RAK831 Pilot Gateway User Manual V1.0

Shenzhen Rakwireless Technology Co., Ltd
www.rakwireless.com
info@rakwireless.com

Contents

1. Build your own RAK831 Pilot gateway

This documents are from a workshop held during the Things Network Conference 2018, in this workshop, many developers build a gateway based on RAK831 and a Raspberry Pi model 3. We can use this tutorial as the reference to understand how to connect RAK831 Pilot gateway to TTN.

NOTE: Never power on the gateway without the antenna connected as shown in the picture

2. Preparations

The workshop requires some tools to be available on your system. Check the list below and follow the link to the installation instructions if you do not have the tool installed.

GIT command line client. windows, OSX, for Linux use the tools from your distribution to install.

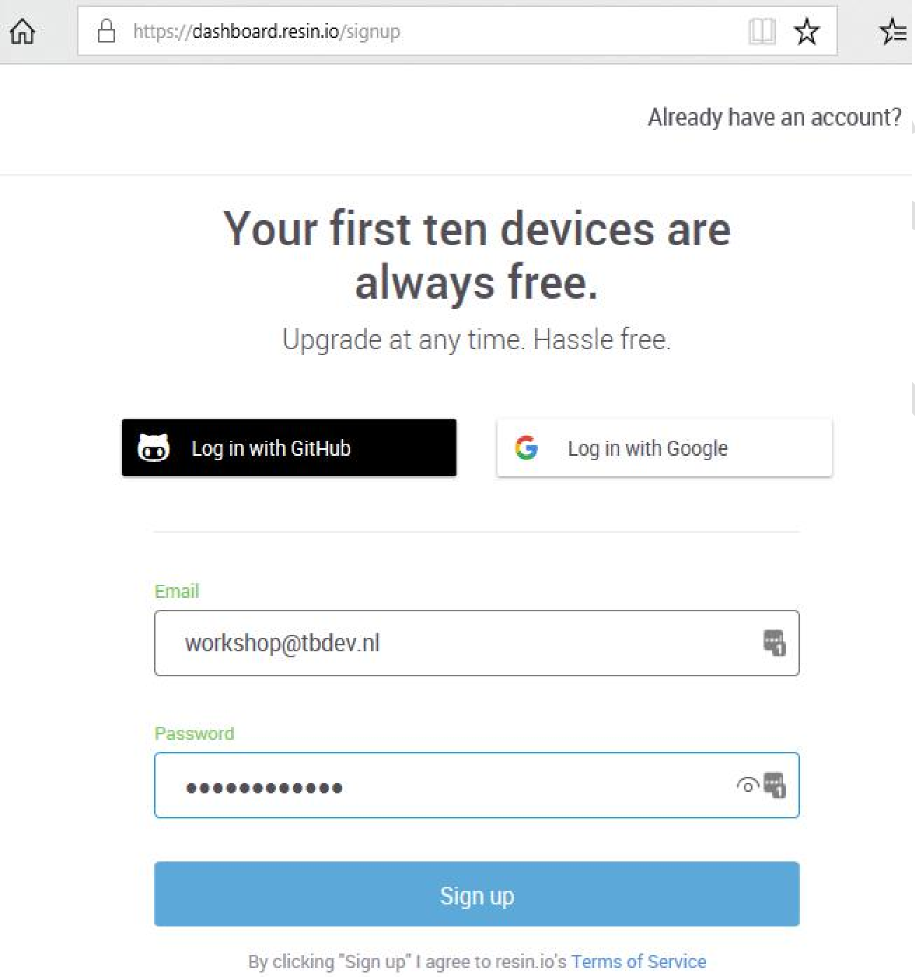
For Windows and OSX: Etcher, SD card writing software. windows, OSX

You will also need a SSH public/private key pair, generate a key pair if you do not have it.

3. Create resin.io account

If you already have a resin.io account, please go to the http://resin.io website and log in. And if you do not have it, now is the time to create it. Go to signup to create it. (If you have an account skip to the next step)

On the first screen enter your e-mail address and a (secure) password.



https://dashboard.resin.io/signup

Already have an account?

Your first ten devices are always free.

Upgrade at any time. Hassle free.

Log in with GitHub

Log in with Google

Email

workshop@tbdev.nl

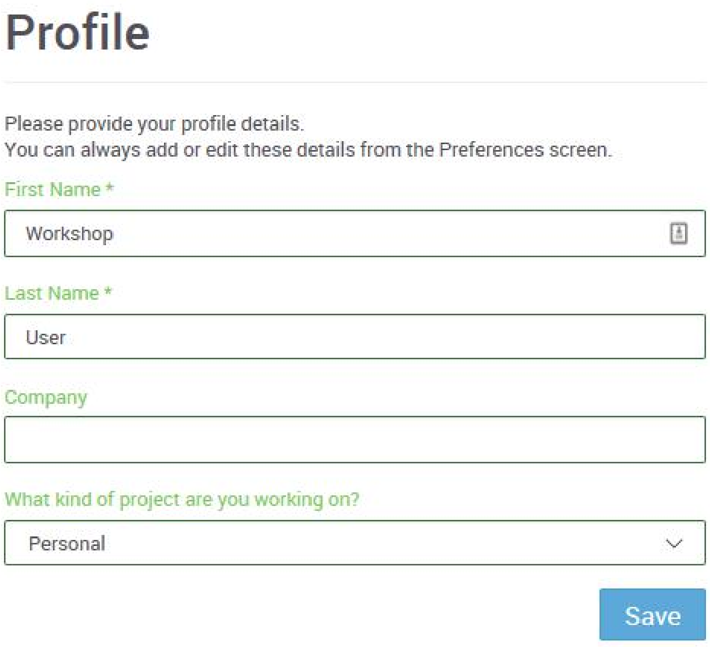
Password

●●●●●●●●●●●●

Sign up

By clicking "Sign up" I agree to resin.io's Terms of Service

Next you provide your first and last name, company if applicable and choose whether you will be using resin.io for a personal or professional project.



Profile

Please provide your profile details.
You can always add or edit these details from the Preferences screen.

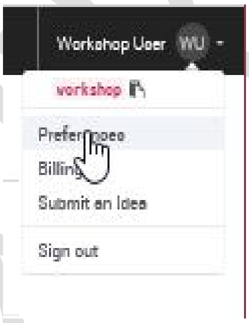
First Name *: Workshop

Last Name *: User

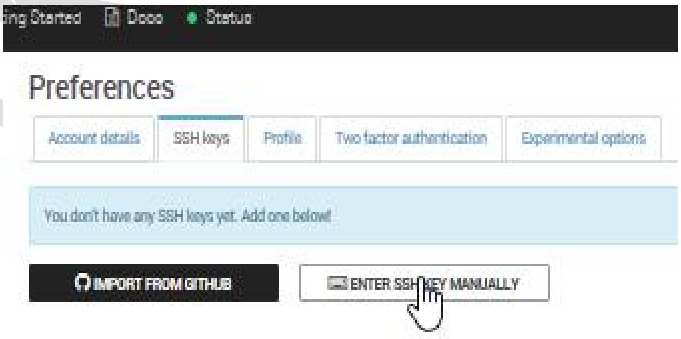
Company:

What kind of project are you working on?: Personal

Now we need to add our public SSH key. Resin.io uses this key when we upload code to run on our devices. In the upper right of the page click on the down arrow next to your name and select ‘Preferences’ from the menu.



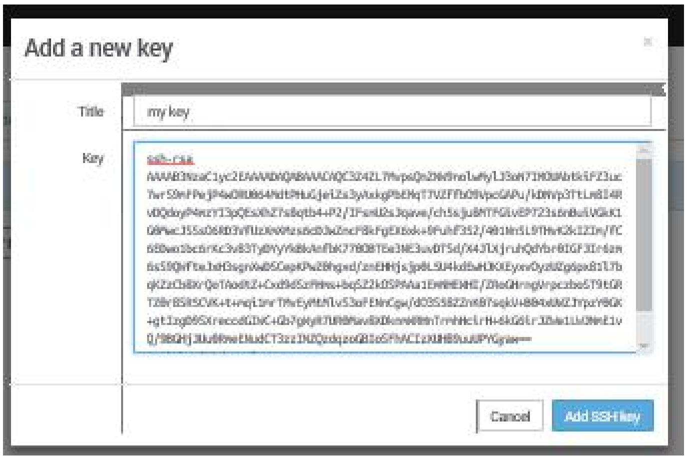
Now select the tab “SSH keys” and click on “Enter SSH key manually”



Preferences

You don't have any SSH keys yet. Add one below!

