RENS

Renewable energy solutions

SolarTracker

User manual

version 1.0

Contents

1 Introduction

1.1 Welcome

Welcome to the User Manual for the RENS SolarTracker. This comprehensive guide is designed to assist you in understanding and effectively operating the software that governs the movement of solar panels in your solar power plant.

2 Getting started

2.1 Initial setup

The RENS SolarTracker consists of the central control unit (CCU), LoRaWAN gateway (gateway), and a number of end devices. The central control unit is a Beckhoff CX5140 running Windows 10 LTSC, the gateway is a Kerlink Wirnet™ iStation 868, and the end devices are custom STM32WL55JC based microcontrollers.
The CCU communicates with the gateway using the UDP protocol, and the gateway communicates with the end devices using LoRaWAN v1.0.2.

Deploying the system requires:

- Powering on the CCU and the gateway and obtaining their IP addresses
  (it is recommended to set up these devices with static local IP addresses)
- Starting the ChirpStack LoRaWAN network server
- Starting the SolarTracker dashboard
- Creating a new ChirpStack application and API key and adding them to the SolarTracker
- Setting up the gateway and adding it to the ChirpStack server
- Creating a device profile for the end devices

2.1.1 LoraWAN gateway setup

You can connect to the gateway via a browser or an SSH client, using its previously obtained IP address

The default login info for the gateway is:

Web interface:
username: admin
password: pwd4admin

SSH:
port: 22
username: root
password: pdmk-0830D3
(generally pmdk-<last 6 characters of the board ID>)

SSH into the gateway using your preferred SSH client and setup the following:

- Open file /etc/lorafwd.toml in a text editor (e.g. vi /etc/lorafwd.toml)
- in the [gwmp] section of the file change the *node* value to the IP address of the ChirpStack server (CCU)
- in the same section change the *service.downlink* and *service.uplink* values to the downlink and uplink ports defined by the ChirpStack server (default: 1700 for both)
- delete any '#' symbol preceding these lines to uncomment them
- save and exit
- run the commands '/etc/init.d/lorafwd restart' and '/etc/init.d/lorad restart'

This will enable LoRa packet forwarding to and from the end devices, and connect the gateway to the ChirpStack LoRaWAN network server.

2.1.2 Central control unit setup

The CCU consists of a ChirpStack LoRaWAN network server running in a Docker container, and a web server, hosting the SolarTracker dashboard.

First, start the docker container from the provided image file. This will also start the ChirpStack dashboard, accessible via a web browser at the address localhost:8080/

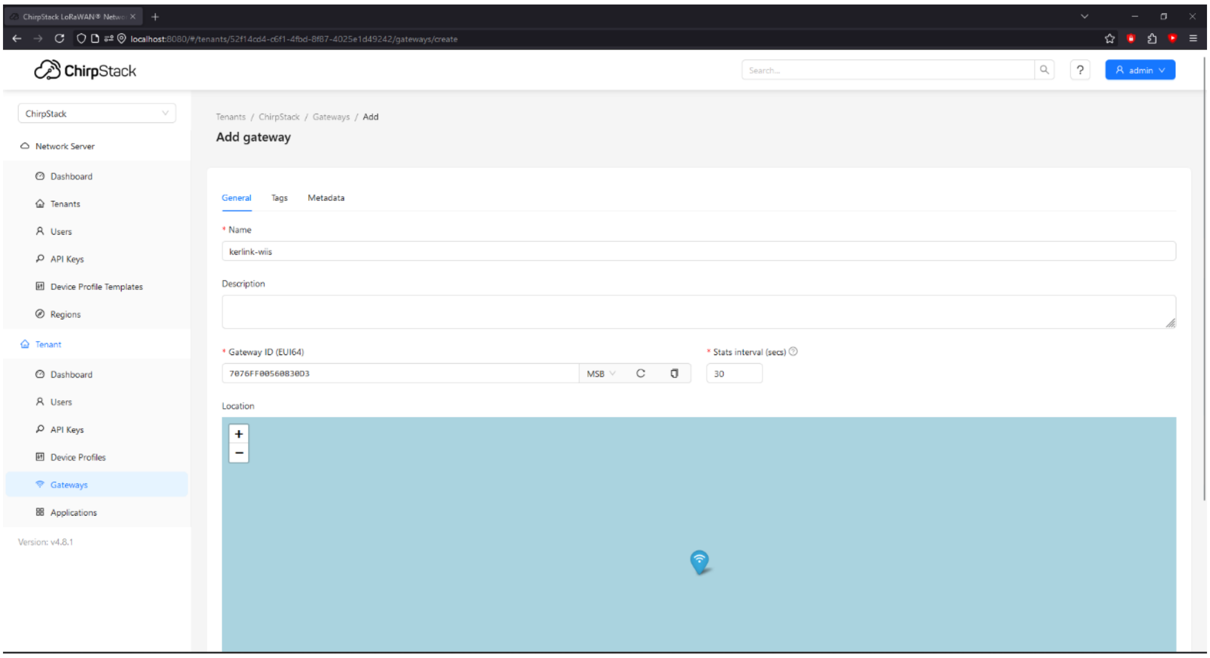
Log into the ChirpStack dashboard using credentials:

username: admin
password: admin

To set up ChirpStack, you will need to do the following:

- Connect the gateway
- Add a device profile for the end devices
- Add an application
- Add an API key

To connect the gateway, click on the *Gateways* button on the sidebar, and then click on the *Add gateway* button in the top right of the newly opened window.



* Name: kerlink-wiis

Description:

* Gateway ID (EUI64): 7076FF00560830D3

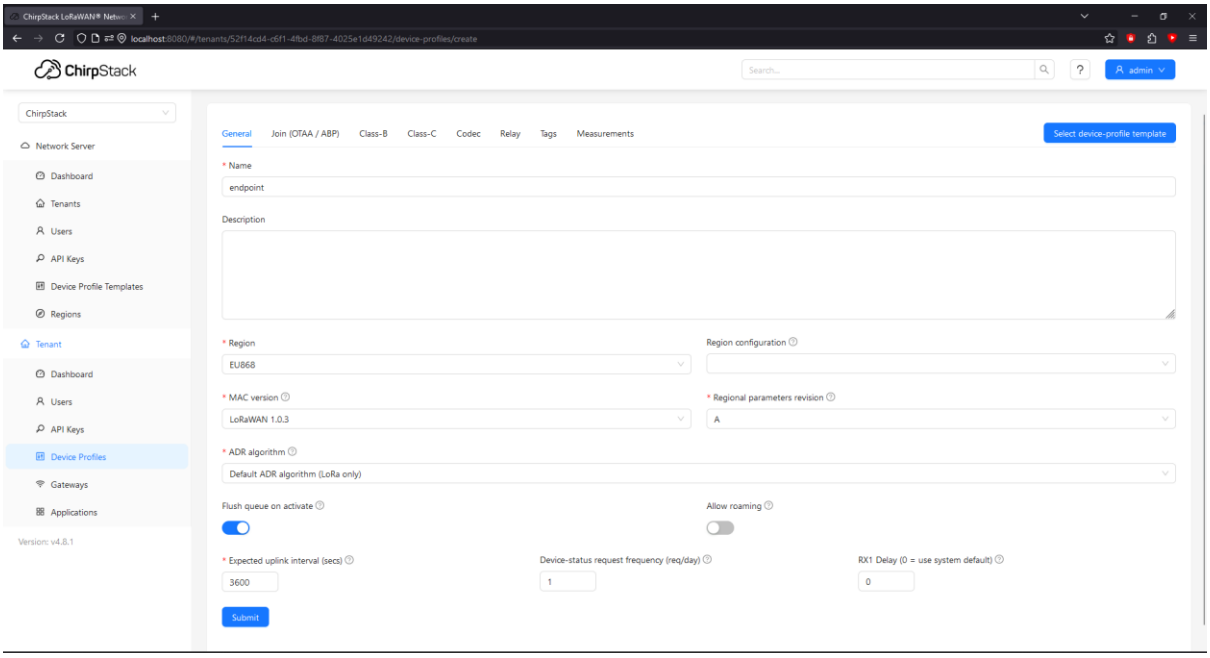
* Stats interval (secs): 30

Figure 1 - Adding a gateway

In this window, in the *General* tab, pick a name of your choosing for the gateway and write in the gateway's EUI64 (written on the side of the gateway, or accessible via ssh command: 'grep EUI /tmp/board_info.json'

All other data should be left default. Click the *Submit* button at the bottom of the page.
After a couple seconds, the gateway will appear as online on the *Dashboard* tab.

