

LORA / LORAWAN TUTORIAL 28

LoRaWAN Gateway



INTRO

- In this tutorial I will explain what the RAK831 Pilot Gateway is, how to configure it and how you can register this gateway with The Things Network.

This video is sponsored by

RAAK

IoT Made Easy

RAKWIRELESS TECHNOLOGY

- Rakwireless Technology (RAK) was established in June 2014 and is based in Shenzhen, China.
- RAK main focus is developing and supplying advanced IoT technology and services.
- Interesting links:
 - RAK website: <https://www.rakwireless.com>
 - RAK document section: <https://www.rakwireless.com/en/download>
 - RAK online shop: <https://www.aliexpress.com/store/2805180>
 - RAK support centre: <http://support.rakwireless.com>

PRESENTATIONS

- This presentation can be found at:
https://www.mobilefish.com/download/lora/lora_part28.pdf
- All my LoRa/LoRaWAN tutorials and presentations can be found at:
https://www.mobilefish.com/developer/lorawan/lorawan_quickguide_tutorial.html

RAK83 I Pilot Gateway Introduction

RAK831 PILOT GATEWAY

- There are many different gateways, one of them is the RAK831 Pilot Gateway.
- This gateway is designed for indoor usage and supports the LoRaWAN 1.0.2 protocol.
- It is ideal for prototyping, Proof-of-Concept demonstration or for evaluation purposes.
- The RAK831 Pilot Gateway is a fully assembled gateway with an aluminium housing and is shipped with a micro SD card with all software pre-installed.
- Instead of the RAK831 Pilot Gateway you can also buy a LoRaWAN Gateway Developer kit and assemble the gateway yourself but without the aluminium housing, thus reducing the price.
- The RAK831 Pilot Gateway can be purchased at the [RAK online shop](#).

RAK831 PILOT GATEWAY

- The RAK831 Pilot Gateway supports the following frequency ranges:

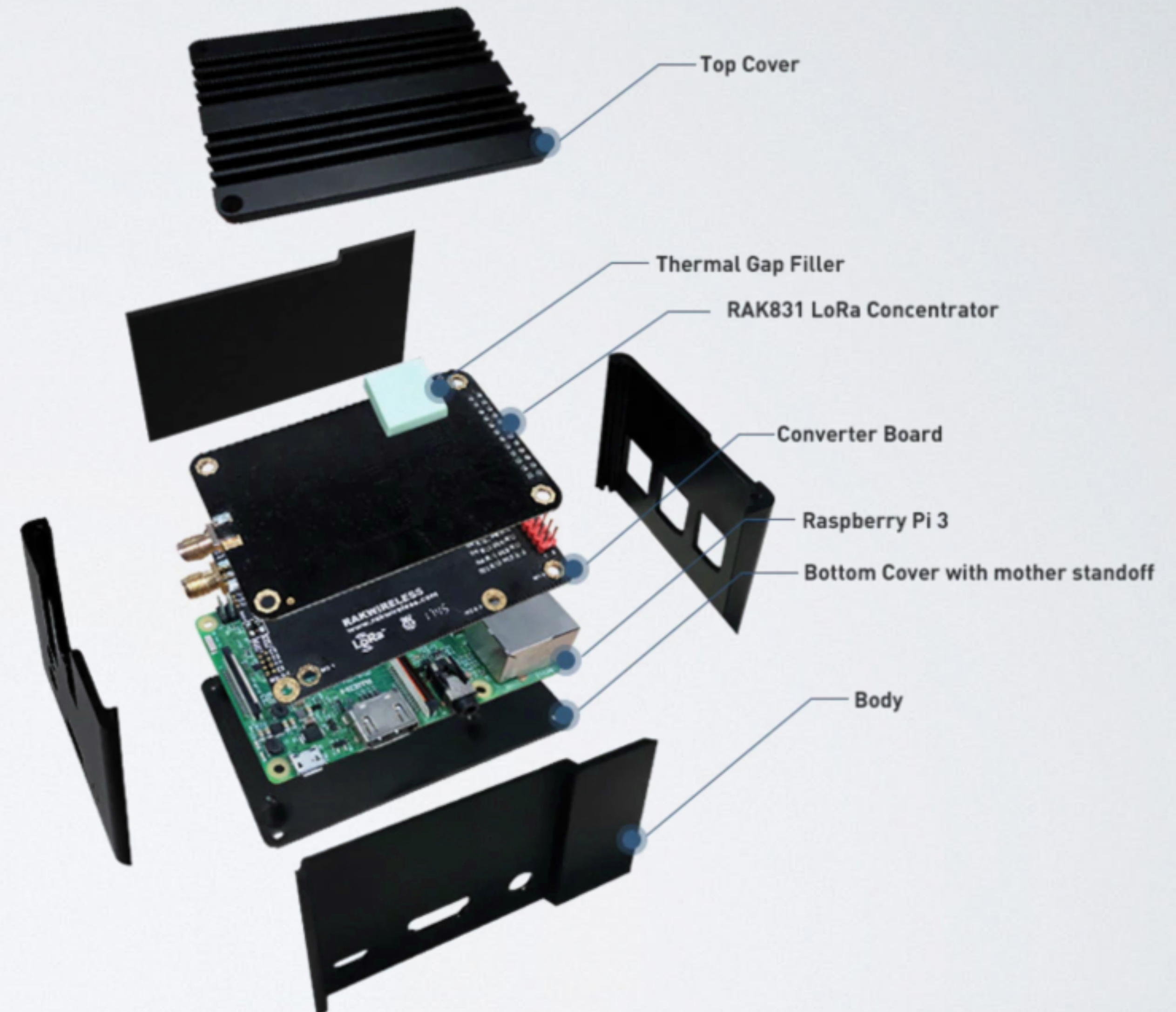
Type	Freq. (MHz)
433	433-435
470	RX: 470-490, TX 490-510
EU868	865-872
AS923	923-925
AU915	915-928
US915	902-928

- Checkout the [RAK document section](#) for more technical specifications.

RAK831 PILOT GATEWAY

The RAK831 Pilot Gateway consists of:

- a Raspberry Pi 3 model B+ (rev1.3),
- a converter board with a GPS (Ublox, MAX-7Q) module,
- a RAK831 LoRa Concentrator,
- an aluminium housing with the top cover serving as a heat sink housing size $D \times W \times H = 69 \times 92 \times 54$ mm,



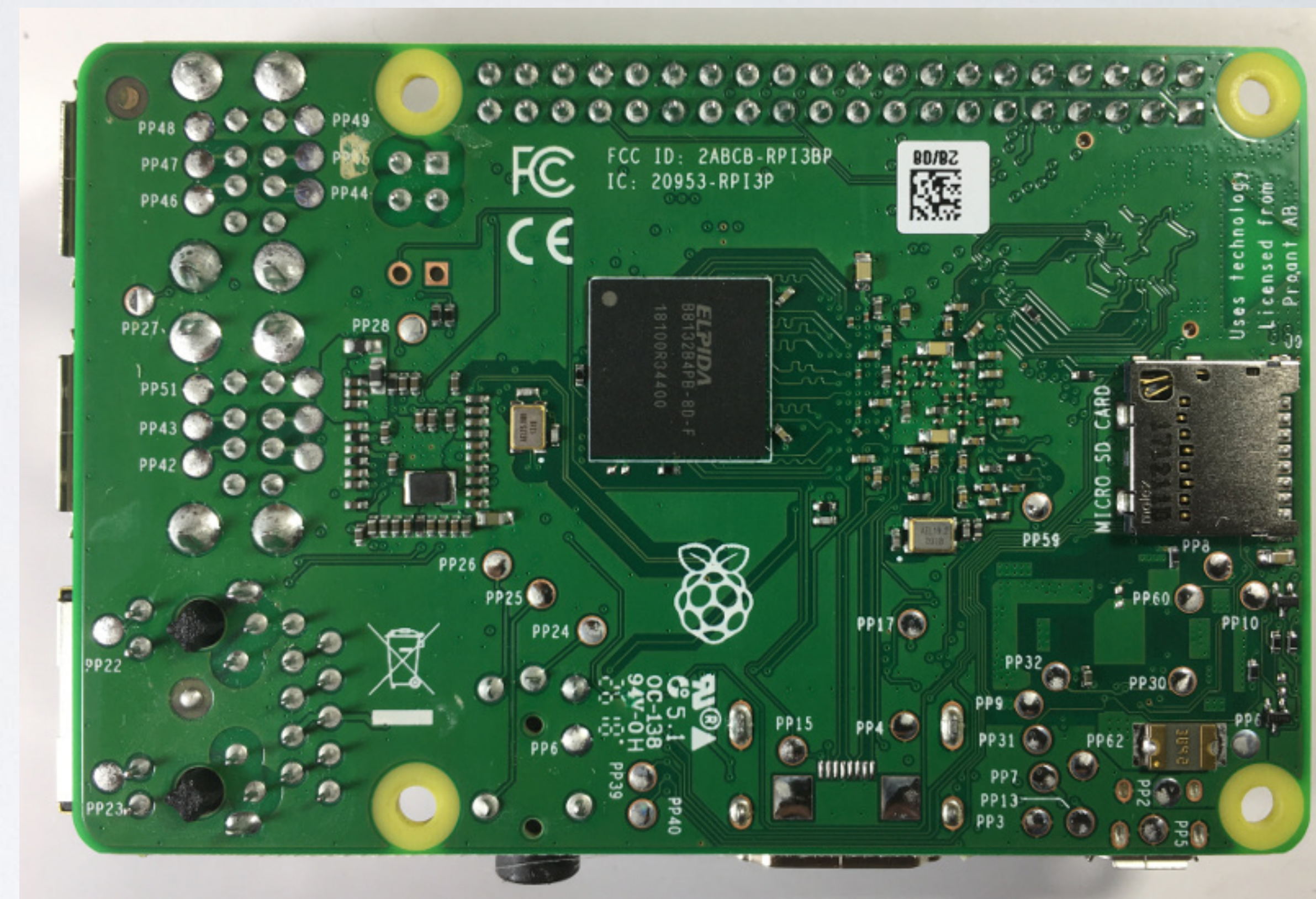
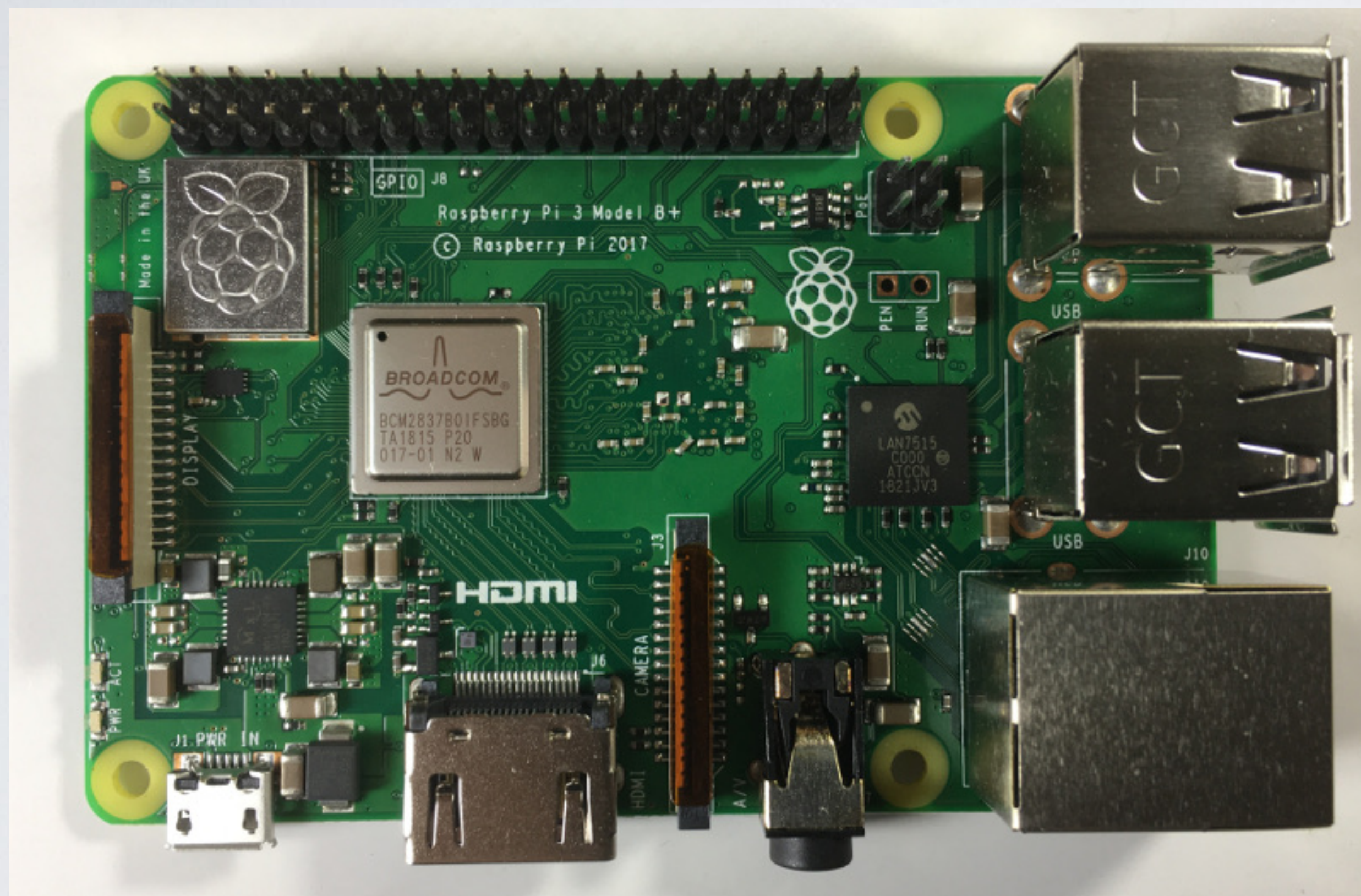
RAK831 PILOT GATEWAY

- a power adapter (5V, 2A) with micro USB, and the cable length is 115 cm,
- an antenna with the correct frequency for the LoRa radio,
- a GPS antenna and the cable length is 150 cm,
- a micro SD card (16 GB, class 10) with pre-installed software.

Note: I have noticed that RAK makes improvements to their products. Your RAK831 Pilot Gateway components may deviate from the one shown in this tutorial.