In the next window you need to paste the public part of your SSH key. This is store in id_rsa.pub in the folder .ssh in your home folder (Windows: c:\users<your username>.ssh\id_rsa.pub, OSX /Users/.ssh/id_rsa.pub, Linux: /home/.ssh/id_rsa.pub) Use your favorite editor to open the file and copy the contents to the form.



Title: my key

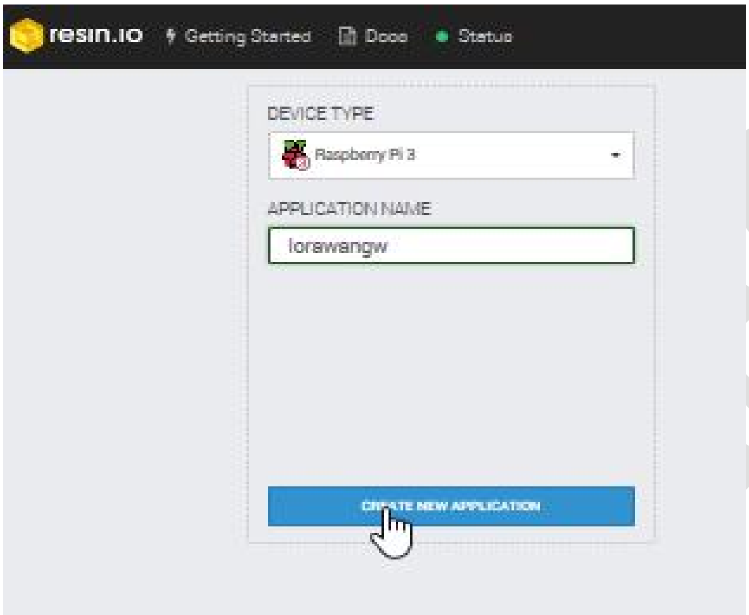
Key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQC3242L7MvpsQn2NW9nolwMyL33oN71MOUAbtkiFZ3uc
7wr59mFPejP4wDRU864MdtPHuGjeiZs3yAxkgPbEMqT7VZFfbO9VpcGAPu/kDWVp3TtLmBI4R
vDQdoyP4mzYI3pQEsXhZ7s8qtb4+P2/IFsmU2sJqave/ch5sjuBMTFGlvEP723s6nBuiVGkK1
G8PwcJ55sO68D3VfUzXmWzs6cD3wZncFBkFgEX6ok+0FuhF352/401Wn5L9THvK2kIZIn/fC
6B0wo1bc6rKc3v83TyDVyYkBkAnfbK77BO8TEe3NE3uvDT5d/X4JlXjruhQdVbr0IGF3Ir4zm
6s59QHfTeJxH3sgnXwD5CepKPw2Bhgxd/znEHHjsjp8L9U4kd5wHOKXEyovOyzUzg6pxB1L7b
qKZzCb8XrQeTAodtZ+Cxd9d5zPHms+bqSZ2kDSPAAa1EMMHEMHE/ZNeGHrngVrpczbe5T9tGR
T28rBSRSCVK+t+nqi1mrTMvEyMtMlvS3oFENnCgw/dO3S5822nKB7sqkU+B84xUWZJrpzYBGK
+gtIzgD9SXrecodGIbC+Gb7gWyR7URMMavBXDknmN8WnTrnhHcIrH+6kG6lr3ZbW1LWOMmE1v
Q/9BGHjJUuMmwEHudCT3zzINZQzdqzoQB1oSFhACIzXUHB9uUPYGgaw==
```

Save with “Add SSH key”. Go back to the main page by click in the resin.io logo in the top left of the page.

4. Create a new application

On the resin.io main screen you are able to create a new application. The device type conveniently defaults to Raspberry Pi 3, which we will be using. Enter ‘lorawangw’ for the application name and proceed with ‘CREATE NEW APPLICATION’.



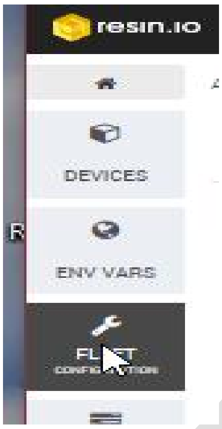
DEVICE TYPE
Raspberry Pi 3

APPLICATION NAME
lorawangw

5. Set Fleet Configuration

The Raspberry Pi model 3 requires some settings to correct the timing and switch serial ports (not used in this setup, but would be required to access a GPS if it had been mounted).

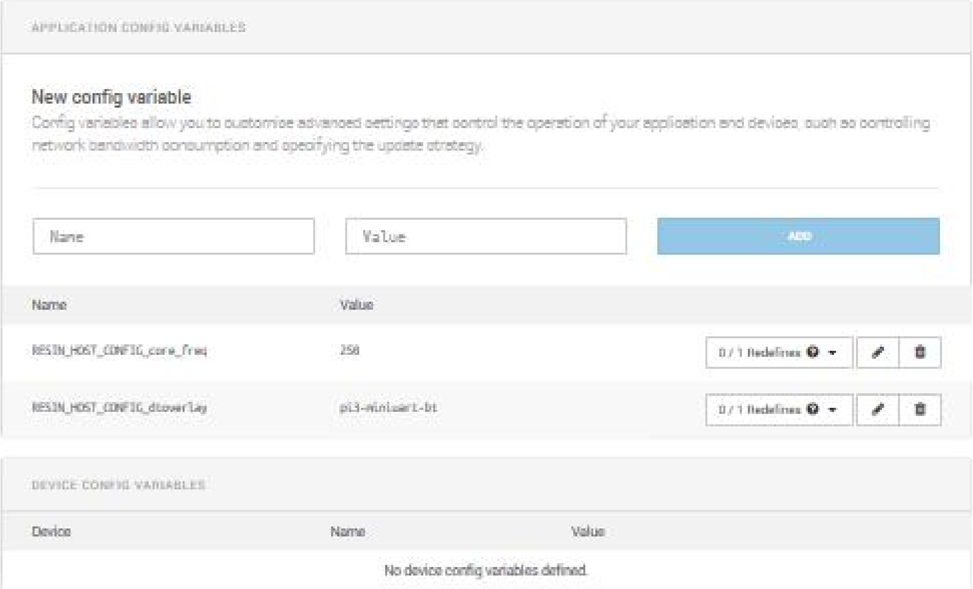
In your browser go to ‘FLEET’, as shown below:



Add the information as shown below:

Name	Value
RESIN_HOST_CONFIG_core_freq	250
RESIN_HOST_CONFIG_dtoverlay	pi3-miniuart-bt

Application config variables:



APPLICATION CONFIG VARIABLES

New config variable

Config variables allow you to customise advanced settings that control the operation of your application and devices, such as controlling network bandwidth consumption and specifying the update strategy.

Name | Value | ADD

Name	Value	
RESIN_HOST_CONFIG_core_freq	250	0 / 1 Redefines
