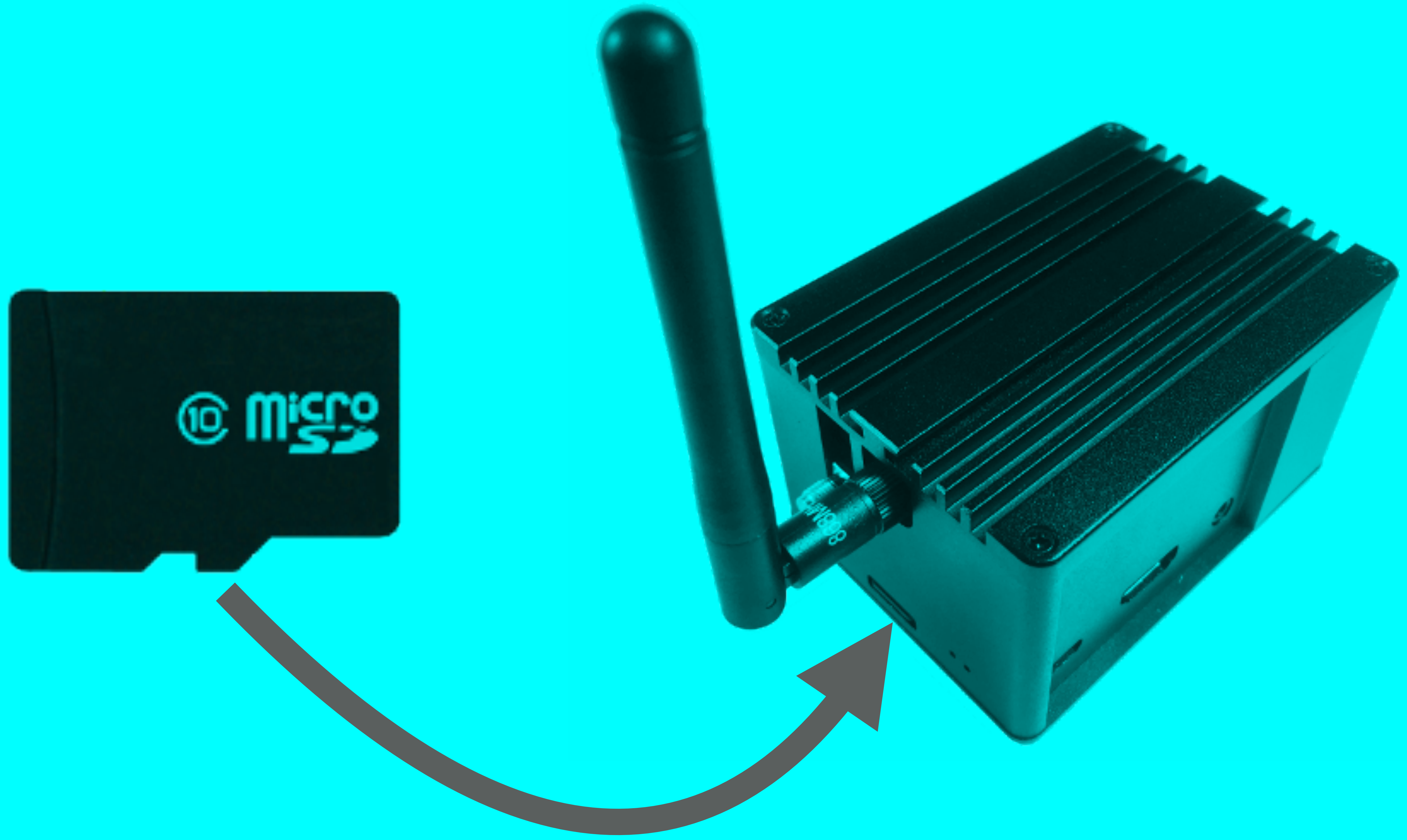
mobilefish.com

LORA / LORAWAN TUTORIAL 28.1

Installing Semtech UDP Packet Forwarder for the RAK831 Pilot Gateway



v1.0.0

INTRO

- A micro SD card is shipped with the RAK831 Pilot Gateway with all software pre-installed as mentioned in Tutorial 28.
- In this tutorial I will show you how to install all the required software on a micro SD card.
- The result is a bootable micro SD card which can be used in the RAK831 Pilot Gateway.

Install Raspbian Lite on micro SD card

DOWNLOAD RASPBIAN LITE

- Download Raspbian Lite.
  Lite is a minimal version of the Raspbian image for the Raspberry Pi.
  The Lite version has only a command line interface (CLI) and no desktop or GUI of any kind.This means fewer modules will load with the kernel thus less of the Raspberry Pi's resources are used.
  https://www.raspberrypi.org/downloads/raspbian/

- After the Raspbian Lite is downloaded, verify the SHA256 checksum.
  The checksum is a hash number and is used to verify the integrity of the file.
  macOS: **`shasum -a 256 <file>`**
  Linux: **`sha256sum <file>`**
  Windows 10: **`certutil -hashfile <file> SHA256`**

WRITE IMAGE TO MICRO SD CARD

- Download and install Etcher.
  Etcher is a tool for Windows, Mac and Linux for flashing images to SD cards and USB drives.
  https://www.balena.io/etcher/

- Flash the Raspbian Lite onto the micro SD card using Etcher.

- Note: If the micro SD card is not formatted, Etcher will format it before writing and verifying the image.

ADD SSH FILE ON MICRO SD CARD

- The micro SD Card has a **`/boot`** partition.
  On macOS, see: **`/Volumes/boot`**

- Goto the **`/boot`** partition and add an empty file with name: **`ssh`**
  When the Raspberry Pi is rebooted it will look for file "ssh".
  When found, it will enable SSH.
  By default SSH is disabled.
  After the reboot, the ssh file will be deleted.