RAK831 PILOT GATEWAY



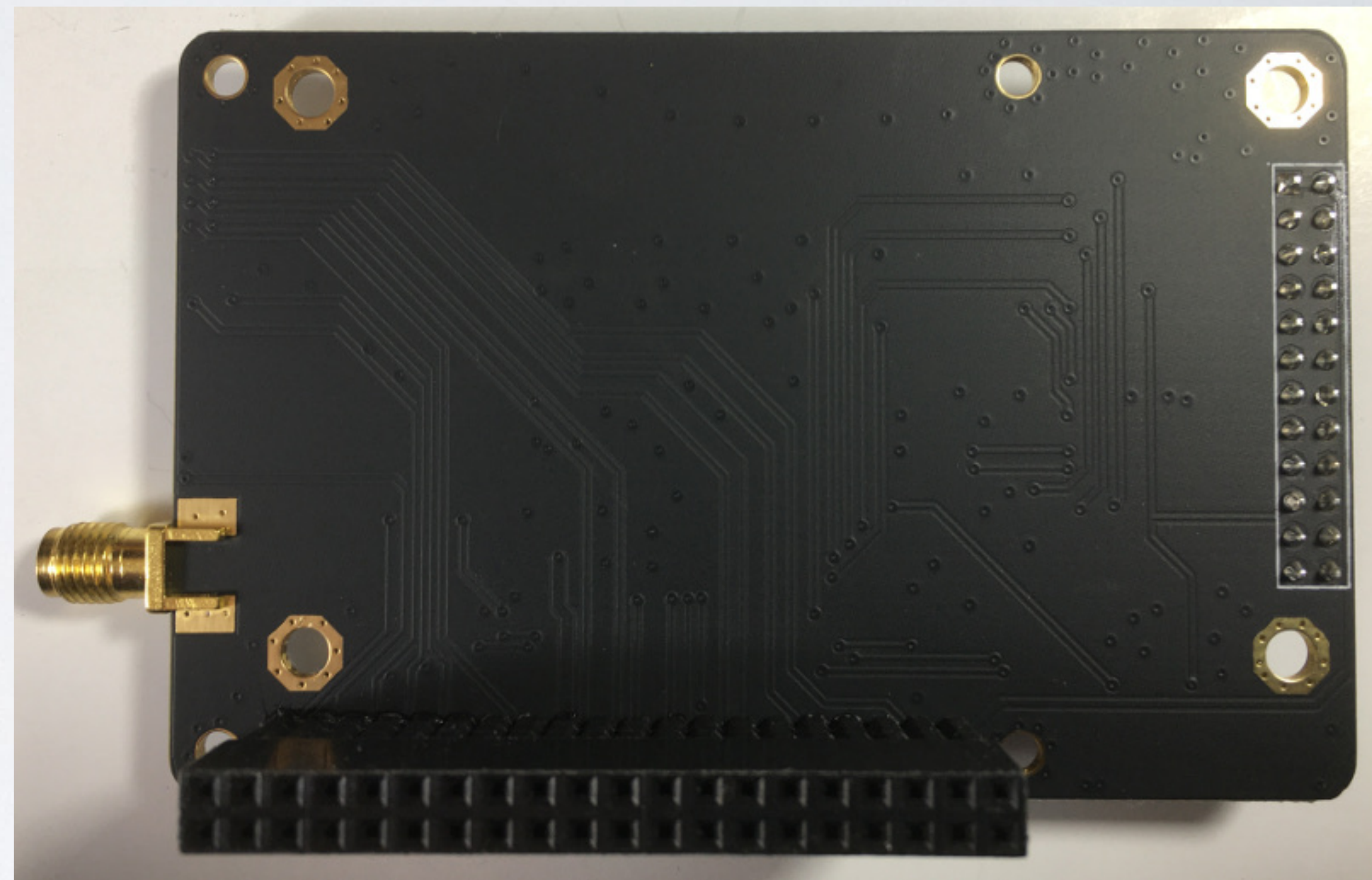
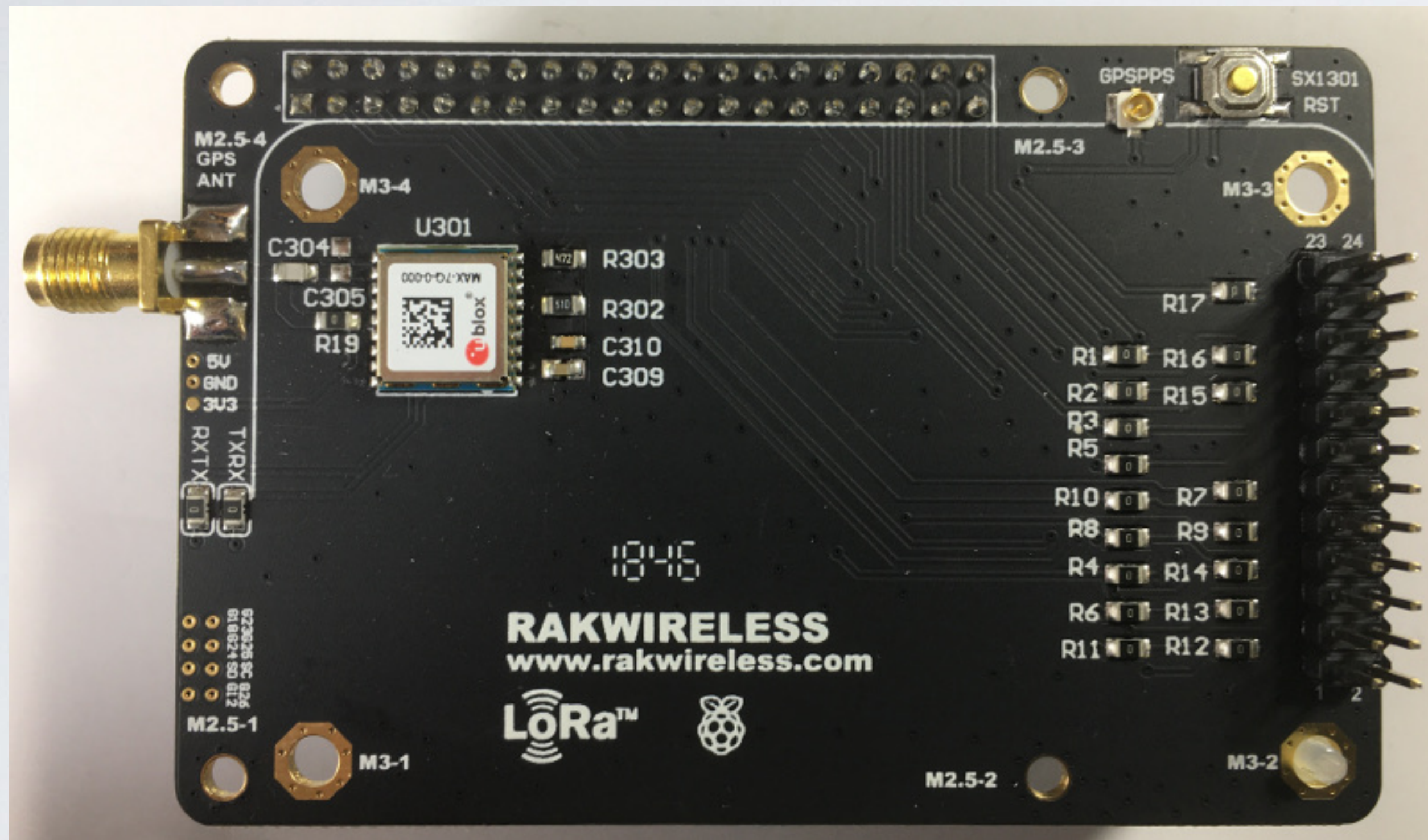
Raspberry Pi 3 Model B+