RESIN_HOST_CONFIG_dtoverlay	pi3-miniuart-bt	0 / 1 Redefines

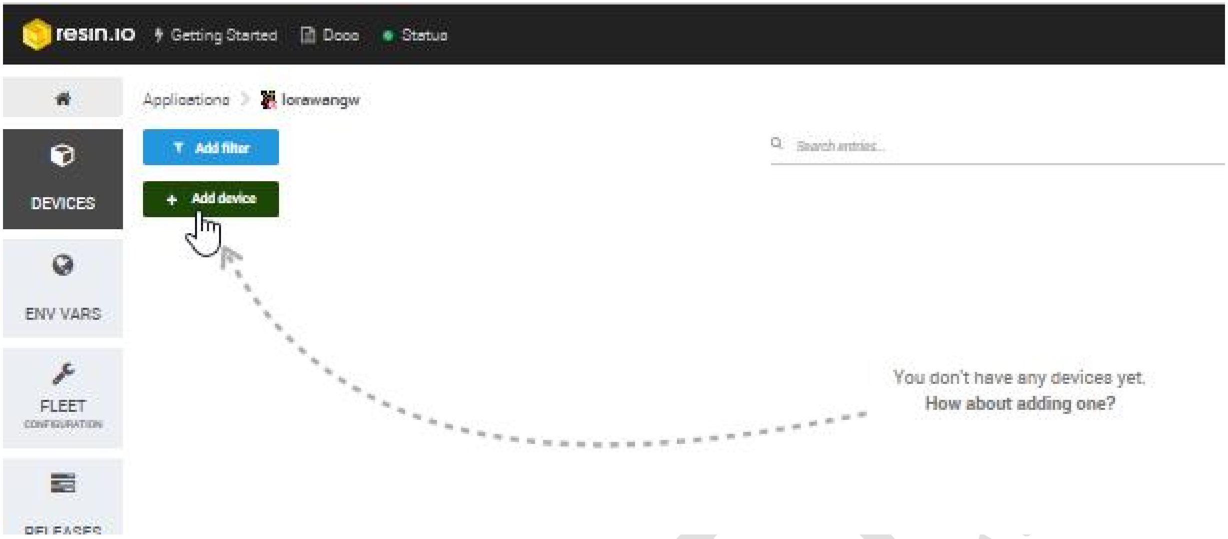
DEVICE CONFIG VARIABLES

Device	Name	Value

No device config variables defined.

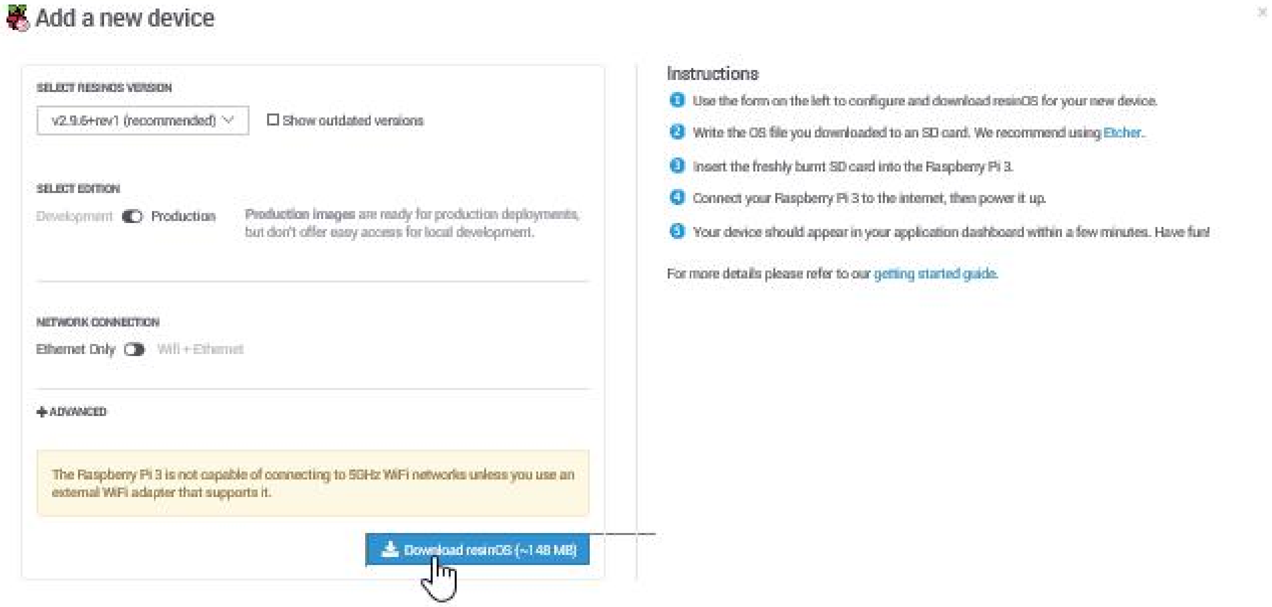
6. Add device

In the application select “Add device”.



You don't have any devices yet.
How about adding one?

Leave all settings set to the default and click “Download resinOS”.



Add a new device

SELECT RESINOS VERSION
v2.9.6+rev1 (recommended)
□ Show outdated versions

SELECT EDITION
Development Production
Production images are ready for production deployments, but don't offer easy access for local development.

NETWORK CONNECTION
Ethernet Only Wifi + Ethernet

The Raspberry Pi 3 is not capable of connecting to 5GHz WiFi networks unless you use an external WiFi adapter that supports it.

Instructions

1. Use the form on the left to configure and download resinOS for your new device.
2. Write the OS file you downloaded to an SD card. We recommend using Etcher.
3. Insert the freshly burnt SD card into the Raspberry Pi 3.
4. Connect your Raspberry Pi 3 to the internet, then power it up.
5. Your device should appear in your application dashboard within a few minutes. Have fun!

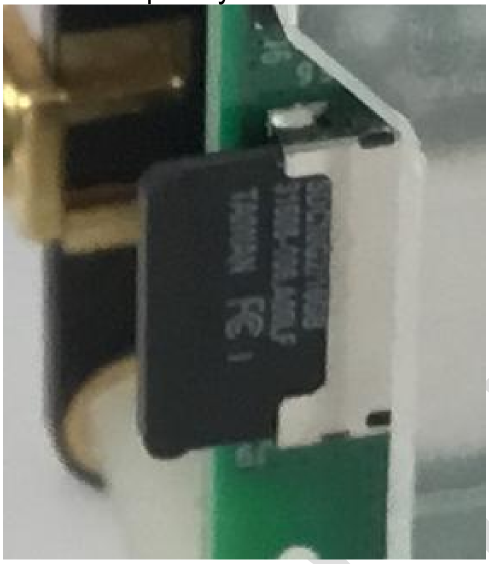
For more details please refer to our getting started guide.

Save the download to disk. (Remember where you save it!)

Once the download is finished, extract the contents of the zip file to disk (keep in mind the extracted file will be almost 2GB in size). In the next step we will write this file to SD card.

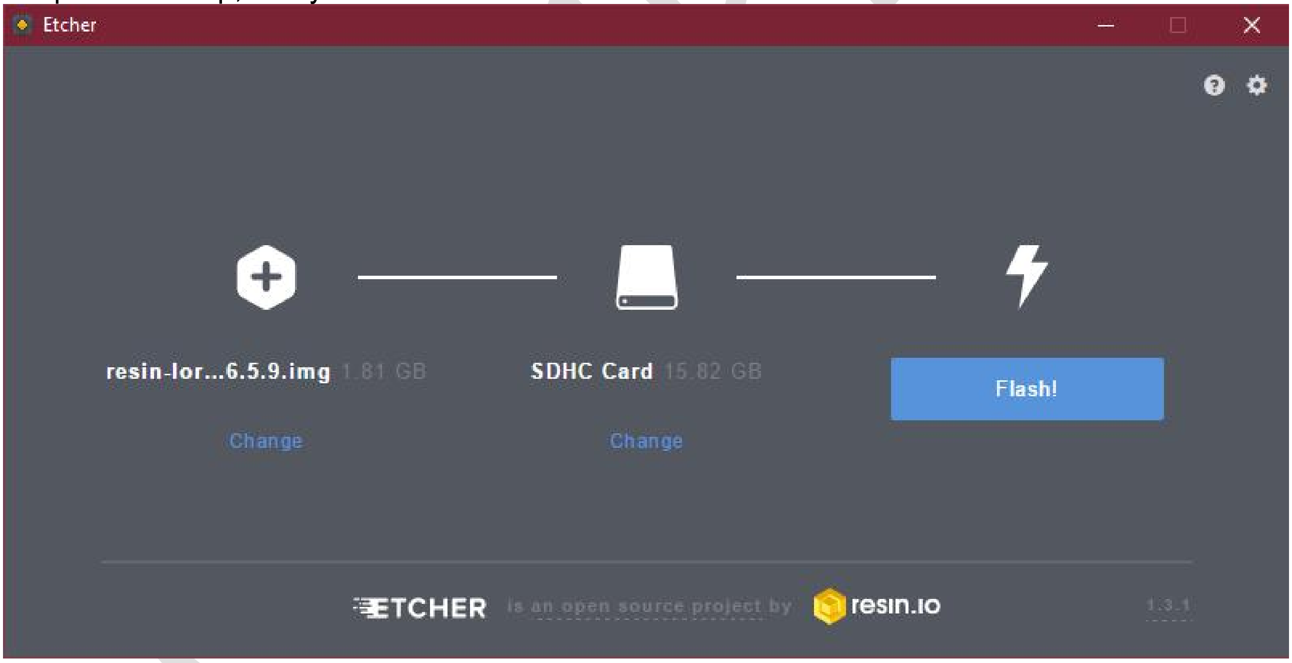
7. Write image to SD card

Get the micro SD card from the Raspberry Pi.



Insert it into the SD card writer (use a full size SD card adapter if required).

Windows & OSX: Open Etcher, click “Select image” and browse to the ISO file extracted in the previous step, verify the correct SD card is chosen.



Etcher
resin-lor...6.5.9.img 1.81 GB
Change
SDHC Card 15.82 GB
Change
Flash!
ETCHER is an open source project by resin.io
1.3.1

Once the right file and device have been click ‘Flash!’. (On Windows this will show the UAC dialog, click ‘Yes’)

Proceed to the next step while Etcher is writing the image.

Linux: Use dmesg to check which device is used for you SD card. Check with mount if any partitions of it are mounted and unmounts if this is the case. Write the image to SD card with dd.

