LoRa TTN gateway with Orange Pi Zero and RAK831.

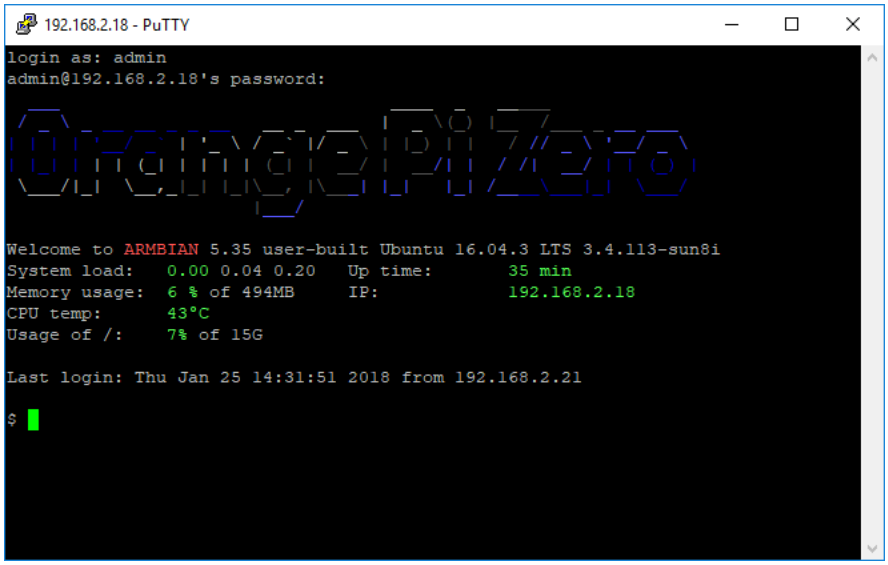
1 Introduction.

This document describes how to build a LoRaWan gateway. For this gateway a RAK831 gateway module and an Orange Pi Zero (the so-called "Backhaul") is used. Info about the RAK31 can be found at *http://www.rakwireless.com*. The RAK831 is compatible to the iC880A-SPI.

The bash-commands in this document start with "#" of een "$" to show if a command must be given as user "root" or as a common user. Do not include this character in the commands.

2 Orange Pi Zero configuration.

For this project an Orange Pi Zero is used.
Other Ornage Pi models are possible, but it is vital that an (extra) SPI connection is available on the connector. The Orange Pi PC Plus lacks this feature.
The used Linux distribution is Armbian 5.35 (see *http://www.armbian.com/orange-pi-zero).* For installation of Armbian an SD card (16 GB) must be prepared. With the SD card the Orange PI Zero can be booted up.
Connect the OPi through an ethernet cable with your local network.
Now the Orange Pi can be connceted through SSH (I used puTTY fort his).
At first you may login as user "root" with password "1234". The password must be changed immediately. Add a user "admin" (with "useradd"), enter a password fort his user and see that it gets "/bin/bash" for the shell.
Update the software with "apt-get update" en "apt-get upgrade". If you log in as "admin", you will see a screen like this:



2.1 Enable WiFi.

For your convenience enable WiFi with the command "nmtui". Here you can also set the hostname, for instance "GwLoRa", but that can also wait untill you run the gateway installscript.

2.2 Timezone.

To show the right clocktime, you have to set the time zone. For the Netherlands (for example) you have to give the following commands as "root":

```
# rm /etc/localtime
# ln -s /usr/share/zoneinfo/Europe/Amsterdam /etc/localtime
```

2.3 Enable SPI.

SPI is normally enabled. Check if /dev/spidev1.0 exists. Spidev0.0 is used for the on-board flash.
You may check the SPI setting with the command:

```
# bin2fex /boot/script.bin | less
```

Look for the paragraph with "[spi1]". You will see the GPIO numbers that are used for SPI: cs0 = PA13, sclk=PA14, mosi=PA15, miso=PA16. You may check the SPI by connecting a LED (in series with a 1 kΩ resistor) at pin 23 (SCLK) or pin 19 (MOSI). Use the command:
cat /dev/random >/dev/spidev1.0
The LED will light and show som flickering. Abort with Ctrl-C.