Next, add a device profile in the *Device profiles* tab on the sidebar



* Name: endpoint

Description:

* Region: EU868

Region configuration:

* MAC version: LoRaWAN 1.0.3

* Regional parameters revision: A

* ADR algorithm: Default ADR algorithm (LoRa only)

Flush queue on activate: on

Allow roaming: off

* Expected uplink interval (secs): 3600

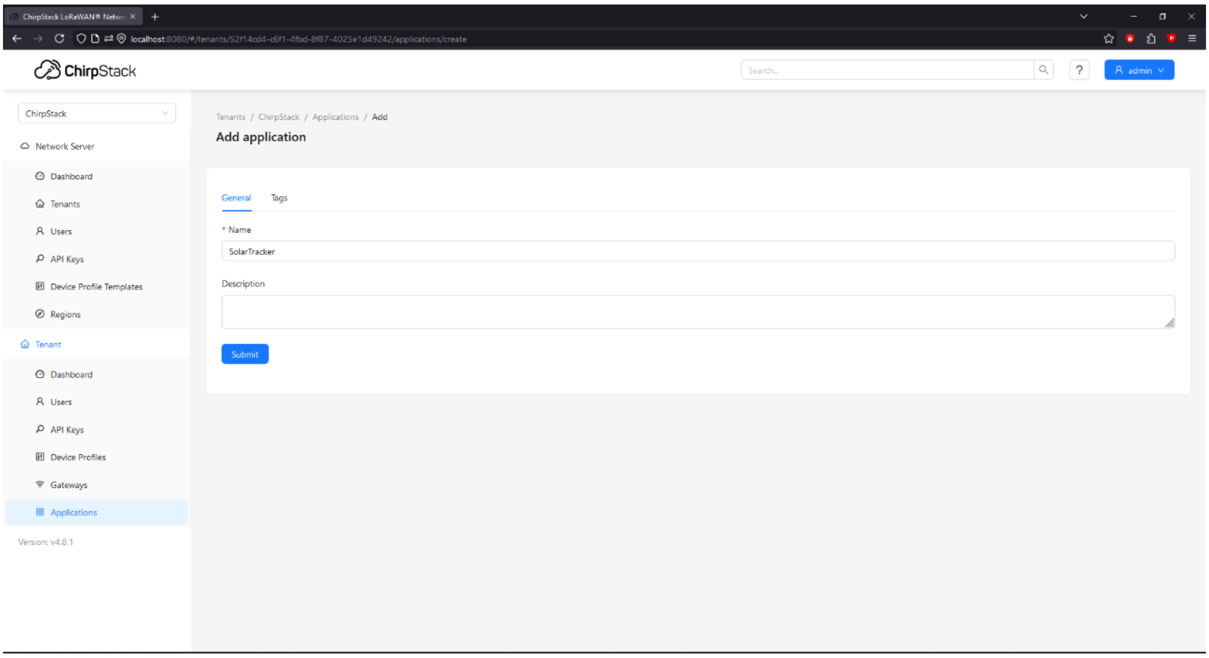
Device-status request frequency (req/day): 1

RX1 Delay (0 = use system default): 0

Figure 2 - Adding a device profile

Pick a name of your choosing for the endpoint device profile and set all other values the same as in the image above (all default).
Click the *Class-C* tab at the top of the screen and toggle this option *on*.
Leave all other options as default, and click the *Submit* button

After this, add an application in the *Applications* tab on the sidebar.



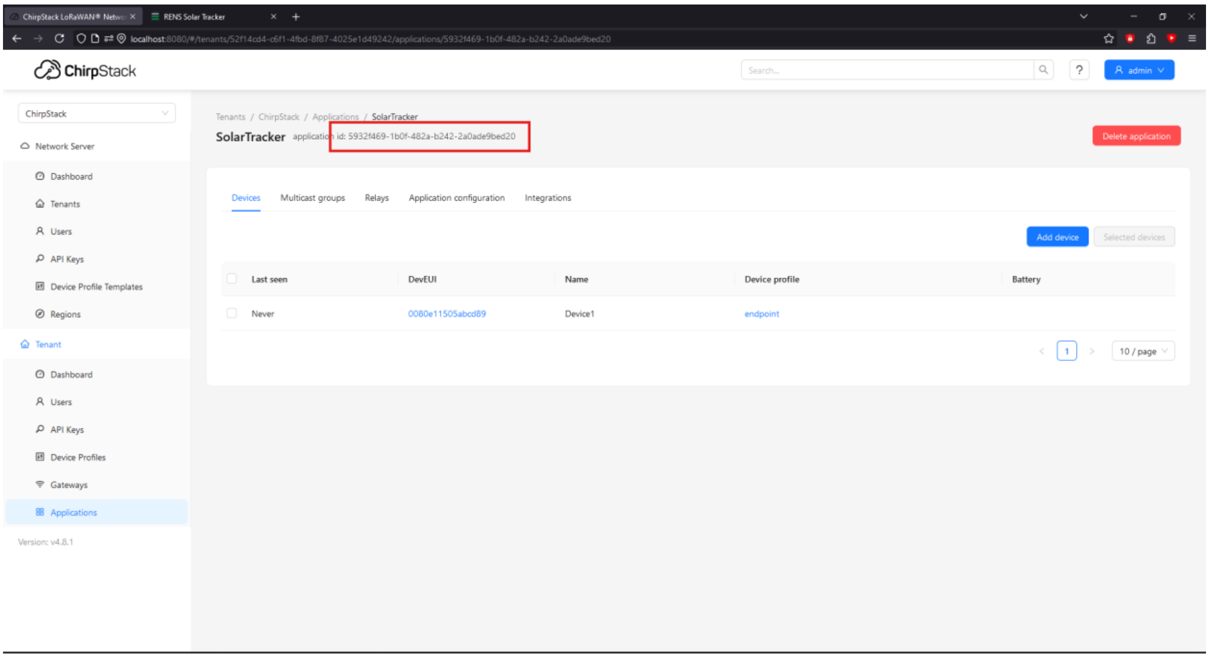
Add application

* Name: SolarTracker

Description:

Figure 3 - Adding an application

Simply choose a name for the application and click the *Submit* button.



SolarTracker application id: 5932f469-1b0f-482a-b242-2a0ade9bed20

Last seen	DevEUI	Name	Device profile	Battery
Never	0080e11505abcd89	Device1	endpoint	

Figure 4 - Application ID

Note the *application id* at the top of the application's page. This will be important for SolarTracker setup later on.
This value can be accessed at any time.

Finally, add an API key by pressing the *API keys* button on the sidebar and then *Add API key* on the top left.
Pick a name of the API key, and click *Submit*
MAKE SURE TO COPY THE API KEY THAT COMES UP because it can only be viewed once.
In case you lose the API key, simply create a new one. Old API keys can be invalidated by deleting them.

We are now ready to start adding end devices to the ChirpStack network server, but before that, let's start and configure the SolarTracker.