```
sudo dd if=/tmp/resin-lorawangw-<revision>.img of=/dev/sdX bs=4M conv=sync
```

NOTE: Take care to select the right device, dd will quite happily overwrite any disk, including your Linux installation !

Proceed to the next step while dd is writing to the SD card.

8. Add software to your application

Earlier we have created the resin.io application. Now we need to add the application code that should be run on the Raspberry Pi. For this we will use a prepared LoRaWAN gateway application available on Github.

A. Open (OSX/Linux) Terminal or (Windows) Git Bash

B. Create a new directory to store the code on you local system go to it

```
mkdir lorawan-software

cd lorawan-software
```

C. Get the code from github

```
git clone https://github.com/jpmeijers/ttn-resin-gateway-rpi.git
```

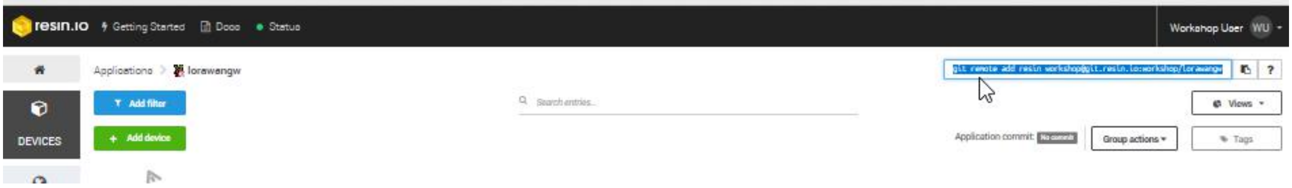
D. The output should resemble:

```
Cloning into 'ttn-resin-gateway-rpi'...

remote: Counting objects: 27, done.

remote: Compressing objects: 100% (20/20), done.

remote: Total 27 (delta 4), reused 27 (delta 4), pack-reused 0

Unpacking objects: 100% (27/27), done.
```

E. Go to the newly created directory

```
cd ttn-resin-gateway-rpi
```

F. Now we need to couple this reposity to resin.io so we are able to send the code there. For this we need to execute the git command listed on our application page. (Top right)



```
git remote add resin workshop@git.resin.io:workshop/lorawangw
```

G. Copy the text marked in the picture (for your application page) and paste it on the command line of your terminal/git bash.

```
git remote add resin <your resin.io account>@git.resin.io:<your resin.io account>/lorawangw.git
```

If the command runs successfully no output will be shown

H. Now we will push the code to resin.io, run

```
git push resin
```

The first time you run this command you will be see a message like

```
The authenticity of host 'git.resin.io (54.165.162.194)' can't be established.

ECDSA key fingerprint is SHA256:NfwmqnKId5cx1RWpebbEuuM87bCJbdyhzRnqFES9Nnw.

Are you sure you want to continue connecting (yes/no)?
```

Reply ‘yes’ to the question.

Next you will be prompted for the passphrase for your SSH key:

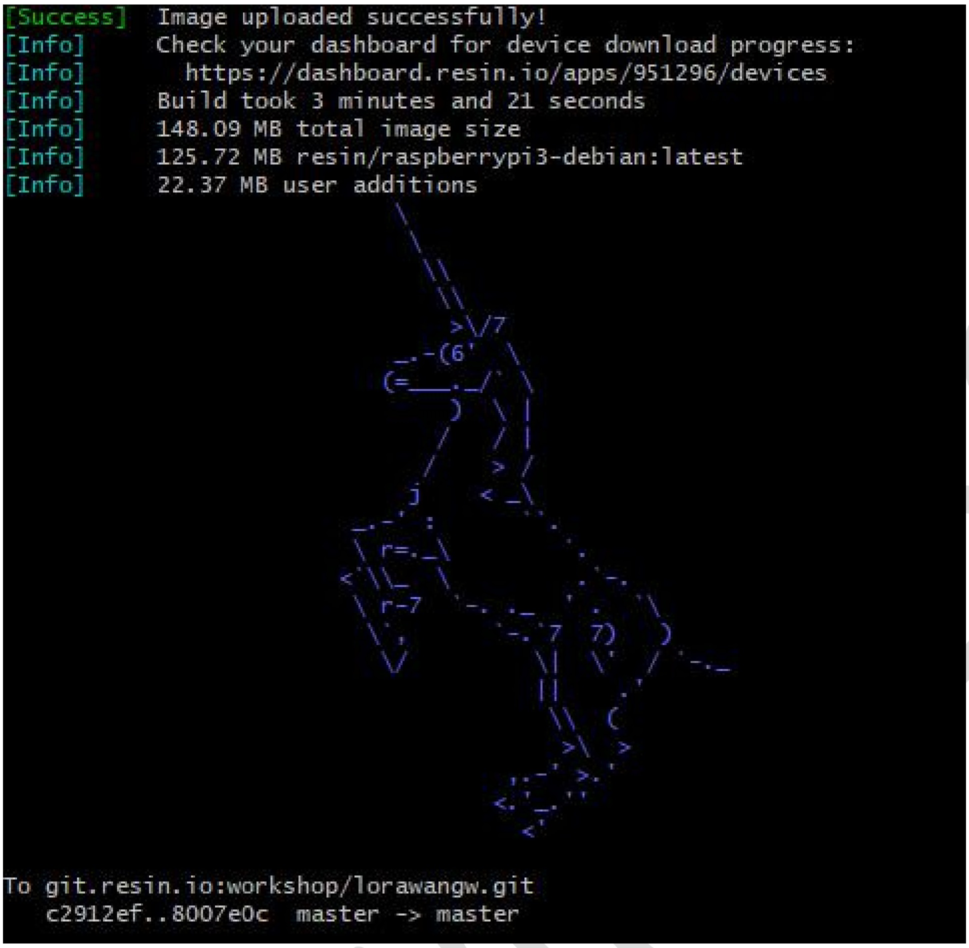
```
Enter passphrase for key '/c/Users/kersing/.ssh/id_rsa':
```

Enter the passphrase.

The upload starts and resin starts to build the image:

```
Counting objects: 27, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (24/24), done.

Writing objects: 100% (27/27), 2.30 MiB | 3.71 MiB/s, done.