2.4 Enable GPIO.

Use the commands:

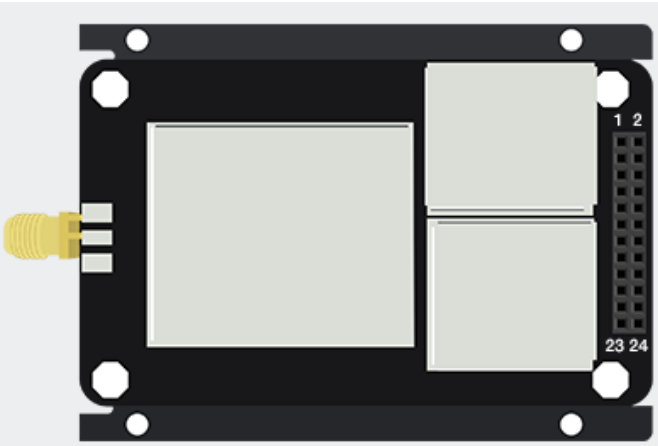
```
# modprobe gpio-sunxi
# echo "gpio-sunxi" >> /etc/modules
```

to make sure that the gpio module will be loaded (also after the next restart).
Now it should be possible to control for instance GPIO10 (pin 26) with the commands:

```
# echo 10  > /sys/class/gpio/export
# echo out > /sys/class/gpio/gpio10/direction
# echo 1   > /sys/class/gpio/gpio10/value
# echo 0   > /sys/class/gpio/gpio10/value
```

3 RAK831 connections.

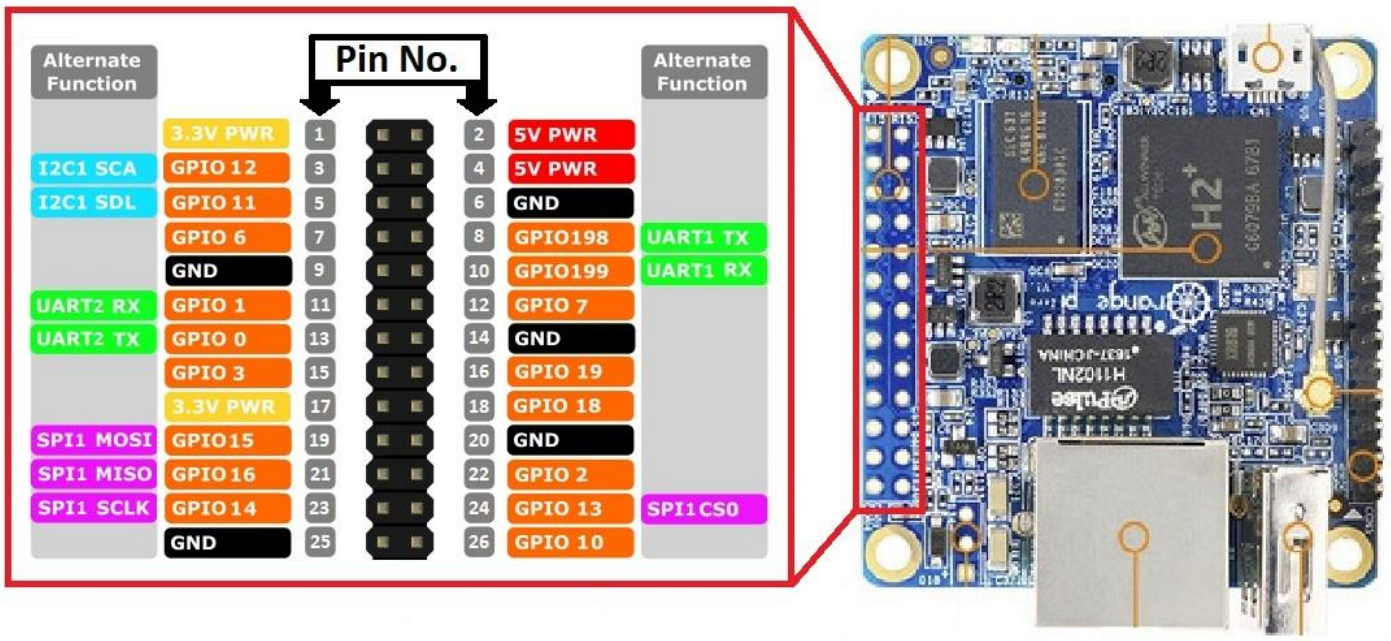
The lay-out of the RAK831 connector looks like this:



