KERLINK Gateway

Description: This document contains information related to the implementation of the KERLINK Gateway.

www.pcvuesolutions.com

FRANCE - Paris
ARC Informatique
Head Office

GERMANY - Munich
PcVue GmbH

ITALY - Milan
PcVue Srl

UK - London
Control Technology International

USA - Boston
PcVue Inc

SINGAPORE - Singapore
PcVue Sea

MALAYSIA – Kuala Lumpur
PcVue Sdn Bhd

CHINA - Shangai
PcVue China

JAPAN - Nagoya
PcVue Japan

Keywords: AA, KERLINK, Gateway

Last Revision Date: 2 April 2025

ARC Informatique
ISO 9001 : 2008
ISO 14001 : 2004
certified

Authorization

	Name	Stamp	Date
Written by	Matthias Risy		2 july 2020
Checked by	Fabien Boissie		29th July 2020
Authorized by			

Revision history

Revision	Author	Action	Editing	Date	Distribution
1.0.a	MARI	Create doc		7th July20	FABO
1.0.b	MARI	Add Reset procedure		28th July 20	FABO
1.1	FABO	Check document		29th July 20	KB
1.2.a	MARI	Update Node-RED configuration		01st April 21	FABO
1.3.a	MARI	Add writing configuration		02nd November 2021	FABO
2.0	FABO	Check document (new KB)		02nd April 2025	FABO

Table of content

1. Prerequisite

To have a KERLINK Wirnet iStation 868 gateway configured as a small private network SPN and with embedded Node-RED

2. Introduction

This document describes the procedure for installing KERLINK Gateway and receiving sensors data.

3. Small Private Network (SPN)

3.1. Connection

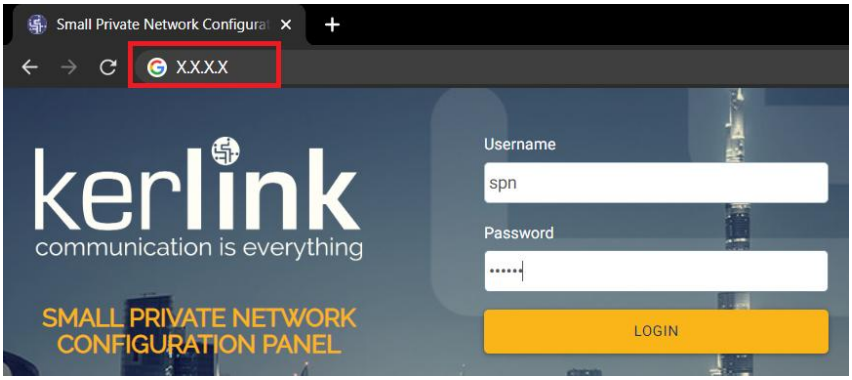

Figure 1 : SPN Connection

To log in, put your gateway IP in URL address bar.
By default, username and password are:

- Username: spn
- Password: spnpwd

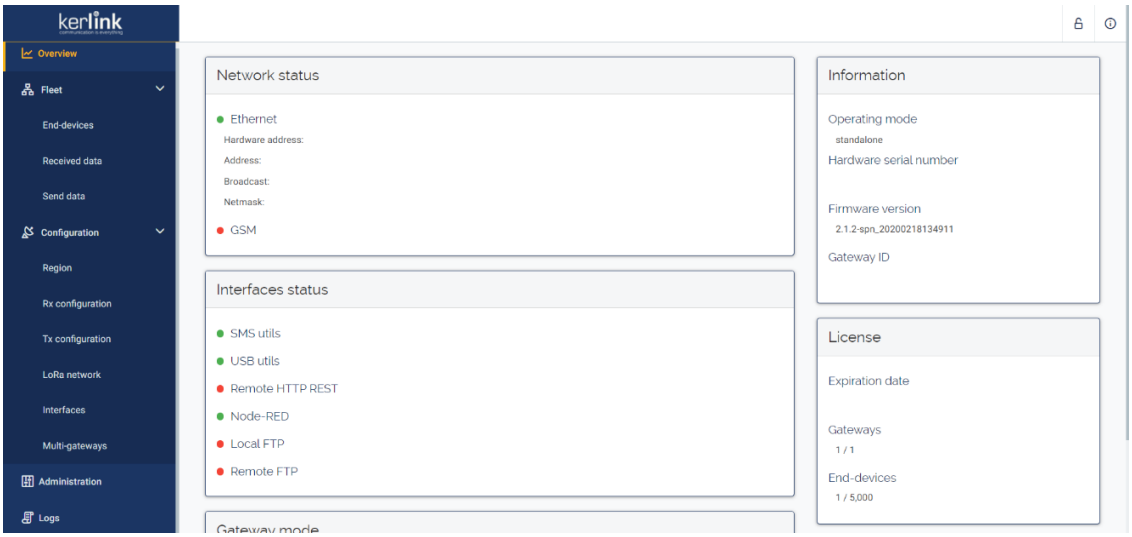

Figure 2 : SPN Overview after connection

3.2. License addition

Once logged in, if not already done, license must be adding. With control panel, on the left, click in administration then license. License file can drop in red zone.

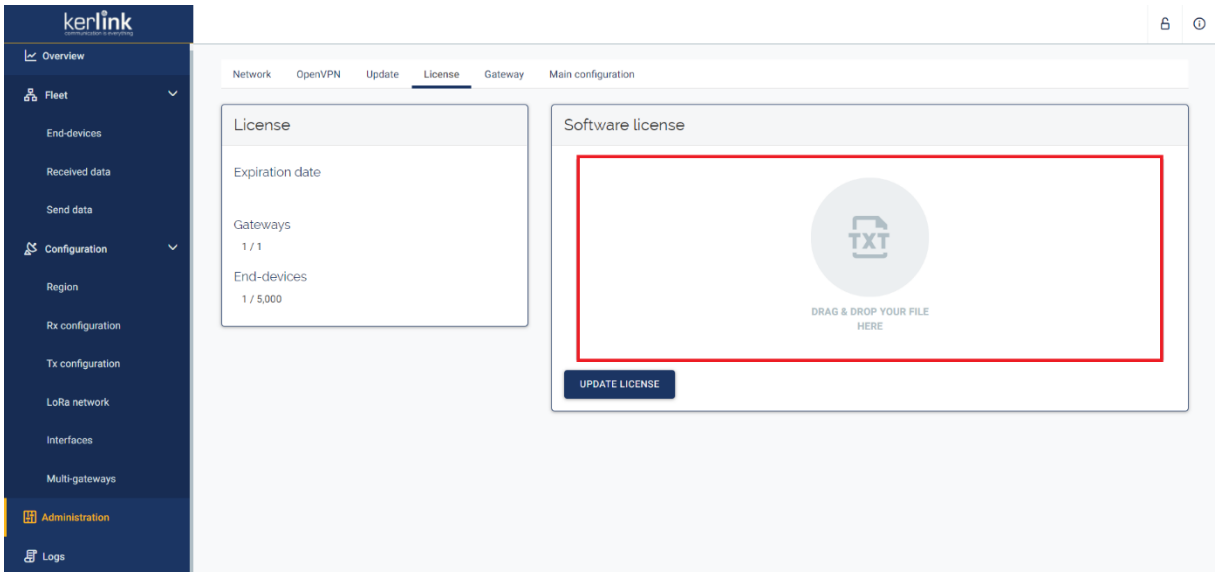

Figure 3 : License addition

It’s probably necessary to reboot the gateway.