Get the gateway IP address

GET THE GATEWAY IP ADDRESS

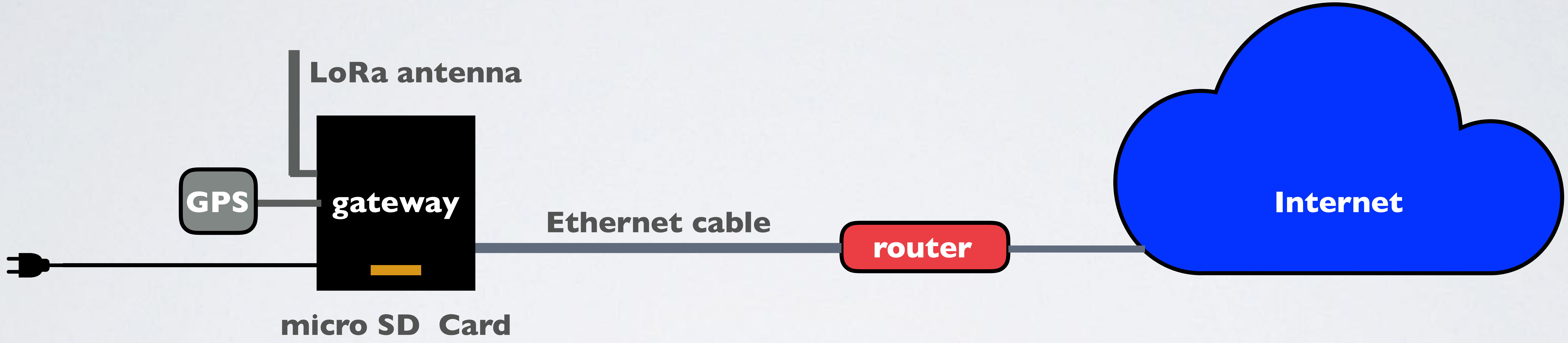
- Insert the micro SD card into the Raspberry Pi.

- Connect the LoRa antenna to the gateway.
  Never power up the gateway without the LoRa antenna, otherwise this can cause damage to your gateway.

- It is not required to connect the GPS antenna because GPS is by default disabled as explained in Tutorial 28.

GET THE GATEWAY IP ADDRESS

- Connect the gateway to a router using an Ethernet cable.



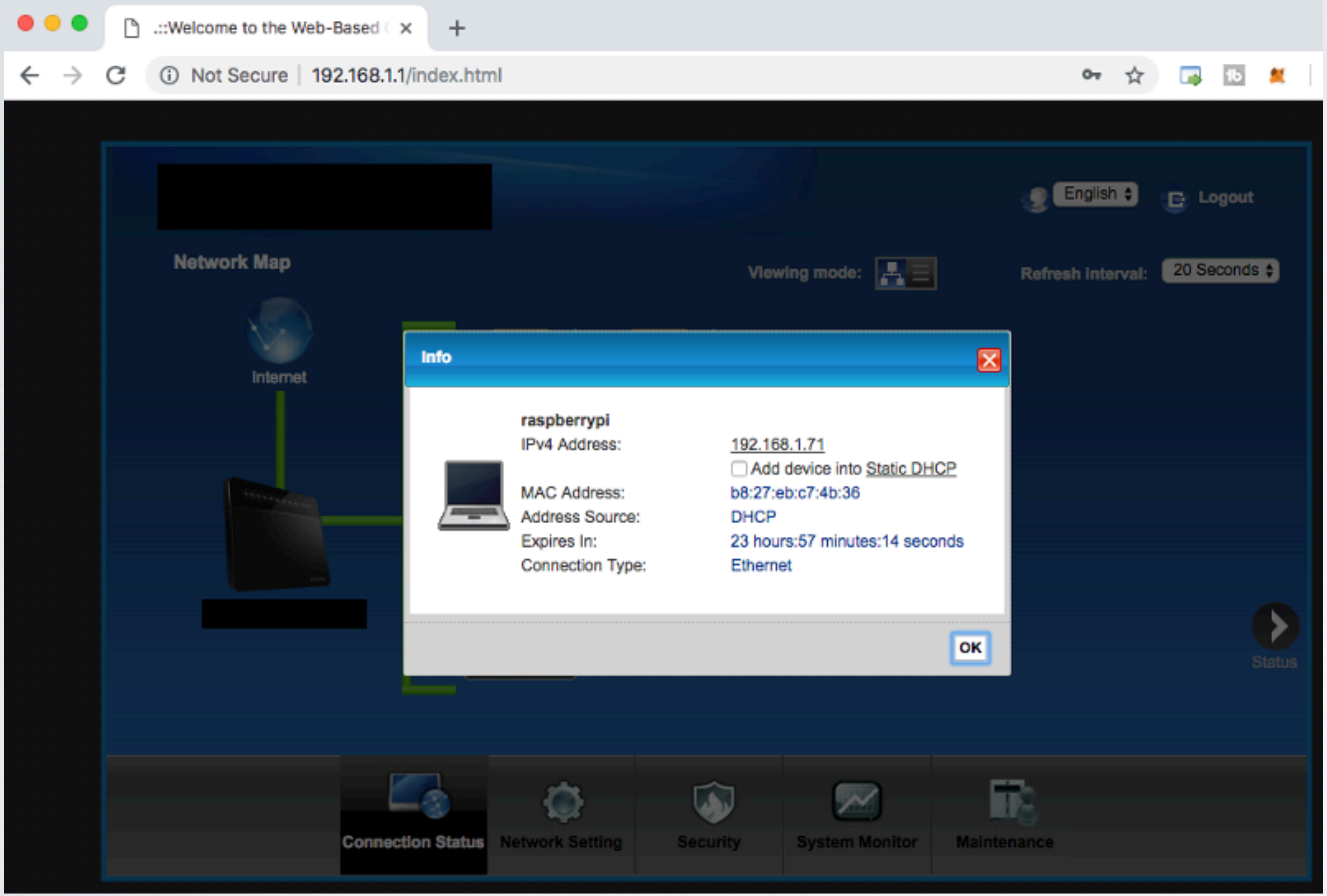
- Power up the gateway using the included power adapter. Do not use a power adapter with less than 2A output.