Total 27 (delta 4), reused 0 (delta 0)

[Info]      Starting build for workshop/lorawangw, user workshop

[Info]      Dashboard link: https://dashboard.resin.io/apps/951296/devices

[Info]      Building on arm02

[Info]      Fetching base images

[==================================================>] 100%

[Info]      Building Dockerfile.template project
```

This will take a few minutes. When the build is finished a unicorn will be shown. Proceed with the next step while the build is running.



```
[Success]  Image uploaded successfully!
[Info]     Check your dashboard for device download progress:
[Info]       https://dashboard.resin.io/apps/951296/devices
[Info]     Build took 3 minutes and 21 seconds
[Info]     148.09 MB total image size
[Info]     125.72 MB resin/raspberrypi3-debian:latest
[Info]     22.37 MB user additions
                            \
                             \
                              \\
                               \\
                                >\/7
                            _.-(6'  \
                           (=___._/` \
                                )  \ |
                               /   / |
                              /    > /
                             j    < _\
                         _.-' :      ``.
                         \ r=._\        `.
                        <`\\_  \         .`-.
                         \ r-7  `-. ._  ' .  `\
                          \`,      `-.`7  7)   )
                           \/         \|  \'  / `-._
                                      ||    .'
                                       \\  (
                                        >\  >
                                    ,.-' >.'
                                   <.'_.''
                                     <'

To git.resin.io:workshop/lorawangw.git
   c2912ef..8007e0c  master -> master
```

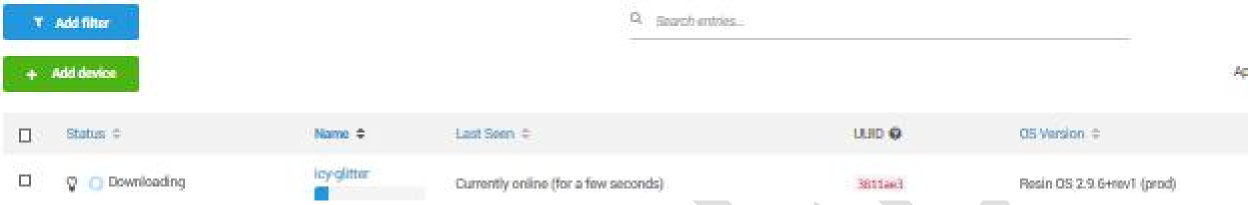
9. Boot the gateway hardware

Eject the micro SD card from your write and insert it into the Raspberry Pi SD card slot. (Metal contacts at PCB side)

Make sure the antenna is connected to the RAK831 card (see picture at the top), connect Ethernet cable and power adaptor to the Raspberry Pi. Now plug the power adaptor into power socket.

The red LED on the Raspberry Pi should light and (after a few seconds) a green LED should start flashing.

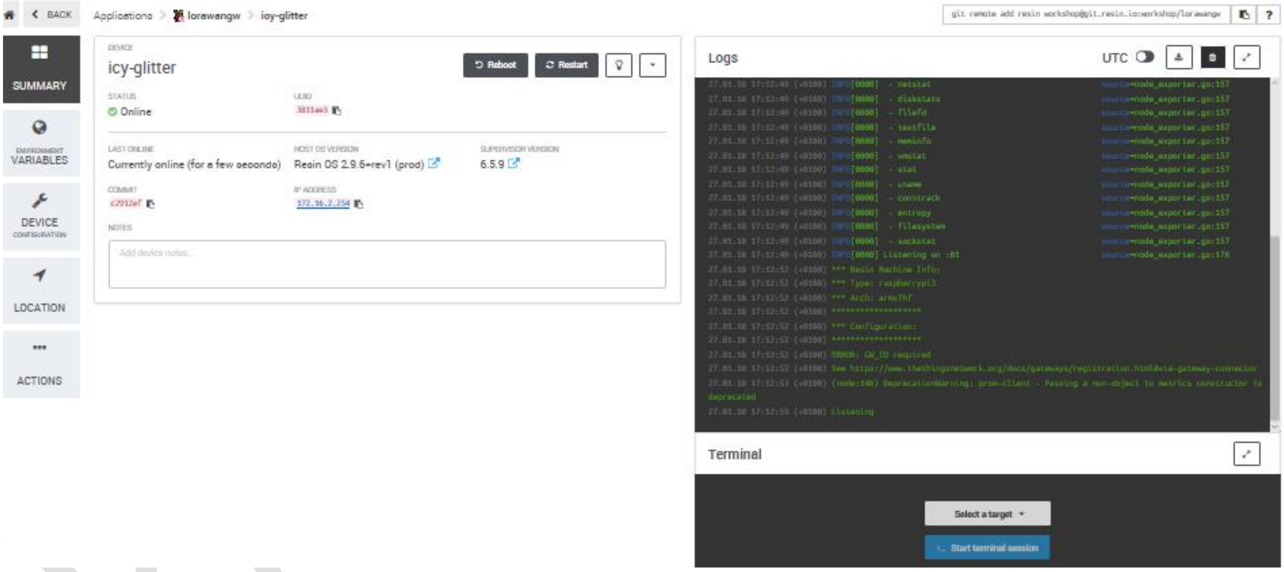
Switch to your browser, select “DEVICES” on the left. After 1-2 minutes a device should appear in on the application page.



☐	Status	Name	Last Seen	UUID	OS Version
☐	Downloading	icy-glitter	Currently online (for a few seconds)	3811ae3	Resin OS 2.9.6+rev1 (prod)

If the build we started in the previous step is finished the node will start downloading, if not it will stay idle.

Click on the device name to open the device details.



```
git remote add resin workshop@git.resin.io:workshop/lorawangw
```

DEVICE
icy-glitter

STATUS: Online
UUID: 3811ae3
LAST ONLINE: Currently online (for a few seconds)
HOST OS VERSION: Resin OS 2.9.6+rev1 (prod)
SUPERVISOR VERSION: 6.5.9
COMMIT: c2912ef
IP ADDRESS: 172.16.2.254
NOTES: Add device notes...

