ANavS Precise Positioning Systems – Technical Reference Guide

Applicable to:

- **A-ROX GNSS-INS tightly coupled positioning system**

ANavS GmbH – Advanced Navigation Solutions
Gotthardstrasse 40
80686 München, Germany
Tel.: +49 - (0)89 - 89056721
Fax: +49 - (0)89 - 89056720
Internet: http://www.anavs.com/

Abstract

The following guide provides a detailed explanation for the usage of the A-ROX GNSS-INS system.

The A-ROX GNSS-INS system is a high-precision tightly coupled sensor fusion solution designed for demanding applications in automotive, railway, and maritime navigation. It combines all-frequency GNSS (GPS, Galileo, Beidou, Glonass), real-time RTK and PPP corrections, a FOG-grade MEMS IMU, and odometry data to deliver centimeter-level positioning accuracy even in GNSS-denied environments.

Built on a modular hardware platform, the A-ROX system offers seamless integration with RTK networks, cloud-based fleet management, and real-time data processing via ROS2, FastAPI, NMEA and ACOM data format. It supports multiple communication protocols, including 5G mobile network connection, Gigabit Ethernet, Wi-Fi, and CAN-FD, ensuring reliable connectivity and data exchange.

A key feature of A-ROX is its seamless RTK-to-PPP handover, ensuring continuous high-accuracy navigation even when RTK corrections are unavailable. The system is optimized for applications such as ADAS testing, autonomous vehicle navigation, georeferencing of LiDAR and camera data, and railway tracking and maintenance operations.

With its web-based configuration, real-time visualization tools, and automated post-processing capabilities, the A-ROX is an ideal solution for precision positioning, mapping, and vehicle localization across multiple industries.

Key Features:

- Best in class RTK & PPP accuracy with all-frequency GNSS
- Seamless RTK-PPP transition for uninterrupted navigation
- Integration with ROS2, ACOM, FAST-API and NMEA formats
- Web-based postprocessing engine for generating ground truth including fleet management
- Optimized for automotive, railway, and maritime applications
- Made in Germany – high reliability and robust performance

Typographical Conventions

`abc -param`	Command-line instructions, e.g., in shell
abc -param	MSRTKF command-line instructions

List of Acronyms

Acronym	Description
PAD	Position- and Attitude Determination Engine
POI	Point of Interests

Document Change Log

Issue	Revision	Sections Affected	Details of Change
1	0	All	Initial version of the document.
1	1	3, 4	Added more explanations to the screenshots
1	2	5.5	Changed NMEA description to reflect changes from v3.01 to v4.10 and newly supported sentences.

1 System Overview

The A-ROX GNSS-INS system is a high-precision tightly coupled sensor fusion solution designed to provide reliable and accurate position, velocity, and attitude information across various applications. It integrates all-frequency GNSS receivers, an advanced FOG-grade MEMS IMU, and odometry data to ensure robust navigation performance, even in environments where GNSS signals are unreliable or unavailable.

This section provides an in-depth look at the applications, architecture, and key system components that make A-ROX a leading choice for automotive, railway and maritime navigation.

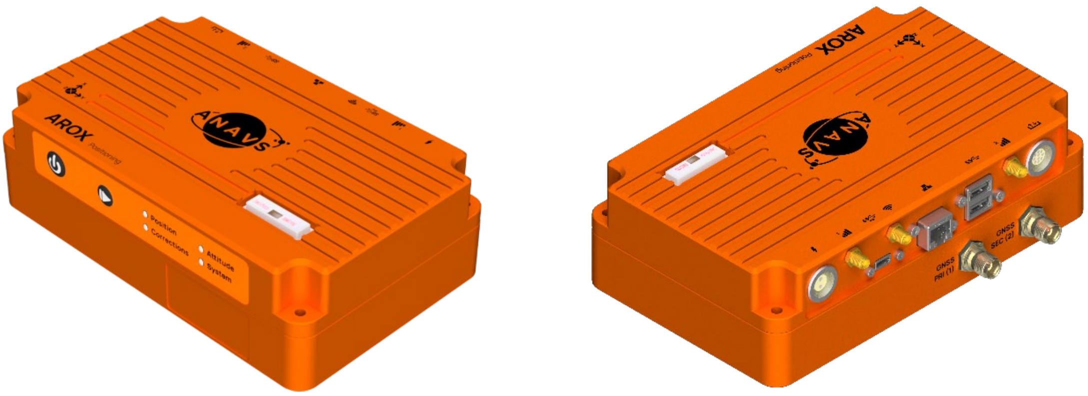


Figure 1: A-ROX system with status panel (left) and interfaces (right)

1.1 Applications

A-ROX is designed for high-precision positioning and navigation in a wide range of industries, including:

- **Automotive & Autonomous Vehicles**
    - Ground truth positioning for ADAS testing and validation
    - High-accuracy localization for autonomous driving applications
    - Vehicle dynamics testing with precise position, velocity, and attitude data
    - SLAM-based navigation support in GNSS-challenged or denied environments
- **Railway**
    - Accurate track surveying and maintenance monitoring
    - Positioning solutions for railway automation and fleet tracking
    - Monitoring of maintenance work
- **Surveying & Mapping**

    - Georeferencing of LiDAR, radar, and cameras for high-resolution mapping
    - GNSS-denied navigation for underground or urban canyon environments
    - Accurate trajectory recording for mobile mapping applications
- **Maritime & Aerospace**
    - Vessel navigation and dynamic positioning in offshore environments
    - Unmanned aerial vehicle (UAV) positioning for autonomous flight

A-ROX excels in scenarios where continuous, high-accuracy positioning is required, providing a seamless transition between RTK and PPP correction sources for uninterrupted operation.

1.2 System Architecture of the A-ROX System

The A-ROX system is built on a flexible, modular hardware platform, allowing it to be tailored to the specific needs of different industries. The architecture consists of:

1. **GNSS-Receiver (s):**
    - Supports all-frequency, multi-constellation GNSS (GPS, Galileo, Beidou, Glonass)
    - Configurable with single, dual or triple GNSS antenna(s)
    - Receives real-time RTK and PPP (also Galileo High-Accuracy-Service (HAS) data) corrections for centimeter-level accuracy
2. **Inertial Measurement Unit (IMU):**
    - High-performance FOG-grade MEMS IMU for precise attitude determination
    - Enables dead reckoning when GNSS signals are lost
    - Delivers stable and drift-free navigation over extended periods
3. **Odometry Integration:**
    - CAN-FD interfaces for wheel-speed sensor data input and output
    - Enhances accuracy in tunnels, dense urban areas, and environments with multipath effects
4. **Processing & Sensor Fusion:**
    - Advanced Extended Kalman Filter (EKF) engine integrates GNSS, IMU, and odometry data

- Web app, easily accessible from different kinds of devices, including laptops and tablets. It is directly hosted on the system with no need for installation of software on your device
- ROS2 based software architecture
- Powerful forward-backward post-processing engine, including an automatic data-sync to the cloud-based backend, to further enhance the accuracy of the position, velocity and attitude solution with all recorded raw sensor data.
- Supports real-time processing at up to 200 Hz (1000 Hz optional)
- Outputs precise position, velocity, and attitude estimates

5. **Communication & Data Interfaces:**
   - Multiple connectivity options: 5G 2x2 MIMO, Gigabit Ethernet, Wi-Fi, USB-C, and CAN-FD (up to 4 channels)
   - Supports ROS2, FAST-API, ACOM and NMEA 0183 formats for data exchange
   - Provides cloud-based fleet management and remote monitoring via a web interface

6. **Time Synchronization & Corrections:**
   - PPS signal output for precise timing synchronization
   - NTRIPv2 and PTP (Precision Time Protocol) support for enhanced accuracy

1.3 Software Architecture of A-ROX System

A forward processed real time solution is calculated on the A-ROX. This is done by the ANavS PAD (Position- and Attitude Determination) engine, which processes all available sensor raw data using a tightly coupled sensor fusion.

First, raw data such as GNSS measurements, IMU measurements, odometry values, system status, and correction data are captured and published as topics by the corresponding software components. Afterwards the PAD software processes the precise position, velocity, attitude parameters (and many more) and publishes that as topics. Different output adapters can then subscribe to those topics (and raw values if needed) to output the solution in different formats, like CAN, NMEA or ACOM. The ANavS Command Center is a powerful web app which can access all topics by FastAPI and visualize them. Further it is possible to configure and manage or download recordings through it.

The following diagram illustrates the data flow of sensor values:

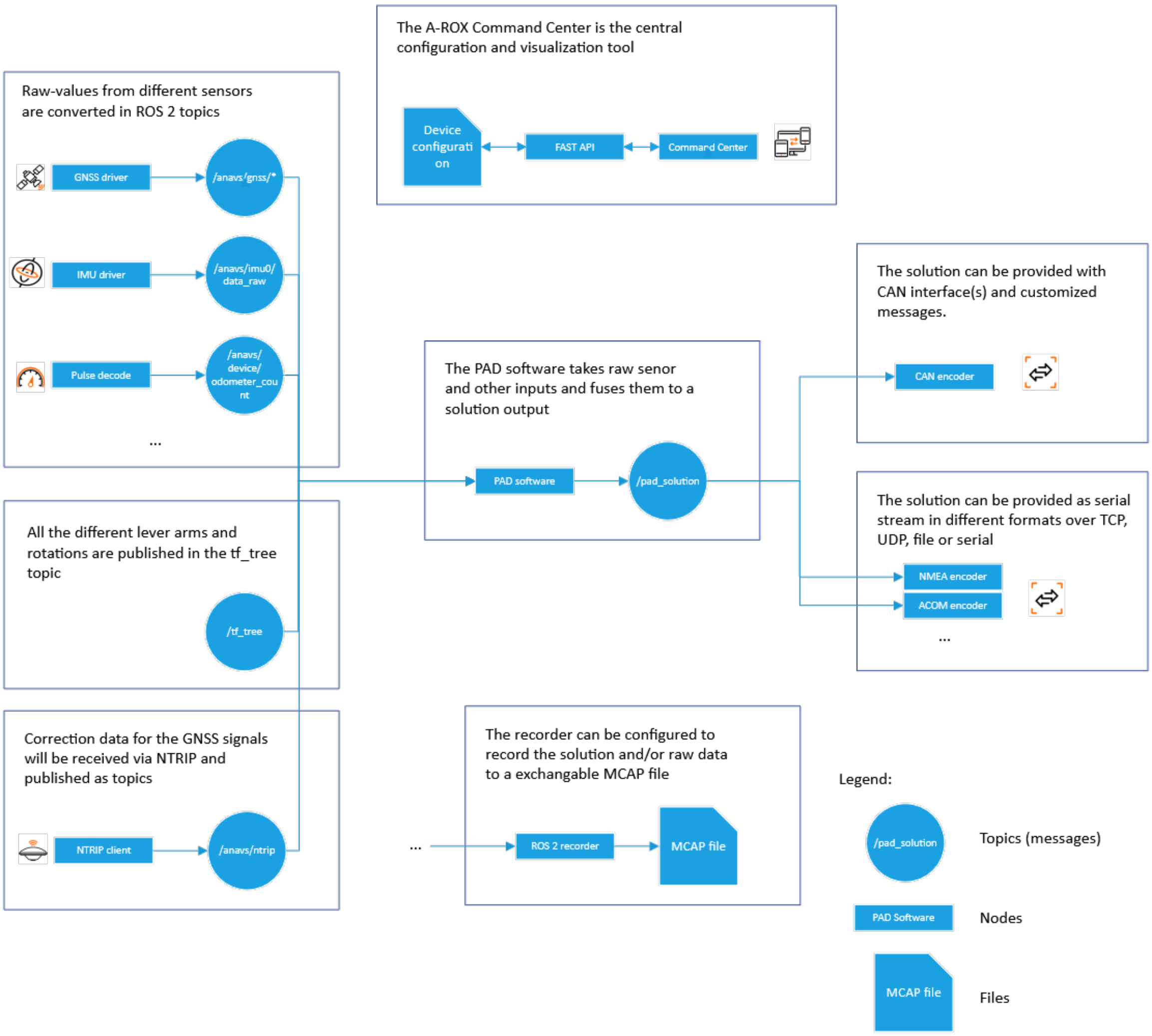


Figure 2: Data flow diagram of live sensor- and solution data

1.4 System Architecture of ANavS Control Hub (Cloud Service/Backend)

The ANavS Control Hub is a powerful extension to the A-ROX devices. It allows you to monitor in real time the solution and status information of your device remotely in the cloud, control settings and access the recordings.

The Control Hub syncs automatically (if active) the recordings to the server. Once uploaded, the synced raw sensor data is processed again for best possible solution results. It does so in two steps. First it generates a forward solution. The topic ***/pad_solution*** was already calculated on the A-ROX. It is generated on the server as well, benefiting from the extended processing power available.

This forward solution is then automatically passed through the postprocessing engine on the backend server, which calculates a forward-backward solution (Rauch-Tung-Striebel Smoother). The result is published in the topic ***/smoother_solution***. This is the best and final solution and can be used for further processing.

The diagram below illustrates this process: From the MCAP file synchronized from the A-ROX, the sensor topics are published and processed by the PAD software. The output of this process is written back into the MCAP file and forwarded to the smoother (forward-backward postprocessing engine). The smoother solution is also written into the MCAP file.

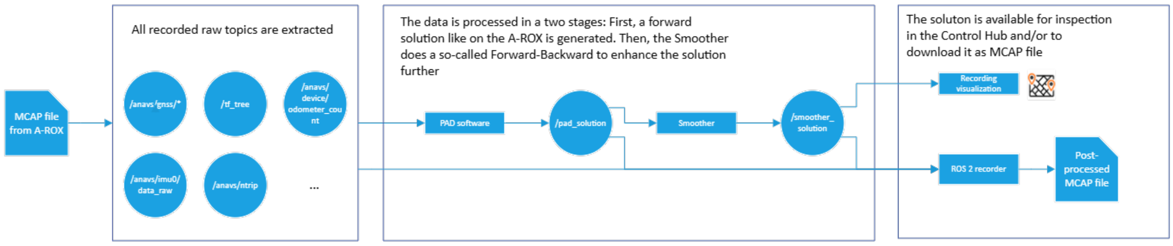


Figure 3: Data flow diagram in the Control Hub

2 Technical Specifications

This chapter provides a detailed breakdown of the hardware, communication interfaces, power requirements, positioning accuracy, time synchronization capabilities, and supported data formats.

2.1 Positioning and Accuracy[1]

A-ROX offers cutting-edge positioning accuracy, leveraging RTK and PPP technologies for centimeter-level precision.

Mode	Horizontal Accuracy (1 Sigma)	Vertical Accuracy (1 Sigma)
RTK	0.006 m	0.010 m
PPP	0.200 m	0.400 m

- RTK Initialization Time: < 7 seconds
- PPP Initialization Time: < 120 seconds
- Velocity Accuracy: 0.03 m/s RMS
- Slip Angle Accuracy: 0.15°
- Attitude Accuracy:
    - Without calibration (2m antenna spacing): 0.10° (Roll/Pitch), 0.10° (Heading)

[1] Depends on environment and quality of used GNSS antenna(s)

- With dynamic pre-calibration: 0.02° (Roll/Pitch), 0.05° (Heading)

2.2 Hardware Specifications

Component	Specification
GNSS Receiver(s)	Multi-constellation, All-frequency (GPS, Galileo, Beidou, Glonass)
Correction Data	RTK (Real-Time Kinematics), PPP (Precise Point Positioning), including HAS-IDD and HAS-SiS
IMU	FOG-grade MEMS IMU, different variants available
Odometry Input	Wheel ticks (rotary encoder) or speed from CAN-FD interface
Data Storage	32 GB onboard Flash (expandable up to 1 TB with SSD)
Communication	5G 2x2 MIMO cellular network, Gigabit Ethernet, Wi-Fi, USB-C, up to 4 CAN-FD channels
Power Consumption	9-36V (Nominal: 12-24V), average: 9W, peak: 16W
IP-Rating	IP65 (dustproof and water-resistant)
Operating Temperature	-20°C to 65°C
Weight	1.2 kg
Dimensions	174 x 123 x 60 mm

2.3 Connectors

On the back side of the A-ROX, all connectors for power- and data transmission can be found.

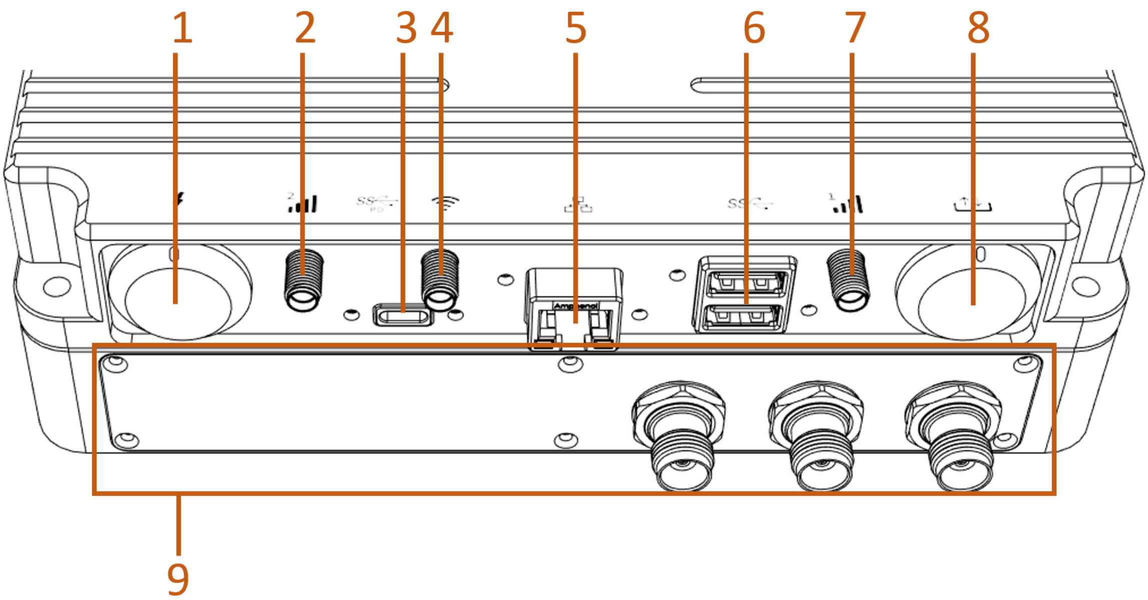


Figure 4 View from back with all connectors

<table>
<tr><th>Connector</th><th>Description</th></tr>
<tr><td>1</td><td>Lemo Power Input
Connector type: Lemo EGG.2T.302.CYM
Mate with: Lemo FGG.2T.302.CLAC85ZN
Pinout:<table><tr><td>Pin No:</td><td>Signal</td></tr><tr><td>1</td><td>PWR-IN</td></tr><tr><td>2</td><td>PWR-GND</td></tr></table></td></tr>
<tr><td>2</td><td>Mobile Radio 2
Secondary antenna connector for mobile radio communication module (5G)
Connector type: SMA female
Mate with: SMA male</td></tr>
<tr><td>3</td><td>USB-C PD
USB-C connector to power the module with USB-PD and data connection via USB 3.1
Supported USB-PD profiles: VMIN = 12V, VMAX = 20V, ISNK = 3A</td></tr>
<tr><td>4</td><td>Wi-Fi antenna connector
Connector type: SMA female
Mate with: SMA male</td></tr>
<tr><td>5</td><td>Ethernet
Gigabit ethernet RJ45 connector. Default IP setting is DHCP client. Can be changed in Command Center</td></tr>
<tr><td>6</td><td>USB Device connector
2x USB-A ports with 3.1 standard for connecting external peripherial devices</td></tr>
<tr><td>7</td><td>Mobile Radio 1
Primary antenna connector for mobile radio communication module (5G)
Connector type: SMA female
Mate with: SMA male</td></tr>
</table>

<table>
<tr><td>8</td><td>Lemo Power Input
Connector type: Lemo EGG.2T.316.CLL
Mate with: Lemo FGG.2T.316.CLAC70ZN
Pinout:
<table>
<tr><td>Pin No:</td><td>Signal</td></tr>
<tr><td>1</td><td>GND</td></tr>
<tr><td>2</td><td>PPS_out</td></tr>
<tr><td>3</td><td>GND</td></tr>
<tr><td>4</td><td>PPS_in</td></tr>
<tr><td>5</td><td>GND</td></tr>
<tr><td>6</td><td>GPIO1</td></tr>
<tr><td>7</td><td>GPIO2</td></tr>
<tr><td>8</td><td>GPIO3</td></tr>
<tr><td>9</td><td>GPIO4</td></tr>
<tr><td>10</td><td>Reserved</td></tr>
<tr><td>11</td><td>CAN1_GND</td></tr>
<tr><td>12</td><td>CAN1_Hi</td></tr>
<tr><td>13</td><td>CAN1_Low</td></tr>
<tr><td>14</td><td>CAN2_GND</td></tr>
<tr><td>15</td><td>CAN2_Hi</td></tr>
<tr><td>16</td><td>CAN2_Low</td></tr>
</table>
Note: CAN busses are unterminated. External termination adapter or terminated cables must be used</td></tr>
<tr><td>9</td><td>GNSS Antenna
Up to three (depending on model) GNSS input connectors. All provide a LNA voltage of 5V and are short circuit and ESD protected
Connector type: TNC female
Mate with: TNC male</td></tr>
</table>

2.4 Power Requirements

The A-ROX system is designed for versatile power supply options, making it suitable for both vehicle, railway and many more applications.

Power Source	Specification
USB-C	USB-C PD (requires 12-20V/3A)
LEMO Connector	9-36V DC (Nominal: 12-24V DC)

It is possible to connect simultaneously two different power sources. The A-ROX will always use the one with the higher voltage and supports seamless handover when one power source fails. This can be used e.g. to connect a power bank via USB-C and supply the A-ROX with 24 V DC. When DC-In powers off, the system stays powered from USB-PD.

The typical power consumption is around 9-10W with peak usage up to 20W. A power supply providing minimum 20W needs to be used.

2.5 Reference point

The A-ROX outputs the position and attitude for a virtual point in its bottom left corner, the so-called reference point. All dimensions are referring to it.

From this point you can define the so-called Point-of-Interests (POI). These additional virtual reference points allow you to get the position and attitude of a remote point, relative to the origin reference point.

Figure 5: Virtual reference point on the bottom left corner

The A-ROX uses a local box coordinate system. This system is printed on the housing and serves as a reference. The solution can be rotated and moved output in the vehicle coordinate system, which always refers to the box coordinate system. This can be configured by the user through settings in the Command Center (section 3.2.5).