- By default the Raspberry Pi has the hostname **`raspberrypi`**

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 (**raspberrypi**) and write down its IP address.

GET THE GATEWAY IP ADDRESS



Info

raspberrypi

IPv4 Address: 192.168.1.71

□ Add device into Static DHCP

MAC Address: b8:27:eb:c7:4b:36

Address Source: DHCP

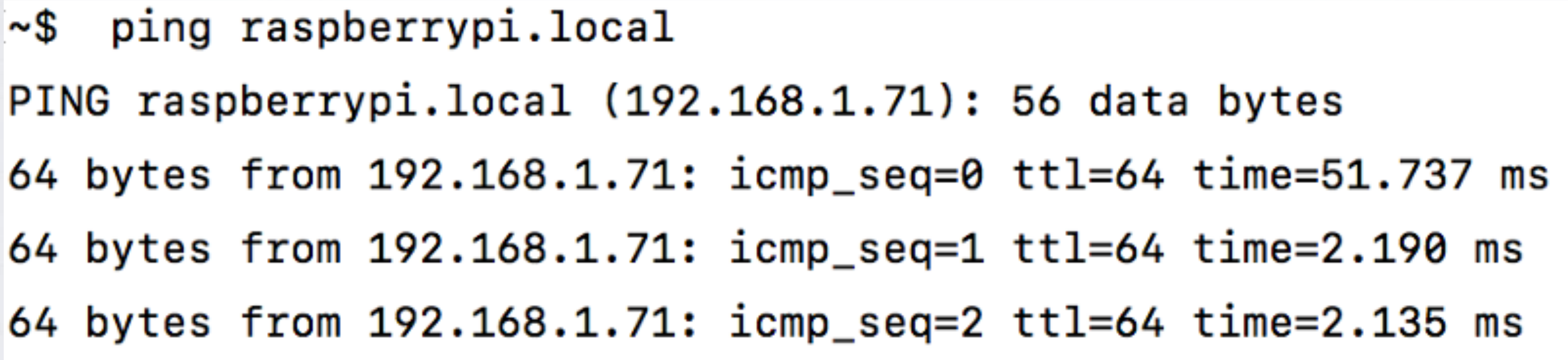
Expires In: 23 hours:57 minutes:14 seconds

Connection Type: Ethernet

OK

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 raspberrypi.local`**

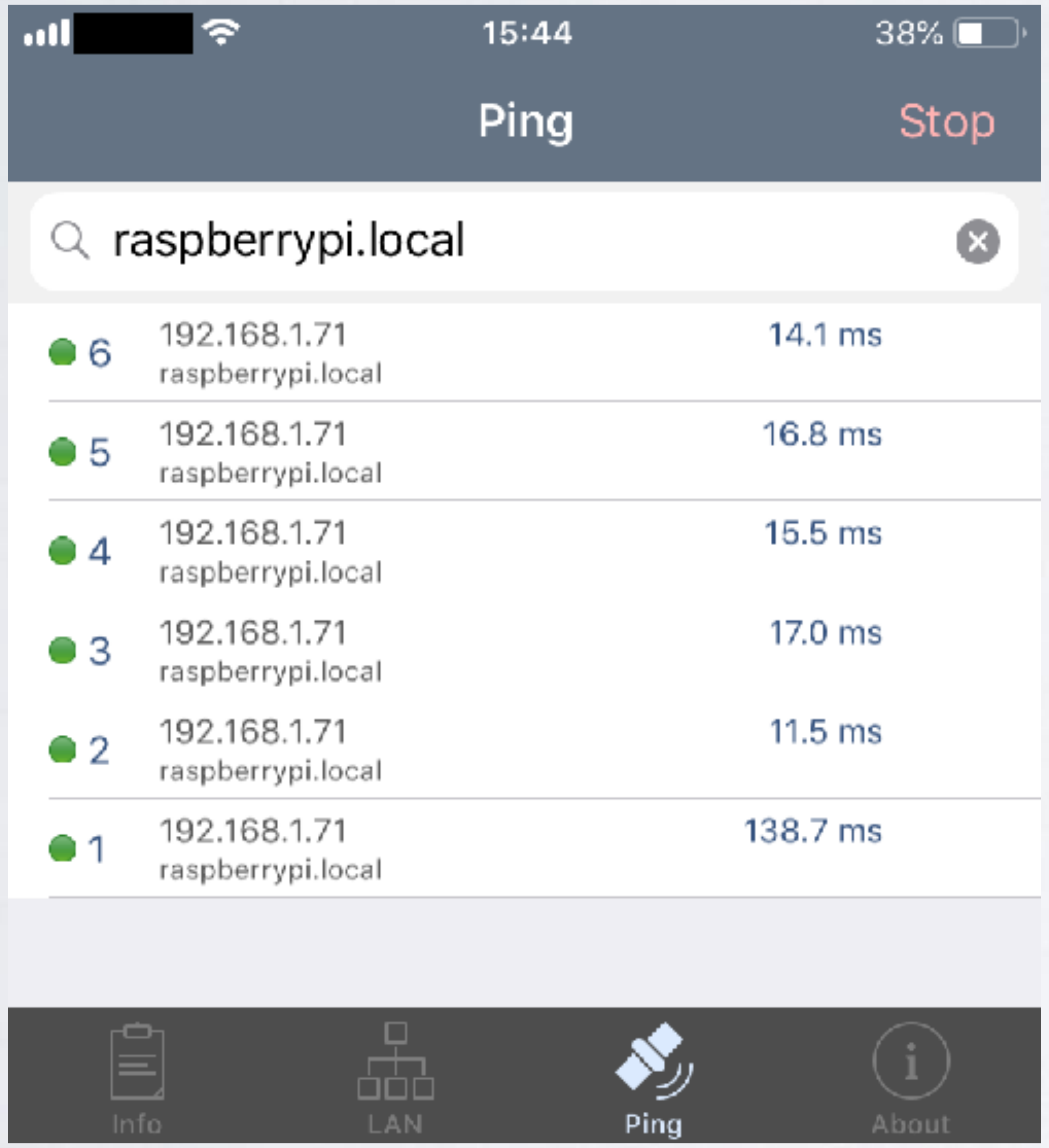


```
~$  ping raspberrypi.local
PING raspberrypi.local (192.168.1.71): 56 data bytes
64 bytes from 192.168.1.71: icmp_seq=0 ttl=64 time=51.737 ms
64 bytes from 192.168.1.71: icmp_seq=1 ttl=64 time=2.190 ms
64 bytes from 192.168.1.71: icmp_seq=2 ttl=64 time=2.135 ms
```