```
27.01.18 17:12:49 (+0100) INFO[0000]  - netstat                     source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - diskstats                   source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - filefd                      source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - textfile                    source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - meminfo                     source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - vmstat                      source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - stat                        source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - uname                       source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - conntrack                   source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - entropy                     source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - filesystem                  source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - sockstat                    source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000] Listening on :81              source=node_exporter.go:176
27.01.18 17:12:52 (+0100) *** Resin Machine Info:
27.01.18 17:12:52 (+0100) *** Type: raspberrypi3
27.01.18 17:12:52 (+0100) *** Arch: armv7hf
27.01.18 17:12:52 (+0100) ********************
27.01.18 17:12:52 (+0100) *** Configuration:
27.01.18 17:12:52 (+0100) ********************
27.01.18 17:12:52 (+0100) ERROR: GW_ID required
27.01.18 17:12:53 (+0100) See https://www.thethingsnetwork.org/docs/gateways/registration.html#via-gateway-connector
27.01.18 17:12:53 (+0100) (node:140) DeprecationWarning: prom-client - Passing a non-object to metrics constructor is
deprecated
27.01.18 17:12:53 (+0100) Listening
```

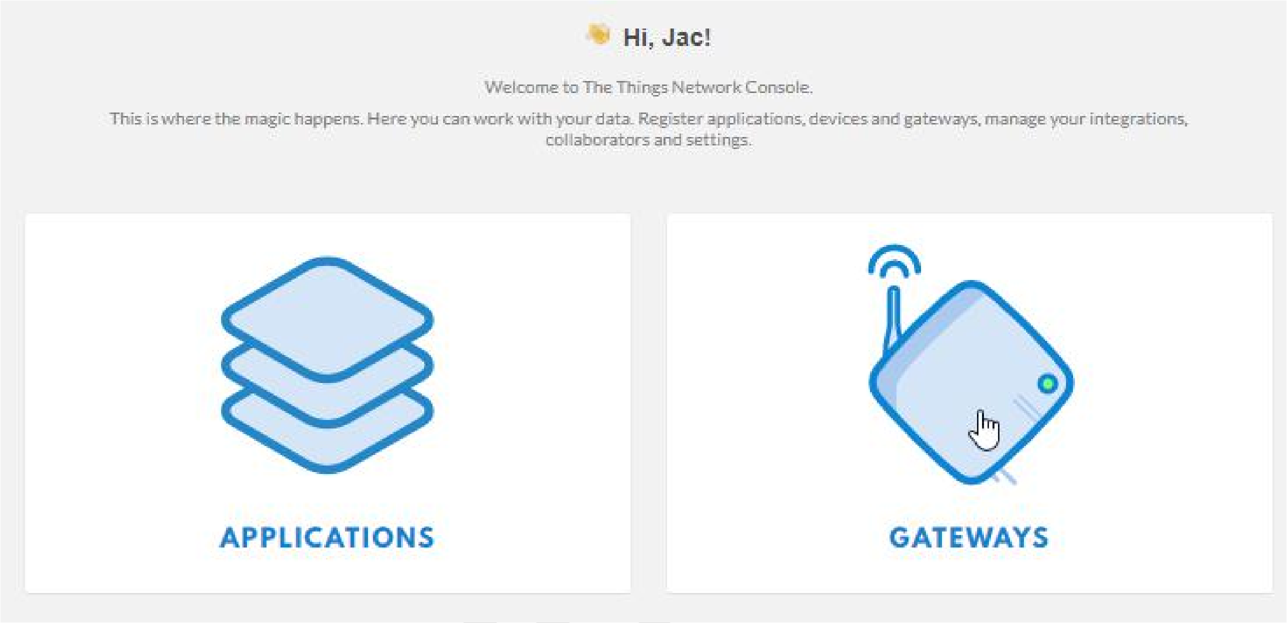
The top window on the right shows the output from the device. In this case it shows an error because the software attempted to start but is missing configuration parameters.

10. Add the gateway to the TTN console

To configure our gateway, we need to add it in the TTN console.

A. Use your TTN credentials to log in to https://console.thethingsnetwork.org/

B. In the main screen click on gateways

