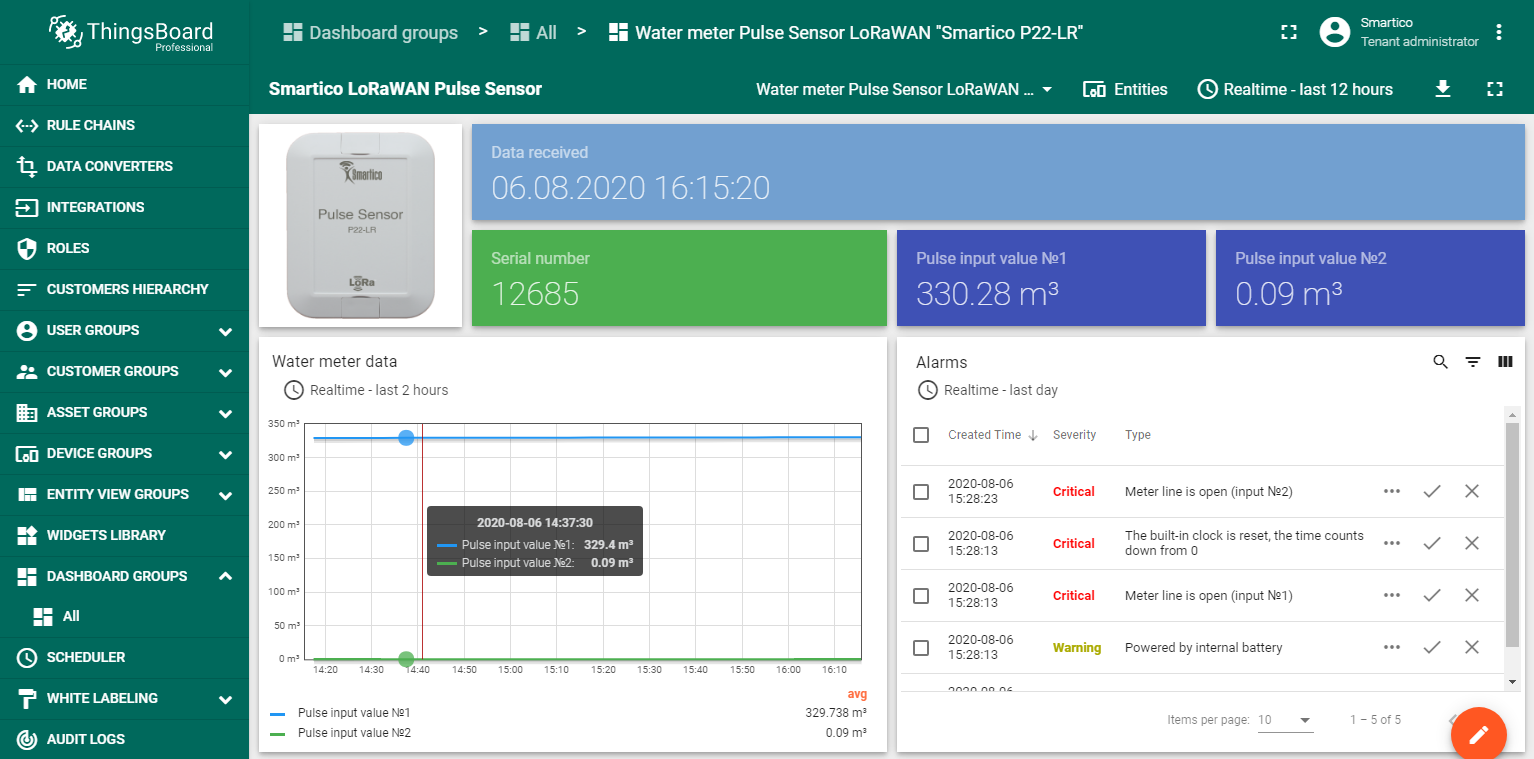
**Water meter Pulse Sensor LoRaWAN "Smartico P22-LR" telemetry upload**

[**Introduction**](https://thingsboard.io/docs/samples/nettrartu+/rtu_temp_sensor/#introduction)