Start the SolarTracker dashboard by opening a PowerShell terminal in the folder where the SolarTracker.py script is located, and running the following commands (requires python3 to be installed):

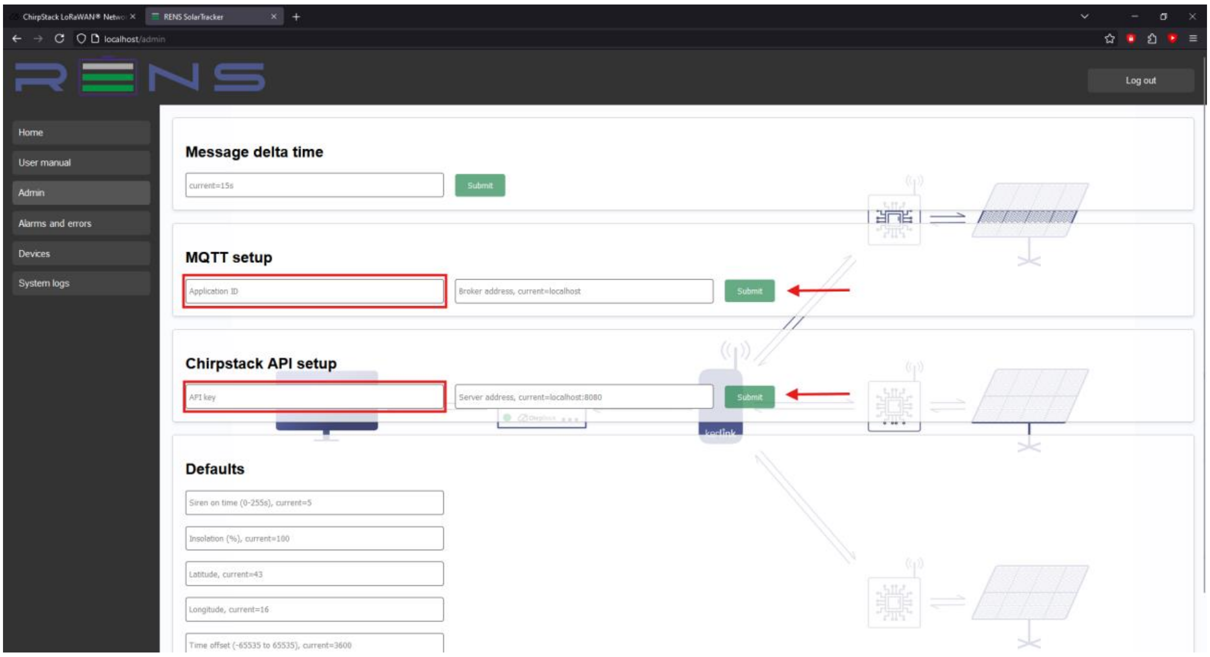
- pip install -r requirements.txt
  (installs all python packages required for running the application)
- python SolarTracker.py
  (runs the application)

Once the application is running, open a web browser of choice, type in the IP address of the CCU (or localhost:80/ if setting up directly from the CCU), and be greeted by the RENS SolarTracker dashboard home page!

From here, click on the *Admin* tab on the left side panel, and log in as admin.

Default admin password: 23456789

On the admin control panel, paste the application ID obtained from ChirpStack to the *Application ID* field in MQTT setup, and click *Submit*
Paste the API key obtained from ChirpStack to the *API key* field in ChirpStack API setup, and click *Submit*



Message delta time

current=15s

MQTT setup

Application ID

Broker address, current=localhost

Chirpstack API setup

API key

Server address, current=localhost:8080

Defaults

Siren on time (0-255s), current=5

Insolation (%), current=100

Latitude, current=43

Longitude, current=16

Time offset (-65535 to 65535), current=3600

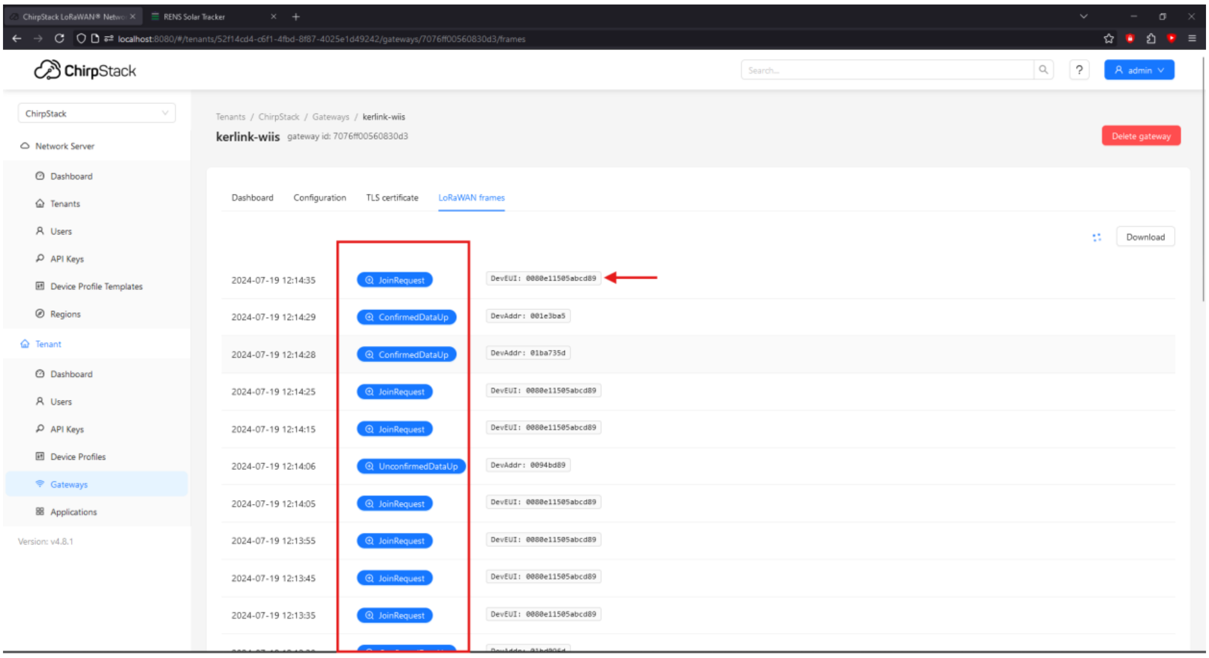
Figure 5 - API setup

2.1.3 Connecting end devices

To connect an end device to ChirpStack we need to turn it on and add it to the *Application* that you created earlier. This can be done via the ChirpStack web interface or via the provided console application, for adding devices in bulk.

2.1.3.1 *Adding devices manually*

Click on *Gateways* on the sidebar and then on the ID of the gateway to open its control panel. Once you turn on an end device (or restart it), *JoinRequest* messages will appear in the *LoRaWAN frames* tab. These indicate that the end device is trying to connect to the gateway



localhost:8080/#/tenants/52f14cd4-c6f1-4fbd-8f87-4025e1d49242/gateways/7076ff00560830d3/frames

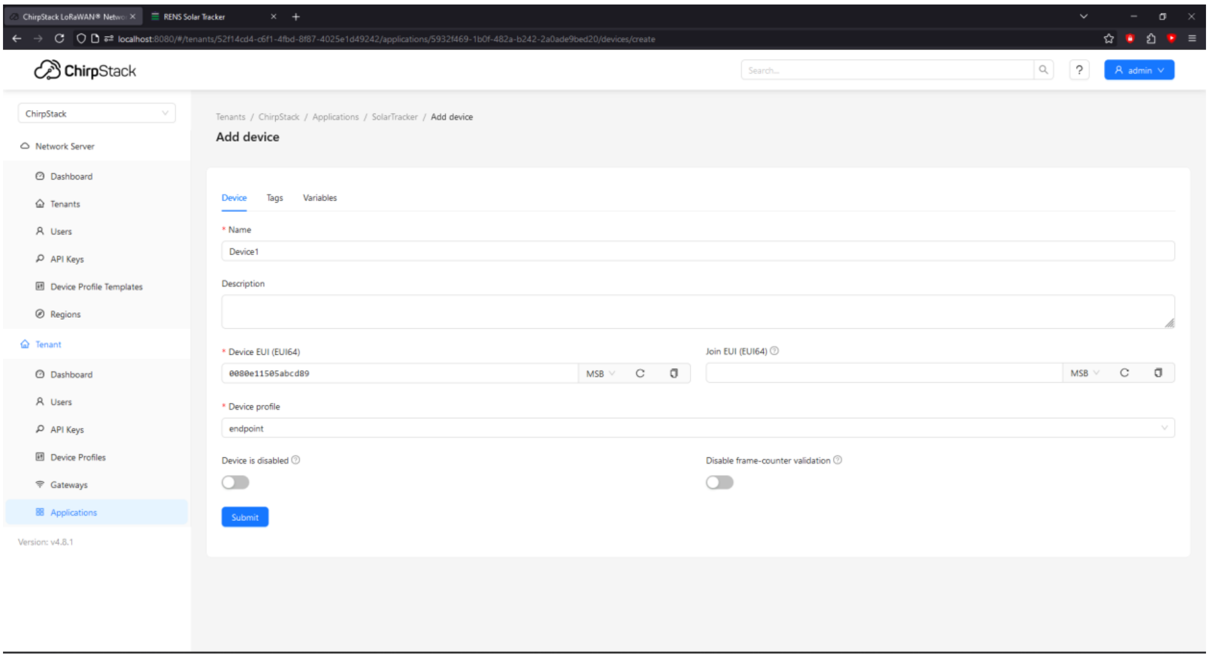
kerlink-wiis gateway id: 7076ff00560830d3