RAK831 PILOT GATEWAY



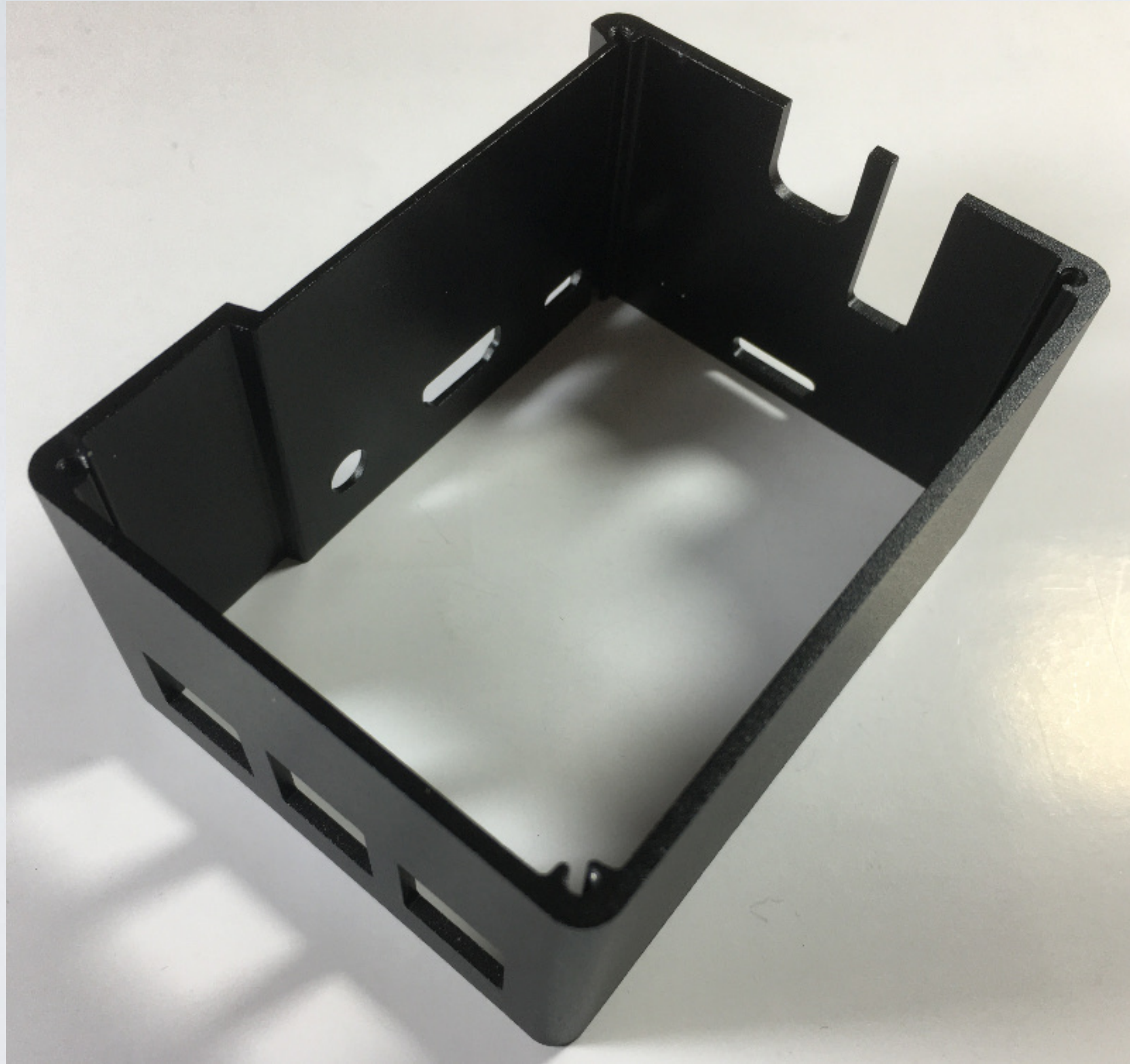
RAK831 LoRa Concentrator

RAK831 PILOT GATEWAY



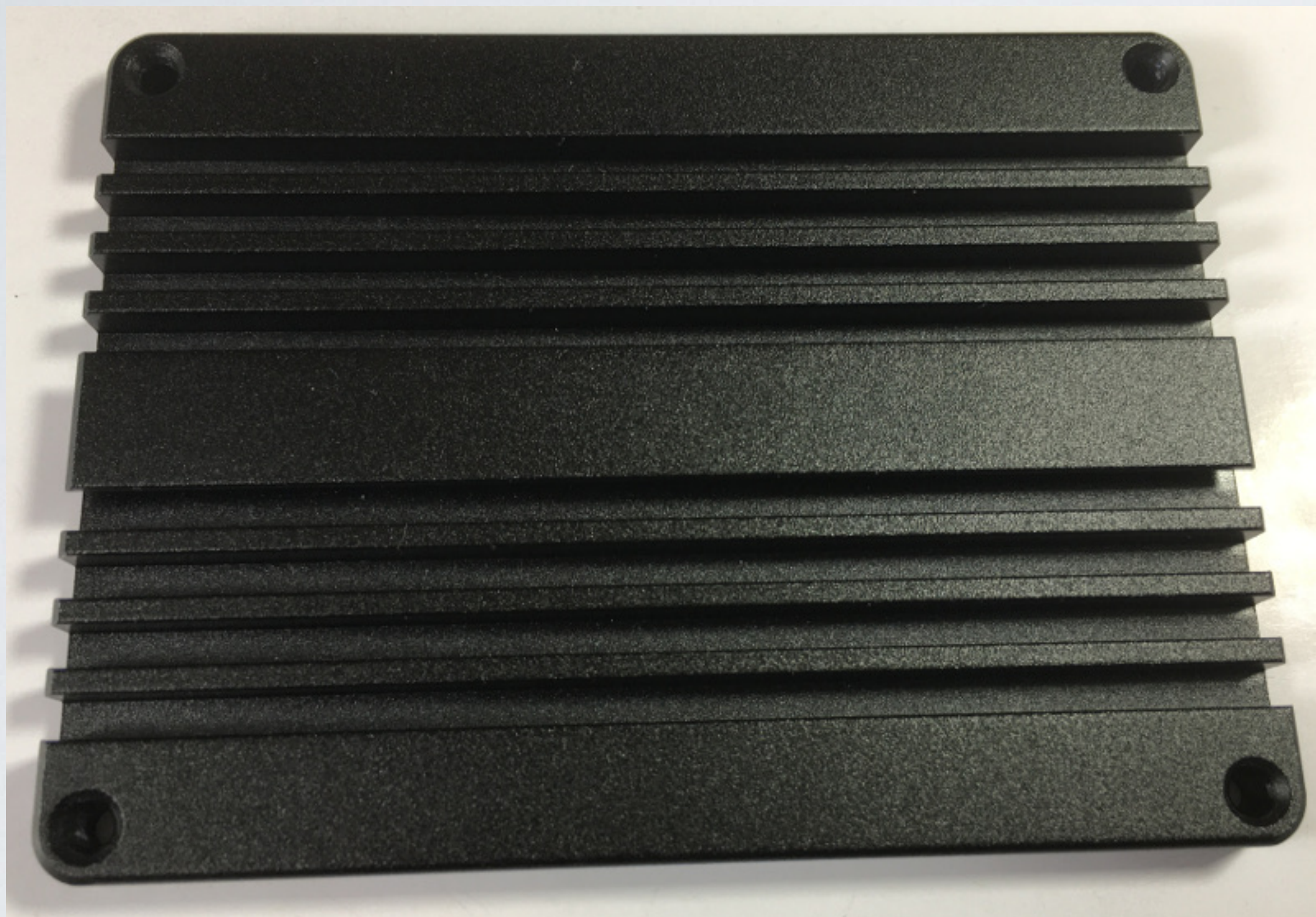
Converter Board

RAK831 PILOT GATEWAY



Enclosure

RAK831 PILOT GATEWAY



Enclosure top plate



Enclosure bottom plate

RAK831 PILOT GATEWAY



Assembly step 1

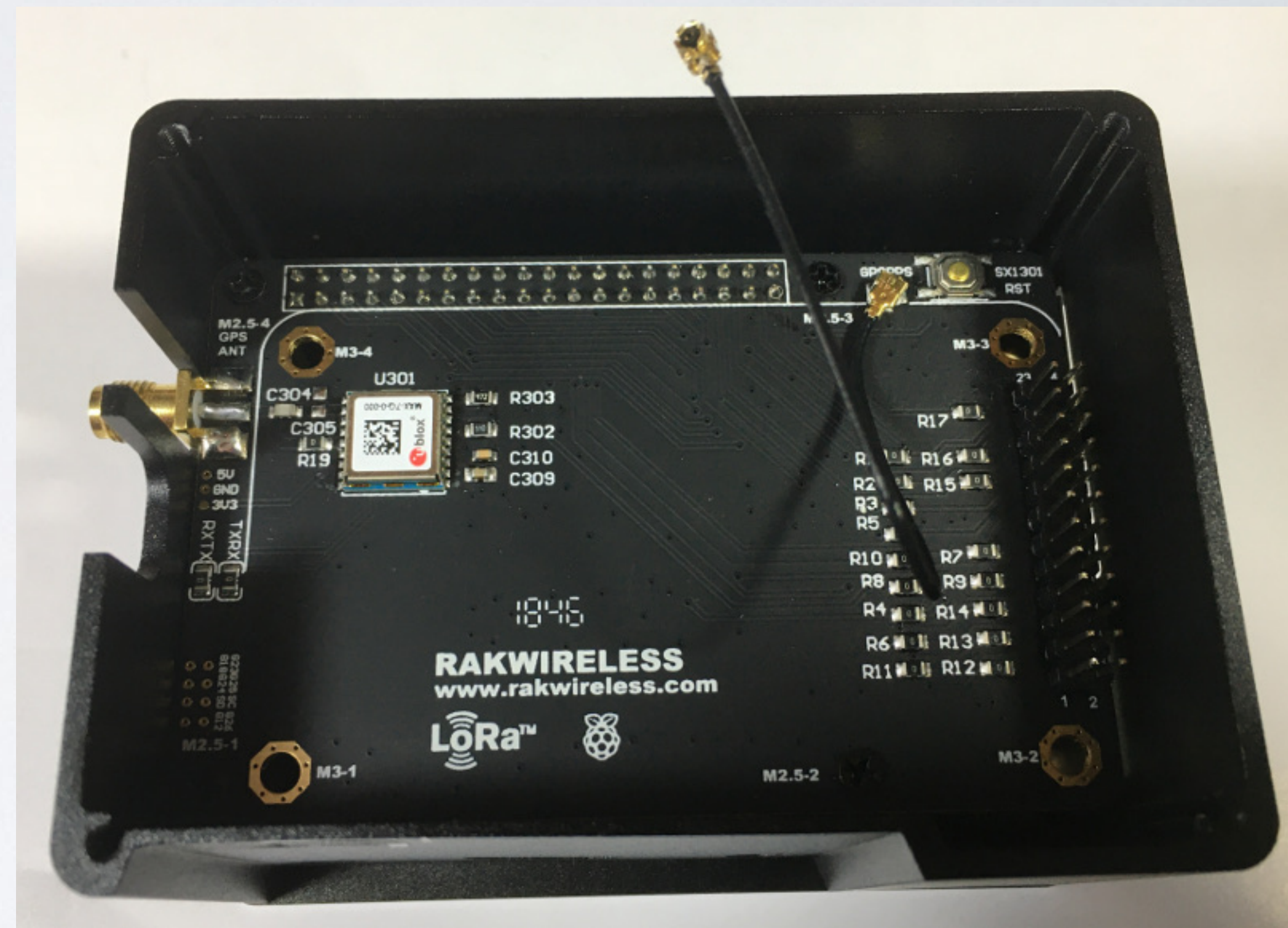


Assembly step 2

RAK831 PILOT GATEWAY

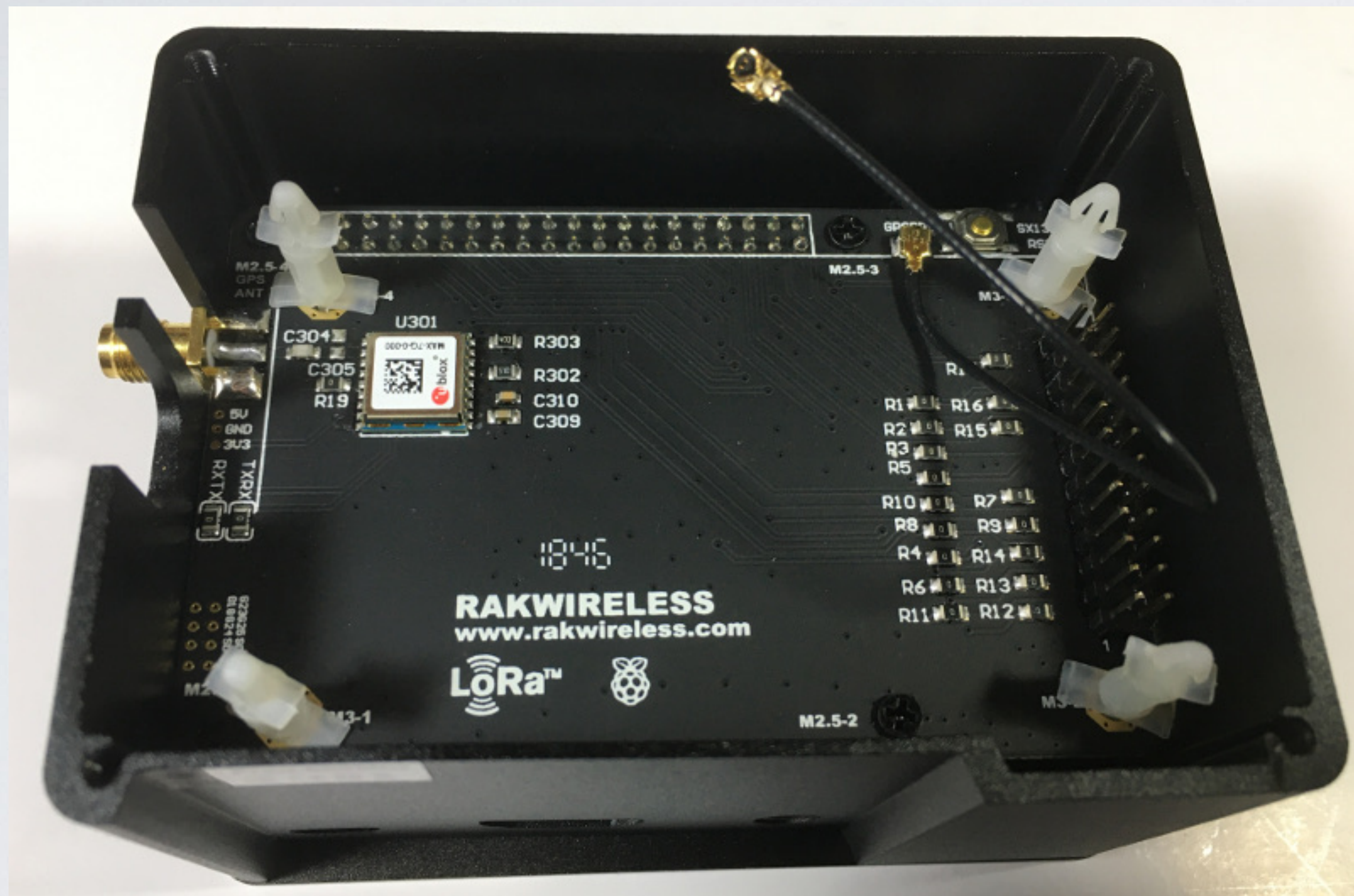


Assembly step 3



Assembly step 4

RAK831 PILOT GATEWAY

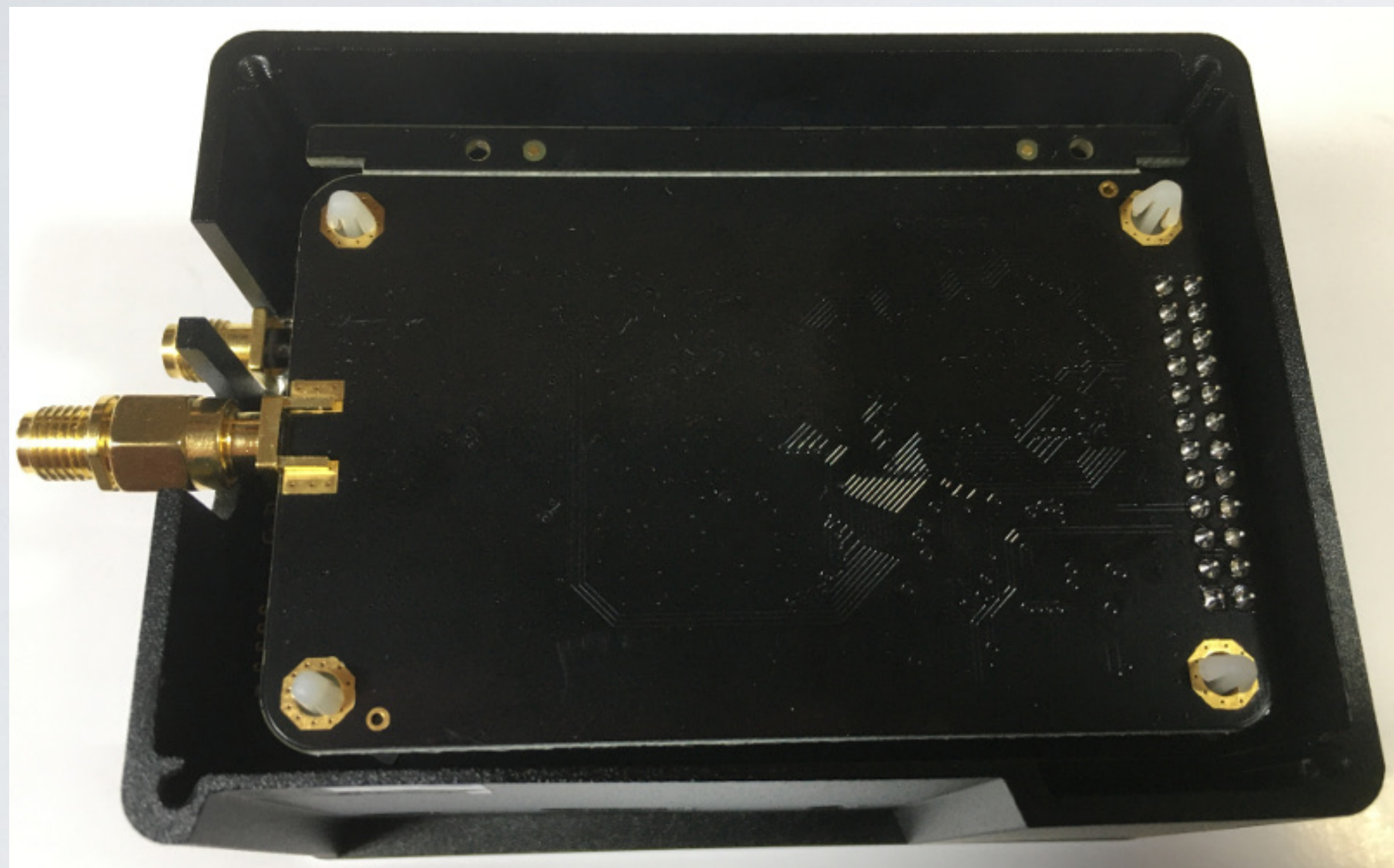


Assembly step 5

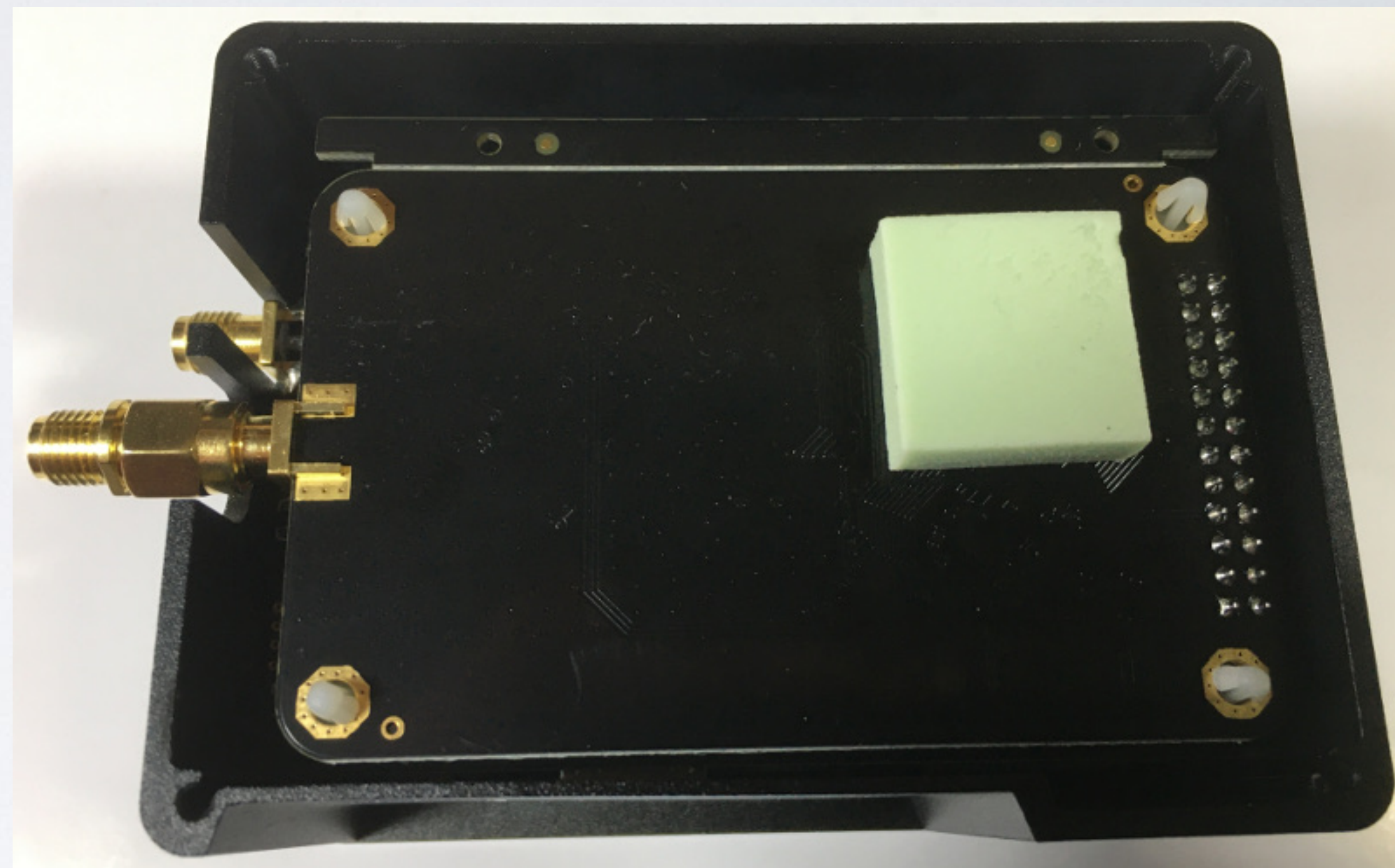


Assembly step 6

RAK831 PILOT GATEWAY



Assembly step 7



Assembly step 8

RAK831 PILOT GATEWAY

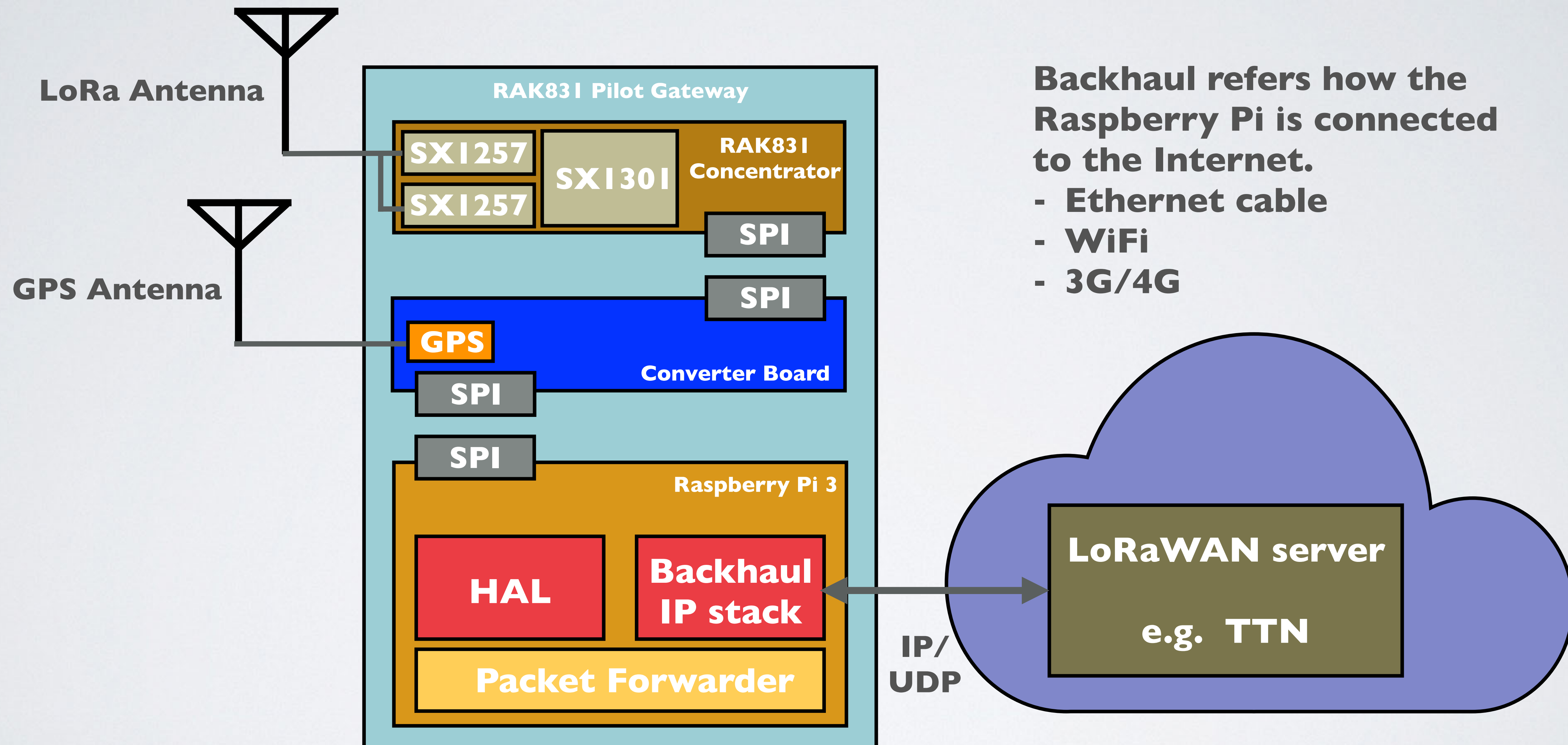


Assembly step 9

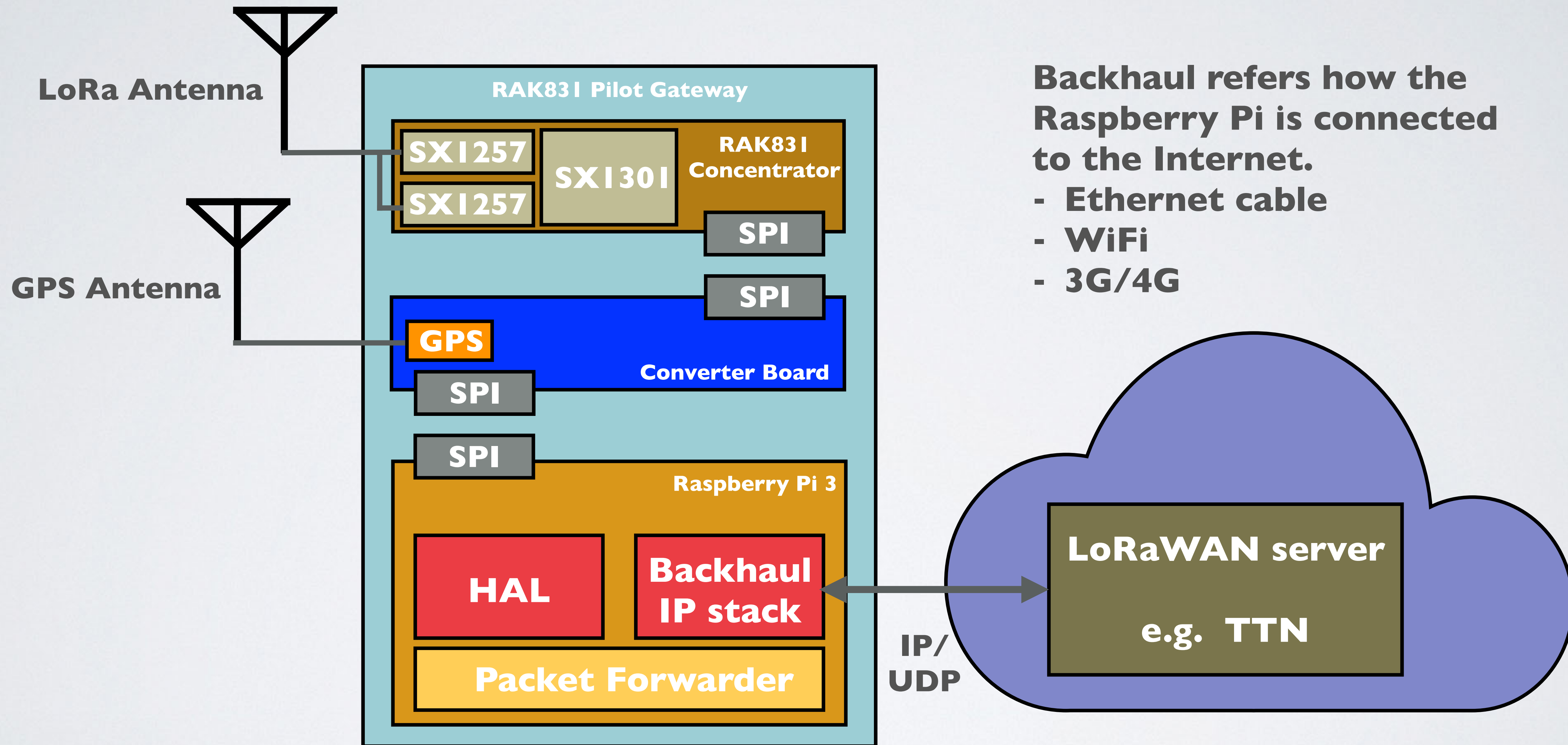
RAK831 PILOT GATEWAY

- The RAK831 concentrator uses the Semtech SX1301 chip and is able to receive up to 8 LoRa packets simultaneously, and can send with different spreading factors on different channels.
- The concentrator is the radio communication module which receives and transmits radio messages.
- The converter board with the GPS module routes the signals between the Raspberry Pi and the RAK831 concentrator.
- The Raspberry Pi processes these radio messages as well as protocol related tasks and sends the processed data to a LoRaWAN server, for example The Things Network server.

RAK831 PILOT GATEWAY



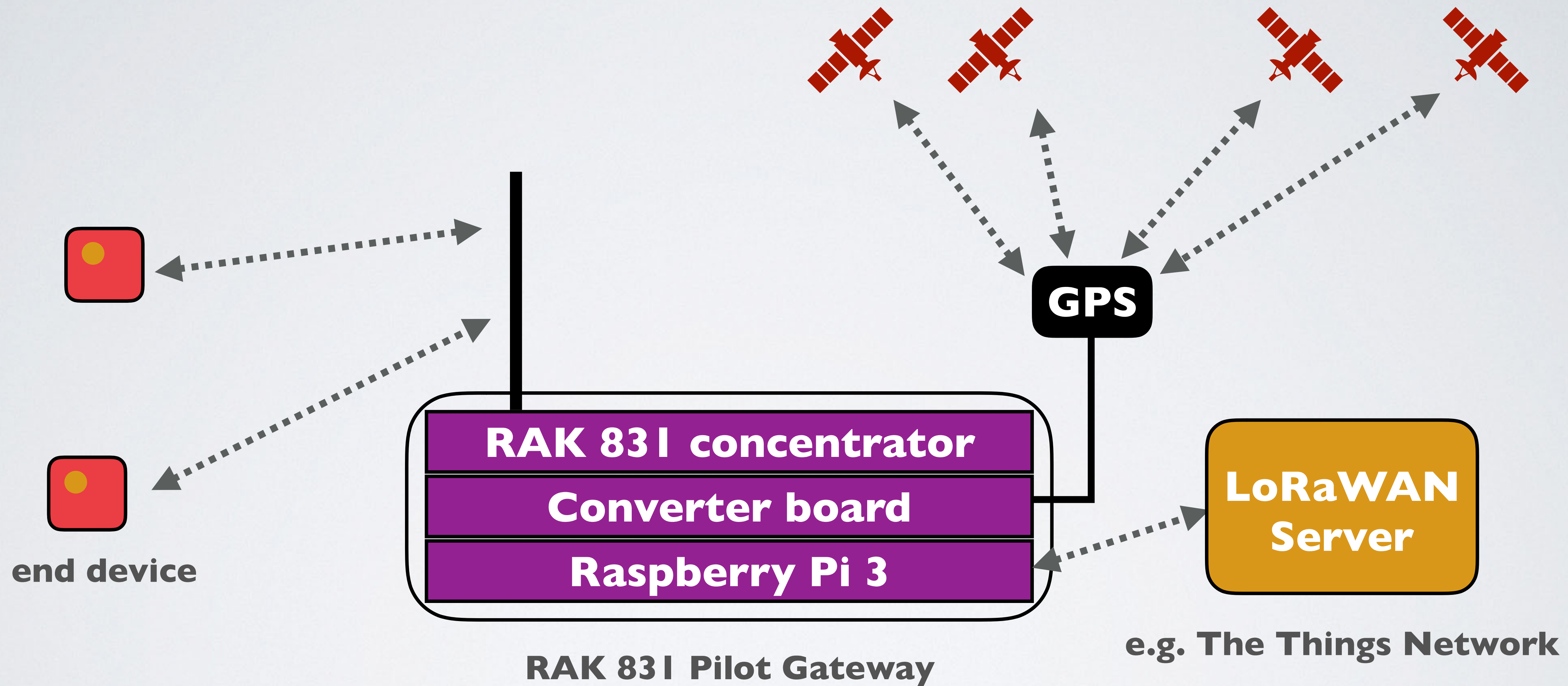
RAK831 PILOT GATEWAY



Backhaul refers how the Raspberry Pi is connected to the Internet.

- Ethernet cable
- WiFi
- 3G/4G

RAK831 PILOT GATEWAY



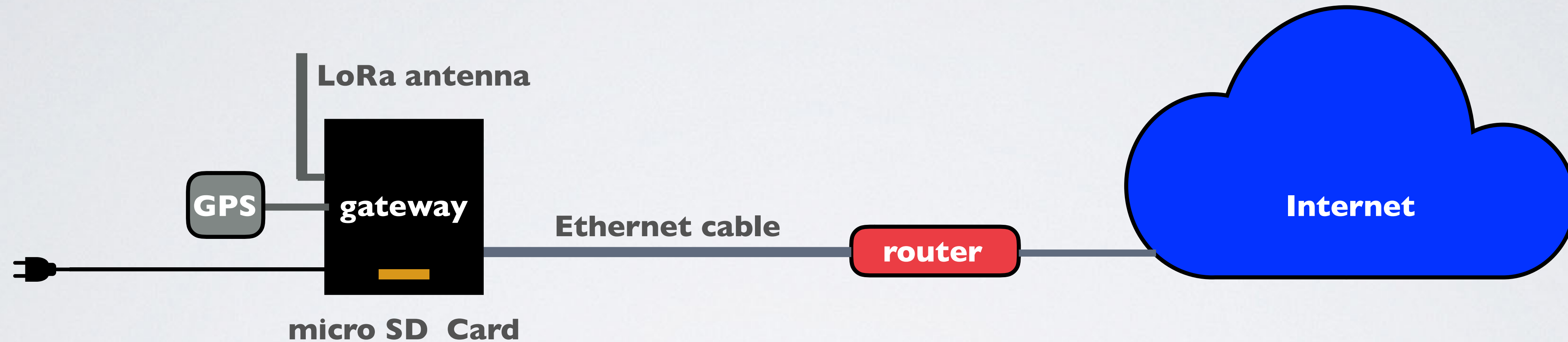
Get the gateway IP address

GET THE GATEWAY IP ADDRESS

- The RAK831 Pilot Gateway is a fully assembled gateway and needs to be configured before it can be used.
- All software are already installed on the micro SD card.
Insert the micro SD card into the Raspberry Pi.
- Connect the LoRa antenna to the gateway.
Never power up the gateway without the LoRa antenna, otherwise this can cause damage to your gateway.