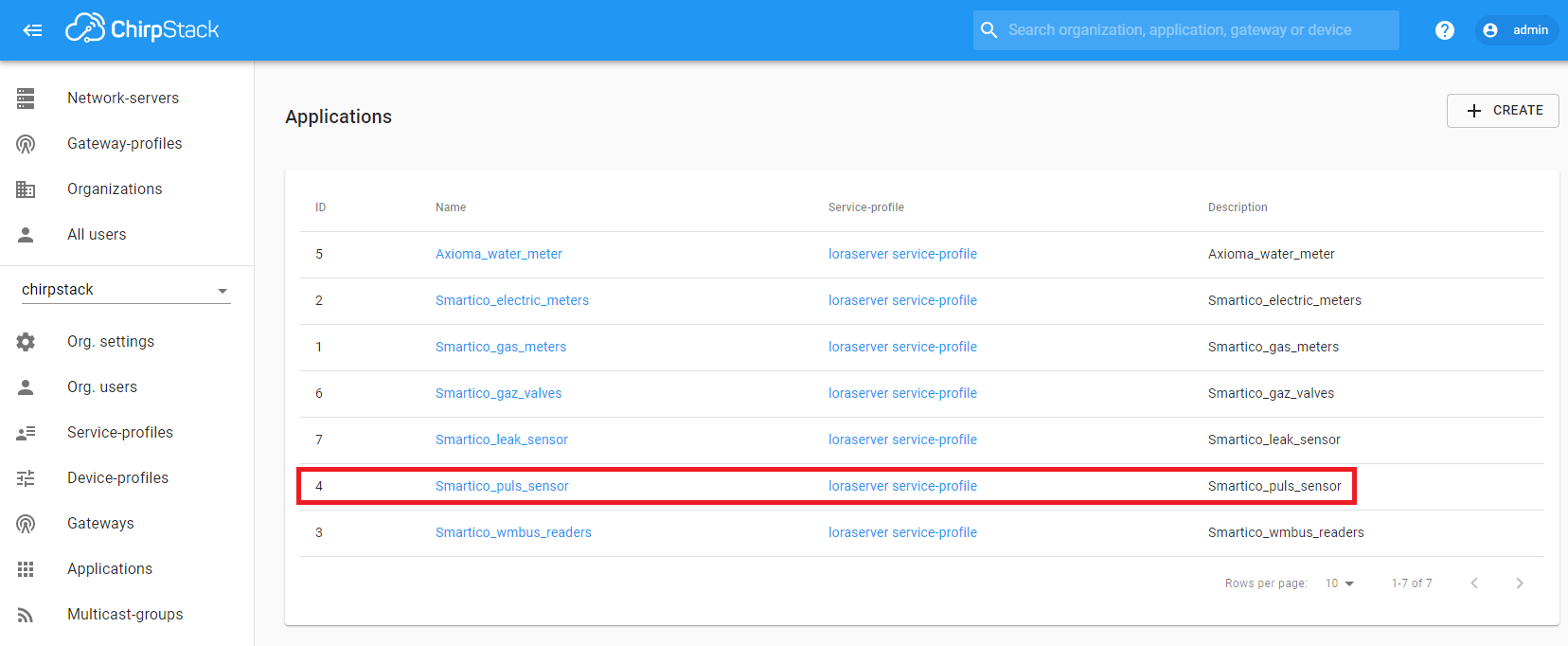
The device Pulse Sensor LoRaWAN “Smartico P22-LR” is used in various fields of industry, utilities and automation for remote data collection and transmission via LoRaWAN networks. The device has two universal pulse inputs with control of the integrity of the communication. In this example, two water meters are connected to the device. The figure shows a dashboard with processed telemetry results.

****

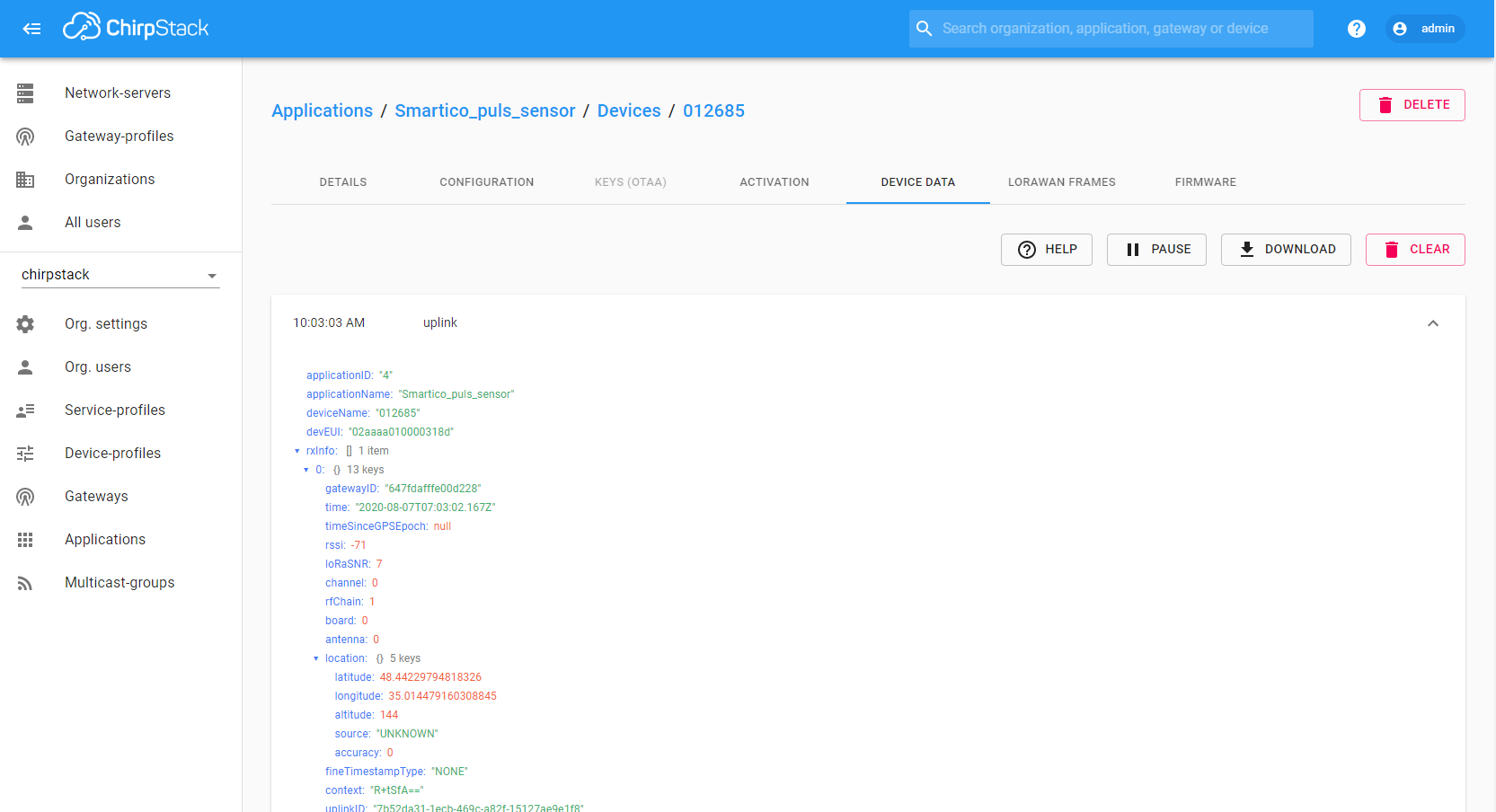
[**Prerequisites**](https://thingsboard.io/docs/user-guide/integrations/sodaq/#prerequisites)