2024-07-19 12:14:35	JoinRequest	DevEUI: 0080e11505abcd89
2024-07-19 12:14:29	ConfirmedDataUp	DevAddr: 001e3ba5
2024-07-19 12:14:28	ConfirmedDataUp	DevAddr: 01ba735d
2024-07-19 12:14:25	JoinRequest	DevEUI: 0080e11505abcd89
2024-07-19 12:14:15	JoinRequest	DevEUI: 0080e11505abcd89
2024-07-19 12:14:06	UnconfirmedDataUp	DevAddr: 0094bd89
2024-07-19 12:14:05	JoinRequest	DevEUI: 0080e11505abcd89
2024-07-19 12:13:55	JoinRequest	DevEUI: 0080e11505abcd89
2024-07-19 12:13:45	JoinRequest	DevEUI: 0080e11505abcd89
2024-07-19 12:13:35	JoinRequest	DevEUI: 0080e11505abcd89

Figure 6 - Join requests

Note the DevEUI value in these requests. You will need this to add the device to the application.

Click on *Applications* in the sidebar, then on the application you created, and then on *Add device* on the top right.



* Name: Device1

Description:

* Device EUI (EUI64): 0080e11505abcd89 MSB

Join EUI (EUI64): MSB

* Device profile: endpoint

Device is disabled: off

Disable frame-counter validation: off

Figure 7 - Adding a device to the application

Choose a name for the device, paste the *DevEUI* to the Device EUI field, and choose the profile you created earlier for the *Device profile.* Leave other fields as default.

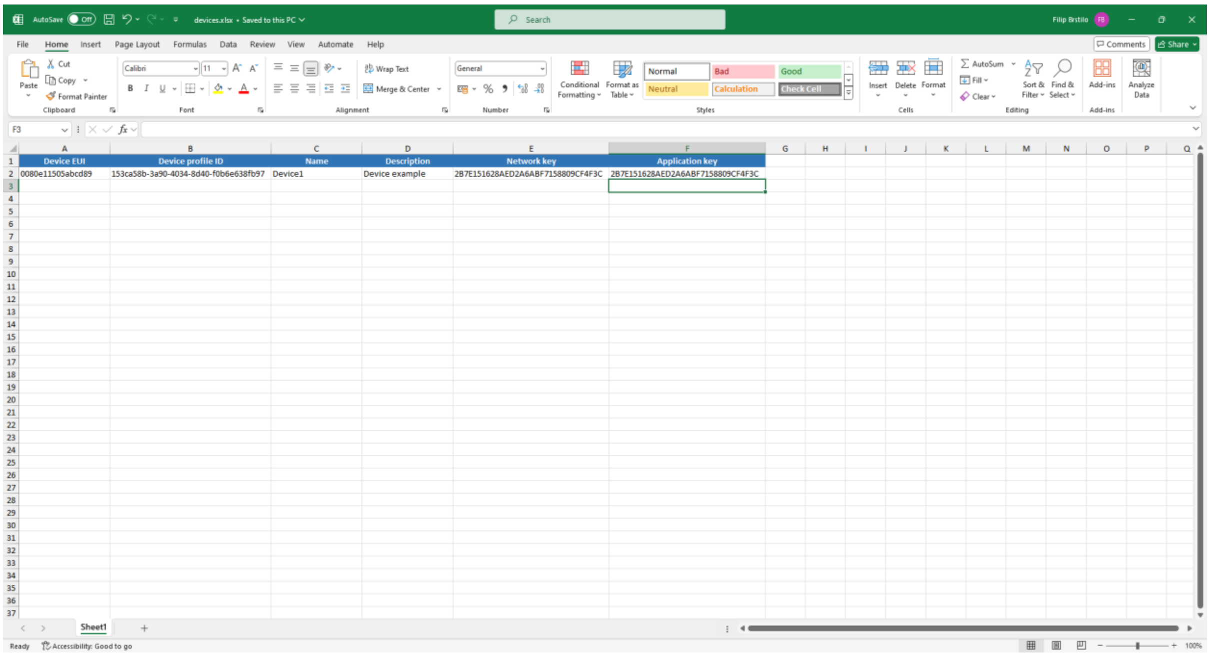
ChirpStack will now ask you for an *Application key*. Input the following value:
2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C

Go back to *Gateways->(gateway)->LoRaWAN frames* and you should see a 'Join request' message, followed by a 'Join accepted' message, meaning that the end device has successfully connected. Refresh the SolarTracker to show the device in the *devices* tab (located in the sidebar). The devices are listed sequentially as *Device 1, Device 2…* in the order that they were initially connected to the gateway.

Repeat this process for each device you wish to add.

2.1.3.2 *Adding devices in bulk*

If you wish to add multiple devices at once to your application (e.g. on initial setup) you can use the provided *bulk-add-devices.py* script.
Before you run the script, you need to create a spreadsheet with the data of all the devices you wish to add. An example spreadsheet is provided along with the script.

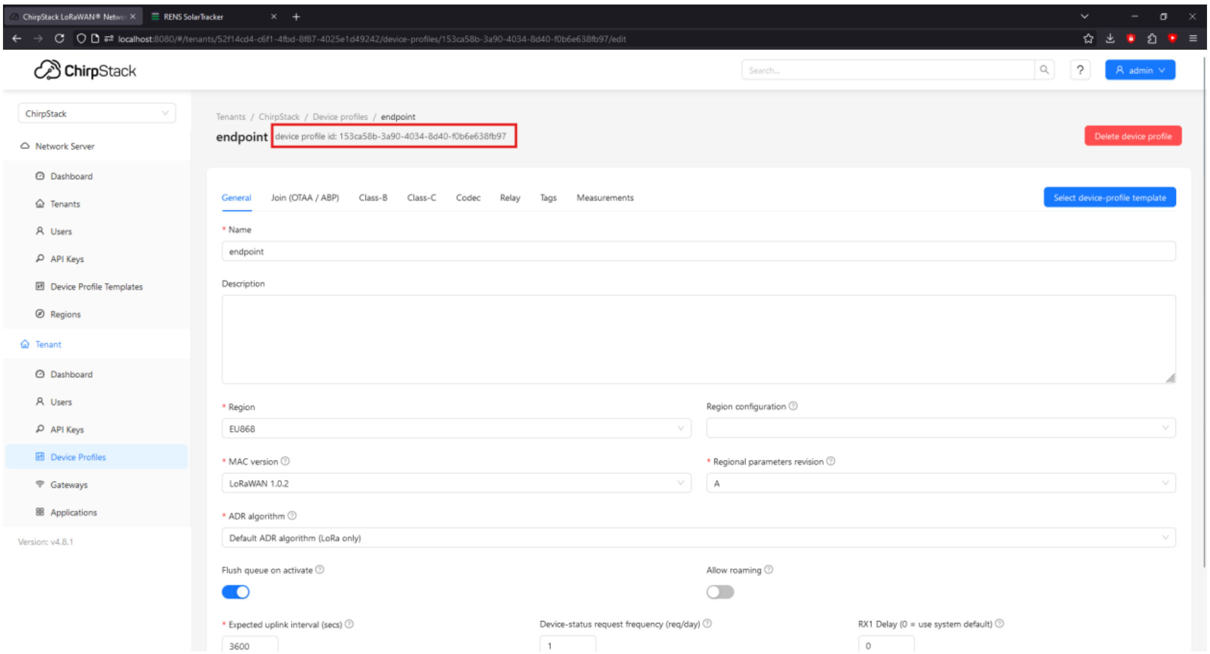


Device EUI	Device profile ID	Name	Description	Network key	Application key
0080e11505abcd89	153ca58b-3a90-4034-8d40-f0b6e638fb97	Device1	Device example	2B7E151628AED2A6ABF7158809CF4F3C	2B7E151628AED2A6ABF7158809CF4F3C

