# Klimator IoT Sensors

Device model descriptions for the BA-project Spring 2024

## Revision history

| Date | Author | Comments | Revision |
|---|---|---|---|
| **2024-02-27** | Mohamed Elmi | Adjusted for the BA-project spring 2024 | 3 |
| **2023-01-23** | Mohamed Elmi | Added data sizes for measurement parameters. Renamed rainFall to precipitation. | 2 |
| **2022-06-14** | Mohamed Elmi | Initial revision | 1 |

# Contents

# Introduction

This document describes the device models of the Klimator IoT sensors, their measurement parameters in terms of parameter name, measurement unit and measurement precision. In addition, any device model specific features are described.

The device model descriptions are grouped into appropriate categories.

# Potential faults and errors

## Heartbeat

The devices are expected to communicate back to our backend at minimum once a day, typically around every 15min for weather stations in the winter season. The water level devices are expected to communicate back in and around every 15min, but the customer can also set a custom transmission interval.

The person-counter devices are different in that they measure and transmit once every hour if there has been any activity in that interval otherwise, they send an empty "heartbeat" message after 24 hours of non-activity registered.

Overall, it is safe to use 24 hours as the minimum "heartbeat" interval.

## Faulty measurements

### Weather stations

For the weather stations we are interested in having the following ranges for the parameters.

#### Road temperature

For the p100 thermocable used for these devices, we are interested in the ranges -30 to +60. We have experienced that it generates faulty < -30C values for broken thermocable that needs to be replaced. Naturally it is possible to have perfectly < -30C values if the conditions are there and the is no brake in the thermocable.

#### Air temperature

We are also aiming for the ranges -30C to +40 to be on the safe side of the current climate and that it is not stuck in given value.

#### Humidity

We have measured in the ranges 30 – 100%. If it is stuck on given value especially 100% then we have experienced that it is due to faulty sensor.

## Water level

The water level devices that use an ultrasonic sensor, during installation we measure the distance from the sensor down to the bottom of the place it is measuring for. This distance from the sensor to the bottom we call boundary distance. Due to various reasons the ultrasonic wave can be reflected off surfaces close by and generates a distance that is larger than the boundary distance. We regard any distance larger than the boundary distance as a reflection.

Also, we have specific distance values generated by the sensor that are particularly faulty. These are anything less than 26cm and everything above 495cm.

## Person counter

Due to the installation procedure and placement, we expect any counts above 100persons in 10min intervals to be of faulty measurement that are triggered by external factors such as IR radiation and irregular movements in front of the sensor.

## Message drop

The vast majority of our device models generate a incremental sequence number for measurements and transmissions. This is used to figure out if we have any brake in the sequence and have a tolerance of around 10%. The message drops can be caused by various issues, such as low network coverage. It is also possible that the device firmware crashes and restarts the sequence number or that it has been serviced.

\

## Battery

There are different types of batteries used in our device models that use different chemistries thus it is not safe to expect all of them to follow a linear discharge curve. For this project we can assume that the safe range is 7.4v – 3v.

# Weather stations

## WSense

WSense represent the weather station device with multiple sensors. The device continuously measures the various parameters based on the defined measurement interval and transmits measurements after the defined transmission interval.

**Parameters**

| Name | Unit | Precision | Data size | Description | Type |
|------|------|-----------|-----------|-------------|------|
| airTemerature | °C | 0.1°C | 2 bytes | Measured air temperature. | Measurement |
| roadTemerature | °C | 0.1°C | 2 bytes | Measured road temperature. | Measurement |
| airHumidity | % | 1 % | 1 byte | Relative air humidity | Measurement |
| batteryLevel | V | 0.01V | 2 bytes | Battery voltage at the time of transmission | Measurement |
| seqNumber | N/A | 1 | 4 bytes | Sequence number of the measurement | Metadata |
| rssi | dBm | 1 dBm | 1 byte | Receiver strength string indicator | Network |
| time | date-time | N/A | 24 bytes | Measurement timestamp in ISO 8061 compliant format | Metadata |
| createdAt | date-time | N/A | 24 bytes | Timestamp when created in the backend in ISO 8061 compliant format | Metadata |

