

Asset Tracker Driver

Rev 3.7.0

17/11/2023

NOTICE
This document contains proprietary and confidential material of Abeeway SAS. This document is provided under and governed by either a license or confidentiality agreement. Any unauthorized reproduction, use, or disclosure of this material, or any part thereof, is strictly prohibited.
The material provided in this document is believed to be accurate and reliable. However, no responsibility is assumed by Abeeway SAS for the use of this material. Abeeway SAS reserves the right to make changes to the material at any time and without notice. This document is intended for information and operational purposes only. No part of this document shall constitute any contractual commitment by Abeeway SAS.

ABEEWAY SAS
www.abeeway.com – contact@abeeway.com

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Hend Affes	Initial version	06/09/2018
1.1	Patrick Beatini	Rework	10/09/2018
1.2	Hend Affes	Update in line with AT 1.7.3	25/02/2019
1.3	PY Gueniffey	Added downlink documentation encoder snippet	12/03/2019
1.4	Mengxuan Zhao	Update in line with AT 1.8.0	05/04/2019
1.5	Mengxuan Zhao	Dockerhub version	22/05/2019
1.6	PY Gueniffey	Refactoring	22/05/2019
1.7	Mengxuan Zhao	Docker v1.2	07/06/2019
1.8	Mengxuan Zhao	-Aligned with User Guide -Iot Driver (Docker) v1.4 -Add Decoding Downlink -Optimization	26/08/2019
1.9	Mengxuan Zhao	Update to AT1.9	27/02/2020
2.0	PY Gueniffey	Fixed version number	16/03/2020
2.1	PY Gueniffey	Update for AT 2.0	04/06/2020
2.1.1	Rohit Gupta	Add a section to use driver hosted in SaaS	22/06/2020
2.2	PY Gueniffey	Update for AT 2.1	08/06/2020
2.2.2	PY Gueniffey	Added new proximity payloads	10/07/2020
2.2.3	PY Gueniffey	Latest AT 2.1 features	10/08/2020
2.2.4	PY Gueniffey	Missing AT 2.1 features	11/08/2020
2.2.5	PY Gueniffey	New AT 2.1 features	18/09/2020
2.2.6	Mengxuan Zhao	Latest update for AT 2.1	20/10/2020
3.1.0	Hend Affes	Update for AT 2.2	24/03/2021
3.1.1	Hend Affes	Latest update for AT 2.2	20/04/2021
3.1.3	Hend Affes	Change minimum values	21/06/2021

3.2.0	Hend Affes	New position payload (AT 2.2.1)	15/07/2021
3.3.0	PY Gueniffey	Added points	25/01/2022
3.4.0	Mengxuan Zhao	Update for AT 2.3	11/03/2022
3.5.0	PY Gueniffey	Update for AT 2.4	17/10/2022
3.5.2	PY Gueniffey	Renamed motionDurationPercentage	23/12/2022
3.5.3	PY Gueniffey	Update for AT 2.4.1 (change range of ble_beacon_timeout	27/01/2023
3.6.0	PY Gueniffey	Update for AT 2.5.0	07/07/2023
3.7.0	Mengxuan Zhao	Update for AT 2.6.0	17/11/2023

Table of Contents

1. Introduction 8
2. Accessing the Abeeway driver using ThingPark X Location Engine (SaaS) 9
2.1. Create a free account 9
2.2. Generate a token 9
2.3. Using the Abeeway Driver hosted in ThingPark X Location Engine (SaaS) 11
2.4. Retrieving the version of driver installed on the SaaS platform 13
3. Standalone Installation of Abeeway driver using docker 14
3.1. Prerequisite 14
3.2. Removing the old docker images 14
3.3. Pull and run the driver with docker 14
3.4. Usage of the Abeeway Driver 15
3.4.1. Retrieving the version information of the Abeeway driver 16
3.4.2. Client code 17
4. Standalone Installation of Abeeway driver using Node.js and npm 19
4.1. Prerequisite 19
4.2. Decoding uplink 19
4.3. Encoding downlink 20
4.4. Decoding downlink 21
5. Decoding an uplink 23
5.1. Description 23
5.1.1. Device configuration object 32
5.1.2. Custom transmit strategy object 46
5.1.3. Measured temperature object 47
5.1.4. Data scan collection object 48
5.1.5. Proximity notification object 49
5.1.6. Proximity daily report object 50
5.1.7. Proximity white listing object 51
5.1.8. Proximity daily response object 51
5.1.9. Angle detection object 52
5.1.10. Geofencing notification object 52
5.1.11. GPS previous object 53

5.1.12. Health Status object 53
5.1.13. SMS object 54
5.2. Examples 54
5.2.1. Position message 54
5.2.2. Heartbeat message 58
5.2.3. Activity Status message 61
5.2.4. Configuration message 64
5.2.5. Shock Detection message 68
5.2.6. Shutdown message 71
5.2.7. Event message 74
5.2.7.1. BLE Connected Event 74
5.2.7.2. Temperature Alert Event 77
5.2.7.3. Angle detection Event 81
5.2.7.4. Geofencing notification Event 85
5.2.8. Data scan collection message 88
5.2.9. Proximity notification message 92
5.2.10. Proximity daily report message 95
5.2.11. Proximity white listing message 99
5.2.12. Proximity daily response message 102
5.2.13. SMS message 105
5.2.14. Debug message 107
6. Decoding multiple payloads 112
7. Encoding a downlink 119
7.1. Overview 119
7.2. Request a position 119
7.2.1. Example 120
7.3. Operational mode change 121
7.3.1. Example 121
7.4. SOS start 122
7.4.1. Example 122
7.5. SOS stop 124
7.5.1. Example 124
7.6. Request temperature status 125
7.6.1. Example 125

7.7. Proximity messages 126
7.7.1. Request proximity record status 126
7.7.1.1. Example 126
7.7.2. Set proximity record white list status 128
7.7.2.1. Example 128
7.7.3. Get proximity daily information 129
7.7.3.1. Example 130
7.7.4. Clear proximity daily information 131
7.7.4.1. Example 131
7.8. Angle detection control 133
7.8.1. Example 133
7.9. Request status 134
7.9.1. Example 134
7.10. Receive SMS 137
7.10.1. Example 137
7.11. Debug command 138
7.11.1. Example 139
7.11.2. Reset Device with reset action value specified 140
7.11.2.1. Example 140
7.11.3. Write New TX Power Index 142
7.11.3.1. Example 142
7.11.4. Set buzzer melody and duration 143
7.11.4.1. Example 143
7.11.5. Configure Startup modes 145
7.11.5.1. Example 145
7.11.6. Start and Stop BLE advertisement 146
7.11.6.1. Example 146
7.12. Parameter read 148
7.12.1. Examples 155
7.13. Parameter modification 157
7.13.1. Examples 157
7.13.1.1. Modify the GPS parameters: 157
7.13.1.2. Modify the BLE parameters: 158
7.13.1.3. Modify the Operational mode Parameters: 158

7.13.1.4. Modify the LORA parameters: 159
7.13.1.5. Modify the side operational mode parameters: 160
7.13.1.6. Modify Misc Parameters 161
7.13.1.7. Modify accelerometer parameters 162
7.13.1.8. Modify temperature parameters 163
7.13.1.9. Modify special parameters 164
7.13.1.10. Modify Custom Transmission Strat parameters 164
7.13.1.11. Modify scan collection parameters 166
7.13.1.12. Modify BLE Position Filtering parameters 167
7.13.1.13. Modify proximity detection parameters 168
7.13.1.14. Modify angle detection parameters 170
7.13.1.15. Modify Quuppa beaconing parameters 172
7.13.1.16. Modify geofencing parameters 173
7.13.1.17. Modify button mapping 174
7.13.1.18. Modify motion GPS parameters 175
8. Decoding a downlink 175
8.1. Example 176
9. References 177

1. Introduction

This document describes the Abeeway Firmware driver (Encoder/Decoder) for the Abeeway Asset Tracker 2 firmware version v2.4. It explains how to decode the uplinks and encode the downlinks messages.

For a full description of the Abeeway trackers, it is advised to read the companion document: Abeeway Reference Guide [1].

The document provides the usage and installation information of the driver. It also provides some example code. To separate the examples from the rest of the documents, they are always enclosed by the following tags:

---------- Snippet begin ----------
---------- Snippet end ----------

The driver is provided as a docker image. Once the container is running, you can access the driver REST API using any REST API tool or client, for example:

There are three ways to use the driver:

1. Driver hosted in ThingPark X Location Engine (Saas): In this scenario, the client application calls the driver hosted inside ThingPark X Location Engine SaaS
2. Driver hosted locally as a docker service: In this scenario, the client application calls the driver hosted inside docker image
3. Link the Javascript driver to your application. The instructions are available here, https://github.com/actility/device-catalog/tree/main/vendors/abeeway/drivers/asset-tracker

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

2. Accessing the Abeeway driver using ThingPark X Location Engine (SaaS)

The easiest way to use Abeeway driver is to use ThingPark X Location Engine hosted in SaaS. The section below describes the steps needed to access the driver hosted in ThingPark X Location Engine.

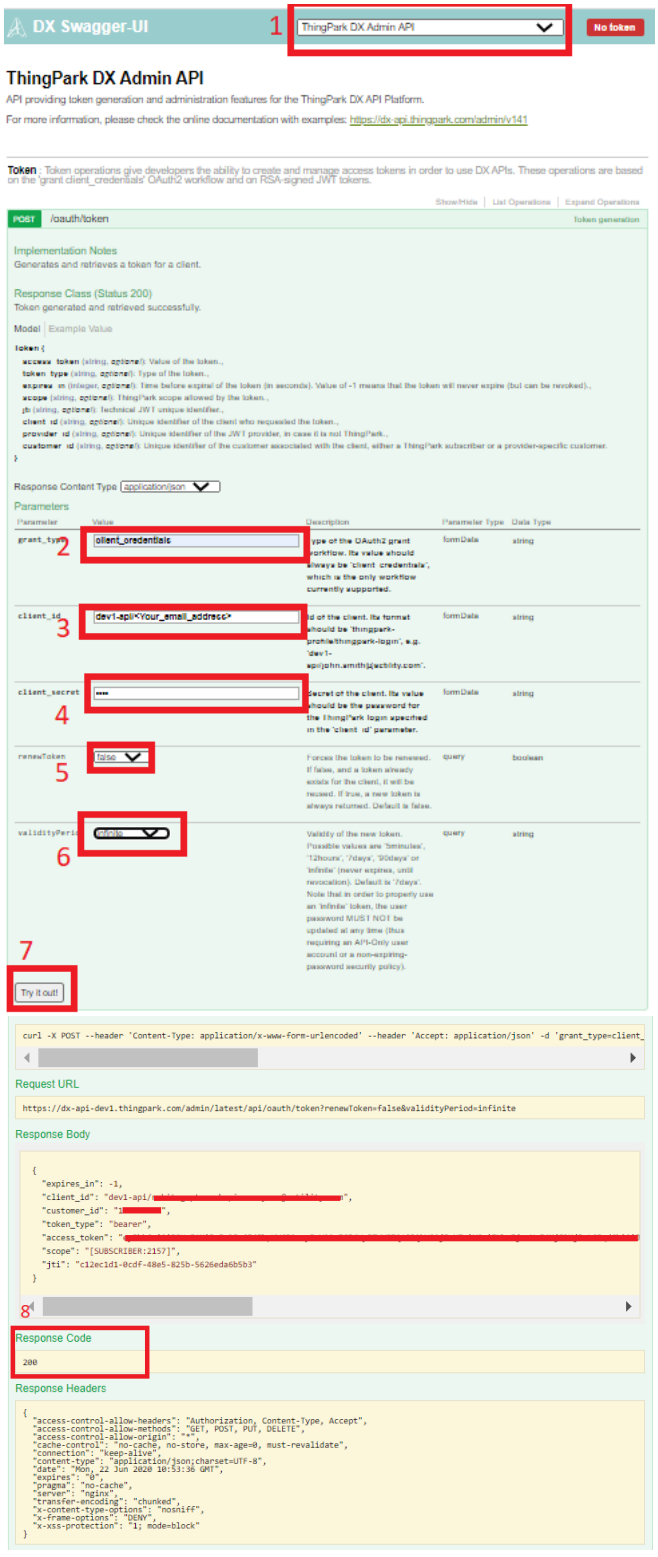
2.1. Create a free account

In this step, the user needs to create free account on ThingPark community portal, https://community.thingpark.org/

2.2. Generate a token

In this step, the user needs to generate a token that can be used with the DX Location API. Here is the URL of DX Admin API page on ThingPark community portal, https://dx-api.thingpark.io/admin/latest/swagger-ui/index.html?shortUrl=tpdx-admin-api-contract.json#!/Token/post_oauth_token. Here are the steps (see figure below) that the user needs to make to generate the token using DX Admin API. The user needs to set *grant_type* to *client_credentials* and enter his login/password in *client_id* and client_secret field respectively in the form below.

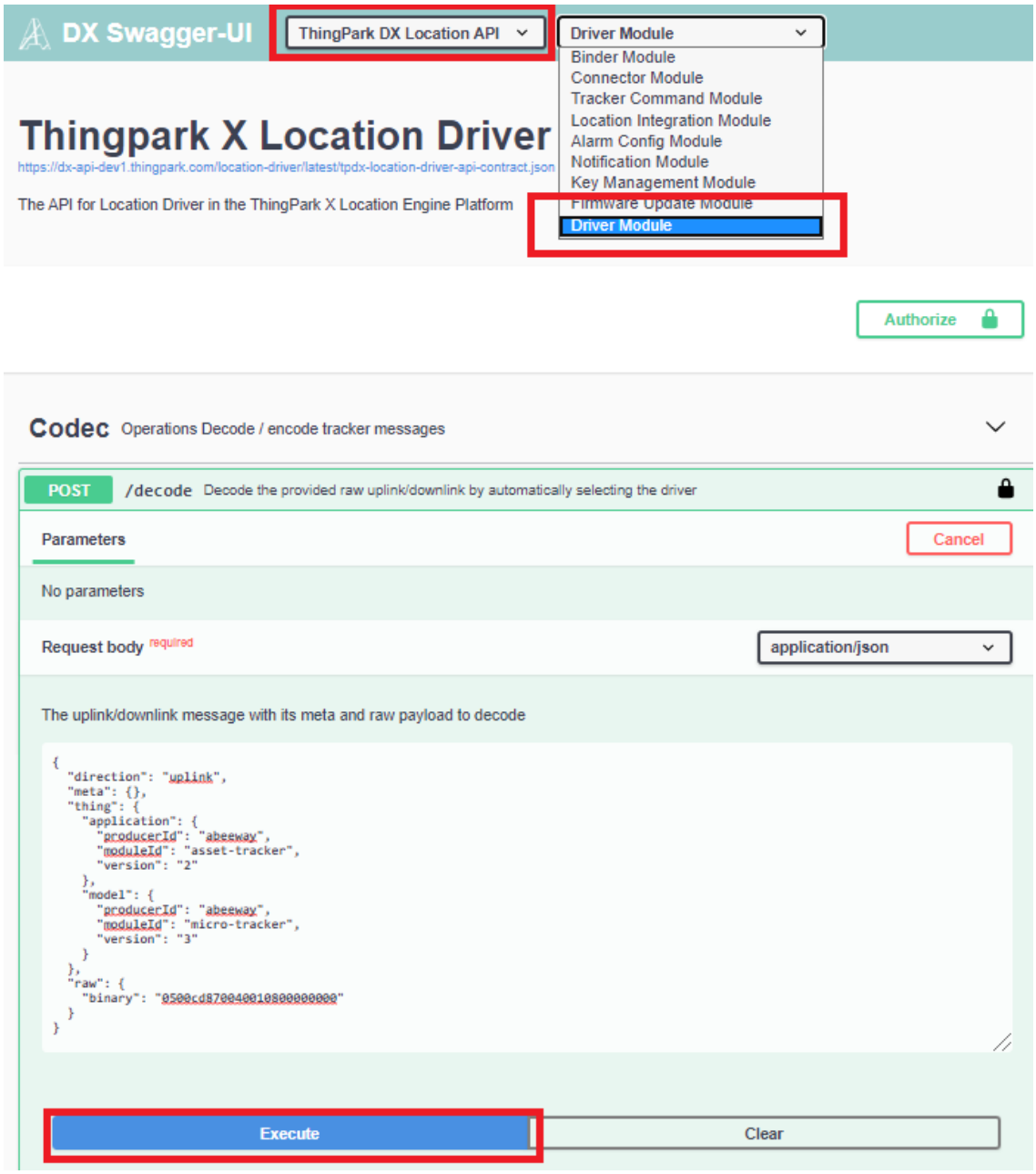
Once the command is executed, you should receive *200* as the response and the token is set on the swagger UI.



Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

2.3. Using the Abeeway Driver hosted in ThingPark X Location Engine (SaaS)

Once the token is set on swagger, then you can can switch to the driver module of ThingPark Location API as shown below which you can use to encode/decode Abeeway tracker payloads. The driver module within DX location API can be accessed from here, https://dx-api.thingpark.io/location-driver/latest/swagger-ui/index.html?shortUrl=tpdx-location-driver-api-contract.json



Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

Responses

Curl

```
curl -X POST "https://dx-api-dev1.thingpark.com/location-driver/latest/api/decode" -H "accept: application/json" -H
"Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzY29wZSI6WyJTVUJTQ1JJQkVSOjIxNTciXSwiZXhwIjozNzI4MjE3MjIwLCJqdGkiOiJjMTJlYzFkMS0wY2R
mLTQ4ZTUtODI1Yi01NjI2ZWRhNmI1YjMiLCJjbGllbnRfaWQiOiJkZXYxLWFwaS9yb2hpdC5ndXB0YStzdWItcGllcnJlLXl2ZXNAYWN0aWxpdHkuY29tIn0.1jv
RIQKalysItQUirzypz5X_jjWOBxrY8hKAKPx0kXvRvNfIChsnJrAsU_w4t644qg37ZfALEaqGaQJFuQjpWA" -H "Content-Type: application/json" -d
"{\"direction\":\"uplink\",\"meta\":{},\"thing\":{\"application\":{\"producerId\":\"abeeway\",\"moduleId\":\"asset-
tracker\",\"version\":\"2\"},\"model\":{\"producerId\":\"abeeway\",\"moduleId\":\"micro-
```

Request URL

```
https://dx-api-dev1.thingpark.com/location-driver/latest/api/decode
```

Server response

Code	Details
200	Response body


```
{
  "direction": "uplink",
  "meta": {},
  "raw": {
    "binary": "0500cd870040010800000000"
  },
  "message": {
    "messageType": "HEARTBEAT",
    "trackingMode": "STAND_BY",
    "batteryVoltage": 3.93,
    "ackToken": 0,
    "firmwareVersion": "1.8.0",
    "bleFirmwareVersion": "0.0.0",
    "resetCause": 40,
    "periodicPosition": false,
    "temperatureMeasure": 24.3,
    "sosFlag": 0,
    "appState": 0,
    "dynamicMotionState": "STATIC",
    "onDemand": false,
    "payload": "0500cd870040010800000000"
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
```


Download

Response headers

```
access-control-allow-headers: Authorization, Content-Type, Accept
access-control-allow-methods: GET, POST, PUT, DELETE
access-control-allow-origin: *
connection: keep-alive
content-type: application/json;charset=utf-8
date: Mon, 22 Jun 2020 12:45:30 GMT
server: nginx
transfer-encoding: chunked
x-application-context: application:21101
```

Responses

Code	Description	Links
200	The uplink/downlink message with its meta, raw payload and decoded message	*No links*

2.4. Retrieving the version of driver installed on the SaaS platform

Here is an example of the command to retrieve the Abeeway Asset Tracker driver that is installed on SaaS https://community.thingpark.io platform. The platform URL must be replaced with the platform currently in use.

---------- Snippet begin ----------

```
curl -s "https://community.thingpark.io/iot-flow/v1/drivers" -H "accept:
application/json" | jq '.[] | select(.id=="abeeway:asset-tracker:3") |
.version'
```

---------- Snippet end ----------

The result shows you the driver version that is installed on the SaaS platform.

---------- Snippet begin ----------

```
3.5.0
```

---------- Snippet end ----------

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

3. Standalone Installation of Abeeway driver using docker

3.1. Prerequisite

To use the driver, you need the docker engine installed on your environment. For more information of the docker engine on your environment, please refer to https://hub.docker.com/search/?type=edition&offering=community

3.2. Removing the old docker images

It is mandatory to remove the older docker images if they already exist. The docker image can be removed as follows:

```
docker images -a
```

```
docker rmi <Image_Name>
```

3.3. Pull and run the driver with docker

Open a command line tool, start by pulling the docker image:

```
docker pull actility/iot-flow-drivers
```

The port 8095 must be exposed from the docker in order to access the API.

---------- Snippet begin ----------

```
docker run -d -p 8095:8095 actility/iot-flow-drivers:latest
```

---------- Snippet end ----------

Note: If the driver is invoked for the first time on localhost, the docker instance needs the Internet to fetch the drivers. There are some instances when the DNS resolution has error when running docker image. If that is the case, the docker run command will fail after few minutes and there will no image when the following command is executed.

---------- Snippet begin ----------

```
sudo docker ps
```

---------- Snippet end ----------

In order to check the reason for failure, you can run the following command below, where <image_name> is present in the execution of the above command:

---------- Snippet begin ----------

```
sudo docker logs <image_name>
```

---------- Snippet end ----------

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

If you see an error in the docker logs that looks something like below, the execute the docker run command with additional parameters described below:

---------- Snippet begin ----------

```
INFO: Downloading drivers...
mirror: drivers.thingpark.com: Name or service not known
ERROR: Mirroring of drivers repository failed. This docker image needs
access to https://drivers.thingpark.com in order to download the drivers,
if you suspect a DNS problem try adding "--add-
host=drivers.thingpark.com:$(nslookup drivers.thingpark.com | awk
'/^Address: / { print $2 }' | head -1)" to the docker run command (e.g.
docker run -d -p 8095:8095 --add-host=drivers.thingpark.com:$(nslookup
drivers.thingpark.com | awk '/^Address: / { print $2 }' | head -1)
actility/iot-flow-drivers)
```

---------- Snippet end ----------

Here is an example to run docker image with additional parameters to resolve DNS issue.