Figure 8 - Bulk devices spreadsheet example

For the bulk adding operation to succeed, the spreadsheet needs to be of the same format as the one provided. The input data is:

- Device EUI – 64 bit HEX value starting with 0080. Can be found written on the device, or using the JoinRequests in ChirpStack, like when Adding devices manually
- Device profile ID – Can be found in the ChirpStack dashboard: Device Profile->(profile you created), at the top of the page. An example image is located below this list
- Name – A unique device name
- Description (optional)
- Network key/Application key – use value 2B7E151628AED2A6ABF7158809CF4F3C



localhost:8080/#/tenants/52f14cd4-c6f1-4fbd-8f87-4025e1d49242/device-profiles/153ca58b-3a90-4034-8d40-f0b6e638fb97/edit

Tenants / ChirpStack / Device profiles / endpoint

endpoint device profile id: 153ca58b-3a90-4034-8d40-f0b6e638fb97

* Name: endpoint

Description:

* Region: EU868

Region configuration:

* MAC version: LoRaWAN 1.0.2

* Regional parameters revision: A

* ADR algorithm: Default ADR algorithm (LoRa only)

Flush queue on activate: on

Allow roaming: off

* Expected uplink interval (secs): 3600

Device-status request frequency (req/day): 1

RX1 Delay (0 = use system default): 0

Figure 9 - Device profile ID

Once you have added all the device information to the spreadsheet, you can add them to the application using the command:
python bulk-add-devices.py --source devices.xlsx
(generally python bulk-add-devices.py --source <spreadsheet_filename>)

2.2 Admin panel setup

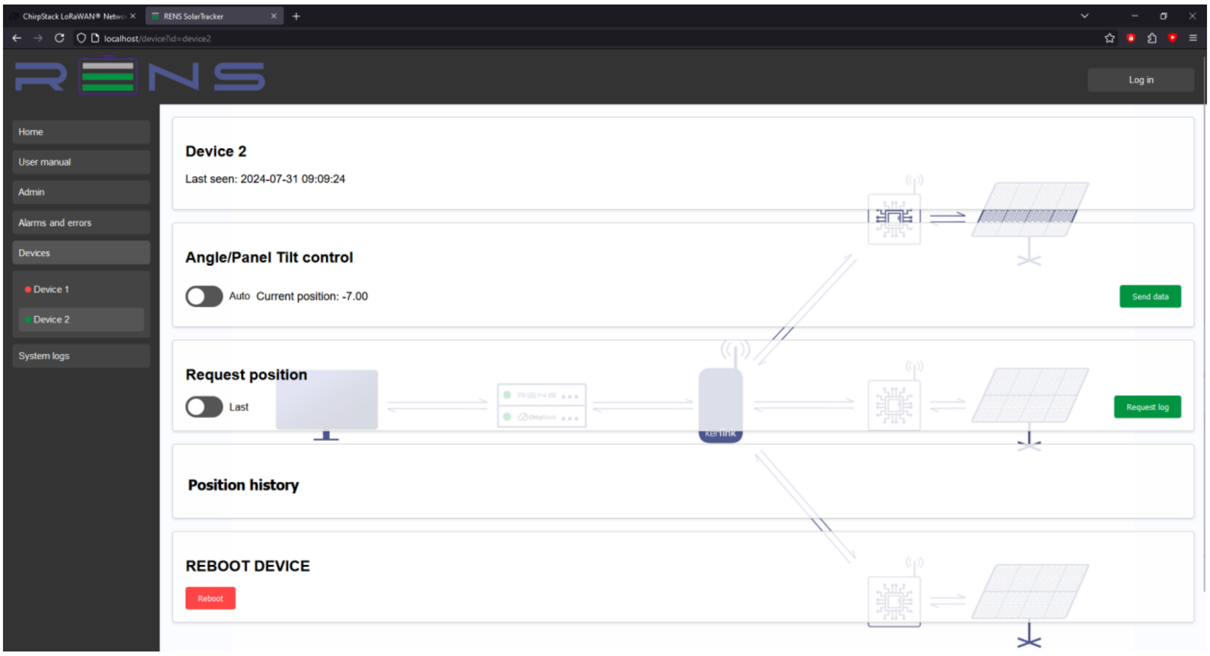
Log in to the SolarTracker dashboard (default admin password: 23456789) and click on the *Admin* button on the sidebar to go to the admin panel.

On the admin panel you can set the following:

- Message delta time – time delay between messages
- MQTT setup and ChirpStack API setup – used for connecting to the ChirpStack network server (refer to the Central control unit setup section of this manual for further help)
- Change Administrator password
- Setup default values – values sent to the end device in case an input field is left empty. (refer to the Parameters section of this manual for further help)

3 Device control – Regular user

Once an end device is connected to the ChirpStack network server, it will also be registered to the SolarTracker dashboard, and it will appear in the *Devices* tab on the sidebar after a refresh. End devices are listed sequentially as *Device 1, Device 2*... sequentially, in the order that they were first registered to the dashboard.



Device 2

Last seen: 2024-07-31 09:09:24

Angle/Panel Tilt control

Current position: -7.00

Request position

Position history

REBOOT DEVICE

Figure 10 - SolarTracker regular user device control panel

Clicking a device's name in the sidebar opens that device's control panel where a non-admin user can control the following:

- Angle/Panel Tilt control
  - Switch to auto – place the toggle to the *Auto* position and click the *Send data* button
  - Manual control – place the toggle to the *Manual* position, input the wanted angle in the text box and click the *Send data* button
- Request position
  - Last – Requests the current position of the device and appends it to a log
  - All – Requests all historical position data on the device and appends it to a log
- Position history
  - View and control logs generated by *Request position*.
  - Logs names are the dates (and hours) of when they were requested
  - Contents of logs and available controls are explained in the Logs, errors and alarms section of this manual
- Reboot device
  - Remotely reboot the end device. Attempting to control the device while it is not fully booted will result in an error

Tilt control, requesting positions, rebooting and parameter settings (available if user is admin) will lock out device control for *Message timeout* seconds (defined in the Admin control panel). This is to give enough time for the end device to respond.

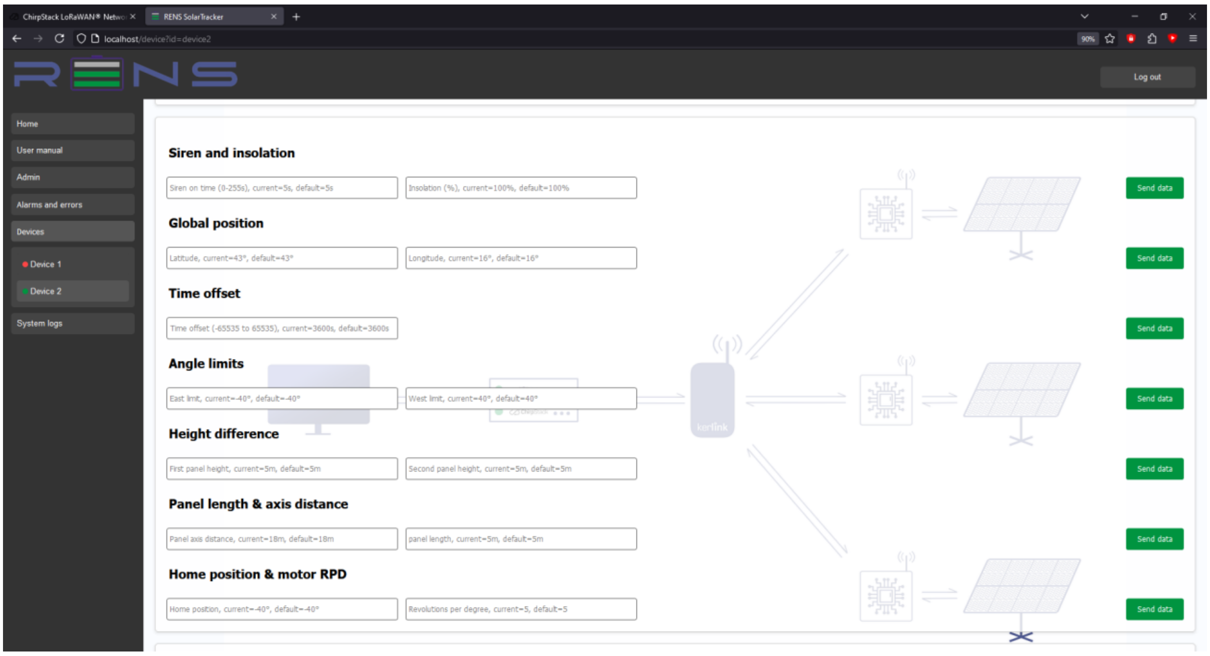
Requesting all position logs and resetting the device parameters to default causes a longer lock out, because these actions require multiple messages to be sent to and from an end device.