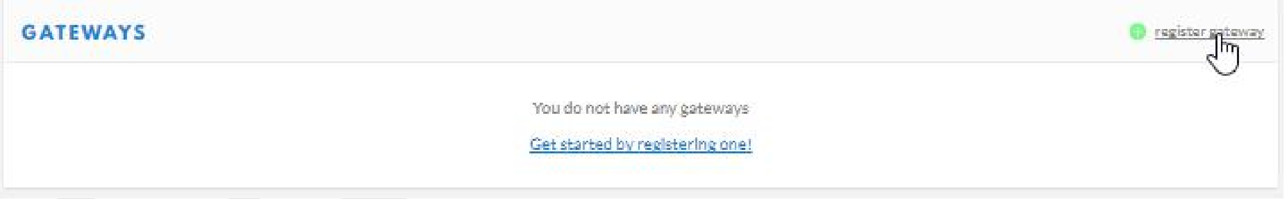


Hi, Jac!

Welcome to The Things Network Console.

This is where the magic happens. Here you can work with your data. Register applications, devices and gateways, manage your integrations, collaborators and settings.

C. Click on "register gateway" (or "Get started by registering one")



You do not have any gateways

Get started by registering one!

D. In the form enter the following values:

For 'Gateway ID', choose a unique name for the gateway, allowed A-Z, a-z, 0-9 and -.

Do not check 'legacy packet forwarder'.

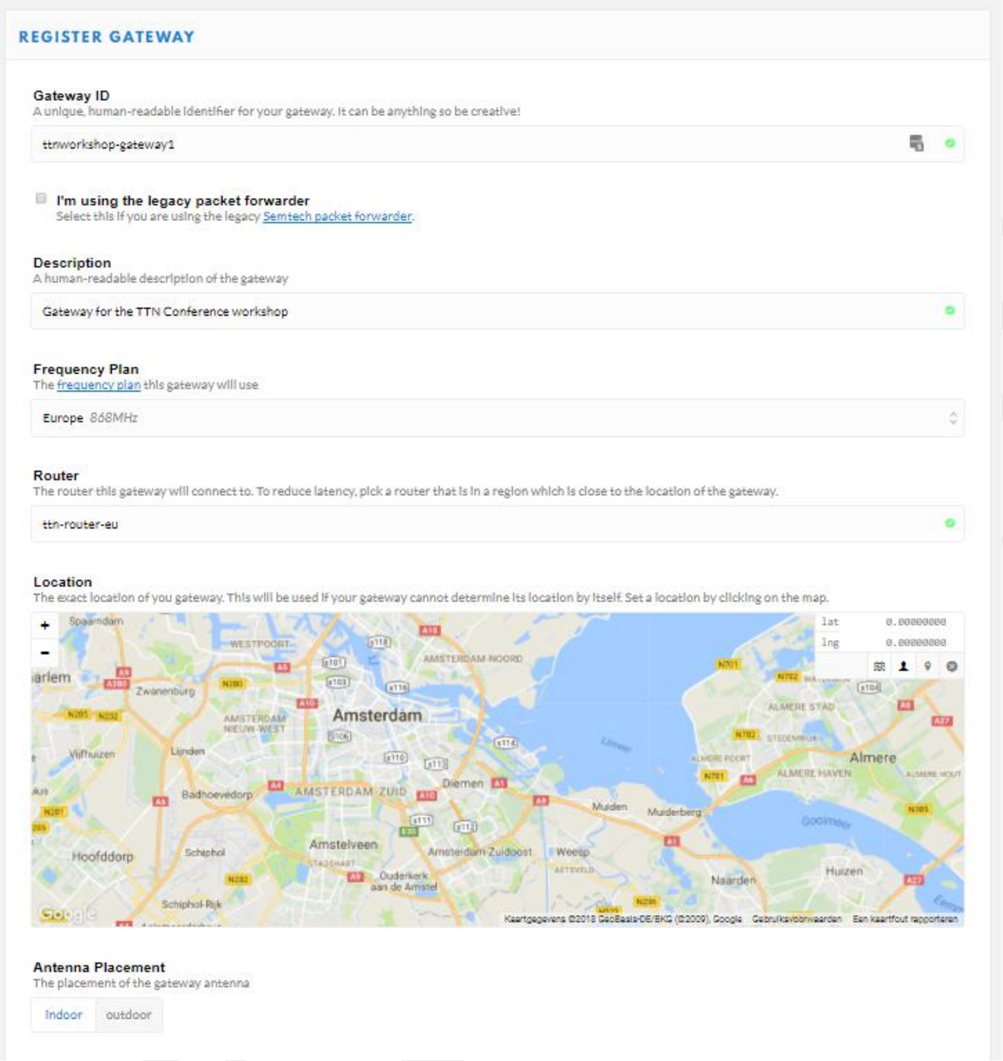
For 'Description', enter a human read-able description.

For 'Frequency Plan', choose the one appropriate for your location, for Amsterdam use Europe 868MHz.

For 'Router', choose the one closest to you. Routers prefixed with 'ttn-' are public ones operated by The Things Network. switch-router is based in Switzerland and meshed-router is Australian.

For 'Location', choose your location on the map.

For ‘Antenna Placement’, choose Indoor/Outdoor as appropriate.



REGISTER GATEWAY

Gateway ID
A unique, human-readable identifier for your gateway. It can be anything so be creative!
ttnworkshop-gateway1

□ 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
Gateway for the TTN Conference workshop

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

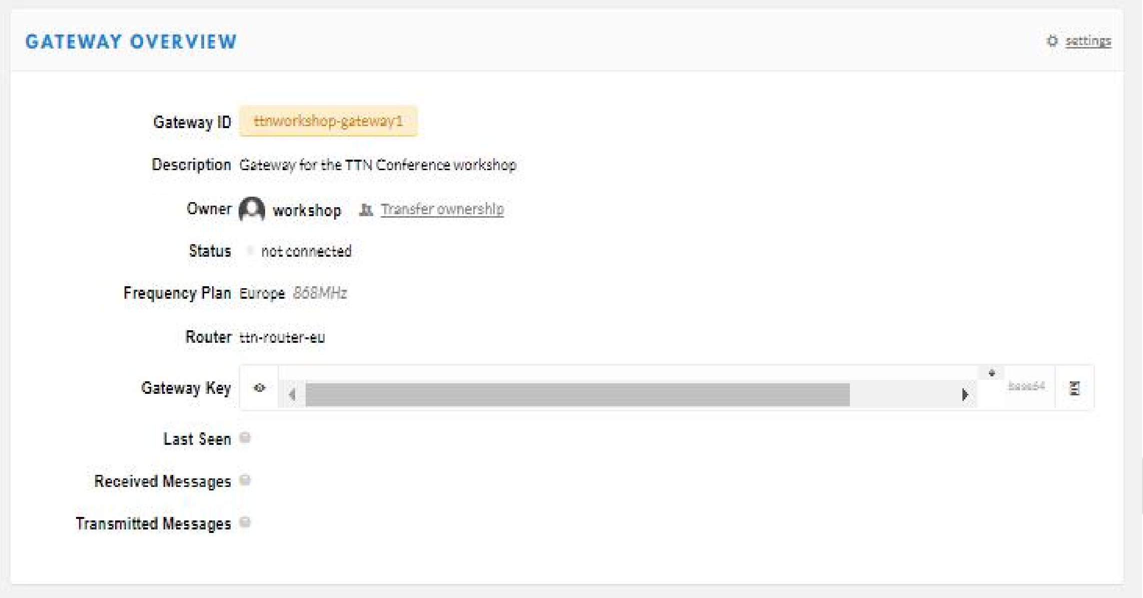
Location
The exact location of you gateway. This will be used if your gateway cannot determine its location by itself. Set a location by clicking on the map.

lat 0.00000000
lng 0.00000000

Antenna Placement
The placement of the gateway antenna
Indoor outdoor

Click “Register Gateway” to proceed.

The result should look like:



GATEWAY OVERVIEW

Gateway ID ttnworkshop-gateway1

Description Gateway for the TTN Conference workshop

Owner workshop

Status not connected

Frequency Plan Europe 868MHz

Router ttn-router-eu

Gateway Key

Last Seen

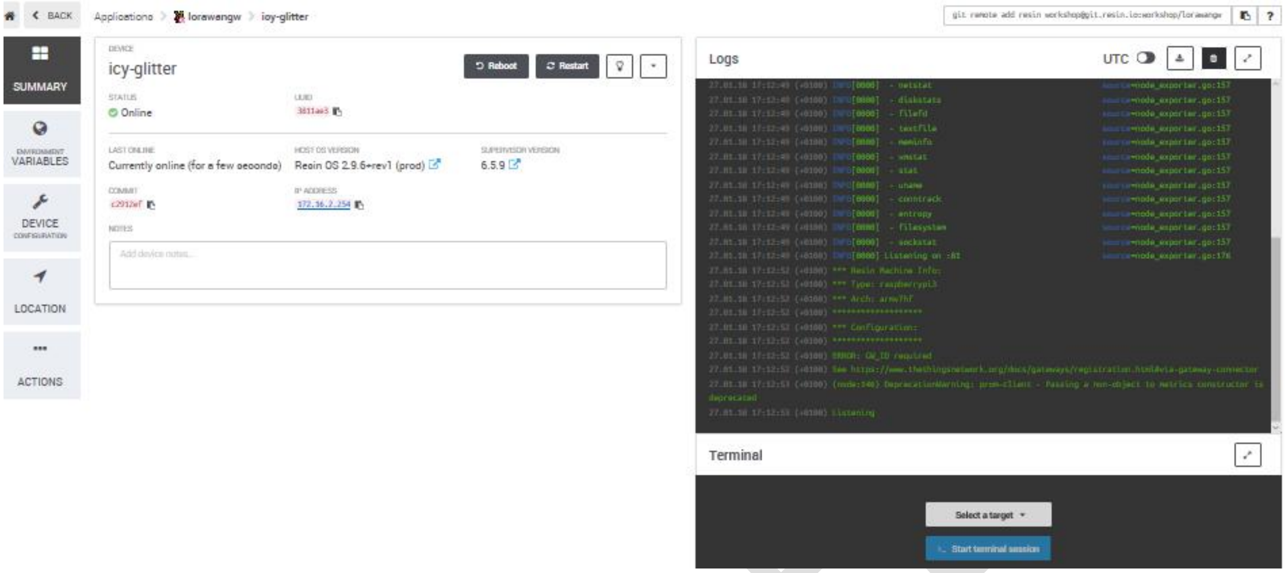
Received Messages

Transmitted Messages

Keep this window open, you need it to copy-and-paste information.

11. Configure your device

Switch to your resin.io page. If you do not have the device summary page open, open it now.