---------- Snippet begin ----------

```
docker run -d -p 8095:8095 --add-host=drivers.thingpark.com:$(nslookup
drivers.thingpark.com | awk '/^Address: / { print $2 }' | head -1)
actility/iot-flow-drivers
```

---------- Snippet end ----------

Note: You will have to wait for few minutes for the drivers to be downloaded to the image first time you run the container. If there are some other issues, please contact Abeeway support and provide us complete logs of docker image.

3.4. Usage of the Abeeway Driver

The driver is accessible at

```
http://localhost:8095/v1/drivers/abeeway:asset-tracker:3
```

If you run the above URL in browser, you should get a response like below to ensure everything is working correctly. If it is not the case, please contact Abeeway support with your error message.


```
{
   "id":"abeeway:asset-tracker:3",
   "name":"asset tracker js driver",
   "type":"thingpark-x-js",
   "moduleId":"asset-tracker",
   "producerId":"abeeway",
   "source":"system",
   "version":"3.7.0",
```

```
    "application":{
       "producerId":"abeeway",
       "moduleId":"asset-tracker",
       "version":"2"
    },
    "private":true,
    "examples": [...]
}
```

If the information of the driver is accessible using the above endpoint, the driver is ready to use.

- To encode a downlink message, send HTTP POST request to

```
https://localhost:8095/v1/encode
```

- To decode an uplink message, send HTTP POST request to

```
http://localhost:8095/v1/decode
```

Request examples are given in the following sections.
A UI is available to make your requests using the API: http://localhost:8095/v1/swagger-ui/

3.4.1. Retrieving the version information of the Abeeway driver

The version of the Abeeway driver that is installed locally can be retrieved as follows:
---------- Snippet begin ----------

```
curl -X GET "http://localhost:8095/v1/drivers" -H  "accept:
application/json" | python3 -m json.tool
```

---------- Snippet end ----------

The following is an example snippet of the response to the command above. The docker image hosts several drivers. The version of Abeeway Asset Tracker driver below is 3.5.0:
---------- Snippet begin ----------

```
    {
        "id": "abeeway:asset-tracker:3",
        "producerId": "abeeway",
        "moduleId": "asset-tracker",
        "version": "3.5.0",
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
….
]
```

---------- Snippet end ----------

3.4.2. Client code

All the examples use `cURL` to reach the driver REST service, however, you are free to use your favorite client, or your favorite programming language to establish your request. NodeJS and Python request examples can be found below.

- The following is an example of the Downlink encoding request in Python (Python3):

---------- Snippet begin ----------

```
import requests

url = "http://localhost:8095/v1/encode"

payload='{"direction":"downlink","meta":{},"thing":{"application":{"producerId":"abeeway","moduleId":"asset-tracker","version":"2"}},"message":{"downMessageType":"SET_PARAM","ackToken":10,"setParameters":{"energyStatusPeriod":604800,"configFlags":{"framePendingMechanism":true,"buttonPressToTurnOFF":false,"doubleClickIsSosModeOrAlert":false,"wifiPayloadWithNoCypher":true,"bleAdvertisingAtStart":false,"selectWifiScanOrGeolocStartMessage":false,"downlinkSetParameterConfirmation":true,"ledBlinkWithGpsFix":false,"startMotionEventUplink":true,"endMotionEventUplink":false,"otaJoinWhenLeavingModeOff":false,"rejectAsymmetricPairing":false,"enableLongWifiPayload":false,"collectionLongReport":true,"autoStart":false,"automaticDatarateSelection":false,"forbidModeOff":false,"sosModeSound":false,"automaticDatarateSelection":false,"extendedPosition":false,"bleCli":false,"blePasskey":false,"wifiDatabase":false,"activeWifiScan":false}}}}'
headers = {'Content-Type': "application/json"}

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

---------- Snippet end ----------

- The following is an example of the Uplink decoding request in NodeJS:

---------- Snippet begin ----------

```
var request = require("request");
```

```
var options = { method: 'POST',
  url: 'http://localhost:8095/v1/decode',
  headers:
   { 'cache-control': 'no-cache',
     'Content-Type': 'application/json' },
  body:
   { direction: 'uplink',
     meta: { lorawan:
             { fPort: 18 } ,
     "driverId": "abeeway:asset-tracker:3"
   },
     thing:
      { application:
         { producerId: 'abeeway',
           moduleId: 'asset-tracker',
           version: '2' }
      },
     raw:
      { binary: '05604b898001010703' }
  },
  json: true
};

request(options, function (error, response, body) {
  if (error) throw new Error(error);

  console.log(body);
});
```

---------- Snippet end ----------

with `request` module installed using `npm install request`

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

4. Standalone Installation of Abeeway driver using Node.js and npm

4.1. Prerequisite

To use the driver using Node.js/npm, you will need to install Node.js and npm by following instructions here. Once installed, perform the following steps:

---------- Snippet begin ----------

```
mkdir my-abeeway-test-project
cd my-abeeway-test-project
npm init --yes
npm install abeeway-asset-tracker-driver
```

---------- Snippet end ----------

4.2. Decoding uplink

Here is an example code to decode an uplink which can be put in file index.js.

---------- Snippet begin ----------

```
// Import the driver
var driver = require("abeeway-asset-tracker-driver");

// Helper function: converts a hex string to a hex array
function parseHexString(str){
    let result = [];
    while (str.length >= 2) {
        result.push(parseInt(str.substring(0, 2), 16));
        str = str.substring(2, str.length);
    }
    return result;
}

// EXAMPLE CODE FOR TESTING
let uplinkPayloadString = "032c5f8c0a00000000000045c800000000039b0000000000002ae";
let input = {
    bytes: parseHexString(uplinkPayloadString),
    fPort: 18
}
let result = driver.decodeUplink(input)
console.log(result.data)
```

---------- Snippet end ----------

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

Here is the output when executing ***node index.js***,

---------- Snippet begin ----------

```
% node index.js

AbeewayUplinkPayload {
  messageType: 'EXTENDED_POSITION_MESSAGE',
  age: 0,
  trackingMode: 'MOTION_TRACKING',
  batteryLevel: 100,
  batteryStatus: 'OPERATING',
  ackToken: 0,
  rawPositionType: 'BLE_BEACON_SCAN',
  periodicPosition: false,
  temperatureMeasure: 28.3,
  sosFlag: 1,
  appState: 1,
  dynamicMotionState: 'MOVING',
  onDemand: false,
  payload: '0e3c648f0700006077719df3b4c76077719d58e0bb6077719d5c8dae6077719d5cbaa6',
  deviceConfiguration: { mode: 'MOTION_TRACKING' },
  bleBssids: [
    BssidInfo { bssid: '60:77:71:9d:f3:b4', rssi: -57 },
    BssidInfo { bssid: '60:77:71:9d:58:e0', rssi: -69 },
    BssidInfo { bssid: '60:77:71:9d:5c:8d', rssi: -82 },
    BssidInfo { bssid: '60:77:71:9d:5c:ba', rssi: -90 }
  ]
}
```

---------- Snippet end ----------

4.3. Encoding downlink

Here is an example code to encode a downlink which can be put in file index.js.

---------- Snippet begin ----------

```
// Import the driver
var driver = require("abeeway-asset-tracker-driver");

let input = {
    "downMessageType": "DEBUG_COMMAND",
    "ackToken": 2,
    "debugCommandType": "RESET"
```

```
}

var result = driver.encodeDownlink(input).bytes
console.log(result)
```

---------- Snippet end ----------

Here is the output when executing ***node index.js***,

---------- Snippet begin ----------

```
% node index.js
[ 255, 2, 1 ]
```

---------- Snippet end ----------

The downlink is encoded in hexadecimal, where:

- byte 0 (LSB) = 1,
- byte 1 = 2,
- byte 2 (MSB) = 255

4.4. Decoding downlink

Here is an example code to decode a downlink which can be put in file index.js.

---------- Snippet begin ----------

```
// Import the driver
var driver = require("abeeway-asset-tracker-driver");

// Helper function: converts a hex string to a hex array
function parseHexString(str){
    let result = [];
    while (str.length >= 2) {
        result.push(parseInt(str.substring(0, 2), 16));
        str = str.substring(2, str.length);
    }
    return result;
}

// EXAMPLE CODE FOR TESTING
let uplinkPayloadString = "ff0102";
let input = {
    bytes: parseHexString(uplinkPayloadString),
    fPort: 18
}
let result = driver.decodeDownlink(input)
```

```
console.log(result)
```

---------- Snippet end ----------

Here is the output when executing ***node index.js***,

---------- Snippet begin ----------

