WIRELESS & SENSING PRODUCTS

User Guide to the LoRa® Corecell Reference Design

Table of Contents

List of Figures

List of Tables

1 Introduction

This user guide introduces the Semtech LoRa® Corecell reference design V1.0 and how to set it up with a Raspberry Pi 3.

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

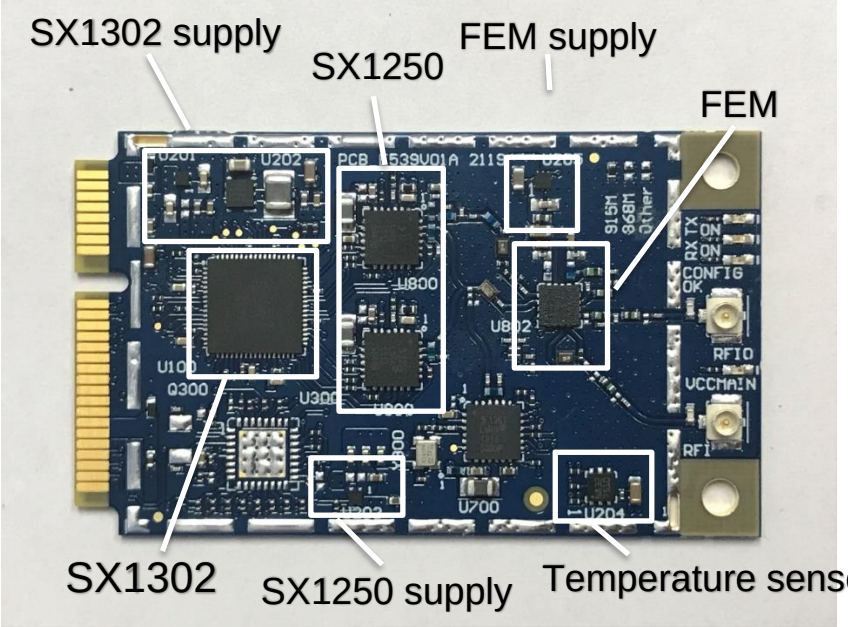

Figure 1: LoRa® Corecell reference design V1.0 (EU898/US915)

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

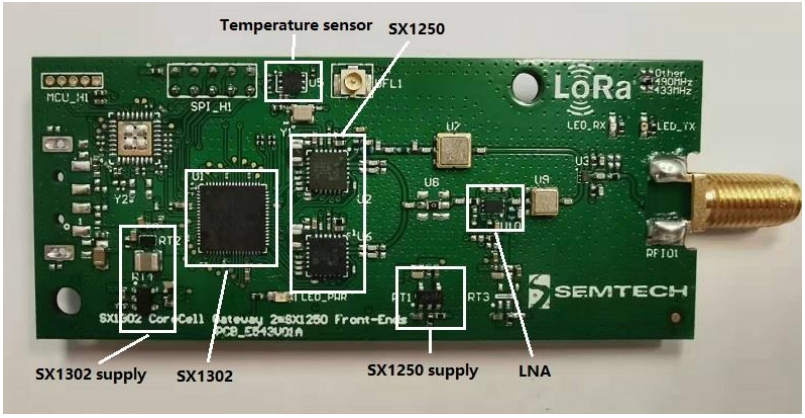

Figure 2 LoRa® Corecell reference design V1.0 (CN490)

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	^{o}C
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 (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 (CN490):
    - -141 dBm at SF12 BW 125 kHz
    - -126 dBm at SF7 BW 125 kHz
    - -111 dBm at FSK 50 kbps
- Ability to work in hostile RF environments such as close to cellular mobile phones, WiFi routers, Bluetooth devices