3.3. Node-RED addition

Under Configuration, Interface and Node-RED, check the square to enable Node-RED.

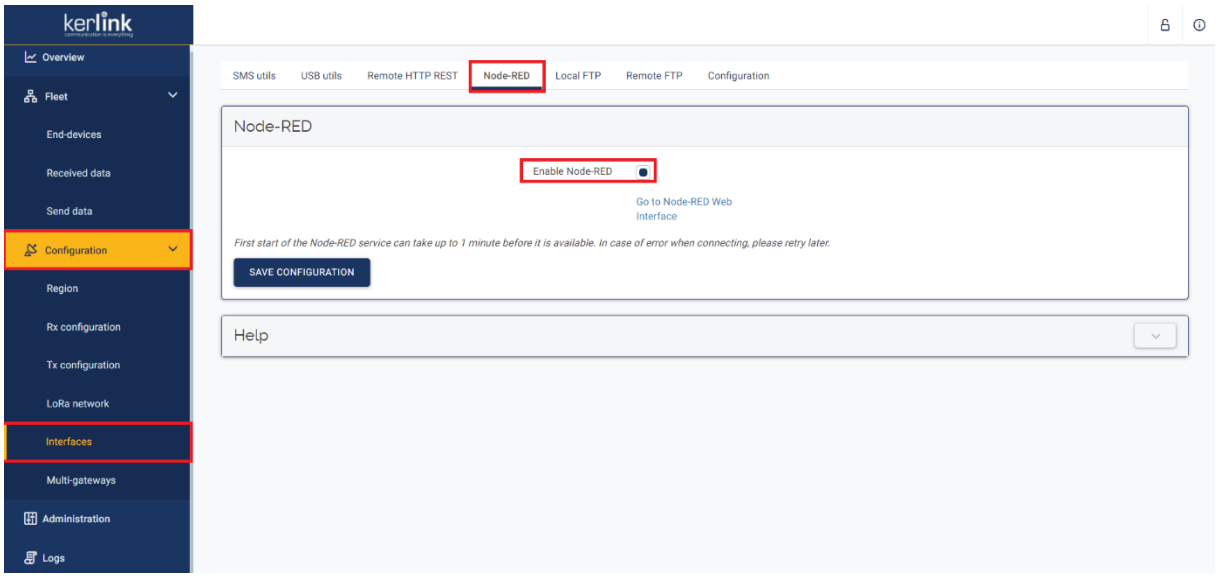

Figure 4 : Node-Red activation

It’s probably necessary to reboot the gateway.

3.4. Sensors addition

3.4.1. Sensors description

Each sensor has a unique EUI identifier. The first 3 bytes are constructor references. They all have a APPEUI and APPKEY.
For example, a TEMP sensor from ADEUNIS:
EUI : 00-18-B2-20-00-00-15-7F
APPEUI : XXXXXXXXXXXXXX
APPKEY : XXXXXXXXXXXXXXXXXXXXXXXXXX

Note: "00-18-B2" is the Adeunis manufacturer's reference.

3.4.2. Sensors addition in SPN

In Fleet, End-device and Manage end-devices, the addition of sensors can be done manually where it will be necessary to inform EUI, APPEUI, APPKEY (figure 5) but it is also possible to add more with a CSV file (figure 6 & 7).

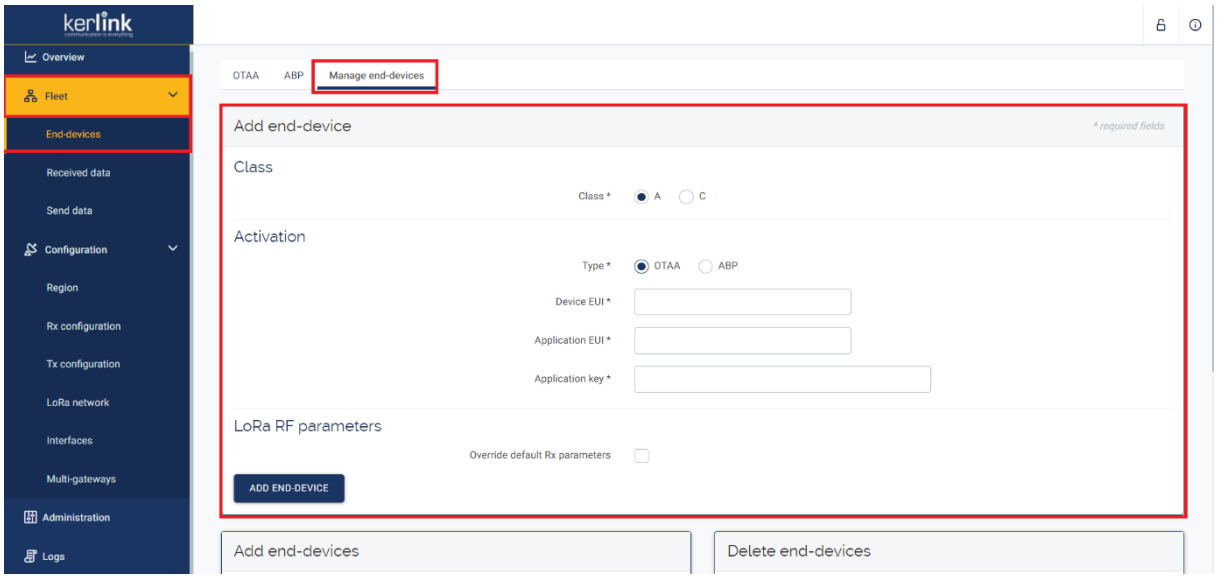

Figure 5 : Sensors addition

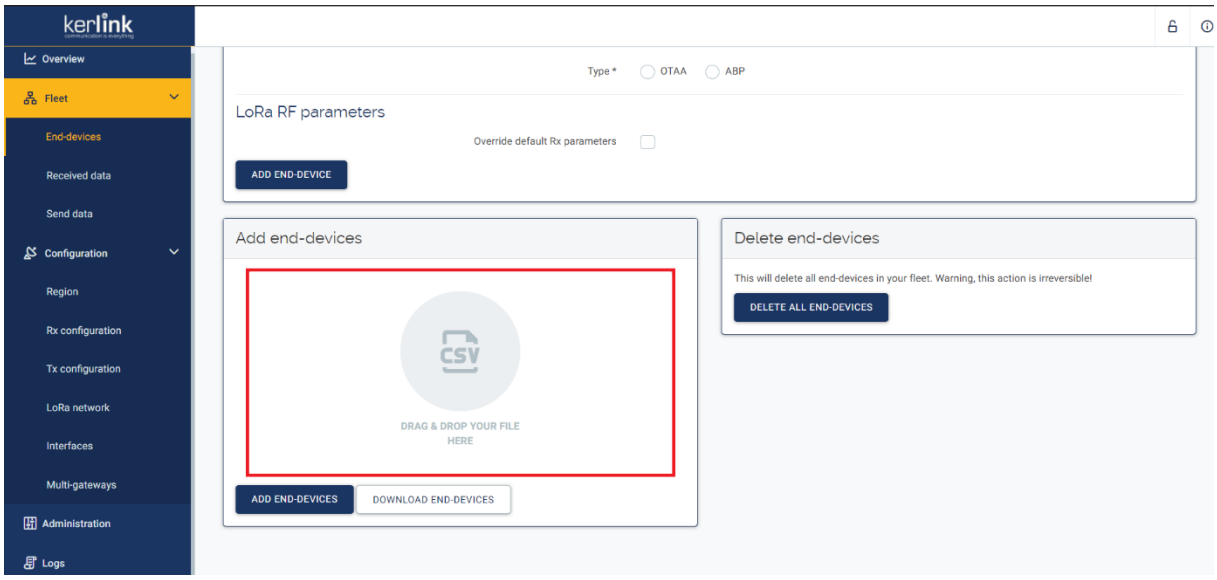

Figure 6 : Drop location of the CSV file

	A	B	C	D	E	F	G	H	I	J	K	L
1	activation_type (IN)	class (IN)	dev_eui (IN)	app_eui (IN)	app_key (IN)	dev_addr (IN)	nwks_key (IN)	apps_key (IN)	rx_window (IN)	rx_frequency (IN)	rx_datarate (IN)	status (OUT)
2	OTAA	A	XXX	XXX	XXX				255	869525000	0	UNKNOWN

Figure 7 : CSV file format

4. Node-RED

4.1. Reading configuration

In this part, Node-RED configuration will be explained.