```
% node index.js
{
  data: AbeewayDownlinkPayload {
    downMessageType: 'DEBUG_COMMAND',
    ackToken: 1,
    modeValue: undefined,
    debugCommandType: 'RESET_BLE_PAIRING',
    listParameterID: undefined,
    listParameterIDNames: undefined,
    statusType: undefined,
    setParameters: undefined,
    resetAction: undefined,
    optionalCommand: undefined,
    txPowerIndex: undefined,
    melodyId: undefined,
    buzzerDuration: undefined,
    proximityMessage: undefined,
    angleDetectionControl: undefined,
    bleAdvertisementDuration: undefined,
    startupModes: undefined,
    sms: undefined,
    payload: 'ff0102'
  },
  errors: [],
  warnings: []
}
```

---------- Snippet end ----------

5. Decoding an uplink

5.1. Description

The input of the decoder is a LoRa uplink payload coming from an Abeeway Tracker using Abeeway Asset Tracker firmware.
The LoRa port number[1] used for transmission must be passed in the “meta” object fed to the driver.
The output of the decoder is a string in the well-known JSON format.
For example, a LoRa heartbeat message contains only the following fields:

- DynamicMotionState
- On-demand
- Battery level
- Temperature
- Ack token
- Reset cause
- Firmware version

[1]Note:

1. Abeeway firmware 2.0+ use LoRaWAN port = 18 for uplinks
2. Abeeway firmware 1.9 and below use LoRaWAN port = 17 for sending uplinks
3. The fport input to the driver when decoding uplinks should be the LoRaWAN port over which the message is received.

The input parameters are described in the table below.

Parameter	**Type**	**Description**	**Supported since FW version**	**Supported HW version**
gpsLatitude	Number	Latitude of the position (expressed in degrees).	1.7	Micro, Industrial, Compact, Badge
gpsLongitude	Number	Longitude of the position (expressed in degrees).	1.7	Micro, Industrial, Compact, Badge
horizontalAccuracy	Number	Estimated Horizontal Position Error, a measure of the error in a GPS position in the horizontal plane It is expressed in meters	1.7	Micro, Industrial, Compact, Badge
messageType	String	Indicates the uplink type:	1.7 Shock	Micro, Industrial, Compact,

		• POSITION_MESSAGE • HEARTBEAT • ENERGY_STATUS • SHUTDOWN • FRAME_PENDING • DEBUG • ACTIVITY_STATUS • CONFIGURATION • SHOCK_DETECTION • EVENT • DATA_SCAN_COLLECTION • PROXIMITY_DETECTION • EXTENDED_POSITION_MESSAGE	detection feature supported only since 1.8 Data scan collection feature supported since 2.0 Proximity detection feature supported since 2.1 Extended position message feature supported since 2.2.1	Badge
age	Number	Age in seconds of message position given by the GPS, WIFI, BLE scan, GPS timeout, BLE failure, WIFI timeout and WIFI failure.	1.7 The age is added to GPS timeout, BLE failure, WIFI timeout and WIFI failure. since 2.21	Micro, Industrial, Compact, Badge
trackingMode	String	Tracker operating mode. Values can be: • STAND_BY • OFF • MOTION_TRACKING • PERMANENT_TRACKING • MOTION_START_END_TRACKING • ACTIVITY_TRACKING	1.7	Micro, Industrial, Compact, Badge

batteryVoltage	Number	The battery voltage. Expressed in Volts. If the value is equal to 0, it means that the battery is charging	1.7, <2.0	Micro, Industrial, Compact, Badge
batteryLevel	Number	The battery level. Expressed in %. If the value is equal to 0, it means that the battery is charging	2.0	Micro, Industrial, Compact, Badge
batteryStatus	String	Tracker battery status. Values can be: • CHARGING • OPERATING • UNKNOWN	2.0	Micro, Industrial, Compact, Badge
ackToken	Number	Acknowledgement token is used to inform the server of the reception of LoRa downlink messages.	1.7	Micro, Industrial, Compact, Badge
firmwareVersion	String	Firmware Version	1.7	Micro, Industrial, Compact, Badge
bleFirmwareVersion	String	BLE Firmware Version	1.8	Micro, Industrial, Compact, Badge
resetCause	String	Reset cause	1.7	Micro, Industrial, Compact, Badge
rawPositionType	String	Fix type describes what kind of data is in the uplink. Values can be: • GPS • GPS_TIMEOUT • ENCRYPTED_WIFI_BSSIDS • WIFI_TIMEOUT • WIFI_FAILURE • XGPS_DATA • XGPS_DATA_WITH_GPS_SW_TIME • BLE_BEACON_SCAN • BLE_BEACON_FAILURE • WIFI_BSSIDS_WITH_NO_CYPHER • BLE_BEACON_SCAN_SHORT_ID • BLE_BEACON_SCAN_LONG_ID	1.7, BLE Beacon ID (short and long) supported since 2.1	Micro, Industrial, Compact, Badge

periodicPosition	Boolean	True if this is a periodic position message.	1.7	Micro, Industrial, Compact, Badge
gpsOnRuntime	Number	The total time in seconds spent by the GPS in state ON since boot. It is transmitted in energy status message	1.7	Micro, Industrial, Compact, Badge
gpsStandbyRuntime	Number	The total time in seconds spent by the GPS in state STANDBY since boot. It is transmitted in energy status message	1.7	Micro, Industrial, Compact, Badge
wifiScanCount	Number	The total counter of WIFI scans since boot. No unit (counter). It is transmitted in energy status message	1.7	Micro, Industrial, Compact, Badge
timeoutCause	String	The GPS timeout cause. Values can be: • USER_TIMEOUT: The GPS was not able to compute a fix before the gps timeout period • T0_TIMEOUT: The GPS geolocation is aborted if there are not enough satellites are in view at the end of the duration indicated in the parameter gps_t0_timeout • T1_TIMEOUT: The GPS geolocation is aborted if there is no GPS fix at the end of the duration indicated in the parameter gps_fix_timeout	1.7 T0_TIMEOUT, T1_TIMEOUT Supported since 2.2	Micro, Industrial, Compact, Badge
bestSatellitesCOverN	Array	when a "GPS Timeout" occurs, the rest of the LoRa message consists of the type of timeout that preempted the scan and the 4 best satellites C/N indicators, the bestSatellitesIndicator is an array.	1.7	Micro, Industrial, Compact, Badge
temperatureMeasure	Number	Board temperature in °C	1.7	Micro, Industrial, Compact, Badge
miscDataTag	String	Values can be: • ACTIVITY_COUNTER • DEVICE_CONFIGURATION • SHOCK_DETECTION • PERIODIC_ACTIVITY	1.7 SHOCK_DETECTION and PERIODIC_ACTIVITY only since 1.8	Micro, Industrial, Compact, Badge

sosFlag	Number	Set if the user alert/SOS has been entered.	1.7	Micro, Industrial, Compact, Badge
appState	Number	Tracking state. 1: tracking. 0: idle	1.7	Micro, Industrial, Compact, Badge
dynamicMotionState	String	MOVING or STATIC	1.7	Micro, Industrial, Compact, Badge
onDemand	Boolean	True if this is an on-demand message.	1.7	Micro, Industrial, Compact, Badge
batteryVoltageMeasures	Array	Voltage measures reported under WIFI scan failures.	1.7	Micro, Industrial, Compact, Badge
errorCode	Number	Error code.	1.7	Micro, Industrial, Compact, Badge
debugErrorCode	Number	Unique error code corresponding to the crash	1.7	Micro, Industrial, Compact, Badge
shutdownCause	String	Shutdown cause. Values can be • USER-ACTION • LOW-BATTERY • DOWNLINK-REQUEST • BLE_REQUEST • BLE_CONNECTED	1.7	Micro, Industrial, Compact, Badge
currentAckTokenValue	Number	Current Ack Token value	1.7	Micro, Industrial, Compact, Badge
payload	string	The uplink payload	1.7	Micro, Industrial, Compact, Badge

debugCrashInfo	String	Crash information in ASCII	1.7	Micro, Industrial, Compact, Badge
activityCount	Number	Every motion detection, the activity counter is increased	1.7	Micro, Industrial, Compact, Badge
deviceConfiguration	Object	Device configuration object. See Device configuration object	1.7	Micro, Industrial, Compact, Badge
wifiBssids	Array of objects	An array of (BSSID, RSSI) objects. Field populated for successful WIFI scan.	1.7	Micro, Industrial, Compact, Badge
bleBssids	Array of objects	An array of (BSSID, RSSI) objects. Field populated for successful BLE scan.	1.7	Micro, Industrial, Compact, Badge
bleBeaconIds	Array of objects	An array of (Beacon ID, RSSI) objects. Field populated for successful BLE scan.	2.1	Micro, Industrial, Compact, Badge
bleBeaconFailure	String	The cause of BLE beacon failure. Values can be: • BLE_NOT_RESPONDING • INTERNAL_ERROR • SHARED_ANTENNA_NOT_AVAILABLE • SCAN_ALREADY_ON_GOING • NO_BEACON_DETECTED • HARDWARE_INCOMPATIBILITY	1.7 (changed in AT2 2.3. If the driver is used with a previous version, this value will not be correctly decoded)	Micro, Industrial, Compact, Badge

eventType	String	Values can be: • START_GEOLOC • MOTION_START • MOTION_END • BLE_CONNECTED • BLE_DISCONNECTED • TEMPERATURE_ALERT • BLE_BOND_DELETED • SOS_MODE_START • SOS_MODE_END • ANGLE_DETECTION • GEOFENCING	1.8, TEMPERATURE_ALERT and BLE_BOND_DELETED since 1.9 SOS_MODE_START, SOS_MODE_END since 2.1 ANGLE_DETECTION and GEOFENCING Since 2.2	Micro, Industrial, Compact, Badge
txPowerIndex	Number	TX Power index used by the device	1.7	Micro, Industrial, Compact, Badge
debugCommandTag	String	Values can be: • DEBUG_CRASH_INFORMATION • TX_POWER_INDEX_VALUE • UPLINK_LENGTH_ERR • GENERIC_ERROR • SPECIFIC_FIRMWARE_ PARAMETERS	1.7, UPLINK_LENGTH_ERR since 1.9 SPECIFIC_FIRMWARE_PARAMETERS Since 2.2	Micro, Industrial, Compact, Badge
nbOfshock	Number	Number of shocks above configured condthresholds	1.8	Micro, Industrial, Compact, Badge
accelerometerShockData	Array	the biggest shock detected during the period	1.8	Micro, Industrial, Compact, Badge
trackerOrientation	Array	orientation of the tracker in End Motion Event	1.8	Micro, Industrial, Compact, Badge
activityReportingWindow	Array	activity in 6 windows	1.8	Micro, Industrial, Compact, Badge

measuredTemperature	Object	data about the measured temperature in a “temperature alert” event. See Measured temperature object	1.9	Micro, Industrial, Compact, Badge
lengthErrCounter	Number	number of payloads discarded due to Uplink Length Error Is returned	1.9	Micro, Industrial, Compact, Badge
dataScanCollection	Object	Information about the data scan collection information. See Data scan collection object	2.0	Micro, Industrial, Compact, Badge
proximityNotification	Object	Information about proximity notification. See Proximity notification object	2.1	Micro, Industrial, Compact, Badge
proximityDailyReport	Object	Information about proximity daily report. See Proximity daily report object	2.1	Micro, Industrial, Compact, Badge
proximityWhiteListing	Object	Information about proximity white listing. See Proximity white listing object	2.1	Micro, Industrial, Compact, Badge
proximityDailyResponse	Object	Information about proximity daily response. See Proximity daily response object	2.1	Micro, Industrial, Compact, Badge
angleDetection	Object	Information about angle detection reported in “ANGLE_DETECTION” event message. See Angle detection object	2.2	Micro, Industrial, Compact, Badge
geofencingNotification	Object	Information about BLE geofencing notification received in “GEOFENCING” event message. See Geofencing notification object	2.2	Micro, Industrial, Compact, Badge
specificFirmwareParameters	Array	Specific firmware parameters are reported when a “SPECIFIC_FIRMWARE_PARAMETERS_REQUEST” debug downlink is received.	2.2	Micro, Industrial, Compact, Badge

gpsAltitude	Number	Altitude of the gps position (expressed in meters)	2.2.1	Micro, Industrial, Compact, Badge
gpsFixStatus	String	GPS flag indicates the type of the fix GPS. Values can be: • FIX_2D • FIX_3D	2.2.1	Micro, Industrial, Compact, Badge
gpsCourseOverGround	Number	The GPS course over ground expressed in 1/100 of degree.	2.2.1	Micro, Industrial Compact, Badge
gpsSpeedOverGround	Number	The GPS speed over ground expressed in cm/seconds	2.2.1	Micro, Industrial Compact, Badge
gpsPrevious	object	Information about the previous fix reported in gps position message. See gpsPrevious object	2.2.1	Micro, Industrial Compact, Badge
gpsPayloadType	Number	payload type. 0 or 1. Impact on the altitude and EHPE format	2.3	Micro, Industrial Compact, Badge
healthStatus	object	Information about the health status of the device See healthStatus object	2.3	Micro, Industrial Compact, Badge
motionDutyCycle	Number	Motion duty cycle (time percentage when the tracker was in motion between motion start and motion end messages).	2.4	Micro, Industrial Compact, Badge
gaddIndex	Number	Highest GADD index	2.4	Micro, Industrial Compact, Badge

5.1.1. Device configuration object

The device configuration object can contain any of the tracker configuration parameters. They are described in the table below:

Parameter name (Firmware)	**Type**	**Field value**	**Supported since FW version**
geolocSensorProfile (geoloc_sensor)	String	• Acceptable values are: • *WIFI_ONLY*: Use only WIFI scans. • *GPS_ONLY:* Use only GPS. • *XGPS_ONLY*: Not used anymore. • *WIFI_FALLBACK_XGPS*: Not used anymore. • *SELF_GOVERNING_HISTORY*: Not used anymore. • *SELF_GOVERNING_TIMEOUT*: Superseded by mode 9. • *WGPS_ONLY*: WIFI then GPS if WIFI failed in one geolocation cycle. • *WXGPS_ONLY*: WIFI then LP-GPS if WIFI failed in one geolocation cycle. • *WGPS_WXGPS_FAILURE*: Not used anymore. • *WGPS_WXGPS_TIMEOUT*: Alternate WIFI/LP-GPS/GPS technologies. • *BLE*: Only BLE beacon scans. • *BGPS_ONLY*: BLE then GPS if BLE fails in one geolocation cycle. • *BXGPS_ONLY:* BLE then XGPS if BLE fails in one geolocation cycle.	*1.7* *BGPS_ONLY since 1.8* *BXGPS_ONLYSince 2.2*
oneshotGeolocMethod (geoloc_method)	String	Acceptable values are: • *WIFI*: Use only WIFI scans. • *GPS:* Use only GPS. • *XGPS*: Use GPS and LP-GPS. • *WGPS*: WIFI then GPS if WIFI failed in one geolocation cycle. • *WXGPS*: WIFI then LP-GPS if WIFI failed in one geolocation cycle. • *BGPS:* BLE then GPS if BLE fails in one geolocation cycle. • *BLE*: Only BLE beacon scans. • *BXGPS:* BLE then XGPS if BLE fails in one geolocation cycle.	1.7, *BGPS since 1.8* *BXGPS since 2.2*
extAntennaProfile (antenna)	String	Acceptable values are: • *PRINTED:* PCB antenna. • *CERAMIC:* Ceramic antenna. • *DYNAMIC:* Dynamic antenna selection.	1.7, This parameter is deprecated since FW 2.0 and should not be set

transmitStrat (transmit_strat)	String	• Acceptable values are: • *SINGLE_FIXED*: Single TX. Use provisioned data rate. • *SINGLE_RANDOM*: Single TX. Rate in • [SF7..SF12]. • *DOUBLE_FIXED*: First TX in [SF7..SF8], next use • provisioned data rate. • *DOUBLE_RANDOM*: First TX with rate in • [SF7..SF8], next TX with rate in [SF9..SF12]. • *NETWORK_ADR*: The LoRa network controls the • data rate and the number of transmissions. • *CUSTOM_STRATEGY*: Based on transmitStratCustom parameter.	1.7
trackingUlPeriod (ul_period)	Number	Integer value (number of seconds) in the range [60..86400] for EU and [30..86400] for US.	1.7 The range is changed since 2.2 [15..86400]
loralivePeriod (lora_period)	Number	Integer value (number of seconds) in the range [300..86400].	1.7
energyStatusPeriod (pw_stat_period)	Number	Integer value (number of seconds) in the range [300..604800].	1.7, This parameter is deprecated since FW 2.0 and should not be set
periodicPositionInterval (periodic_pos_period)	Number	Integer value (number of seconds) in the range [900..604800].	1.7
motionStartEndNbTx (motion_nb_pos)	Number	Integer value.	1.7
gpsTimeout (gps_timeout)	Number	Integer value (number of seconds) in the range [30..300].	1.7
xgpsTimeout (agps_timeout)	Number	Integer value (number of seconds) in the range [15..250].	1.7 The range is changed since 2.2 [15..250]
gpsEHPE (gps_ehpe)	Number	Integer value (number of meters) in the range [0..100].	1.7
gpsConvergence (gps_convergence)	Number	Integer value (number of seconds) in the range [0..300].	1.7

bleBeaconCount (ble_beacon_cnt)	Number	Integer value in the range [1..4].	1.7
bleBeaconTimeout (ble_beacon_timeout)	Number	Integer values (number of seconds) in the range [1..5].	1.7
bleRssiFilter (ble_rssi_filter)	Number	Integer values (dBm) in the range [-100..-40].	1.8
gpsStandbyTimeout (gps_standby_timeout)	Number	Integer value (number of seconds) in the range [10..7200].	1.7 The range is changed since 2.2 [0..43200] The range is changed again since 2.4 [0.. 2147483647]
confirmedUplink (confirmed_ul_bitmap, confirmed_ul_retry)	Object	The device can be configured to request LoRa confirmation for a collection of uplink message types. The parameter confirmedUplink is used to select the message types that requires a confirmation. Only message types in the range 0x00 to 0x0F can be selected. confirmedUlRetry gives the number of retransmissions that the tracker should do in the case where the LoRaWAN™ confirmation is not received. A value 0 means that the uplink will request the LoRaWAN™ confirmation but will not retry in case of a failure. Example Confirmed uplink of message types 0x0 and 0x3: bitmap = $2^0 + 2^3 = 1 + 8 = 9$. { "confirmedUlBitmap": 9, "confirmedUlRetry": 2, "framePending": true, "position": true, "energyStatus": false, "heartbeat": false, "activityStatus": false, "configuration": false, "shockDetection": false, "shutdown": false, "event": false, "scanCollection ": false,	1.7, 2.0 (scanCollection), 2.2.1 (extendedPosition), 2.5.0 (sms)

		"extendedPosition": false, "sms": false, "debug": false }	
configFlags (config_flags)	Object	Configuration flags: bit 0: Frame pending mechanism bit 1: Activate long button press to switch to off mode bit 2: Double short press configuration: Alert vs SOS bit 3: Send a configuration uplink message in response to a configuration modification downlink. Used to confirm the action. bit 4: WIFI payload with Cypher or without Cypher bit 5: Activate BLE communication interface bit 6: First WIFI scan when geolocation starts. If disabled, WIFI position is replaced by a "geoloc start" uplink message bit 7: Led blinks when a GPS fix is received. Set to enable the feature, reset to disable bit 8: Set to enable the sending of a motion start message bit 9: Set to enable the sending of a motion end message bit 10: Set to enable a new OTA join when leaving the mode OFF bit 11: Asymmetric BLE pairing: 0 accepted, 1 rejected bit 12: Set to enable Long Wifi Payload bit 13: When set the number of entries carried in the scan collection uplink message is larger and use a payload size of 91 bytes. When reset, the usual payload size is used (36 bytes). bit 14: Set to enable device autostart after initialization, without a button press. bit 15: Set to enable forbid Mode OFF Bit 16: Set to enable SOS Mode Sound Bit 17: Set to enable Automatic datarate selection Bit 18: Set to enable extended position format Bit 19: reserved Bit 20: Set to activate BLE CLI Bit 21: Set to require the passkey for BLE bonding procedure Bit 22: Set to build Wi-Fi database using GPS positions Bit 23: Set to enable Wi-Fi beaconing Bit 24: Set to enable gnssHoldOn mode { "framePendingMechanism":true, "buttonPressToTurnOFF":false, "doubleClickIsSosModeOrAlert":false, "downlinkSetParameterConfirmation":false, "wifiPayloadWithNoCypher":true, "bleAdvertisingAtStart":false, "selectWifiScanOrGeolocStartMessage":false, "ledBlinkWithGpsFix":false,	1.7 (bit0 - bit5) 1.7.3 (bit6 - bit7) 1.8 (bit8 - bit11), 1.9 (bit12), 2.0 (bit13,14) 2.2.1 (bit 15,16,17) 2.3 (bit 18,19,20) 2.4 (bit 21) 2.5 (bit 22,23) 2.6 (bit 24)

		"startMotionEventUplink":true, "endMotionEventUplink":false, "otaJoinWhenLeavingModeOff":false, "rejectAsymmetricPairing":false, "enableLongWifiPayload":false, "collectionLongReport":false, "autoStart":false, "forbidModeOff":false, "sosModeSound":false, "automaticDatarateSelection":false, "extendedPosition":false, "bleCli": false, "blePasskey": false, "wifiDatabase": false, "activeWifiScan": false, " gnssHoldOnEnable": false }	

motionSensitivity
(motion_sensitivity)
	Object	Accelerometer configuration.
		Attributes:
		- sensitivity:
		• *Value range [1-30]: Mode High sensitivity, low power. Step 0.063g*
		• *Value 100: Regular backward compatibility.*
		• *Value range [101..199] => Backward compatibility with sensitivity (as before)*
		- outputDataRate: ODR
		• *12_5_HZ: 12.5 Hz*
		• *25_HZ: 25 Hz*
		• *50_HZ: 50 Hz*
		• *100_HZ: 100 Hz*
		• *200_HZ: 200 Hz*
		- fullScale: full-scale
		• *2_G: 2G*
		• *4_G: 4G*
		• *8_G: 8G*
		• *16_G: 16G*
		Example:
		{
		"fullScale":"16_G",
		"outputDataRate":"100_HZ",
		"sensitivity":1
		}
			1.8
			The format has changed since 2.4

shockDetection (shock_detection)	Number	This parameter provides the configuration of the sensitivity of the shock detection. [0-128]	1.8 Range changed in 2.3 Range changed in 2.4

periodicActivityPeriod (periodic_activity_period)	Number	Integer value in the range [0,1800-86400], seconds	1.8
motionDuration (motion_duration)	Number	Integer value in the range [60-3600], seconds	1.8
geofencingScanPeriod (geofencing_scan_period)	Number	Period (in seconds) for position message reporting while in Geofencing Active state. [0-300]	1.8
geofencingCollectPeriod (geofencing_collect_period)	Number	Period (in seconds) for position message reporting while in Geofencing Active state. [0, 15-3600]	2.2 (called geofencing ul period before)
bleRssiFilter (ble_rssi_filter)	Number	RSSI value to filter BLE beacons with BGPS. [-100- -40]	1.8
temperatureHigh (temperature_high)	Number	Configure High Temperature for Temperature Monitoring [-44- 85,255], ºC	1.9
temperatureLow (temperature_low)	Number	Configure Low Temperature for Temperature Monitoring [-44- 85,255] , ºC	1.9
temperatureAction (temperature_action)	String	Configure the action to be done when entering critical state Acceptable values are: • NO_ACTION: No action. • TEMPERATURE_HIGH: Geolocation disabled if temperature_high is reached. • TEMPERATURE_HIGH_TEMPERATURE_LOW: Geolocation disabled if temperature_low or temperature_high is reached. • TEMPERATURE_LOW: Geolocation disabled if temperature_low is reached.	1.9
transmitStratCustom (transmit_strat_custom)	Object	Custom transmit strategy. See Custom transmit strategy object	1.9
reedSwitchConfiguration (reed_switch_configuration)	String	Reed switch status. Values can be: • DISABLE_REED_SWITCH: disable (no action) • NORMAL_REED_SWITCH: Reset • BUTTON_REED_SWITCH: Act as the usual button • START_BLE_ADVERTISEMENT_REED_SWITCH: Start BLE advertisement	2.0
batteryCapacity (battery_capacity)	Number	Battery capacity in mAh	2.0
collectionScanType (collection_scan_type)	String	Provide the type of the collection to be done. Values can be: • NO_COLLECTION_SCAN: disable (no action)	2.0

		• BLE_BEACONS: A BLE scan will be done after the usual geolocation. The Lora reporting of this scan will be done after the usual geolocation • WIFI_BSSID: A WIFI scan will be done after the usual geolocation. The Lora reporting of this scan will be done after the usual geolocation	
collectionBleFilterType (collection_ble_filter_type)	String	BLE filter type used for BLE collection only. Values can be: • NO_FILTER: MAC addresses reported in the collection uplink • EDDYSTONE_UID: Eddystone UUID only. Beacons ID reported in the collection uplink • EDDYSTONE_URL: Eddystone URL only. MAC addresses reported in the collection uplink • ALL_EDDYSTONE: Eddystone all. MAC addresses reported in the collection uplink • IBEACON_ONLY: iBeacons. Beacon IDs reported in the collection uplink • ALTBEACON_ONLY: altBeacons. Beacon IDs reported in the collection uplink	2.0
collectionBleFilterMain1 (collection_ble_filter_main_1)	Number	BLE main filter part 1 used for BLE collection only	2.0
collectionBleFilterMain2 (collection_ble_filter_main_1)	Number	BLE main filter part 2 used for BLE collection only	2.0
collectionBleFilterSecValue (collection_ble_filter_sec_value)	Number	Value used for the BLE secondary filter (BLE collection only)	2.0
collectionBleFilterSecMask (collection_ble_filter_sec_mask)	Number	Mask used for the BLE secondary filter (BLE collection only)	2.0
collectionNbEntry (collection_nb_entry)	Number	Number of entries in the collection [1-20]	2.0
networkTimeoutCheck (network_timeout_check)	Number	LoRa network monitoring. A timer using this duration is restarted after each last downlink received. Once elapsed, the tracker includes a link check request in all of this uplink. Once the answer is received, the timer restarts. A null value disables this monitoring. Duration expressed in seconds [0,86400-5184000].	2.0
networkTimeoutReset (network_timeout_reset)	Number	LoRa network monitoring. A timer using this duration is restarted after each downlink received. Once elapsed, the tracker resets. This parameter is usually used with network_timeout_check. Duration expressed in seconds [0,21600-2592000].	2.0

gnssConstellation (gnss_constellation)	String	GNSS constellation set to be used for making fixes. Values can be: • GPS_ONLY: GPS only • GLONASS_ONLY: GLONASS only • GPS_GLONASS: GPS and GLONASS • GPS_GALILEO: GPS and GALILEO • GPS_GLONASS_GALILEO: GPS, GLONASS and GALILEO • BEIDOU_ONLY: BEIDOU only • GPS_BEIDOU: GPS and BEIDOU	2.0
proximityMinimumScanPower (prox_scan_pwr_min)	Number	Minimum power (RSSI) of a scan entry to be considered, dBm [-90 ... 10].	2.1
proximityDistanceCoefficient (prox_distance_coef)	Number	N=Prox_distance_coef/10 Distance d is evaluated from RSSI P (dBm), and reference distance d1 and power p1 perd=d1*10^((P1-P)/N)In free space N=20. For proximity detection needs to be tuned in 1-4m region depending on indoor multipath effects [1 – 256].	2.1
proximityScanFrequency (prox_scan_frequency)	Number	Number of scans in an hour [2 – 3600].	2.1
proximityBacktraceMaximumAge (prox_backtrace_max_age)	Number	Delay after which an entry is removed from short term buffer, and record trigger is evaluated, seconds [1 – 256].	2.1
proximityDistanceSlidingWindow (prox_dist_sliding_window)	Number	Duration of sliding window over which distance_average will be evaluated, seconds [1 – 256].	2.1
proximityExposure50 (Prox_exposure_50)	Number	Marginal exposure index for [0,50cm] range during 1 second [0 – 256].	2.1
proximityExposure100 (Prox_exposure_100)	Number	Marginal exposure index for [50,100cm] range during 1 second [0 – 256].	2.1
proximityExposure150 (Prox_exposure_150)	Number	Marginal exposure index for [100,150cm] range during 1 second [0 – 256].	2.1
proximityExposure200 (Prox_exposure_200)	Number	Marginal exposure index for [150,200cm] range during 1 second [0 – 256].	2.1