2.3Corecell reference design block diagram

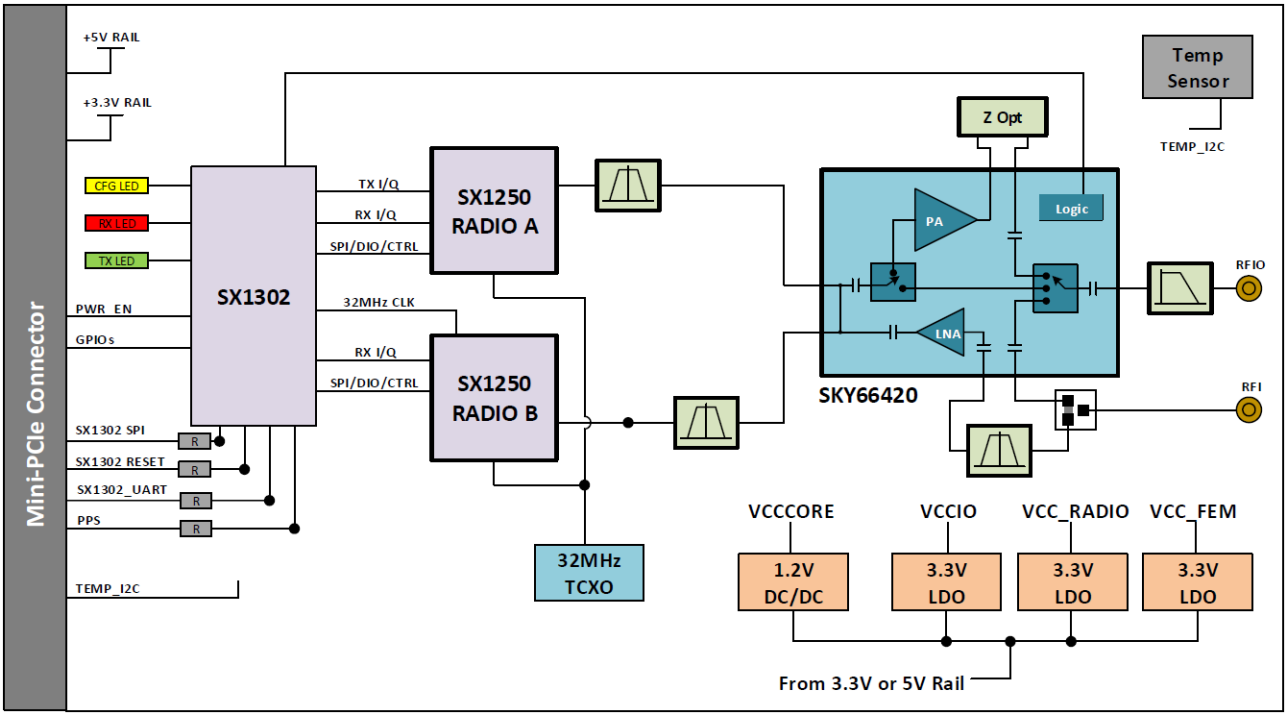

Figure 3: Corecell reference design V1.0 Block Diagram (EU868/US915)