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: **raspberrypi.local**



[redacted] 15:44 38%

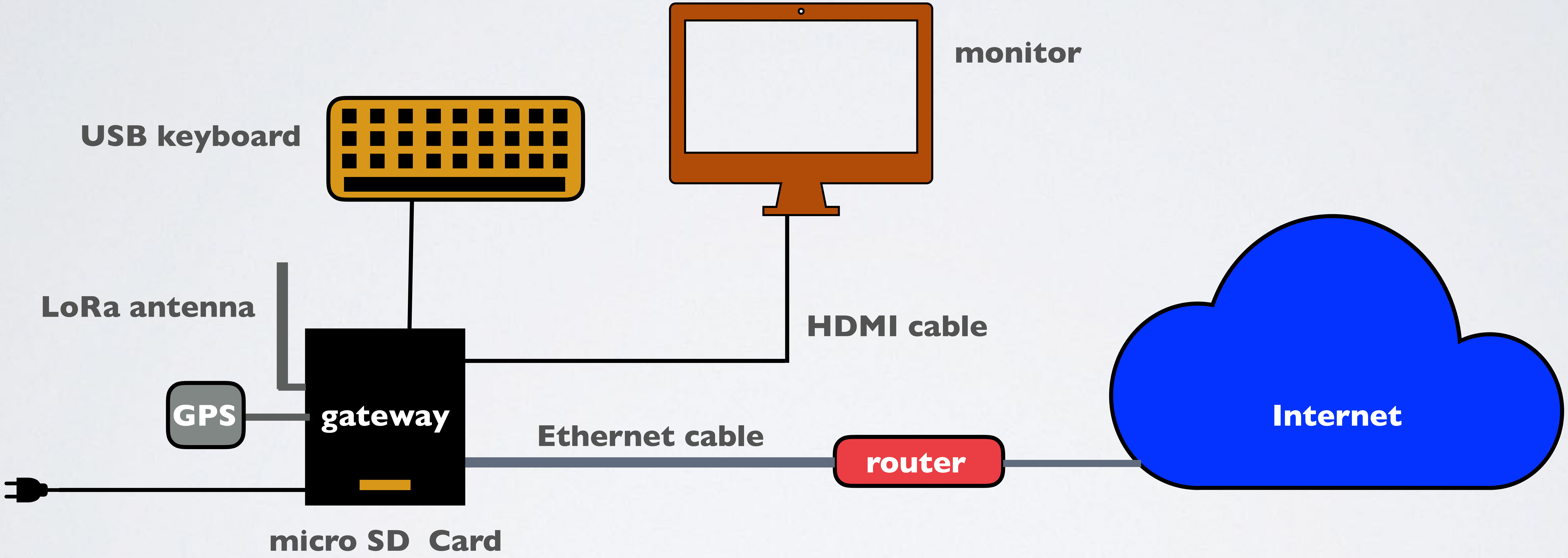
Ping Stop

raspberrypi.local

```
6  192.168.1.71        14.1 ms
   raspberrypi.local
5  192.168.1.71        16.8 ms
   raspberrypi.local
4  192.168.1.71        15.5 ms
   raspberrypi.local
3  192.168.1.71        17.0 ms
   raspberrypi.local
2  192.168.1.71        11.5 ms
   raspberrypi.local
1  192.168.1.71       138.7 ms
   raspberrypi.local
```

GET THE GATEWAY IP ADDRESS

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



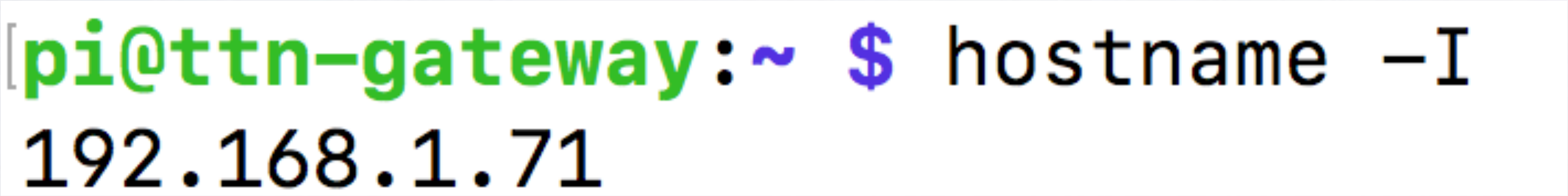
GET THE GATEWAY IP ADDRESS

- You will see the login prompt.
  The default username=**`pi`** and password=**`raspberry`**

- Login with above mentioned credentials.