LoRaWAN technology is used to transfer data from the Pulse Sensor P22-LR to the ThingsBoard platform. This is the wireless communication technology that allows small amounts of data to be exchanged over a long distance. First of all, you need to configure the LoRaWAN server and make sure that data from the device goes to the server. This guide uses [ChirpStack](https://www.chirpstack.io/) open-source LoRaWAN Network Server (<https://www.chirpstack.io/application-server/>).

After finishing the server configuration on the Applications page, an entry with the device type should appear in the table.



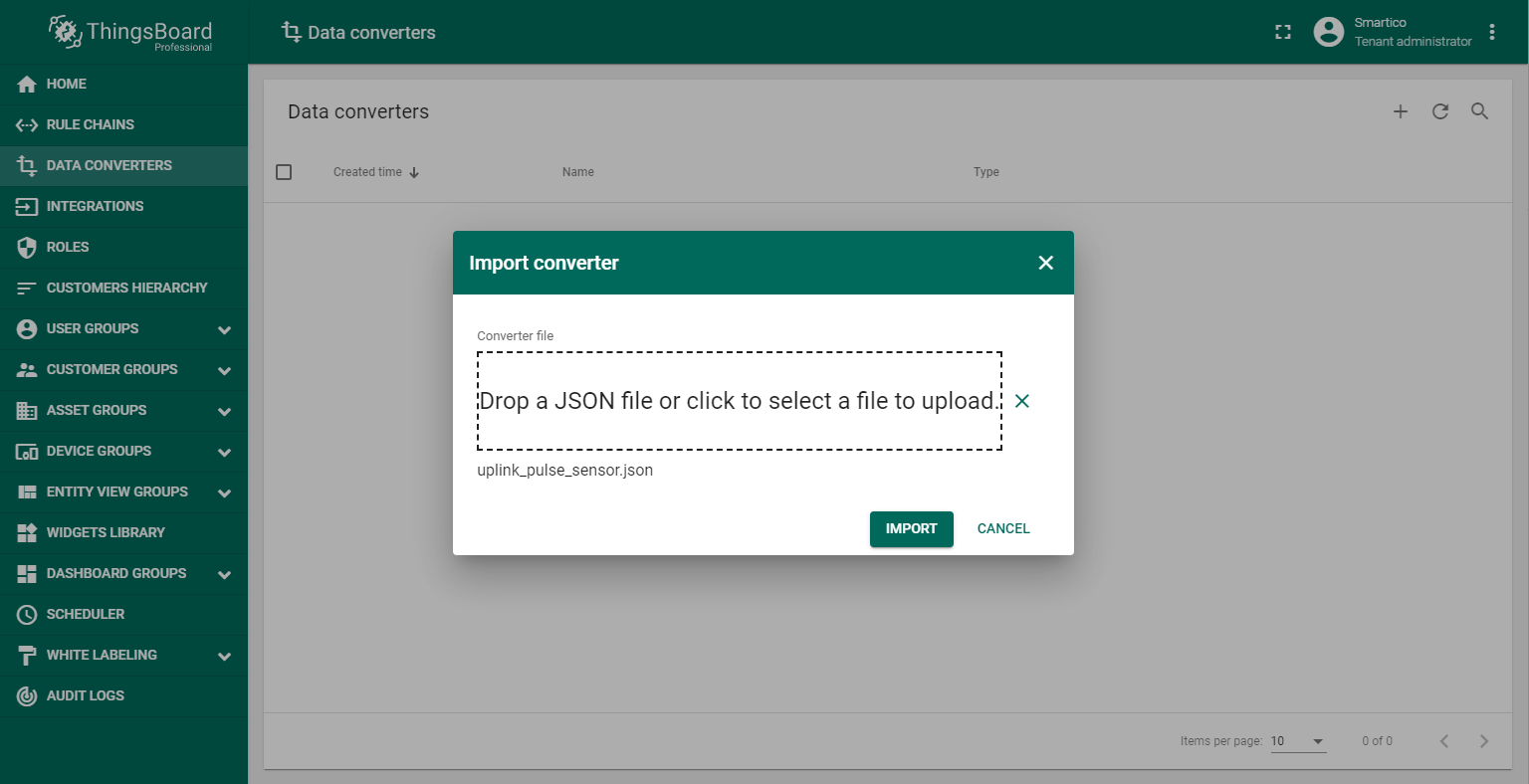
For example, we connected a device with the serial number 12685. With the correct configuration of the LoRaWAN server, we should see the data flow from the device. The frequency of data transmission from the device depends on the Pulse Sensor settings.