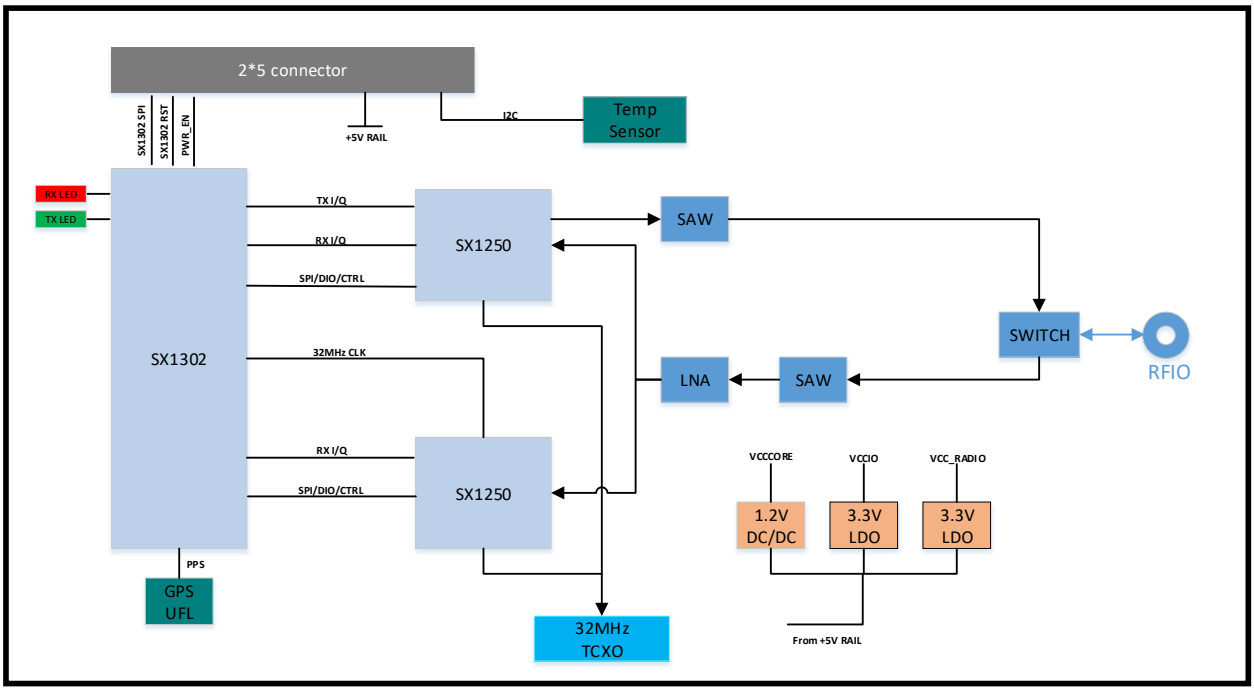

Figure 4 Corecell reference design V1.0 Block Diagram (CN490)

- 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.
- One SX1302 and two SX1250 transceivers are used to complete an eight channel LoRa concentrator.

For EU868/US915, the control signals from/to the Mini PCIe and the SX1302 device are described below:

On-board Mother board main requirements:

- 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) that described below:

- 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 split in two 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 with a concentrator board based on Semtech SX1302 multi-channel modem and SX1250 RF transceivers.

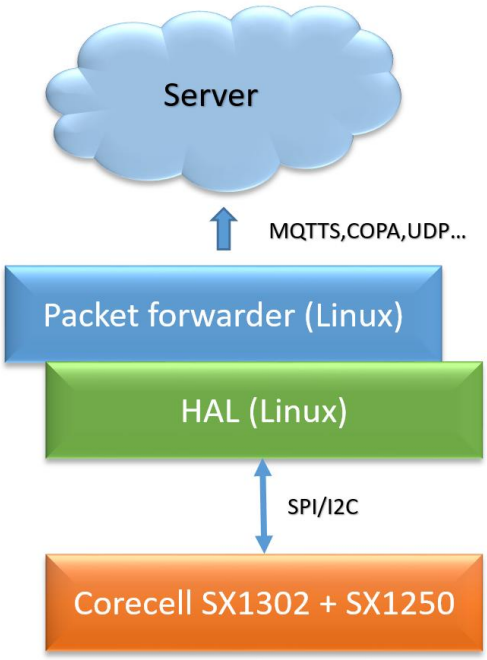

Figure 5: GW Software Overview

The packet_forwarder (gateway application) as well as sx1302_hal (SX1302 control library) source code can be found under LoRa® Github:

https://github.com/Lora-net/sx1302_hal

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

- sx1302_hal
- sx1302_hal/libloragw
- sx1302_hal/packet_forwarder
- sx1302_hal/util_net_downlink
- sx1302_hal/util_chip_id

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 located in the following directory: [PATH]/sx1302_hal/packet_forwarder.

4 Use with Raspberry Pi

The Semtech LoRa® Concentrator reference design has been tested with Raspberry Pi 3 model B

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

4.1 Corecell reference design + Interface board + Raspberry Pi Connection

For EU868 / US915, 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:

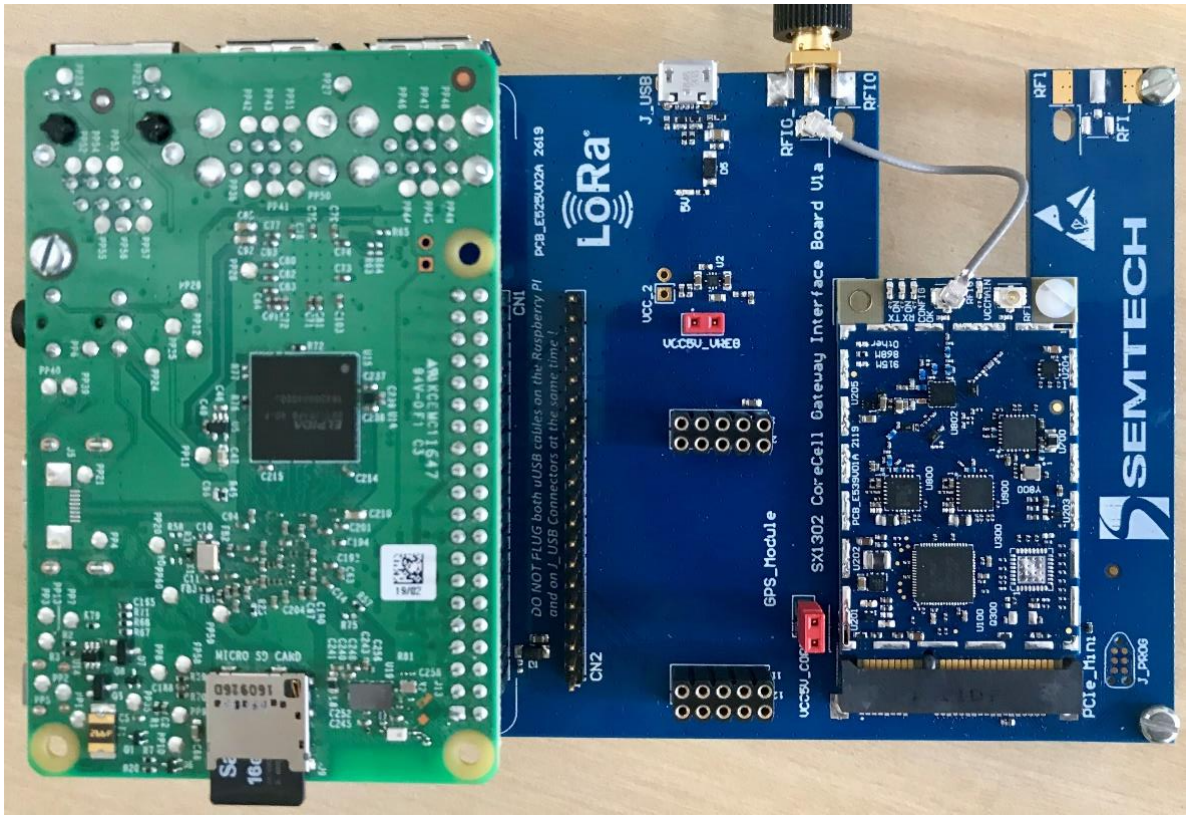

Figure 6: Corecell reference design + interface board + raspberry pi Connection (EU868/US915)

For CN490, simply connect the Corecell GW to the adapter board through the 2*5 connector and connect to the Raspberry pi as depicted on the picture below:

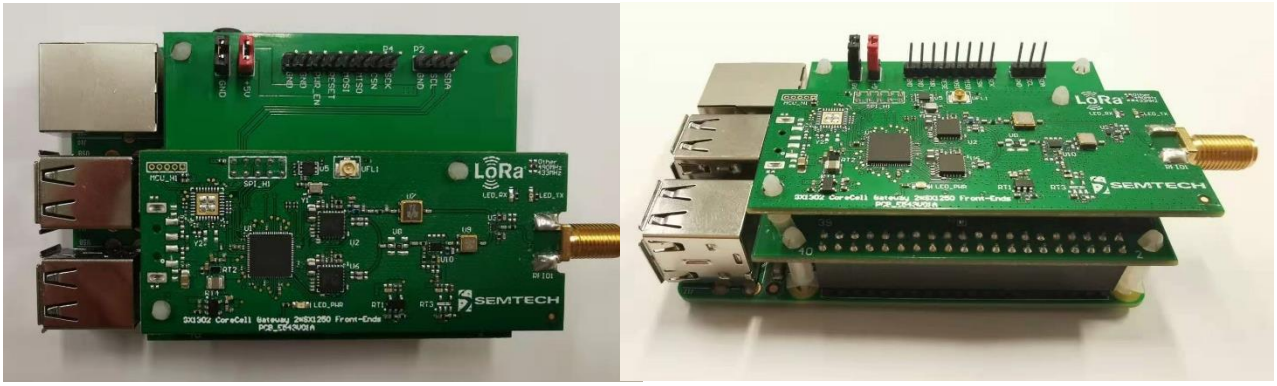

Figure 7: Corecell GW + interface board + raspberry pi Connection(CN490)

4.2Raspberry Pi Image Software Installation

- Download the Raspbian image:
  - Go to address https://www.raspberrypi.org/downloads/raspbian/
  - Choose “RASPBIAN BUSTER LITE”
- Refer to 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/