- Enter command: **`hostname -I`**

- The IP address will be shown, for example: 192.168.1.71



```
[pi@ttn-gateway:~ $ hostname -I
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=**raspberry**

CHECK CURRENT RASPBERRY PI CONFIGURATION

- Check time zone.

```
date
```

- Check the disk space. The full space of the micro SD card is used.

```
df -h
```

- Check if SPI is enabled.

```
ls /dev/*spi*
```

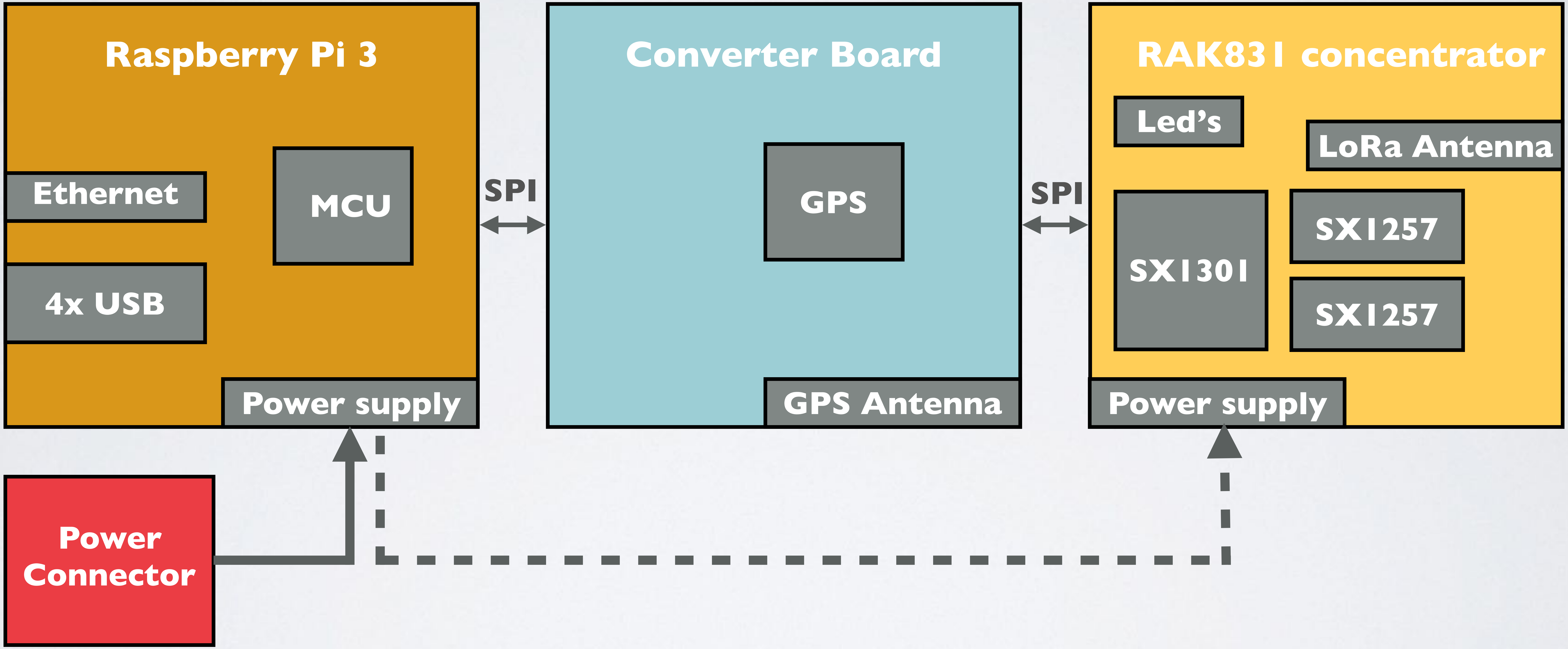
SPI is not enabled if you see:

```
ls: cannot access '/dev/*spi*': No such file or directory
```

- Check current language settings.