To be able to receive data via the MQTT protocol, you need to integrate the LoRaWAN server and the [Mosquitto](http://mosquitto.org/) MQTT broker (<https://www.chirpstack.io/application-server/integrations/mqtt/>).

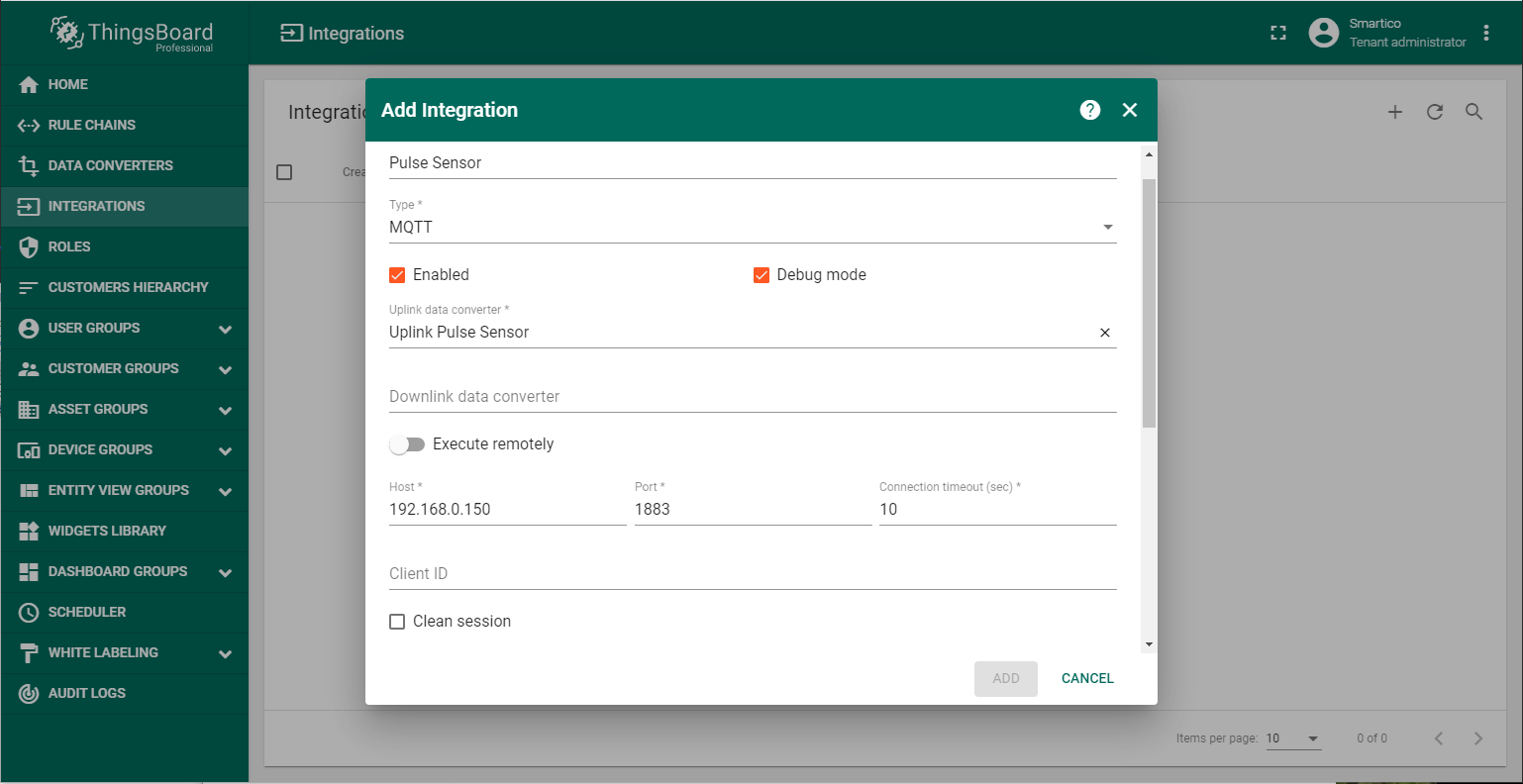
**Step 1. Creation UpLink Data Converters**

First, you should create the [Uplink Data Converter](https://thingsboard.io/docs/user-guide/integrations/#uplink-data-converter) according to the device protocol. The converter will decode incoming telemetry payload data from Pulse Sensor LoRaWAN “Smartico P22-LR” that contains in encoded Base64 string to human readable, simplified ThingsBoard data format. Import following json file with Uplink data converter (*uplink\_pulse\_sensor.json*).