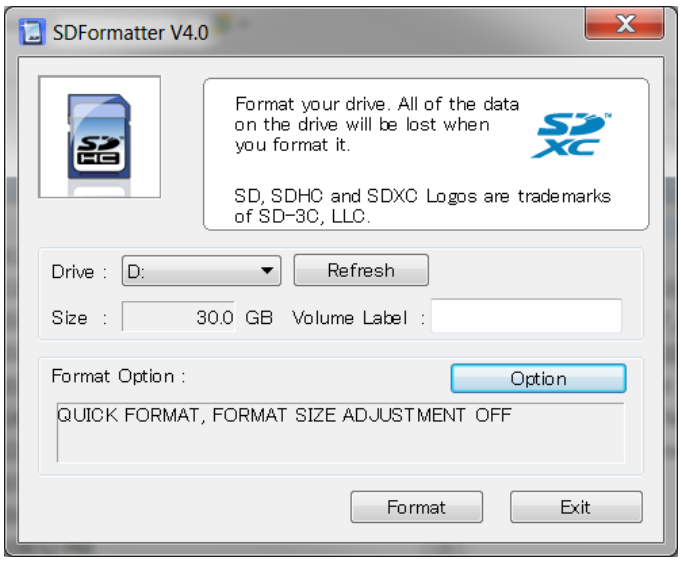

Figure 8: SDFormatter

- Write the image previously downloaded on the SD card:

  https://sourceforge.net/projects/win32diskimager/

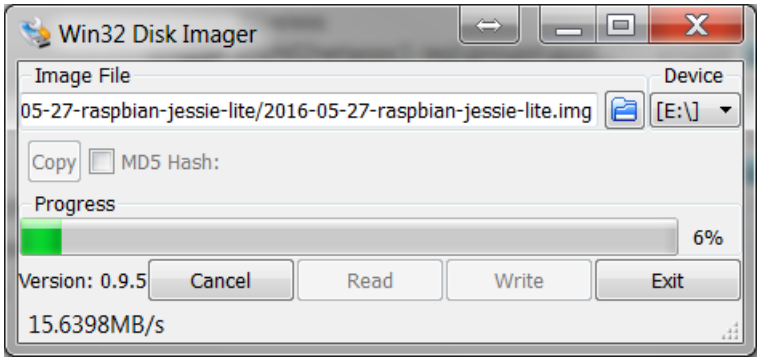

Figure 9: Win32 Disk Imager

4.3Starting Raspberry Pi

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

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

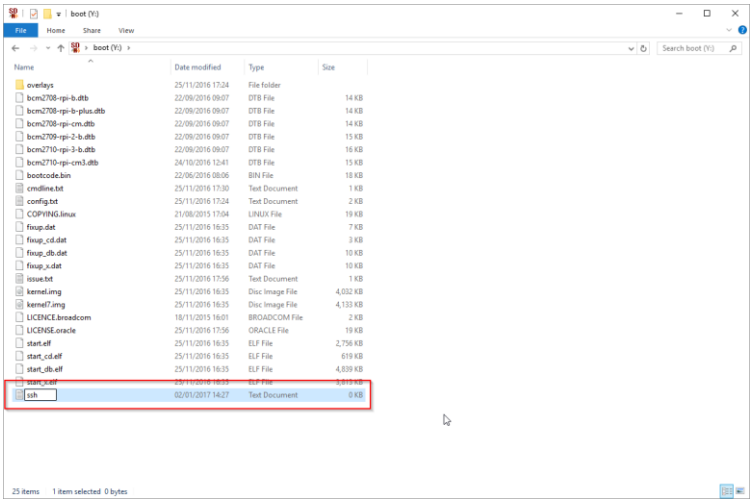

Figure 10 enable SSH connection on RPI

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

4.3.1 Login: pi and Password: raspberry

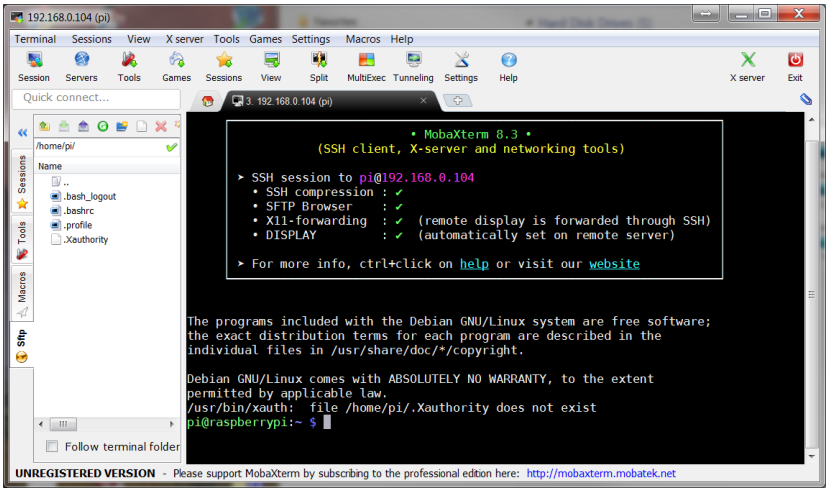

Figure 11: MobaXterm SSH Client

4.3.2 Resize Partition / FS

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

  $ sudo raspi-config

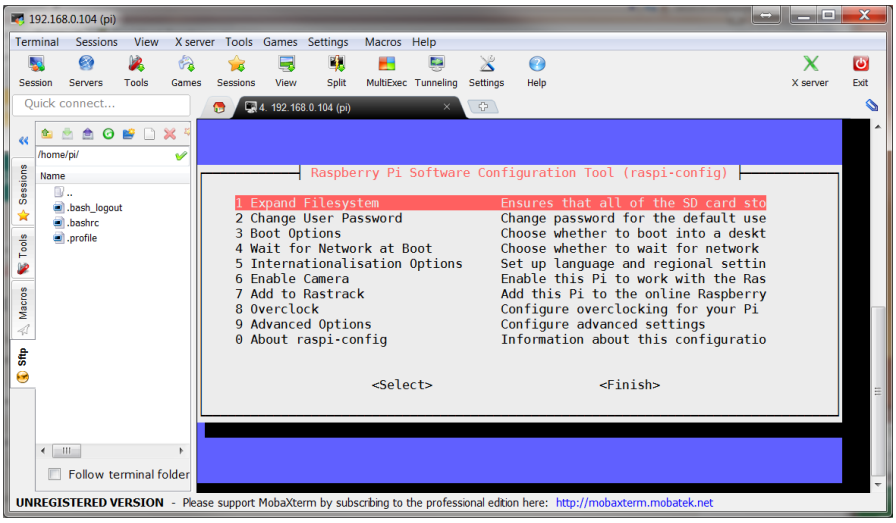

Figure 12: raspi-config Menu

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

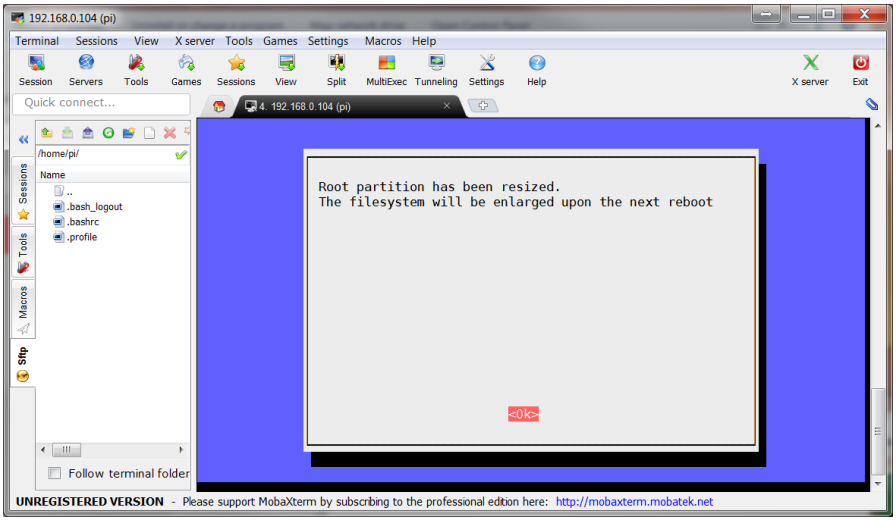

Figure 13: raspi-config "Expand Filesystem"

- The system must be then rebooted:

  $ sudo reboot

For more details, go to the following address:
https://www.raspberrypi.org/documentation/configuration/raspi-config.md

4.3.3 Update and configure the RPI

Update

Enter the following commands:

- *$* sudo apt-get update
- *$* sudo apt-get upgrade
- *$* sudo apt-get dist-upgrade
- *$* sudo rpi-update

Install Git

- *$* sudo apt install git

Enable SPI/I2C/UART

- *$* sudo raspi-config:
    - Interfacing options :
        - SPI
        - I2C
        - Serial