Internal components may be installed in different orientations and therefore initially have a different coordinate system. The software accounts for this, making it mostly irrelevant for the user. However, when analyzing raw values, this must be taken into consideration. The correct transformations from the sensor coordinate system to the box coordinate system are stored in the ***tf_tree***.

3 Description of the A-ROX Command Center (Local Frontend)

The following chapter describes the A-ROX Command Center, a web-based user interface accessible via LAN or Wi-Fi directly from a browser. This interface allows users to configure the system, monitor sensor data, and manage recordings. If you are connected via the integrated Wi-Fi Access Point, you can access it by default via https://192.168.42.1/.

A modern browser needs to be used which supports Modern JavaScript and provides full WebGL2 support. Check here if your browser is on the minimum supported list. Note that using the up-to-date versions of these browsers will ensure the best experience with maximized performance of the various interactive and data-intensive functionalities by the Command Center. NO internet explorer support and as of the nature of modern browsers we recommend a minimum of 16GB of RAM and plenty of processing power.

3.1 Home

The Home tab provides a comprehensive system overview and serves as the main dashboard for monitoring the real-time status of the device. It includes various sections displaying key information about the hardware, software, and sensor data.

3.1.1 Overview

At the top, users can see the system's status, including hardware information, software version, and connectivity state. A map view is integrated into the dashboard, showing the device's current location and trajectory. Additionally, a log panel records system messages and status updates, helping users track recent events. Each module on the dashboard can be expanded to reveal detailed information and real-time plots of data from the last two minutes.

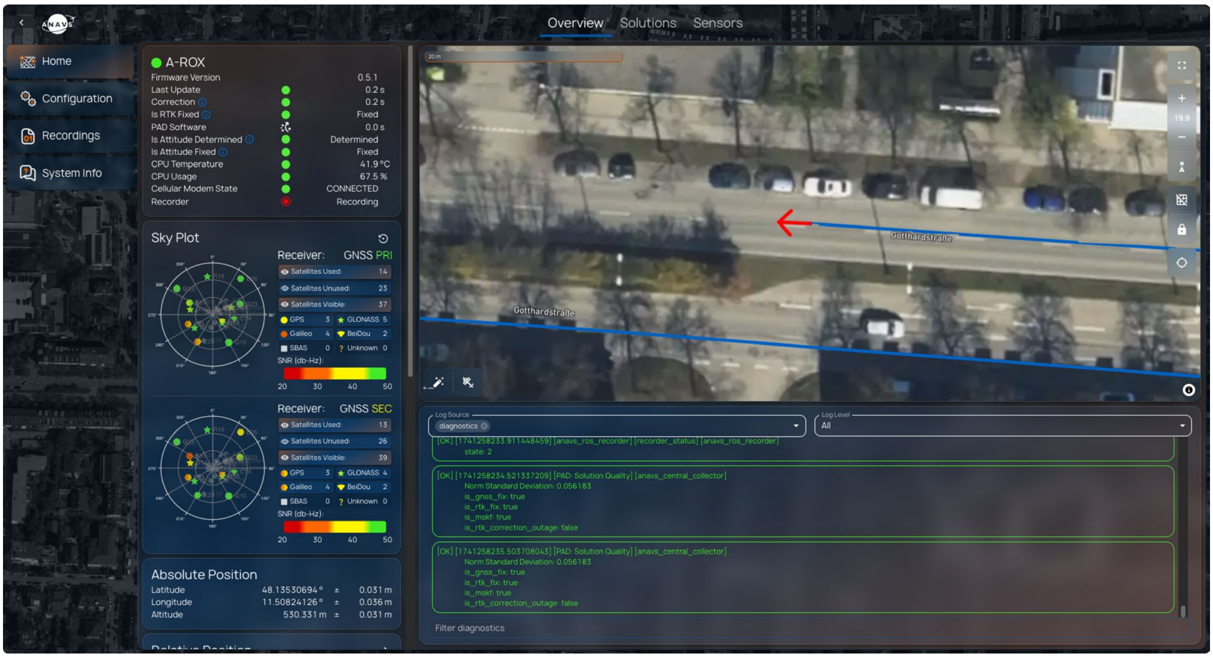


Figure 6: Landing page of the Command Center including a hardware, software and sensor overview with Map and Logs

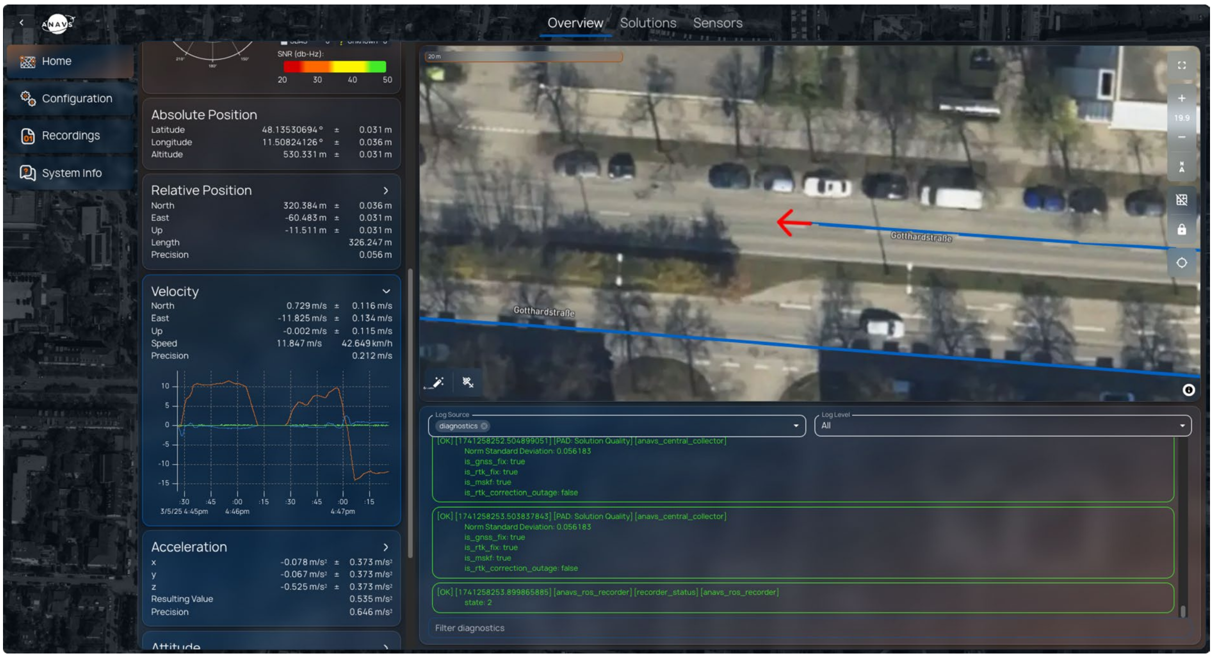


Figure 7: Selected solution containers can be expanded to reveal a plot of the data from the last two minutes

3.1.2 Solutions

An important page of the Home tab is the Solutions section, where users can access detailed insights into the system's calculations. Each of these categories includes expandable containers that reveal live plots of the last two minutes of data, allowing users to monitor trends and detect anomalies.

3.1.2.1 Position

This page shows the estimated location in the local NED coordinate system with standard deviation.

Figure 8: Relative position sensor page with std. dev, length

3.1.2.2 Velocity

This page displays the velocity in the local NED coordinate system with standard deviation.

Figure 9: PAD Solution Velocity page with speed and std.

3.1.2.3 Acceleration

This page displays real-time acceleration, adapted regarding bias, lever-arm and misalignment (also using a DaKKS calibration), on all three axes (X, Y, Z).

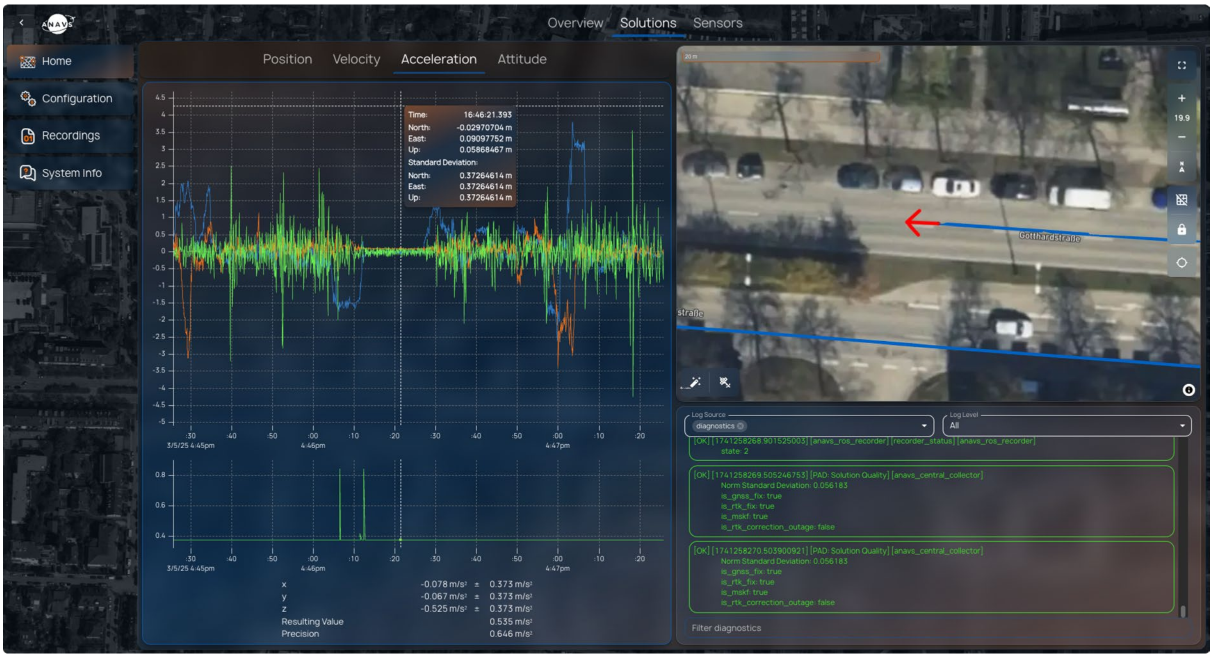


Figure 10: PAD Solution Acceleration page with std dev. and resulting value

3.1.2.4 Attitude

This page visualizes roll, pitch, and yaw angle of your vehicle.

Figure 11 PAD Solution Attitude page with std. dev

3.1.3 Home – Sensors

Additionally, the Sensors section provides raw measurement data, such as GNSS signal reception across multiple satellite constellations (GPS, Galileo, BeiDou, and GLONASS), frequencies and overall equipped GNSS receivers (up to 3). This is crucial for assessing the quality and reliability of the positioning solution. For example, it is also possible to directly detect whether the GNSS signals are being interfered with by other sensors (e.g. LiDAR).

Overall, the Home tab acts as the central hub for monitoring the A-ROX system's performance, checking real-time data, and diagnosing potential issues, ensuring seamless operation in high-precision navigation applications.

3.1.3.1 GNSS 1 / Multi-GNSS Raw Measurement Data

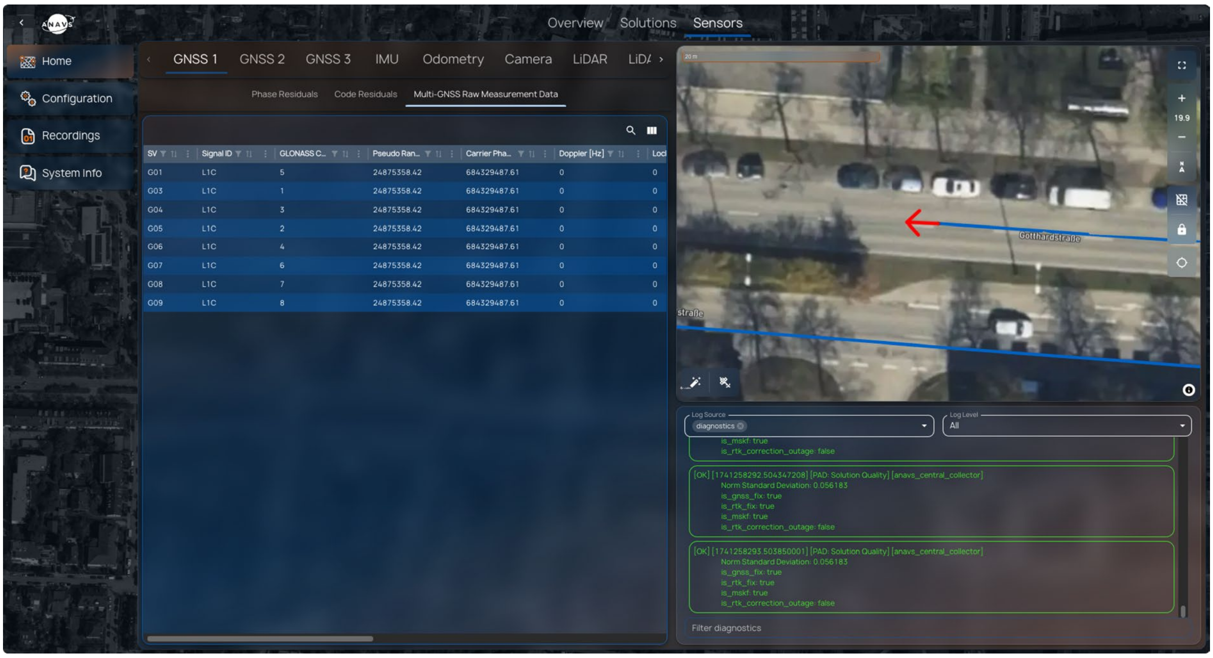


Figure 12: Multi-GNSS RAW Measurement Data page

3.2 Configuration

The Configuration tab in the A-ROX Command Center is a crucial section where users can set up and customize various system parameters, including connectivity, data input and output, algorithms, and device dimensions. It ensures that the system is correctly integrated with external networks, sensors, and vehicles, optimizing its performance for high-precision navigation applications.

3.2.1 Connectivity

The Connectivity section allows users to configure the different communication interfaces available on the A-ROX system.

3.2.1.1 Cellular

Users can manage the built-in modem for mobile network communication. The interface displays key details such as signal strength, connection status, network type (e.g., 4G/5G), and SIM card information. This is essential for applications requiring real-time corrections via NTRIP or remote access to the system.

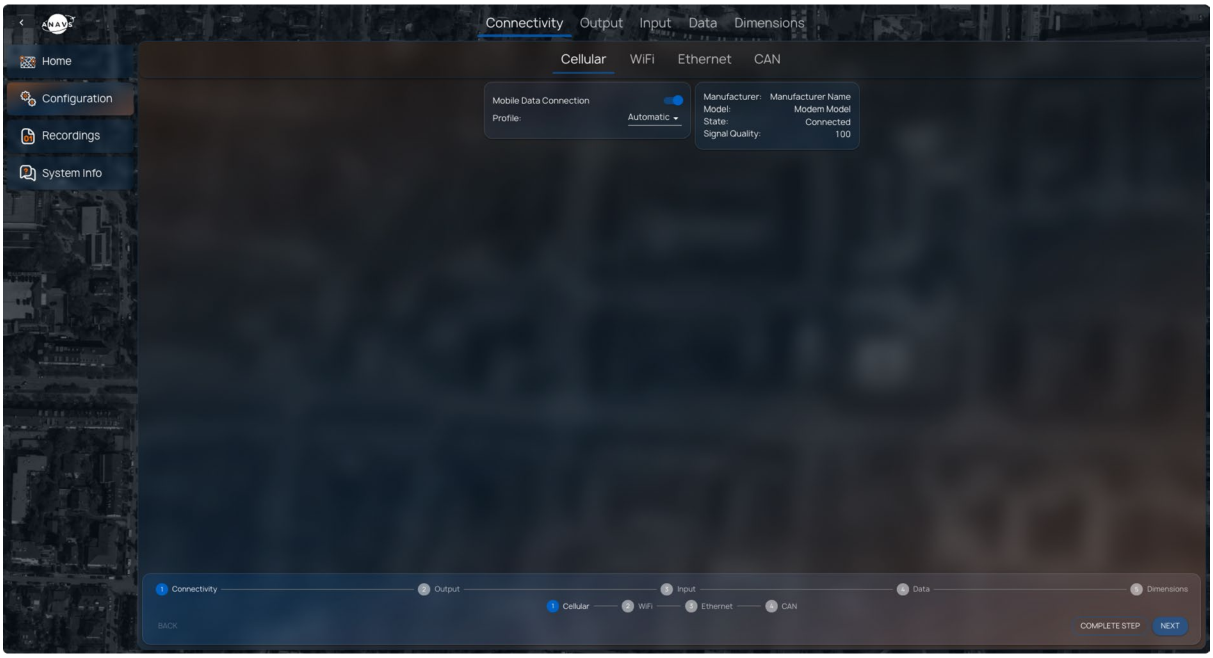


Figure 13: Cellular Modem Connectivity configuration form with modem information and current state

3.2.1.2 Wi-Fi

The A-ROX device can operate as either a Wi-Fi access point or a client. The Wi-Fi settings page allows users to select networks, enter credentials, and check the connection status. A green indicator confirms successful connectivity.

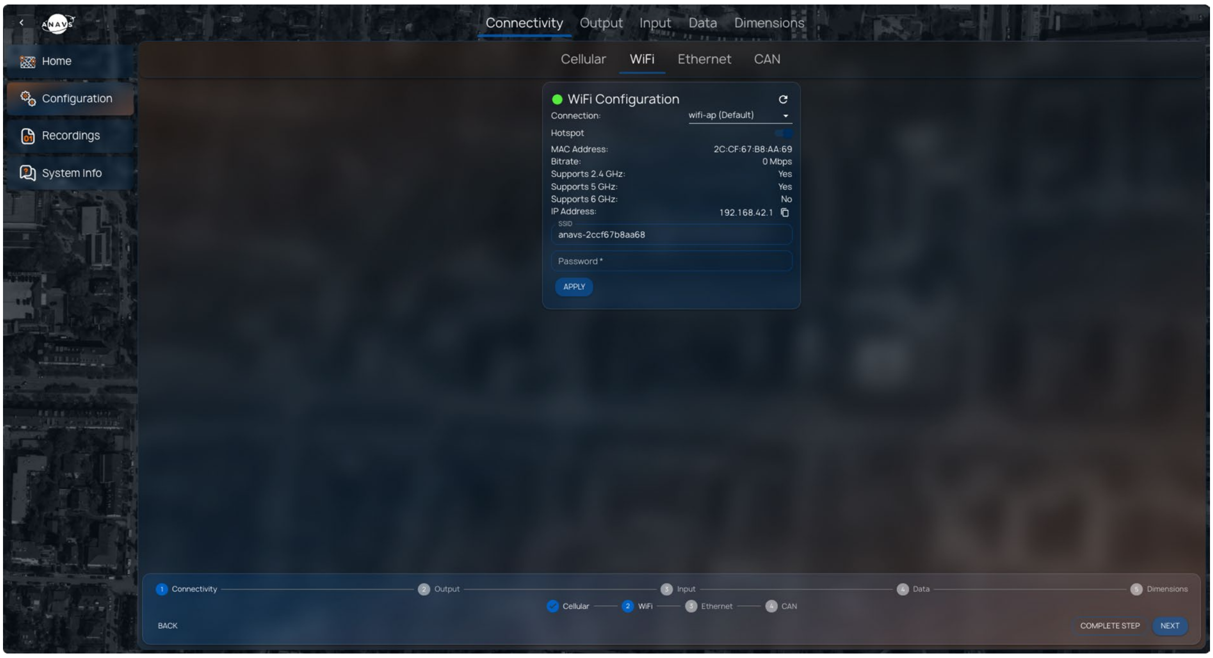


Figure 14: Wi-Fi connectivity configuration page; green icon indicating availability state

3.2.1.3 Ethernet

If a wired network connection is preferred, users can configure Ethernet settings, including DHCP or static IP configuration. This option is useful for stable, high-speed data transfer setups with low latency requirements.

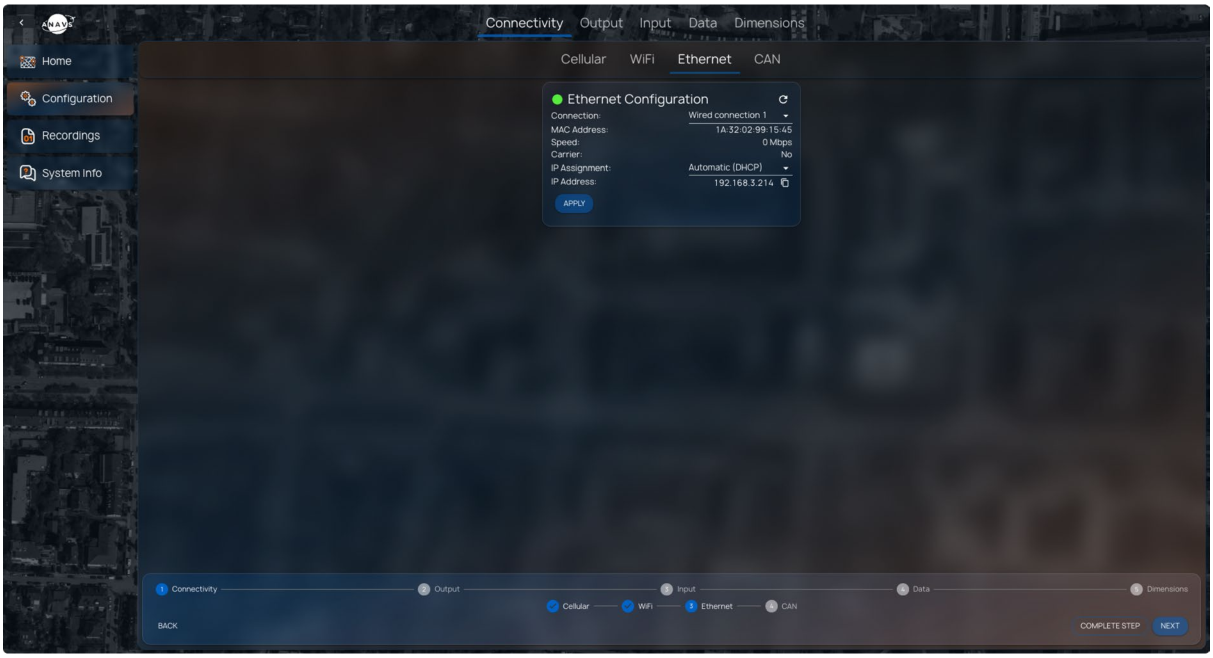


Figure 15: Ethernet connectivity configuration page displaying availability and current state

3.2.1.4 CAN

The CAN (Controller Area Network) settings enable communication with vehicle sensors, odometry systems, and other external devices. This is where users can define CAN-FD parameters, ensuring the system correctly interprets and transmits vehicle data. Detailed information in chapter 5.3.