4 Device control – Admin

To access the admin device control panel, log in with the administrator password, and select the device you wish to control.

The admin device control panel contains all the options as the regular user control panel (refer to the Device control – Regular user section of this manual for further help), as well as the options:

- Parameter control
  - Explanations of these parameters are in the Parameters section of this manual
  - Sending parameters with empty input boxes will leave those values unchanged
- Reset to defaults
  - Reset all device parameters to their default values (defined in the Admin control panel)
- Delete device
  - Unregister a device from the SolarTracker



Siren and insolation

Siren on time (0-255s), current=5s, default=5s

Insolation (%), current=100%, default=100%

Global position

Latitude, current=43°, default=43°

Longitude, current=16°, default=16°

Time offset

Time offset (-65535 to 65535), current=3600s, default=3600s

Angle limits

East limit, current=-40°, default=-40°

West limit, current=40°, default=40°

Height difference

First panel height, current=5m, default=5m

Second panel height, current=5m, default=5m

Panel length & axis distance

Panel axis distance, current=18m, default=18m

panel length, current=5m, default=5m

Home position & motor RPD

Home position, current=-40°, default=-40°

Revolutions per degree, current=5, default=5

Figure 11 - SolarTracker admin device control panel

5 Parameters

Siren on time – After a reboot, the device will sound an alarm for this many seconds before moving to its home position

Insolation – Maximum area of the next solar panel that this solar panel is allowed to ‘hide’ from the Sun, expressed as a percentage:
0% - next panel is fully hidden from the Sun
100% - next panel is fully exposed to the Sun
Used during automatic panel tilt control to maximise energy production

Global position - Set the latitude and longitude of the system. Used together with the current time to calculate the appropriate panel tilt angle

Time offset - Set the device time offset in seconds. Used together with the current UTC time to adjust for the actual time zone

Angle limits - Set the maximum east/west panel angle. Used to protect the system from physical damage. The device will never exceed these limits, even if manually ordered to

Height difference - Set the heights of this (first) and next (second) panel, as viewed from the Sun. This data is used to calculate the insolation of the next panel

Panel length and axis distance - Set the panel length and distance between panel axes. This data is used to calculate the insolation percentage of the next panel

Home position and motor revolutions per degree
Home position – the position where the solar panel will first go to after booting up
Motor RPD - ratio between motor revolutions and solar panel angle change

6 Logs, errors and alarms

Logs are divided into 3 categories – alarms and errors, position logs, and general system logs.

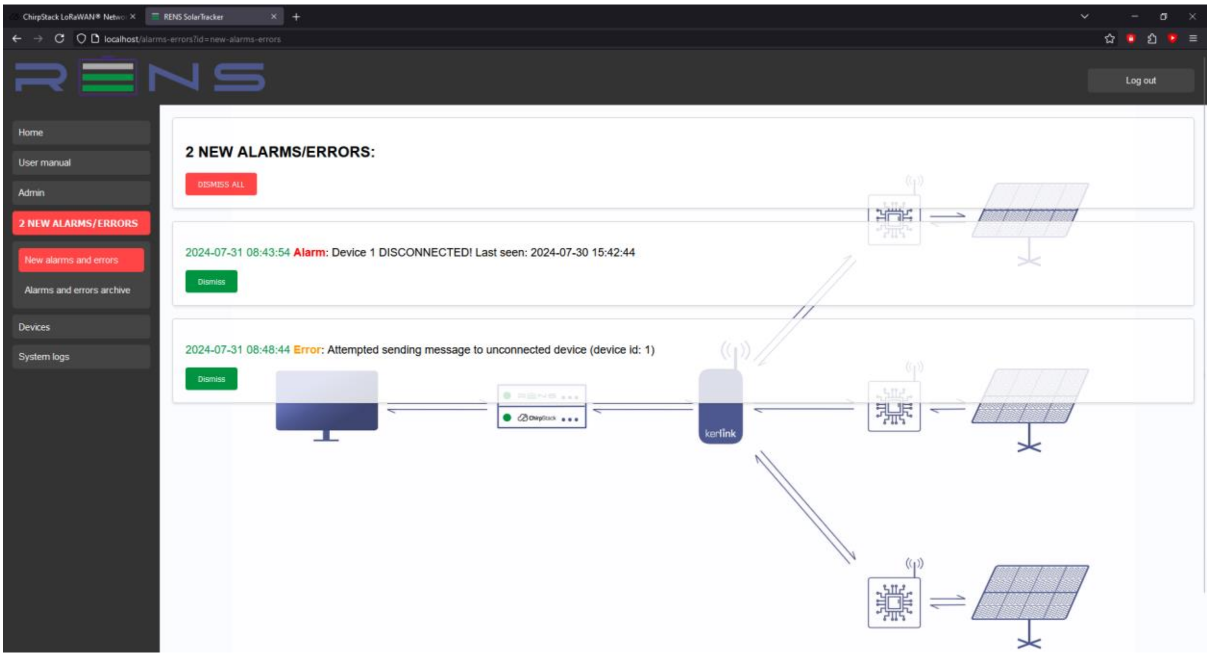
Logs are stored on the server as .log files, and can be viewed, downloaded and deleted from the SolarTracker web interface.

All logs older than 30 days are automatically deleted

6.1 Alarms and errors

Once SolarTracker receives an alarm or error message from an end device, the log will be stored in the *Alarms and errors archive*, and held for manual review in the *New alarms/errors* tab. Both new and archived alarms are accessible through the *Alarms and errors* button on the sidebar.