proximityExposure250 (Prox_exposure_250)	Number	Marginal exposure index for [200,250cm] range during 1 second [0 – 256].	2.1
proximityExposure300 (Prox_exposure_300)	Number	Marginal exposure index for [250,300cm] range during 1 second [0 – 256].	2.1

proximityExposure400 (Prox_exposure_400)	Number	Marginal exposure index for [300,400cm] range during 1 second [0 – 256].	2.1
proximityImmediateAlarmDistance (prox_alarm_dist_immediate)	Number	Detection threshold in meters for immediate alarm state trigger [0-10].	2.1
proximityAlarmExposure (prox_alarm_exposure)	Number	Cumulated Exposure threshold to trigger alarm state [0-65535].	2.1
proximityImmediateWarningDistance (prox_warn_dist_immediate)	Number	Detection threshold in meters for immediate warning state trigger [0-10].	2.1
proximityWarningExposure (prox_warn_exposure)	Number	Cumulated Exposure threshold to trigger warning state [0-65535].	2.1
proximityImmediateRecordDistance (prox_record_dist_immediate)	Number	Detection threshold in meters for Contact recording [0-10].	2.1
proximityRecordExposure (prox_record_exposure)	Number	Cumulated Exposure threshold to trigger record action [0-65535].	2.1
proximityAlarmBuzzerDuration (prox_alarm_buz_dur)	Number	Buzzer duration in seconds [0 – 256].	2.1
proximityWarningBuzzerDuration (prox_warn_buz_dur)	Number	Buzzer duration in seconds [0 – 256].	2.1

proximityContactPolicy (prox_contact_policy)	Object	Contact policy.
		Attributes:
		- enable: boolean
		- uplink: boolean
		- store: Boolean
		Example:
		{
		" enable ":true,
		"uplink":false,
		"store":true
		}
			2.1

proximityScanDuration (prox_scan_duration)	Number	Duration in seconds of multichannel scan [0-60].	2.1
proximityScanWindow (prox_scan_window)	Number	Duration in milliseconds of scan on each channel [10-10240].	2.1
proximityScanInterval (prox_scan_Interval)	Number	BLE scan interval in milliseconds [15-10245].	2.1

proximityAlarmRemanence (prox_alarm_remanence)	Number	Seconds a Contact will remain in Alarm list even if no longer detected [0-256].	2.1
proximityWarningRemanence (prox_warn_remanence)	Number	Seconds a Contact will remain in Alarm list even if no longer detected [0-256].	2.1
proximityBeaconRepeat (prox_bcn_repeat)	Number	Beacon repeat frequency, in milliseconds [0-65535].	2.1
proximityBeaconTXPower (prox_bcn_tx_power)	Number	Advertised calibrated TX power at 1m, in dBm [-60.. -20].	2.1
proximityReminderPeriod (prox_reminder_period)	Number	Seconds, periodicity of reminder beep during warning or alarm states. 0 disables the reminder beep [0-256].	2.1
proximityReminderDistance (prox_reminder_distance)	Number	Distance in meters beyond which the reminder beep is disabled when no tags in alarm or warning state are within this radius [0-25.6].	2.1
proximityWarningDisableDistance (prox_warn_disable_dist)	Number	Distance in meters beyond which the warning soft state timer will not be reset [1-25.6].	2.1
proximityAlarmDisableDistance (prox_alarm_disable_dist)	Number	Distance in meters beyond which the warning softstate timer will not be reset [1-25.6].	2.1
proximityMaxSpeedFilter (prox_max_speed_filter)	Number	Maximum speed of a pedestrian. Used to filter out the outliers meters/s [0-4].	2.1
proximityMaxUpdate (prox_max_update)	Number	Maximum duration in seconds for a record without sending any information [300 - 43200].	2.1
bleFilterType (position_ble_filter_type)	String	BLE filter type. Values can be: • NO_FILTER: MAC addresses reported in the collection uplink • EDDYSTONE_UID: Eddystone UUID only. Beacons ID reported in the collection uplink • EDDYSTONE_URL: Eddystone URL only. MAC addresses reported in the collection uplink • ALL_EDDYSTONE: Eddystone all. MAC addresses reported in the collection uplink • IBEACON_ONLY: iBeacons. Beacon IDs reported in the collection uplink • ALTBEACON_ONLY: altBeacons. Beacon IDs reported in the collection uplink	2.1
bleFilterMain1 (position_ble_filter_main_1)	Number	BLE main filter part 1	2.1
bleFilterMain2	Number	BLE main filter part 2	2.1

(position_ble_filter_main_2)			
bleFilterSecValue (position_ble_filter_sec_value)	Number	Value used for the BLE secondary filter	2.1
bleFilterSecMask (position_ble_filter_sec_mask)	Number	Mask used for the BLE secondary filter	2.1
bleFilterReportType (position_ble_report_type)	String	Configure BLE data to report in payloads. Values can be: • MAC_ADDRESS • SHORT_ID • LONG_ID • SHORT_ID_MAJOR_MINOR_FIELD	2.1 SHORT_ID_MAJOR_MINOR_FIELD Since 2.2
buzzerVolume (buzzer_volume)	Number	Buzzer Volume by modifying the buzzer duty cycle. This value is not linear and the effect changes according to the type of buzzer [0-100].	2.1
angleDetectionMode (angle_detect_mode)	String	Defines the operational mode of the angle detection feature. Values can be: • DISABLED • CRITICAL_ANGLE_DETECTION • ANGLE_DEVIATION_DETECTION • SHOCK	2.2
angleReferenceAcquisition (angle_ref_acq)	String	Defines the mode of the reference acquisition. Values can be: • MANUAL • CONFIGURED • AUTOMATIC • ASSISTED	2.2
angleReferenceAccelerationX (angle_ref_acc_x)	Number	Acceleration in mG belonging to the X axis. First coordinate of the reference orientation vector.	2.2
angleReferenceAccelerationY (angle_ref_acc_y)	Number	Acceleration in mG belonging to the Y axis. Second coordinate of the reference orientation vector.	
angleReferenceAccelerationZ (angle_ref_acc_z)	Number	Acceleration in mG belonging to the Z axis. Third coordinate of the reference orientation vector.	2.2
angleCritical (angle_critical)	Number	Critical angle in degrees.	2.2
angleCriticalHysteresis (angle_critical_hyst)	Number	Critical angle hysteresis in degrees.	2.2
angleReportingMode (angle_report_mode)	Object	Angle reporting mode defines which Angle Detection events are reported, defined as a bitmap where 1 enables a reporting and 0 disables it.:	2.2

		Bit 0: Report the transitions normal → critical.
		Bit 1: Report the transitions critical → normal.
		Bit 2: Report the transitions learning → normal.
		Bit 3: Report the transitions normal → learning.
		Bit 4: Report the transitions: critical → learning.
		
		Example:
		{
		"normalToCritical":true,
		"criticalToNormal":false,
		"learningToNormal":false,
		"normalToLearning":false,
		"criticalToLearning":false
		}	

angleReportingPeriod (angle_report_period)	Number	Angle detection reporting period between repetitions in seconds.	2.2
angleReportingRepeat (angle_report_repeat)	Number	Defines how many times an angle detection report is repeated	2.2
angleRisingTime (angle_rising_time)	Number	Rising time phase duration in seconds.	2.2
angleFallingTime (angle_falling_time)	Number	Falling time phase duration in seconds.	2.2
angleLearningTime (angle_learning_time)	Number	Learning time phase duration in seconds.	2.2
angleAccelerationAccuracy (angle_acc_accuracy)	Number	Accuracy in mg applied to the acceleration measures.	2.2
angleDeviationDelta (angle_deviation_delta)	Number	Angle from previously reported orientation which triggers a new orientation report.	2.2
angleDeviationMinInterval (angle_deviation_min_interval)	Number	No report sent before the delay from the previous report, any deviation before this delay will be ignored.	2.2
angleDeviationMaxInterval (angle_deviation_max_interval)	Number	No report sent after this delay from previous report, an update of current gravity vector will be sent regardless of delta value after this time.	2.2
defaultProfile (default_profile)	String	Defines the initialization value of the profile. It is used when the device is powered on for the first time or resets. Values can be: • UNKNOWN • SLEEP • ECO	2.2

		• INTENSIVE **READ_ONLY**	
passwordCommandLineInterface (password)	Number	Password used to configure User CLI access.	2.2
gpsT0Timeout (gps_t0_timeout)	Number	Time in seconds to abort the GPS geoloc when not enough satellites are in view.	2.2
gpsFixTimeout (gps_fix_timeout)	Number	Time in seconds to abort the current GPS geoloc if there is no GPS position at the end of the configured period.	2.2
geofencingScanDuration (geofencing_scan_duration)	Number	Time in milliseconds for geofencing BLE scan duration	2.2
beaconingType (beaconing_type)	Number	Represents the beaconing advertising type. Values can be: • DISABLED • PROXIMITY • QUUPPA • EDDYSTONE_UID • IBEACON • ALT_BEACON	2.2
beaconingTxPower (beaconing_tx_power)	String	Beaconing TX power in dBm. Values can be : • PLUS_4_DBM • PLUS_3_DBM • 0_DBM • MINUS_4_DBM • MINUS_8_DBM • MINUS_12_DBM • MINUS_16_DBM • MINUS_20_DBM • MINUS_40_DBM	2.2
beaconingStaticInterval (beaconing_static_interval)	Number	When beaconing is active and the tracker is static, the beaconing static interval represents the time in milliseconds for beacon advertising period.	2.2
beaconingMotionInterval (beaconing_motion_interval)	Number	When beaconing is active and the tracker is moving, the beaconing motion interval represents the time in milliseconds for beacon advertising period.	2.2
beaconingMotionDuration(beaconing_motion_duration)	Number	When Beaconing is active, time in seconds to be considered in motion when receiving a motion event.	2.2
bleConnectivityAdvertisingDuration (ble_cnx_adv_duration)	Number	Time in seconds for BLE advertisement duration when the device is not bonded.	2.2

beaconId0 (beacon_id_0)	Number	Beacon ID advertised in BLE advertisement payload, part 0	2.3
beaconId1 (beacon_id_1)	Number	Beacon ID advertised in BLE advertisement payload, part 1	2.3
beaconId2 (beacon_id_2)	Number	Beacon ID advertised in BLE advertisement payload, part 2	2.3
beaconId3 (beacon_id_3)	Number	Beacon ID advertised in BLE advertisement payload, part 3	2.3
beaconId4 (beacon_id_4)	Number	Beacon ID advertised in BLE advertisement payload, part 4	2.3
sosPeriod (sos_period)	Number	SOS period duration, in seconds.	2.3
motionDebounce (motion_debounce)	Number	Minimum duration for a movement to be detected. For an initial movement, the duration is given by the formula: motion_debounce * 160ms. For a continuous motion, the duration is given by the formula: motion_debounce * 80ms.	2.3
buttonMapping (button_mapping)	Object	Configure button actions according to the bitmap description. Bit 0-3: button press action Bit 4-7: one button click action Bit 8-11: two button clicks action Bit 12-15: three button clicks or more action Bit 16-19: button press duration in seconds Acceptable values for action are: • NO_ACTION • BATTERY_LEVEL_WITH_LED • START_STOP_SOS • START_ALERT • WHITELIST_WITH_PROXIMITY • ANGLE_DETECTION_MANUL_TRIGGER	2.3
defaultDatarate (default_datarate)	Number	Configure Default Datarate. Range is [–1,0-7]	2.3
gpsEHPEMotion (gps_ehpe_motion)	Number	Acceptable GPS horizontal error for GPS geolocation. Applied when the tracker is in motion. [0..100]	2.4
gpsConvergenceMotion (gps_convergence_motion)	Number	Time let to the GPS module to refine the calculated position. Applied when the tracker is in motion. [0..300]	2.4

gpsT0TimeoutMotion (gps_t0_timeout_motion)	Number	Time in seconds to abort the GPS geolocation when not enough satellites are in view. Applied when the tracker is in motion. [0..300]	2.4
dynamicProfile (profile)	String	It must be used to change dynamically the profile. The value is not stored in flash, and the device will restart with default Profile after an empty battery or a reset. Values can be: • Unknown • Sleep • Eco • Intensive	2.2
powerConsumption (consumption)	Number	Tracker current consumption in mAh [0-4294967295].	2.0
bleBond	String	BLE bond status. Values can be: • BONDED • NOT_BONDED Only NOT_BONDED value can be set.	1.9
bleCliActive	Number	Enable(1)/disable(0) CLI over BLE stream	2.3
mode (mode)	String	Tracker operating mode. Values can be: • STAND_BY • OFF • MOTION_TRACKING • PERMANENT_TRACKING • MOTION_START_END_TRACKING • ACTIVITY_TRACKING	1.9
xAccelerometerValue	Number	Accelerometer X axis value in mg (read only).	1.9
yAccelerometerValue	Number	Accelerometer Y axis value in mg (read only).	1.9
zAccelerometerValue	Number	Accelerometer Z axis value in mg (read only).	1.9
bleFirmwareVersion	String	BLE firmware version. Read only.	1.8
firmwareVersion	String	MCU firmware version. Read only.	1.8

5.1.2. Custom transmit strategy object

The custom transmit strategy object describes the custom transmission strategy used when the transmitStrat parameter is set to *CUSTOM_STRATEGY*. The fields are described in the table below:

Parameter name	Type	Field value	Supported since FW version
ADREnabled	Boolean	ADR activation in static	1.9
transmissionType	String	Acceptable values are: • *SINGLE*: Single transmission. • *DOUBLE:* Double transmission.	1.9
firstTransmissionDatarateDistribution	String	Datarate distribution for the first transmission • *RANDOM* • *BELL_CURVE* • *RING*	1.9
secondTransmissionDatarateDistribution	String	Datarate distribution for the second transmission • *RANDOM* • *BELL_CURVE* • *RING*	1.9

firstTransmissionDatarate	Object	Object to select each datarate for the first transmission.
		Example:
		{
		"dr0": true,
		"dr1": false,
		"dr2": false,
		"dr3": true,
		"dr4": false,
		"dr5": false,
		"dr6": false,
		"dr7": true
		}
			1.9

secondTransmissionDatarate	Object	Object to select each datarate for the second transmission.
		Example:
		{
		"dr0": true,
		"dr1": true,
		"dr2": true,
		"dr3": false,
		"dr4": false,
		"dr5": false,
		"dr6": false,
		"dr7": false
		}
			1.9

5.1.3. Measured temperature object

The measured temperature object contains information about temperature alerts recorded in the tracker. The properties are described in the table below:

Parameter name	Type	Field value	Supported since FW version
state	String	Possible values are (string format): • *NORMAL*: No temperature alert • *HIGH:* High temperature alert • *LOW:* Low temperature alert • *FEATURE_NOT_ACTIVATED*: Temperature measure feature is disabled.	1.9
max	Number	Maximum measured temperature	1.9
min	Number	Minimum measured temperature	1.9
highCounter	Number	From the time a high temperature event is sent, this counter is incremented every 10 minutes	1.9
lowCounter	Number	From the time a low temperature event is sent, this counter is incremented every 10 minutes	1.9

5.1.4. Data scan collection object

The data scan collection object describes the information received with a data scan collection uplink message. The properties are described in the table below:

Parameter name	Type	Field value	Supported since FW version
scanType	String	Possible values are: • *BLE_BEACONS*: BLE beacons scan • *WIFI_BSSID:* Wifi BSSID scan • *BLE_BEACONS_COLLECTION:* The collection will be done and BLE beacons will be reported.	2.0 BLE_BEACONS_COLLECTION Since 2.2
again	Boolean	Again flag	2.0
dataFormat	String	Possible values are: • *BEACON_ID* • *MAC_ADDRESS*	2.0
fragmentIdentification	Number	Identifier of the current fragment	2.0
collectionIdentifier	Number	Identifier of the current collection	2.0
hash	Number	Hash of the matching position payload	2.0

beaconIdData	Array of objects	Array of beacon ID, RSSI couples.
		Example:
		[
		{
		"beaconId": "CAFEBB",
		"rssi": -128
		},
		{
		"beaconId": "2BAFFE",
		"rssi": -56
		}
		]
			2.0

macAddressData	Array of objects	Array of BSSID, RSSI couples.
		Example:
		[
		{
		"bssid": "6e:aa:1b:1a:ec:11",
		"rssi": -128
		},
		{
		"bssid": "4f:c9:cd:b8:d3:09",
		"rssi": -56
		}
		].
			2.0

5.1.5. Proximity notification object

The proximity notification object describes the information received with a proximity notification uplink message. The properties are described in the table below:

Parameter name	**Type**	**Field value**	**Supported since FW version**
notificationType	String	Possible values are: • *WARNING_MESSAGE* • *ALERT_MESSAGE* • *RECORD_MESSAGE*	2.1
encrypted	Boolean	When set, the RPI as well as the metadata fields are encrypted.	2.1
recordAction	String	Possible values are: • *RECORD_START* • *RECORD_UPDATE* • *RECORD_STOP*	2.1

rollingProximityIdentifier	String	RPI: Rolling proximity identifier. Hexadecimal string.	2.1
closestDistanceRecording	Number	Dmin: Closest distance recorded. Meters	2.1
distanceAverageRecorded	Number	Davg: Distance average recorded. Meters	2.1
cumulatedExposure	Number	Cexpo: Cumulated exposure. No unit	2.1
metadata	String	Metadata: metadata. Hexadecimal string	2.1
cumulatedContactDuration	Number	CCD: Cumulated contact duration. Seconds	2.1
currentDailyExposure	Number	CDE: current daily exposure. Seconds	2.1

5.1.6. Proximity daily report object

The proximity daily report object describes the information received with a proximity daily report uplink message. The properties are described in the table below:

Parameter name	**Type**	**Field value**	**Supported since FW version**
dailyAlertDay0	Number	DA0: Daily alert day 0 (just ended day). Number of times the tag triggered an alert in the day.	2.1
dailyWarningDay0	Number	DW0: Daily warning day 0 (just ended day). Number of times the tag triggered a warning in the day.	2.1
dailyExposureDay0	Number	DE0: Daily exposure day 0 (just ended day). Cumulated exposure for the day (all contacts cumulated).	2.1
dailyAlertDay1	Number	DA1: Daily alert day -1 (previous day). Number of times the tag triggered an alert in day -1.	2.1
dailyWarningDay1	Number	DW1: Daily warning day -1 (previous day). Number of times the tag triggered a warning in day -1.	2.1
dailyExposureDay1	Number	DE1: Daily exposure day -1 (previous day). Cumulated exposure for day -1	2.1
dailyAlertDay2	Number	DA2: Daily alert day -2. Number of times the tag triggered an alert in day -2.	2.1
dailyWarningDay2	Number	DW2: Daily warning day -2. Number of times the tag triggered a warning in day -2.	2.1
dailyExposureDay2	Number	DE2: Daily exposure day -2. Cumulated exposure for day -2	2.1

5.1.7. Proximity white listing object

The proximity white listing object describes the information received with a proximity white listing uplink message. The properties are described in the table below:

Parameter name	Type	Field value	Supported since FW version
encrypted	Boolean	When set, the RPI as well as the metadata fields are encrypted.	2.1
rollingProximityIdentifier	String	RPI: Rolling proximity identifier. Hexadecimal string.	2.1
list	String	Possible values are: • *PEER_LIST* • *WARNING_LIST* • *ALERT_LIST*	2.1
recordStatus	String	Possible values are: • *NOT_WHITE_LISTED* • *WHITE_LISTED*	2.1
solicited	Boolean	When set, the message is a reply to a LoRaWAN downlink request	2.1

5.1.8. Proximity daily response object

The proximity daily response object describes the information received with a proximity daily response uplink message. The properties are described in the table below:

Parameter name	Type	Field value	Supported since FW version
dayIdentifier	Number	Day identifier matching the request	2.1
dailyAlert	Number	DA: Daily alert. Number of times the tag triggered an alert in the day.	2.1
dailyWarning	Number	DW: Daily warning. Number of times the tag triggered a warning in the day.	2.1
dailyExposure	Number	DE: Daily exposure. Cumulated exposure for the day (all contacts cumulated).	2.1

5.1.9. Angle detection object

The angle detection object describes the information received with an angle detection event uplink message. The properties are described in the table below:

<table>
<tr><th>Parameter name</th><th>Type</th><th>Field value</th><th>Supported since FW version</th></tr>
<tr><td>flags</td><td>Object</td><td>Flags providing status and report reason, the attributes are described in the table below:

<table>
<tr><th>Attribute name</th><th>Type</th><th>Field value</th></tr>
<tr><td>transition</td><td>String</td><td>Values can be:
LEARNING_TO_NORMAL
NORMAL_TO_LEARNING
NORMAL_TO_CRITICAL
CRITICAL_TO_NORMAL
CRITICAL_TO_LEARNING</td></tr>
<tr><td>triggerType</td><td>String</td><td>Values can be:
CRITICAL_ANGLE_REPORTING
ANGLE_DEVIATION_REPORTING
SHOCK_TRIGGER
RFU</td></tr>
<tr><td>notificationIdentifier</td><td>Number</td><td>Notification identifier</td></tr>
</table></td><td>2.2</td></tr>
<tr><td>age</td><td>Number</td><td>Age of the measure in seconds.</td><td>2.2</td></tr>
<tr><td>referenceVector</td><td>Array of number</td><td>Reference gravity vector with the following format: RV = [Rx, RY, RZ]
Rx: Acceleration in mg on X axis used for the reference vector.
RY: Acceleration in mg on Y axis used for the reference vector.
RZ: Acceleration in mg on Z axis used for the reference vector.</td><td>2.2</td></tr>
<tr><td>criticalVector</td><td>Array of number</td><td>Critical gravity vector with the following format: CV = [Cx, CY, CZ]
Cx: Acceleration in mg on X axis from last measurement.
CY: Acceleration in mg on Y axis from last measurement.
CZ: Acceleration in mg on Z axis from last measurement.</td><td>2.2</td></tr>
<tr><td>angle</td><td>Number</td><td>Angle in degrees between the reference vector and the last measurement.</td><td>2.2</td></tr>
</table>

5.1.10. Geofencing notification object

The geofencing notification object describes the information received with a geofencing event uplink message. The properties are described in the table below:

Parameter name	Type	Field value	Supported since FW version

geofencingFormat	Number	Geofencing format	2.2
geofencingType	String	Geofencing notification type. Values can be: • SAFE_AREA: The badge moved from hazardous area to a safe aera. • ENTRY: The badge moved from non-monitored aera to monitored aera. • EXIT: The badge moved from monitored aera to non-monitored aera. • HAZARD: The badge moved to hazardous area.	2.2
id	Number	Identifier: its size depends on the geofencing format.	2.2

5.1.11. GPS previous object

The GPS previous object describes the previous GPS fix received in the GPS position message.
The properties are described in the following table:

Parameter name	Type	Field value	Supported since FW version
age	Number	Age of the previous GPS fix expressed in seconds	2.2.1
dynamicMotionState	String	Movement status of the previous GPS fix. It can be MOVING or STATIC	2.2.1
latitude	Number	Latitude of the previous GPS fix expressed in degrees.	2.2.1
longitude	Number	Longitude of the previous GPS fix expressed in degrees.	2.2.1

5.1.12. Health Status object

The Health status object describes the total power consumption of the tracker.
The properties are described in the following table:

Parameter name	Type	Field value	Supported since FW version
totalConsumption	Number	Total consumption value in mAh.	2.3
maxTemperature	Number	Signed Value on 8 bits, temperature in °C, same encoding as value in header.	2.3
maxTemperature	Number	Signed Value on 8 bits, temperature in °C, same encoding as value in header.	2.3
loraPowerConsumption	Number	Cumulated consumption for the Lora radio, in mAh.	2.3
blePowerConsumption	Number	Cumulated consumption for the BLE, in mAh.	2.3

gpsPowerConsumption	Number	Cumulated consumption for the GPS, in mAh.	2.3
wifiPowerConsumption	Number	Cumulated consumption for the WIFI, in mAh.	2.3
batteryVoltage	Number	Battery voltage, in mV.	2.3

5.1.13. SMS object

The SMS object describes a SMS message sent over LoRa.
The properties are described in the following table:

Parameter name	**Type**	**Field value**	**Supported since FW version**
destinationId	Number	ID of the SMS recipient.	2.5
message	String	SMS message.	2.5

5.2. Examples

5.2.1. Position message

This example shows the decoding of an uplink position message.
The LoRa payload to be parsed is the following:
"032c36844926c8675e842cd7c76cc217d827bebc34dbfdad0e31b4c8675e8200d4b2"
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"

  },
  "thing": {
```

```
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary":
"032c36844926c8675e842cd7c76cc217d827bebc34dbfdad0e31b4c8675e8200d4b2"
  }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
{
  "direction": "uplink",
  "sourceTime": 1647255378016,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary":
"032c36844926c8675e842cd7c76cc217d827bebc34dbfdad0e31b4c8675e8200d4b2"
  },
  "message": {
    "messageType": "POSITION_MESSAGE",
    "age": 304,
    "trackingMode": "MOTION_TRACKING",
    "batteryLevel": 54,
    "batteryStatus": "OPERATING",
    "ackToken": 4,
    "rawPositionType": "WIFI_BSSIDS_WITH_NO_CYPHER",
    "periodicPosition": false,
    "temperatureMeasure": 22.8,
    "sosFlag": 0,
    "appState": 1,
    "dynamicMotionState": "MOVING",
```

```
    "onDemand": false,
    "payload":
"032c36844926c8675e842cd7c76cc217d827bebc34dbfdad0e31b4c8675e8200d4b2",
    "deviceConfiguration": {
      "mode": "MOTION_TRACKING"
    },
    "wifiBssids": [
      {
        "bssid": "c8:67:5e:84:2c:d7",
        "rssi": -57
      },
      {
        "bssid": "6c:c2:17:d8:27:be",
        "rssi": -68
      },
      {
        "bssid": "34:db:fd:ad:0e:31",
        "rssi": -76
      },
      {
        "bssid": "c8:67:5e:82:00:d4",
        "rssi": -78
      }
    ]
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "POSITION_MESSAGE"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "MOVING"
```

```
      },
      "appState": {
        "type": "int64",
        "record": 1
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
      "onDemand": {
        "type": "boolean",
        "record": false
      },
      "ackToken": {
        "type": "int64",
        "record": 4
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "MOTION_TRACKING"
      },
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
        "record": 54
      },
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 22.8
      },
      "deviceMode": {
        "type": "string",
        "record": "MOTION_TRACKING"
      },
      "age": {
```

```
        "unitId": "s",
        "type": "double",
        "record": 304
      },
      "rawPositionType": {
        "type": "string",
        "record": "WIFI_BSSIDS_WITH_NO_CYPHER"
      }
    }
  }
}
```