To launch Node-Red, click on "Go to Node-RED Web Interface" in the node-RED tab.

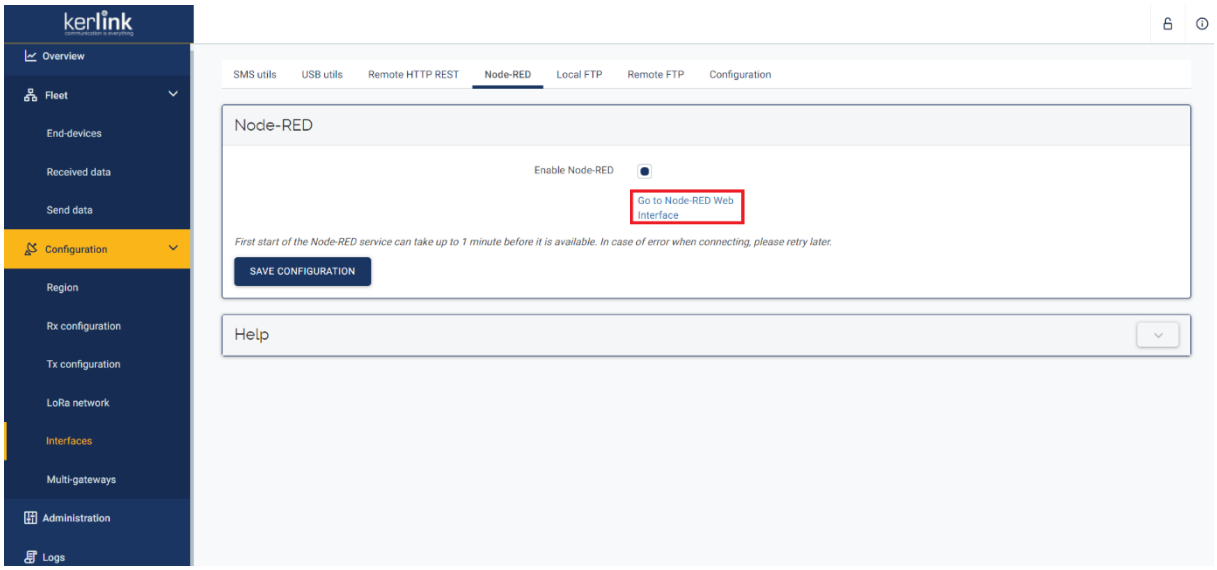

Figure 8 : Node-RED launcher

The following configuration allows the data received by the sensors to be sent directly into PcVue.

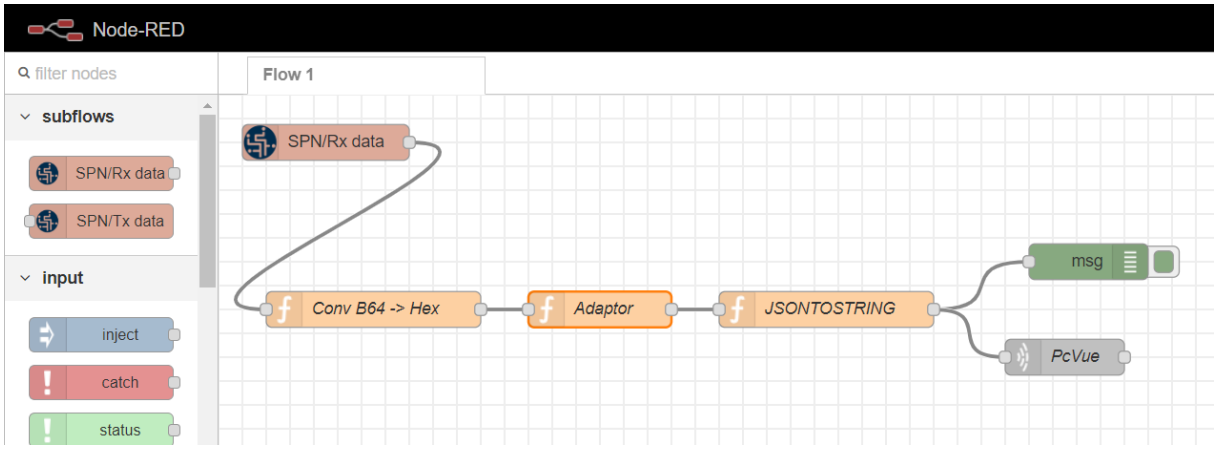

Figure 9 : Node-RED configuration

To start this configuration, take "SPN/Rx data" node on the left, 3 "function" nodes, 1 "output TCP" node, and one "output debug" node. And connect as shown in Figure 9.

"SPN/Rx data" is an input node for Gateway communication, i.e. each time a LoRaWAN frame is received by the Gateway a signal is sent to the next node.

4.1.1. Details "Conv B64 -> Hex" node

This function node allows you to do whatever you want with a script. Here we convert the payload initially received in base 64 into hexadecimal.

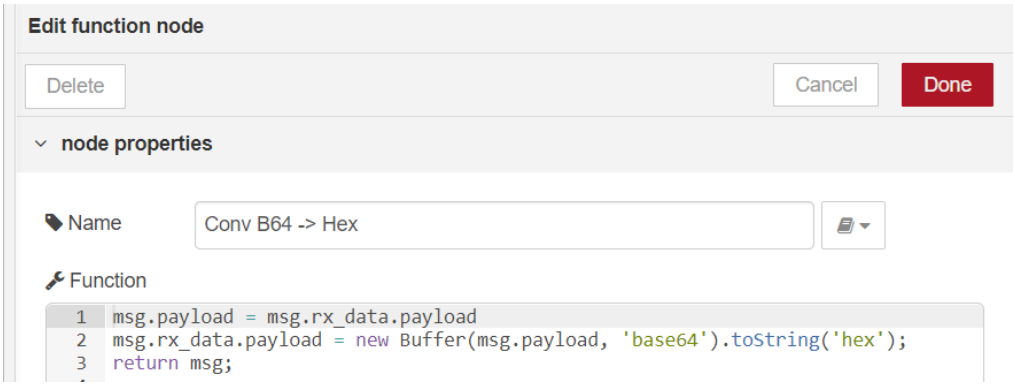

Figure 10 : "Conv B64 -> Hex" node

4.1.2. Details “Adaptor” node

This second function node is used to adapt the format of the id received by the gateway to the format requested by PcVue. Moreover, it transforms the hexadecimal payload written as a string into an array of decimal digits.