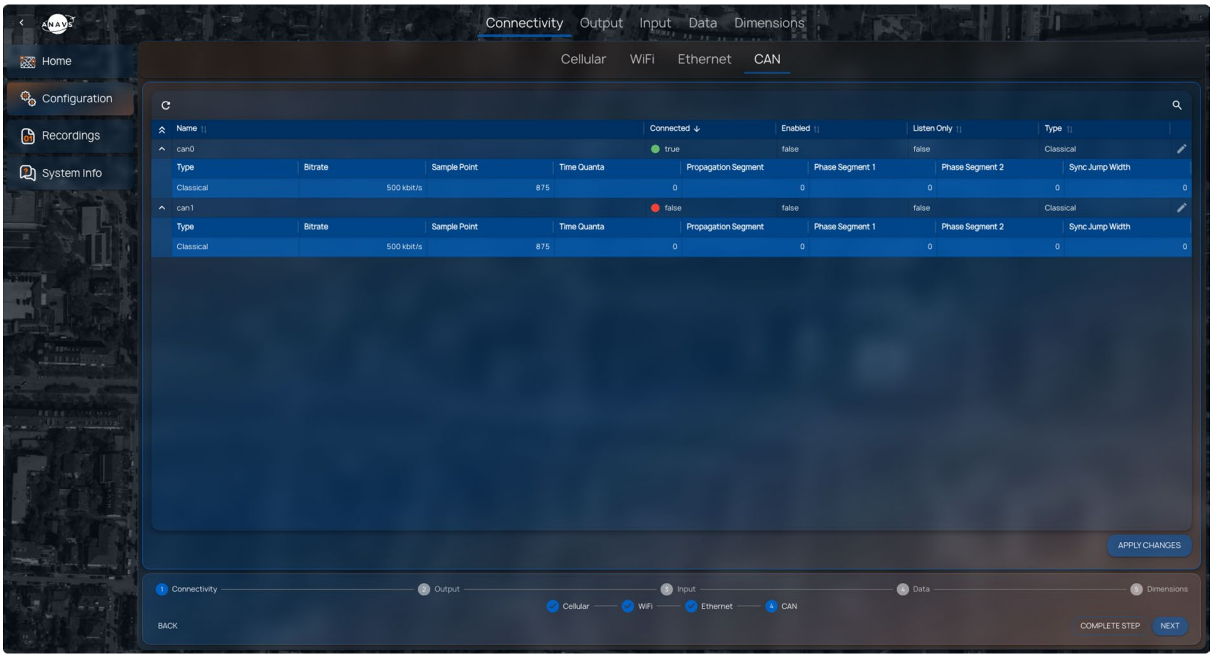


Figure 16: CAN connectivity configuration page

3.2.2 Output

The Output section defines how data is transmitted from the A-ROX system.

3.2.2.1 Recording

This section configures the data logging settings. Users can specify which GNSS, IMU, and odometry data should be recorded, as well as set storage preferences.

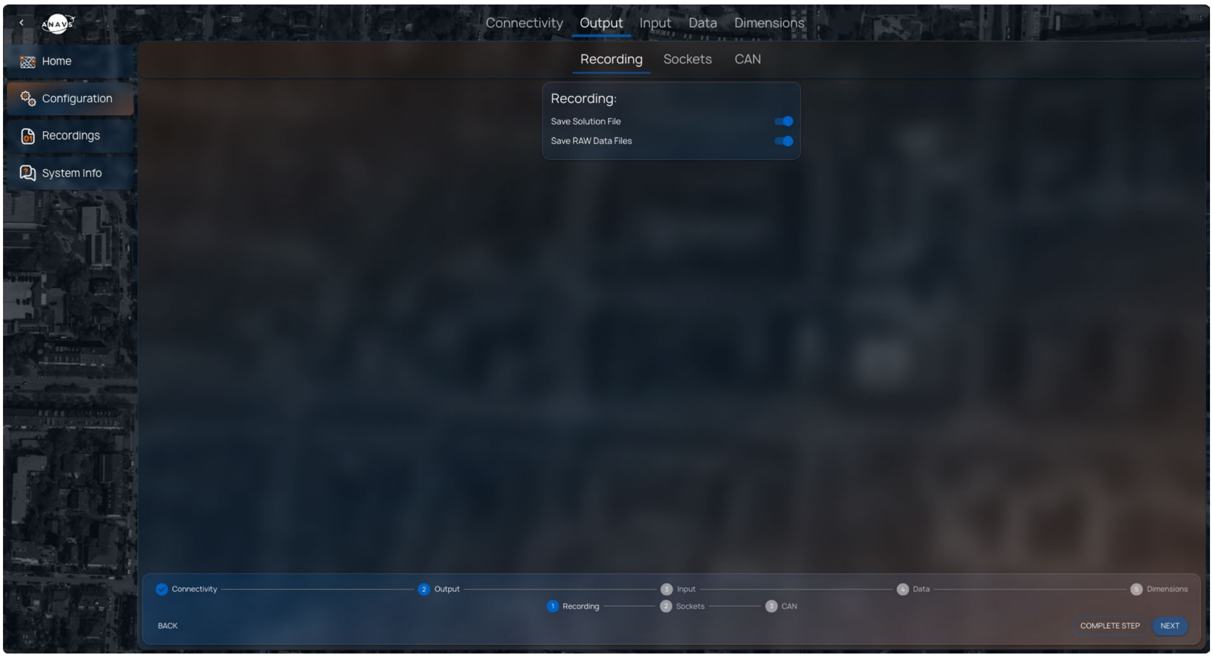


Figure 17: Recording Output configuration page

3.2.2.2 *Sockets*

The system supports real-time data transmission via TCP or UDP sockets, using formats such as ACOM and NMEA. This configuration allows external applications to access positioning data in real-time.

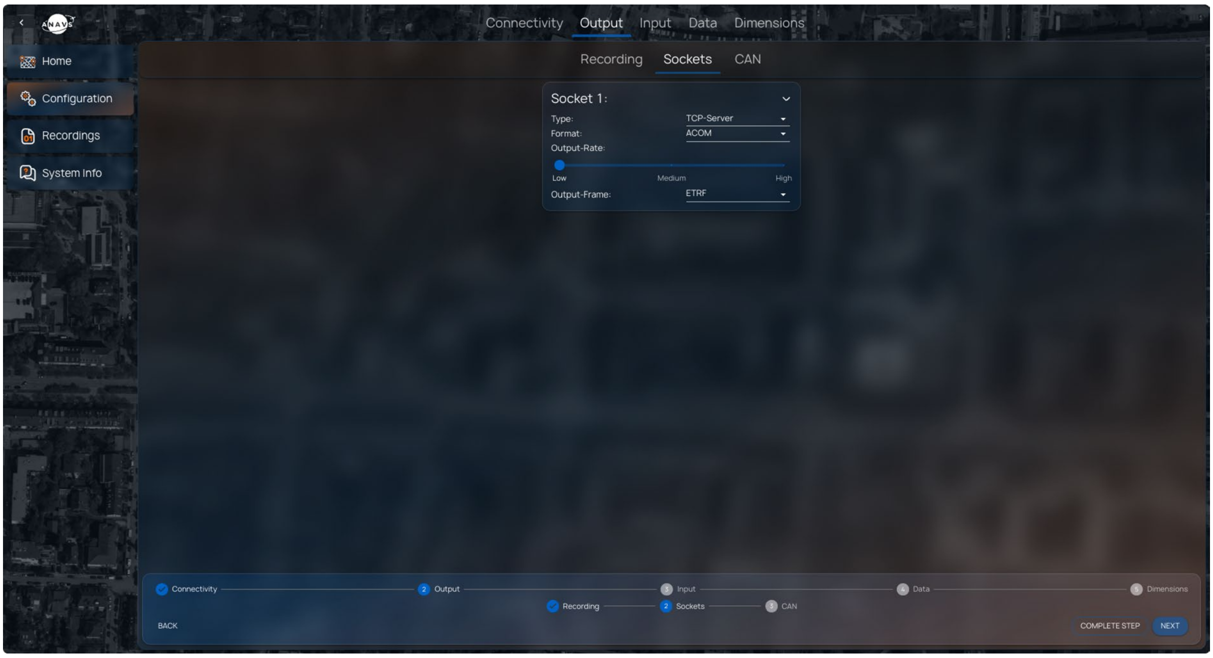


Figure 18: Socket Output configuration page providing the socket types (TCP, UDP) and the socket formats (ACOM, NMEA)

3.2.2.3 CAN

The A-ROX system can send calculated position, velocity, attitude and many more data over the CAN bus. Users can define custom CAN messages by mapping data fields and applying scaling and offset parameters. Additionally, DBC file exports allow seamless integration with vehicle networks. Detailed information can be found in chapter 5.3.

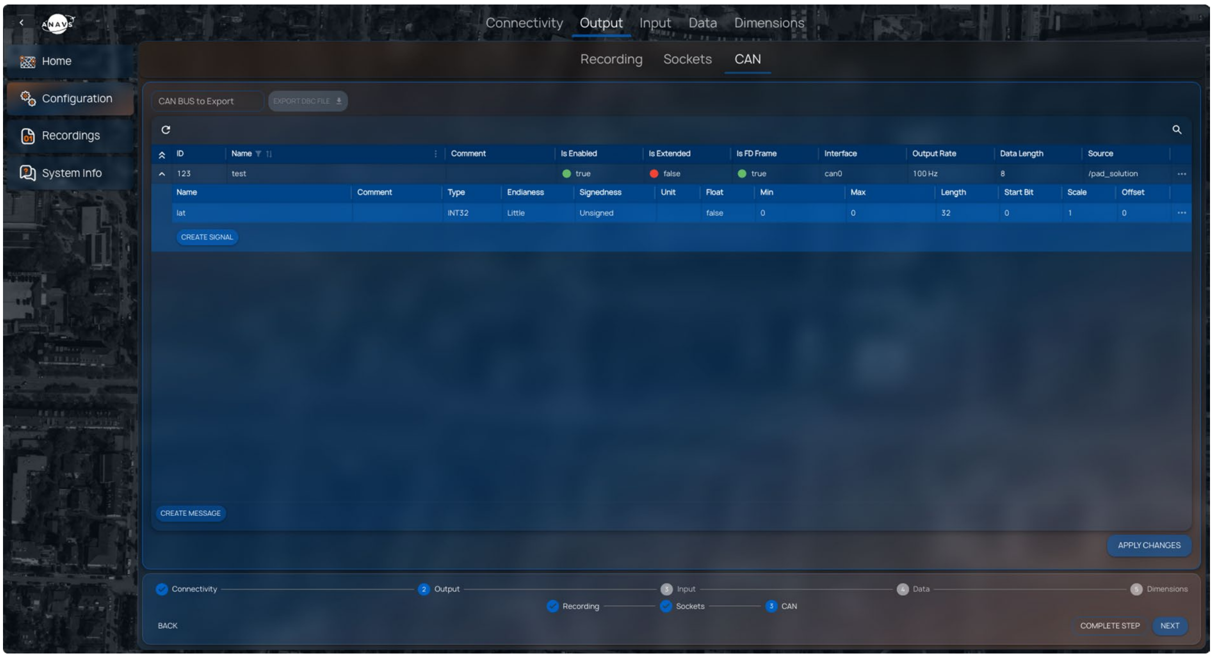


Figure 19: CAN output message configuration page with optional dbc file export

3.2.3 Input

The Input section allows users to configure external data sources that enhance positioning accuracy.

3.2.3.1 CAN Odometry

The A-ROX system with up to 4 CAN-FD channels can directly feed the wheel speed and steering information into the PAD engine. However, a matching DBC file from the vehicle is always required to set the necessary information in this dialog.

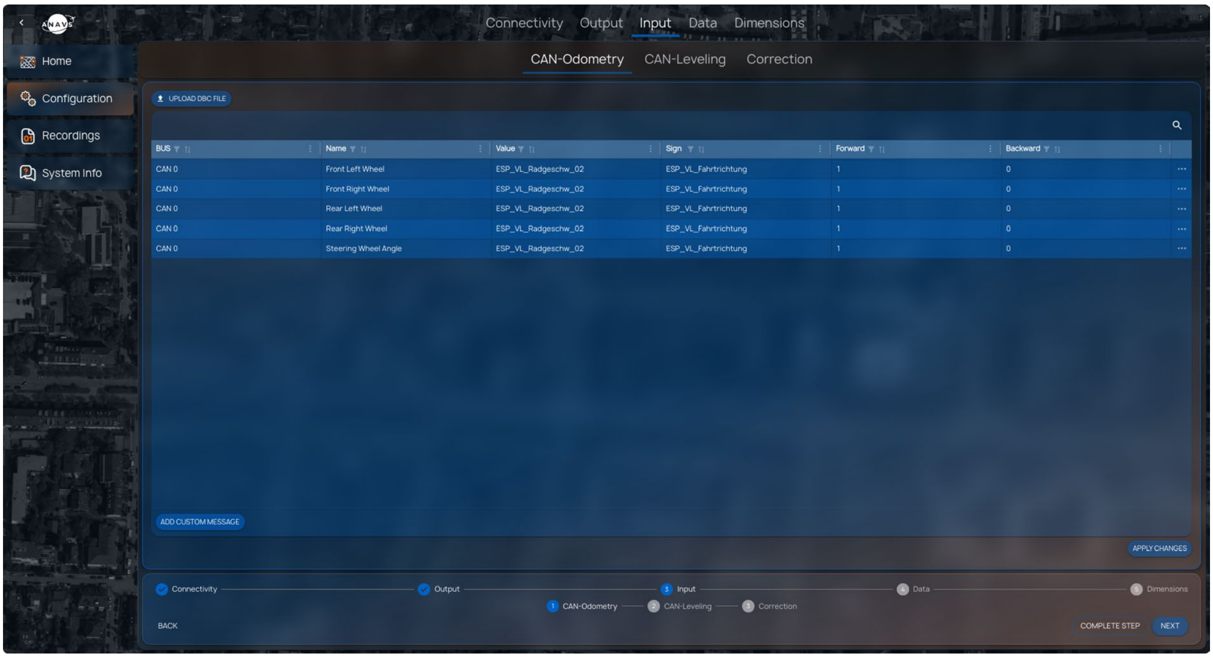


Figure 20: CAN Odometer input message mapping page

3.2.3.2 CAN Leveling

3.2.3.3 Correction

This section is critical for RTK (Real-Time Kinematic) and PPP (Precise Point Positioning) corrections. The A-ROX system includes an NTRIPv2 client, which allows users to configure an RTCM 3.X correction data stream for centimeter-level accuracy. Users can specify NTRIP credentials, caster addresses, and mount points.

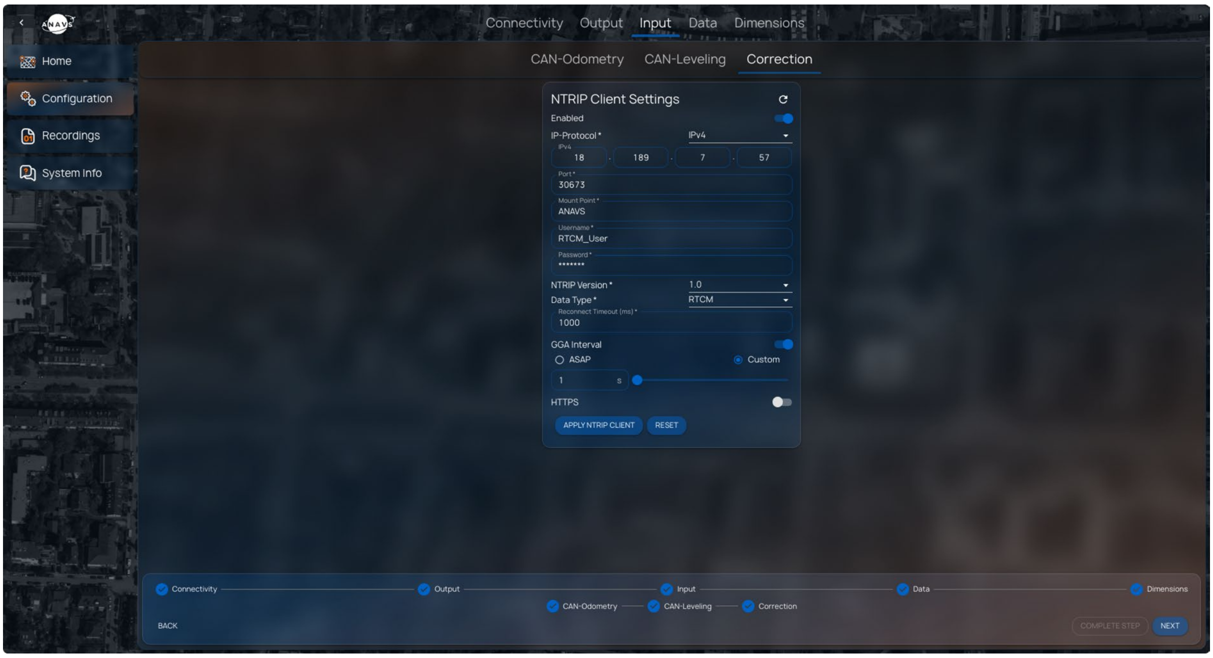


Figure 21: Correction data input configuration page enabling NTRIP client

3.2.4 Data

This section fine-tunes the sensor fusion algorithm and system behavior.

3.2.4.1 Algorithm

Users can adjust parameters of the PAD (Position and Attitude Determination) engine, which processes data from GNSS, IMU, and odometry sensors. This ensures optimal performance across different motion profiles and environments.

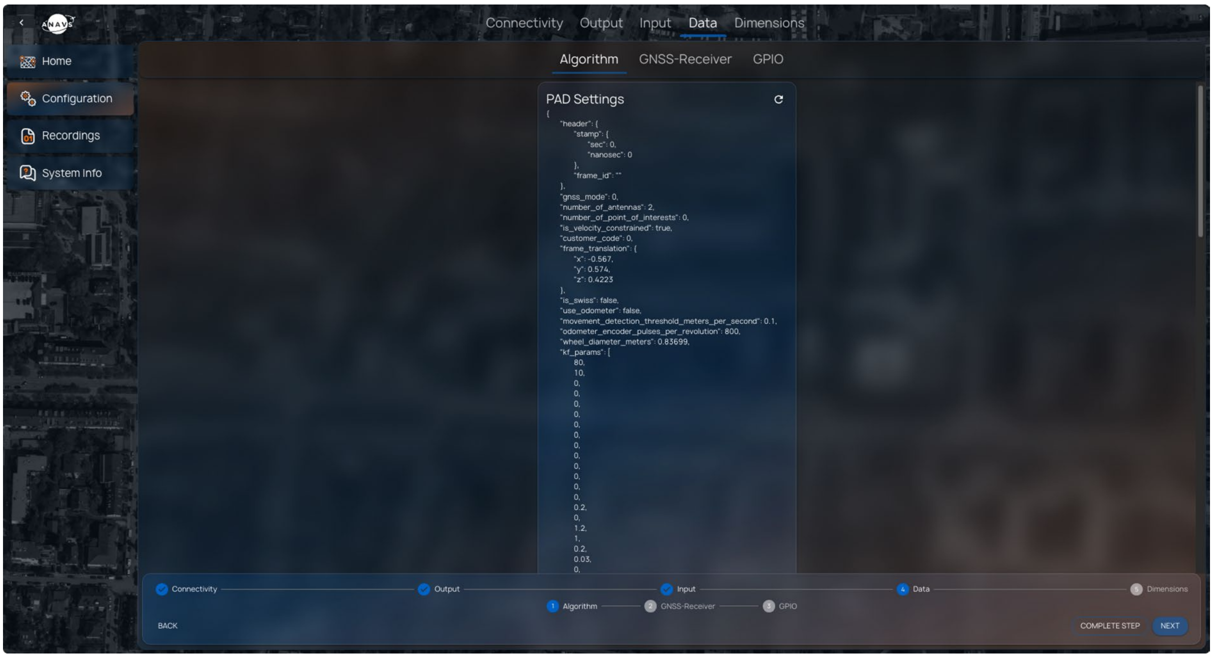


Figure 22: Algorithm data page for viewing the PAD Settings

3.2.4.2 GNSS-Receiver

The GNSS settings page allows enabling or disabling specific frequency bands (L1, L2, L5, etc.) from different satellite constellations. This is useful for optimizing performance based on regional signal availability or interference conditions.

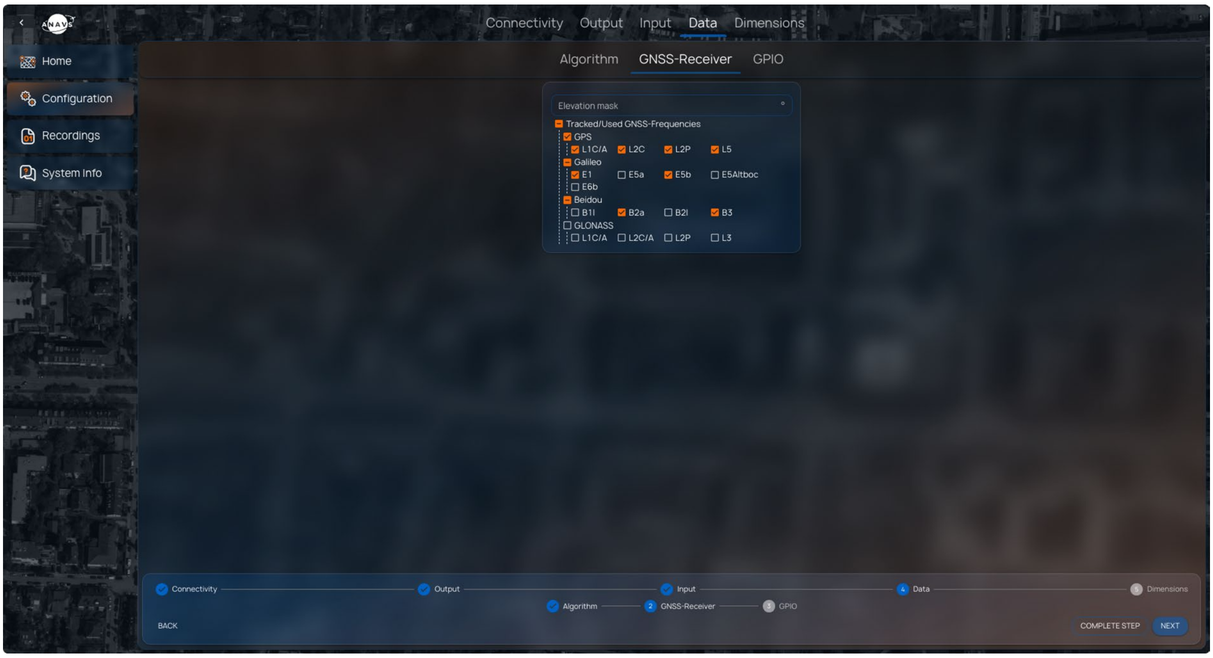


Figure 23: GNSS Receiver data configuration page for en-/disabling GNSS-Frequencies

3.2.4.3 GPIO

The A-ROX device has four configurable GPIO (General Purpose Input/Output) ports. These can be used for external sensor input, event triggers, or output signals. A typical use case is connecting a quadrature encoder for wheel tick input for improved sensor fusion performance.