---------- Snippet end ----------

Where `message` is the decoded payload.

5.2.2. Heartbeat message

This example shows the decoding of an uplink heartbeat message.
The LoRa payload to be parsed is the following: "05003683e0010109c3"
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
},
"thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary": "05805f7f70400202c7030205"
  }
```

```
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------


```
{
  "direction": "uplink",
  "sourceTime": 1647255543630,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary": "05805f7f70400202c7030205"
  },
  "message": {
    "messageType": "HEARTBEAT",
    "trackingMode": "ACTIVITY_TRACKING",
    "batteryLevel": 95,
    "batteryStatus": "OPERATING",
    "ackToken": 7,
    "firmwareVersion": "2.2.199",
    "bleFirmwareVersion": "3.2.5",
    "resetCause": 64,
    "periodicPosition": false,
    "temperatureMeasure": 20.2,
    "sosFlag": 0,
    "appState": 0,
    "dynamicMotionState": "STATIC",
    "onDemand": false,
    "payload": "05805f7f70400202c7030205",
    "deviceConfiguration": {
      "mode": "ACTIVITY_TRACKING"
    }
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "HEARTBEAT"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "STATIC"
      },
      "appState": {
        "type": "int64",
        "record": 0
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
      "onDemand": {
        "type": "boolean",
        "record": false
      },
      "ackToken": {
        "type": "int64",
        "record": 7
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "ACTIVITY_TRACKING"
      },
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
```

```
        "type": "double",
        "record": 95
      },
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 20.2
      },
      "deviceMode": {
        "type": "string",
        "record": "ACTIVITY_TRACKING"
      },
      "firmwareVersion": {
        "type": "string",
        "record": "2.2.199"
      },
      "bleFirmwareVersion": {
        "type": "string",
        "record": "3.2.5"
      },
      "resetCause": {
        "type": "int64",
        "record": 64
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.3. Activity Status message

This example shows the decoding of an uplink Activity Status message.
The LoRa payload to be parsed is the following: "07004586000100000000a"
The LoRa transmission port number is 18.

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
```

```
  "meta": {
    "lorawan": {
      "fPort": 18
    },
   "driverId": "abeeway:asset-tracker:3"
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary": "0700458600010000000a"
  }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
{
  "direction": "uplink",
  "sourceTime": 1647255590181,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary": "0700458600010000000a"
  },
  "message": {
    "messageType": "ACTIVITY_STATUS",
    "trackingMode": "STAND_BY",
    "batteryLevel": 69,
    "batteryStatus": "OPERATING",
    "ackToken": 0,
    "periodicPosition": false,
    "temperatureMeasure": 23.8,
    "miscDataTag": "ACTIVITY_COUNTER",
    "sosFlag": 0,
```

```
    "appState": 0,
    "dynamicMotionState": "STATIC",
    "onDemand": false,
    "payload": "07004586000100000000a",
    "activityCount": 10,
    "deviceConfiguration": {
      "mode": "STAND_BY"
    }
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "ACTIVITY_STATUS"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "STATIC"
      },
      "appState": {
        "type": "int64",
        "record": 0
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
      "onDemand": {
        "type": "boolean",
        "record": false
      },
      "ackToken": {
        "type": "int64",
        "record": 0
```

```
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "STAND_BY"
      },
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
        "record": 69
      },
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 23.8
      },
      "deviceMode": {
        "type": "string",
        "record": "STAND_BY"
      },
      "miscDataTag": {
        "type": "string",
        "record": "ACTIVITY_COUNTER"
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.4. Configuration message

This example shows the decoding of an uplink configuration message.
The LoRa payload to be parsed is the following: "`0784327d8002010000025815000000 64`"
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary": "0784327d8002010000025815000000064"
  }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
{
  "direction": "uplink",
  "sourceTime": 1647255677379,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary": "0784327d8002010000025815000000064"
  },
  "message": {
    "messageType": "CONFIGURATION",
    "trackingMode": "ACTIVITY_TRACKING",
    "batteryLevel": 50,
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
    "batteryStatus": "OPERATING",
    "ackToken": 8,
    "periodicPosition": false,
    "temperatureMeasure": 19.2,
    "miscDataTag": "DEVICE_CONFIGURATION",
    "sosFlag": 0,
    "appState": 0,
    "dynamicMotionState": "MOVING",
    "onDemand": false,
    "payload": "0784327d8002010000025815000000064",
    "deviceConfiguration": {
      "loralivePeriod": 600,
      "shockDetection": 100,
      "mode": "ACTIVITY_TRACKING"
    }
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "CONFIGURATION"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "MOVING"
      },
      "appState": {
        "type": "int64",
        "record": 0
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
      "onDemand": {
        "type": "boolean",
        "record": false
      },
      "ackToken": {
        "type": "int64",
        "record": 8
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "ACTIVITY_TRACKING"
      },
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
        "record": 50
      },
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 19.2
      },
      "deviceMode": {
        "type": "string",
        "record": "ACTIVITY_TRACKING"
      },
      "miscDataTag": {
        "type": "string",
        "record": "DEVICE_CONFIGURATION"
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.5. Shock Detection message

This example shows the decoding of an uplink shock detection message.
The LoRa payload to be parsed is the following: "07842080f00301ff41ff5903df"
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
   "driverId": "abeeway:asset-tracker:3"
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary": "07842080f00301ff41ff5903df"
  }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
{
  "direction":"uplink",
  "sourceTime":"2022-10-18T14:03:08.85Z",
  "meta":{
    "lorawan":{
      "fPort":18
    },
    "driverId":"abeeway:asset-tracker:3"
```

```
  },
  "raw":{
    "binary":"07842080f00301ff41ff5903df0000cafe"
  },
  "message":{
    "messageType":"SHOCK_DETECTION",
    "trackingMode":"ACTIVITY_TRACKING",
    "batteryLevel":32,
    "batteryStatus":"OPERATING",
    "ackToken":15,
    "periodicPosition":false,
    "temperatureMeasure":20.8,
    "miscDataTag":"SHOCK_DETECTION",
    "sosFlag":0,
    "appState":0,
    "dynamicMotionState":"MOVING",
    "onDemand":false,
    "payload":"07842080f00301ff41ff5903df0000cafe",
    "deviceConfiguration":{
      "mode":"ACTIVITY_TRACKING"
    },
    "nbOfshock":1,
    "accelerometerShockData":[
      -191,
      -167,
      991
    ],
    "gaddIndex":51966
  },
  "thing":{
    "application":{
      "producerId":"abeeway",
      "moduleId":"asset-tracker",
      "version":"2"
    },
    "points":{
      "messageType":{
        "type":"string",
        "record":"SHOCK_DETECTION"
      },
      "dynamicMotionState":{
        "type":"string",
        "record":"MOVING"
      },
```

```
      "appState":{
        "type":"int64",
        "record":0
      },
      "sosFlag":{
        "type":"int64",
        "record":0
      },
      "onDemand":{
        "type":"boolean",
        "record":false
      },
      "ackToken":{
        "type":"int64",
        "record":15
      },
      "periodicPosition":{
        "type":"boolean",
        "record":false
      },
      "trackingMode":{
        "type":"string",
        "record":"ACTIVITY_TRACKING"
      },
      "batteryStatus":{
        "type":"string",
        "record":"OPERATING"
      },
      "batteryLevel":{
        "unitId":"%",
        "type":"double",
        "record":32
      },
      "temperature":{
        "unitId":"Cel",
        "type":"double",
        "record":20.8
      },
      "deviceMode":{
        "type":"string",
        "record":"ACTIVITY_TRACKING"
      },
      "miscDataTag":{
        "type":"string",
```

```
        "record":"SHOCK_DETECTION"
      },
      "gaddIndex":{
        "type":"int64",
        "record":51966
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.6. Shutdown message

This example shows the decoding of an uplink shutdown message.
The LoRa payload to be parsed is the following: "090025860003"
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary": "090025860003"
  }
}'
```

---------- Snippet End ----------

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

The response is:

---------- Snippet begin ----------


```
{
  "direction": "uplink",
  "sourceTime": 1647255796566,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary": "090025860003"
  },
  "message": {
    "messageType": "SHUTDOWN",
    "trackingMode": "STAND_BY",
    "batteryLevel": 37,
    "batteryStatus": "OPERATING",
    "ackToken": 0,
    "periodicPosition": false,
    "temperatureMeasure": 23.8,
    "sosFlag": 0,
    "appState": 0,
    "dynamicMotionState": "STATIC",
    "onDemand": false,
    "shutdownCause": "BLE_REQUEST",
    "payload": "090025860003",
    "deviceConfiguration": {
      "mode": "STAND_BY"
    }
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
```

```
    "points": {
      "messageType": {
        "type": "string",
        "record": "SHUTDOWN"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "STATIC"
      },
      "appState": {
        "type": "int64",
        "record": 0
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
      "onDemand": {
        "type": "boolean",
        "record": false
      },
      "ackToken": {
        "type": "int64",
        "record": 0
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "STAND_BY"
      },
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
        "record": 37
      },
      "temperature": {
        "unitId": "Cel",
```

```
        "type": "double",
        "record": 23.8
      },
      "deviceMode": {
        "type": "string",
        "record": "STAND_BY"
      },
      "shutdownCause": {
        "type": "string",
        "record": "BLE_REQUEST"
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.7. Event message

5.2.7.1. BLE Connected Event

This example shows the decoding of an uplink event message.
The LoRa payload to be parsed is the following: "0a2430864003"
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary": "0a2430864003"
  }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
{
  "direction": "uplink",
  "sourceTime": 1647256071337,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary": "0a2430864003"
  },
  "message": {
    "messageType": "EVENT",
    "trackingMode": "MOTION_TRACKING",
    "batteryLevel": 48,
    "batteryStatus": "OPERATING",
    "ackToken": 4,
    "periodicPosition": false,
    "temperatureMeasure": 23.8,
    "sosFlag": 0,
    "appState": 0,
    "dynamicMotionState": "MOVING",
    "onDemand": false,
    "payload": "0a2430864003",
    "deviceConfiguration": {
      "mode": "MOTION_TRACKING"
    },
    "eventType": "BLE_CONNECTED"
  },
  "thing": {
```

```
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "EVENT"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "MOVING"
      },
      "appState": {
        "type": "int64",
        "record": 0
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
      "onDemand": {
        "type": "boolean",
        "record": false
      },
      "ackToken": {
        "type": "int64",
        "record": 4
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "MOTION_TRACKING"
      },
      "batteryStatus": {
```

```
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
        "record": 48
      },
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 23.8
      },
      "deviceMode": {
        "type": "string",
        "record": "MOTION_TRACKING"
      },
      "eventType": {
        "type": "string",
        "record": "BLE_CONNECTED"
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.7.2. Temperature Alert Event

This example shows the decoding of a TEMPERATURE_ALERT event uplink message:
“0a00306c000502876c00000001”
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
```

```
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary": "0a00306c000502876c00000001"
  }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
{
  "direction": "uplink",
  "sourceTime": 1647256132687,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary": "0a00306c000502876c00000001"
  },
  "message": {
    "messageType": "EVENT",
    "trackingMode": "STAND_BY",
    "batteryLevel": 48,
    "batteryStatus": "OPERATING",
    "ackToken": 0,
    "periodicPosition": false,
    "temperatureMeasure": 10.6,
    "sosFlag": 0,
    "appState": 0,
    "dynamicMotionState": "STATIC",
    "onDemand": false,
```

```
    "payload": "0a00306c000502876c00000001",
    "deviceConfiguration": {
      "mode": "STAND_BY"
    },
    "eventType": "TEMPERATURE_ALERT",
    "measuredTemperature": {
      "state": "LOW",
      "max": 24.3,
      "min": 10.6,
      "highCounter": 0,
      "lowCounter": 1
    }
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "EVENT"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "STATIC"
      },
      "appState": {
        "type": "int64",
        "record": 0
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
      "onDemand": {
        "type": "boolean",
        "record": false
```

```
      },
      "ackToken": {
        "type": "int64",
        "record": 0
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "STAND_BY"
      },
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
        "record": 48
      },
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 10.6
      },
      "deviceMode": {
        "type": "string",
        "record": "STAND_BY"
      },
      "eventType": {
        "type": "string",
        "record": "TEMPERATURE_ALERT"
      },
      "temperatureState": {
        "type": "string",
        "record": "LOW"
      },
      "maxTemperature": {
        "unitId": "Cel",
        "type": "double",
        "standardNaming": "unsupported",
        "record": 24.3
```

```
      },
      "minTemperature": {
        "unitId": "Cel",
        "type": "double",
        "standardNaming": "unsupported",
        "record": 10.6
      },
      "highTemperatureCounter": {
        "type": "int64",
        "record": 0
      },
      "lowTemperatureCounter": {
        "type": "int64",
        "record": 1
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.7.3. Angle detection Event

This example shows the decoding of an ANGLE_DETECTION event uplink message:
“0a2c646c0009420258fffffc10001502d6fd80ffff31”
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"

  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
  },
  "raw": {
    "binary": "0a2c646c0009420258fffffc10001502d6fd80ffff31"
  }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------


```
{
  "direction": "uplink",
  "sourceTime": 1647256185789,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary": "0a2c646c0009420258fffffc10001502d6fd80ffff31"
  },
  "message": {
    "messageType": "EVENT",
    "trackingMode": "MOTION_TRACKING",
    "batteryLevel": 100,
    "batteryStatus": "OPERATING",
    "ackToken": 0,
    "periodicPosition": false,
    "temperatureMeasure": 10.6,
    "sosFlag": 0,
    "appState": 1,
    "dynamicMotionState": "MOVING",
    "onDemand": false,
    "payload": "0a2c646c0009420258fffffc10001502d6fd80ffff31",
    "deviceConfiguration": {
      "mode": "MOTION_TRACKING"
    },
    "eventType": "ANGLE_DETECTION",
    "angleDetection": {
      "flags": {
        "transition": "NORMAL_TO_CRITICAL",
        "triggerType": "CRITICAL_ANGLE_REPORTING",
        "notificationIdentifier": 2
      },
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
      "age": 600,
      "referenceVector": [
        -1,
        -1008,
        21
      ],
      "criticalVector": [
        726,
        -640,
        -1
      ],
      "angle": 49
    }
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "EVENT"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "MOVING"
      },
      "appState": {
        "type": "int64",
        "record": 1
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
      "onDemand": {
        "type": "boolean",
        "record": false
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
      },
      "ackToken": {
        "type": "int64",
        "record": 0
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "MOTION_TRACKING"
      },
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
        "record": 100
      },
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 10.6
      },
      "deviceMode": {
        "type": "string",
        "record": "MOTION_TRACKING"
      },
      "eventType": {
        "type": "string",
        "record": "ANGLE_DETECTION"
      },
      "angle": {
        "unitId": "deg",
        "type": "double",
        "record": 49
      },
      "angleAge": {
        "unitId": "s",
        "type": "double",
        "standardNaming": "unsupported",
        "record": 600
```

```
      },
      "angleTriggerType": {
        "type": "string",
        "record": "CRITICAL_ANGLE_REPORTING"
      },
      "angleTransition": {
        "type": "string",
        "record": "NORMAL_TO_CRITICAL"
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.7.4. Geofencing notification Event

This example shows the decoding of a GEOFENCING event uplink message: “0a04538a000a01b5dffb”
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary": "0a04538a000a01b5dffb"
  }
}'
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------


```
{
  "direction": "uplink",
  "sourceTime": 1647256253334,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary": "0a04538a000a01b5dffb"
  },
  "message": {
    "messageType": "EVENT",
    "trackingMode": "STAND_BY",
    "batteryLevel": 83,
    "batteryStatus": "OPERATING",
    "ackToken": 0,
    "periodicPosition": false,
    "temperatureMeasure": 25.8,
    "sosFlag": 0,
    "appState": 0,
    "dynamicMotionState": "MOVING",
    "onDemand": false,
    "payload": "0a04538a000a01b5dffb",
    "deviceConfiguration": {
      "mode": "STAND_BY"
    },
    "eventType": "GEOFENCING",
    "geofencingNotification": {
      "geofencingFormat": 0,
      "geofencingType": "ENTRY",
      "id": "0xb5dffb"
    }
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "EVENT"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "MOVING"
      },
      "appState": {
        "type": "int64",
        "record": 0
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
      "onDemand": {
        "type": "boolean",
        "record": false
      },
      "ackToken": {
        "type": "int64",
        "record": 0
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "STAND_BY"
      },
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
```

```
        "record": 83
      },
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 25.8
      },
      "deviceMode": {
        "type": "string",
        "record": "STAND_BY"
      },
      "eventType": {
        "type": "string",
        "record": "GEOFENCING"
      },
      "geofencingFormat": {
        "type": "int64",
        "record": 0
      },
      "geofencingType": {
        "type": "string",
        "record": "ENTRY"
      },
      "geofencingId": {
        "type": "string",
        "record": "0xb5dffb"
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.8. Data scan collection message

This example shows the decoding of an uplink data scan collection message.
The LoRa payload to be parsed is the following:
"0b01648600c00048c66eaa1b1aec11c34fc9cdb8d309c363d8bab2432ac2581f7c370a48"
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
```

```
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
   "driverId": "abeeway:asset-tracker:3"
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary":
"0b01648600c00048c66eaa1b1aec11c34fc9cdb8d309c363d8bab2432ac2581f7c370a48"
  }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
{
  "direction": "uplink",
  "sourceTime": 1647256289762,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary":
"0b01648600c00048c66eaa1b1aec11c34fc9cdb8d309c363d8bab2432ac2581f7c370a48"
  },
  "message": {
    "messageType": "DATA_SCAN_COLLECTION",
    "trackingMode": "STAND_BY",
    "batteryLevel": 100,
    "batteryStatus": "OPERATING",
```

```
    "ackToken": 0,
    "periodicPosition": false,
    "temperatureMeasure": 23.8,
    "sosFlag": 0,
    "appState": 0,
    "dynamicMotionState": "STATIC",
    "onDemand": true,
    "payload":
"0b01648600c00048c66eaa1b1aec11c34fc9cdb8d309c363d8bab2432ac2581f7c370a48"
,
    "deviceConfiguration": {
      "mode": "STAND_BY"
    },
    "dataScanCollection": {
      "scanType": "BLE_BEACONS",
      "again": true,
      "dataFormat": "MAC_ADDRESS",
      "fragmentIdentification": 0,
      "collectionIdentifier": 0,
      "hash": 72,
      "macAddressData": [
        {
          "bssid": "6e:aa:1b:1a:ec:11",
          "rssi": -58
        },
        {
          "bssid": "4f:c9:cd:b8:d3:09",
          "rssi": -61
        },
        {
          "bssid": "63:d8:ba:b2:43:2a",
          "rssi": -61
        },
        {
          "bssid": "58:1f:7c:37:0a:48",
          "rssi": -62
        }
      ]
    }
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "DATA_SCAN_COLLECTION"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "STATIC"
      },
      "appState": {
        "type": "int64",
        "record": 0
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
      "onDemand": {
        "type": "boolean",
        "record": true
      },
      "ackToken": {
        "type": "int64",
        "record": 0
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "STAND_BY"
      },
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
```

```
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
        "record": 100
      },
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 23.8
      },
      "deviceMode": {
        "type": "string",
        "record": "STAND_BY"
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.9. Proximity notification message

This example shows the decoding of an uplink proximity notification message.
The LoRa payload to be parsed is the following:
"0c0164860000cafebabedeadbeefcafebabedeadbeef000a00200020cafebabe233100233122"
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
```

```
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary":
"0c046289000000000000000000020635f01e10000b7000a000a002c000000000002001de
56a

}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
{
  "direction": "uplink",
  "sourceTime": 1647256350069,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary":
"0c046289000000000000000000020635f01e10000b7000a000a002c000000000002001de
56a"
  },
  "message": {
    "messageType": "PROXIMITY_DETECTION",
    "trackingMode": "STAND_BY",
    "batteryLevel": 98,
    "batteryStatus": "OPERATING",
    "ackToken": 0,
    "periodicPosition": false,
    "temperatureMeasure": 25.3,
    "sosFlag": 0,
    "appState": 0,
    "dynamicMotionState": "MOVING",
    "onDemand": false,
    "payload":
"0c046289000000000000000000020635f01e10000b7000a000a002c000000000002001de
56a",
```

```
    "deviceConfiguration": {
      "mode": "STAND_BY"
    },
    "proximityNotification": {
      "notificationType": "WARNING_MESSAGE",
      "encrypted": false,
      "recordAction": "RECORD_START",
      "rollingProximityIdentifier": "00000000000000020635f01e10000b7",
      "closestDistanceRecording": 1,
      "distanceAverageRecorded": 1,
      "cumulatedExposure": 44,
      "metadata": "00000000",
      "cumulatedContactDuration": 2,
      "currentDailyExposure": 1959274
    }
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "PROXIMITY_DETECTION"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "MOVING"
      },
      "appState": {
        "type": "int64",
        "record": 0
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
```

```
      "onDemand": {
        "type": "boolean",
        "record": false
      },
      "ackToken": {
        "type": "int64",
        "record": 0
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "STAND_BY"
      },
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
        "record": 98
      },
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 25.3
      },
      "deviceMode": {
        "type": "string",
        "record": "STAND_BY"
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.10. Proximity daily report message

This example shows the decoding of an uplink proximity daily report message.

The LoRa payload to be parsed is the following:
"0c48628d2003000000000000000300001f2400000000000000000000000000000000000000000000000000"
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary":
"0c48628d2003000000000000000300001f2400000000000000000000000000000000000000000000000000"
  }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
{
  "direction": "uplink",
  "sourceTime": 1647256384760,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
```

```
  "raw": {
    "binary":
"0c48628d200300000000000000300001f2400000000000000000000000000000000000000000000000"
  },
  "message": {
    "messageType": "PROXIMITY_DETECTION",
    "trackingMode": "PERMANENT_TRACKING",
    "batteryLevel": 98,
    "batteryStatus": "OPERATING",
    "ackToken": 2,
    "periodicPosition": false,
    "temperatureMeasure": 27.3,
    "sosFlag": 0,
    "appState": 1,
    "dynamicMotionState": "STATIC",
    "onDemand": false,
    "payload":
"0c48628d200300000000000000300001f2400000000000000000000000000000000000000000000000",
    "deviceConfiguration": {
      "mode": "PERMANENT_TRACKING"
    },
    "proximityDailyReport": {
      "dailyAlertDay0": 0,
      "dailyWarningDay0": 3,
      "dailyExposureDay0": 7972,
      "dailyAlertDay1": 0,
      "dailyWarningDay1": 0,
      "dailyExposureDay1": 0,
      "dailyAlertDay2": 0,
      "dailyWarningDay2": 0,
      "dailyExposureDay2": 0
    }
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
```

```
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "PROXIMITY_DETECTION"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "STATIC"
      },
      "appState": {
        "type": "int64",
        "record": 1
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
      "onDemand": {
        "type": "boolean",
        "record": false
      },
      "ackToken": {
        "type": "int64",
        "record": 2
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "PERMANENT_TRACKING"
      },
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
        "record": 98
      },
```

```
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 27.3
      },
      "deviceMode": {
        "type": "string",
        "record": "PERMANENT_TRACKING"
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.11. Proximity white listing message

This example shows the decoding of an uplink proximity white listing message.
The LoRa payload to be parsed is the following:
"0c4864902005000000000000000020635f017100000d00"
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
    "binary": "0c4864902005000000000000000020635f017100000d00"
  }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------


