click on the display.bat batch file to execute it.
This batch file simply launches the python interpretation of the display_histogram.py script. You may have to modify the batch file to point to your python interpreter if your installation is not standard.
When the script executes, a file selector window opens, select the csv file you would like to visualize.



The 2D plot should appear



This a color map of the spectral power density.

Xaxis is the frequency

Yaxis , the channel integrated power in dBm

The color represents probability.

For example , this plot tells us that there is an interferer at 868.5Mhz present roughly 30% of the time , the rest of the time the channel power is -120dBm which is the receiver input noise floor.

5. Annex.

a. Installing “pyzo”

A Python based scientific calculation IDE.

The distribution used is at: http://www.pyzo.org/downloads.html

It is a self-installer and should run nicely on any windows / linux computer
It does not require administrator privilege.
To be sure all the required modules get impored successfully , you must run the IDE once in administrator mod.
To do this, right click on the desktop “pyzo” icon, then select “run as admin”
Once the IDE opens, open and execute the display_histogram.py script.
This will import all required python libraries. You only have to do this once.