**Step 2. Integration configuration.**

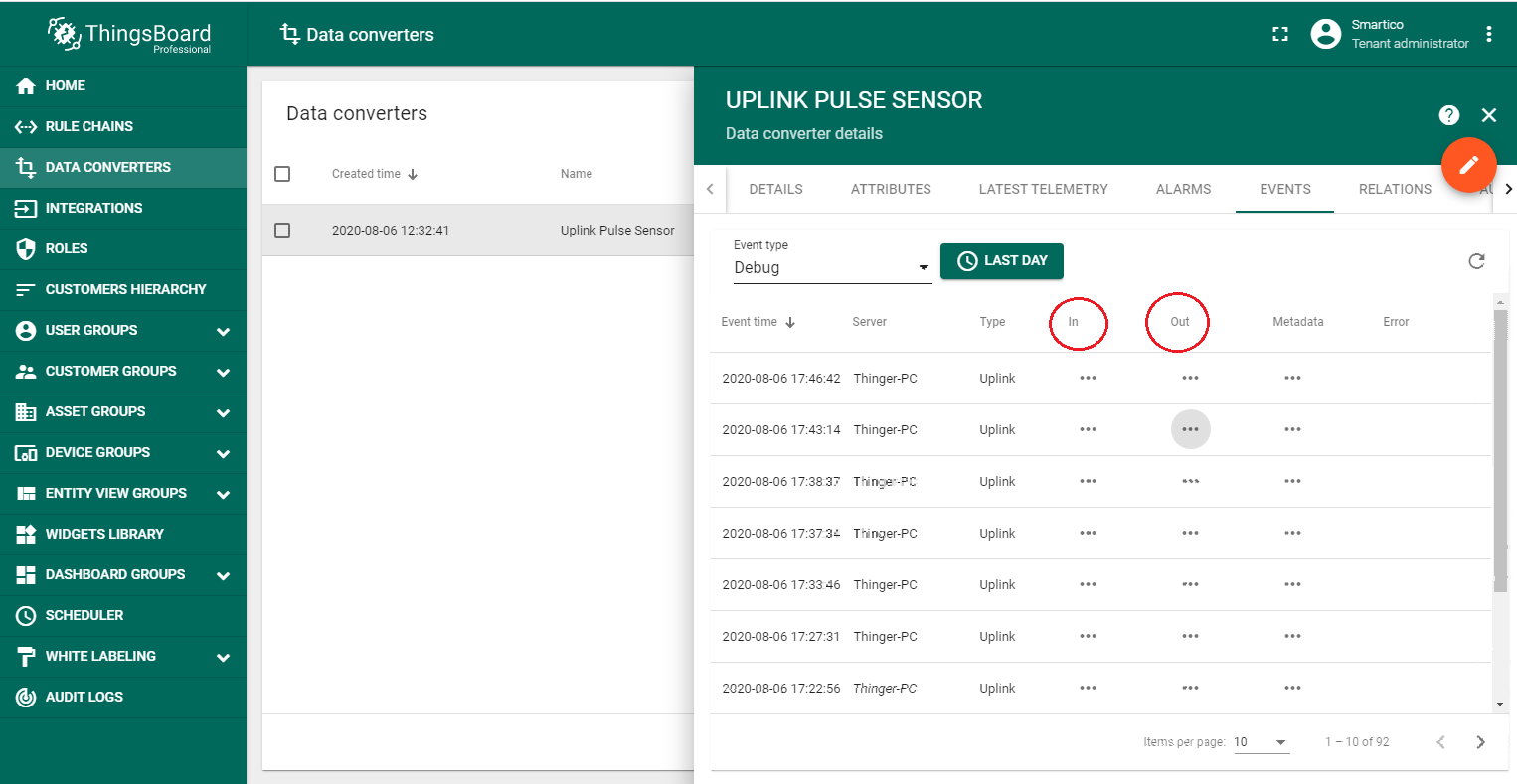
To integrate Pulse Sensor LoRaWAN “Smartico P22-LR” into ThingsBoard platform you should create a new integration as shown on the figure.



Also below you should add the topic filter according to LoRaWAN server configuration (in this example application/4/device/+/rx ). In the Host and Port fields, enter the ip-address where the MQTT broker is installed and port for working with it.

**Step 3. Verifying the receipt of data from the device.**

Connect Pulse Sensor P22-LR to transfer information. If the integration was performed without errors, after the transmission of the first telemetry, a new device with the name “012685” will appear in the DEVICE GROUPS →All. Also you can verify the input and output data, respectively, before and after conversion in DATA CONVERTERS → Uplink Pulse Sensor → EVENTS.



Input data from Pulse Sensor looks like this:

{

"applicationID": "4",

"applicationName": "Smartico\_puls\_sensor",

"deviceName": "012685",

"devEUI": "02aaaa010000318d",

"rxInfo": [{

"gatewayID": "647fdafffe00d228",

"uplinkID": "7ee7327f-9bde-4039-b993-4f8a2d593166",

"name": "tectelic\_micro\_lite\_TECH",

"rssi": -72,

"loRaSNR": 4,

"location": {

"latitude": 48.44229794818326,

"longitude": 35.014479160308845,

"altitude": 144

}

}],

"txInfo": {

"frequency": 868500000,

"dr": 0

},

"adr": true,

"fCnt": 6368,

"fPort": 7,

"data": "YgASAACBBAAAAAk="

}

The payload is contained in the “data” field and encrypted in Base64. After decoding output data will look like this:

{

"deviceName": "012685",

"deviceType": "Water Pulse",

"attributes": {

"integrationName": "Pulse Sensor Integration"