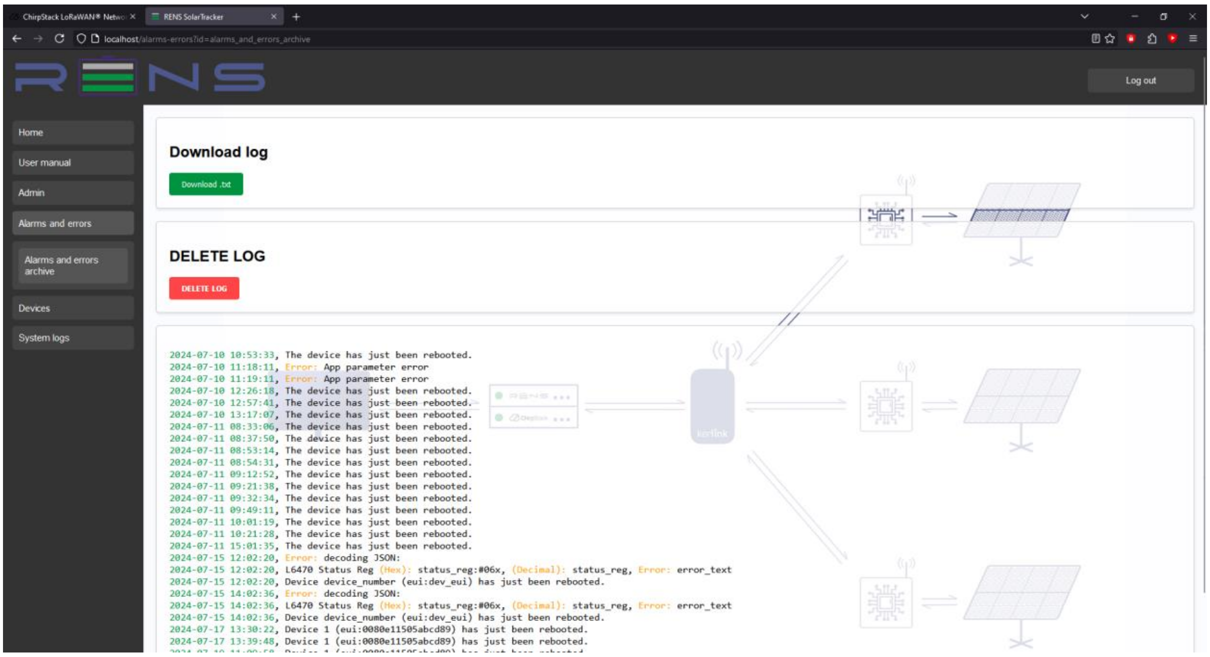
If there are any alarms/errors that were not manually reviewed and dismissed, the *Alarms and errors* button will show up as red, and will contain the *New alarms/errors* tab.



2024-07-31 08:43:54 Alarm: Device 1 DISCONNECTED! Last seen: 2024-07-30 15:42:44

2024-07-31 08:48:44 Error: Attempted sending message to unconnected device (device id: 1)

Figure 12 - New alarms/errors received

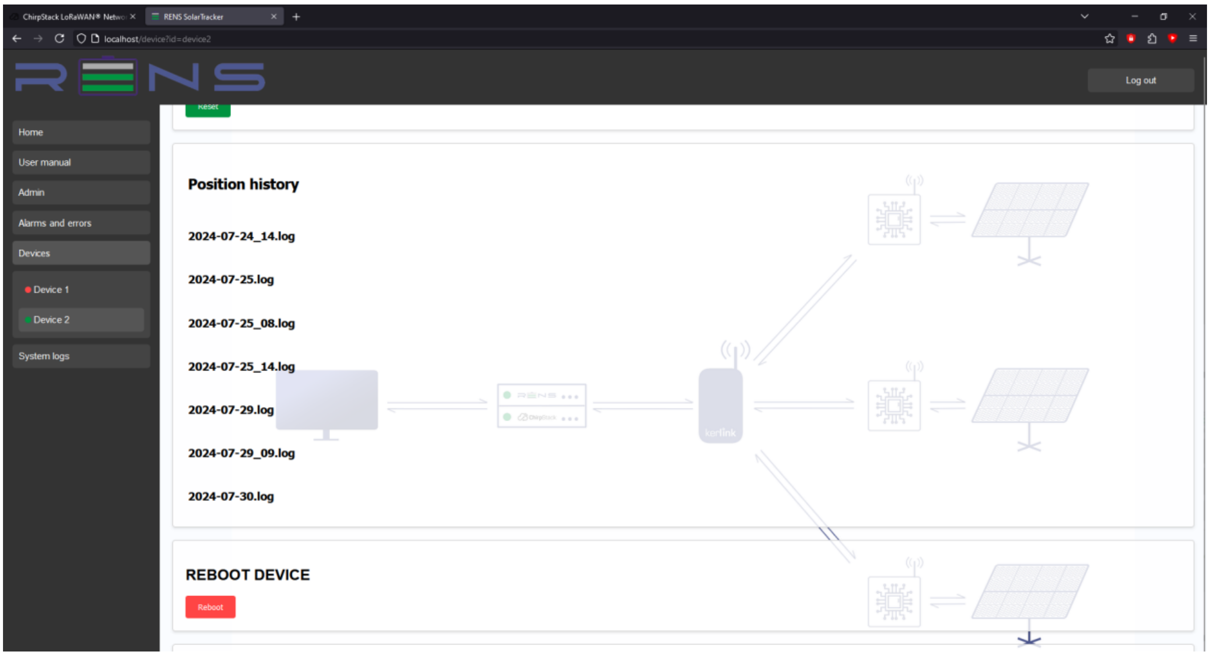


```
2024-07-10 10:53:33, The device has just been rebooted.
2024-07-10 11:18:11, Error: App parameter error
2024-07-10 11:19:11, Error: App parameter error
2024-07-10 12:26:18, The device has just been rebooted.
2024-07-10 12:57:41, The device has just been rebooted.
2024-07-10 13:17:07, The device has just been rebooted.
2024-07-11 08:33:06, The device has just been rebooted.
2024-07-11 08:37:50, The device has just been rebooted.
2024-07-11 08:53:14, The device has just been rebooted.
2024-07-11 08:54:31, The device has just been rebooted.
2024-07-11 09:12:52, The device has just been rebooted.
2024-07-11 09:21:38, The device has just been rebooted.
2024-07-11 09:32:34, The device has just been rebooted.
2024-07-11 09:49:11, The device has just been rebooted.
2024-07-11 10:01:19, The device has just been rebooted.
2024-07-11 10:21:28, The device has just been rebooted.
2024-07-11 15:01:35, The device has just been rebooted.
2024-07-15 12:02:20, Error: decoding JSON:
2024-07-15 12:02:20, L6470 Status Reg (Hex): status_reg:#06x, (Decimal): status_reg, Error: error_text
2024-07-15 12:02:20, Device device_number (eui:dev_eui) has just been rebooted.
2024-07-15 14:02:36, Error: decoding JSON:
2024-07-15 14:02:36, L6470 Status Reg (Hex): status_reg:#06x, (Decimal): status_reg, Error: error_text
2024-07-15 14:02:36, Device device_number (eui:dev_eui) has just been rebooted.
2024-07-17 13:30:22, Device 1 (eui:0080e11505abcd89) has just been rebooted.
2024-07-17 13:39:48, Device 1 (eui:0080e11505abcd89) has just been rebooted.
```

Figure 13 - Alarms/errors archive

6.2 Position logs

A device’s position logs are accessible through the device’s control panel, in the *Position history* tab. Click on the *Position history* tab to show all available logs.