- You do not need to connect the GPS antenna because GPS is by default disabled.
I recommend not to connect the GPS antenna.
I will explain this later.

GET THE GATEWAY IP ADDRESS

- Connect the gateway to a router using an Ethernet cable (this is not included).



- Power up the gateway using the included power adapter. Do not use a power adapter with less than 2A output.
- The gateway has the hostname **ttn-gateway**

GET THE GATEWAY IP ADDRESS

- There are several methods to get the gateway IP address, I will only demonstrate 3.
- **Method 1. Router devices list**
 - Open a browser and navigate to your router's IP address e.g. `http://192.168.1.1`
The IP address is usually printed on a label on your router or check your router's manual.
 - To access your router's configuration interface you need its username and password.
Check your router's manual.
 - Locate the gateway (ttn-gateway) and write down its IP address.

GET THE GATEWAY IP ADDRESS

The screenshot shows a web browser window with the address bar displaying "192.168.1.1/index.html". The browser's address bar also shows "Not Secure" and "192.168.1.1/index.html". The browser's address bar also shows "Not Secure" and "192.168.1.1/index.html".

The main content area of the browser displays a network management interface. At the top right, there are options for "English" and "Logout". The interface features a network diagram with a laptop icon and a "Status" button. A modal dialog box titled "Info" is open, displaying the following details for the "ttn-gateway":

ttn-gateway	
IPv4 Address:	192.168.1.71
MAC Address:	b8:27:eb:c7:4b:36
Address Source:	DHCP
Expires In:	0 hours:58 minutes:0 seconds
Connection Type:	Ethernet

Below the dialog box, there is a checkbox labeled "Add device into Static DHCP" which is currently unchecked. An "OK" button is located at the bottom right of the dialog box. The bottom navigation bar of the interface includes icons and labels for "Connection Status", "Network Setting", "Security", "System Monitor", and "Maintenance".