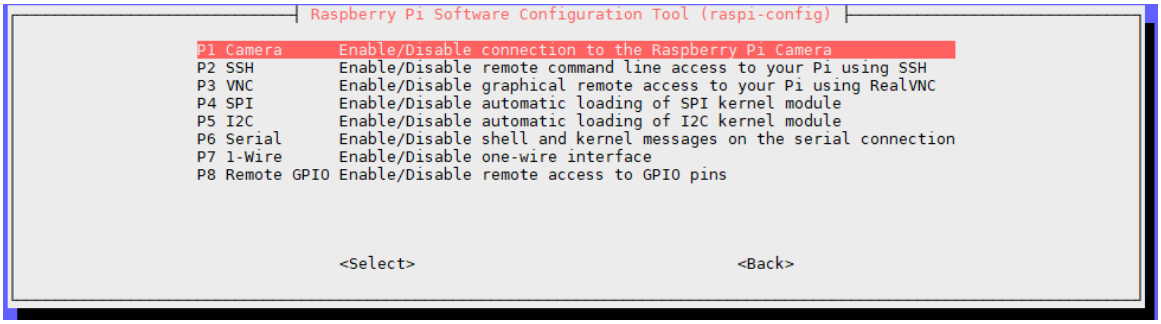

Figure 14 Enable SPI/I2C/UART

- The system must be then rebooted:

  $ sudo reboot

4.3.4 Compile Semtech HAL + Packet Forwarder

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

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

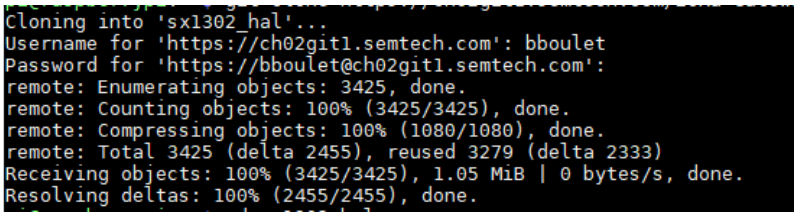

Figure 15: Git clone

- *$ cd ~/sx1302_hal/*
- *$ make clean all*
- *$ ssh-keygen -t rsa*
- *$ ssh-copy-id -i ~/.ssh/id_rsa.pub pi@localhost*
    - These both commands just above are executed in order to avoid entering the user password when installing the files
- *$ make install*
    - You shall enter the password (raspberry) several times. ➔ install all programs
- *$ make install_conf*
    - install global_conf.json

The executables are be copied in the *bin* folder.

```
pi@raspberrypi:~/sx1302_hal/bin $ ls -l
total 1532
-rwxr-xr-x 1 pi pi 119124 Jul 12 12:58 chip_id
-rw-r--r-- 1 pi pi   4092 Jul 12 12:58 global_conf.json.sx1250
-rw-r--r-- 1 pi pi   4630 Jul 12 12:59 global_conf.json.sx1257
-rwxr-xr-x 1 pi pi 199284 Jul 12 12:58 lora_pkt_fwd
-rwxr-xr-x 1 pi pi  50016 Jul 12 12:58 net_downlink
-rwxr-xr-x 1 pi pi   1672 Jul 12 12:58 reset_lgw.sh
-rwxr-xr-x 1 pi pi 136580 Jul 12 12:58 test_loragw_cal
-rwxr-xr-x 1 pi pi  31020 Jul 12 12:58 test_loragw_capture_ram
-rwxr-xr-x 1 pi pi 119452 Jul 12 12:58 test_loragw_counter
-rwxr-xr-x 1 pi pi 129460 Jul 12 12:58 test_loragw_gps
-rwxr-xr-x 1 pi pi 123560 Jul 12 12:58 test_loragw_hal_rx
-rwxr-xr-x 1 pi pi 128192 Jul 12 12:58 test_loragw_hal_tx
-rwxr-xr-x 1 pi pi 119304 Jul 12 12:58 test_loragw_i2c
-rwxr-xr-x 1 pi pi 119348 Jul 12 12:58 test_loragw_reg
-rwxr-xr-x 1 pi pi 119420 Jul 12 12:58 test_loragw_spi
-rwxr-xr-x 1 pi pi 119424 Jul 12 12:58 test_loragw_spi_sx1250
```

Figure 16 executables

4.3.5 Semtech HAL Compilation Check

The program *test_loragw_spi* is used to 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: press Ctrl+C to stop the application.