+5V	1	+5V	2
GND	3	LNA_EN_A	4
GND	5	GND	6
RADIO_EN_A	7	PA_G8	8
RADIO_EN_B	9	PA_G16	10
PA_EN_A	11	GND	12
RADIO_RST	13	GND	14
CSN	15	MOSI	16
MISO	17	SCK	18
RESET	19	GPIO0	20
GPIO1	21	GPIO2	22
GPIO3	23	GPIO4	24

3.1 Orange Pi Zero I/O pins.

The lay-out of the Orange Pi connector looks like this:

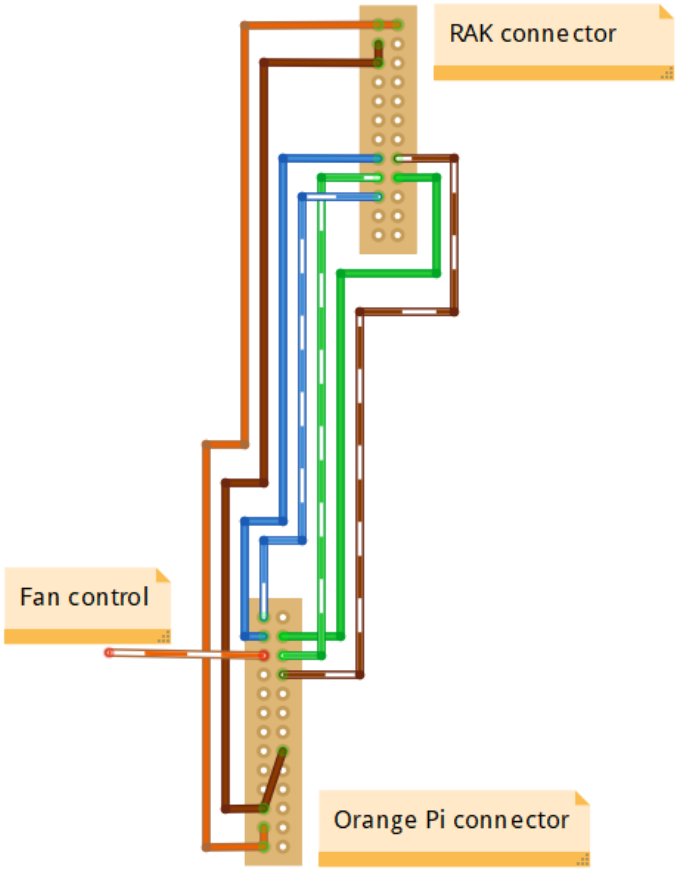


3.2 Connecting the RAK and the Pi.

The powersupply (5 Volt and ground) is connected to pins 2 and 6 of the Orange Pi respectively. The micro USB connector will not be used. For the connections between the RAK831 en Orange Pi, a cable is prepared with the following wiring:

RAK831 pin	Description	Orange Pi pin	Description	Color
1,2	+5 Volt	2, 4		Orange
3,5	Ground	6,9		Brown
15	CSN (chip select)	24	GPIO13 - SPI1 CS0	Blue
18	SCP (SPI clock)	23	GPIO14 – SPI1 SCLK	Green
16	MOSI	19	GPIO15 – SPI1 MOSI	Brown-white
17	MISO	21	GPIO16 – SPI1 MISO	Green-white
19	RST (Reset)	26	GPIO10	Blue-white
		22	GPIO2 - Fan control	Orange-white

The cable looks like this:



The 24 pin RAK connector is “male”, the 26 pin Pi conncetor is “female”.

When the cable is connected and the powersupply is switched on, a red LED will light near pin 3 of the RAK831 header. For a picture of the cable, see the pictures at thye end of this document.