```
locale
```

RAK831 PILOT GATEWAY



Configure Raspberry Pi

CONFIGURE RASPBERRY PI

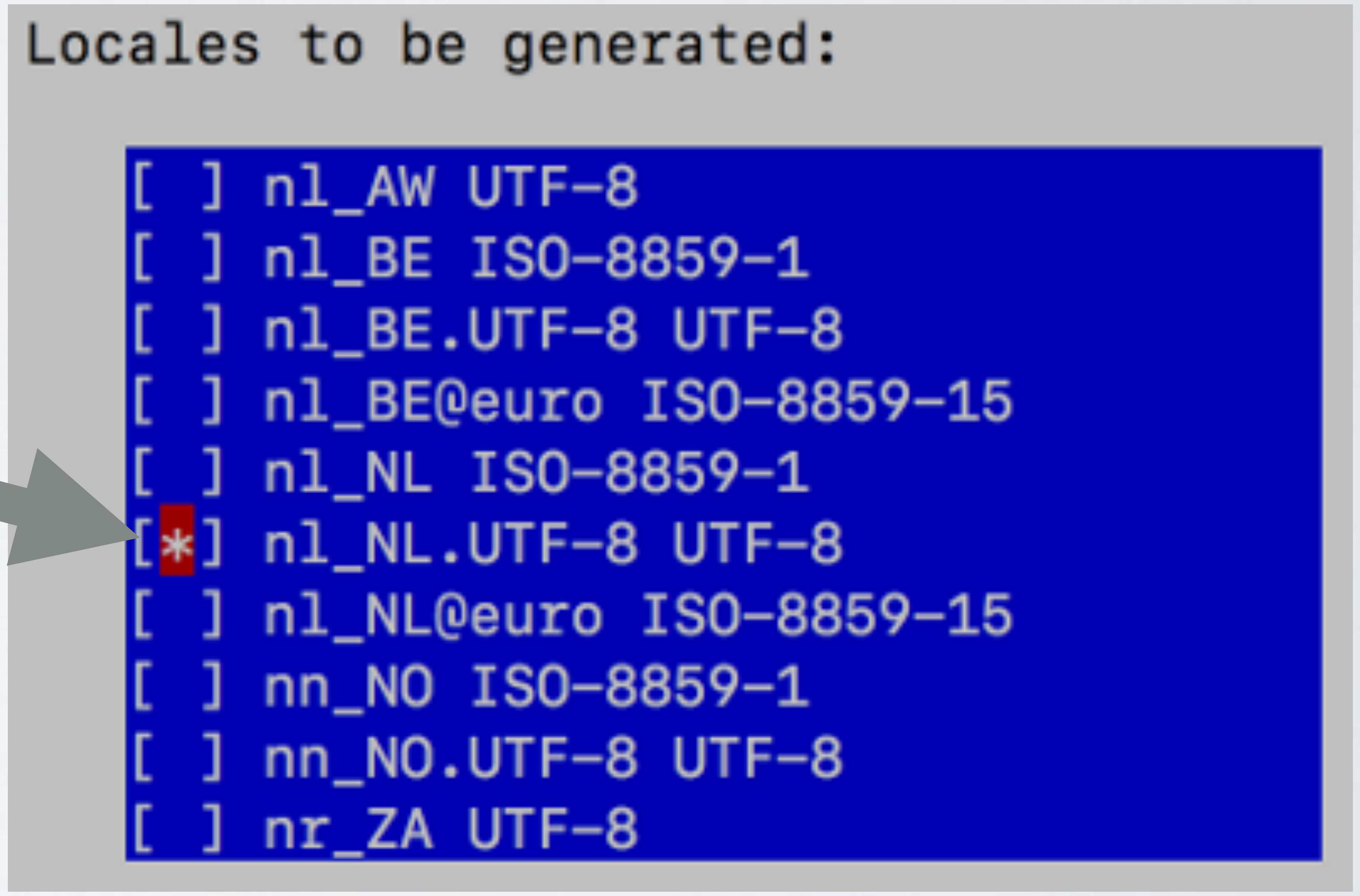
- The Raspberry Pi needs to be configured. I will change:
  - user pi default password
  - the language
  - the time zone
  - and enable SPI
- Open the Raspberry Pi Software Configuration Tool:

```
sudo raspi-config
```

CONFIGURE RASPBERRY PI

- Change user pi password.
  Select menu: **`1 Change User Password`**
  Note: For security reasons you must change the user pi default password.

- Optionally change the language (default is English: en_GB.UTF-8).
  Select menu: **`4 Localisation Options`**
  Select menu: **`I1 Change Locale`**
  Scroll the list and use the scrollbar to select.
  Recommended: xx_XX.**UTF-8 UTF-8**
  Press Enter key when done with the selection.

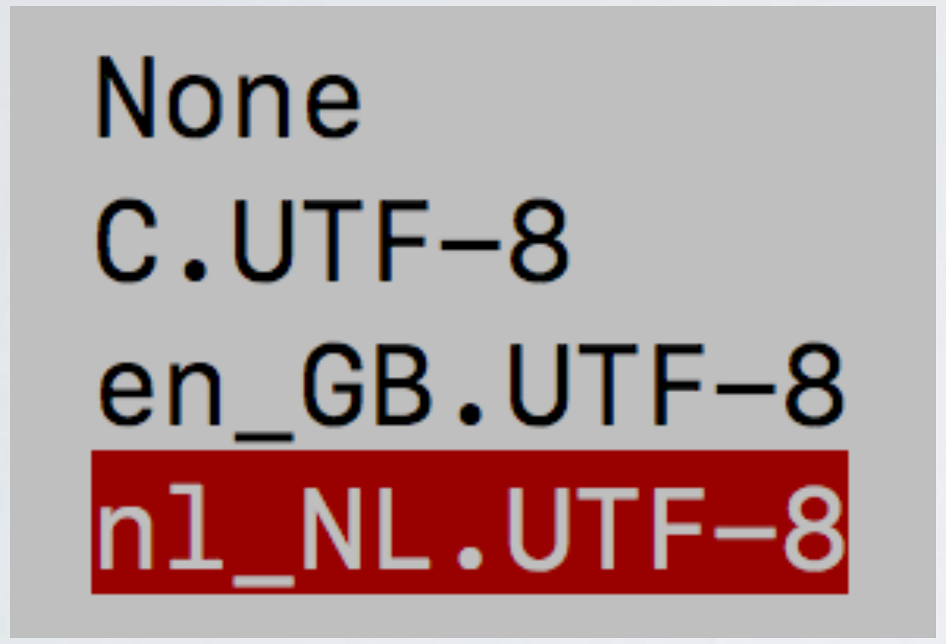


```
Locales to be generated:

    [ ] nl_AW UTF-8
    [ ] nl_BE ISO-8859-1
    [ ] nl_BE.UTF-8 UTF-8
    [ ] nl_BE@euro ISO-8859-15
    [ ] nl_NL ISO-8859-1
    [*] nl_NL.UTF-8 UTF-8
    [ ] nl_NL@euro ISO-8859-15
    [ ] nn_NO ISO-8859-1
    [ ] nn_NO.UTF-8 UTF-8
    [ ] nr_ZA UTF-8
```

CONFIGURE RASPBERRY PI

In the next screen select the default locale for the system environment. Select for example nl_NL.UTF-8 and press the Enter key.