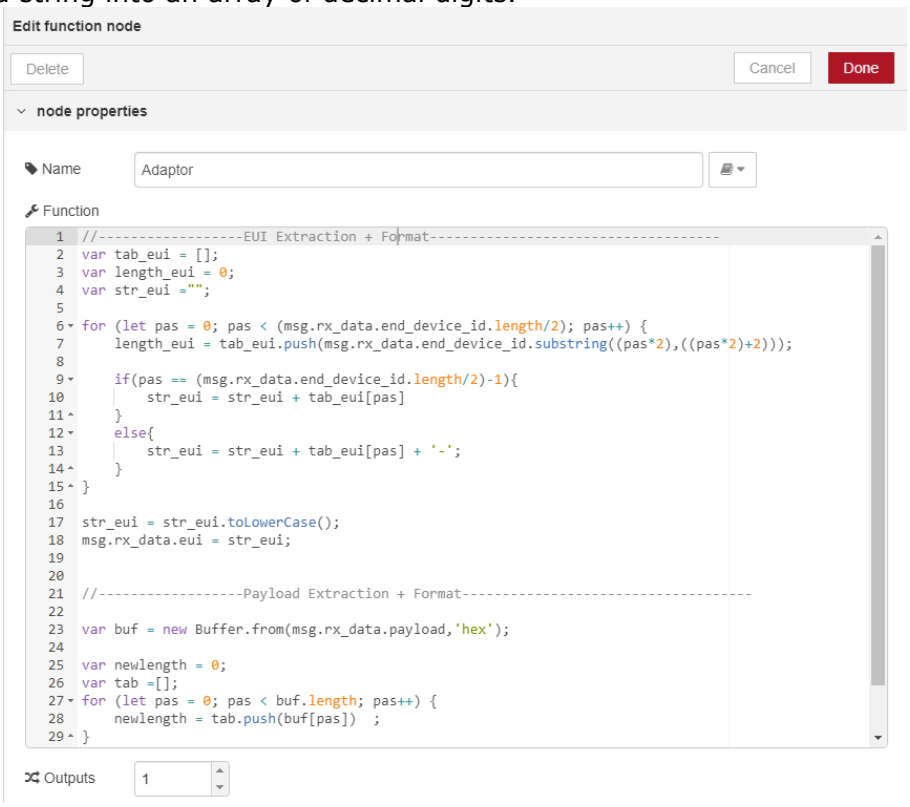

Figure 11 : "Adaptor" node

4.1.3. Details “JSONTOSTRING” node

This third and last function node transforms a JSON into text to be compatible with PcVue.

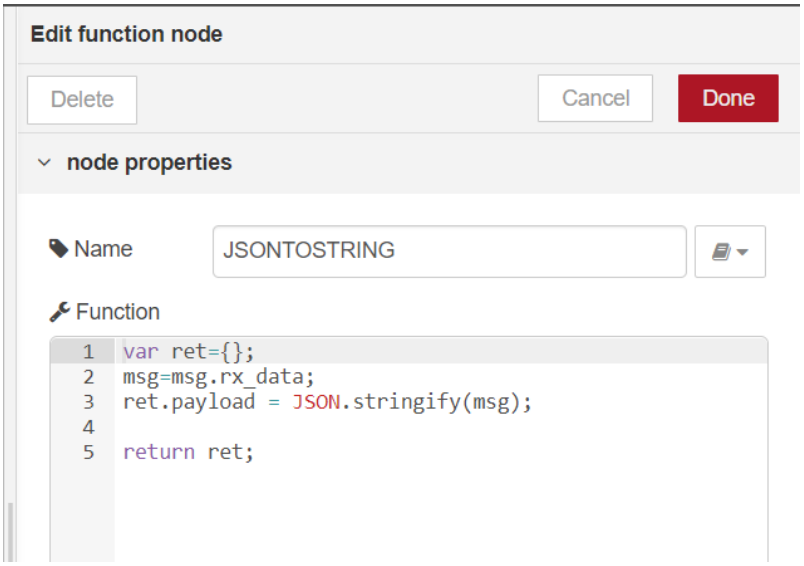

Figure 12 : "JSONTOSTRING" node

4.1.4. Details “PcVue” node

The "PcVue" node allows you to know the destination station. It is enough to inform him the IP (here X.X.X.X) and the port (2055 for the LoRaWAN communication in PcVue).

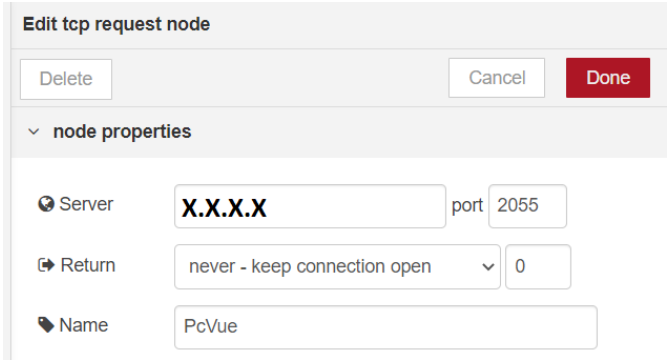

Figure 13 : "PcVue" node

4.1.5. Details “msg” node

This node makes it possible to observe what happens in the Node-RED debug tab.

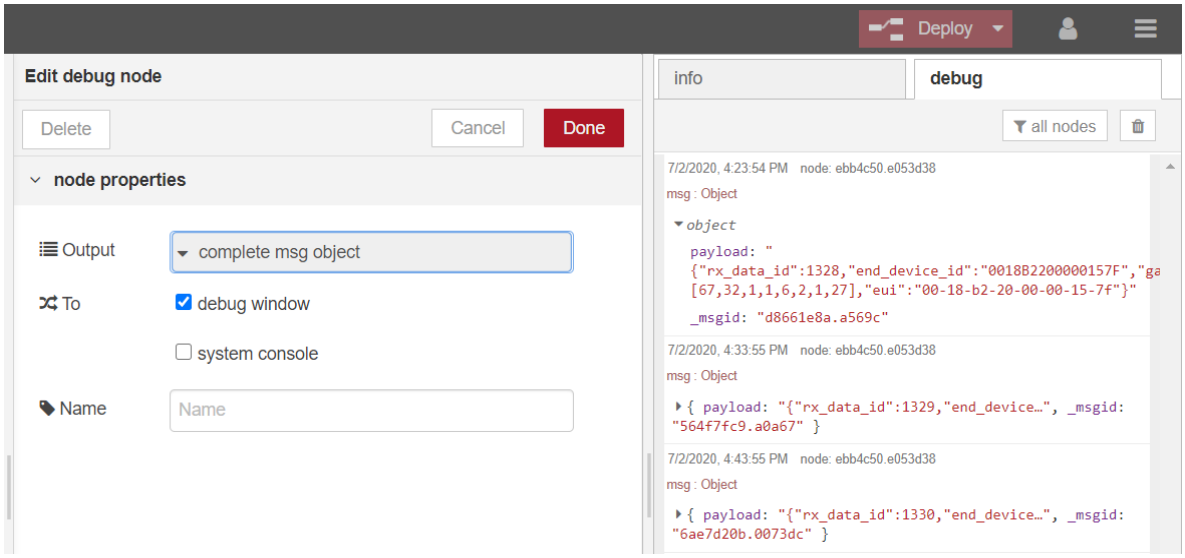

Figure 14 : "msg" node with debug display

Below is a typical message received by the debug window:

```
payload: "{"rx_data_id":1336,"end_device_id":"0018B2200000157F","gateway_id":"7076FF005605
019C","received_time":1593704635,"sequence_number":194,"port":1,"radio_id":0,"channel":5,"
rssi_dbm":-
34,"snr_db":10.2,"frequency_hz":868100000,"modulation_type":"LoRa","data_rate":"SF7BW125",
"coding_rate":"4/5","adr":true,"payload":[67,32,1,1,6,2,1,20],"eui":"00-18-b2-20-00-00-15-
7f"}"
_msgid: "5da6e3fd.c6d87c"
```