},

"telemetry": {

"NAME\_DEV": "Water meter Pulse Sensor LoRaWAN 'Smartico P22-LR'",

"SN": "012685",

"REAL\_TIME": "06.08.2020 16:15:20",

"WATER\_PULSE\_VALUE\_1": 330.28000000000003,

"WATER\_PULSE\_VALUE\_2": 0.09,

"FRAUD\_1": 2,

"FRAUD\_2": 2,

"FLG\_ERR\_PULSE\_1": 0,

"FLG\_ERR\_PULSE\_2": 0,

"FLG\_LOW\_BAT": 0,

"FLG\_MOTION\_DETECT": 1,

"FLG\_MAGNET\_DETECT": 0,

"FLG\_TAMPER\_DETECT": 0,

"FLG\_POWER\_ON": 0,

"FLG\_POWER\_BAT": 1,

"FLG\_ERR\_TIME": 1,

"FLG\_CFG\_DONE": 0

}

}

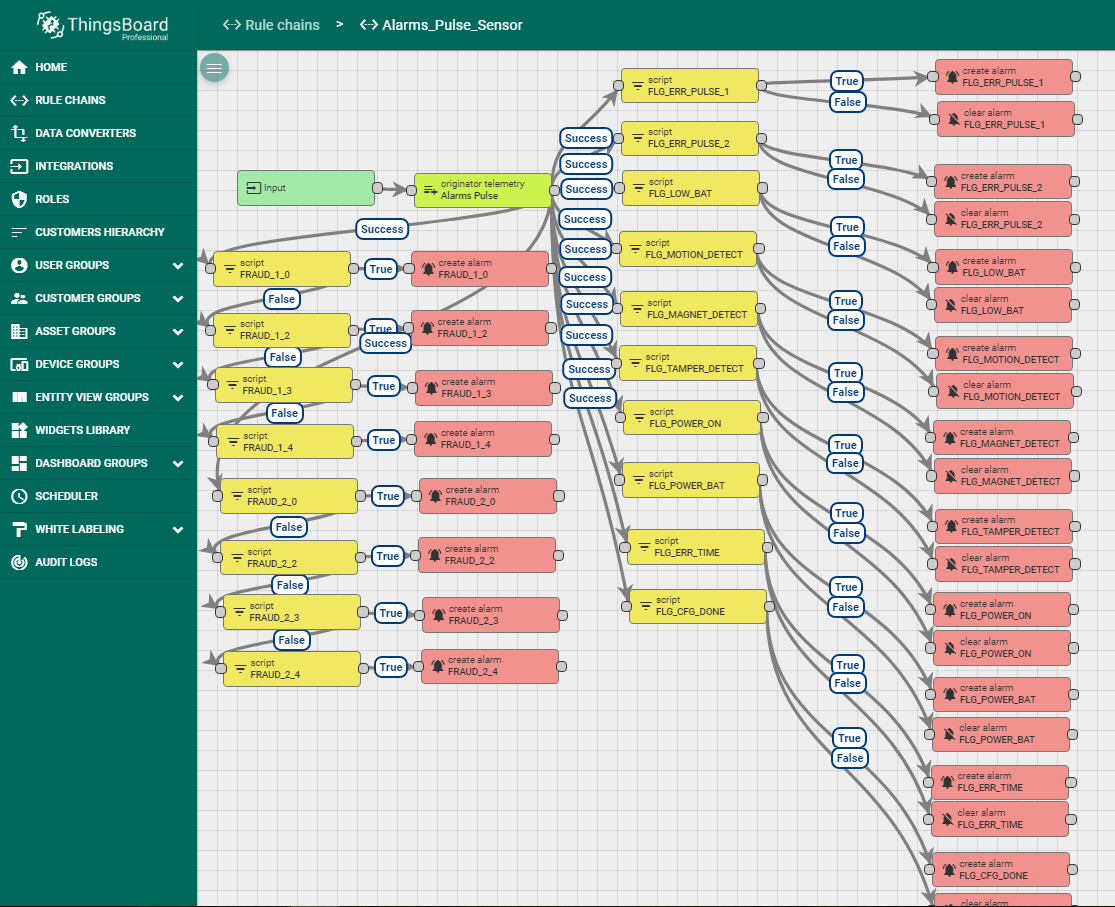
**Step 4. Creation Pulse Sensor Asset.**

To be able to display data in the dashboard, you should first create an asset and add device 012685 in the RELATIONS, as shown in the figures.