- *$ cd ~/sx1302_hal/bin*
- *$ ./test_loragw_spi*

The output looks like this:

```
pi@raspberrypi:~/sx1302/sx1302_hal/libloragw $ ./test_loragw_spi
Accessing CoreCellSX1302 reset pin through GPIO23...
Accessing CoreCellSX1302 power enable pin through GPIO18...
Beginning of test for loragw_spi.c
SX1302 version: 0x10
Cycle 0 > did a 980-byte R/W on a data buffer with no error
Cycle 1 > did a 638-byte R/W on a data buffer with no error
Cycle 2 > did a 748-byte R/W on a data buffer with no error
Cycle 3 > did a 275-byte R/W on a data buffer with no error
Cycle 4 > did a 426-byte R/W on a data buffer with no error
Cycle 5 > did a 781-byte R/W on a data buffer with no error
Cycle 6 > did a 907-byte R/W on a data buffer with no error
Cycle 7 > did a 422-byte R/W on a data buffer with no error
Cycle 8 > did a 293-byte R/W on a data buffer with no error
Cycle 9 > did a 589-byte R/W on a data buffer with no error
Cycle 10 > did a 317-byte R/W on a data buffer with no error
Cycle 11 > did a 243-byte R/W on a data buffer with no error
Cycle 12 > did a 990-byte R/W on a data buffer with no error
Cycle 13 > did a 216-byte R/W on a data buffer with no error
Cycle 14 > did a 385-byte R/W on a data buffer with no error
Cycle 15 > did a 226-byte R/W on a data buffer with no error
Cycle 16 > did a 22-byte R/W on a data buffer with no error
Cycle 17 > did a 566-byte R/W on a data buffer with no error
Cycle 18 > did a 48-byte R/W on a data buffer with no error
```

Figure 17 : test_loragw_spi

4.3.6 Get the Unique ID to the Gateway

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

$ cd ~/sx1302_hal/bin

$ ./chip_id

Return a unique ID like the following:

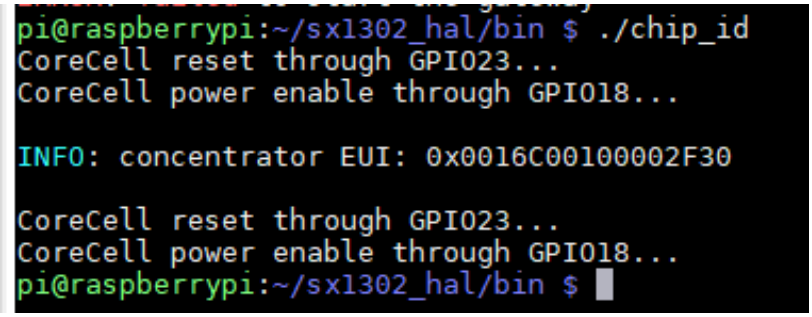

Figure 18 util chip ID

The gateway ID could be then replaced in the global_conf.json.sx1250 file within the repository:

~/sx1302_hal/bin/global_conf.json.sx1250

```
"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 Test the HAL TX

The program *./test_loragw_hal_tx* is used to test the emission of the Corecell reference design with settings set by the user.

The tests run endlessly or until an error is detected: press Ctrl+C to stop the application.

- *$ cd ~/sx1302_hal/bin*
- *$* ./test_loragw_hal_tx -k0 -c0 -r1250 -f868.1 -mLORA --pa 1 -l12 --pwid 14 -s7 -b125 -z16 -n10000 -t 100

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

For more information, enter:

$ ./test_loragw_hal_tx -h

4.3.8 Run Packet Forwarder

The Packet Forwarder is a program running on the host of a LoRa® Gateway that forward RF packets received by the concentrator to a server through an IP/UDP link, and emits RF packets that are sent by the server.

Run Packet Forwarder for a functional check:

$ 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 19: Packet Forwarder

5 JSON file for RF Parameter Tuning

Edit the file *~/sx1302_hal/bin/global_conf.json.sx1250* to update the following RF parameters:

- *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 available gain tables *tx_lut_12* until *tx_lut_27* to tune Tx output power thanks to the 3 following parameters:
    - *pa_gain* : [0 1] PA Enable Corecell reference design V1.0, 0 means PA bypassed, 1 means 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*.

A typical Corecell reference design *global_conf.json* file looks like this:

```
{
        "SX130x_conf": {
                "spidev_path": "/dev/spidev0.0",
                "lorawan_public": true,
                "clksrc": 0,
                "antenna_gain": 0, /* antenna gain, in dBi */
                "full_duplex": false,
                "precision_timestamp": {
                        "enable": false,
                        "max_ts_metrics": 255,
                        "nb_symbols": 1
                },
                "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_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": 128,
                    "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"
        }
}
```

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 how of the "lorawan_public" field of lgw_conf_board_s is set.

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	July 2019	First Release
1.1	September 2019	Minor corrections
1.2	November 2019	Add CN490

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

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.

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
www.semtech.com