```
{
  "direction": "uplink",
  "sourceTime": 1647256897679,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary": "0c4864902005000000000000000020635f017100000d00"
  },
  "message": {
    "messageType": "PROXIMITY_DETECTION",
    "trackingMode": "PERMANENT_TRACKING",
    "batteryLevel": 100,
    "batteryStatus": "OPERATING",
    "ackToken": 2,
    "periodicPosition": false,
    "temperatureMeasure": 28.8,
    "sosFlag": 0,
    "appState": 1,
    "dynamicMotionState": "STATIC",
    "onDemand": false,
    "payload": "0c4864902005000000000000000020635f017100000d00",
    "deviceConfiguration": {
      "mode": "PERMANENT_TRACKING"
    },
    "proximityWhiteListing": {
      "encrypted": false,
      "rollingProximityIdentifier": "00000000000000020635f017100000d",
      "list": "PEER_LIST",
      "recordStatus": "NOT_WHITE_LISTED",
      "solicited": true
    }
  },
```

```
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "PROXIMITY_DETECTION"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "STATIC"
      },
      "appState": {
        "type": "int64",
        "record": 1
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
      "onDemand": {
        "type": "boolean",
        "record": false
      },
      "ackToken": {
        "type": "int64",
        "record": 2
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "PERMANENT_TRACKING"
      },
```

```
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
        "record": 100
      },
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 28.8
      },
      "deviceMode": {
        "type": "string",
        "record": "PERMANENT_TRACKING"
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.12. Proximity daily response message

This example shows the decoding of an uplink proximity daily response message.
The LoRa payload to be parsed is the following: "0c486490100600000000000000000000015fa"
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
```

```
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary": "0c48649010060000000000000000000015fa"
  }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
{
  "direction": "uplink",
  "sourceTime": 1647256923560,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary": "0c48649010060000000000000000000015fa"
  },
  "message": {
    "messageType": "PROXIMITY_DETECTION",
    "trackingMode": "PERMANENT_TRACKING",
    "batteryLevel": 100,
    "batteryStatus": "OPERATING",
    "ackToken": 1,
    "periodicPosition": false,
    "temperatureMeasure": 28.8,
    "sosFlag": 0,
    "appState": 1,
    "dynamicMotionState": "STATIC",
    "onDemand": false,
    "payload": "0c48649010060000000000000000000015fa",
    "deviceConfiguration": {
      "mode": "PERMANENT_TRACKING"
    },
```

```
    "proximityDailyResponse": {
      "dayIdentifier": 0,
      "dailyAlert": 0,
      "dailyWarning": 0,
      "dailyExposure": 5626
    }
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "PROXIMITY_DETECTION"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "STATIC"
      },
      "appState": {
        "type": "int64",
        "record": 1
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
      "onDemand": {
        "type": "boolean",
        "record": false
      },
      "ackToken": {
        "type": "int64",
        "record": 1
      },
      "periodicPosition": {
```

```
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "PERMANENT_TRACKING"
      },
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
        "record": 100
      },
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 28.8
      },
      "deviceMode": {
        "type": "string",
        "record": "PERMANENT_TRACKING"
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.13. SMS message

This example shows the decoding of an uplink SMS message.
The LoRa payload to be parsed is the following: "0d0000001c68656c6c6f"
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary": "0d0000001c68656c6c6f"
  }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
{
  "direction":"uplink",
  "sourceTime":1647256923560,
  "meta":{
    "lorawan":{
      "fPort":18
    },
    "driverId":"abeeway:asset-tracker:3"
  },
  "raw":{
    "binary":"0d0000001c68656c6c6f"
  },
  "message":{
    "messageType":"SMS",
    "ackToken":0,
    "payload":"0d0000001c68656c6c6f",
    "sms":{
      "destinationId":28,
      "message":"hello"
    }
  },
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
  "thing":{
    "model":{
      "producerId":"abeeway",
      "moduleId":"micro-tracker",
      "version":"3"
    },
    "application":{
      "producerId":"abeeway",
      "moduleId":"asset-tracker",
      "version":"2"
    },
    "points":{
      "messageType":{
        "type":"string",
        "record":"SMS"
      },
      "ackToken":{
        "type":"int64",
        "record":0
      },
      "smsDestinationId":{
        "type":"int64",
        "record":28
      },
      "smsMessage":{
        "type":"string",
        "record":"hello"
      }
    }
  }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

5.2.14. Debug message

This example shows the decoding of an uplink debug message.
The LoRa payload to be parsed is the following: "0f0030860001434F4445204552524F52"
The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/decode \
  -H 'Content-Type: application/json' \
  -d '{
  "direction": "uplink",
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  },
  "raw": {
    "binary": "0f0030860001434f4445204552524f52"
  }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
{
  "direction": "uplink",
  "sourceTime": 1647256979072,
  "meta": {
    "lorawan": {
      "fPort": 18
    },
    "driverId": "abeeway:asset-tracker:3"
  },
  "raw": {
    "binary": "0f0030860001434f4445204552524f52"
  },
  "message": {
    "messageType": "DEBUG",
    "trackingMode": "STAND_BY",
    "batteryLevel": 48,
    "batteryStatus": "OPERATING",
```

```
    "ackToken": 0,
    "periodicPosition": false,
    "temperatureMeasure": 23.8,
    "sosFlag": 0,
    "appState": 0,
    "dynamicMotionState": "STATIC",
    "onDemand": false,
    "debugErrorCode": 1,
    "payload": "0f0030860001434f4445204552524f52",
    "debugCrashInfo": "CODE ERROR",
    "deviceConfiguration": {
      "mode": "STAND_BY"
    },
    "debugCommandTag": "DEBUG_CRASH_INFORMATION"
  },
  "thing": {
    "model": {
      "producerId": "abeeway",
      "moduleId": "micro-tracker",
      "version": "3"
    },
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    },
    "points": {
      "messageType": {
        "type": "string",
        "record": "DEBUG"
      },
      "dynamicMotionState": {
        "type": "string",
        "record": "STATIC"
      },
      "appState": {
        "type": "int64",
        "record": 0
      },
      "sosFlag": {
        "type": "int64",
        "record": 0
      },
      "onDemand": {
```

```
        "type": "boolean",
        "record": false
      },
      "ackToken": {
        "type": "int64",
        "record": 0
      },
      "periodicPosition": {
        "type": "boolean",
        "record": false
      },
      "trackingMode": {
        "type": "string",
        "record": "STAND_BY"
      },
      "batteryStatus": {
        "type": "string",
        "record": "OPERATING"
      },
      "batteryLevel": {
        "unitId": "%",
        "type": "double",
        "record": 48
      },
      "temperature": {
        "unitId": "Cel",
        "type": "double",
        "record": 23.8
      },
      "deviceMode": {
        "type": "string",
        "record": "STAND_BY"
      },
      "debugCommandTag": {
        "type": "string",
        "record": "DEBUG_CRASH_INFORMATION"
      },
      "debugCrashInfo": {
        "type": "string",
        "record": "CODE ERROR"
      },
      "debugErrorCode": {
        "type": "int64",
        "record": 1
```

```
        }
      }
    }
}
```

---------- Snippet End ----------

Where `message` is the decoded payload.

6. Decoding multiple payloads

The locally hosted docker image allows
This example shows the decoding of an uplink position message.
The LoRa payload to be parsed is the following: "0700649250024C0000012C " and
“032c36844926c8675e842cd7c76cc217d827bebc34dbfdad0e31b4c8675e8200d4b2”

The LoRa transmission port number is 18.

---------- Snippet begin ----------

```
curl -L -X POST ' http://localhost:8095/v1/decode/batch ' -H 'Content-
Type: application/json' -d
'[
    {
        "id": "correlationId1",
        "input": {
            "direction": "uplink",
            "meta": {
                "lorawan": {
                    "fPort": 18
                }
            },
            "thing": {
                "application": {
                    "producerId": "abeeway",
                    "moduleId": "asset-tracker",
                    "version": "2"
                }
            },
            "raw": {
                "binary": "0700649250024C0000012C"
            }
        }
    },
    {
        "id": "correlationId2",
        "input": {
            "direction": "uplink",
            "meta": {
                "lorawan": {
                    "fPort": 18
                }
            },
            "thing": {
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
                "application": {
                    "producerId": "abeeway",
                    "moduleId": "asset-tracker",
                    "version": "2"
                }
            },
            "raw": {
                "binary":
"032c36844926c8675e842cd7c76cc217d827bebc34dbfdad0e31b4c8675e8200d4b2"
            }
        }
    }
]'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
[
    {
        "id": "correlationId1",
        "output": {
            "direction": "uplink",
            "sourceTime": "2022-03-14T11:25:24.654Z",
            "meta": {
                "lorawan": {
                    "fPort": 18
                },
                "driverId": "abeeway:asset-tracker:3"
            },
            "raw": {
                "binary": "0700649250024C0000012C"
            },
            "message": {
                "messageType": "CONFIGURATION",
                "trackingMode": "STAND_BY",
                "batteryLevel": 100,
                "batteryStatus": "OPERATING",
                "ackToken": 5,
                "periodicPosition": false,
                "temperatureMeasure": 29.9,
                "miscDataTag": "DEVICE_CONFIGURATION",
                "sosFlag": 0,
```

```
                "appState": 0,
                "dynamicMotionState": "STATIC",
                "onDemand": false,
                "payload": "0700649250024c0000012c",
                "deviceConfiguration": {
                    "proximityMaxUpdate": 300,
                    "mode": "STAND_BY"
                }
            },
            "thing": {
                "points": {
                    "messageType": {
                        "record": "CONFIGURATION",
                        "type": "string"
                    },
                    "dynamicMotionState": {
                        "record": "STATIC",
                        "type": "string"
                    },
                    "appState": {
                        "record": 0,
                        "type": "int64"
                    },
                    "sosFlag": {
                        "record": 0,
                        "type": "int64"
                    },
                    "onDemand": {
                        "record": false,
                        "type": "boolean"
                    },
                    "ackToken": {
                        "record": 5,
                        "type": "int64"
                    },
                    "periodicPosition": {
                        "record": false,
                        "type": "boolean"
                    },
                    "trackingMode": {
                        "record": "STAND_BY",
                        "type": "string"
                    },
                    "batteryStatus": {
```

```
                        "record": "OPERATING",
                        "type": "string"
                    },
                    "batteryLevel": {
                        "record": 100,
                        "type": "double",
                        "unitId": "%"
                    },
                    "temperature": {
                        "record": 29.9,
                        "type": "double",
                        "unitId": "Cel"
                    },
                    "deviceMode": {
                        "record": "STAND_BY",
                        "type": "string"
                    },
                    "miscDataTag": {
                        "record": "DEVICE_CONFIGURATION",
                        "type": "string"
                    }
                },
                "application": {
                    "producerId": "abeeway",
                    "moduleId": "asset-tracker",
                    "version": "2"
                }
            }
        }
    },
    {
        "id": "correlationId2",
        "output": {
            "direction": "uplink",
            "sourceTime": "2022-03-14T11:25:24.654Z",
            "meta": {
                "lorawan": {
                    "fPort": 18
                },
                "driverId": "abeeway:asset-tracker:3"
            },
            "raw": {
                "binary":
"032c36844926c8675e842cd7c76cc217d827bebc34dbfdad0e31b4c8675e8200d4b2"
```

```
            },
            "message": {
                "messageType": "POSITION_MESSAGE",
                "age": 304,
                "trackingMode": "MOTION_TRACKING",
                "batteryLevel": 54,
                "batteryStatus": "OPERATING",
                "ackToken": 4,
                "rawPositionType": "WIFI_BSSIDS_WITH_NO_CYPHER",
                "periodicPosition": false,
                "temperatureMeasure": 22.8,
                "sosFlag": 0,
                "appState": 1,
                "dynamicMotionState": "MOVING",
                "onDemand": false,
                "payload":
"032c36844926c8675e842cd7c76cc217d827bebc34dbfdad0e31b4c8675e8200d4b2",
                "deviceConfiguration": {
                    "mode": "MOTION_TRACKING"
                },
                "wifiBssids": [
                    {
                        "bssid": "c8:67:5e:84:2c:d7",
                        "rssi": -57
                    },
                    {
                        "bssid": "6c:c2:17:d8:27:be",
                        "rssi": -68
                    },
                    {
                        "bssid": "34:db:fd:ad:0e:31",
                        "rssi": -76
                    },
                    {
                        "bssid": "c8:67:5e:82:00:d4",
                        "rssi": -78
                    }
                ]
            },
            "thing": {
                "points": {
                    "messageType": {
                        "record": "POSITION_MESSAGE",
                        "type": "string"
```

```
                    },
                    "dynamicMotionState": {
                        "record": "MOVING",
                        "type": "string"
                    },
                    "appState": {
                        "record": 1,
                        "type": "int64"
                    },
                    "sosFlag": {
                        "record": 0,
                        "type": "int64"
                    },
                    "onDemand": {
                        "record": false,
                        "type": "boolean"
                    },
                    "ackToken": {
                        "record": 4,
                        "type": "int64"
                    },
                    "periodicPosition": {
                        "record": false,
                        "type": "boolean"
                    },
                    "trackingMode": {
                        "record": "MOTION_TRACKING",
                        "type": "string"
                    },
                    "batteryStatus": {
                        "record": "OPERATING",
                        "type": "string"
                    },
                    "batteryLevel": {
                        "record": 54,
                        "type": "double",
                        "unitId": "%"
                    },
                    "temperature": {
                        "record": 22.8,
                        "type": "double",
                        "unitId": "Cel"
                    },
                    "deviceMode": {
```

```
                        "record": "MOTION_TRACKING",
                        "type": "string"
                    },
                    "age": {
                        "record": 304,
                        "type": "double",
                        "unitId": "s"
                    },
                    "rawPositionType": {
                        "record": "WIFI_BSSIDS_WITH_NO_CYPHER",
                        "type": "string"
                    }
                },
                "application": {
                    "producerId": "abeeway",
                    "moduleId": "asset-tracker",
                    "version": "2"
                }
            }
        }
    }
]
```

---------- Snippet end ----------

Where `message` is the decoded payload.

7. Encoding a downlink

7.1. Overview

The encoder takes a JSON format String and converts it into the downlink hexadecimal payload.

Each downlink message should contain the following fields:

- *DownMessageType*: Type of downlink messages (string format). Acceptable values are:
  - *POS_ON_DEMAND*: Request a position
  - *SET_MODE*: Change the operation mode of a tracker.
  - *START_SOS*: Start the SOS mode.
  - *STOP_SOS*: Stop the SOS mode.
  - *DEBUG_COMMAND*: Send debug commands
  - *SET_PARAM*: Modify tracker's parameter(s).
  - *REQUEST_CONFIG*: Read the tracker configuration.
  - *REQUEST_TEMPERATURE_STATUS*: request a temperature status
  - *REQUEST_STATUS*: request a status message (energy status or health status)
  - *PROXIMITY*: Send a proximity request message
  - *ANGLE_DETECTION*: Control angle detection
  - *CLEAR_MOTION_PERCENTAGE*: Clear the motion percentage
- *AckToken*: Random integer ranging from 0 to 15 and used by the tracker to acknowledge the downlink message. Refer to the abeeway-reference-guide-fw1.8.pdf document for more details on setting/processing this field.

Notes:

- Each field name should be coded as a string.
- Extra fields should be provided according to the downlink message type.
- **All types of messages are supported since AssetTracker V1.7 except REQUEST_TEMPERATURE_STATUS and CLEAR_MOTION_DETECTION. However, some parameters are new since v1.8 or v1.9 or v2.1 or v2.2 or v2.3 in SET_PARAM, REQUEST_CONFIG and DEBUG_COMMAND messages**

The following sections detail each downlink message types and their required fields.

7.2. Request a position

This message requests a position to the tracker. The request does not require extra fields.

7.2.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "POS_ON_DEMAND",
        "ackToken": 10
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2019-06-07T14:28:52.016Z",
    "meta": {
        "driverId": "abeeway:asset-tracker:2"
    },
    "raw": {
        "binary": "010a"
    },
    "message": {
        "downMessageType": "POS_ON_DEMAND",
        "ackToken": 10
    },
    "thing": {
        "application": {
            "producerId": "abeeway",
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
            "moduleId": "asset-tracker",
            "version": "2"
        }
    }
}
```

---------- Snippet end ----------

Where `"010a"` in `"raw"` is the encoded payload

7.3. Operational mode change

This message changes the tracker operational mode. The request requires an extra field called *ModeValue*, which can take the following string values:

- *STAND_BY*: Set the tracker in standby.
- *MOTION_TRACKING*: Set the tracker in motion tracking.
- *PERMANENT_TRACKING*: Set the tracker in permanent tracking.
- *MOTION_START_END_TRACKING*: Set the tracker in start/end mode
- *ACTIVITY_TRACKING*: Set the tracker in activity tracking mode.
- *OFF*: Set the tracker in off mode. Note that the master tracker will interpret this mode as *STANDBY* (since it cannot be woke up using the button).

7.3.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "SET_MODE",
        "modeValue": "MOTION_TRACKING",
        "ackToken": 3
    }
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2019-06-07T14:31:07.453Z",
    "meta": {
        "driverId": "abeeway:asset-tracker:2"
    },
    "raw": {
        "binary": "020301"
    },
     "message": {
        "downMessageType": "SET_MODE",
        "modeValue": "MOTION_TRACKING",
        "ackToken": 3
    },
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    }
}
```

---------- Snippet end ----------

Where `"020301"` is the encoded payload

7.4. SOS start

This message turns on the SOS mode of the tracker. The request does not require extra fields.

7.4.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
```

```
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "START_SOS",
        "ackToken": 2
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2019-06-07T14:35:38.363Z",
    "meta": {
        "driverId": "abeeway:asset-tracker:2"
    },
    "raw": {
        "binary": "0402"
    },
    "message": {
        "downMessageType": "START_SOS",
        "ackToken": 2
    },
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    }
}
```

---------- Snippet end ----------

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

Where `"0402"` is the encoded payload

7.5. SOS stop

This message turns off the SOS mode of the tracker. The request does not require extra fields.

7.5.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "STOP_SOS",
        "ackToken": 3
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2019-06-07T14:36:43.727Z",
    "meta": {
        "driverId": "abeeway:asset-tracker:2"
    },
    "raw": {
        "binary": "0503"
    },
    "message": {
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
        "downMessageType": "STOP_SOS",
        "ackToken": 3
    },
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    }
}
```

---------- Snippet end ----------

Where `"0503"` is the encoded payload

7.6. Request temperature status

This message requests the temperature alert status of the device. The request does not require extra fields.

7.6.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "REQUEST_TEMPERATURE_STATUS
        "ackToken

}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

The response is:

---------- Snippet begin ----------

```
{
  "direction": "downlink",
  "sourceTime": "2020-07-15T11:14:32.169Z",
  "meta": {
    "driverId": "abeeway:asset-tracker:2"
  },
  "raw": {
    "binary": "0603"
  },
  "message": {
    "downMessageType": "REQUEST_TEMPERATURE_STATUS",
    "ackToken": 3
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  }
}
```

---------- Snippet end ----------

Where `"0603"` is the encoded payload

7.7. Proximity messages

The proximity requests require an extra object called *proximityMessage*.

7.7.1. Request proximity record status

This message requests the status of the record matching the Rolling Proximity Identifier (RPI).
The *ProximityMessage* object requires two fields:

- *type: "GET_RECORD_STATUS"*
- *rollingProximityIdentifier:* Rolling proximity identifier hexadecimal string.

7.7.1.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "PROXIMITY",
        "proximityMessage": {
            "type": "GET_RECORD_STATUS",
            "rollingProximityIdentifier":
"0000000000000000020635f017100000d"
        },
        "ackToken": 3
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
  "direction": "downlink",
  "sourceTime": "2020-07-15T11:25:33.939Z",
  "meta": {
    "driverId": "abeeway:asset-tracker:2"
  },
  "raw": {
    "binary": "0703000000000000000000020635f017100000d"
  },
  "message": {
    "downMessageType": "PROXIMITY",
    "proximityMessage": {
      "type": "GET_RECORD_STATUS",
      "rollingProximityIdentifier": "0000000000000000020635f017100000d"
    },
    "ackToken": 3
```

```
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  }
}
```

---------- Snippet end ----------

Where `"07030000000000000000020635f017100000d"` is the encoded payload

7.7.2. Set proximity record white list status

This message sets the status of the record matching the Rolling Proximity Identifier (RPI).
The *ProximityMessage* object requires three fields:

- *type: SET_WHITE_LIST_STATUS*
- *rollingProximityIdentifier:* Rolling proximity identifier hexadecimal string.
- *recordStatus: RESET_WHITE_LISTING* or *SET_WHITE_LISTING*

7.7.2.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "PROXIMITY",
        "proximityMessage": {
            "type": "SET_WHITE_LIST_STATUS",
            "rollingProximityIdentifier":
"00000000000000020635f017100000d",
            "recordStatus": "SET_WHITE_LISTING"
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
        },
        "ackToken": 3
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
  "direction": "downlink",
  "sourceTime": "2020-07-15T11:42:18.057Z",
  "meta": {
    "driverId": "abeeway:asset-tracker:2"
  },
  "raw": {
    "binary": "07030100000000000000020635f017100000d01"
  },
  "message": {
    "downMessageType": "PROXIMITY",
    "proximityMessage": {
      "type": "SET_WHITE_LIST_STATUS",
      "rollingProximityIdentifier": "00000000000000020635f017100000d",
      "recordStatus": "SET_WHITE_LISTING"
    },
    "ackToken": 3
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  }
}
```

---------- Snippet end ----------

Where `"07030100000000000000020635f017100000d01"` is the encoded payload

7.7.3. Get proximity daily information

This message requests the proximity daily information of the tracker.

The *ProximityMessage* object requires three fields:

- *type: GET_DAILY_INFORMATION*
- *dayIdentifier:* Day for which we are requesting the information.

7.7.3.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "PROXIMITY",
        "proximityMessage": {
            "type": "GET_DAILY_INFORMATION",
            "dayIdentifier": 2
        },
        "ackToken": 3
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
  "direction": "downlink",
  "sourceTime": "2020-07-15T11:46:50.754Z",
  "meta": {
    "driverId": "abeeway:asset-tracker:2"
  },
  "raw": {
    "binary": "07030202"
  },
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
  "message": {
    "downMessageType": "PROXIMITY",
    "proximityMessage": {
      "type": "GET_DAILY_INFORMATION",
      "dayIdentifier": 2
    },
    "ackToken": 3
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  }
}
```

---------- Snippet end ----------

Where `"07030202"` is the encoded payload

7.7.4. Clear proximity daily information

This message clears the proximity daily information of the tracker.
The *ProximityMessage* object requires three fields:

- *type: CLEAR_DAILY_INFORMATION*
- *dayIdentifier:* Day for which we are requesting the information.