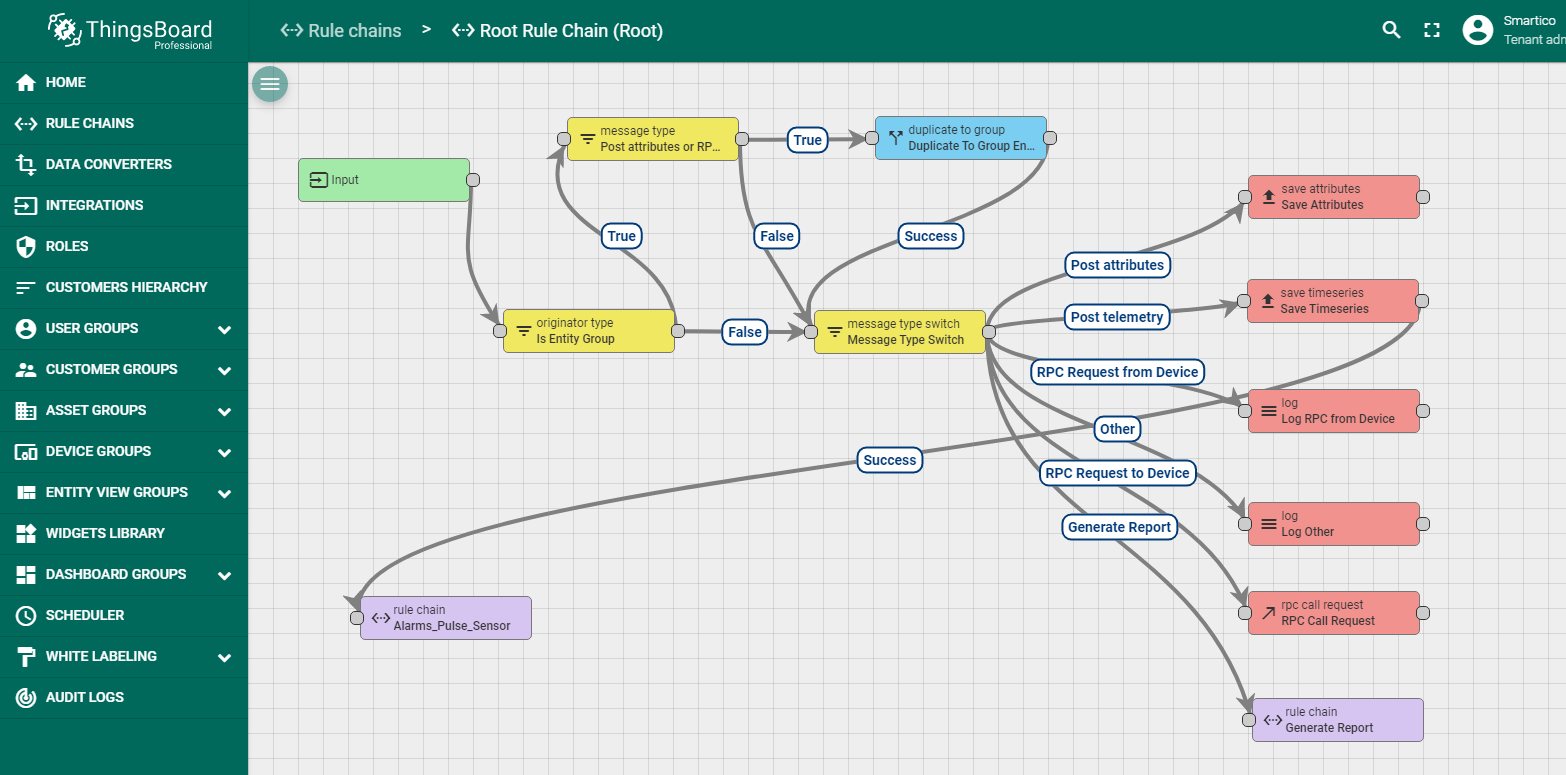
|  |  |
| --- | --- |
|  |  |

**Step 5. Rule chain import and configuration.**

In addition to meter readings, it is possible to monitor the status of the device. For example, you can get information about a low battery, opening the device case, exposure to a magnetic field, and others. This information is displayed in the Alarm widget. Therefore, you should set up Rule chain first. Import following json file with alarms (*alarms\_pulse\_sensor.json*) and save the configuration of the Rule chain in ThingsBoard.