GET THE GATEWAY IP ADDRESS

• **Method 2. Resolving <hostname>.local with mDNS**

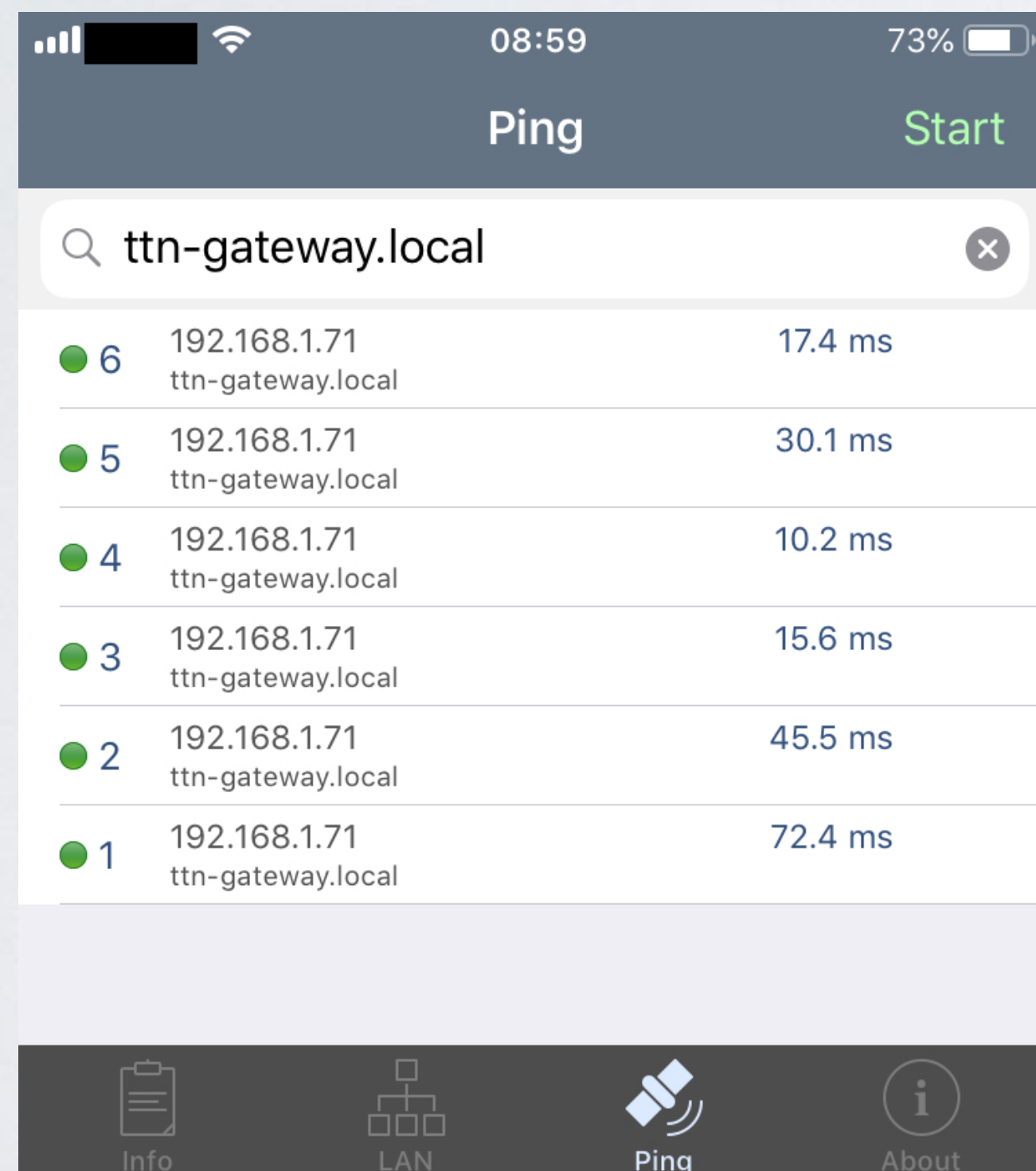
- On Raspbian, multicast DNS is supported out-of-the-box by the Avahi service.
- If your computer supports mDNS, you can reach your gateway (Raspberry Pi) by using its hostname and the **.local** suffix.

Enter command: **ping ttn-gateway.local**

```
[~$ ping ttn-gateway.local
PING ttn-gateway.local (192.168.1.71): 56 data bytes
64 bytes from 192.168.1.71: icmp_seq=0 ttl=64 time=3.754 ms
64 bytes from 192.168.1.71: icmp_seq=1 ttl=64 time=2.757 ms
64 bytes from 192.168.1.71: icmp_seq=2 ttl=64 time=2.762 ms
```

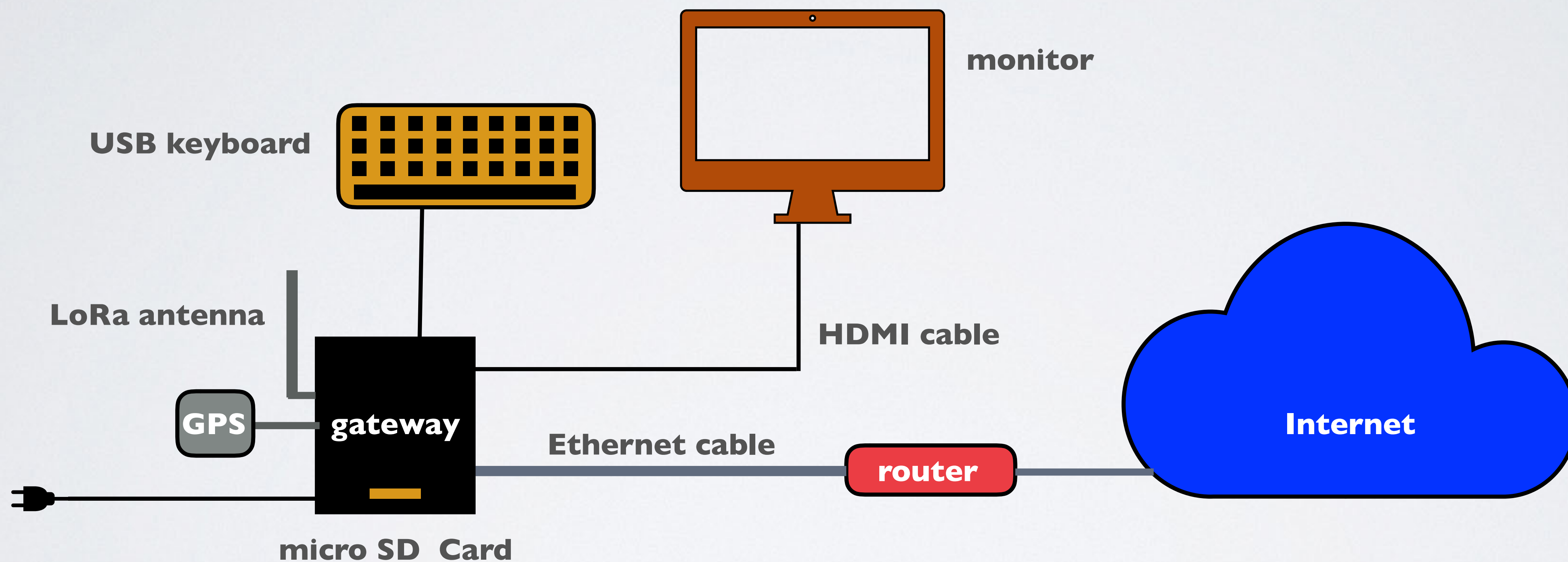
GET THE GATEWAY IP ADDRESS

- If your computer does not support mDNS, install the **Net Analyzer** app (iOS and Android) on your mobile.
- Select the Ping option and enter: **ttn-gateway.local**



GET THE GATEWAY IP ADDRESS

- **Method 3. Connect monitor and keyboard**



GET THE GATEWAY IP ADDRESS

- You will see the login prompt.

The default username=**pi** and password=**raspberrry**

Note: The pi password may have changed to **11111111** in the shipped micro SD card.

- Login with above mentioned credentials.
- Enter command: **hostname -I**
- The gateway IP address will be shown, for example: 192.168.1.71

ACCESS THE RASPBERRY PI

- On Linux and Mac the SSH command can be used to access the Raspberry Pi.
- On Windows 10, SSH can be used or instead use Putty, <https://www.putty.org>.
- To enable SSH functionality on the Windows 10 PowerShell:
 - Open Settings
 - View the Apps
 - Go to the Manage optional features
 - Click Add a feature
 - Select OpenSSH Client
 - Wait, then reboot

ACCESS THE RASPBERRY PI

- Access the Raspberry Pi.

```
ssh pi<ip_address_gateway>
```

For example: **ssh pi@192.168.1.71**

- The default username=**pi** and password=**raspberrypi**

Note: The pi password may have changed to **11111111** in the shipped micro SD.

Configure Raspberry Pi

CONFIGURE RASPBERRY PI

- Open the Raspberry Pi Software Configuration Tool.
sudo raspi-config
- Change the user password.
Select menu: **1 Change User Password**
- Press Enter and follow the steps.
Do not forget your password!
- For security reasons you must change the user pi default password.

CONFIGURE RASPBERRY PI

- Change localisation.

Select menu: **4 Localisation Options**

- Select menu: **I1 Change Locale**

For example I live in the Netherlands:

`nl_NL.UTF-8`

For demonstration purposes I will keep the default locale **en_GB.UTF-8**

- Select menu: **I2 Change Timezone**

For example:

Geographic area: **Europe**

Time zone: **Amsterdam**

CONFIGURE RASPBERRY PI

- Check if the disk space matches the micro SD card size.

df -h

- Expand the filesystem on the micro SD card.

Select menu: **7 Advanced Options**

- Select menu: **A1 Expand Filesystem**

- This ensures that all of the SD card storage is available to the Operating System.

CONFIGURE RASPBERRY PI (NOT IMPORTANT)

- NOT important configurations you might want to apply:
- Disable the annoying “You have new mail.” message in the console.

Type: `cd /etc/pam.d/`

Type: `sudo nano sshd`

Comment out line:

```
#session optional pam_mail.so standard noenv
```


REBOOT AND LOGIN RASPBERRY PI

- The Raspberry Pi Software Configuration Tool can reboot the Raspberry Pi or you can do it your self:

```
sudo reboot
```

- Again access the Raspberry Pi.

```
ssh pi<ip_address_gateway>
```

```
password=<your_new_password>
```

VERIFY CHANGES

- Check the disk space.

df -h

- Check time zone.

date

Get the gateway EUI

GET GATEWAY MAC ADDRESS

- To register your gateway with The Things Network you need its gateway EUI (Extended Unique Identifier).
- Once a gateway EUI is registered with The Things Network it can not be used by someone else.
- The gateway EUI can be any random 8 bytes number but it is recommended that the gateway EUI is based on the gateway MAC (Media Access Control) address.
- A MAC address is a hardcoded identification number which is assigned to your network interface card.
- The Raspberry Pi 3 has a Ethernet port and built in WiFi.

GET GATEWAY MAC ADDRESS

- Each network interface (Ethernet and WiFi) has its own MAC address.
eth0 is the name of the first Ethernet interface.
wlan0 is the name of the first wireless interface.
- If the gateway is connected to a router using the Ethernet cable, the eth0 network interface is used. To get the eth0 network interface MAC address:
ip link show eth0
- If the gateway is connected to a router using WiFi, the wlan0 network interface is used. To get the wlan0 network interface MAC address:
ip link show wlan0
- Please note: WiFi is by default disabled in the RAK831 Pilot Gateway.

GET GATEWAY MAC ADDRESS

- For example:

```
pi@ttn-gateway:~ $ ip link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether b8:27:eb:c7:4b:36 brd ff:ff:ff:ff:ff:ff
```

MAC address

- All Raspberry Pis have MAC addresses which starts with: **b8:27:eb**

GATEWAY EUI

- MAC address = **b8:27:eb:c7:4b:36**
- Insert **fffe** after the 3rd byte: **b8:27:eb:fffec7:4b:36**
- Remove all colons: **b827ebfffec74b36**
- Change to uppercase: **B827EBFFFE74B36**
- The gateway EUI = **B827EBFFFE74B36**
- *Write down the gateway EUI, you will need it later when registering the gateway with The Things Network console.*

Gateway installed software

GATEWAY INSTALLED SOFTWARE

- The gateway is shipped with a micro SD card with the following software pre-installed.
 - Raspbian Stretch Lite
<https://www.raspberrypi.org/downloads/raspbian>
 - Semtech LoRa library (V5.0.1)
https://github.com/Lora-net/lora_gateway
/opt/ttn-gateway/lora_gateway
 - Semtech UDP Packet Forwarder (V4.0.1)
https://github.com/Lora-net/packet_forwarder
/opt/ttn-gateway/packet_forwarder

GATEWAY INSTALLED SOFTWARE

- In case the Semtech LoRa library (lora_gateway) and Semtech UDP Packet Forwarder (packet_forwarder) are installed in two different folders, for example:

/home/pi/ttn-gateway/opt/ttn-gateway

and

/opt/ttn-gateway

...than always use the one installed in folder **/opt/ttn-gateway** and ignore the other.

- Note: Instead of the Semtech UDP Packet Forwarder, the RAK831 Pilot Gateway can also work with other packet forwarders, but that is a subject for another video.

RASPBIAN

- Raspbian is the operating system which runs on the Raspberry Pi.
- To check which Raspbian release is installed:

cat /etc/os-release

```
pi@ttn-gateway:~ $ cat /etc/os-release
PRETTY_NAME="Raspbian GNU/Linux 9 (stretch)"
NAME="Raspbian GNU/Linux"
VERSION_ID="9"
VERSION="9 (stretch)"
ID=raspbian
ID_LIKE=debian
HOME_URL="http://www.raspbian.org/"
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"
```

RASPBIAN

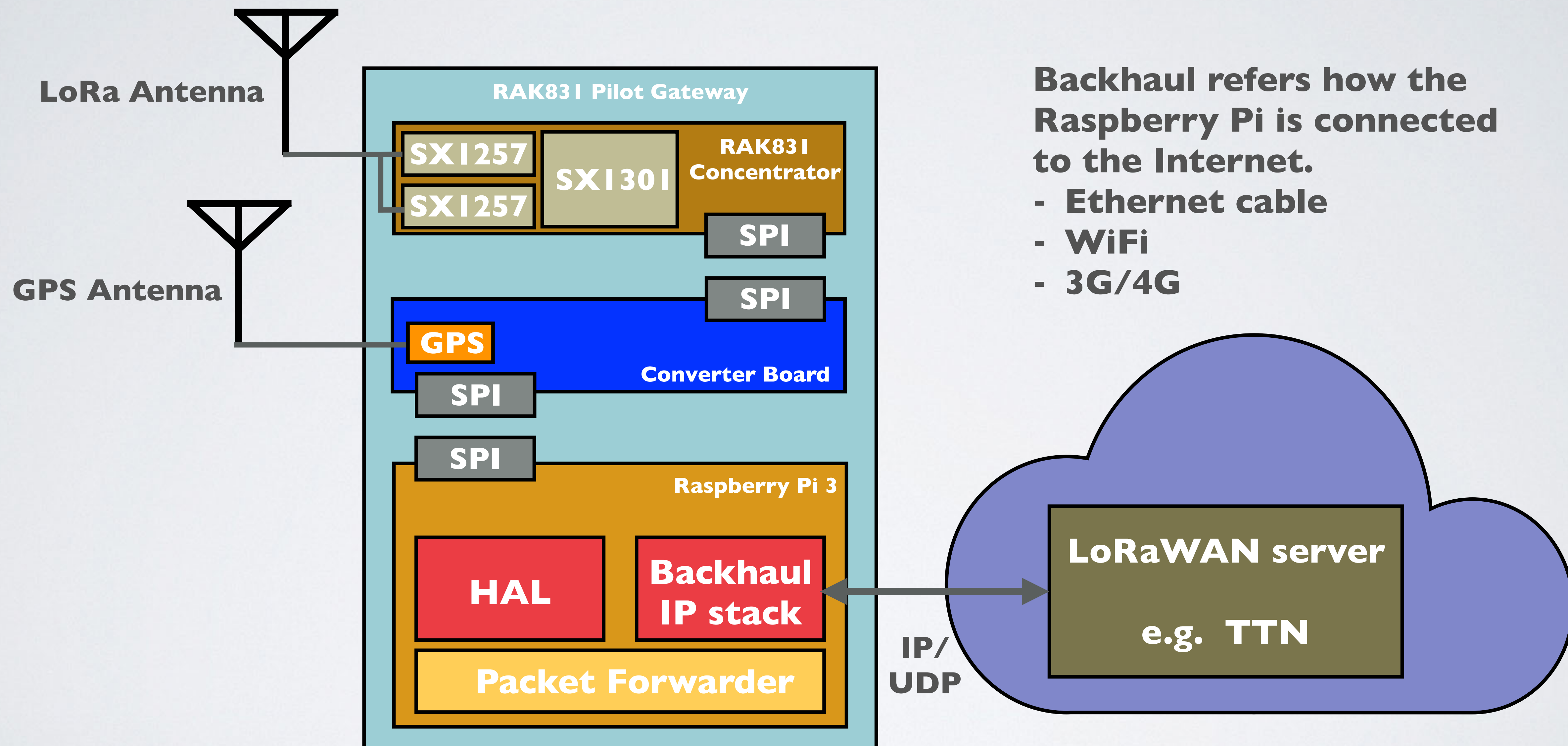
- A Raspbian Lite version was installed, meaning no desktop. To verify if Raspberry Pi desktop is not installed, check for package “raspberrypi-ui-mods”. The Raspberry Pi desktop requires this package.

```
dpkg -l | grep raspberrypi-ui-mods
```