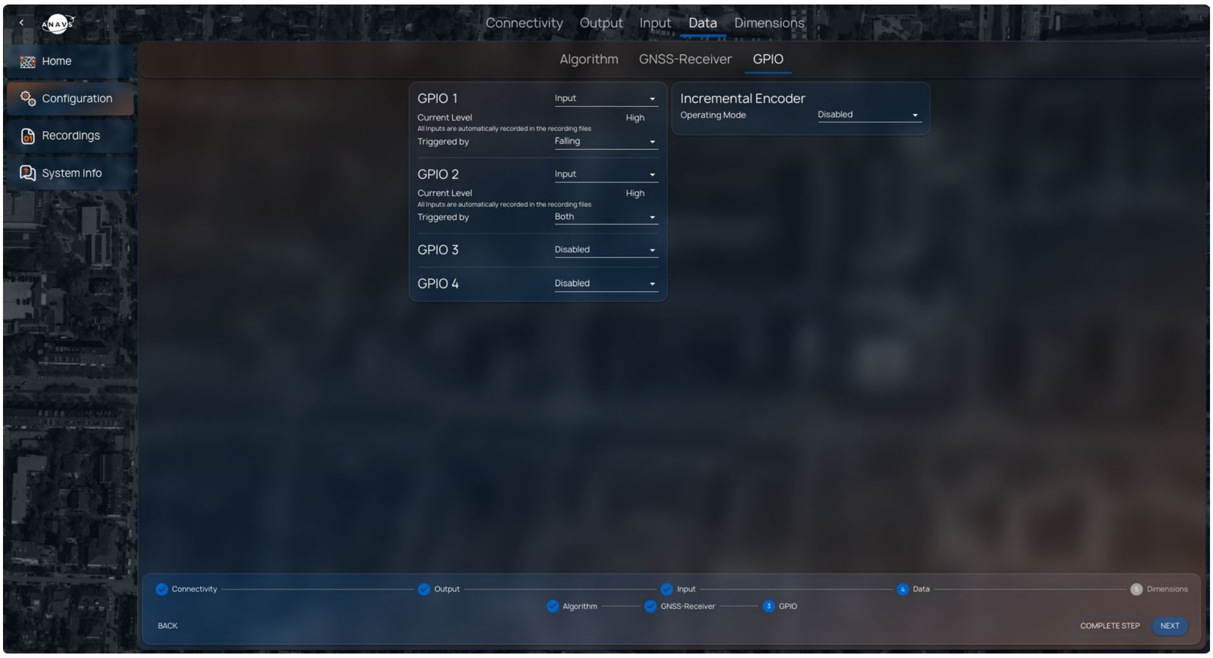


Figure 24 GPIO data configuration page

3.2.5 Dimensions

This is a very important step of the setup and has direct impact on the Position- and Attitude performance of the sensor fusion. The placement of the GNSS antennas defines the attitude-baseline, that means the yaw, pitch, and roll angle of your rover. Please mount the antennas in accordance with your use-case and carefully measure the distances between the GNSS antennas (measurement-error <= 1 cm).

The interface includes a 3D model of the A-ROX system, helping users verify the entered dimensions and alignment before deployment. Once confirmed, the system automatically applies these parameters in the PAD software for sensor fusion.

Please pay attention to the following:

- The minimum distance between two GNSS antennas is 30 cm.
- Please note that the attitude accuracy increases with the distance between the GNSS antennas (sigma=0.25° absolute attitude-accuracy per meter antenna spacing). That means at 2-meter baseline-length a sigma of 0.125° for the heading/pitch/roll.
- Please align your antennas in the same way to minimize the impact of antenna phase center offsets. The high-grade antennas have typically an arrow marked on the bottom or rear side of the antenna. The orientation of the low-cost patch antennas is simply defined by the orientation of the cable-connector.
- Please consider and measure also a height-offset between the mounted GNSS antennas. We recommend a placement at the same height level for the beginning as this prevents any sign errors of the differential height and typically enables the best performance.

- For setups with three GNSS antennas (3D-Setup for yaw, pitch, and roll angle determination), please make sure that distance measurements in perpendicular directions are having a 90° angular spacing as otherwise the distance measurements do not fit to the actual geometry. This would prevent any reliable solution.

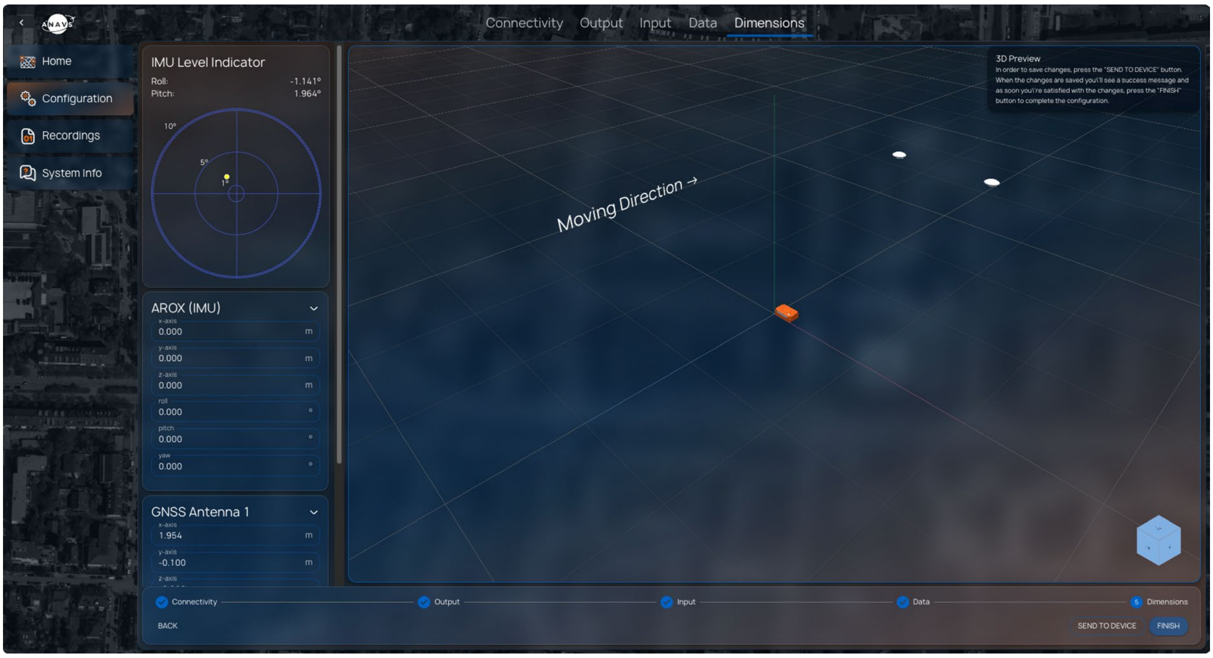


Figure 25: Dimensions configuration page with 3D space component representation and live level

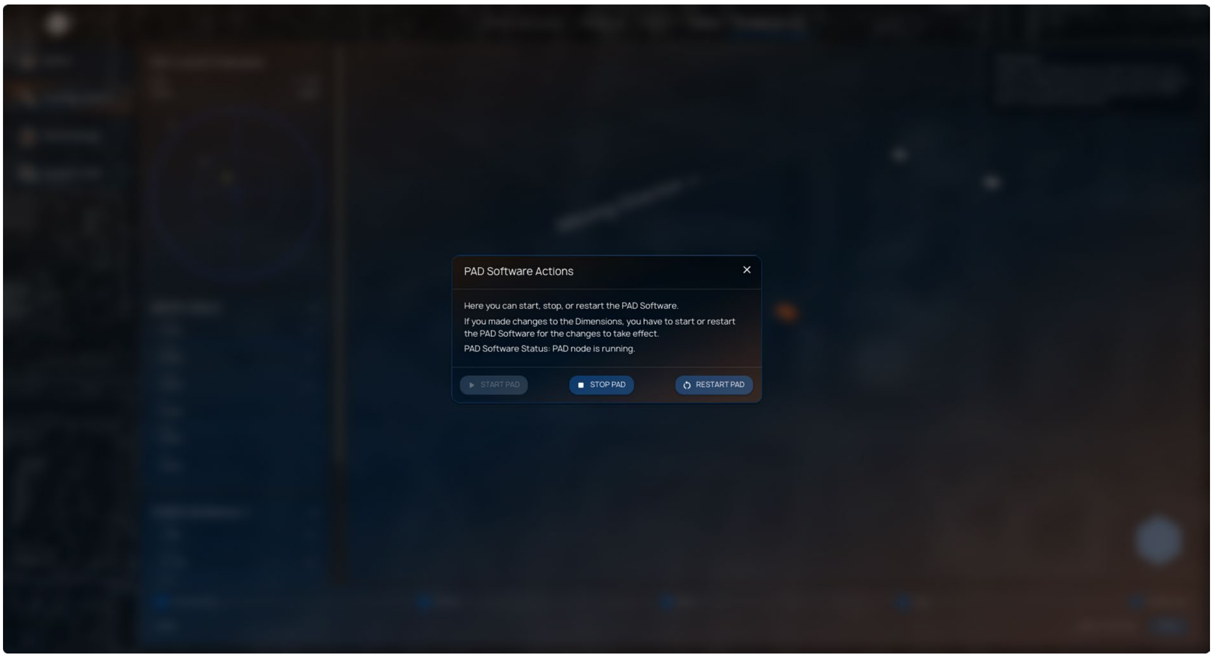


Figure 26: Finished Dimensioning Modal providing instructions/Options for PAD Software handling

3.3 Recordings

The Recording Browser provides an overview of all stored GNSS, IMU, and odometry recordings. Users can browse, download, or delete sessions, making it easier to manage and analyze historical data.

3.3.1 Recording Browser

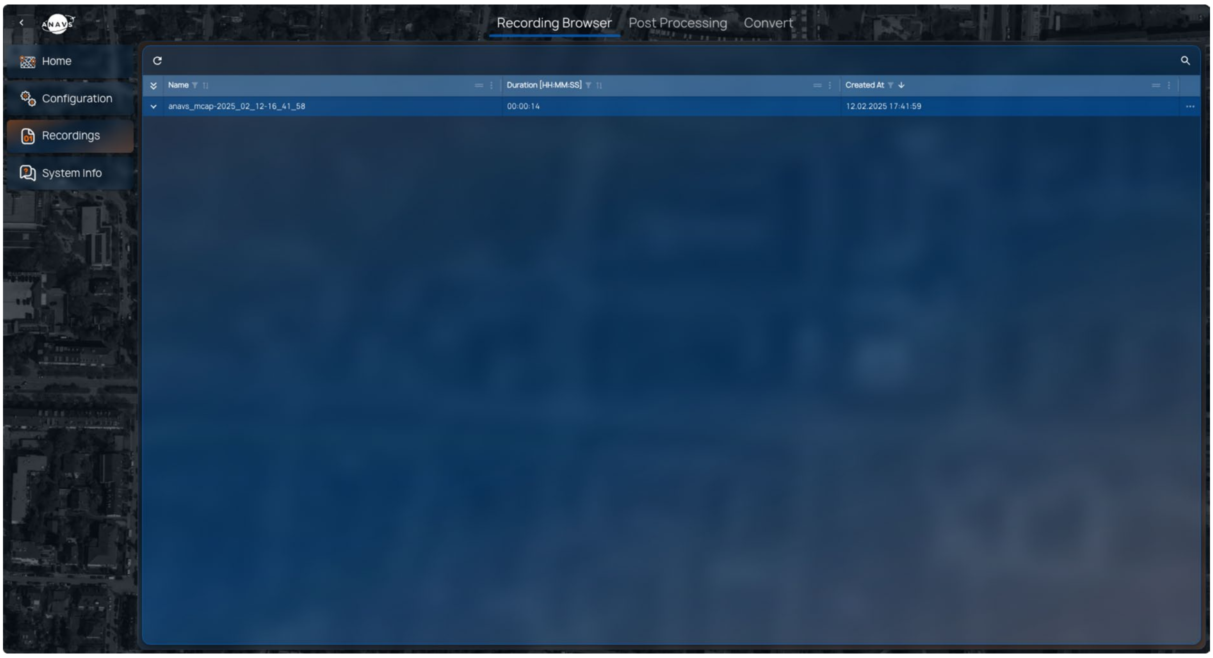


Figure 27: Recordings Browser page for downloading recorded sessions

4 Description of the A-ROX Control Hub (Cloud Service/Backend)

The following chapter provides a detailed description of the A-ROX Control Hub, the cloud-based backend system that enables remote monitoring, configuration, and data management for the A-ROX device. The Control Hub is accessible via ***https://hub.anavs.com/*** and allows users to manage fleets, post-process recorded data with the powerful forward-backward engine and configure devices from anywhere.

Your data is hosted on servers in Europe and protected via 2FA.

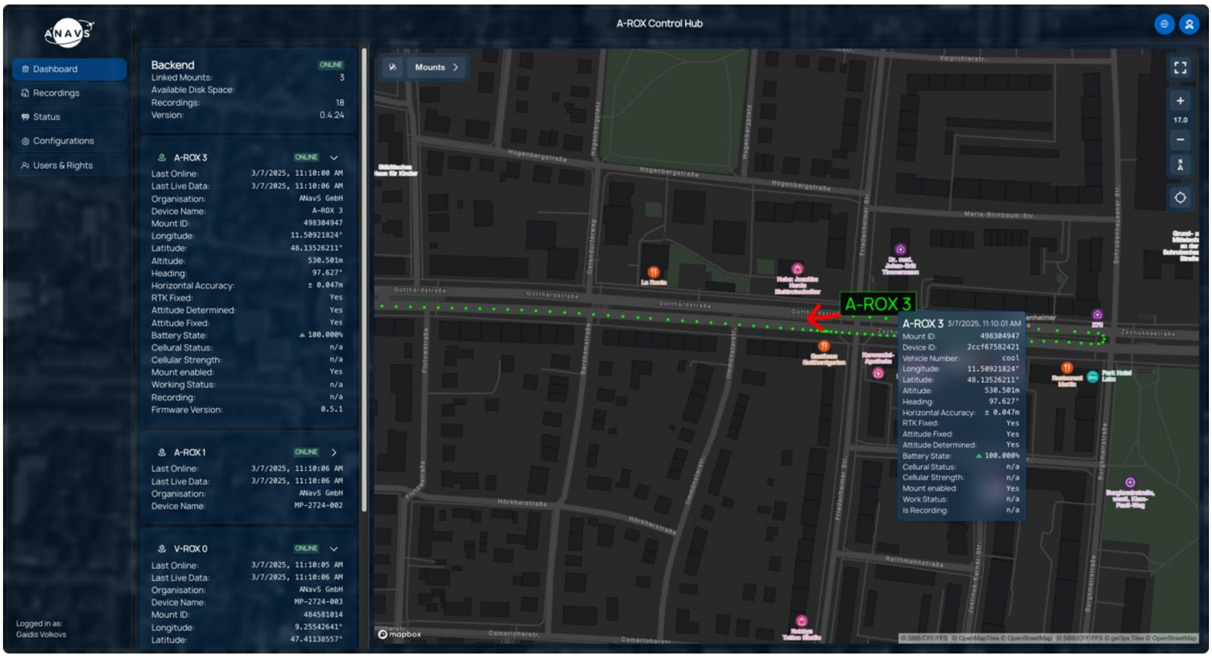


Figure 28: Screenshot of the A-ROX Control Hub landing page with live-view of all your A-ROX devices

4.1 Authentication

To access the Control Hub, users must first log in via a secure authentication process. The login page presents a standard email and password input field, ensuring that only authorized personnel can manage the system. Once the credentials are entered, users with two-factor authentication (2FA) enabled are prompted to scan a QR code using an authentication app, such as Google Authenticator, to generate a one-time security token. This added layer of security protects against unauthorized access. After scanning the QR code, the system requests the generated one-time token, which must be entered before full access is granted.

4.1.1 Login Page

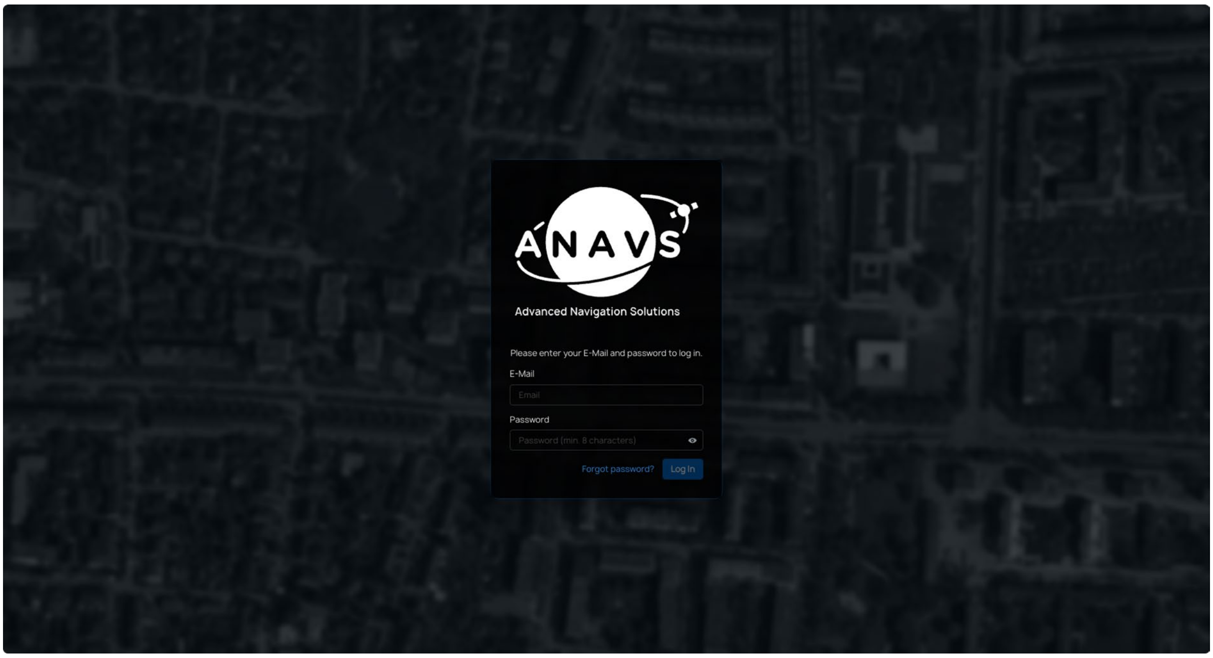


Figure 29: User login with email and password

4.1.2 Two Factor Authentication Setup

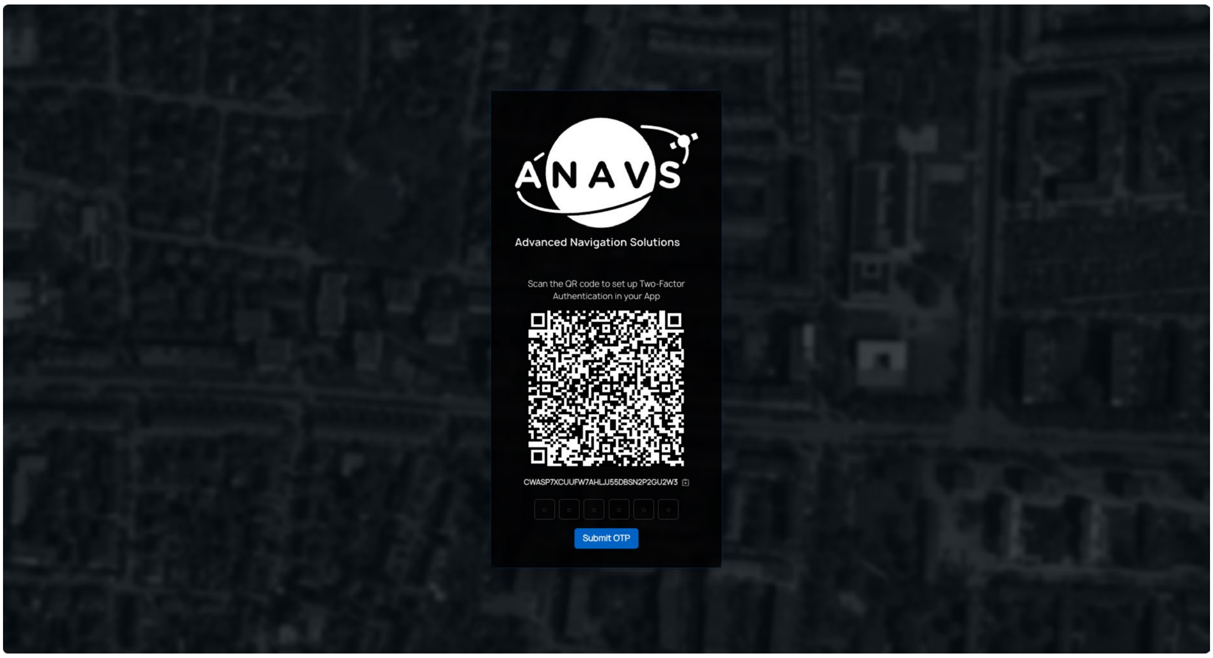
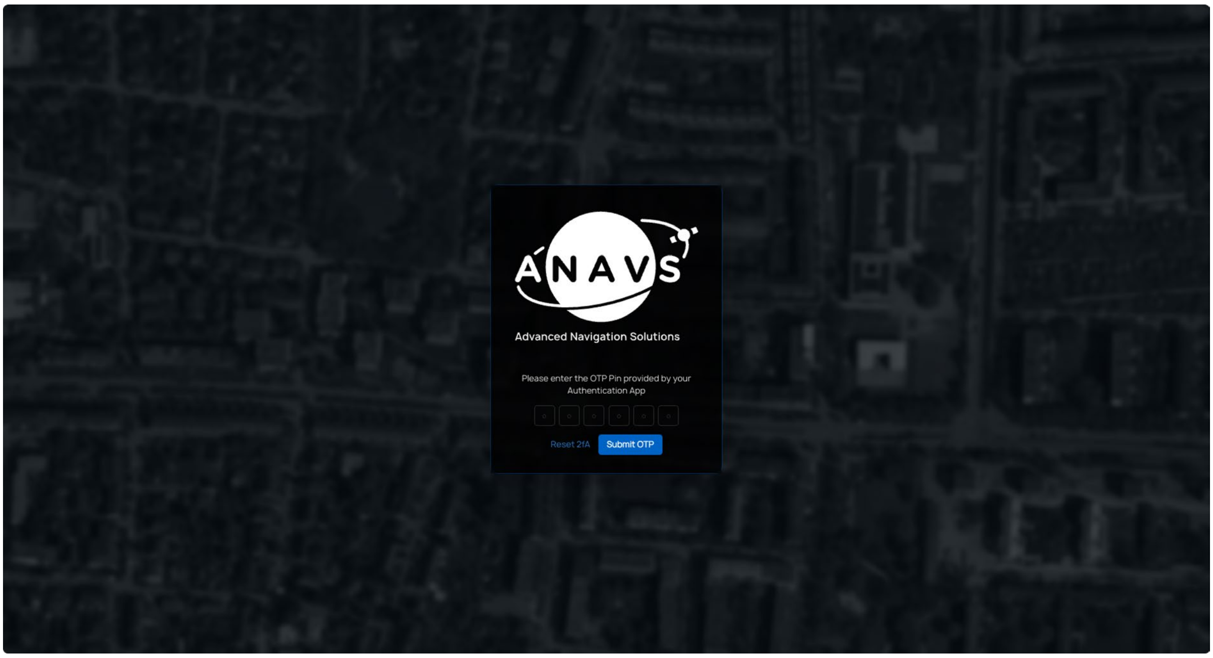


Figure 30: Two-Factor setup with QR-Code holding the Time-based one-time secret

4.1.3 Onetime Token Input



4.2 Recordings

Once logged in, users can navigate to the Recordings section, where all previously sessions (mcap with solution, raw GNSS, raw IMU, raw odometry data, status, diagnostics, etc.) are stored. The interface provides an organized list, showing timestamps, status, and available post-processing options. If a recording has already been processed, a highlighted preview appears, indicating that the data has been refined using the powerful forward-backward postprocessing engine. Users can choose to download, delete, or request additional processing of specific sessions. This feature is particularly useful for reviewing past navigation data and verifying accuracy.