## Payload structure

```
[{
    "networkId": "14484CA85F912",
    "airTemperature": -1.5,
    "roadTemperature": -2.5,
    "airHumidity": 95,
    "batteryLevel": 6.5,
    "time": "2023-12-01T00:03:00.000Z",
    "createdAt": "2023-12-01T00:04:38.502Z"
  },
  {

    "networkId": "14484CA85F912",
    "createdAt": "2023-12-01T00:04:36.418Z",
    "time": "2023-12-01T00:04:36.322Z",
    "rssi": 25,
    "rsrq": 4,
    "seqNumber": 6687,
    "acxti": 1,
    "acmi": 5,
    "firmwareVersion": 1.7,
    "eventNumber": null,
    "messageType": "cfg"
  }]
```

Device object structure

```json
{
  "name": "Mårbyleden, Märsta",
  "networkId": "14484CA85F912",
  "geo": {
    "lat": 59.62838453711938,
    "lon": 17.885008770989145
  },
  "properties": {
    "probeTempOffset": -0.9
  },
  "lastSeenAt": "2023-03-08T15:08:06.328Z"
}
```

## WRSense

WRSense represent the weather station device with multiple sensors. The device continuously measures the various parameters based on the defined measurement interval and transmits measurements after the defined transmission interval. In addition, this device model has also a rain sensor attached to it.

**Parameters**

| Name | Unit | Precision | Data size | Description | Type |
|------|------|-----------|-----------|-------------|------|
| **airTemerature** | °C | 0.1°C | 2 bytes | Measured air temperature. | Measurement |
| **roadTemerature** | °C | 0.1°C | 2 bytes | Measured road temperature. | Measurement |
| **airHumidity** | % | 1 % | 1 byte | Relative air humidity | Measurement |
| **precipitation** | mm | 0.2mm | 2 bytes | Measured precipitation in between measurement intervals. | Measurement |
| **batteryLevel** | V | 0.01V | 2 bytes | Battery voltage at the time of transmission | Measurement |
| **seqNumber** | N/A | 1 | 4 bytes | Sequence number of the measurement | Metadata |
| **transmissionSeqNumber** | N/A | 1 | 4 bytes | Transmission sequence number | Metadata |
| **rssi** | dBm | 1 dBm | 1 byte | Receiver strength string indicator | Network |
| **time** | date-time | N/A | 24 bytes | Measurement timestamp in ISO 8061 compliant format | Metadata |
| **createdAt** | date-time | N/A | 24 bytes | Timestamp when created in the backend in ISO 8061 compliant format | Metadata |

Payload structure

```json
[
  {
    "roadTemperature": 1.04,
    "seqNumber": 1,
    "airTemperature": 2.79,
    "airHumidity": 85,
    "precipitation": 0,
    "batteryLevel": 5.02,
    "rssi": -76,
    "rssiMapped": 19,
    "transmissionSeqNumber": 1,
    "firmwareVersion": "v2.5",
    "networkId": "E067-6EB9-0614-9339",
    "deviceTypeNumber": 11,
    "time": "2023-12-25T19:01:13.000Z",
    "createdAt": "2023-12-25T19:03:47.094Z"
  },
  {
    "roadTemperature": 0.93,
    "seqNumber": 2,
    "airTemperature": 2.46,
    "airHumidity": 86,
    "precipitation": 0,
    "batteryLevel": 5.03,
    "rssi": -75,
    "rssiMapped": 19,
    "transmissionSeqNumber": 2,
    "firmwareVersion": "v2.5",
    "networkId": "E067-6EB9-0614-9339",
    "deviceTypeNumber": 11,
    "time": "2023-12-25T19:05:34.000Z",
    "createdAt": "2023-12-25T19:08:41.772Z"
  }
]
```

Device object structure

```json
{
  "name": "Kirkholt + Kamera + regn",
  "networkId": "14484CA85F3FB",
  "geo": {
    "lat": 57.30122089593072,
    "lon": 10.156051102833912
  },
  "properties": {
    "batteryLevel": 5.03,
    "firmwareVersion": "v2.5",
    "probeTempOffset": 0.5
  }
}
```

# Water level sensors

## USense

USense represent the water level sensors that use ultrasonic sensor to measure the distance from sensor to water surface, which then is computed to actual relative water level.