7.7.4.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
        "downMessageType": "PROXIMITY",
        "proximityMessage": {
            "type": "CLEAR_DAILY_INFORMATION",
            "dayIdentifier": 2
        },
        "ackToken": 3
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
  "direction": "downlink",
  "sourceTime": "2020-07-15T11:49:48.896Z",
  "meta": {
    "driverId": "abeeway:asset-tracker:2"
  },
  "raw": {
    "binary": "07030302"
  },
  "message": {
    "downMessageType": "PROXIMITY",
    "proximityMessage": {
      "type": "CLEAR_DAILY_INFORMATION",
      "dayIdentifier": 2
    },
    "ackToken": 3
  },
  "thing": {
    "application": {
      "producerId": "abeeway",
      "moduleId": "asset-tracker",
      "version": "2"
    }
  }
}
```

---------- Snippet end ----------

Where `"07030302"` is the encoded payload

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

7.8. Angle detection control

This message is used to stop or start the angle detection. The request requires an extra field called angleDetectionControl, which can take the following string values:

- STOP_ANGLE_DETECTION: Stop angle detection and reset the reference vector.
- START_ANGLE_DETECTION: Start angle detection and relearn.

7.8.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
       "downMessageType": "ANGLE_DETECTION",
       "angleDetectionControl": "STOP_ANGLE_DETECTION",
       "ackToken": 3
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2021-03-25T10:12:02.520Z",
    "meta": {
        "lorawan": {
            "fPort": 2
        }
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
    },
    "thing": {
        "points": {},
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "raw": {
        "binary": "080300"
    },
    "message": {
        "downMessageType": "ANGLE_DETECTION",
        "angleDetectionControl": "STOP_ANGLE_DETECTION",
        "ackToken": 3
    }
}
```

---------- Snippet end ----------

Where "080300" is the encoded payload

7.9. Request status

This message is used to get a status message. An optional value specifying the type of status can be added to the request:

- ENERGY_STATUS.
- HEALTH_STATUS.

If the value is omitted, a health status message will be returned by the tracker as response to the downlink request

7.9.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
            "version": "2"
        }
    },
    "message": {
       "downMessageType": "REQUEST_STATUS",
       "ackToken": 3
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2021-03-25T10:12:02.520Z",
    "meta": {
        "lorawan": {
            "fPort": 2
        }
    },
    "thing": {
        "points": {},
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "raw": {
        "binary": "0903"
    },
    "message": {
        "downMessageType": "REQUEST_STATUS",
        "ackToken": 3
    }
}
```

---------- Snippet end ----------

Where `"0903"` is the encoded payload

---------- Snippet begin ----------

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
       "downMessageType": "REQUEST_STATUS",
       "ackToken": 3,
       "statusType": "ENERGY_STATUS"
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2021-03-25T10:12:02.520Z",
    "meta": {
        "lorawan": {
            "fPort": 2
        }
    },
    "thing": {
        "points": {},
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "raw": {
        "binary": "090300"
    },
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
    "message": {
        "downMessageType": "REQUEST_STATUS",
        "ackToken": 3,
        "statusType": "ENERGY_STATUS"
    }
}
```

---------- Snippet end ----------

Where `"090300"` is the encoded payload

7.10. Receive SMS

This message is used to receive a SMS message. The request requires an extra object called sms with the following fields:

- senderId: Number identifying the sender
- message: Message string

7.10.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
      "downMessageType":"SMS",
      "ackToken":0,
      "sms":{
        "senderId":123456,
        "message":"You're talking to me?"
      }
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2021-03-25T10:12:02.520Z",
    "meta": {
        "lorawan": {
            "fPort": 2
        }
    },
    "thing": {
        "points": {},
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "raw": {
        "binary": "0d0001e240596f752772652074616c6b696e6720746f206d653f "
    },
    "message": {
        "downMessageType": "REQUEST_STATUS",
        "ackToken": 3
    }
}
```

---------- Snippet end ----------

Where `"0d0001e240596f752772652074616c6b696e6720746f206d653f"` is the encoded payload

7.11. Debug command

This message performs debug actions on the tracker. The request requires an extra field called *DebugCommandType*, which can take the following string values:

- *RESET*: Reset the tracker. Optional values can be given to specify the reset action.
- *READ_CURRENT_ERROR_AND_SEND_IT*: Read the last registered error and send it via LoRa.

- MAKE_TRACKER_RING: make the device ring
- *TRIGGER_AN_ERROR*: Generate an error. Internal use only.
- *RESET_BLE_PAIRING*: Delete the BLE bonding.
- TRIGGER_HEARTBEAT_MESSAGE: Trigger a heartbeat message
- READ_TX_POWER_INDEX: Read TX Power Index and send a TX power debug uplink
- WRITE_TX_POWER_INDEX: Write New TX Power Index (For this type of debug command, a TxPowerIndex value is needed)
- TRIGGER_BLE_BOOTLOADER: Trigger BLE bootloader
- SPECIFIC_FIRMWARE_PARAMETERS_REQUEST: Specific firmware parameters request
- CONFIGURE_STARTUP_MODES: Configure startup modes
- START_AND_STOP_BLE_ADVERTISEMENT: Start and stop BLE advertisement

7.11.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "DEBUG_COMMAND",
        "debugCommandType": "RESET",
        "ackToken": 2
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
```

```
    "direction": "downlink",
    "sourceTime": "2019-06-07T14:38:23.938Z",
    "meta": {
        "driverId": "abeeway:asset-tracker:2"
    },
    "raw": {
        "binary": "ff0201"
    },
    "message": {
        "downMessageType": "DEBUG_COMMAND",
        "debugCommandType": "RESET",
        "ackToken": 2
    },
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    }
}
```

---------- Snippet end ----------

Where `"ff0201"` is the encoded payload

7.11.2. Reset Device with reset action value specified

This message resets the device with a reset action specified. Possible values are: RESET_DEVICE, DELETE_CONFIG_RESET, DELETE_CONFIG_BLE_BOND_RESET. This value is optional. If it is missing, the command is equivalent to the one with the value set to RESET_DEVICE. **ResetAction is new from AT v1.8.**

7.11.2.1. Example

The Json data content of the request is:

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
```

```
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "DEBUG_COMMAND",
        "debugCommandType": "RESET",
        "ackToken": 2,
        "resetAction": "DELETE_CONFIG_BLE_BOND_RESET"
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2019-06-07T14:39:36.800Z",
    "meta": {
        "driverId": "abeeway:asset-tracker:2"
    },
    "raw": {
        "binary": "ff020103"
    },
    "message": {
        "downMessageType": "DEBUG_COMMAND",
        "debugCommandType": "RESET",
        "ackToken": 2,
        "resetAction": "DELETE_CONFIG_BLE_BOND_RESET"
    },
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    }
}
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

---------- Snippet end ----------

Where `"ff020103"` is the encoded payload

7.11.3. Write New TX Power Index

This message performs write new TX Power Index . The request requires an extra field called *DebugCommandType* containing "*WRITE_TX_POWER_INDEX*" and a second the field called TxPowerIndex containing the new value of TX Power Index.

7.11.3.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "DEBUG_COMMAND",
        "debugCommandType": "WRITE_TX_POWER_INDEX",
        "ackToken": 2,
        "txPowerIndex": 2
    }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2019-06-07T14:40:32.511Z",
    "meta": {
```

```
        "driverId": "abeeway:asset-tracker:2"
    },
    "raw": {
        "binary": "ff020702"
    },
    "message": {
        "downMessageType": "DEBUG_COMMAND",
        "debugCommandType": "WRITE_TX_POWER_INDEX",
        "ackToken": 2,
        "txPowerIndex": 2
    },
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    }
}
```

---------- Snippet end ----------

Where `"ff020702"` is the encoded payload

7.11.4. Set buzzer melody and duration

This message rings the tracker with a specific melody and duration . The request requires an extra field called *DebugCommandType* containing " *MAKE_TRACKER_RING*". Second field "*melodyId*" and third field "*buzzerDuration*" are both optional.

7.11.4.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
        }
    },
    "message":{
      "downMessageType":"DEBUG_COMMAND",
      "ackToken":2,
      "debugCommandType":"MAKE_TRACKER_RING",
      "melodyId":"BLE_ADVERTISING",
      "buzzerDuration":100
   }
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2019-06-07T14:40:32.511Z",
    "meta": {
        "driverId": "abeeway:asset-tracker:2"
    },
    "raw": {
        "binary": "ff02030764"
    },
    "message": {
        "downMessageType": "DEBUG_COMMAND",
        "debugCommandType": "MAKE_TRACKER_RING",
        "ackToken": 2,
       "melodyId":"BLE_ADVERTISING",
        "buzzerDuration": 100
    },
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    }
}
```

---------- Snippet end ----------

Where `"ff02030764"` is the encoded payload

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

7.11.5. Configure Startup modes

This message configures the startup modes. The request requires an extra field called DebugCommandType containing "CONFIGURE_STARTUP_MODES" and a second field called startupModes containing the new values of the startupModes.

7.11.5.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "DEBUG_COMMAND",
        "debugCommandType": "CONFIGURE_STARTUP_MODES",
        "ackToken": 2,
        "startupModes":{
                "manufacturing": true,
                "shipping": false
                }
}
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2021-03-25T10:30:52.390Z",
    "meta": {
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
        "lorawan": {
            "fPort": 2
        }
    },
    "thing": {
        "points": {},
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "raw": {
        "binary": "ff020a01"
    },
    "message": {
        "downMessageType": "DEBUG_COMMAND",
        "debugCommandType": "CONFIGURE_STARTUP_MODES",
        "ackToken": 2,
        "startupModes": {
            "manufacturing": true,
            "shipping": false
        }
    }
}
```

---------- Snippet end ----------

Where "ff020a01" is the encoded payload

7.11.6. Start and Stop BLE advertisement

This message is used to start and stop BLE advertisement. The request requires an extra field called debugCommandType containing " START_AND_STOP_BLE_ADVERTISEMENT " and a second field called bleAdvertisementDuration containing the value of the duration of the advertisement.

7.11.6.1. Example

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "DEBUG_COMMAND",
        "debugCommandType": "START_AND_STOP_BLE_ADVERTISEMENT",
        "ackToken": 2,
        "bleAdvertisementDuration":100
}
}'
```

---------- Snippet End ----------

Where `message` is the downlink request in JSON

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2021-03-25T10:43:36.722Z",
    "meta": {
        "lorawan": {
            "fPort": 2
        }
    },
    "thing": {
        "points": {},
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "raw": {
        "binary": "ff020b0064"
    },
    "message": {
        "downMessageType": "DEBUG_COMMAND",
        "debugCommandType": "START_AND_STOP_BLE_ADVERTISEMENT",
        "ackToken": 2,
        "bleAdvertisementDuration": 100
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
        }
}
```

---------- Snippet end ----------

Where `"ff020b0064"` is the encoded payload

7.12. Parameter read

The driver provides a smart way to retrieve the tracker configuration. With a single command (downlink), you can either read the value of one or more parameters or retrieve the whole configuration.
The request (downlink message) is built as usual with the *downMessageType* field set to the string value *REQUEST_CONFIG*.
An optional field called *listParameterID* can be also provided:

- When provided: its value contains the list of parameter identifiers that is expected.
- When omitted: the full configuration will be requested.

Note that, the tracker will answer this downlink message with one or more uplinks. Each related uplinks will have the *messageType* set to *CONFIGURATION* and the *miscDataTag* set to *DEVICE_CONFIGURATION.* The field *deviceConfiguration* will be filled with the parameters list and their values.
The tracker is able to send up to 5 parameters value in one uplink. This means that if more than 5 parameters (or the full configuration) is requested, the tracker will send the required number of uplinks to fulfill the request.

The following table provide the mapping between the parameter names used by the decoder, the names used by the tracker and the identifiers.

Parameter identifier (See [2])	Decoder name	Config name (See [2])	Range	Unit	Supported since FW version
0	TRACKING_UL_PERIOD	ul_period	15-86400	Seconds	1.7
1	LORALIVE_PERIOD	lora_period	300-86400	Seconds	1.7
2	ENERGY_STATUS_PERIOD	pw_stat_period	0,300-604800	Seconds	1.7
3	PERIODIC_POSITION_PERIOD	periodic_pos_period	0,900-604800	Seconds	1.7
4	GPS_SCAN_MODE	NA	0	NA	1.7
5	GEOLOC_SENSOR_PROFILE	geoloc_sensor	0-12	Enum	1.7
6	ONESHOT_GEOLOC_METHOD	geoloc_method	0-7	Enum	1.7
7	EXT_ANTENNA_PROFILE	antenna	0-1	Enum	1.7, Deprecated since 2.0
8	MOTION_START_END_NB_TX	motion_nb_ps	1-60	NA	1.7
9	GPS_TIMEOUT	gps_timeout	30-300	Seconds	1.7
10	XGPS_TIMEOUT	agps_timeout	15-250	Seconds	1.7
11	GPS_EHPE	gps_ehpe	0-100	Meters	1.7
12	GPS_CONVERGENCE	gps_convergence	0-300	Seconds	1.7
13	CONFIG_FLAGS	config_flags	NA	NA	1.7
14	TRANSMIT_STRAT	transmit_strat	0-5	Enum	1.7
15	BLE_BEACON_COUNT	ble_beacon_count	1-4	NA	1.7
16	BLE_BEACON_TIMEOUT	ble_beacon_timeout	1-21	Seconds	1.7
17	GPS_STANDBY_TIMEOUT	gps_standby_timeout	0-43200	Seconds	1.7
18	CONFIRMED_UL_BITMAP	confirmed_ul_bitmap	NA	NA	1.7
19	CONFIRMED_UL_RETRY	confirmed_ul_retry	0-8	NA	1.7
20	MOTION_SENSITIVITY	motion_sensitivity	0-111	NA	1.8
21	SHOCK_DETECTION	shock_detection	1-30, 100-200	Percent	1.8
22	PERIODIC_ACTIVITY_PERIOD	periodic_activity_period	0,1800-86400	Seconds	1.8

23	MOTION_DURATION	motion_duration	60-3600	Seconds	1.8
24	GEOFENCING_SCAN_PERIOD	geofencing_scan_period	0-300	Seconds	1.8
25	GEOFENCING_COLLECT_PERIOD	geofencing_collect_period	0, 15-3600	Seconds	2.2
26	BLE_RSSI_FILTER	ble_rssi_filter	-100- -40	dBm	1.8
27	TEMPERATURE_HIGH	temperature_high	-44- 85,255	°C	1.9
28	TEMPERATURE_LOW	temperature_low	-44- 85,255	°C	1.9
29	TEMPERATURE_ACTION	temperature_action	0-3	Enum	1.9
30	TRANSMIT_STRAT_CUSTOM	transmit_strat_custom	NA	NA	1.9
31	NETWORK_TIMEOUT_CHECK	network_timeout_check	0,86400-5184000	Seconds	2.0
32	NETWORK_TIMEOUT_RESET	network_timeout_reset	0,21600-2592000	Seconds	2.0
33	COLLECTION_SCAN_TYPE	collection_scan_type	0-2	Enum	2.0
34	COLLECTION_NB_ENTRY	collection_nb_entry	1-20	NA	2.0
35	COLLECTION_BLE_FILTER_TYPE	collection_ble_filter_type	0-5	Enum	2.0
36	COLLECTION_BLE_FILTER_MAIN_1	collection_ble_filter_main_1	NA	NA	2.0
37	COLLECTION_BLE_FILTER_MAIN_2	collection_ble_filter_main_2	NA	NA	2.0
38	COLLECTION_BLE_FILTER_SEC_VALUE	collection_ble_filter_sec_value	NA	NA	2.0
39	COLLECTION_BLE_FILTER_SEC_MASK	collection_ble_filter_sec_mask	NA	NA	2.0
40	BATTERY_CAPACITY	battery_capacity	-1,0-65535	mAh	2.0
41	REED_SWITCH_CONFIGURATION	reed_switch_configuration	0-3	Enum	2.0
42	GNSS_CONSTELLATION	gnss_constellation	0-6	Enum	2.0
43	PROXIMITY_MINIMUM_SCAN_POWER	prox_scan_pwr_min	-90 ... 10	dBm	2.1

44	PROXIMITY_DISTANCE_COEFFICIENT	prox_distance_coef	1 – 256	NA	2.1
45	PROXIMITY_SCAN_FREQUENCY	prox_scan_frequency	2 – 3600	NA	2.1
46	PROXIMITY_BACKTRACE_MAXIMUM_AGE	prox_backtrace_max_age	1 – 256	Seconds	2.1
47	PROXIMITY_DISTANCE_SLIDING_WINDOW	prox_dist_sliding_window	1 – 256	Seconds	2.1
48	PROXIMITY_EXPOSURE_50	prox_exposure_50	0 – 256	NA	2.1
49	PROXIMITY_EXPOSURE_100	prox_exposure_100	0 – 256	NA	2.1
50	PROXIMITY_EXPOSURE_150	prox_exposure_150	0 – 256	NA	2.1
51	PROXIMITY_EXPOSURE_200	prox_exposure_200	0 – 256	NA	2.1
52	PROXIMITY_EXPOSURE_250	prox_exposure_250	0 – 256	NA	2.1
53	PROXIMITY_EXPOSURE_300	prox_exposure_300	0 – 256	NA	2.1
54	PROXIMITY_EXPOSURE_400	prox_exposure_400	0 – 256	NA	2.1
55	PROXIMITY_IMMEDIATE_ALARM_DISTANCE	prox_alarm_dist_immediate	0-10	Meters	2.1
56	PROXIMITY_ALARM_EXPOSURE	prox_alarm_exposure	0-65535	NA	2.1
57	PROXIMITY_IMMEDIATE_WARNING_DISTANCE	prox_warn_dist_immediate	0-10	Meters	2.1
58	PROXIMITY_WARNING_EXPOSURE	prox_warn_exposure	0-65535	NA	2.1
59	PROXIMITY_IMMEDIATE_RECORD_DISTANCE	prox_record_dist_immediate	0-10	Meters	2.1
60	PROXIMITY_RECORD_EXPOSURE	prox_record_exposure	1-65535	NA	2.1
61	PROXIMITY_ALARM_BUZZER_DURATION	prox_alarm_buz_dur	0-256	Seconds	2.1
62	PROXIMITY_WARNING_BUZZER_DURATION	prox_warn_buz_dur	0-256	Seconds	2.1
63	PROXIMITY_CONTACT_POLICY	prox_contact_policy	NA	NA	2.1
64	PROXIMITY_SCAN_DURATION	prox_scan_duration	0-60	Seconds	2.1
65	PROXIMITY_SCAN_WINDOW	prox_scan_window	10-10240	Milliseconds	2.1

66	PROXIMITY_SCAN_INTERVAL	prox_scan_Interval	15-10245	Millisecond	2.1
67	PROXIMITY_ALARM_REMANENCE	prox_alarm_remanence	0-256	Seconds	2.1
68	PROXIMITY_WARNING_REMANENCE	prox_warn_remanence	0-256	Seconds	2.1
69	PROXIMITY_BEACON_REPEAT	prox_bcn_repeat	0-65535	Milliseconds	2.1
70	PROXIMITY_BEACON_TX_POWER	prox_bcn_tx_power	-60..-20	dBm	2.1
71	PROXIMITY_REMINDER_PERIOD	prox_reminder_period	0-256	Seconds	2.1
72	PROXIMITY_REMINDER_DISTANCE	prox_reminder_distance	0-25.6	Meters	2.1
73	PROXIMITY_WARNING_DISABLE_DISTANCE	prox_warn_disable_dist	1-25.6	Meters	2.1
74	PROXIMITY_ALARM_DISABLE_DISTANCE	prox_alarm_disable_dist	1-25.6	Meters	2.1
75	PROXIMITY_MAX_SPEED_FILTER	prox_max_speed_filter	0 – 40	Meters/s	2.1
76	PROXIMITY_MAX_UPDATE	prox_max_update	300 – 43200	Seconds	2.1
77	BLE_FILTER_TYPE	position_ble_filter_type	0-5	Enum	2.1
78	BLE_FILTER_MAIN_1	position_ble_filter_main_1	NA	NA	2.1
79	BLE_FILTER_MAIN_2	position_ble_filter_main_2	NA	NA	2.1
80	BLE_FILTER_SEC_FILTER_VALUE	position_ble_filter_sec_value	NA	NA	2.1
81	BLE_FILTER_SEC_FILTER_MASK	position_ble_filter_sec_mask	NA	NA	2.1
82	BLE_FILTER_REPORT_TYPE	position_ble_report_type	0-3	Enum	2.1
83	BUZZER_VOLUME	buzzer_volume	0-100	NA	2.1
84	ANGLE_DETECTION_MODE	angle_detect_mode	0-3	Enum	2.2

85	ANGLE_REFERENCE_ACQUISITION	angle_ref_acq	0-3	Enum	2.2
86	ANGLE_REFERENCE_ACCELERATION_X	angle_ref_acc_x	65535, -1000-1000	mG	2.2
87	ANGLE_REFERENCE_ACCELERATION_Y	angle_ref_acc_y	65535, -1000-1000	mG	2.2
88	ANGLE_REFERENCE_ACCELERATION_Z	angle_ref_acc_z	65535, -1000-1000	mG	2.2
89	ANGLE_CRITICAL	angle_critical	5-175	Degrees	2.2
90	ANGLE_CRITICAL_HYSTERESIS	angle_critical_hyst	0-180	Degrees	2.2
91	ANGLE_REPORTING_MODE	angle_report_mode	NA	NA	2.2
92	ANGLE_REPORTING_PERIOD	angle_report_period	0, 30-3600	Seconds	2.2
93	ANGLE_REPORTING_REPEAT	angle_report_repeat	0-7	NA	2.2
94	ANGLE_RISING_TIME	angle_rising_time	0-3600	Seconds	2.2
95	ANGLE_FALLING_TIME	angle_falling_time	0-3600	Seconds	2.2
96	ANGLE_LEARNING_TIME	angle_learning_time	0-3600	Seconds	2.2
97	ANGLE_ACCELERATION_ACCURACY	angle_acc_accuracy	0-1000	mG	2.2
98	ANGLE_DEVIATION_DELTA	angle_deviation_delta	0-175	Degrees	2.2
99	ANGLE_DEVIATION_MIN_INTERVAL	angle_deviation_min_interval	0-1800	Seconds	2.2
100	ANGLE_DEVIATION_MAX_INTERVAL	angle_deviation_max_interval	0-86400	Seconds	2.2
101	DEFAULT_PROFILE	default_profile	0-3	Enum	2.2

102	PASSWORD_COMMAND_LINE_INTERFACE	password	100-999999	NA	2.2
103	GPS_T0_TIMEOUT	gps_t0_timeout	0-300	Seconds	2.2
104	GPS_FIX_TIMEOUT	gps_fix_timeout	0-300	Seconds	2.2
105	GPS_SCAN_DURATION	gps_scan_duration	370-3000	Milliseconds	2.2
106	BEACONING_TYPE	beaconing_type	0-5	enum	2.2
107	BEACONING_TX_POWER	beaconing_tx_power	NA	dBm	2.2
108	BEACONING_STATIC_INTERVAL	beaconing_static_interval	100-10000	Milliseconds	2.2
109	BEACONING_MOTION_INTERVAL	beaconing_motion_interval	100-10000	Milliseconds	2.2
110	BEACONING_MOTION_DURATION	beaconing_motion_duration	4-255	Seconds	2.2
111	BLE_CONNECTIVITY_ADVERTISING_DURATION	ble_cnx_adv_duration	30-18000	Seconds	2.2
112	BEACON_ID_0	beacon_id_0	NA	NA	2.3
113	BEACON_ID_1	beacon_id_1	NA	NA	2.3
114	BEACON_ID_2	beacon_id_2	NA	NA	2.3
115	BEACON_ID_3	beacon_id_3	NA	NA	2.3
116	BEACON_ID_4	beacon_id_4	NA	NA	2.3
117	SOS_PERIOD	sos_period	15-300	seconds	2.3
118	MOTION_DEBOUNCE	motion_debounce	0-255	NA	2.3
119	BUTTON_MAPPING	button_mapping	NA	NA	2.3

120	DEFAULT_DATARATE	default_datarate	-1 ,0-7	NA	2.3
121	GPS_EH_PE_MOTION	gps_ehpe_motion	0-100	Meters	2.4
122	GPS_CONVERGENCE_MOTION	gps_convergence_motion	0-300	Seconds	2.4
123	GPS_T0_TIMEOUT_MOTION	gps_t0_timeout_motion	0-300	Seconds	2.4

Special parameters					
245	BLE_CLI_ACTIVE	NA	0,1	NA	2.3
246	DYNAMIC_PROFILE	dynamic_profile	0-3	Enum	2.2
247	POWER_EH	power_consumption	0-4294967295	mA	2.0
248	BLE_BOND	delete_ble_bond	NA	NA	1.9
249	GET_MODE	mode	NA	NA	1.9
250	X_ACCELEROMETER_VALUE	NA	NA	mg	1.8
251	Y_ACCELEROMETER_VALUE	NA	NA	mg	1.8
252	Z_ACCELEROMETER_VALUE	NA	NA	mg	1.8
253	BLE_VERSION	NA	NA	NA	1.7
254	FW_VERSION	NA	NA	NA	1.7

7.12.1. Examples

1. Request the full configuration

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
```

```
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "REQUEST_CONFIG",
        "ackToken": 1
    }
}'
```