```
git remote add resin workshop@git.resin.io:workshop/lorawangw
```

DEVICE
icy-glitter

STATUS: Online

UUID: 3811ae3

LAST ONLINE: Currently online (for a few seconds)

HOST OS VERSION: Resin OS 2.9.6+rev1 (prod)

SUPERVISOR VERSION: 6.5.9

COMMIT: c2912ef

IP ADDRESS: 172.16.2.254

NOTES: Add device notes...

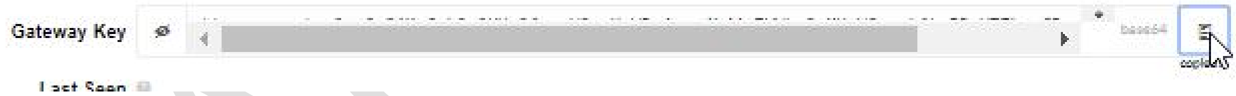
```
27.01.18 17:12:49 (+0100) INFO[0000]  - netstat                source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - diskstats              source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - filefd                 source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - textfile               source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - meminfo                source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - vmstat                 source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - stat                   source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - uname                  source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - conntrack              source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - entropy                source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - filesystem             source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000]  - sockstat               source=node_exporter.go:157
27.01.18 17:12:49 (+0100) INFO[0000] Listening on :81          source=node_exporter.go:176
27.01.18 17:12:52 (+0100) *** Resin Machine Info:
27.01.18 17:12:52 (+0100) *** Type: raspberrypi3
27.01.18 17:12:52 (+0100) *** Arch: armv7hf
27.01.18 17:12:52 (+0100) *******************
27.01.18 17:12:52 (+0100) *** Configuration:
27.01.18 17:12:52 (+0100) *******************
27.01.18 17:12:52 (+0100) ERROR: GW_ID required
27.01.18 17:12:52 (+0100) See https://www.thethingsnetwork.org/docs/gateways/registration.html#via-gateway-connector
27.01.18 17:12:53 (+0100) (node:140) DeprecationWarning: prom-client - Passing a non-object to metrics constructor is
deprecated
27.01.18 17:12:53 (+0100) Listening
```

Click on “ENVIRONMENT VARIABLES”.

You need to add three variables:

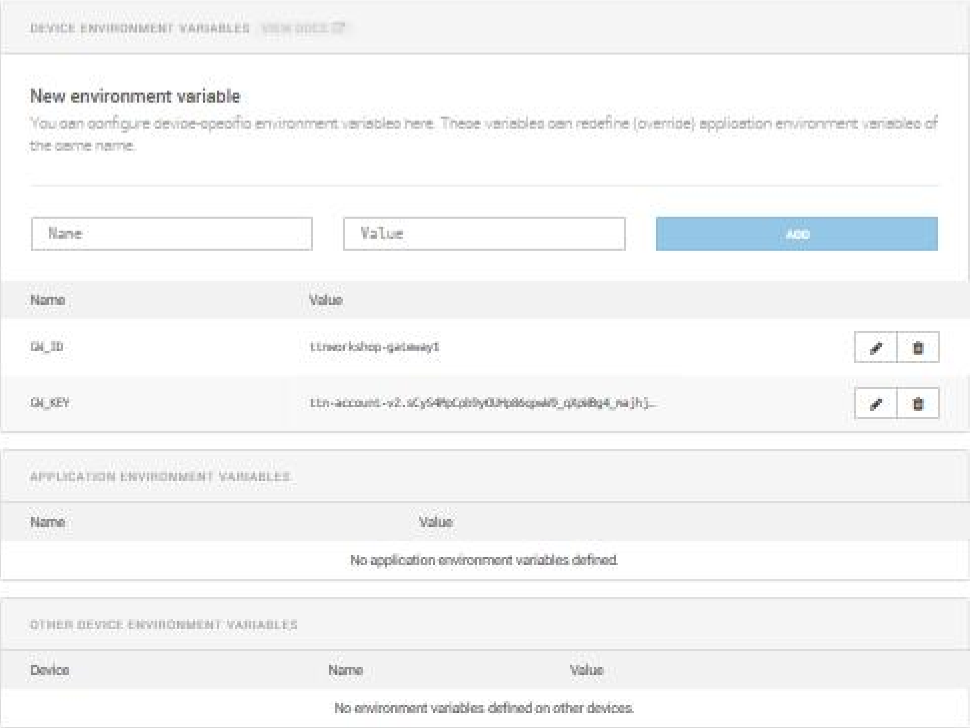
A. Name the first one “GW_ID” (no quotes) and copy the value listed for the “Gateway ID” in the TTN console.

B. Name the second one “GW_KEY” (no quotes), on the TTN console click on the little eye icon to the right of “Gateway Key”, this will display the value. Now use the icon at the end of the line to copy the value (and display “copied”)



C. Paste this value and add.

D. Enter “GW_RESET_PIN” (no quotes) and value 11.



DEVICE ENVIRONMENT VARIABLES VIEW DOCS

New environment variable

You can configure device-specific environment variables here. These variables can redefine (override) application environment variables of the same name.

Name | Value | ADD

Name	Value
GW_ID	ttnworkshop-gateway1
GW_KEY	ttn-account-v2.sCyS4RpCpb9yOUHp86qwW9_qKpBg4_najhj…

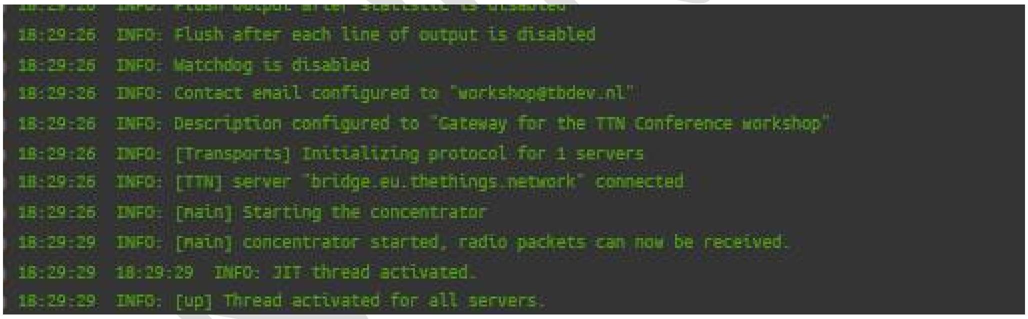
APPLICATION ENVIRONMENT VARIABLES

Name	Value
No application environment variables defined	

OTHER DEVICE ENVIRONMENT VARIABLES

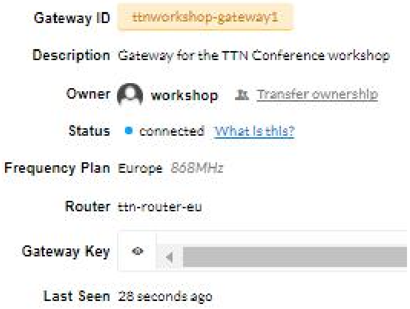
Device	Name	Value
No environment variables defined on other devices.		

Once set (and the software download has finished) the software will start. If the software starts correctly you will see “concentrator started” in the output in the resin.io Logs.



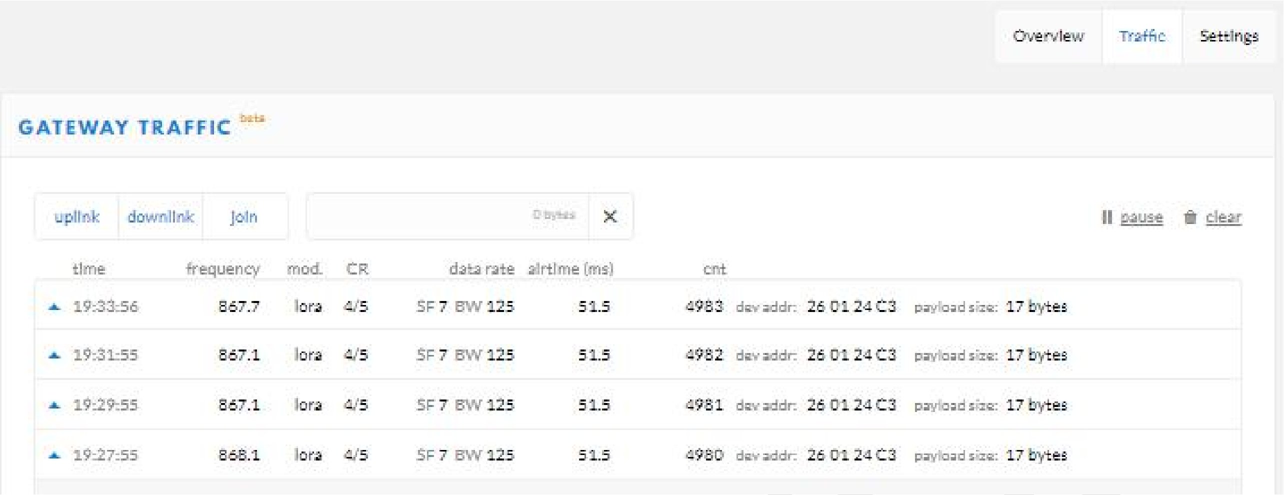
```
18:29:26  INFO: Flush output after statistic is disabled
18:29:26  INFO: Flush after each line of output is disabled
18:29:26  INFO: Watchdog is disabled
18:29:26  INFO: Contact email configured to "workshop@tbdev.nl"
18:29:26  INFO: Description configured to "Gateway for the TTN Conference workshop"
18:29:26  INFO: [Transports] Initializing protocol for 1 servers
18:29:26  INFO: [TTN] server "bridge.eu.thethings.network" connected
18:29:26  INFO: [main] Starting the concentrator
18:29:29  INFO: [main] concentrator started, radio packets can now be received.
18:29:29  18:29:29  INFO: JIT thread activated.
18:29:29  INFO: [up] Thread activated for all servers.
```

In the TTN console the “Last seen” status should change to something between 0 and 60 seconds.



Gateway ID ttnworkshop-gateway1
Description Gateway for the TTN Conference workshop
Owner workshop Transfer ownership
Status ● connected What is this?
Frequency Plan Europe 868MHz
Router ttn-router-eu
Gateway Key
Last Seen 28 seconds ago

In the TTN console switch to “Traffic”. If there are any nodes nearby sending data (should be the case during the conference), packets will show.



	time	frequency	mod.	CR	data rate	airtime (ms)	cnt		
▲	19:33:56	867.7	lora	4/5	SF 7 BW 125	51.5	4983	dev addr: 26 01 24 C3	payload size: 17 bytes
▲	19:31:55	867.1	lora	4/5	SF 7 BW 125	51.5	4982	dev addr: 26 01 24 C3	payload size: 17 bytes
▲	19:29:55	867.1	lora	4/5	SF 7 BW 125	51.5	4981	dev addr: 26 01 24 C3	payload size: 17 bytes
▲	19:27:55	868.1	lora	4/5	SF 7 BW 125	51.5	4980	dev addr: 26 01 24 C3	payload size: 17 bytes

Congratulations, the gateway is now operational !

12. Contact information

Shenzhen Business

E-Mail: ken.yu@rakwireless.com

Address: Room 506, Bldg. 3, Minqi Technology Park, No.65 Taoyuan Road,
Xili Block, Nanshan District, Shenzhen

Shenzhen Technical

E-Mail: steven.tang@rakwireless.com

Tel : 0755-86108311

Address: Room 506, Bldg. 3, Minqi Technology Park, No.65 Taoyuan Road,
Xili Block, Nanshan District, Shenzhen

13. Change Note

Version	Date	Modify content	Arthur
V1.0	2018-04-11	Create the document	Farce