4 Install the gateway software.

For this the installer from ttn-zh can be used. This installer is meant for a iC880A, but it will also work for a RAK831. During the installation the latitude en longitude of the position of the gateway must be entered. These numbers can be found at *https://www.latlong.net*. For example: N = 51.65571 en E = 5.037537. These numbers are stored in /opt/ttn-gateway/bin/local_conf.json after installation.
Install the softwaer as follows:

```
# git clone https://github.com/ttn-zh/ic880a-gateway.git /root/gateway
```

The install script is now in /root/gateway. Go tot his directory:

```
# cd /root/gateway
```

Start the script with:

```
# ./install.sh spi
```

Choose "N" as the answer to the question "Do you want to use remote settings file?". During the script you have to fill in some more data. At the end of the installscript the next info will be showed (example):

```
Gateway configuration:
Detected EUI D29076FFFEB8D554 from eth0
Do you want to use remote settings file? [y/N] N
       Host name [ttn-gateway]:GwLoRa
       Descriptive name [ttn-ic880a]:ttn-RAK831
       Contact email: info@example.com
       Latitude [0]: 51.7557100
       Longitude [0]: 5.1375370
       Altitude [0]: 6
```

The gateway EUI is needed later on to register the gateway to TTN. The Orange Pi will reboot automatically. However it will not work in the current configuration. Read on.

The file /opt/ttn-gateway/bin/local_conf.json contains now for instance:

```
{
        "gateway_conf": {
                "gateway_ID": " D29076FFFEA3E4D554",
                "servers": [ { "server_address": "router.eu.thethings.network",
                "serv_port_up": 1700, "serv_port_down": 1700, "serv_enabled": true } ],
                "ref_latitude": 51.7557100,
                "ref_longitude": 5.1375370,
                "ref_altitude": 6,
                "contact_email": "info@example.com",
                "description": "ttn-RAK831"
        }
}
```

4.1 Reset signal for the RAK831.

After installation and reboot, the software will be started automatically through /lib/systemd/system/ttn-gateway.service.
The command:

```
# systemctl | grep ttn
Or
# service ttn-gateway status
```

Will show the running service.
Stop the service witht:

```
# service ttn-gateway stop
```

The gateway needs to be reset before it is started. The reset line should stay low under normal circumstances. During reset the line the line must be pulsed with a short positive signal. For this you have toe dit the script at /opt/ttn-gateway/bin/start.sh. AT the beginning of this script you will find some line to generate a reset on pin 25. Pin 25 does not exist fort the Orange Pi. Change the pin number from "25" to "10".

4.2 Correction SPI device.

The communication with the RAK831 uses “native” SPI. However, in the standard setting, the wrong SPI device is used, “spidev0.0” instead of “spidev1.0”. That is why the service “ttn-gateway” will not start properly. In the logging at /var/log/syslog you will find the famous messages:

```
Feb 26 11:28:27 GwLoRa ttn-gateway[6294]: INFO: [main] Starting the concentrator
Feb 26 11:28:27 GwLoRa ttn-gateway[6294]: ERROR: [main] failed -1 to start the concentrator
```

Edit the file “/opt/ttn-gateway/lora_gateway/libloragw/inc/imst_rpi.h” and replace “spidev0.0” door “spidev1.0”.
Then recompile the software. Use the next commands:

```
# cd /opt/ttn-gateway/lora_gateway
# make clean ; make
# cd ../packet_forwarder
# make clean ; make
```

Restart the service with:

```
# service ttn-gateway restart
```

Afer some time the “RX” LED will light at the RAK831.

5 Switch service on/off.

You may stop the gateway service with:

```
# service ttn-gateway stop
```

Switch on again with:

```
# service ttn-gateway start
```

6 If it doesn’t work…

If the gateway does not start, you will find in the logging: “ERROR: [main] failed to start the concentrator”. In that case the SPI bug is probably not functioning. First thing to do is: check the wiring.
/opt/ttn-gateway/lora_gateway/util_spi_stress/util_spi_stress is a utility to check the SPI bus.
When you start this, a message “ERROR: lgw_connect() did not return SUCCESS” may show. That points in the direction of an SPI bus problem.
lgw_connect.c can be found at /opt/ttn-gateway/lora_gateway/libloragw/src/loragw_reg.c.
The definition of the SPI device is in /opt/ttn-gateway/lora_gateway/libloragw/inc/imst_rpi.h. Change the definition of “SPI_DEV_PATH” to “/dev/spidev1.0” and recompile everything.
You may set all debug-options in ./lora_gateway/libloragw/library.cfg and recompile to more trouble shooting.