LORA LIBRARY AND UDP PACKET FORWARDER

- The Semtech LoRa library contains drivers/HAL (Hardware Abstraction Layer) to be used in gateways using a concentrator board based on Semtech SX1301 multi-channel modem and SX1257/SX1255 RF transceivers.
- The Semtech UDP Packet Forwarder is a program running on the host of a LoRa gateway (in my case the Raspberry Pi) that forwards RF packets received by the concentrator to a LoRaWAN server through an IP/UDP link.
The packet forwarder can also transmit RF packets which are sent by the LoRaWAN server.

LORA GATEWAY AND PACKET FORWARDER



**Change the frequencies by
updating the global_conf.json file**

<REGIONAL>-GLOBAL_CONF.JSON

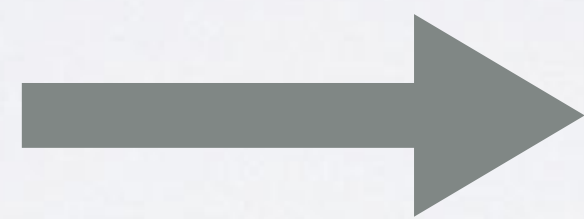
- In this tutorial the gateway will be connected to The Things Network server.
- A <regional>-global_conf.json file is needed which can be downloaded from:
<https://github.com/TheThingsNetwork/gateway-conf>
- This regional configuration file sets the frequencies which applies to a specific region and it also sets The Things Network server address.
- Depending on the region your gateway will operate in, download the correct <regional>-global_conf.json file.
For example, if your gateway operate in:
 - Europe, download EU-global_conf.json
 - China, download CN-global_conf.json
 - etc...

<REGIONAL>-GLOBAL_CONF.JSON

- If the downloaded <regional>-global_conf.json file contains the “lbt_cfg” key than change the “enable” value from true to false.
This will disable the Listen-Before-Talk feature (LBT).

```
"lbt_cfg": {  
  "enable": true,  
  :
```

Change



```
"lbt_cfg": {  
  "enable": false,  
  :
```

- You can connect the gateway to another LoRaWAN server, instead of TTN server, for example lorasever.io. But additional changes need to be made to the <regional>-global_conf.json. How this is done, see:

<https://www.thethingsnetwork.org/docs/gateways/packet-forwarder/semtech-udp.html>

REPLACE GLOBAL_CONF.JSON

- The Raspberry Pi folder

`/opt/ttn-gateway/packet_forwarder/lora_pkt_fwd`

contains a `global_conf.json` file.

- Rename `global_conf.json` file.

`cd /opt/ttn-gateway/packet_forwarder/lora_pkt_fwd`

`mv global_conf.json global_conf.json_orig`

- Download the `<regional>-global_conf.json` file. For example:

`wget https://raw.githubusercontent.com/TheThingsNetwork/gateway-conf/master/EU-global_conf.json`

- Rename the file.

`mv EU-global_conf.json global_conf.json`

REPLACE GLOBAL_CONF.JSON

- An alternative method.

Download the <regional>-global_conf.json to your computer and copy the json file to the Raspberry Pi lora_pkt_fwd folder and name this file global_conf.json.

```
scp ./<regional>-global_conf.json  
pi@<ip_address_gateway>:/opt/ttn-gateway/  
packet_forwarder/lora_pkt_fwd/global_conf.json
```

```
[~/Downloads$ ls EU-global_conf.json  
EU-global_conf.json  
[~/Downloads$ scp ./EU-global_conf.json pi@192.168.1.71:/opt/ttn-g  
ateway/packet_forwarder/lora_pkt_fwd/global_conf.json  
[pi@192.168.1.71's password:  
EU-global_conf.json          100% 4263    925.3KB/s   00:00
```

REPLACE GLOBAL_CONF.JSON

- Reboot and login into the Raspberry Pi.

```
sudo reboot
```

```
ssh pi@<ip_address_gateway>
```

- Check ttn-gateway service status

```
systemctl status ttn-gateway -l
```

```
pi@ttn-gateway:~ $ systemctl status ttn-gateway -l
• ttn-gateway.service - The Things Network Gateway
  Loaded: loaded (/lib/systemd/system/ttn-gateway.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2019-01-10 11:14:46 CET; 1min 30s ago
  Main PID: 543 (start.sh)
  CGroup: /system.slice/ttn-gateway.service
          └─543 /bin/bash /opt/ttn-gateway/packet_forwarder/lora_pkt_fwd/start.sh
            └─562 ./lora_pkt_fwd

Jan 10 11:15:44 ttn-gateway ttn-gateway[543]: ### [JIT] ###
Jan 10 11:15:44 ttn-gateway ttn-gateway[543]: # SX1301 time (PPS): 31581897
Jan 10 11:15:44 ttn-gateway ttn-gateway[543]: src/jitqueue.c:448:jit_pr...queue(): INFO: [jit] queue :
Jan 10 11:15:44 ttn-gateway ttn-gateway[543]: ### [GPS] ###
Jan 10 11:15:44 ttn-gateway ttn-gateway[543]: # GPS sync is disabled
Jan 10 11:15:44 ttn-gateway ttn-gateway[543]: ##### END #####
Jan 10 11:15:44 ttn-gateway ttn-gateway[543]: JSON up: {"stat":{"time":"2019-01-10 10:15:28 GMT","rxnb
Jan 10 11:15:44 ttn-gateway ttn-gateway[543]: INFO: [up] PUSH_ACK received in 25 ms
Jan 10 11:15:44 ttn-gateway ttn-gateway[543]: INFO: [down] PULL_ACK received in 23 ms
Jan 10 11:15:44 ttn-gateway ttn-gateway[543]: INFO: [down] PULL_ACK received in 24 ms
```

There are no errors
and GPS is disabled.

Also see:

```
/var/log/syslog
```

**Optionally enable GPS by
updating the local_conf.json file**

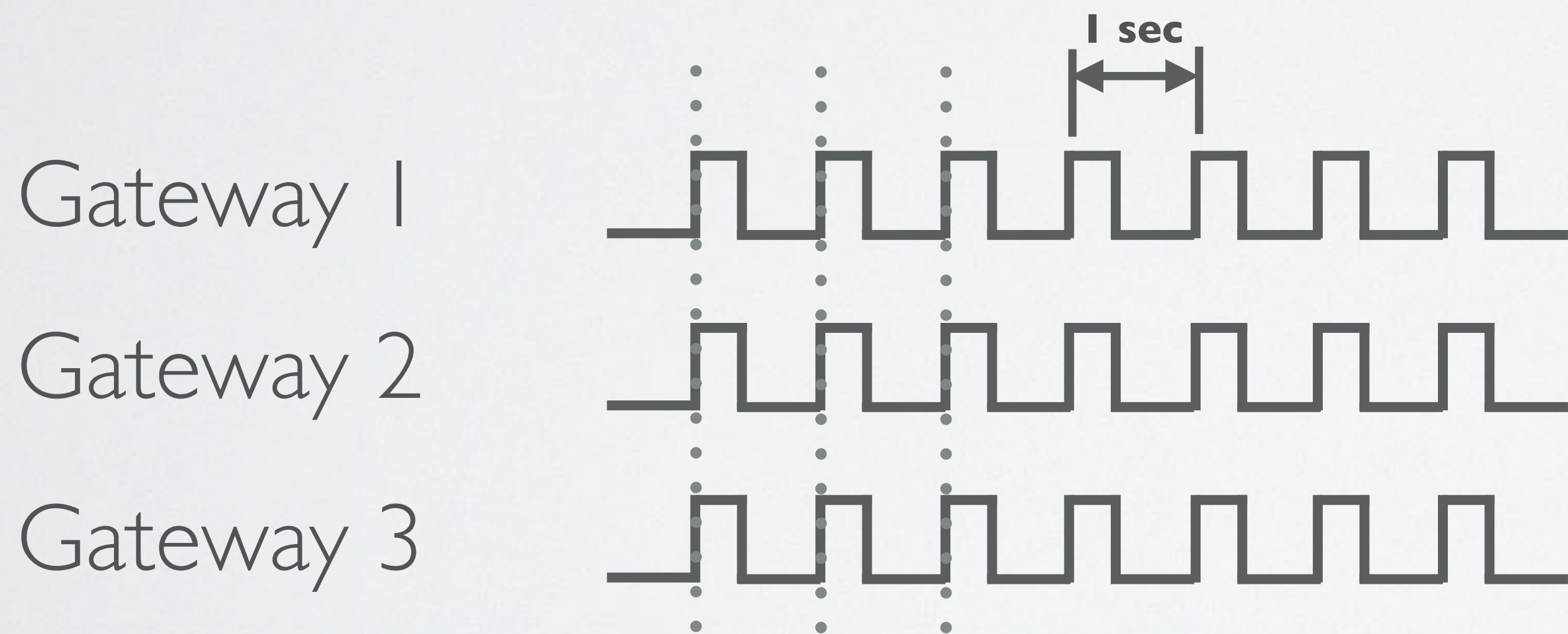
OPTIONALLY ENABLE GPS

- The RAK831 Pilot Gateway is shipped with a converter board with an Ublox MAX-7Q GPS (Global Positioning System) module soldered to it and a GPS antenna.
- The GPS module is connected via a UART/serial port (Rx/Tx) to the Raspberry Pi.
- GPS satellites broadcasts navigational messages to Earth, which contains very accurate time stamps and their position.



OPTIONALLY ENABLE GPS

- The gateway GPS module receives these navigational messages. With this information the GPS module calculates its position on Earth.
- The GPS module can use these accurate time stamps to create a PPS (Pulse Per Second) signal that is used for timing/synchronisation purposes. Every LoRaWAN gateway with a GPS module generates the same synchronised PPS signal.



OPTIONALLY ENABLE GPS

- A gateway PPS signal is used to create time synchronised beacons which are send to class B end devices.
- As explained in Tutorial 4, class B devices opens extra receive slots at scheduled times in addition to class A's random receive slots.
- In order for the end device to open its receive slots at scheduled times, it requires a time synchronised beacon. This allows the gateway to know when the node is listening.

OPTIONALLY ENABLE GPS

- The RAK831 Pilot Gateway has a GPS module but it is disabled. This is done because class B functionality is not implemented yet in LoRaWAN servers, such as The Things Network servers. This video is made in Jan 2019.
- It is therefore not useful yet to enable GPS in the RAK831 Pilot Gateway. If GPS is disabled, the GPS antenna does not need to be attached.
- Gateways usually do not change position, so its coordinates can be specified in the TTN console.
- In the packet forwarder software you can configure if a real, fake or no GPS is used. In the following slides I will demonstrate how to enable GPS and set some beaconing parameters, in case you want to know how it is done. Note: I can not offer you more information regarding class B devices and beaconing.

OPTIONALLY ENABLE GPS

- To set the beaconing parameters I have used parameters found in:
https://github.com/Lora-net/packet_forwarder/tree/master/lora_pkt_fwd/cfg
The RAK831 Pilot Gateway uses this packet forwarder.
I live in the Netherlands, so I use this file:
https://github.com/Lora-net/packet_forwarder/blob/master/lora_pkt_fwd/cfg/global_conf.json.PCB_E286.EU868.beacon
- Make a backup of your original local_conf.json file:
cd /opt/ttn-gateway/packet_forwarder/lora_pkt_fwd
cp local_conf.json local_conf.js_orig
- Modify the local_conf.json file:
sudo nano local_conf.json

OPTIONALLY ENABLE GPS

- Add the following lines:

```
"gateway_conf": {  
    :
```

```
    "gps_tty_path": "/dev/ttyAMA0",  
    "fake_gps": false,
```

```
    /* Beaconsing parameters */
```

```
    "beacon_period": 128,  
    "beacon_freq_hz": 869525000,  
    "beacon_datarate": 9,  
    "beacon_bw_hz": 125000,  
    "beacon_power": 14,  
    "beacon_infodesc": 0
```

```
    :
```

```
}
```

Enable GPS

Set beaconsing parameters

OPTIONALLY ENABLE GPS

- This is what my local_conf.json looks like after the changes:

```
{  
  "gateway_conf": {  
    "gateway_ID": "B827EBFFEC74B36",  
    "serv_port_up": 1700,  
    "serv_port_down": 1700,  
    "gps_tty_path": "/dev/ttyAMA0",  
    "fake_gps": false,  
    "ref_latitude": 0,  
    "ref_longitude": 0,  
    "ref_altitude": 0,  
  
    /* Beaconsing parameters */  
    "beacon_period": 128,  
    "beacon_freq_hz": 869525000,  
    "beacon_datarate": 9,  
    "beacon_bw_hz": 125000,  
    "beacon_power": 14,  
    "beacon_infodesc": 0  
  }  
}
```

OPTIONALLY ENABLE GPS

- It is recommended to make changes in the local_conf.json file. If some parameters are defined in both global and local configuration files, the local definition overwrites the global definition.
- Place the GPS antenna outside, with a clear view to the sky.
If the GPS antenna is inside the altitude information will be way off.
- Reboot and login into the Raspberry Pi.
sudo reboot
ssh pi@<ip_address_gateway>