Figure 31: Recordings page with highlighted post-processed solution preview

4.3 Status

The Status page offers a real-time overview of all connected A-ROX devices. The interface is structured like a dashboard, presenting essential system metrics, including GNSS signal strength, availability of correction data, and general device health. This page helps users quickly identify and troubleshoot potential issues, such as signal loss or misconfigured network settings. By providing a live snapshot of each device’s operational state, the Status page plays a key role in system monitoring.

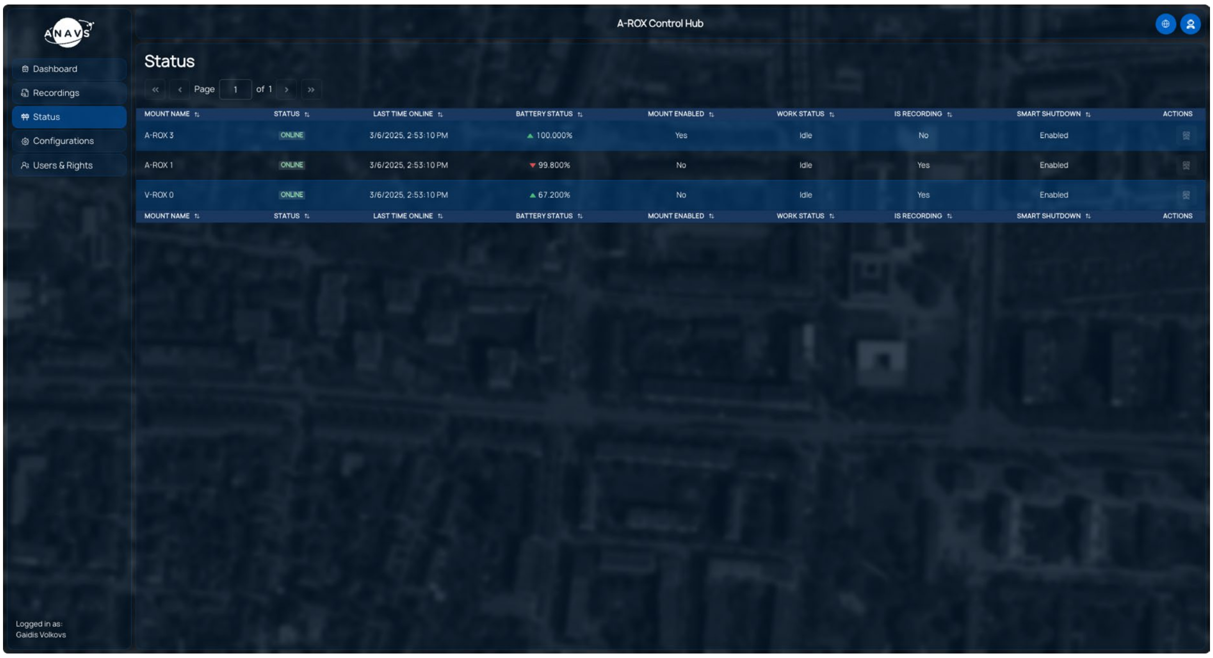


Figure 32: Status page with consolidated status information

4.4 Configuration

The Configuration section is where users can adjust various system settings to tailor A-ROX to specific applications. One of the key features is the Mounts configuration, which allows users to assign devices to specific vehicles or platforms, such as cars, ships, or robotic systems. Each mount can be configured individually, ensuring that its reference points and positioning data align with real-world applications.

Within the Work/Motion Profile settings, users can define automatic motion detection rules. This enables the system to start or stop data recording based on movement, preventing unnecessary data collection when the vehicle is stationary. Additionally, the Odometer settings page provides a way to configure wheel speed measurements and distance calculations, ensuring accurate positioning when GNSS signals are unavailable.

The Dimensions page within the Configuration section offers a visual representation of the system's reference points. Users can define custom Points of Interest (POIs) that allow the system to track specific positions relative to the A-ROX unit. The interface includes a 3D model, making it easier to verify correct positioning before deployment.

Another critical aspect of the Configuration section is the Devices page, which provides an overview of all registered A-ROX units. Here, users can check connection status, firmware versions, and update settings remotely. In addition, the Correction page enables the configuration of NTRIP correction sources, allowing the system to receive RTK and PPP data for centimeter-level positioning accuracy.

The Links section allows users to dynamically assign devices to mounts, ensuring that each A-ROX unit is properly linked to the corresponding vehicle or application environment.

4.4.1 Mounts

Figure 33: Managable Mounts eg.: cars, ships

4.4.1.1 Work/Motion Profile

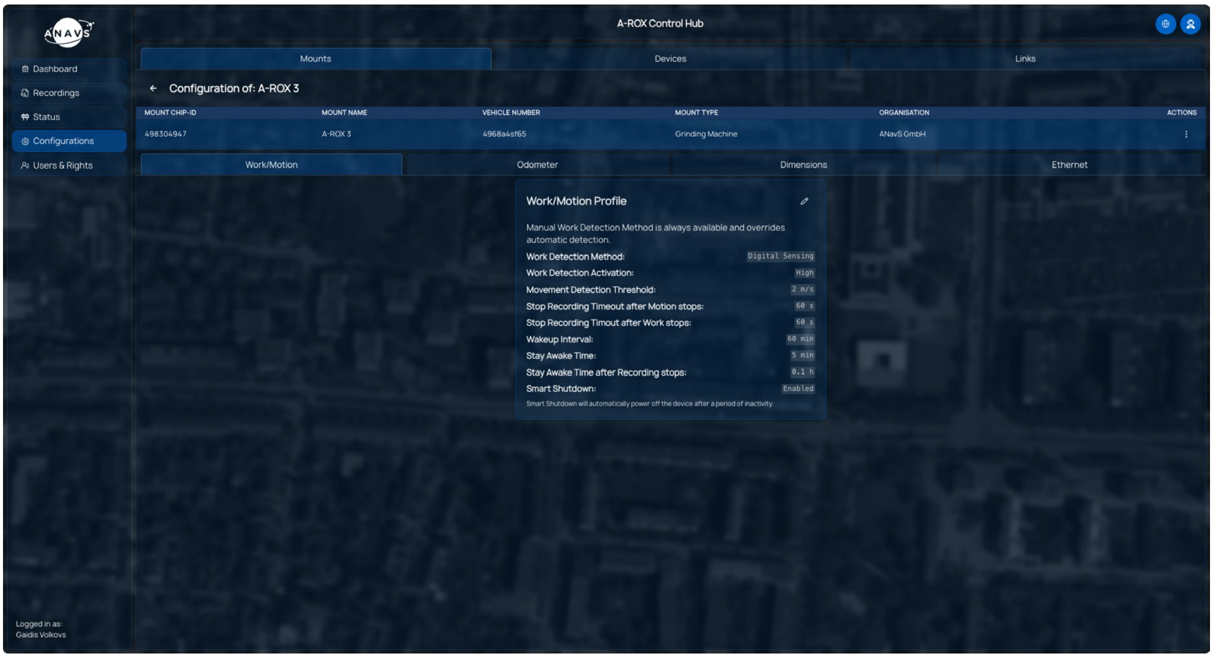


Figure 34: Work/Motion Profile management, sort out automatic detections and smart shutdown

4.4.1.2 Odometer

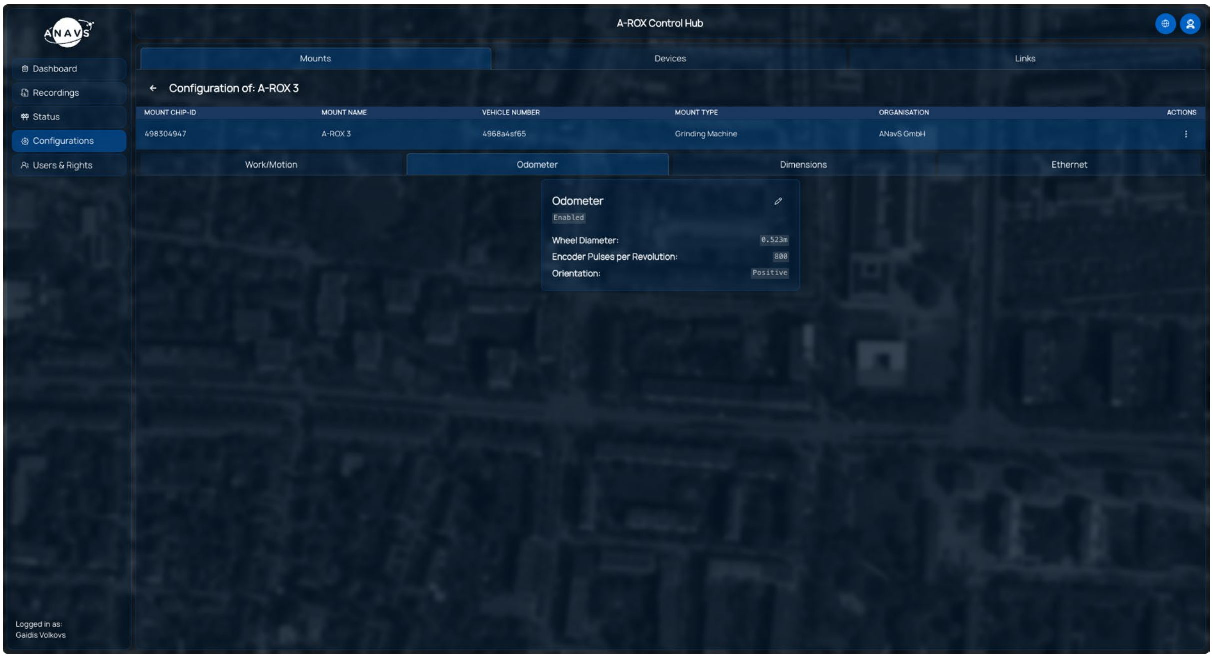


Figure 35 Configurable Odometer settings

4.4.1.3 Dimensions

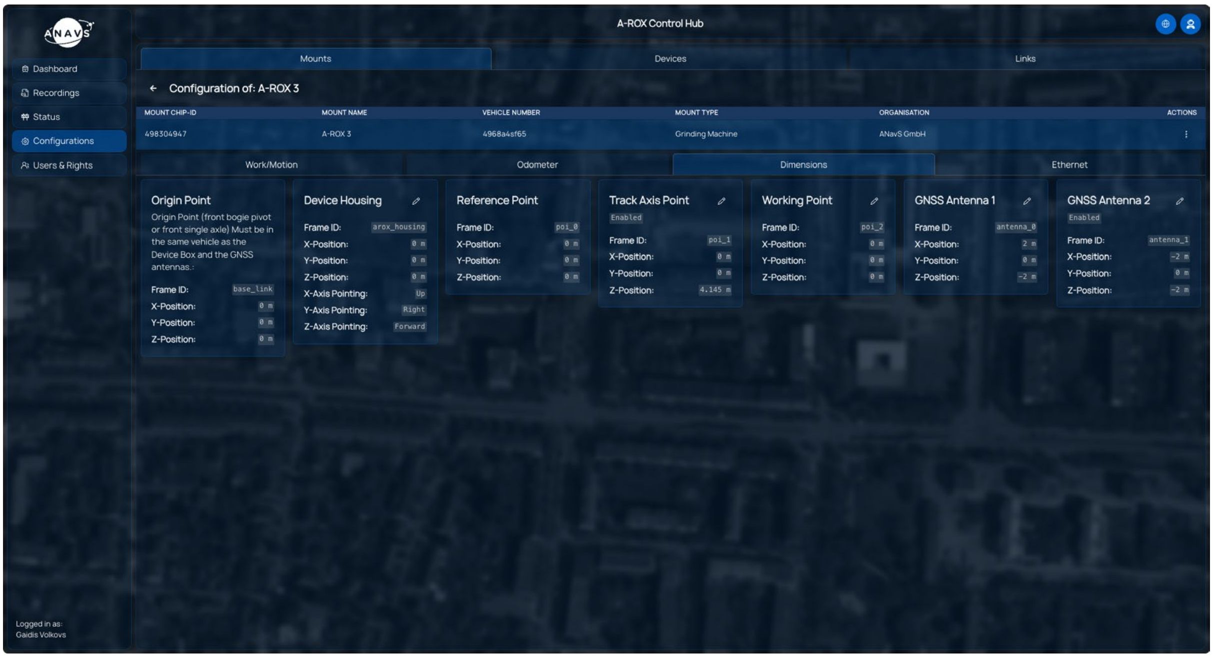


Figure 36: Dimensions with to exemplary additional point of interests (POI). Those can be configured individually

4.4.2 Devices

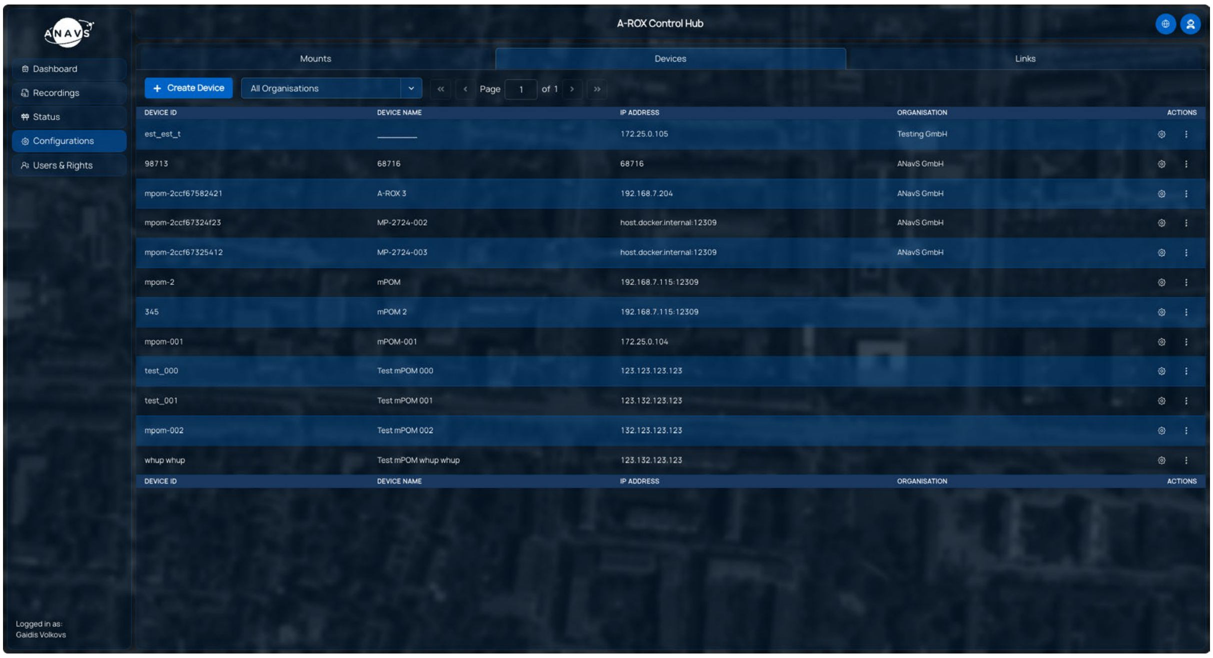


Figure 37: Single overview and initial setup place for A-ROX, V-ROX and G-ROX devices

4.4.2.1 Correction

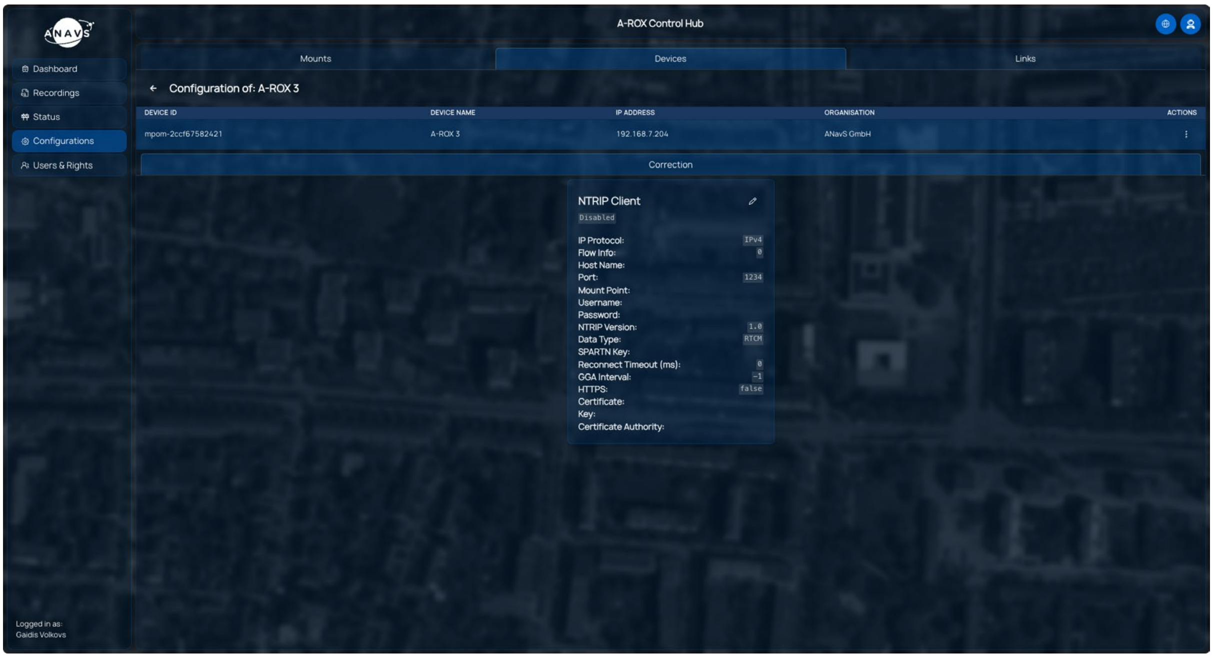


Figure 38: Configuration of the NTRIP Caster of your device

4.4.3 Links

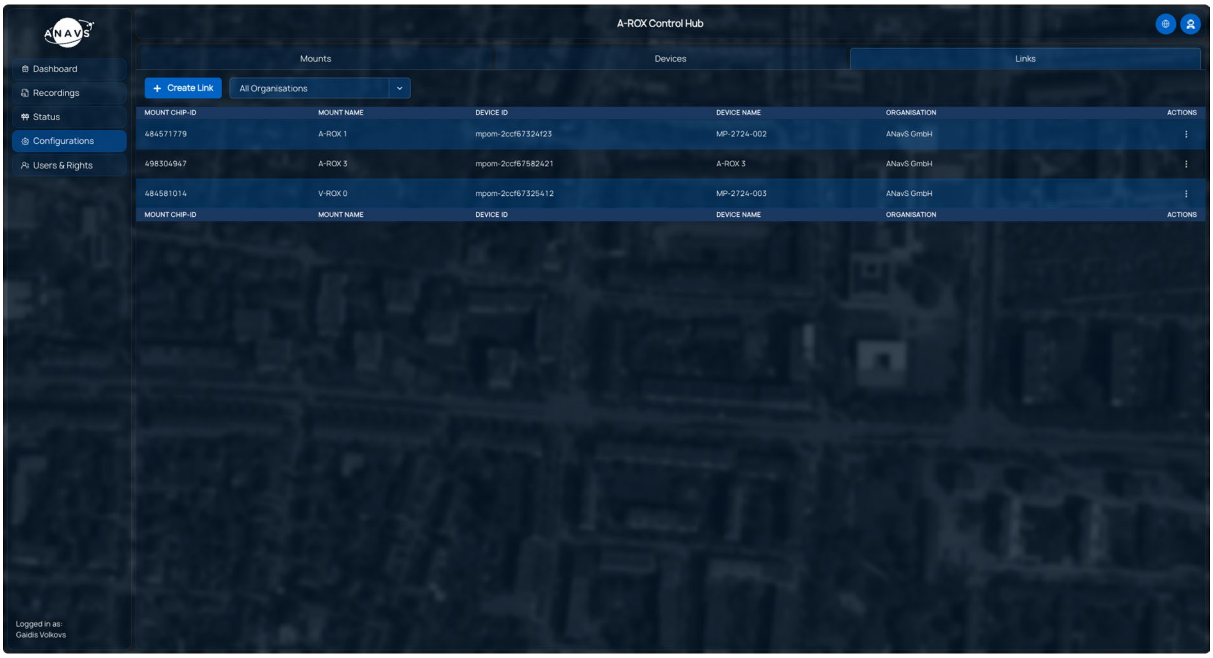


Figure 39: Dynamic linking between Mount and Device

4.5 Users & Rights

Managing access to the Control Hub is essential, especially for organizations with multiple users. The Users & Rights page provides a structured interface where administrators can assign roles and permissions. This ensures that only authorized personnel have the ability to modify critical settings or access sensitive data. Each user's access level can be adjusted according to their role, providing a secure and controlled operating environment.