**Parameters**

| Name | Unit | Precision | Data size | Description | Type |
|---|---|---|---|---|---|
| **waterLevel** | m | 0.01m | 2 bytes | Computed water level based on measured distance to water surface. In proper elevation if elevation parameter is set, otherwise elevation 0 is assumed – effectively the relative water level. | Measurement |
| **distance** | m | 0.01m | 2 bytes | Distance from sensor to water level surface. | Measurement |
| **batteryLevel** | V | 0.01V | 2 bytes | Battery voltage at the time of transmission | Measurement |
| **seqNumber** | N/A | 1 | 4 bytes | Sequence number of the measurement | Metadata |
| **rssi** | dBm | 1dBm | 1 byte | Receiver strength string indicator | Network |
| **time** | date-time | N/A | 24 bytes | Measurement timestamp in ISO 8061 compliant format | Metadata |
| **createdAt** | date-time | N/A | 24 bytes | Timestamp when created in the backend in ISO 8061 compliant format | Metadata |

### Device object structure

```
{
  "name": "Svanevej 17 – Overløb",
  "networkId": "14484CA7BED21",
  "geo": {
    "lat": 55.79046820230201,
    "lon": 9.734068773521798
  },
  "properties": {
    "boundary_distance": 186,
    "overFlowThreshold": 64,
    "firmwareVersion": 1.4,
    "batteryLevel": 6.1
  }
}
```

Payload structure

```json
[
  {
    "networkId": "14484CA7BED21",
    "createdAt": "2023-12-11T08:02:00.088Z",
    "time": "2023-12-11T08:01:59.943Z",
    "rssi": 13,
    "seqNumber": 3,
    "firmwareVersion": 1.4,
    "messageType": "cfg"
  },
  {
    "networkId": "14484CA7BED21",
    "distance": 1.66,
    "waterLevel": 0.03,
    "batteryLevel": 6.3,
    "time": "2023-12-11T08:02:05.000Z",
    "createdAt": "2023-12-11T08:02:25.120Z"
  }
]
```

# Person counters

## PeopleSense

PeopleSense represents the sensor used to count people passing in front of the sensor. The sensor aggregates the counts in 10min intervals and sends measurements once in hour if there has been any activity.

**Parameters**

| Name | Unit | Precision | Data size | Description | Type |
|------|------|-----------|-----------|-------------|------|
| **personCount** | N/A | 1 | 2 bytes | 10min aggregated count of persons passing in front of the sensor. | Measurement |
| **batteryLevel** | V | 0.01V | 2 bytes | Battery voltage at the time of transmission | Measurement |
| **seqNumber** | N/A | 1 | 4 bytes | Sequence number of the measurement | Metadata |
| **rssi** | dBm | 1dBm | 1 byte | Receiver strength string indicator | Network |
| **createdAt** | date-time | N/A | 24 bytes | Timestamp when created in the backend in ISO 8061 compliant format | Metadata |

## Device object structure

```json
{
  "name": "Tirsbæk Strand",
  "networkId": "1FA5528",
  "geo": {
    "lat": 55.7090972,
    "lon": 9.6317105
  },
  "properties": {}
}
```

Payload structure

```json
[
  {
    "rssi": -140,
    "batteryLevel": 5.65,
    "networkId": "1FA5528",
    "personCount": 2,
    "seqNumber": 1237,
    "time": "2023-12-01T05:29:52.000Z",
    "createdAt": "2023-12-01T06:19:54.066Z"
  },
  {
    "rssi": -140,
    "batteryLevel": 5.65,
    "networkId": "1FA5528",
    "personCount": 5,
    "seqNumber": 1237,
    "time": "2023-12-01T05:39:52.000Z",
    "createdAt": "2023-12-01T06:19:54.067Z"
  }
]
```