Position history

2024-07-24_14.log

2024-07-25.log

2024-07-25_08.log

2024-07-25_14.log

2024-07-29.log

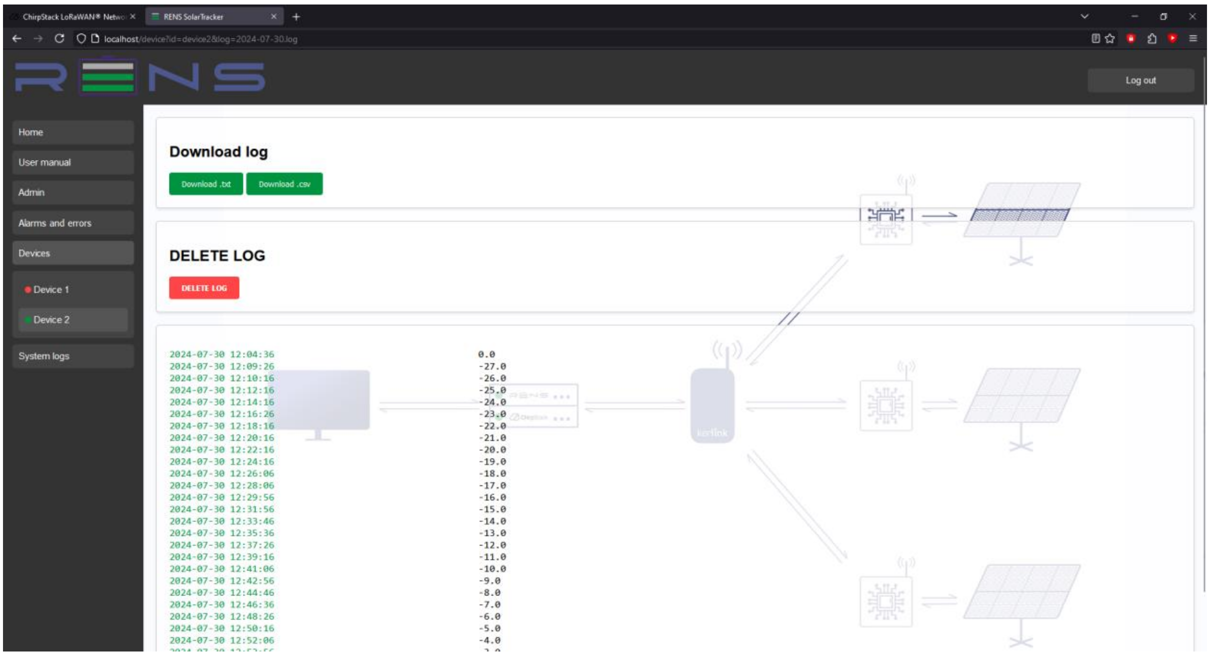
2024-07-29_09.log

2024-07-30.log

REBOOT DEVICE

Figure 14 - Expanded list of a device's position history logs

The log names are timestamps of when they were requested, specific to the day (in case of requesting all position logs) or to the hour (in case of requesting the last log)



Download log

DELETE LOG

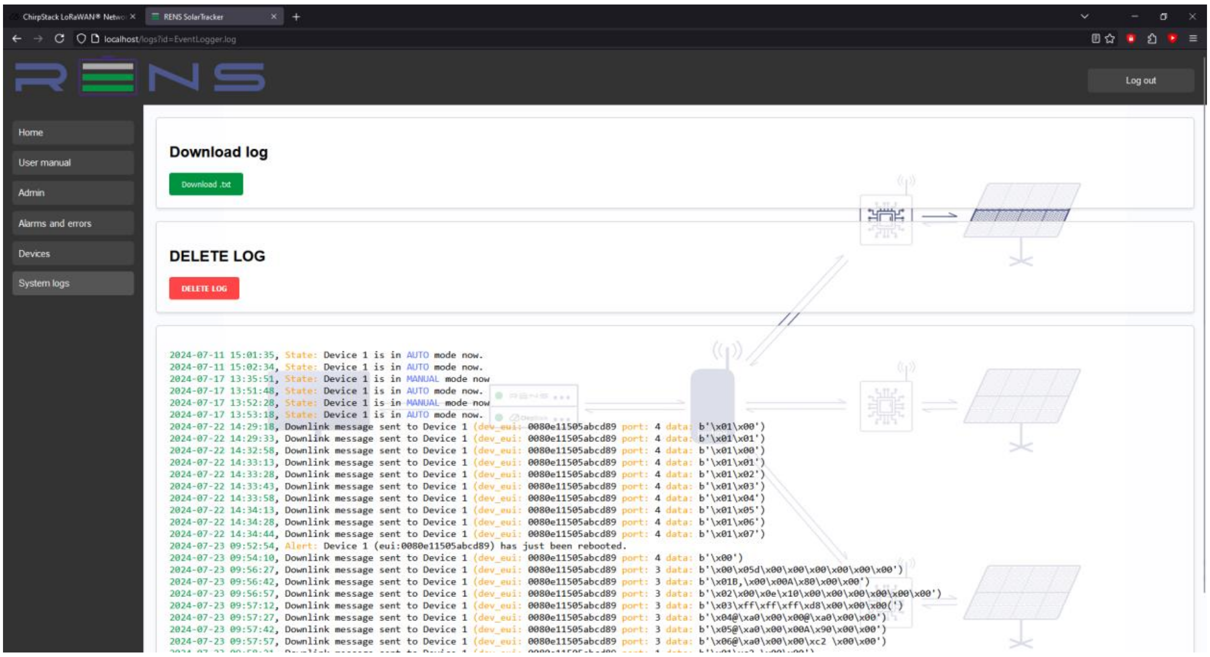
```
2024-07-30 12:04:36                                 0.0
2024-07-30 12:09:26                                 -27.0
2024-07-30 12:10:16                                 -26.0
2024-07-30 12:12:16                                 -25.0
2024-07-30 12:14:16                                 -24.0
2024-07-30 12:16:26                                 -23.0
2024-07-30 12:18:16                                 -22.0
2024-07-30 12:20:16                                 -21.0
2024-07-30 12:22:16                                 -20.0
2024-07-30 12:24:16                                 -19.0
2024-07-30 12:26:06                                 -18.0
2024-07-30 12:28:06                                 -17.0
2024-07-30 12:29:56                                 -16.0
2024-07-30 12:31:56                                 -15.0
2024-07-30 12:33:46                                 -14.0
2024-07-30 12:35:36                                 -13.0
2024-07-30 12:37:26                                 -12.0
2024-07-30 12:39:16                                 -11.0
2024-07-30 12:41:06                                 -10.0
2024-07-30 12:42:56                                 -9.0
2024-07-30 12:44:46                                 -8.0
2024-07-30 12:46:36                                 -7.0
2024-07-30 12:48:26                                 -6.0
2024-07-30 12:50:16                                 -5.0
2024-07-30 12:52:06                                 -4.0
2024-07-30 12:53:56                                 -3.0
```

Figure 15 - Position log example

The log entries consist of the date and time (specific to the second), and the solar panel tilt value at that time.

In addition to downloading these logs as a .txt file, SolarTracker also provides a *download as .csv* option, for easier data analysis using external programs.

6.3 System logs



Download log

Download .txt

DELETE LOG

DELETE LOG

```
2024-07-11 15:01:35, State: Device 1 is in AUTO mode now.
2024-07-11 15:02:34, State: Device 1 is in AUTO mode now.
2024-07-17 13:35:51, State: Device 1 is in MANUAL mode now
2024-07-17 13:51:48, State: Device 1 is in AUTO mode now.
2024-07-17 13:52:28, State: Device 1 is in MANUAL mode now
2024-07-17 13:53:18, State: Device 1 is in AUTO mode now.
2024-07-22 14:29:18, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 4 data: b'\x01\x00')
2024-07-22 14:29:33, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 4 data: b'\x01\x01')
2024-07-22 14:32:58, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 4 data: b'\x01\x00')
2024-07-22 14:33:13, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 4 data: b'\x01\x01')
2024-07-22 14:33:28, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 4 data: b'\x01\x02')
2024-07-22 14:33:43, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 4 data: b'\x01\x03')
2024-07-22 14:33:58, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 4 data: b'\x01\x04')
2024-07-22 14:34:13, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 4 data: b'\x01\x05')
2024-07-22 14:34:28, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 4 data: b'\x01\x06')
2024-07-22 14:34:44, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 4 data: b'\x01\x07')
2024-07-23 09:52:54, Alert: Device 1 (eui:0080e11505abcd89) has just been rebooted.
2024-07-23 09:54:10, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 4 data: b'\x00')
2024-07-23 09:56:27, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 3 data: b'\x00\x05d\x00\x00\x00\x00\x00\x00')
2024-07-23 09:56:42, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 3 data: b'\x01B,\x00\x00A\x80\x00\x00')
2024-07-23 09:56:57, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 3 data: b'\x02\x00\x0e\x10\x00\x00\x00\x00\x00\x00')
2024-07-23 09:57:12, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 3 data: b'\x03\xff\xff\xff\xd8\x00\x00\x00(')
2024-07-23 09:57:27, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 3 data: b'\x04@\xa0\x00\x00@\xa0\x00\x00')
2024-07-23 09:57:42, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 3 data: b'\x05@\xa0\x00\x00A\x90\x00\x00')
2024-07-23 09:57:57, Downlink message sent to Device 1 (dev_eui: 0080e11505abcd89 port: 3 data: b'\x06@\xa0\x00\x00\xc2 \x00\x00')
```

Device state changes (auto/manual), device reboots, and other non-critical system messages are stored in the *System logs* tab, also accessible through the sidebar. These are placed here for easier overlook of the whole system.

Just like all other logs, these can be downloaded for further analysis, or deleted if deemed unnecessary.