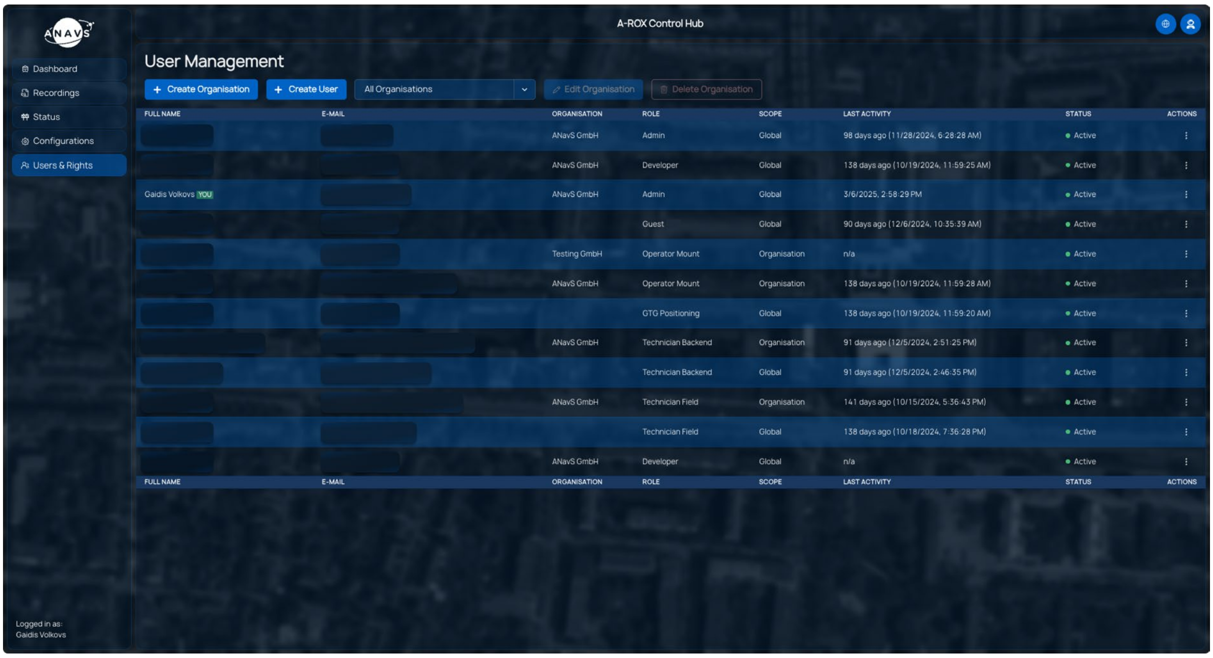


Figure 40: Single place to manage all Persons which have access to the Control Hub space

5 Definition of the Output Formats

The A-ROX GNSS-INS system provides multiple output formats to ensure flexibility and compatibility with various applications, from real-time vehicle navigation to high-precision data logging and post-processing. These formats include standard GNSS formats, proprietary sensor fusion outputs, and real-time data streams for integration with robotics, autonomous systems, and industrial applications

5.1 ROS 2 Message Format Description

Interface providing this output: Gigabit Ethernet

Recommended Version ROS2: ROS2 Humble

The boxes (automatically) record data, which is transferred to the ANavS Control Hub by the synchronizer and post-processed there (if this feature has been booked). The recordings are stored in the **MCAP format**, which contains messages, called **topics**, that are asynchronously logged in the file. The **MCAP file** is created on the box, storing all configured topics. After synchronization with the ANavS Control Hub, additional topics are added there.

5.1.1 Format

A detailed list of the individual topics and their sources can be found in the table below.

Topic	Source	Description
/anavs/algorithm/pad_status	Server	Provides various status messages from the **PAD (Position and Attitude Determination) software** (positioning algorithm).
/anavs/chip_id	A-ROX	**Customer specific topic:** Contains the ID of the connected serial number chip, which is read every few seconds and updated on change.
/anavs/device/apt_status	A-ROX	Includes an initial query of installed software packages (APT packages) along with version numbers. This information is published once at startup.
/anavs/device/battery_status	A-ROX	**Customer specific topic:** Contains battery information published by the Battery Management System (BMS) in mPOM Energy. This data is published cyclically and includes battery charge percentage, cell voltages, and temperature.
/anavs/device/cellular_status	A-ROX	This message is published cyclically and contains the signal strength and connection status of the integrated cellular modem.
/anavs/device/device_status	A-ROX	This message displays basic device status information, including the antenna status (OK, short circuit, open) and PAD software status. This data is also used for LED control and status display.

Topic	Source	Description
/anavs/device/hardware_status	A-ROX	This message is published cyclically and contains voltage and current measurements on the circuit board and processor parameters for error diagnosis and detection.
/anavs/device/moving_flag	A-ROX	Boolean value indicating whether motion is detected. Used for automatic data recording and energy management. This message is published cyclically.
/anavs/device/odometer_count	A-ROX	Incremental tick count of the connected odometer. This message is published cyclically.
/anavs/device/user_gpio	A-ROX	Displays the digital value at digital input 2 (User GPIO).
/anavs/device/vs_zug	A-ROX	**Customer specific topic:** Displays the digital value of the VS_ZUG signal, which indicates whether the GBM is active.
/anavs/gnss/_bdsnav	Server	Internal messages of satellite parameters for the PAD software. This message is published cyclically.
/anavs/gnss/_galnav	Server	Internal messages of satellite parameters for the PAD software. This message is published cyclically.
/anavs/gnss/_glonav	Server	Internal messages of satellite parameters for the PAD software. This message is published cyclically.
/anavs/gnss/_gpsnav	Server	Internal messages of satellite parameters for the PAD software. This message is published cyclically.
/anavs/gnss/_measepoch	Server	Internal messages of satellite parameters for the PAD software. This message is published cyclically.
/anavs/gnss/gnss_obs	A-ROX	Raw satellite measurement data from both antennas and correction data for the PAD software. This message is published cyclically.
/anavs/gnss/gpsfix	A-ROX	Raw satellite measurement data from satellite signals. This message is published cyclically.
/anavs/gnss/keplerian_ephemeris	A-ROX	Internal messages for Kepler parameters for the PAD software. This message is published cyclically
/anavs/gnss/pvtcartesian	A-ROX	Internal messages of satellite parameters containing raw measurements of satellite signals in Cartesian coordinates. This message is published cyclically.
/anavs/gnss/rtcm_raw	A-ROX	Internal message. Logs the raw stream of the correction data service.
/anavs/gnss/sbf_raw	A-ROX	Internal message. Logs the raw stream from the GNSS receiver.
/anavs/imu0/data_raw	A-ROX	Raw measurements from the integrated IMU, used by the PAD software in sensor fusion. This includes linear and angular accelerations across three axes. This message is published cyclically.

Topic	Source	Description
/anavs/mpom_box_settings	A-ROX	This message is published once and contains settings for work mode detection, GBM information, recording settings, and standby settings.
/anavs/ntrip_client_settings	A-ROX	This message is published once and contains settings for the correction data service, such as login credentials.
/anavs/ros_recorder_status	A-ROX	Indicates as a binary value whether a recording is currently taking place. This message is published upon status change.
/anavs/work_status	A-ROX	**Customer specific topic:** Displays the source used for automatic work mode detection (automatic_work_detection_method) and whether it is currently active (work_detected_automatic). If a work operation was manually started (button press), it is logged in work_detected_manual. This message is published upon status change.
/anavs/work_status_raw	A-ROX	**Customer specific topic:** Publishes current values for individual work mode detection sources. This message is published upon status change.
/diagnostics_agg	A-ROX	Reserved for various software development information.
/pad_filter_state	Server	Internal messages of satellite parameters for the PAD software.
/pad_settings	A-ROX	Contains all settings of the PAD (Position and Attitude Determination) software (positioning algorithm). These are typically written and applied once at startup.
/pad_solution	Server	Outputs the "Forward"-processed position and attitude solution from the PAD software. The software differentiates between various data sources (GNSS, IMU, odometry), and this message combines all sources. This output is used by the forward-backward postprocessing engine.
/pad_solution_gnss	Server	Outputs the "Forward"-processed position and attitude solution from the PAD software. Only solutions based on GNSS updates are included in this message. This topic is only published when satellite signals are available and typically has a lower update rate (5 Hz) compared to /pad_solution_imu.
/pad_solution_imu	Server	Outputs the "Forward"-processed position and attitude solution from the PAD software. Only IMU-based updates are included in this message.
/pad_status	Server	Provides various status messages related to the positioning process (PAD software).
/rosout	A-ROX	Logs from individual nodes are published here, categorized by severity level.

Topic	Source	Description
/smoother_solution	A-ROX	Outputs the "Forward-Backward"-processed position and attitude solution. This is the final processed solution, which can be used for further steps.
/tf_static	A-ROX	Transformations (TFs) define the spatial relationship (translation and rotation) of individual components on the train, such as the position and orientation of the GNSS antennas, mPOM box, odometer, reference point, and work equipment.

5.1.2 Point of Interests

The lever arms and rotations of individual components in the system are represented by various transformations in frames. The entire system includes approximately 20 such transforms, which are published in the /tf_static topic.

A special role is played by the so-called **Points of Interest (POI)** which can be defined arbitrarily by the customer. POIs allow to get the position and attitude of a projected point from the A-ROX, e.g. in the center or on other positions:

The **/pad_solution** topic is calculated for all frames (e.g. **"poi_0", "poi_1"**, …) and written sequentially into the same topic.

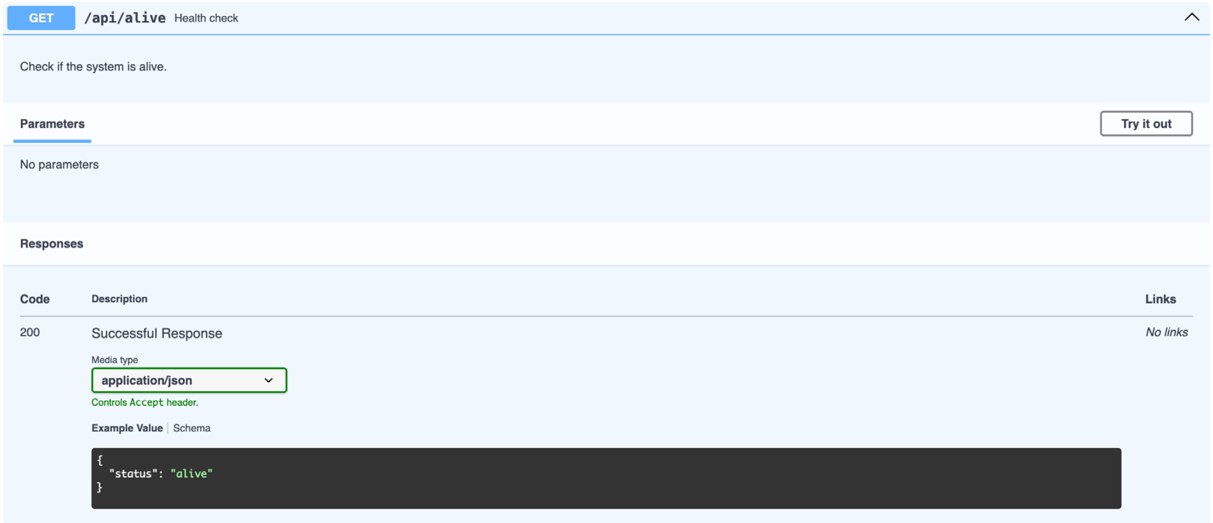
5.2 FAST-API Format Description

Interface providing this output: Gigabit Ethernet, Wi-Fi, 5G

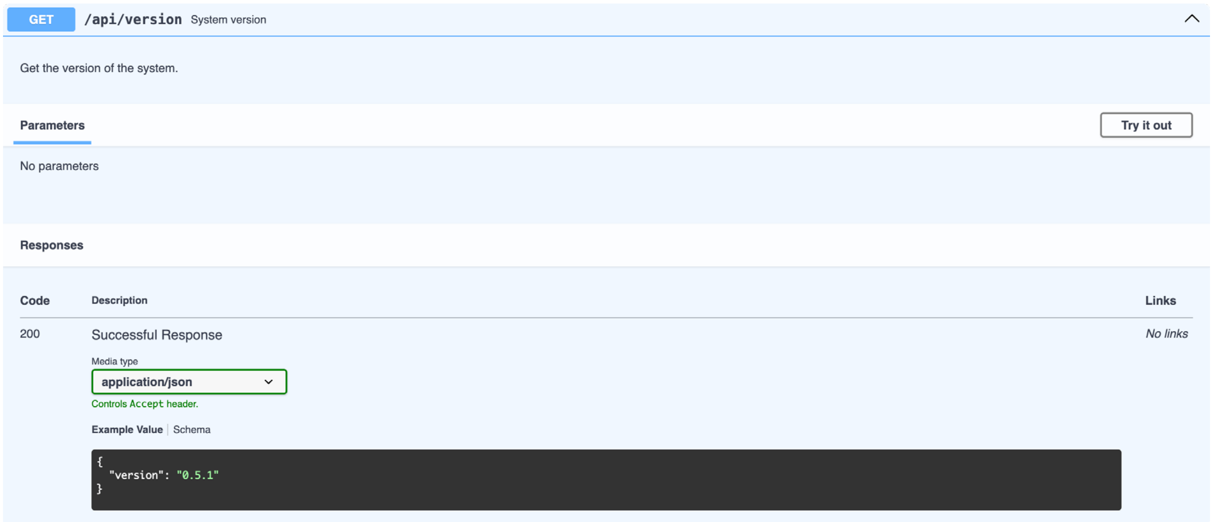
The API follows the RESTful architecture style with an interactive SwaggerUI and static ReDoc documentation created on basis of the OpenAPI specification.

5.2.1 Base

5.2.1.1 Alive



5.2.1.2 Version



5.2.2 Live Data

5.2.2.1 Full Stream

GET `/api/live_data/full_stream` Stream the full live data feed.

Full Live Data Stream Endpoint

This endpoint streams the entire live data feed from the ROS2 node. It delivers continuous updates to the client using Server-Sent Events (SSE).

Overview

- **Purpose:** Provides a real-time, full feed of live data for monitoring or further processing.
- **Mechanism:** Uses SSE to maintain a persistent connection for data streaming.
- **Note:** SSE streams are not testable within Swagger UI due to their continuous nature.

Examples

CURL Example

```
curl -N \\
  -H "Accept: application/json" \\
  "http://{AROX_IP_ADDRESS}/api/live_data/full_stream"
```

Responses

- **200 OK:** The live data stream is successfully delivered.
- **503 Service Unavailable:** No live data is available at the moment; please try again later.

5.2.2.2 PAD Solution

GET `/api/live_data/pad_solution` Stream the pad solution based on a specific point of interest.

Stream Pad Solution Endpoint

This endpoint streams a specific pad solution based on a given point of interest. The point of interest corresponds to the index of the pad solution within the available list.

Overview

- **Purpose:** Streams a single pad solution from a list of available solutions in real time.
- **Mechanism:** Utilizes Server-Sent Events (SSE) to continuously deliver updates to the client.
- **Protocol Note:** Due to the nature of SSE, this endpoint is not testable via Swagger UI.

Parameters

- **point_of_interest** (int): The index specifying which pad solution to stream.
  Example: If set to `0`, the first solution in the list is streamed. Ensure the index is within the valid range.

Examples

CURL Example

```
curl -N \\
  -H "Accept: application/json" \\
  "http://{AROX_IP_ADDRESS}/api/live_data/pad_solution?point_of_interest=0"
```

Parameters

- **point_of_interest** (int): The index specifying which pad solution to stream.
  Example: If set to `0`, the first solution in the list is streamed. Ensure the index is within the valid range.

Responses

- **200 OK:** Successfully streams the requested pad solution.
- **400 Bad Request:** The provided point of interest is out of bounds.
- **503 Service Unavailable:** No pad solution is currently available; please try again later.

5.2.3 Network

5.2.3.1 Network Interface Status

GET `/api/configuration/network-interface-status` Retrieve the current network interface status.

Retrieve Network Interface Status Endpoint

This endpoint retrieves the current status of all network interfaces available on the device. The response includes details such as interface name, active connection, available connections, interface status (up or down), and additional configuration information.

Responses

- **200 OK:** A JSON object conforming to `NetworkInterfaceListPydantic` containing the list of network interfaces and their status.
- **404 Not Found:** Returned when network status information is not available.
- **500 Internal Server Error:** Returned if an unexpected error occurs while retrieving the network interface status.

5.2.3.2 Network Connections

GET `/api/configuration/network-connections` Get Network Connections

Get Network Connections Endpoint

This endpoint sends a request to the underlying ROS2 service to retrieve the current network connection statuses. The response includes connection details for both Ethernet and WiFi, as applicable.

Query Parameters

- **connection_id** (str): An optional identifier for a specific network connection. If omitted or empty, information for all connections is retrieved.

Responses

- **200 OK:** Returns a JSON object conforming to `GetNetworkConnectionsResponsePydantic` that contains the network connection statuses.
- **404 Not Found:** Returned when the requested network connections are not available.
- **500 Internal Server Error:** Returned if an unexpected error occurs during the retrieval process.

5.2.3.3 Ethernet Interface

POST `/api/configuration/ethernet-settings` Configure the Ethernet interface.

Configure Ethernet Interface Endpoint

This endpoint configures the Ethernet interface with the specified settings. The request must include details such as the IP address, subnet mask, gateway, and the DHCP setting.

Request Body

- **connection_id** (str): Identifier for the Ethernet connection (e.g., `main-ethernet`).
- **dhcp** (bool): Flag indicating whether DHCP should be enabled.
- **ip_address** (str): The IP address to assign to the interface.
- **subnet_mask** (str): The subnet mask for the network.
- **gateway** (str): The gateway address.

Responses

- **200 OK:** Returns a `BaseResponse` object indicating the success or failure of the configuration operation.
- **500 Internal Server Error:** Returned if an error occurs during the configuration process.

5.2.3.4 WiFi Interface

POST `/api/configuration/wifi-settings` Configure the WiFi interface.

Configure WiFi Interface Endpoint

This endpoint configures the WiFi interface with the provided settings. The request must include details such as the connection identifier, SSID, password, and IP configuration.

Request Body

- **connection_id** (str): Identifier for the WiFi connection (e.g., `wifi-ap`).
- **hotspot** (bool): Flag indicating whether the connection should be configured as a hotspot.
- **ssid** (str): The SSID of the WiFi network.
- **password** (str): The password for the WiFi network.
- **ip_address** (str): The IP address to assign (if a static IP is used).
- **subnet_mask** (str): The subnet mask for the network.
- **gateway** (str): The gateway address.

Responses

- **200 OK:** Returns a `BaseResponse` object indicating the result of the WiFi configuration operation.
- **500 Internal Server Error:** Returned if an error occurs during the configuration process.

5.2.4 CAN

5.2.4.1 Retrieve CAN BUS Settings

GET `/api/configuration/can-bus-settings` Retrieve CAN BUS settings

Retrieve CAN BUS Settings Endpoint

This endpoint retrieves all the CAN BUS configurations from the system. The response includes the complete settings for all CAN buses, following the `AllCanBusSettings` schema. The configuration data includes messages and signals related to the CAN bus.

Responses

- **200 OK:** The CAN bus settings are successfully retrieved.
- **500 Internal Server Error:** An unexpected error occurred while retrieving the settings.
- **503 Service Unavailable:** The settings file is not found or the service is unavailable.

5.2.4.2 Update CAN BUS Settings

PUT `/api/configuration/can-bus-settings` Update CAN BUS Settings

Update CAN BUS Settings Endpoint

This endpoint updates the CAN BUS configurations using the provided settings in the request body. The request must adhere to the `AllCanBusSettings` schema. The response is formatted according to the `CanManagerApplyResponse` schema.

Request Body

- **AllCanBusSettings Schema:** The complete set of CAN bus configuration settings to be applied.

Responses

- **200 OK:** The CAN bus settings are successfully updated.
- **400 Bad Request:** The provided data is in an invalid format or contains invalid content.
- **500 Internal Server Error:** An unexpected error occurred during the update process.
- **503 Service Unavailable:** The settings file is not found or the service is unavailable.

5.2.4.3 Retrieve CAN Data Output Settings

GET `/api/configuration/can-bus-data-output-settings` Retrieve CAN data output settings

Retrieve CAN Data Output Settings Endpoint

This endpoint retrieves all the CAN data output settings from the system. The response includes detailed configurations related to CAN data output, including messages and signals, conforming to the `AllCanDataOutputSettings` schema.

Responses

- **200 OK:** The CAN data output settings are successfully retrieved.
- **500 Internal Server Error:** An unexpected error occurred while retrieving the settings.
- **503 Service Unavailable:** The settings file is not found or the service is unavailable.

5.2.4.4 Update CAN Data Output Settings

PUT `/api/configuration/can-bus-data-output-settings` Update CAN data output settings

Update CAN Data Output Settings Endpoint

This endpoint updates the CAN data output settings with the provided configuration. The request body must follow the `AllCanDataOutputSettings` schema. The response is structured according to the `CanManagerApplyResponse` schema.

Request Body

- **AllCanDataOutputSettings Schema:** The complete CAN data output settings to be applied.

Responses

- **200 OK:** The CAN data output settings are successfully updated.
- **400 Bad Request:** The provided data is in an invalid format or contains invalid content.
- **500 Internal Server Error:** An unexpected error occurred during the update process.
- **503 Service Unavailable:** The settings file is not found or the service is unavailable.

5.2.4.5 Retrieve Message Sources

GET `/api/configuration/can-data-output-settings/message-sources` Retrieve all the message sources

Retrieve Message Sources Endpoint

This endpoint retrieves all the available message sources for CAN data output settings. The data is loaded from a JSON file containing the configuration of data streams.

Responses

- **200 OK:** Returns a list of data streams representing the message sources.
- **503 Service Unavailable:** The settings file is not found.
- **500 Internal Server Error:** An unexpected error occurred while retrieving the message sources.

5.2.4.6 Retrieve CAN Topics

GET `/api/configuration/can-data-output-settings/output_list` Retrieve all the available CAN topics

Retrieve CAN Topics Endpoint

This endpoint retrieves all the available CAN topics for CAN data output settings. The response is structured according to the `CanTopicList` schema and includes a list of topic descriptions.

Responses

- **200 OK:** The CAN topics are successfully retrieved.
- **503 Service Unavailable:** The settings file is not found.
- **500 Internal Server Error:** An unexpected error occurred while retrieving the CAN topics.