7 Temperatute control.

The temperature in degrees Celcius can be made visible by:

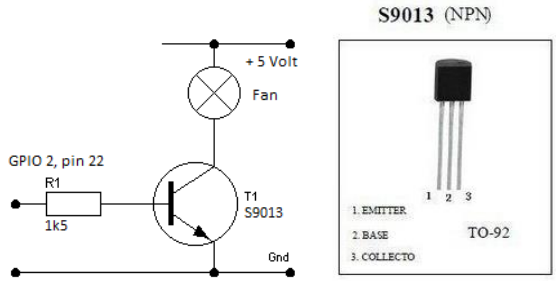
```
$ cat /etc/armbianmonitor/datasources/soctemp
```

More info is available with:

```
# armbianmonitor -m
```

This command is not very reliable. You may try the command with "-M". After several tries i twill work suddenly.

Then you see the CPU frequencye, the system load and the temperature. We can control the temperature by switching a small fan on and off. We use GPIO 2 for this function. The diagram is as follows:



In parallel of the transistor (collector en emitter) I added a 39 Ohm resistor in order to switch between low and high revolutions.

A script to control the fan looks like this:

```
#!/bin/bash
#
# Power-up the fan if temperature is too high.
# 27-02-2018, Ed Smallenburg
#

GPIO=2
TEMP=40

# Check if gpio is already exported
if [ ! -d /sys/class/gpio/gpio${GPIO} ]
then
  echo ${GPIO} > /sys/class/gpio/export
  sleep 1 ;# Short delay while GPIO permissions are set up
fi

# Set pin to OUTPUT
echo out > /sys/class/gpio/gpio${GPIO}/direction

while true; do
  if [[ $(cat /sys/class/thermal/thermal_zone0/temp) -gt ${TEMP} ]]; then
    echo 1 > /sys/class/gpio/gpio${GPIO}/value
  else
    echo 0 > /sys/class/gpio/gpio${GPIO}/value
  fi
  sleep 10
done
```

You may put this script at /usr/local/sbin/fancontrol.sh (chmod 700) and run it permanently by adding a configuration file at "/lib/systemd/system/fancontrol.service" with the following contents:

```
[Unit]
Description=Activate fan on high temperature

[Service]
WorkingDirectory=/usr/local/sbin
ExecStart=/usr/local/sbin/fancontrol.sh
SyslogIdentifier=fancontrol
Restart=on-failure
RestartSec=5

[Install]
WantedBy=multi-user.target
```

Then execute the following command:
systemctl enable fancontrol

8 Error logging.

In /var/log/syslog you can see the logging of the gateway. Relevant line contain the tekst “ttn-gateway”.

9 TCP logging.

Het commando:
```
# tcpdump -XUq port 1700
```
Can be used to inspect the traffic between the gateway and TTN.
If no node is active, you will see a status message to TTN every 30 seconds as well as a keep-alive message every 10 seconds.

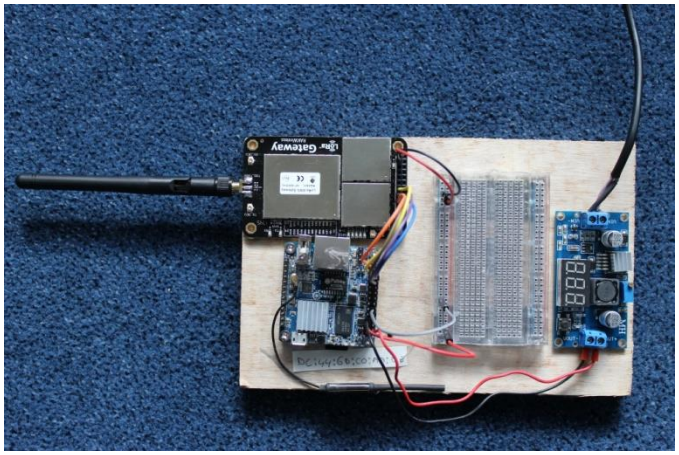
When an (OTAA) node is switched on, i twill do a JOIN REQUEST. That will activate a message of about 243 bytes. An ACK of 4 bytes will be returned. After 4 seconden TTN will return a JOIN ACCEPT of about 209 bytes.