Figure 15 : typical debug message

4.1.6. Import & Deploy

4.1.6.1. Import

Below is the complete Node-RED clipboard that can be imported. Don't forget to change the IP address (in bold red).


```
[{"id":"a181cb8.ab12038","type":"subflow","name":"SPN/Rx data","info":"Provides Rx data in **JSON format**.\n\n# Outputs\n\n - **msg.rx_data** _JSON object_ Rx
data:\n  - **msg.rx_data.rx_data_id** _number_ Rx data unique identifier, auto incremented\n  - **msg.rx_data.end_device_id** _string_ Device EUI (OTAA) or Device
Address (ABP). Hexadecimal value, upper case\n  - **msg.rx_data.received_time** _number_ Reception time, Unix timestamp\n  - **msg.rx_data.sequence_number**
_number_ Sequence number\n  - **msg.rx_data.port** _number_ Port in [0;224]\n  - **msg.rx_data.radio_id** _number_ Radio identifier in [0;1]\n  -
**msg.rx_data.channel** _number_ Channel identifier in [0;9]. '0' to '7' are the 8 LoRa multi SF channels, '8' is the LoRa mono SF channel, '9' is the FSK channel (RFU)\n  -
**msg.rx_data.snr_db** _number_ LoRa SNR ratio in dB\n  - **msg.rx_data.rssi_dbm** _number_ RSSI in dBm\n  - **msg.rx_data.frequency_hz** _number_ Rx central
frequency in Hz\n  - **msg.rx_data.modulation_type** _string_ Modulation type 'LoRa' or 'FSK'\n  - **msg.rx_data.data_rate** _string_ If 'LoRa' modulation, 'SFxBWy'
format is used where 'x' is spreading factor in [7;12] and 'y' bandwidth in kHz. If 'FSK' modulation, FSK bitrate in bps.\n  - **msg.rx_data.coding_rate** _string_ ECC coding
rate (not available with 'FSK' modulation). Always '4/5'\n  - **msg.rx_data.adr** _boolean_ Adaptive data rate\n  - **msg.rx_data.payload** _string_ Payload data. Binary,
base64 encoded\n\n# Details\n\nRx data are automatically provided each time a Rx data is received by
SPN.","in":[],"out":[{"x":816,"y":204,"wires":[{"id":"9a14deb5.f354f","port":0}]}]},{"id":"d09be48c.0ceaf8","type":"zeromq
in","z":"a181cb8.ab12038","name":"ZMQUT/uplink","topic":"rx_data","fields":"rx_data","server":"ipc:///var/run/zmqut/uplink","output":"string","isserver":false,"intype":"su
b","x":173,"y":204,"wires":[["1049eec0.6e6e61"]]},{"id":"1049eec0.6e6e61","type":"change","z":"a181cb8.ab12038","name":"Remove rx_data
topic","rules":[{"t":"set","p":"rx_data","pt":"msg","to":"$substring(rx_data,
7)","tot":"jsonata"}],"action":"","property":"","from":"","to":"","reg":false,"x":383,"y":204,"wires":[["9a14deb5.f354f"]]},{"id":"9a14deb5.f354f","type":"json","z":"a181cb
8.ab12038","name":"Convert to JSON
object","property":"rx_data","action":"obj","pretty":false,"x":626,"y":204,"wires":[[]]},{"id":"df74940e.7e62c8","type":"subflow:a181cb8.ab12038","z":"ff9d593c.3844b8",
"name":"","x":210,"y":220,"wires":[["b9ec5e44.b419f"]]},{"id":"44af849c.d1d7ec","type":"debug","z":"ff9d593c.3844b8","name":"","active":true,"tosidebar":true,"console
":false,"tostatus":false,"complete":"true","x":830,"y":320,"wires":[]},{"id":"b9ec5e44.b419f","type":"function","z":"ff9d593c.3844b8","name":"Conv B64 ->
Hex","func":"msg.payload = msg.rx_data.payload\nmsg.rx_data.payload = new Buffer(msg.payload, 'base64').toString('hex');\nreturn
msg;\n","outputs":1,"noerr":0,"x":210,"y":360,"wires":[["b71b667.48abe98"]]},{"id":"81b7fafa.670558","type":"function","z":"ff9d593c.3844b8","name":"JSONTOSTRING
","func":"var ret={};\nmsg=msg.rx_data;\nret.payload = JSON.stringify(msg);\n\nreturn
ret;","outputs":1,"noerr":0,"x":630,"y":360,"wires":[["44af849c.d1d7ec","e33a10d4.14f84","a90f56d3.ce20f8"]]},{"id":"b71b667.48abe98","type":"function","z":"ff9d593c.
3844b8","name":"Adaptor","func":"//----------------EUI Extraction + Format----------------------------------\nvar tab_eui = [];\nvar length_eui = 0;\nvar str_eui
=\"\";\n\nfor (let pas = 0; pas < (msg.rx_data.end_device_id.length/2); pas++) {\n length_eui =
tab_eui.push(msg.rx_data.end_device_id.substring((pas*2),((pas*2)+2)));\n \n if(pas == (msg.rx_data.end_device_id.length/2)-1){\n str_eui = str_eui +
tab_eui[pas]\n }\n else{\n str_eui = str_eui + tab_eui[pas] + '-';\n }\n}\n\nstr_eui = str_eui.toLowerCase();\nmsg.rx_data.eui = str_eui;\n\n\n//----------
---Payload Extraction + Format----------------------------------\n\nvar buf = new Buffer.from(msg.rx_data.payload,'hex');\n\nvar newlength = 0;\nvar tab =[];\nfor (let pas
= 0; pas < buf.length; pas++) {\n newlength = tab.push(buf[pas]) ;\n}\nmsg.rx_data.payload = tab;\n\nreturn
msg;","outputs":1,"noerr":0,"x":400,"y":360,"wires":[["81b7fafa.670558"]]},{"id":"e33a10d4.14f84","type":"tcp
request","z":"ff9d593c.3844b8","server":"X.X.X.X","port":"2055","out":"sit","splitc":"0","name":"PcVue","x":830,"y":400,"wires":[[]]}]
```


Figure 16 : clipboard text

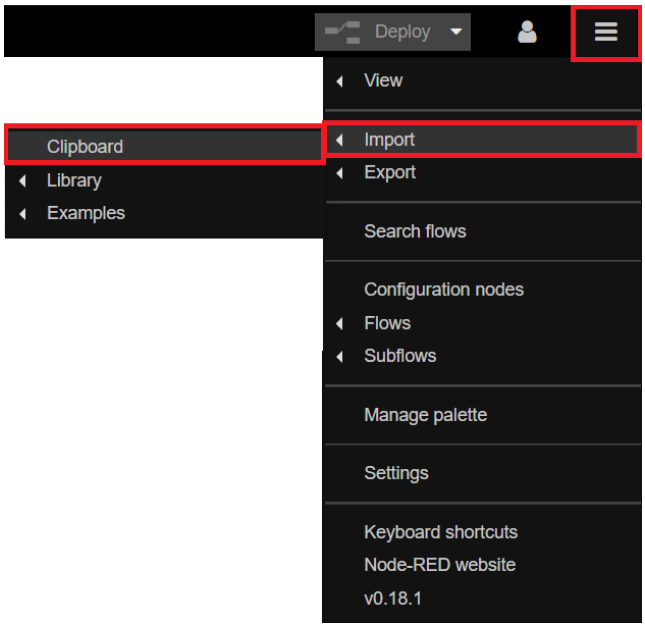

Figure 17 : Import procedure

To import, click on the 3 horizontal bars at the top right. Select Import then Clipboard.
A window opens, it is possible to paste the import text of a clipboard containing all the nodes and their parameters (see figure 18).