---------- Snippet end ----------

`"0301"` is the encoded payload

2. Request only the parameters having the identifier 1,2,3,4 and 5.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "REQUEST_CONFIG",
        "listParameterID": [
            1,
            2,
            3,
            4,
            5
        ],
        "ackToken": 2
    }
}'
```

---------- Snippet end ----------

`"03020102030405"` is the encoded payload

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

7.13. Parameter modification

This message allows the modification of a configuration parameter. It requires systematically an extra field called *ParameterName* containing the string name of the parameter and a second field containing the new value. The following table provides the list of accepted parameter names as well as the second field name and its value.
For the list of parameters, see Device configuration object

Note: Abeeway Driver allows you to set max 5 parameters at the same time in the same downlink

7.13.1. Examples

7.13.1.1. Modify the GPS parameters:

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "SET_PARAM",
        "ackToken": 10,
        "setParameters": {
            "gpsTimeout": 120,
            "xgpsTimeout": 250,
            "gpsEHPE": 100,
            "gpsStandbyTimeout": 10
        }
    }
}'
```

---------- Snippet end ----------

`"0b0a09000000780a000000fa0b00000064110000000a"` is the encoded payload

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

7.13.1.2. Modify the BLE parameters:

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "SET_PARAM",
        "ackToken": 10,
        "setParameters": {
            "bleBeaconCount": 3,
            "bleRssiFilter": -80,
            "bleBeaconTimeout": 5
        }
    }
}'
```

---------- Snippet end ----------

`"0b0b0f000000031affffffb01000000005"` is the encoded payload

7.13.1.3. Modify the Operational mode Parameters:

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
     "message": {
        "downMessageType": "SET_PARAM",
        "ackToken": 2,
        "setParameters": {
            "trackingUlPeriod": 86400,
            "loralivePeriod": 3600,
            "geolocSensorProfile": "WIFI_ONLY",
            "motionStartEndNbTx": 30,
            "motionDuration": 3600
        }
    }
}'
```

---------- Snippet end ----------

`"0b020000015180010000e10050000000080000001e1700000e10"` is the encoded payload

7.13.1.4. Modify the LORA parameters:

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
     "message": {
        "downMessageType": "SET_PARAM",
        "ackToken": 1,
        "setParameters": {
            "confirmedUplink": {
               "position": true,
               "activityStatus": false,
               "configuration": false,
```

```
                "debug": false,
                "energyStatus": false,
                "healthStatus": false,
                "event": false,
                "framePending": false,
                "extendedPosition": false,
                "heartbeat": false,
                "shockDetection": false,
                "shutdown": false,
                "scanCollection": false,
                "confirmedUlBitmap": 8,
                "confirmedUlRetry": 8
            },
            "transmitStrat": "DOUBLE_FIXED",
            "defaultDatarate": -1
        }
    }
}'
```

---------- Snippet end ----------

`"0b011200000008130000000080e0000000378ffffffff"` is the encoded payload

Note:

If the parameter "confirmedUlBitmap" is present in "confirmedUplink" object, the values of each specific bit in the same object are ignored.

7.13.1.5. Modify the side operational mode parameters:

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
     "message": {
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
        "downMessageType": "SET_PARAM",
        "ackToken": 5,
        "setParameters": {
            "periodicPositionInterval": 900,
            "oneshotGeolocMethod": "BLE",
            "periodicActivityPeriod": 1800
        }
    }
}'
```

---------- Snippet end ----------

`"0b0503000003840600000051600000708"` is the encoded payload

7.13.1.6. Modify Misc Parameters

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{

   },
   "thing":{
      "application":{
         "producerId":"abeeway",
         "moduleId":"asset-tracker",
         "version":"2"
      }
   },
   "message":{
      "downMessageType":"SET_PARAM",
      "ackToken":10,
      "setParameters":{
         "energyStatusPeriod":604800,
         "extAntennaProfile":"CERAMIC",
         "configFlags":{
            "framePendingMechanism":true,
            "buttonPressToTurnOFF":false,
            "doubleClickIsSosModeOrAlert":false,
            "downlinkSetParameterConfirmation":true,
            "wifiPayloadWithNoCypher":true,
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
            "bleAdvertisingAtStart":false,
            "selectWifiScanOrGeolocStartMessage":false,
            "ledBlinkWithGpsFix":false,
            "startMotionEventUplink":true,
            "endMotionEventUplink":false,
            "otaJoinWhenLeavingModeOff":false,
            "rejectAsymmetricPairing":false,
            "enableLongWifiPayload":false,
            "collectionLongReport":true,
            "autoStart":false,
            "forbidModeOff":false,
            "sosModeSound":false,
            "automaticDatarateSelection":true,
            "extendedPosition":false,
            "bleCli":false,
            "blePasskey":true,
            "wifiDatabase":true,
            "activeWifiScan":true
         },
         "buzzerVolume":100,
         "sosPeriod":120
      }
   }
}'
```

---------- Snippet end ----------

`"0b0a0200093a8007000000010d00222119530000006475000000078"` is the encoded payload

7.13.1.7. Modify accelerometer parameters

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{

   },
   "thing":{
      "application":{
         "producerId":"abeeway",
         "moduleId":"asset-tracker",
         "version":"2"
```

```
      }
   },
   "message":{
      "downMessageType":"SET_PARAM",
      "ackToken":10,
      "setParameters":{
         "motionSensitivity":{
            "fullScale":"2_G",
            "outputDataRate":"12_5_HZ",
            "sensitivity":105
         },
         "shockDetection":100,
         "motionDebounce":101
      }
   }
}'
```

---------- Snippet end ----------

`"0b0a14000000691500000064760000065"` is the encoded payload

7.13.1.8. Modify temperature parameters

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "SET_PARAM",
        "ackToken": 6,
        "setParameters": {
            "temperatureHigh": 60,
            "temperatureLow": -40,
            "temperatureAction": "TEMPERATURE_HIGH_TEMPERATURE_LOW"
        }
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
    }

}'
```

---------- Snippet end ----------

`"0b061b0000003c1cffffffd81d00000003"` is the encoded payload

7.13.1.9. Modify special parameters

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "message": {
        "downMessageType": "SET_PARAM",
        "ackToken": 10,
        "setParameters": {
            "mode": "ACTIVITY_TRACKING",
            "bleBond": "NOT_BONDED"
        }
    }

}'
```

---------- Snippet end ----------

`"0b06f800000000f900000004"` is the encoded payload

7.13.1.10. Modify Custom Transmission Strat parameters

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
```

```
    "meta":{},
    "thing":{
       "application":{
          "producerId":"abeeway",
          "moduleId":"asset-tracker",
          "version":"2"
       }
    },
    "message":{
       "downMessageType":"SET_PARAM",
       "ackToken":6,
       "setParameters":{
          "transmitStratCustom":{
             "firstTransmissionDatarateDistribution":"RANDOM",
             "secondTransmissionDatarateDistribution":"BELL_CURVE",
             "firstTransmissionDatarate":{
                "dr0":false,
                "dr1":true,
                "dr2":false,
                "dr3":true,
                "dr4":false,
                "dr5":false,
                "dr6":false,
                "dr7":false
             },
             "secondTransmissionDatarate":{
                "dr0":false,
                "dr1":false,
                "dr2":true,
                "dr3":false,
                "dr4":true,
                "dr5":false,
                "dr6":false,
                "dr7":false
             },
             "transmissionType":"DOUBLE",
             "adrenabled":true
          }
       }
    }
}'
```

---------- Snippet end ----------

`"0b061e00140a22"` is the encoded payload

Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

7.13.1.11. Modify scan collection parameters

Note: BLE position filtering requires update of 7 parameters so we need to send two downlinks as we can only set 5 parameters at a time for one downlink message

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{},
   "thing":{
      "application":{
         "producerId":"abeeway",
         "moduleId":"asset-tracker",
         "version":"2"
      }
   },
   "message":{
      "downMessageType": "SET_PARAM",
      "ackToken": 10,
      "setParameters": {
         "collectionScanType": "BLE_BEACONS",
         "collectionBleFilterMain1": 51966,
         "collectionBleFilterMain2": 57005,
         "collectionBleFilterSecValue": 47834,
         "collectionBleFilterSecMask": 47806
      }
   }
}'
```

---------- Snippet end ----------

`"0b0a2100000001240000cafe250000dead260000bada270000babe"` is the encoded payload

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{},
   "thing":{
      "application":{
```

```
            "producerId":"abeeway",
            "moduleId":"asset-tracker",
            "version":"2"
         }
      },
      "message": {
         "downMessageType": "SET_PARAM",
         "ackToken": 10,
         "setParameters": {
            "collectionBleFilterType": "EDDYSTONE_UID",
            "collectionNbEntry": 7
         }
      }
}'
```

---------- Snippet end ----------

`"0b0a22000000072300000001"` is the encoded payload

7.13.1.12. Modify BLE Position Filtering parameters

Note: BLE position filtering requires update of 7 parameters so we need to send two downlinks as we can only set 5 parameters at a time for one downlink message

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{},
   "thing":{
      "application":{
         "producerId":"abeeway",
         "moduleId":"asset-tracker",
         "version":"2"
      }
   },
   "message": {
      "downMessageType": "SET_PARAM",
      "ackToken": 10,
      "setParameters": {
         "bleFilterType": "EDDYSTONE_UID",
         "bleFilterMain1": 51966,
         "bleFilterMain2": 57005,
         "bleFilterSecValue": 47834,
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
            "bleFilterSecMask": 47806
        }
    }
}'
```

---------- Snippet end ----------

`"0b0a4d000000014e0000cafe4f0000dead500000bada510000babe"` is the encoded payload

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{},
   "thing":{
      "application":{
         "producerId":"abeeway",
         "moduleId":"asset-tracker",
         "version":"2"
      }
   },
   "message": {
      "downMessageType": "SET_PARAM",
      "ackToken": 10,
      "setParameters": {
         "bleBeaconCount": 4,
         "bleFilterReportType": "SHORT_ID"
      }
   }
}'
```

---------- Snippet end ----------

`" 0b0a0f000000045200000001"` is the encoded payload

7.13.1.13. Modify proximity detection parameters

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
```

```
    "meta":{},
    "thing":{
       "application":{
          "producerId":"abeeway",
          "moduleId":"asset-tracker",
          "version":"2"
       }
    },
    "message":{
       "downMessageType": "SET_PARAM",
       "ackToken": 10,
       "setParameters": {
          "proximityContactPolicy": {
             "enable":true,
             "store":true,
             "uplink":true
          },
          "proximityScanDuration": 100,
          "proximityScanWindow": 10005

       }
    }
}'
```

---------- Snippet end ----------

`"0b0a3f0000000740000003e84100002715"` is the encoded payload

7.13.1.14. Modify angle detection parameters

Note: angle detection requires update of 17 parameters so we need to send 4 downlinks as we can only set 5 parameters at a time for one downlink message

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{},
   "thing":{
      "application":{
         "producerId":"abeeway",
         "moduleId":"asset-tracker",
         "version":"2"
      }
   },
   "message": {
      "downMessageType": "SET_PARAM",
      "ackToken": 10,
      "setParameters": {
         "angleDetectionMode": "CRITICAL_ANGLE_DETECTION",
          "angleReferenceAcquisition" : "ASSISTED",
          "angleReferenceAccelerationX" : 80,
          "angleReferenceAccelerationY": 96,
        "angleReferenceAccelerationZ": 0

      }
   }
}'
```

---------- Snippet end ----------

`"0b0a5400000001550000000356000000050570000006058000000000"` is the encoded payload

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{},
   "thing":{
      "application":{
         "producerId":"abeeway",
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
            "moduleId":"asset-tracker",
            "version":"2"
         }
      },
      "message": {
         "downMessageType": "SET_PARAM",
         "ackToken": 10,
         "setParameters": {
            "angleCritical" : 30,
            "angleCriticalHysteresis" : 5,
            "angleReportingPeriod" : 300,
            "angleReportingRepeat" : 1,
            "angleRisingTime": 5
         }
      }
}'
```

---------- Snippet end ----------

"0b0a590000001e5a000000055c0000012c5d000000015e00000005" is the encoded payload.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{},
   "thing":{
      "application":{
         "producerId":"abeeway",
         "moduleId":"asset-tracker",
         "version":"2"
      }
   },
   "message": {
      "downMessageType": "SET_PARAM",
      "ackToken": 10,
      "setParameters": {
           "angleFallingTime" : 5,
           "angleLearningTime" : 5,
           "angleAccelerationAccuracy" : 100,
           "angleDeviationDelta" : 0,
           "angleDeviationMinInterval": 10
      }
   }
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
}'
```

---------- Snippet end ----------

"0b045f00000005600000000561000000646200000000630000000a" is the encoded payload.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{},
   "thing":{
      "application":{
         "producerId":"abeeway",
         "moduleId":"asset-tracker",
         "version":"2"
      }
   },
   "message":{
      "downMessageType": "SET_PARAM",
      "ackToken": 10,
      "setParameters": {
         "angleReportingMode":
           {
            "normalToCritical": true,
            "criticalToNormal": false,
           "learningToNormal": false,
            "normalToLearning": false,
            "criticalToLearning": false
            },
            "angleDeviationMaxInterval": 10

      }
   }
}'
```

---------- Snippet end ----------

"0b0a5b00000001640000000a" is the encoded payload.

7.13.1.15. Modify Quuppa beaconing parameters

Note: Quuppa beaconing requires update of 5 parameters so we need to send 1 downlink as we can only set 5 parameters at a time for one downlink message.

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{},
   "thing":{
      "application":{
         "producerId":"abeeway",
         "moduleId":"asset-tracker",
         "version":"2"
      }
   },
   "message": {
      "downMessageType": "SET_PARAM",
      "ackToken": 10,
      "setParameters": {
         "beaconingType": "QUUPPA",
          "beaconingTxPower" : "0",
          "beaconingStaticInterval" : 10000,
          "beaconingMotionDuration": 20,
        "beaconingMotionInterval": 333
      }
   }
}'
```

---------- Snippet end ----------

"0b0a6a000000026b000000006c000027106e000000146d0000014d" is the encoded payload

7.13.1.16. Modify geofencing parameters

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{},
   "thing":{
      "application":{
         "producerId":"abeeway",
         "moduleId":"asset-tracker",
         "version":"2"
      }
   },
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
   "message": {
      "downMessageType": "SET_PARAM",
      "ackToken": 10,
      "setParameters": {
         "geofencingScanPeriod": 120,
         "geofencingCollectPeriod": 600,
         "collectionScanType": "BLE_BEACONS"
      }
   }
}'
```

---------- Snippet end ----------

"0b0a18000000781900000258" is the encoded payload

7.13.1.17. Modify button mapping

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{},
   "thing":{
      "application":{
         "producerId":"abeeway",
         "moduleId":"asset-tracker",
         "version":"2"
      }
   },
   "message": {
      "downMessageType": "SET_PARAM",
      "ackToken": 9,
      "setParameters": {
         "buttonMapping": {
            "buttonPress": "START_STOP_SOS",
            "oneButtonClick": "WHITELIST_WITH_PROXIMITY",
            "twoButtonClick": "BATTERY_LEVEL_WITH_LED",
            "threeButtonClick": "NO_ACTION",
            "buttonPressDuration": 5
         }
      }
   }
}'
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

---------- Snippet end ----------

"0b097700050142" is the encoded payload

7.13.1.18. Modify motion GPS parameters

---------- Snippet begin ----------

```
curl -X POST \
  http://localhost:8095/iot-flow-driver/v1/encode \
  -H 'Content-Type: application/json' \
  -d '{
   "direction":"downlink",
   "meta":{

   },
   "thing":{
      "application":{
         "producerId":"abeeway",
         "moduleId":"asset-tracker",
         "version":"2"
      }
   },
   "message":{
      "downMessageType":"SET_PARAM",
      "ackToken":9,
      "setParameters":{
         "gpsEHPEMotion":100,
         "gpsConvergenceMotion":250,
         "gpsT0TimeoutMotion":250
      }
   }
}'
```

---------- Snippet end ----------

"0b0979000000647a000000fa7b000000fa" is the encoded payload

8. Decoding a downlink

The input of the decoder is a LoRa downlink payload sent to an Abeeway Tracker using a firmware Asset Tracker version 1.7 or more.
The output of the decoder is a string in the well known JSON format.
The following example illustrate the decoding of a downlink message.

8.1. Example

This example shows the decoding of a downlink message.
The LoRa payload to be parsed is the following: "0b0a150000005a"

---------- Snippet begin ----------

```
curl -X POST \
  http://dx-api.thingpark.com/location-driver/latest/api/decode \
  -H 'Content-Type: application/json' \
  -d '{
    "direction": "downlink",
    "meta": {},
    "thing": {
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    },
    "raw": {
        "binary": "0b0a150000005a"

    }
}'
```

---------- Snippet End ----------

The response is:

---------- Snippet begin ----------

```
{
    "direction": "downlink",
    "sourceTime": "2021-03-25T11:18:48.080Z",
    "meta": {
        "driverId": "abeeway:asset-tracker:3"
    },
    "raw": {
        "binary": "0b0a150000005A"
    },
    "message": {
        "downMessageType": "SET_PARAM",
        "ackToken": 10,
        "setParameters": {
            "shockDetection": 90
```


Abeeway SAS, 2000 Route des Lucioles, Sophia-Antipolis, 06410 Biot, France
Restricted distribution

```
        },
        "payload": "0b0a150000005a"
    },
    "thing": {
        "points": {},
        "application": {
            "producerId": "abeeway",
            "moduleId": "asset-tracker",
            "version": "2"
        }
    }
}
```

---------- Snippet end ----------

Where message is the decoded payload.

9. References

[1] Abeeway Firmware Reference Guide. Link: here
[2] Abeeway Firmware Parameters. Link: here