5.2.4.7 Export CAN BUS Configuration to DBC

POST `/api/configuration/can-export-dbc` Export CAN BUS configuration to DBC

Export CAN BUS Configuration to DBC Endpoint

This endpoint exports the CAN BUS configuration to a DBC file. The request must specify the CAN bus for which the configuration should be exported. The exported DBC file contains the configuration details in a text format suitable for further analysis or import into other tools.

Request Body

- **can_bus** (str): The identifier of the CAN bus to be exported.

Responses

- **200 OK:** Returns an `ExportDbcResponse` object containing the DBC file content and a success status.
- **500 Internal Server Error:** Returned if an error occurs during the export process.

5.2.5 Recording

5.2.5.1 Start Recording

POST `/api/configuration/start-recording` Start recording

Start Recording Endpoint

This endpoint starts recording data with the specified file name. If no file name is provided, a default name based on the current date and time will be used.

Parameters

- **file_name** (str): The name of the file to save the recording. If empty, a default name with the format `anavs_mcap_YYYY-MM-DD_HH-MM-SS` will be generated.

Returns

- **BaseResponse**: A model containing the success status, message, and the file name of the recording.

Errors

- **HTTP 500**: Internal server error.

5.2.5.2 Stop Recording

POST `/api/configuration/stop-recording` Stop recording

Stop Recording Endpoint

This endpoint stops the current recording.

Returns

- **BaseResponse**: A model containing the success status and message.

Errors

- **HTTP 500**: Internal server error.

5.2.5.3 Get Recordings List

GET `/api/recordings/` Get a list of recordings

Get Recordings List Endpoint

This endpoint retrieves a list of recordings available on the server. Each recording is represented by a `Rosbag2BagfileInformation` object that includes metadata extracted from a YAML file.

Responses

- **200 OK:** Returns a list of recordings.
- **404 Not Found:** The recordings directory was not found.

5.2.5.4 Download Recording

GET `/api/recordings/{recording_name}/download` Download a recording by its name

Download Recording Endpoint

This endpoint allows you to download a recording as a ZIP archive by specifying its name. The recording directory is compressed into a ZIP file and returned as a streaming response.

Parameters

- **recording_name** (str): The name of the recording to download.

Responses

- **200 OK:** Returns the recording as a ZIP file.
- **404 Not Found:** The specified recording directory does not exist.

5.2.5.5 Delete Recording

DELETE `/api/recordings/{recording_name}` Delete a recording by its name

Delete Recording Endpoint

This endpoint deletes a recording identified by its name. On successful deletion, a response indicating success is returned.

Parameters

- **recording_name** (str): The name of the recording to delete.

Responses

- **200 OK:** The recording was successfully deleted.
- **404 Not Found:** The recording was not found.
- **500 Internal Server Error:** An error occurred during the deletion process.

5.2.6 Algorithm

5.2.6.1 Retrieve PAD Settings

GET `/api/configuration/pad_settings` Retrieve PAD Settings

Retrieve PAD Settings Endpoint

This endpoint retrieves the current PAD settings from the ROS2 node. The response contains the algorithm configuration used by the PAD system, formatted according to the `PADSettingsPydantic` schema.

Responses

- **200 OK:** Returns the PAD settings.
- **500 Internal Server Error:** An unexpected error occurred while retrieving the settings.

5.2.6.2 Check PAD Software

GET `/api/configuration/check_pad` Check PAD Software

Check PAD Software Endpoint

This endpoint checks whether the PAD software is currently running on the system. It returns a status message indicating the operational state of the PAD software.

Responses

- **200 OK:** Returns a `BaseResponse` object with the status.
- **500 Internal Server Error:** An unexpected error occurred while checking the software.

5.2.6.3 Start PAD Software

POST `/api/configuration/start_pad` Start PAD Software

Start PAD Software Endpoint

This endpoint starts the PAD software on the device. It initiates the PAD system and returns a response indicating whether the operation was successful.

Responses

- **200 OK:** The PAD software has been successfully started.
- **500 Internal Server Error:** An error occurred while starting the PAD software.

5.2.6.4 Stop PAD Software

POST `/api/configuration/stop_pad` Stop PAD Software

Stop PAD Software Endpoint

This endpoint stops the currently running PAD software on the device. It returns a response indicating the success or failure of the stop operation.

Responses

- **200 OK:** The PAD software has been successfully stopped.
- **500 Internal Server Error:** An error occurred while stopping the PAD software.

5.2.6.5 Restart PAD Software

POST `/api/configuration/restart_pad` Restart PAD Software

Restart PAD Software Endpoint

This endpoint restarts the PAD software on the device. It stops the PAD software (if running) and then starts it again, returning a response with the result of the operation.

Responses

- **200 OK:** The PAD software has been successfully restarted.
- **500 Internal Server Error:** An error occurred during the restart process.

5.2.7 Dimensions

5.2.7.1 Retrieve Dimensions Configuration

GET `/api/configuration/dimensions-settings` Retrieve Dimensions Configuration

Retrieve Dimensions Configuration Endpoint

This endpoint retrieves the current dimensions configuration by loading the static tf2 tree from the ROS2 node. The response is structured according to the `GetTransformsResponse` schema and contains the list of static transforms.

Responses

- **200 OK:** Returns the dimensions settings as a JSON object following the `GetTransformsResponse` schema.
- **500 Internal Server Error:** An unexpected error occurred while retrieving the dimensions settings.
- **503 Service Unavailable:** The settings file was not found.

5.2.7.2 Update Dimensions Configuration

PUT `/api/configuration/dimensions-settings` Update Dimensions Configuration

Update Dimensions Configuration Endpoint

This endpoint updates the dimensions configuration by accepting a new set of transforms in the request body. The request body must follow the `SetTransformsRequest` schema, and the response is formatted according to the `SetTransformsResponse` schema.

Request Body

- **transforms**: A list of transforms to set as the new static configuration.

Responses

- **200 OK:** The dimensions settings have been successfully updated.
- **500 Internal Server Error:** An unexpected error occurred while updating the dimensions settings.
- **503 Service Unavailable:** The settings file was not found.

5.2.8 Input

5.2.8.1 External Time Reference

PUT `/api/input/time-reference` Provide an external time reference

Provide External Time Reference Endpoint

This endpoint accepts a measurement from an external time source that is not actively synchronized with the system clock. When a time reference is provided, the system timestamps it with the current system time and records the measurement.

Rate Limiting:
The endpoint is rate-limited to one request every 5 seconds to prevent excessive updates.

Request Body

- **time_ref** (Time): The current time from the external time source.
- **source** (str): A string identifier for the external time source.

Responses

- **200 OK:** Time reference successfully provided.
- **400 Bad Request:** Invalid data format or content.
- **500 Internal Server Error:** An error occurred while processing the time reference.

5.2.9 Utilities

5.2.9.1 Hardware Status

GET `/api/utilities/hardware_status` Get The Hardware Status.

Get Hardware Status Endpoint

This endpoint retrieves the latest hardware status from the ROS2 node. The hardware status includes key metrics such as CPU temperature, ADC temperature, various voltage levels, current measurements, and usage statistics (e.g., CPU, memory, HDD).

Responses

- **200 OK:** Returns a `HardwareStatusPydantic` model with detailed hardware status information.
- **503 Service Unavailable:** Returned when no hardware status is available from the ROS2 node.

5.2.9.2 Cellular Status

GET `/api/utilities/cellular_status` Get Cellular Status

Get Cellular Status Endpoint

This endpoint retrieves the current status of the cellular modem from the ROS2 node. The response includes details about the cellular module, such as the manufacturer, model, operational state, and signal quality.

Responses

- **200 OK:** Returns a JSON object conforming to the `CellularStatusPydantic` model containing the cellular status details.

5.2.9.3 Recorder Status

GET `/api/utilities/recorder_status` Get the current status of the recorder.

Get Recorder Status Endpoint

This endpoint returns the current status of the recorder. It provides information about the recorder's operational state and any associated error messages.

Recorder States

- **0**: The recorder is ready and currently not recording.
- **1**: The recorder is starting to record.
- **2**: The recorder is currently recording.
- **3**: The recorder is stopping the recording.

Responses

- **200 OK:** Returns a `RecordingStatusPydantic` model with the current recorder status.

5.2.10 System Info

5.2.10.1 Stream PAD Logs

GET `/api/system_info/pad_logs` Stream the current PAD logs

Stream PAD Logs Endpoint

This endpoint streams the latest PAD logs from the ROS2 node.

Note: This endpoint is not testable in the Swagger UI due to the nature of the EventSource protocol.

CURL Example

```
curl -N \\
  -H "Accept: application/json" \\
  "http://{AROX_IP_ADDRESS}/api/system_info/pad_logs"
```

Returns

- **EventSourceResponse**: A streaming response containing PAD logs.

5.2.10.2 Stream ROSOUT Logs

GET `/api/system_info/rosout` Stream the current ROSOUT logs

Stream ROSOUT Logs Endpoint

This endpoint streams the latest ROSOUT logs from the ROS2 node.

Note: This endpoint is not testable in the Swagger UI due to the nature of the EventSource protocol.

CURL Example

```
curl -N \\
  -H "Accept: application/json" \\
  "http://{AROX_IP_ADDRESS}/api/system_info/rosout"
```

Returns

- **EventSourceResponse**: A streaming response containing ROSOUT logs.

5.2.10.3 Stream Diagnostics

GET `/api/system_info/diagnostics` Stream the current diagnostics

Stream Diagnostics Endpoint

This endpoint streams the latest diagnostic data from the ROS2 node.

Note: This endpoint is not testable in the Swagger UI due to the nature of the EventSource protocol.

CURL Example

```
curl -N \\
  -H "Accept: application/json" \\
  "http://{AROX_IP_ADDRESS}/api/system_info/diagnostics"
```

Returns

- **EventSourceResponse**: A streaming response containing diagnostics data.

5.3 CAN Output Format Description

Interface providing this output: CAN/CAN-FD channel(s)

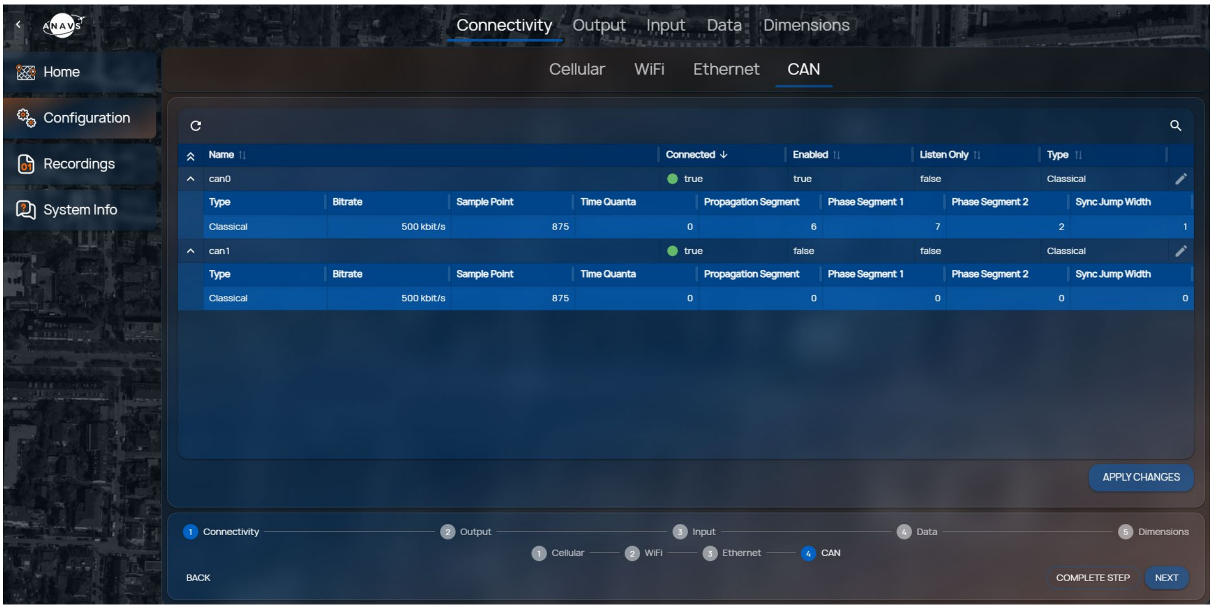
CAN is a robust and widely spread data bus standard which is among other applications also used in almost all modern vehicles for communication between the various controllers in it. The CAN output provided by the A-ROX is highly configurable and can be adjusted by the customer through the web application to meet their specific needs. In general, the customer can define multiple CAN/CAN-FD messages. For each CAN message, a data source needs to be selected. This data source then provides multiple values, which are always transmitted together at a specific rate. Using these values, the customer can define the signals of the CAN message according to their requirements. These signals can be modified using scaling, offsets, and a specified number of bits, and they can be represented as either floats or integers.
Available sources:

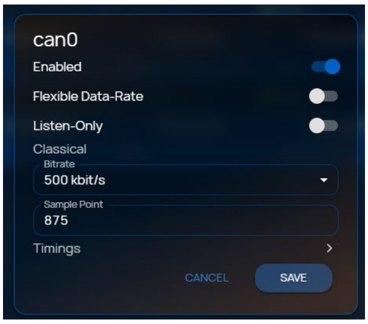
Customers can choose from the following data sources, each of which includes a timestamp:

- IMU raw data
- GNSS raw solution
- General device states (e.g., antenna state, correction availability)
- PAD solution (tightly coupled sensor fusion solution)

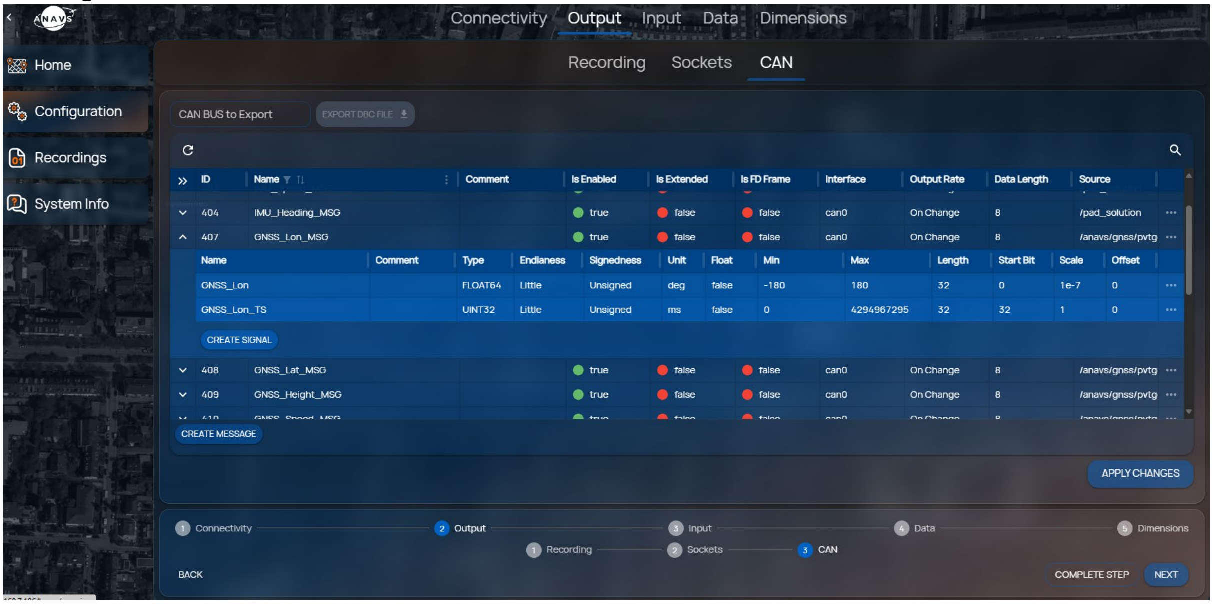
5.3.1 Configuring CAN

CAN Bus settings can be configured in the A-ROX Control Center in the corresponding tabs. Under Connectivity > CAN you can set all physical bus settings all physical settings for each bus.





In Configuration > Output > CAN you can configure your own customized can messages and add signals to them



5.4 ACOM Binary Format Description

Interface providing this output: Gigabit Ethernet (socket type: UDP)

The ACOM binary format is a data format designed for efficient communication of navigation measurements and other data. It is a very compact format and only includes core measurements, which makes it particularly suitable for inertial navigation systems.

The ACOM format was developed specifically for an easy change from OxTS systems to ANavS systems. The ACOM is not covered in this manual as it has its own dedicated manual.
Please ask our support team for more information.

5.5 NMEA 0183 Format Description

Interface providing this output: Gigabit Ethernet, Wi-Fi

This protocol is based on the international standard for maritime navigation and radio communication, equipment and systems and digital interfaces (IEC 61161-1)[2]. This standard adopted the de-facto standards for interfacing with marine electronic devices, known as NMEA 0183. The data is transmitted in sentences of variable length with a specified sentence structure.

5.5.1 Sentence Structure

- Address field
- Data fields
- Checksum field
- Terminating field
- All sentences contain only ASCII characters
- The maximum length of a sentence is 82 characters
- All fields are separated by delimiters

5.5.2 Address field

The address field starts with "$" followed by the talker ID and a sentence identifier. The used talker IDs are:

- **GP** for GPS only solutions
- **GL** for GLONASS only solutions
- **GA** for GALILEO only solutions
- **GB** for BEIDOU only solutions
- **GQ** for QZSS only solutions
- **GN** for multi GNSS solutions
- **--** for no satellites solution

The used sentence identifiers are:

- **GGA** – Global Positioning System Fix Data
- **GNS** – GNSS Fix Data
- **GSA** – GNSS DOP and Active Satellites

[2] IEC 61162-1 ed.2: http://read.pudn.com/downloads151/ebook/657722/IEC%2061162-1%20ed.2%20(2000).pdf

- **GST** – GNSS pseudorange error statistics
- **GSV** – GNSS Satellites in View
- **PASHR** – Attitude Data
- **RMC** – Recommended Minimum Specific GNSS Data
- **ROT** – Rate of Turn Data
- **THS** – True Heading State Data
- **VTG** – Course over Ground and Ground Speed
- **ZDA** – Time and Date

5.5.3 Data fields

Data fields must always be separated by ",". They can contain alpha, numeric, and alphanumeric values all coded in ASCII characters. The length of a data field can be constant, variable, or can contain a fixed and variable portion. This differs for each sentence.

5.5.4 Checksum field

The Checksum field starts with "*" followed by the checksum of the sentence. The Checksum is generated with a bitwise exclusive OR of all fields including the "," delimiters, between but not including the "$" and the "*" characters. The hexadecimal value of the checksum is then converted to two ASCII characters.

5.5.5 Terminating field

The terminating sequence contains the two ASCII characters <CR> and <LF> without any delimiter.

5.5.6 Satellite Numbering

- GPS: 1-37
- GLONASS: 38-61 & 63-68
- GALILEO: 71-106
- BEIDEOU: 141-177
- QZSS: 181-187

5.5.7 Sentence specification

The following sections describe the NMEA sentences supported by ANavS. These formats are defined by the NMEA 0183 Standard Version 4.10 (06/2012). Each section contains a table with the fields of the sentence, their format, an example, and a description of the field. The table is followed by an example sentence. All descriptions and examples provided are for reference only and are subjective to change. ANavS does not populate all fields as described by the NMEA standard, either to hardware limitations or other reasons. These fields are marked as "**Not Supported"** in the description table.

5.5.8 GGA – Global positioning system (GPS) fix data

TALKER ID	XX	All talker IDs usable
SENTECE ID	GGA	
UTC of position	hhmmss.ss	Fixed length 2 digits after dot "h" hours, "m" minutes, "s" seconds
Latitude	ddmm.mmmmmm	Fixed length 4 digits before and 6 after dot "d" degrees, "m" minutes
Hemisphere of latitude	N/S	N if value of latitude is positive
Longitude	dddmm.mmmmmm	Fixed length 5 digits before and 6 after dot
Hemisphere of longitude	E/W	E if value of longitude is positive
GPS quality indicator	X	0: GNSS fix not available 1: GNSS fix valid 4: RTK fixed ambiguities 5: RTK float ambiguities
Number of satellites used for positioning	XX	Fixed length 01 for single digits
HDOP	XX.X	Variable/fixed length 1 digit after dot, variable before
Altitude geoid height	(-)X.XX	Variable/fixed length 2 digits after dot, variable before
Unit of altitude	M	
Geoidal separation	(-)X.XX	Variable/fixed length 2 digits after dot, variable before
Unit of geoidal separation	M	
Age of differential data	XX.X	Age of RTK corrections. Empty if RTK corrections never received. Variable/fixed length: 1 digit after dot, variable before
Differential reference station ID		**-Not Supported-**

Example:
$GNGGA,185833.80,4808.7402397,N,01133.9325039,E,5,15,1.1,470.50,M,45.65,M,,*75

5.5.9 GNS – GNSS Fix Data

TALKER ID	XX	All talker IDs usable
SENTENCE ID	GNS	
UTC of position	hhmmss.ss	Fixed length 2 digits after dot "h" hours, "m" minutes, "s" seconds
Latitude	N/S	Fixed length 4 digits before and 6 after dot "d" degrees, "m" minutes
Hemisphere of Latitude	X.XX	N if value of latitude is positive
Longitude	M	Fixed length 5 digits before and 6 after dot "d" degrees, "m" minutes
Hemisphere of Longitude	X.XX	E if value of longitude is positive
Positioning Mode Indicators	XXXXX	Mode Indicators for each constellation with index 0: GPS 1: GLONASS 2: GALILEO 3: BEIDOU 4: QZSS Values can be: A: Autonomous N: Invalid
Satellites	XX	Number of satellites used for positioning Fixed length: 01 for single digit
HDOP (Horizontal Dilution of precision)	X.X	Fixed length 1 digit before and 1 after dot
Altitude Geoid Height	(-)X.X	Fixed length 1 digit before and 1 after dot
Geoidal Separation	(-)X.X	Fixed length 1 digit before and 1 after dot
Age of Differential Data	XX.X	Age of RTK corrections. Empty if RTK corrections never received. Variable/fixed length: 1 digit after dot, variable before
Reference Station Id		-Not Supported-
Navigation Status	V	-Not Supported-

Example:
$GNGNS,112900.00,4808.157676,N,01130.249576,E,ANANN,08,5.7,517.19,45.65,,,V*6C

5.5.10 GSA – GNSS DOP and active satellites

TALKER ID	XX	All talker IDs usable*
SENTENCE ID	GSA	
Fix Mode	X	A: Automatic M: forced to operate in 3D

Fix Type	X	1: GNSS fix not available 3: 3D
Satellite ID number	XX XX XX XX XX XX XX XX XX XX XX XX	Fixed length: 01 for single digit Up to 12 satellites per constellation Empty field if not used for positioning GPS: 1-37 GLONASS: 38-61 & 63-68 GALILEO: 71-106 BEIDEOU: 141-177 QZSS: 181-187
PDOP HDOP VDOP	XX.XX	Fixed length 2 digits before and after dot
System ID	X	0: QZSS 1: GPS 2: GLONASS 3: GALILEO 4: BEIDOU

*If GN is used for Talker ID a separate sentence must be created for each GNSS constellation, all starting with the Talker ID for multi GNSS. The corresponding System ID field indicates which constellation the sentence belongs to.