OPTIONALLY ENABLE GPS

- Check the syslog and see if GPS is enabled and see beaconing information.

```
tail -f /var/log/syslog
```

```
INFO: GPS serial port path is configured to "/dev/ttyAMA0"  
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: fake GPS is disabled  
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: [main] TTY port /dev/ttyAMA0 open for GPS synchronization  
:  
INFO: Beacon queued (count_us=250268644, freq_hz=869525000, size=17):  
=> 00 00 80 70 63 49 A4 82 00 00 00 00 00 00 00 00  
INFO: Beacon queued (count_us=378268516, freq_hz=869525000, size=17):  
=> 00 00 00 71 63 49 AC 68 00 00 00 00 00 00 00 00  
INFO: Beacon queued (count_us=506268388, freq_hz=869525000, size=17):  
=> 00 00 80 71 63 49 94 B5 00 00 00 00 00 00 00 00  
:  
### [JIT] ###  
# SX1301 time (PPS): 152268922  
src/jitqueue.c:452:jit_print_queue(): INFO: [jit] queue contains 3 packets:  
src/jitqueue.c:453:jit_print_queue(): INFO: [jit] queue contains 3 beacons:  
src/jitqueue.c:459:jit_print_queue(): - node[0]: count_us=250268644 - type=3  
src/jitqueue.c:459:jit_print_queue(): - node[1]: count_us=378268516 - type=3  
src/jitqueue.c:459:jit_print_queue(): - node[2]: count_us=506268388 - type=3  
### [GPS] ###  
# Valid time reference (age: 0 sec)  
# GPS coordinates: latitude 52.45173, longitude 4.81116, altitude -3 m  
##### END #####
```

Beaconing information

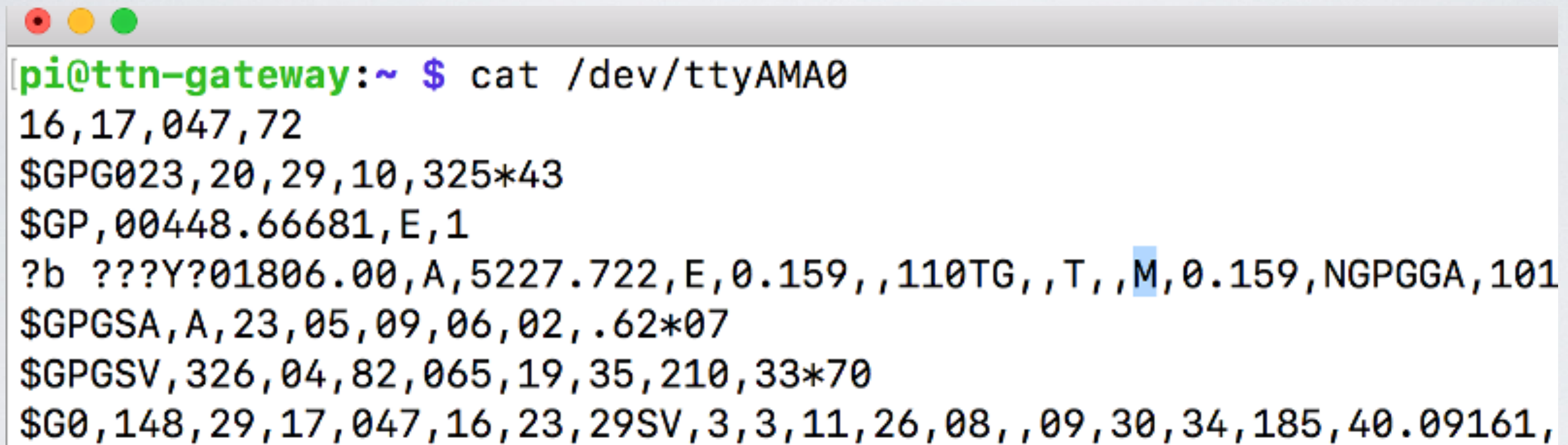
GPS enabled.

Altitude is wrong, antenna was inside.

OPTIONALLY ENABLE GPS

- You can also read GPS data in NMEA (National Marine Electronics Association)

format: **cat /dev/ttyAMA0**



```
[pi@ttn-gateway:~ $ cat /dev/ttyAMA0
16,17,047,72
$GPG023,20,29,10,325*43
$GP,00448.66681,E,1
?b ???Y?01806.00,A,5227.722,E,0.159,,110TG,,T,,M,0.159,NGPGGA,101
$GPGSA,A,23,05,09,06,02,.62*07
$GPGSV,326,04,82,065,19,35,210,33*70
$G0,148,29,17,047,16,23,29SV,3,3,11,26,08,,09,30,34,185,40.09161,
```

- After this test I have disabled GPS by restoring my original local_conf.json file and detached the GPS antenna.

Register the gateway to The Things Network

REGISTER THE GATEWAY TO TTN

- Goto The Things Network <https://www.thethingsnetwork.org/> and login.
- In the console screen select Gateways



APPLICATIONS




GATEWAYS


REGISTER THE GATEWAY TO TTN

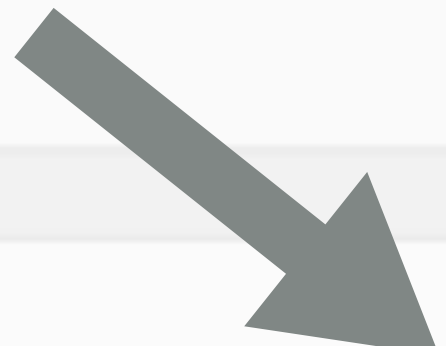
- Select register gateway

Gateways

GATEWAYS

 [register gateway.](#)

 ● not connected [EU_863_870](#)



REGISTER THE GATEWAY TO TTN

- Enter the Gateway EUI, enable legacy packet forwarder and enter a description.

REGISTER GATEWAY

Gateway EUI

The EUI of the gateway as read from the LoRa module

B8 27 EB FF FE C7 4B 36

✓ 8 bytes

I'm using the legacy packet forwarder

Select this if you are using the legacy [Semtech packet forwarder](#).

Description

A human-readable description of the gateway

Mobilefish RAK831 Gateway

✓

REGISTER THE GATEWAY TO TTN

- Select the frequency plan. If GPS is not enabled enter the gateway location.

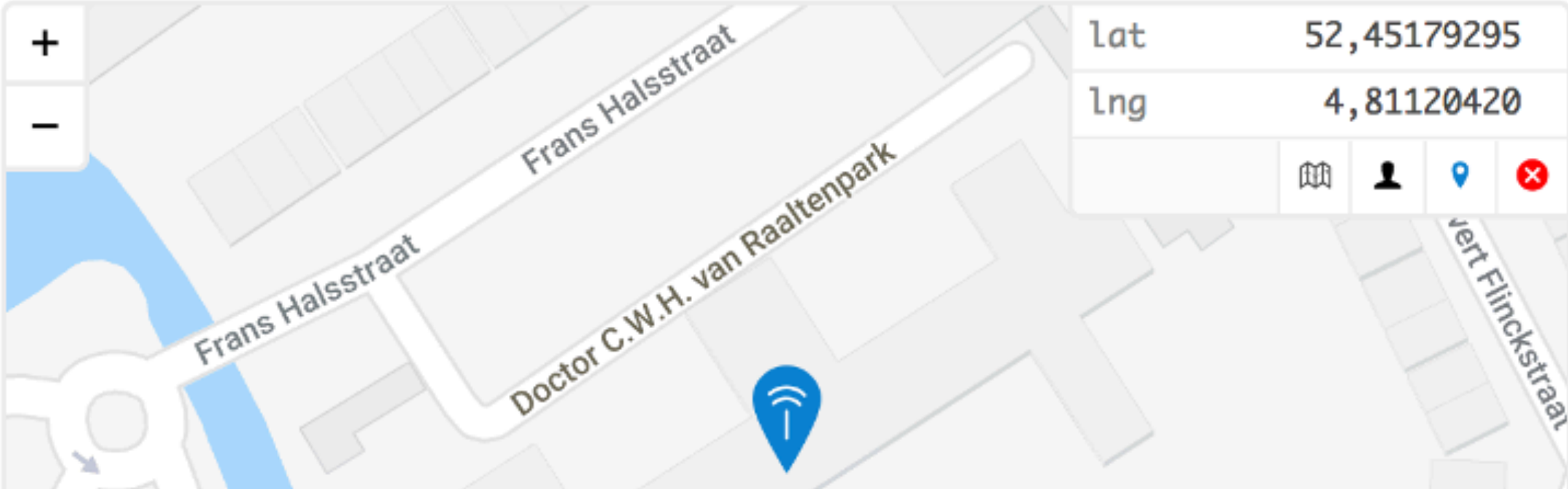
Frequency Plan
The [frequency plan](#) this gateway will use

Europe 868MHz

Router
The router this gateway will connect to. To reduce latency, pick a router that is in a region which is close to the location of the gateway.

ttn-router-eu

Location
The exact location of you gateway. This will be used if your gateway cannot determine its location by itself. Set a location by clicking on the map.



lat 52,45179295
lng 4,81120420

Frans Halsstraat
Doctor C.W.H. van Raaltenpark
vert Fijnckstraat

In the <regional>-global_conf.json file, the router is specified.

REGISTER THE GATEWAY TO TTN

- Select the gateway antenna placement and press the Register Gateway button.

Antenna Placement
The placement of the gateway antenna

indoor outdoor

Cancel



REGISTER THE GATEWAY TO TTN

- After a few minutes, the gateway status is connected.

GATEWAY OVERVIEW ⚙️ [settings](#)

Gateway ID eui-b827ebffec74b36

Description Mobilefish RAK831 Gateway

Owner  **robertlie**  [Transfer ownership](#)

Status ● connected

Frequency Plan Europe 868MHz

Router ttn-router-eu

Gateway is connected to TTN

Enable WiFi

ENABLE WIFI

- By default the gateway WiFi is disabled. Ethernet is preferred above WiFi because you want to reduce additional radio signal noise inside the gateway enclosure. When you login into the Raspberry Pi you see this message:

```
[~$ ssh pi@192.168.1.71
[pi@192.168.1.71's password:
Linux ttn-gateway 4.14.71-v7+ #1145 SMP Fri Sep 21 15:38:35 BST 2018 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jan 10 14:20:48 2019 from 192.168.1.68

Wi-Fi is disabled because the country is not set. ←
Use raspi-config to set the country before use.
```


ENABLE WIFI

- To enable WiFi, open the Raspberry Software Configuration Tool:
sudo raspi-config
- Select menu: **4 Localisation Options**
- Select menu: **I4 Change Wi-fi Country**
- Select the country in which the Pi is used. In my case I select the Netherlands.
Select menu: **NL Netherlands**
- Save the changes

ENABLE WIFI

- Edit file wpa_supplicant.conf

```
cd /etc/wpa_supplicant  
sudo nano wpa_supplicant.conf
```

- Add the last 4 lines and enter your router ssid and password:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev  
update_config=1  
country=NL
```