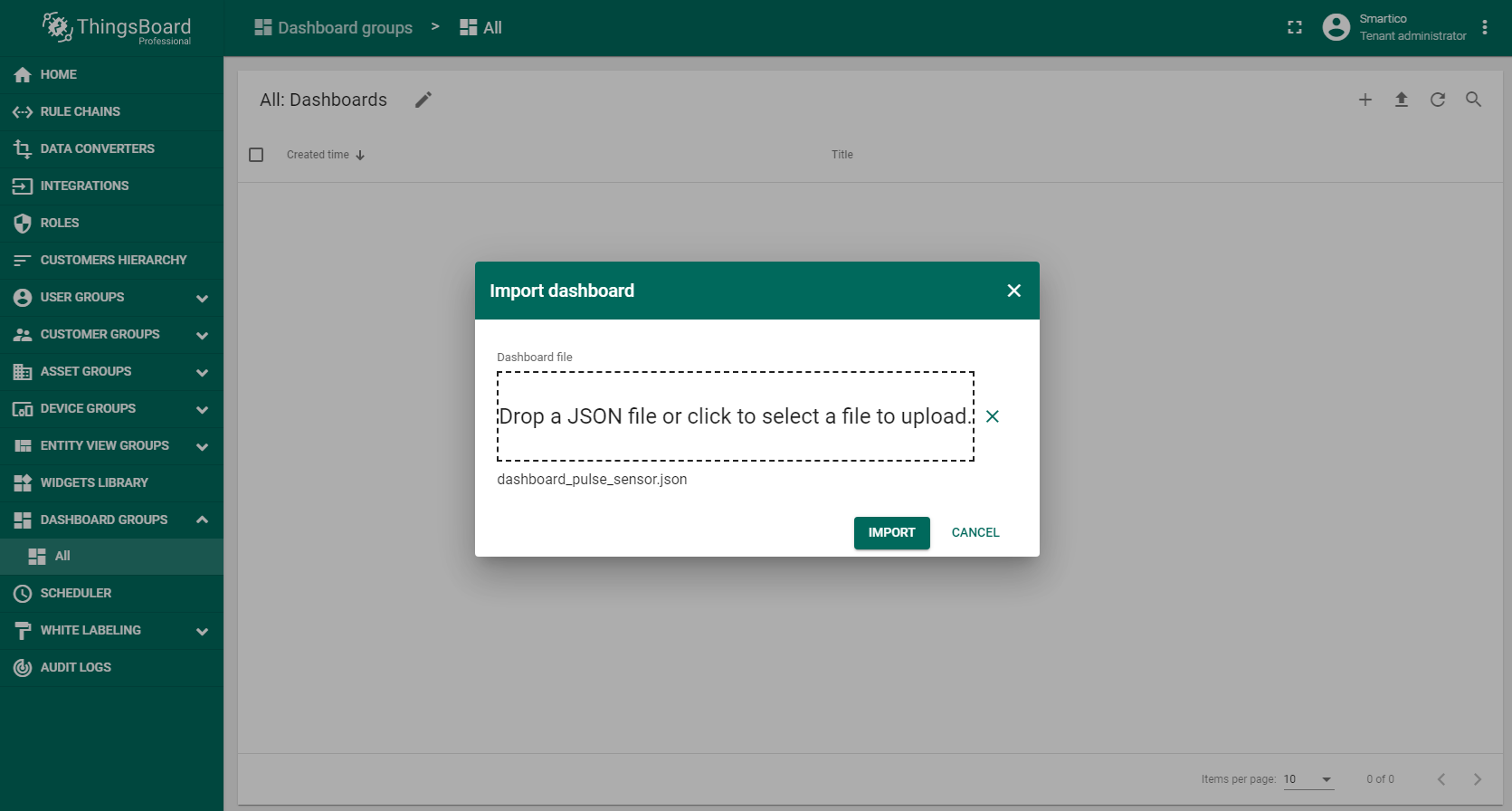
****

Then configure Root Rule chain. You should add in Root Rule chain Alarms Pulse Sensor as it shown on the figure.

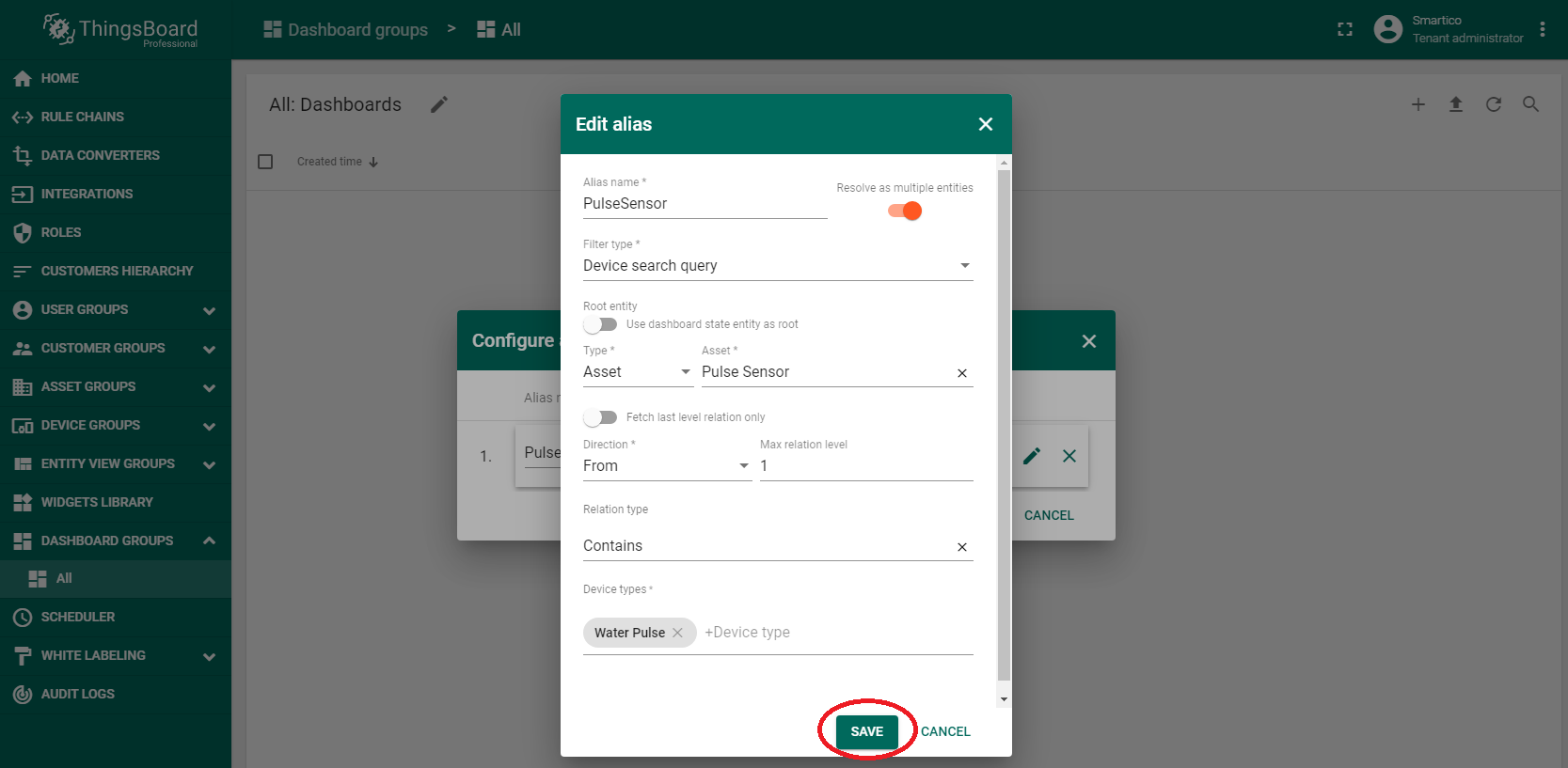


**Step 6. Dashboard import and configuration.**

To display data to users, you need to create a dashboard that can be imported from the following json file (*dashboard\_pulse\_sensor.json*).



When importing a dashboard, it will be necessary to create an alias, as shown in the figure.



If everything was done correctly, in DASHBOARD GROUPS → All you will see the new dashboard **Water meter Pulse Sensor LoRaWAN "Smartico P22-LR"** that was provided at the beginning of the guide.