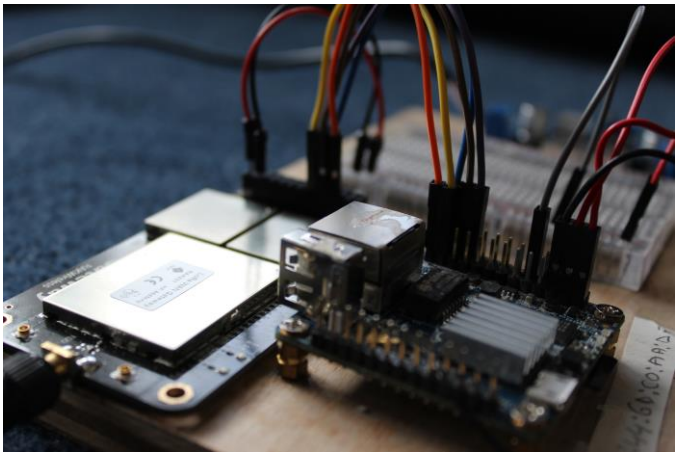
10 Issues.

Sometimes SSH login via WiFi fails. When you send a ping to the IP-adres of the gateway, it will take some time before the gateway reacts. Aftre that the login will succeed.

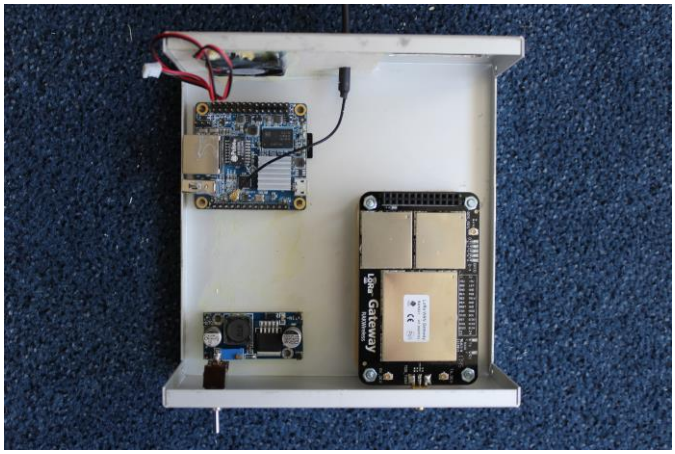
11 Pictures.

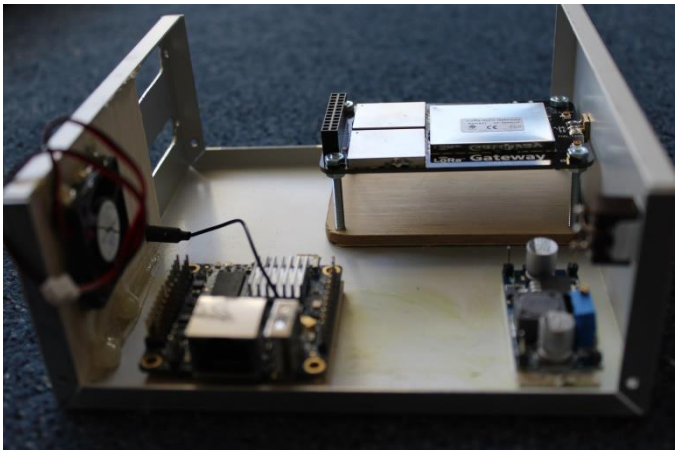
Test setting:



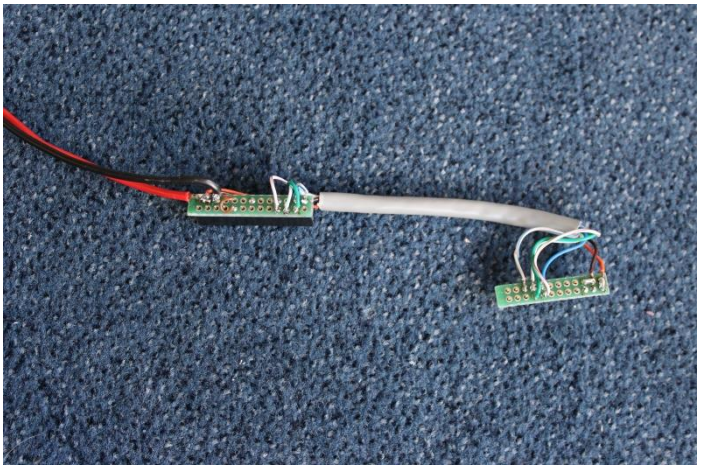


In a box:

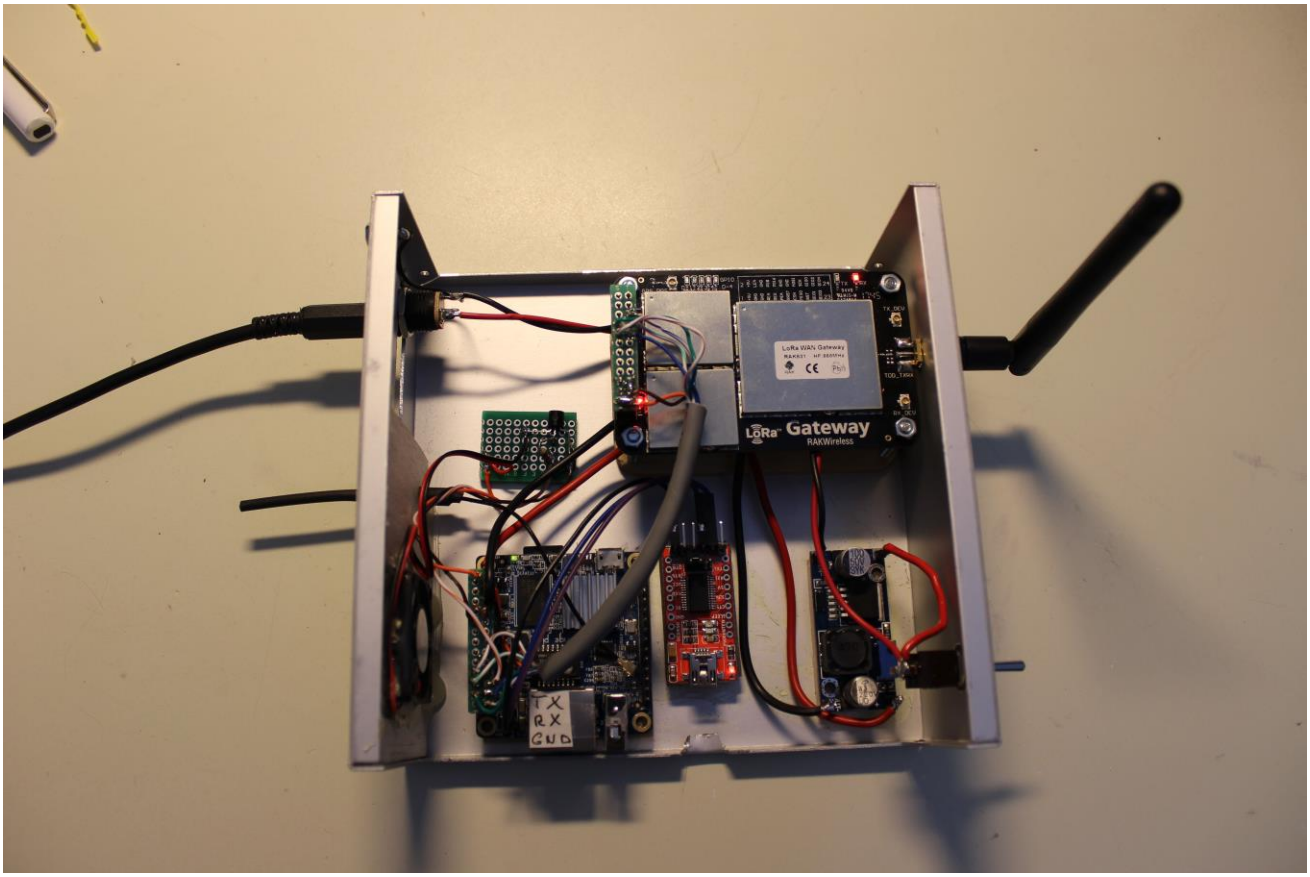


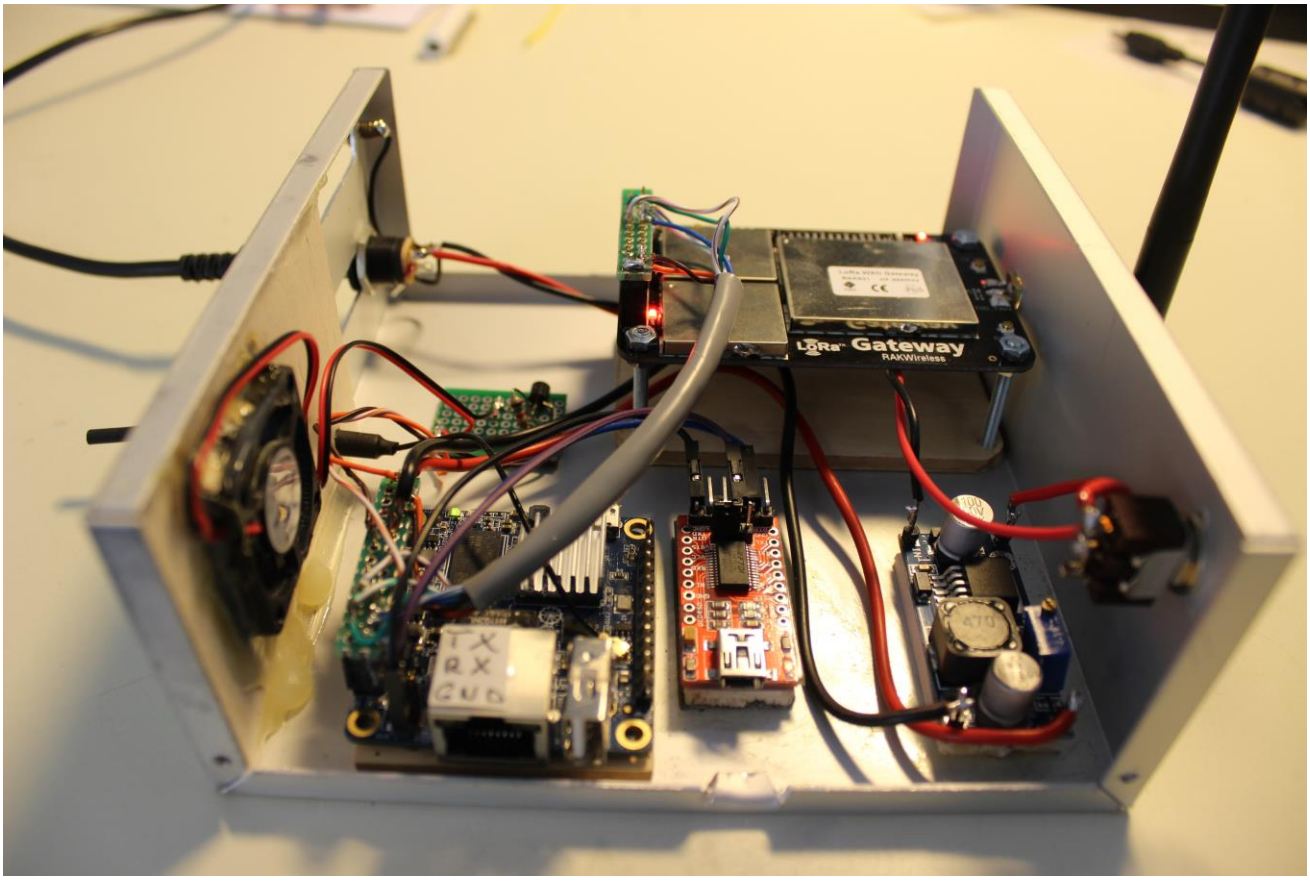


Cable between RAK831 and Orange Pi Zero:



Geheel bedraad:





The red module in the middle is used temporarily to make a connection to the serial port over USB.