Example:
$GNGSA,M,3,27,32,,,,,,,,,,,11.86,5.75,10.38,1*0A

$GNGSA,M,3,77,96,99,,,,,,,,,,11.86,5.75,10.38,3*03

5.5.11 GST – GNSS Pseudorange Error Statistics

TALKER ID	XX	All talker IDs usable
SENTENCE ID	GST	
UTC of position	hhmmss.ss	Fixed length 2 digits after dot “h” hours, “m” minutes, “s” seconds
RMS Pseudorange Residuals	X.XXX	Fixed length 3 digits after dot
Error Ellipse Semi-Major Axis	X.XXX	Fixed length 3 digits after dot
Error Ellipse Semi-Minor Axis	X.XXX	Fixed length 3 digits after dot

Error Angle Semi-Major Axis	X.XXX	Fixed length 3 digits after dot
Error Latitude	X.XXX	Fixed length 3 digits after dot
Error Longitude	X.XXX	Fixed length 3 digits after dot
Error Altitude	X.XXX	Fixed length 3 digits after dot

Example:
$GNGST,112900.00,9.7283,1.7815,0.2781,003.3933,1.7268,0.5190,2.3143*7A

5.5.12 GSV – GNSS satellites in view

TALKER ID	XX	GN must not be used*
SENTENCE ID	GSV	
Total number of messages	X	1-9
Message number	X	1-9
Total number of satellites in view	XX	Fixed length: 01 for single digit
Satellite ID number	XX	Fixed length: 01 for single digit Empty field if not used for positioning GPS: ID is PRN (1-32) GLONASS: ID is slot number + 64 GALILEO: PRN + 300
Elevation	XX	Fixed length: 00 for 0° elevation Values from 0 to 90 Empty if not used
Azimuth	XXX	Fixed length: 000 for 0° azimuth Values from 000 to 360 Empty if not used
SNR	XX	Fixed length: 05 for 5 db/Hz Values from 0-99 Empty if not used
Signal ID		**-Not Supported-**

*If multi GNSS is used, a separate GSV sentence must be created for each constellation starting with the constellation specific talker ID.

This block is repeated 4 times per sentence in total. For more than multiples of 4 a new sentence is started each time. Blocks are left empty if the number of satellites in view is lower than a multiple of 4.

Example:

$GPGSV,2,1,05,10,34,053,46,32,43,090,44,08,61,190,26,27,30,157,38,*53

$GPGSV,2,2,05,22,11,327,24,,,,,,,,,,,,*60

$GAGSV,1,1,04,99,36,123,30,103,78,023,26,96,33,085,29,77,30,058,41,*7C

5.5.13 PASHR – Attitude Data

TALKER ID		No talker ID
SENTENCE ID	PASHR	
UTC time	hhmmss.ss	Fixed length 2 digits after dot “h” hours, “m” minutes, “s” seconds
Heading*	XXX.XX	Fixed value: 000.00 for 0° 3 digits before dot, 2 after
	T	True heading
Roll angle*	(-)XXX.XX	Fixed value: 000.00 for 0° 3 digits before dot, 2 after
Pitch angle*	(-)XXX.XX	
Heave		**-Not Supported-**
Roll standard deviation*	XX.XXX	Fixed value 2 digits before dot 3 after
Pitch standard deviation*	XX.XXX	
Heading standard deviation*	XX.XXX	
Quality flag	X	0: No position 1: RTK float position 2: RTK fixed position

* attitude angles and corresponding deviation values are only filled for defined setup (3D, 2D)

Example:

$PASHR,190558.56,107.09,T,,-0.16,,,0.067,0.056,2*34

5.5.14 RMC – Recommended minimum specific GNSS data

TALKER ID	XX	All talker IDs usable
SENTENCE ID	RMC	
UTC time	hhmmss.ss	Fixed length 2 digits after dot “h” hours, “m” minutes, “s” seconds
Status	X	A: data valid
Latitude	ddmm.mmmmmm	Fixed length 4 digits before and 6 after dot ”d” degrees, ”m” minutes
Hemisphere of latitude	N/S	N if value of latitude is positive
Longitude	dddmm.mmmmmm	Fixed length 5 digits before and 6 after dot ”d” degrees, ”m” minutes
Hemisphere of longitude	E/W	E if value of longitude is positive
Speed over ground	X.XX	Variable/fixed length 2 digits after dot, variable before
Course over ground	X.XX	Variable/fixed length 2 digits after dot, variable before Values from 0 to 359.99
Date	ddmmyy	“d” day, “m” month, “y” year

Magnetic variation		**-Not Supported-**
	E/W (unit)	**-Not Supported-**
Mode indicator		A: Autonomous N: Invalid
Navigation Status		**-Not Supported-**

Example:
$GNRMC,085220.60,A,4808.151472,N,01130.274893,E,0.02,1.97,240625,,,A,V*3B

5.5.15 ROT – Rate of Turn Data

TALKER ID	XX	
SENTENCE ID	ROT	
Rate of turn	(-)X.XX	In degrees / minute. Variable/fixed length: 2 digits after dot, variable before
Valid/Invalid	X	A: Valid data V: Invalid data

Example:
$GNROT,35.6,A*4E

5.5.16 THS – True Heading State Data

TALKER ID	XX	
SENTENCE ID	ROT	
True Heading	XXX.XX	Fixed value: 000.00 for 0° 3 digits before dot, 2 after
Valid/Invalid	X	A: Valid data V: Invalid data

Example:
$GNTHS,35.6,A*4E

5.5.17 VTG – Course over ground and ground speed

TALKER ID	XX	All talker IDs usable
SENTENCE ID	VTG	
True Course over ground	X.XX	Variable/fixed length 2 digits after dot, variable before Values from 0 to 359.99
Unit	T	Degrees (true course)
Magnetic Course over ground	X.XX	**-Not Supported-**
Unit	M	Degrees (magnetic course)
Speed over ground (Knots)	X.XX	Variable/fixed length 2 digits after dot, variable before
Unit	N	Knots
Speed over ground (Km/h)	X.XX	Variable/fixed length 2 digits after dot, variable before

Unit	K	Km/h
Mode indicator	X	A: Autonomous mode N: Invalid

The VTG sentence is empty ($GNVTG,,T,,M,,N,,K,A*3D) until attitude baseline is valid. Course over ground equals heading.

Example:
$GNVTG,112.99,T,109.99,M,0.15,N,0.08,K,A*3B

5.5.18 ZDA – Time and date

TALKER ID	XX	All talker Ids usable
SENTECE ID	ZDA	
UTC time	hhmmss.ss	Fixed length 2 digits after dot “h” hours, “m” minutes, “s” seconds
Day	XX	Fixed length 01 to 31
Month	XX	Fixed length 01 to 12
Year	XXXX	
GMT Offset hours		**-Not Supported-**
GMT Offset minutes		**-Not Supported-**

Example:
$GNZDA,185823.40,13,01,2017,,*7E

6 Foxglove Studio

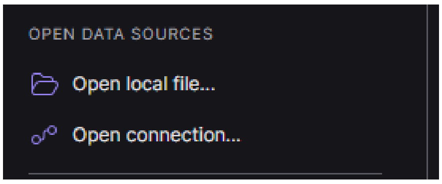
The free to use software Foxglove Studio can be used to visualize the data in replay modus. It is both possible to playback recorded MCAP files and visualize live data

- Download: [Foxglove Studio]
- Install Foxglove Studio.
- Create an account.

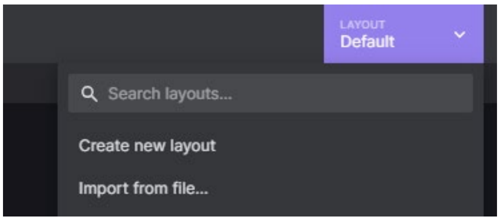
Foxglove allows you to build customized views and analyses that fit your specific need. For getting started, a template is available. Contact ANavS for further details.

To use this layout:

- Download the “A-ROX Foxglove Layout” layout.
- Start Foxglove Studio.
- Open the desired MCAP file (*Open local file...*) or open a new live connection.



- In the top-right corner, open “Layout” and select “Import from file...”.



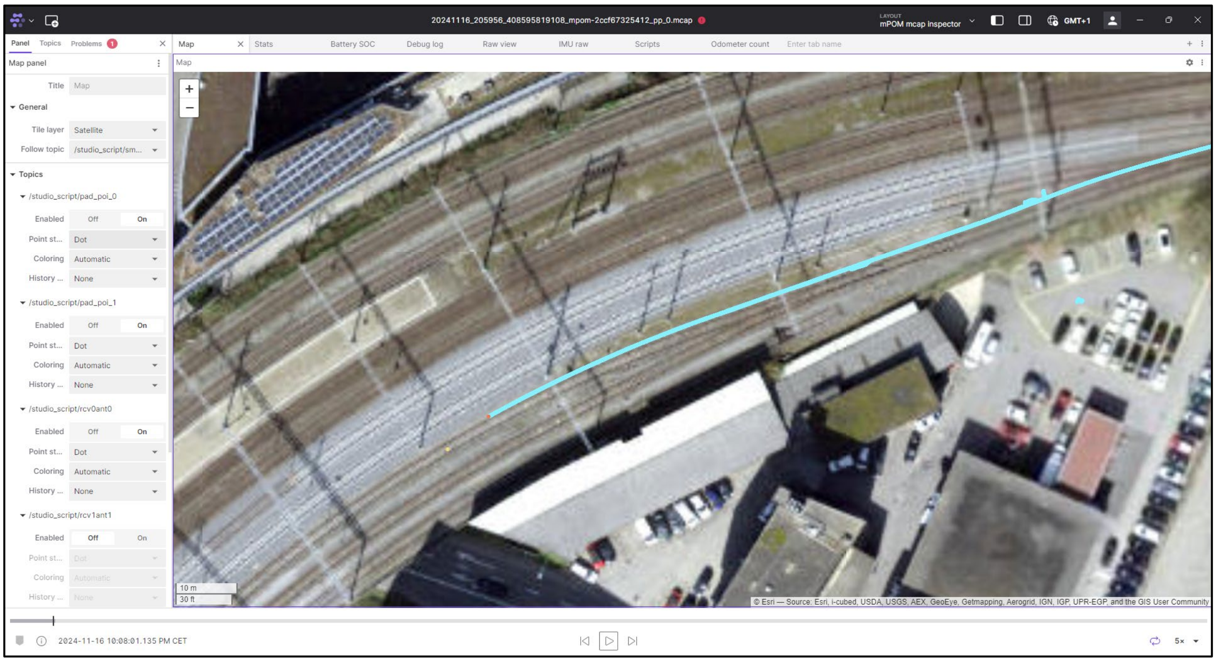
- Select the layout “A-ROX Foxglove Layout”.

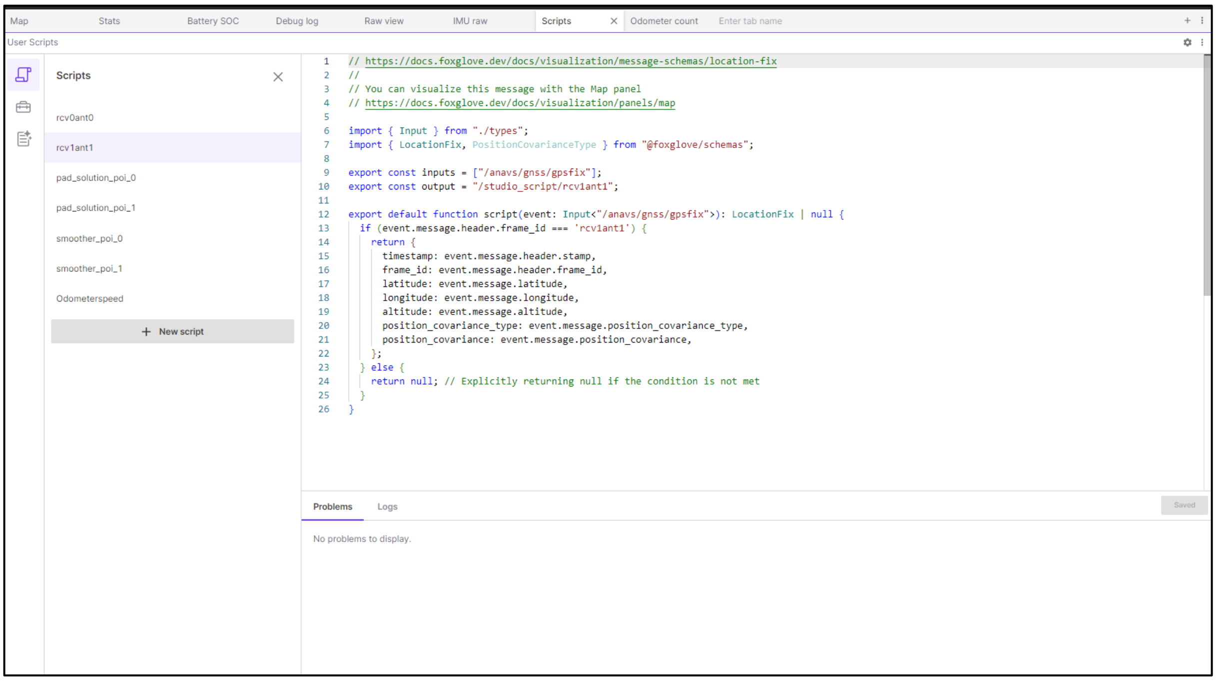
The layout contains multiple preconfigured tabs to visualize different topics. It is highly customizable, allowing users to modify views for detailed analysis.

6.1 Map view

The “Map” tab provides a preconfigured map view, where the solution for different POIs can be displayed. It is also possible to visualize intermediate processing steps and raw antenna positions. To modify the display, click on the map and use the left panel to select how and which data sources should be displayed.

Note:
To display a data source on the map, the format must be adjusted using a script. These scripts can be accessed and modified in the “Scripts” tab.





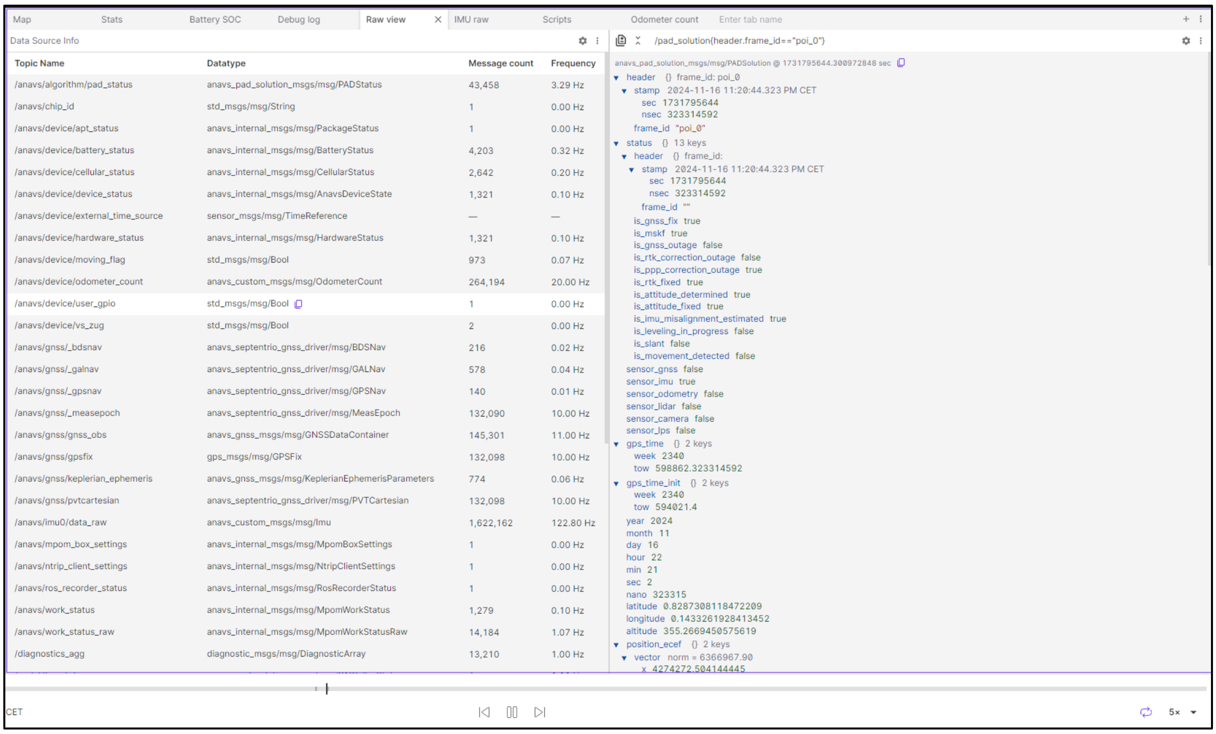
6.2 Raw Data Output

The **"Raw View" tab** contains:

- A **list of all topics** in the MCAP file (left panel).
- **Metadata** such as the **number of messages per topic**.

To **visualize a topic**, enter its name in the **address bar** in the right panel.

Useful feature: Filtering topics.
For example, to **display only POI 0 from the topic /pad_solution**, use the **message path syntax**. See: **Message Path Syntax | Foxglove | Docs**.



6.3 Lever Arms & Rotations

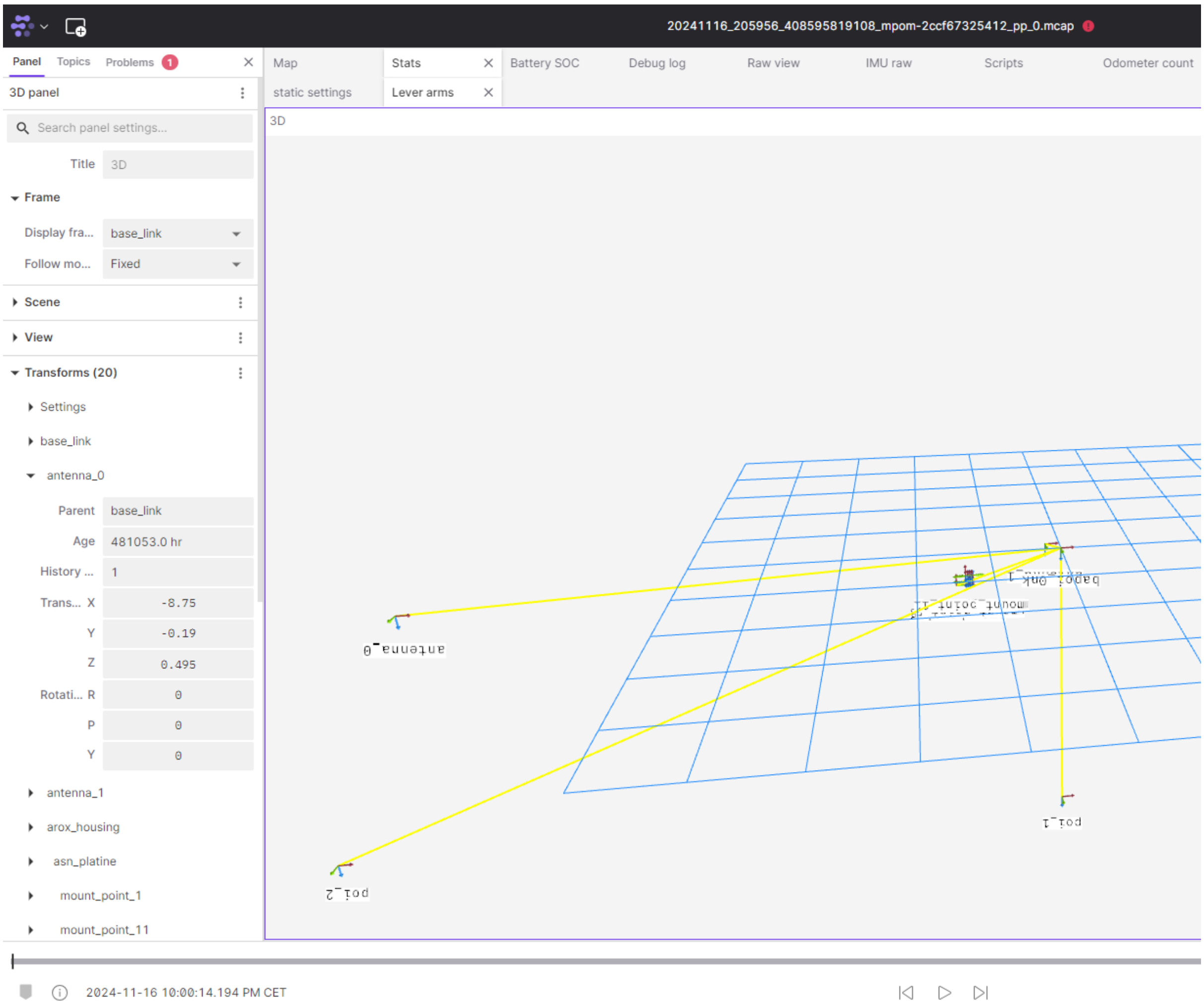
The **positions and rotations** of the **system components** are stored in the **/tf_static topic**.

- Messages are **recorded once per position** at the beginning.
- To visualize them, open: **"Stats" → "Lever arms"**.

Clicking inside this section will allow users to view detailed **transform values** in the **left panel under "Transforms"**.

Important Notice:

- **Foxglove Studio uses a left-hand rule coordinate system**.
- **mPOM uses a right-hand rule coordinate system**.
- This means that **the Z-axis is mirrored** in the **Foxglove visualization**.


20241116_205956_408595819108_mpom-2ccf67325412_pp_0.mcap
Panel
Topics
Problems
Map
Stats
Battery SOC
Debug log
Raw view
IMU raw
Scripts
Odometer count
static settings
Lever arms
3D panel
3D
Title
Frame
Display fra...
base_link
Follow mo...
Fixed
Scene
View
Transforms (20)
Settings
base_link
antenna_0
Parent
Age
481053.0 hr
History ...
Trans... X
-8.75
Y
-0.19
Z
0.495
Rotati... R
P
antenna_1
arox_housing
asn_platine
mount_point_1
mount_point_11
2024-11-16 10:00:14.194 PM CET

Appendix-1: Drawing A-ROX Housing