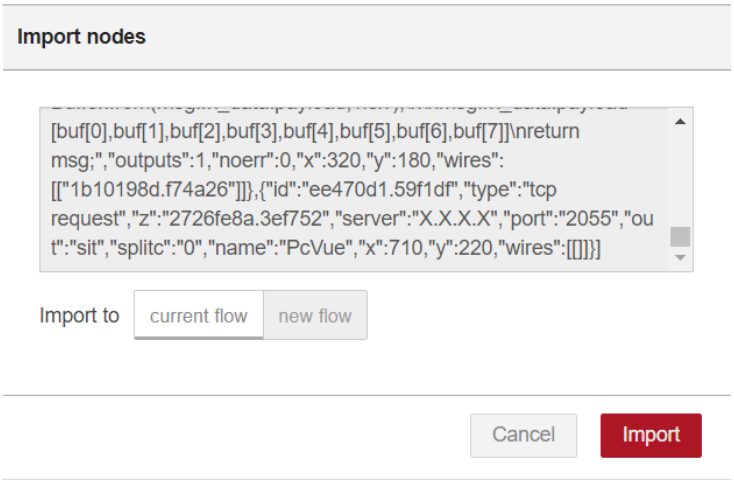

Figure 18 : Clipboard import window

4.1.6.2. Deploy

Once finished, don't forget to deploy the configuration by right-clicking on deploy in the top right corner.

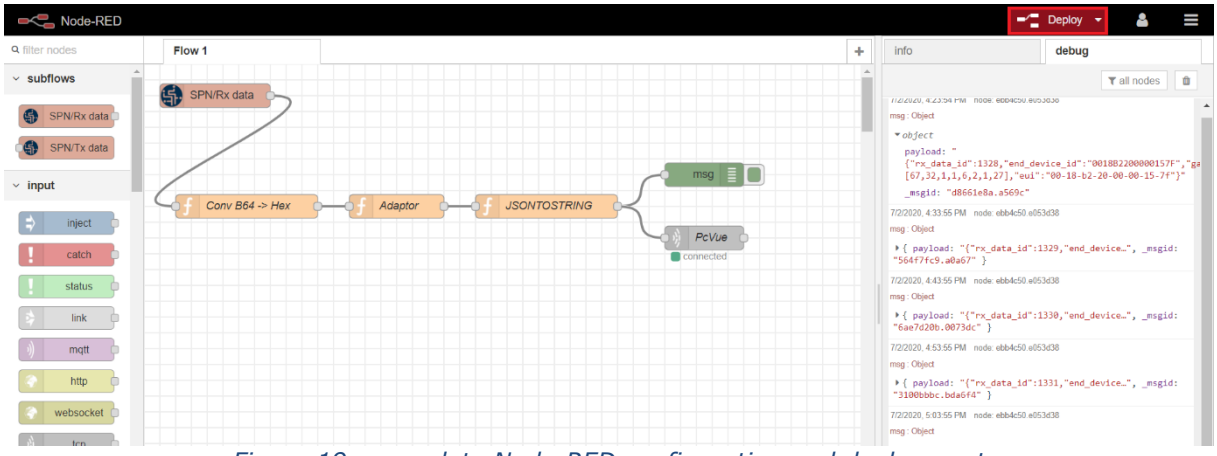

Figure 19 : complete Node-RED configuration and deployment

4.2. Writing configuration

To send a PcVue command to a sensor, Node-Red is configured with three nodes. A TCP node as input with your PcVue IP address, a Function node to convert the data to JSON format, the second Function node to send the data in the correct format, and a SPN/Tx_data for LoRa output.

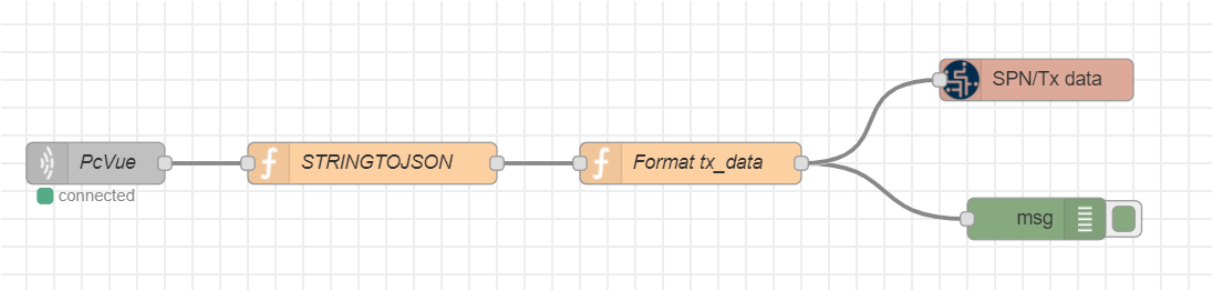

Figure 20 : Node-Red LoRa writing configuration

To start this configuration, take a "TCP in" node, a "function" nodes, an "output LoRa" node, and an "output debug" node, then connect as shown in Figure 23.

4.2.1. Details "PcVue" node

The "PcVue" node allows you to know the source station. Give the PcVue IP (here X.X.X.X) and the port (2055 for the LoRaWAN communication in PcVue).

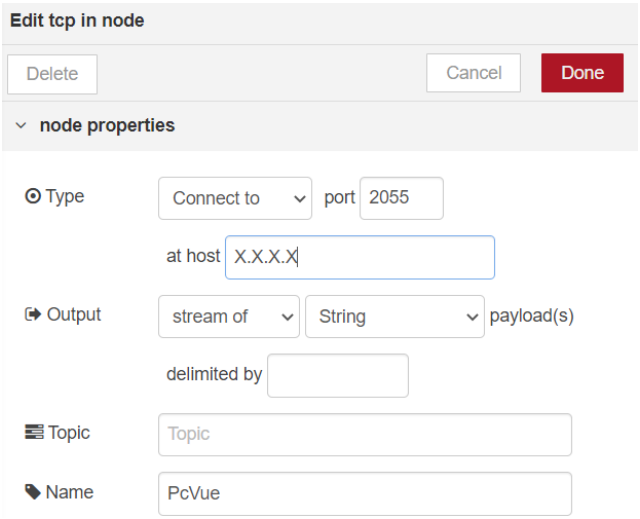

Figure 21 : "PcVue" node configuration

Check the type: "**Connect to**"
and the output: stream of **String**

4.2.2. Details “STRINGTOJSON” node

This second node: function. Transforms a String into JSON to be compatible with PcVue.