```
network={  
ssid="YOUR_ROUTER_SSID"  
psk="YOUR_ROUTER_PASSWORD"  
}
```

ENABLE WIFI

- Shutdown Raspberry Pi
sudo shutdown
- After you have entered the shutdown command, wait a few minutes and pull out the power adapter and the Ethernet cable.
- Wait a few seconds and power up the gateway.
- Your router will assign a new IP address to the gateway.
Go back in this tutorial and watch how to get the gateway IP address.

ENABLE WIFI



- Check if you can still login into the Raspberry Pi with the new IP address:
ssh pi@ <ip_address_gateway>
- Also check if the gateway is connected to The Things Network.
Go to The Things Network console, select the RAK gateway and check its status.

GATEWAY OVERVIEW

⚙️ [settings](#)

Gateway ID eui-b827ebffec74b36

Description Mobilefish RAK831 Gateway

Owner  **robertlie**  [Transfer ownership](#)

Status ● connected

Frequency Plan Europe 868MHz

Router ttn-router-eu

**Testing ABP and OTAA
and
uplink and downlink messages**

TESTING ABP, OTAA, UPLINK AND DOWNLINK MESSAGES

- If the RAK831 Pilot Gateway is up and running and the gateway is registered with The Things Network you may want to conduct several tests:
 - Test if the gateway can handle Activation-By-Personalisation, use the procedure demonstrated in [Tutorial 24](#).
 - Test if the gateway can handle Over-The-Air-Activation, use the procedure demonstrated in [Tutorial 25](#).
 - Test if the gateway can handle uplink and downlink messages, use the procedure demonstrated in [Tutorial 25](#).

Range tests

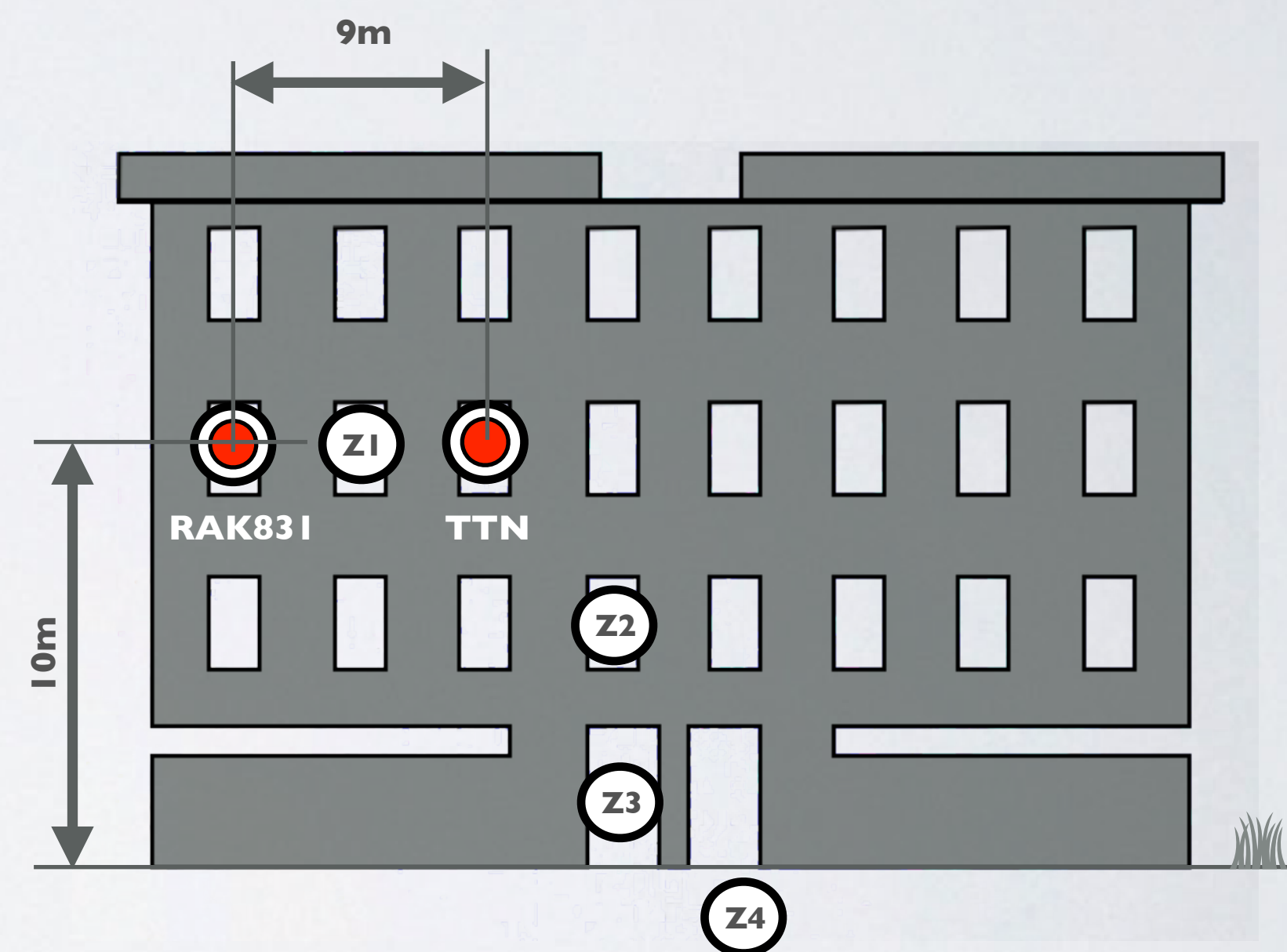
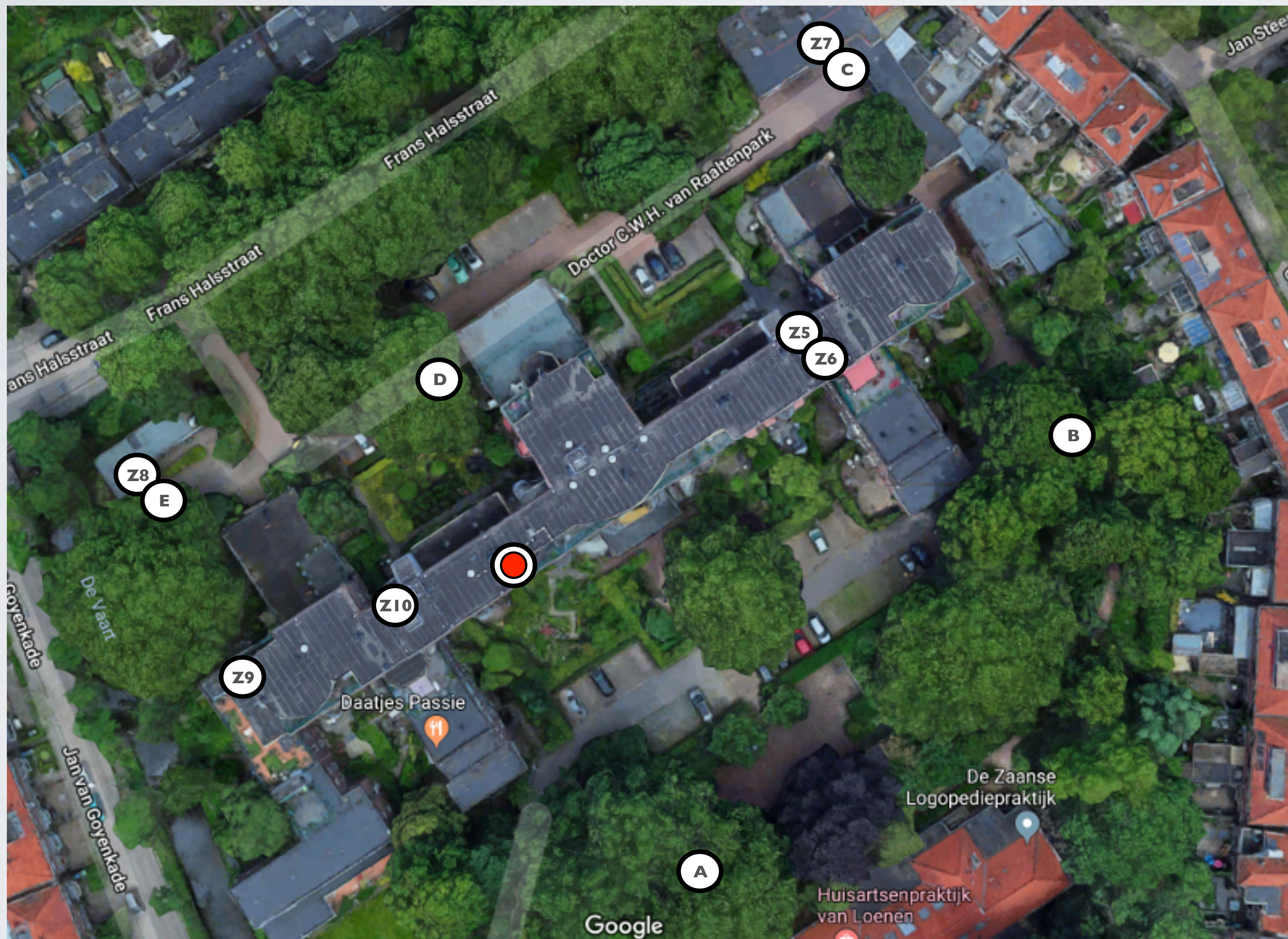
RANGE TESTS

- As explained in [Tutorial 7](#), if a gateway antenna is outside, at a high location, and there is a direct line of sight with an end device then this gateway can receive signals from this end device over a large distance.
- But what if the gateway antenna is inside, approximately 10 m above ground and surrounded by buildings what will the range be?
- I conducted a simple test where I used two gateways the RAK831 Pilot Gateway and the Things Gateway, both are designed for indoor usage.
- As an end device I used The Things Node and went to different locations and recorded the received signals from both gateways. In [Tutorial 27](#) is explained how to retrieve and store sensor data from The Things Network.

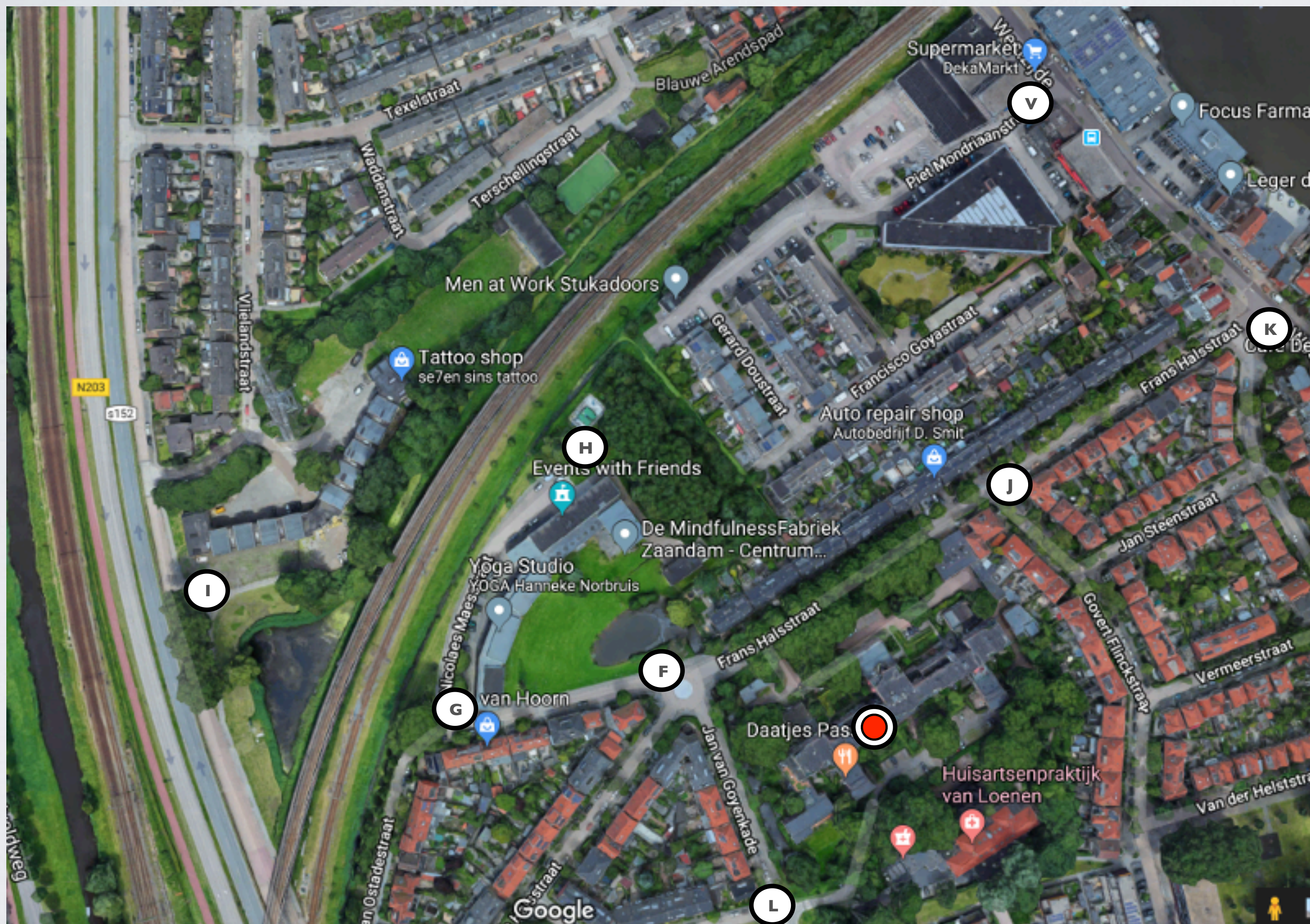
RANGE TESTS

- For the range test both gateways were placed in front of a window, separated 9m from each other and both are located 10m above ground.
- All measure points starting with Z means the end device was inside a building.
- All recorded data can be found at:
https://www.mobilefish.com/download/lora/range_tests.txt
- All measure points can also be found in a shared Google Map:
<https://drive.google.com/open?id=1gq9WIAmmGDnAcrsEaCvpUXYSXRCBdq&usp=sharing>

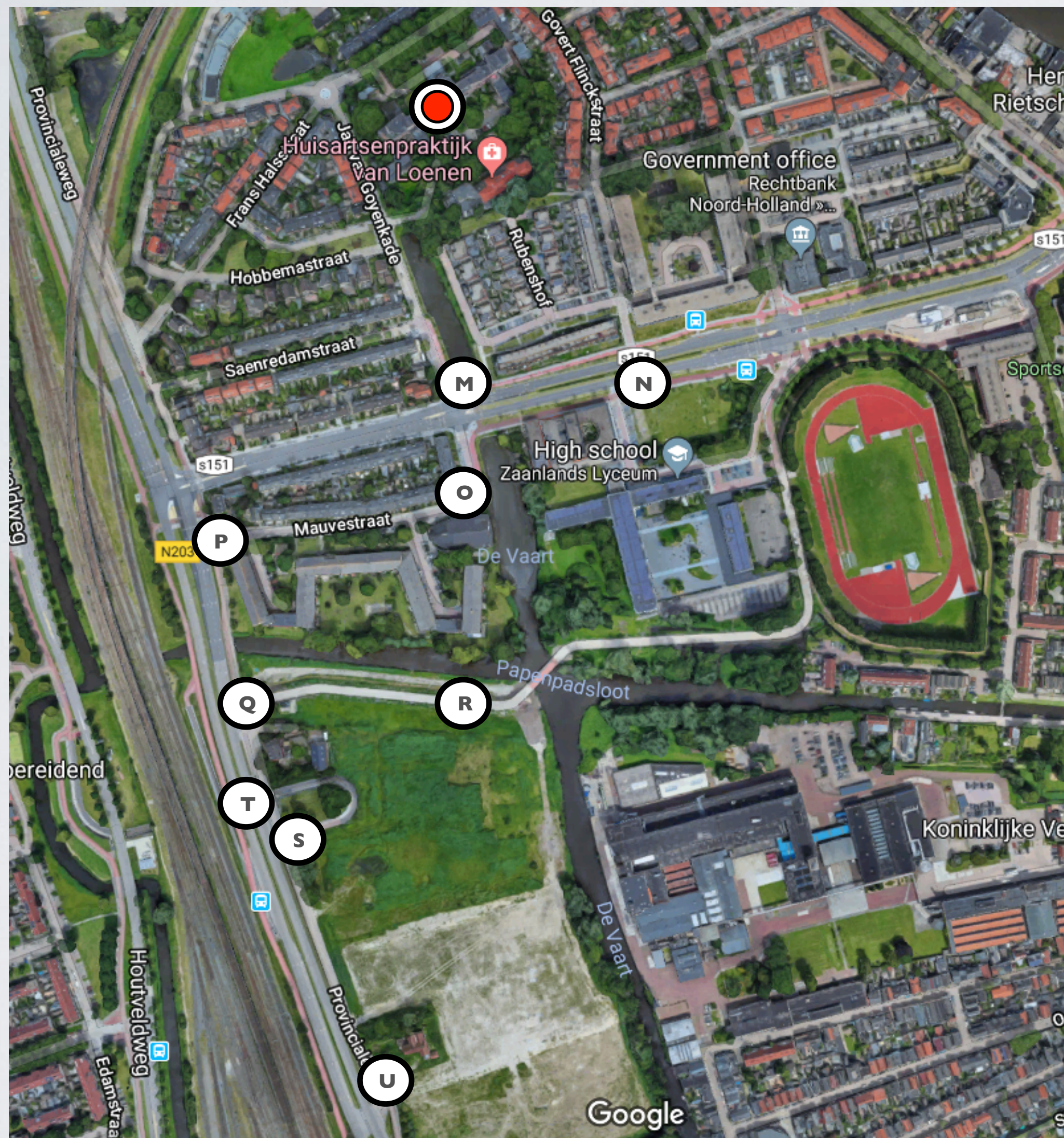
RANGE TESTS



RANGE TESTS



RANGE TESTS








RANGE TESTS

- If the RAK831 Pilot Gateway is placed in front of a window, monitor the gateway temperature. Especially in the summer period (high ambient temperature) and when there is direct sunlight on the gateway. To monitor the Raspberry Pi CPU core temperature, use command:













```
/opt/vc/bin/vcgencmd measure_temp
```

- It is recommended to keep the RAK831 Pilot Gateway operating temperature between 0°C to 70°C.










RANGE TESTS

Location		RAK831 Gateway (rssi)	TTN Gateway (rssi)
Z1, ~4m		-63	-83
Z2, 1st floor		-87	-78
Z3, Ground floor		-79	-84
Z4, Basement		-107	-105
A, ~50m		-66	-75
B, ~72m		-85	-93
C, ~78m		-101	-103
D, ~30m		-90	-79
E, ~50m		-102	-99
F, ~94m		-97	-102
G, ~179m		-111	-111

RANGE TESTS

Location		RAK831 Gateway (rssi)	TTN Gateway (rssi)
H, ~171m		No measurement	-113
I, ~291m		-110	-111
J, ~100m		-112	-114
K, ~200m		-112	No measurement
L, ~104m		-83	-89
M, ~198m		-85	-106
N, ~241m		-103	-109
O, ~271m		-88	-108
P, ~317m		-106	-113
Q, ~434m		No measurement	-111
R, ~416m		-110	No measurement
S, ~519m		-113	-113

RANGE TESTS

Location		RAK831 Gateway (rssi)	TTN Gateway (rssi)
T, ~502m		No measurement	No measurement
U, ~644m		-114	No measurement
V, ~262m		-113	-112
Z5, ~46m (ground fl.)		-106	-108
Z6, ~46m (1st fl.)		-102	-115
Z7, ~78m		-113	-113
Z8, ~49m		-93	-99
Z9, ~38m (ground fl.)		-112	-109
Z10, ~15m (ground fl.)		-87	-99

RANGE TESTS

- I hope that these ranges tests give you a realistic picture of the actual gateway range when you place the gateway antenna in a building at a height of about 10 m and the gateway is surrounded by buildings.
- After carrying out these range tests, I conclude that in my case, both gateways have a range of about 520 m.
- Please note: This is not a scientific test, it is just a simple test to give me an idea of the possible range of my gateways.
- To increase the gateway range, if you can: 1. Place the gateway antenna outside, 2. Place the gateway antenna as high as possible and 3. Have a direct line of sight with the end device.