```
None
C.UTF-8
en_GB.UTF-8
nl_NL.UTF-8
```

CONFIGURE RASPBERRY PI

- Change the timezone.
  Select menu: **`4 Localisation Options`**
  Select menu: **`I2 Change Timezone`**
  For example:
  Geographic area: **`Europe`**
  Time zone: **`Amsterdam`**

CONFIGURE RASPBERRY PI

- Enable SPI.
  Select menu: **`5 Interfacing Options`**
  Select menu: **`P4 SPI`**

- This will enable SPI (Serial Peripheral Interface) on the Raspberry Pi.
  An SPI bus is used to send data between microcontroller(s) and peripherals.
  In this case sending data to and from the Raspberry Pi and RAK831 concentrator via the Converter Board.

- Exit the Raspberry Pi configuration tool.

MODIFY .BASHRC FILE

- If you changed the language using the raspi-config tool, one extra change is needed.
  Goto pi's home directory and edit the .bashrc file

  ```
  cd ~
  nano .bashrc
  ```

  At the end of the .bashrc file add the following line with YOUR language choice:

  ```
  export LC_ALL=nl_NL.UTF-8
  ```

  Save the file

REBOOT AND LOGIN RASPBERRY PI

- Reboot the Raspberry Pi:

  ```
  sudo reboot
  ```

- Again access the Raspberry Pi:

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

  password=**`<your_new_password>`**

VERIFY CHANGES

- Check time zone:

```
date
```

- Check the disk space:

```
df -h
```

- Check if SPI is enabled:

```
ls /dev/*spi*
```

You will see:

```
/dev/spidev0.0   /dev/spi/dev0.1
```

- Check current language settings.

```
locale
```

Install software

INSTALL SOFTWARE

- Upgrade the Raspberry Pi packages:

```
sudo apt-get update && sudo apt-get upgrade -y
```

- Install Git.
  Git is a distributed version control system.

```
sudo apt-get install git
```

INSTALL RAK831-LORAGATEWAY-RPI

- I have forked https://github.com/RAKWireless/RAK831-LoRaGateway-RPi and simplified the installation procedure.

- The repository https://github.com/robertlie/RAK831-LoRaGateway-RPi contains just a few files. The install.sh script:
  - Creates the gateway EUI.
  - Allows the user to set the gateway hostname.
  - Allows the user to select the region the gateway will operate in. Dependant on the selected region the correct global_conf.json is copied from the configuration_files folder.
  - The local_conf.json is copied from the configuration_files folder and the gateway EUI is set in this file.

INSTALL RAK831-LORAGATEWAY-RPI

- Allows the user to set the gateway latitude and longitude coordinates and its altitude.
- Installs the Semtech LoRa library and the Semtech UDP Packet Forwarder and build both packages.
- Makes the packet_forwarder a service, which means when the Raspberry Pi boots the packet_forwarder is started.
- Disables the onboard Raspberry Pi bluetooth.

- Before you start with the installation procedure, you must know which frequency plan to use in your country. See the list of frequency plans by country list: https://www.thethingsnetwork.org/docs/lorawan/frequencies-by-country.html

- The frequency plans can be found at: https://www.thethingsnetwork.org/docs/lorawan/frequency-plans.html

INSTALL RAK831-LORAGATEWAY-RPI

- If you know the frequency plan, look at the table and find the corresponding Region.

Region	Frequency Plan
AS1	AS920-923
AS2	AS923-925
AU	AU915-928
CN	CN470-510
EU	EU863-870
IN	IN865-867
KR	KR920-923
RU	-
US	US902-928

INSTALL RAK831-LORAGATEWAY-RPI REPOSITORY

- Installation procedure:
```
git clone https://github.com/robertlie/RAK831-LoRaGateway-RPi ~/rak831-loragateway
```

- This repository is installed in: **`/home/pi/rak831-loragateway`**

EXECUTE INSTALL SCRIPT

- Execute the install script:

```
cd ~/rak831-loragateway
sudo ./install.sh
```

- The following is displayed. Press Enter to keep the default value or change it:

```
Host name [ttn-gateway]: <Enter>
Region AS1, AS2, AU, CN, EU, IN, KR, RU, US [EU]: EU
Latitude [0]: <Enter>
Longitude [0]: <Enter>
Altitude [0]: <Enter>
```

- I will keep the default hostname, latitude, longitude and altitude.
  The latitude, longitude and altitude will be set later in TTN console.

INSTALL LORA LIBRARY AND PACKET FORWARDER

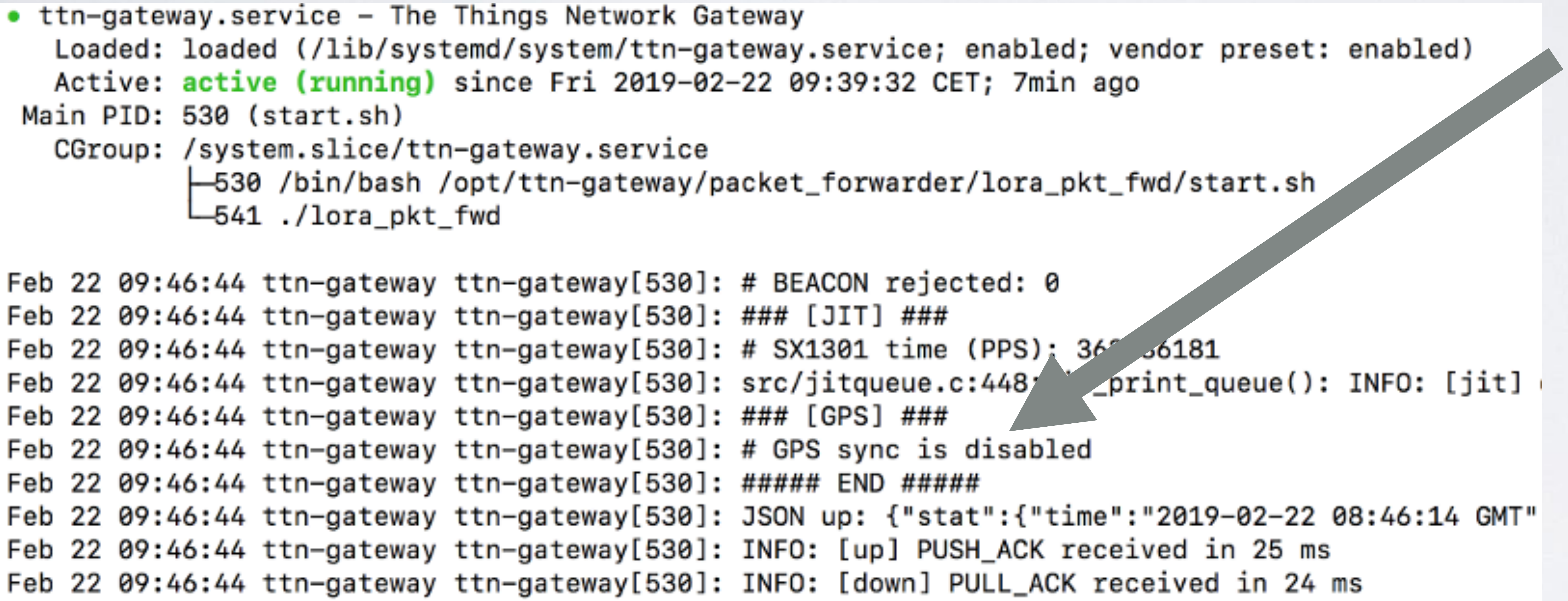
- As mentioned earlier, the install.sh script installs the following git repositories and build these packages.

  - 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`

CHECK TTN-GATEWAY SERVICE STATUS

- Check ttn-gateway service status:
  `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 Fri 2019-02-22 09:39:32 CET; 7min ago
 Main PID: 530 (start.sh)
   CGroup: /system.slice/ttn-gateway.service
           ├─530 /bin/bash /opt/ttn-gateway/packet_forwarder/lora_pkt_fwd/start.sh
           └─541 ./lora_pkt_fwd

Feb 22 09:46:44 ttn-gateway ttn-gateway[530]: # BEACON rejected: 0
Feb 22 09:46:44 ttn-gateway ttn-gateway[530]: ### [JIT] ###
Feb 22 09:46:44 ttn-gateway ttn-gateway[530]: # SX1301 time (PPS): 36[illegible]6181
Feb 22 09:46:44 ttn-gateway ttn-gateway[530]: src/jitqueue.c:448:[illegible]_print_queue(): INFO: [jit]
Feb 22 09:46:44 ttn-gateway ttn-gateway[530]: ### [GPS] ###
Feb 22 09:46:44 ttn-gateway ttn-gateway[530]: # GPS sync is disabled
Feb 22 09:46:44 ttn-gateway ttn-gateway[530]: ##### END #####
Feb 22 09:46:44 ttn-gateway ttn-gateway[530]: JSON up: {"stat":{"time":"2019-02-22 08:46:14 GMT"
Feb 22 09:46:44 ttn-gateway ttn-gateway[530]: INFO: [up] PUSH_ACK received in 25 ms
Feb 22 09:46:44 ttn-gateway ttn-gateway[530]: INFO: [down] PULL_ACK received in 24 ms
```

Check if there are no errors and GPS is disabled.

Also see:
`/var/log/syslog`

global_conf.json & local_conf.json

GLOBAL_CONF.JSON & LOCAL_CONF.JSON

- The RAK831 Pilot Gateway can be connected to any LoRa network servers.

- In this tutorial the RAK831 Pilot Gateway will be connected to The Things Network server.

- The Semtech Packet Forwarder is configured via a file called global_conf.json and if provided an additional file called local_conf.json.

- The global_conf.json is the main configuration file and contains for example the LoRa network server address, which uplink and downlink ports to use, which frequencies to use and the TX power LookUp Table (LUT).

- The local_conf.json file contains more gateway specific parameters.
  The local_conf.json will override the settings in the global_conf.json.

GLOBAL_CONF.JSON & LOCAL_CONF.JSON

- The **~/rak831-loragateway/install.sh** script creates the global_conf.json and local_conf.json files in this folder:
  /opt/ttn-gateway/packet_forwarder/lora_pkt_fwd

- Several other global_conf.json file examples can be found in this folder:
  /opt/ttn-gateway/packet_forwarder/lora_pkt_fwd/cfg
  but these files are not used.

- The https://github.com/robertlie/RAK831-LoRaGateway-RPi/blob/master/configuration_files/README.md file explains where the global configuration files originates from and what modifications were made to these files.

GLOBAL_CONF.JSON & LOCAL_CONF.JSON

- In Tutorial 28 is explained which parameters to set in the local_conf.json file to enable GPS and for beaconing. In the local_conf.json these parameters are commented out. Uncomment these parameters if needed.

- In Tutorial 30 you can find an explanation of all the parameters used in the global_conf.json and local_conf.json files.

- When the global_conf.json or local_conf.json file is changed, always restart the gateway.

GLOBAL_CONF.JSON & LOCAL_CONF.JSON

- Before restarting the gateway, optionally empty the /var/log/syslog:

```
sudo su
> /var/log/syslog
exit
```

- Restart the ttn-gateway service:

```
sudo service ttn-gateway restart
```

- Check ttn-gateway service status:

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

- Check system log for errors:

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

NEXT STEPS….

- The gateway is now running without errors.
- Next steps:
  - Register the gateway to The Things Network, watch: https://youtu.be/bea7g5isD0w?t=1779
  - Optionally enable WiFi, watch: https://youtu.be/bea7g5isD0w?t=1844