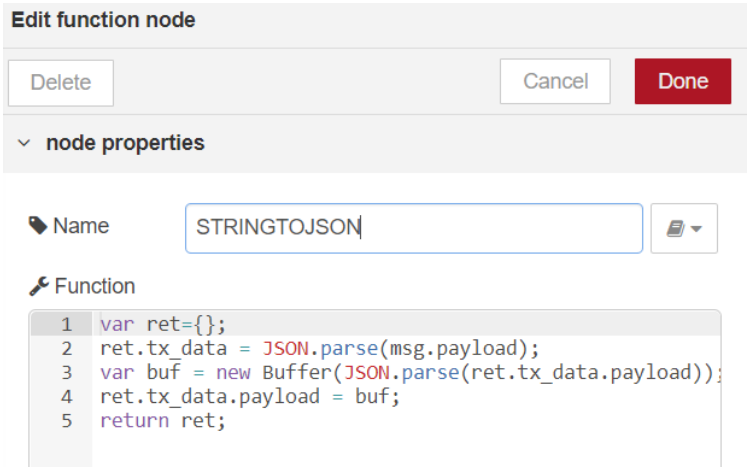

Figure 22 : "STRINGTOJSON" node configuration

4.2.3. Details “Format tx_data”

This second function node, format the payload to be compatible with kerlink data sending.

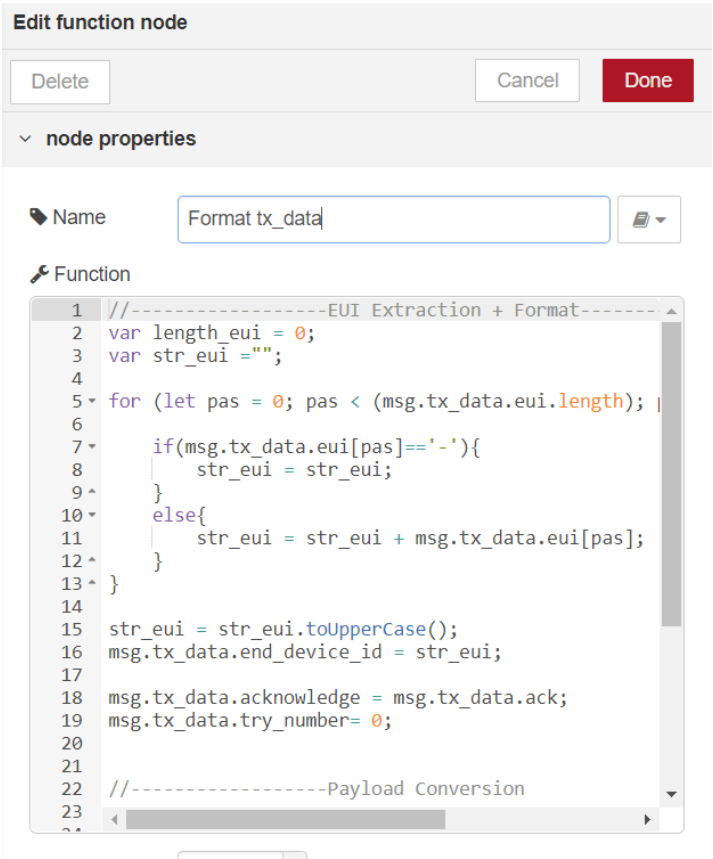

Figure 23 : "format tx_data" node configuration

4.2.4. Details “msg” and SPN/Tx data nodes

This node makes it possible to observe what happens in the right part of Node-RED.

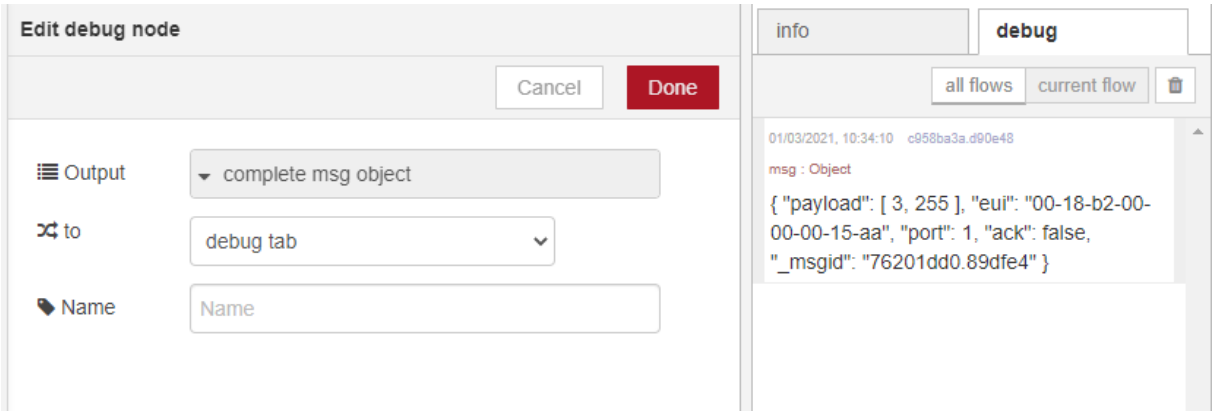

Figure 24 : "msg" node configuration with debug display

Below is a typical message received by the debug window:

```
{ "payload": [ 3, 255 ], "eui": "0018B200000015AA", "port": 1, "ack": false, "_msgid": "76201dd0.89dfe4" }
```

Figure 25 : Typical LoRa writing debug message

To end, put the node SPN/Tx data and link it with “Format tx_data” node like figure 20.

4.2.5. Import

Below is the complete Node-RED clipboard that can be imported. Don't forget to change the IP address (in bold red).

```
[{"id":"410c622e.15f15c","type":"subflow","name":"SPN/Tx data","info":"Adds Tx data in **JSON format**.\n\n# Inputs\n\n - **msg.tx_data** _JSON object_ Tx data:\n - **msg.tx_data.end_device_id** _string_ Device EUI (OTAA, 16 digits) or Device Address (ABP, 8 digits). Hexadecimal value, upper case\n - **msg.tx_data.port** _number_ Port\n - **msg.tx_data.acknowledge** _boolean_ true: acknowledged data ; false: not acknowledged data\n - **msg.tx_data.try_number** _number_ Number of retry before aborting\n - **msg.tx_data.payload** _string_ Payload data in base64. The payload length is limited to 222 bytes (296 characters in base 64)\n\n# Details\n\nTx data are added automatically to SPN.","in":[{"x":215,"y":85,"wires":[{"id":"490c33a2.988e9c"}]}],"out":[]},{"id":"c8fd8cc3.92bad","type":"zeromq out","z":"410c622e.15f15c","name":"ZMQUT/downlink","topic":"","fields":"tx_data","server":"ipc:///var/run/zmqut/downlink","isserver":true,"intype":"pub","x":824,"y":85,"wires":[]},{"id":"51013ee3.f086d",
```

Figure 26 : clipboard LoRa writing text

Refer to **4.1.6**. to import this clipboard and don't forget to deploy to confirm the changes.

5. PcVue Configuration

5.1. Project configuration

Add in the C directory of your PcVue project (if the project already exists) the file svmgrMultiTechLoRa.ini. It has been modified as follows:

```
[ConfigParameters]
AttributeNumber = 10
ListeningPort = 2055
NumberOfGateways = 1


[Gateway_1]
IpAddress = X.X.X.X
Name = GW1
```

Figure 27 : Project configuration

The IP address, the TCP port of the Gateway and an extended name and attribute that is not used must be entered.

We decide to use the attribute 10 which is not used, the (Gateway) IP address X.X.X.X with the port 2055 for the Gateway named GW1.

To map a LoRa variable in PcVue, the extended attributes must be configured with keywords found in the dll documentation.

5.2. Sensor Mapping

In the rest of this technical documentation, the example of the TEMP sensor from ADEUNIS will be taken.

4.1.6 Data Frame

This frame (0x43) contains the values measured on the different sensors.

0	1	2	3	4	5	6	7	8	9	10
Code	Status					PAYLOAD				
0x43	Cf Status	internal sensor identifier	Value read on the internal sensor		External sensor identifier	Value read on the internal sensor				X
0x43	0xA2	0xD1	0x015E		0x82	0xFF06				

Figure 28 : Sensor data sheet

With the help of the documentation, the external temperature of the sensor can be found in bytes 6 and 7, so the following key is entered in attribute 10 in register variable.

Text attribute 10

LORA#GW1#00-18-B2-20-00-00-15-7F#payload#6:2#0:0x43

Figure 29 : Extended attribute

As see figure 22, the beginning of the filter expression LoRaWAN is:

"#LORA#***YourGatewayName***#***YourSensorEUI***"

The next part of the filter corresponds to the byte(s) of payload data to be retrieved and associated with a variable.

"#payload#x:y"

To finish the filter, it is necessary to indicate which frame to retrieve in order to look at the beginning of the frame (this is indicated on the data sheet).

"#**0:0x??**"

They are 3 different filters.

#x:y
This filter is to retrieve the **y** number byte from the byte number **x**.

#x.y
This filter is to retrieve the **y** bit of the byte number **x**.

#0:0x??
This filter is to retrieve the frame beginning with "0:0x??".

In this example, the register variable used, "@KerlinkGateway.Temp.Tout", receives the data sent by the sensor. During this time, a division by 10 must be made to obtain the temperature. To get done, an expression on variable must be created.

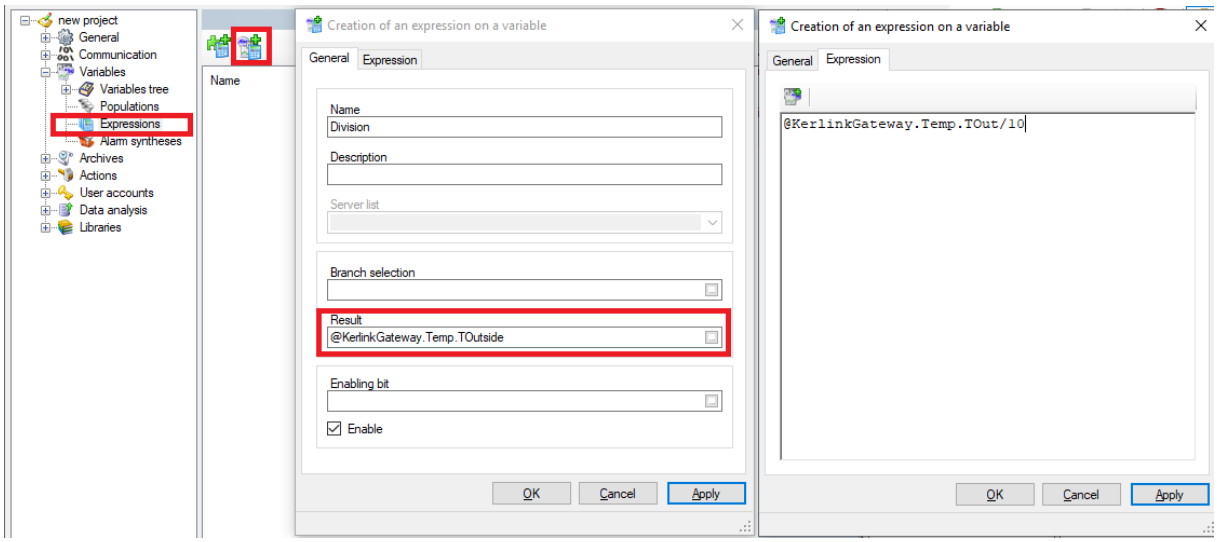

Figure 30 : Expression to obtain the temperature

Go to Expressions, click on "add an expression on variable ..." then configure the window as follows:

Give the expression the name you want, enter the result variable, and enter the following expression:

@KerlinkGateway.Temp.Tout/10

Simply create a graphical object with animation to display the temperature obtained in the variable: "@KerlinkGateway.Temp.TOutside"

Name	Description	Value	Timestamp	Quality
TOut		262.00	03/07/20 11:20:34.336 : PLC	
TOutside		26.20	03/07/20 11:20:34.336 : PLC	

Figure 31 : Results

5.3. Writing sensor value

For this part we will take the example of the DRYCONTACT sensor v2 from Adeunis.

4.2.3 Output control frame

This frame allows us to change the state of the Channels configured as outputs.

0	1	2	3	4	5	6	7
Code	PAYLOAD						
0x03	output status	X	X	X	X	X	X
0x03	0x05	X	X	X	X	X	X

Description of the frame:

- Byte 1: States to be applied to the Channels
  - Bit 0: State to be applied to Channel1
  - Bit 1: State to be applied to Channel2
  - Bit 2: State to be applied to Channel3
  - Bit 3: State to be applied to Channel4
- Bits 4 to 7: reserved

If the channel is configured as an output, the corresponding bit in the downlink frame 0x03 is taken into account, otherwise it is simply ignored.

Figure 32 : DRYCONTACT sensor data sheet

With the help of the documentation, the control bits can be found in byte 1, so that the following key is entered in attribute 10 of the register variable

Text attribute 10

LORA#GW2#00-18-b2-00-00-00-15-aa

Figure 33 : Extended attribute for DRYCONTACT sensor

Just 3 attributes are used and the variable configuration is done:

LORA#YourGatewayName#YourSensorEUI

The message sent by the supervisor is JSON formatted as shown below:

{ "payload": [3, 5], "eui": "xx-xx-xx-xx-xx-xx-xx", "port": 2, "ack": false, "_msgid": "a403987c.5bfc68" }

The value sent to the variable must follow the following format:

Payload | Port | Ack

For instance, if the variable is set with the value: [3, 5] | 2 | false the message sent will be the one just above.

If the value sent to the variable don't respect the given format, the message sent to the gateway will set to "Unknown" the port and ack.

Please note that the variable must be a **text** set as "command".

6. Annexes

6.1. Reset Procedure

- Stop the gateway by pressing the ON/OFF button for more than 5 seconds.
- Make sure the power light is off and release the button.
- Press and hold the ON/OFF button to turn on the gateway and hold it down until the power and status lights flash alternately.
- At this point, the gateway is ready to perform a system restore operation. It waits for 10 seconds to confirm.
- Release the button and press again to confirm the operation (the lights will stop flashing alternately).
- Wait for the end of this operation (LED behaviour). This takes about 2 minutes.

7. Glossary

SPN:	Small Private Network. Web interface of the network generated by the gateway.
Node-RED:	flow-based development tool for visual programming. It's integrated into the gateway.
Payload:	is the part of transmitted data that is the actual intended message.
LoRa:	(Long Range) Technology based on a compromise between signal range, power consumption and data security. This technology is governed by an alliance.
LoRaWAN:	Radio-based communication protocol.
TCP:	Transmission Control Protocol is a transmission protocol used on IP networks.
KERLINK:	Gateway builder.

8. Conclusion

In this document the following points have been dealt with:

SPN interface
Node-RED interface
PcVue configuration
LoRaWAN data acquisition in PcVue