Atlas User Guide

Version 1.7

Revision History

Version	Date	SW Version	Description
1.0	2020-11-1	v1.0.0+	Initial version
1.1	2020-12-1	v1.8.0+	● Added coarse calibration specification and calibration procedure details ● Documented REST API
1.2	2021-3-4	v1.12.0+	● Changed REST device status `error` field to `notices` array ● Added additional modes to REST `clear_state` message ● Added details about recording FusionEngine output
1.3	2021-5-28	v1.13.0+	● Changed `/v1/device/status` response ○ Renamed `developer_mode_enabled` to `developer_features_enabled` ○ Moved `notices` to the top of the message
1.4	2021-8-27	v1.14.0+	● Changed `/v1/log/status` response ○ Added `log_sequence_num` to logs entries
1.5	2022-2-8	v1.15.1+	● Updated device panel drawings ● Added REST commands for settings import/export and GNSS corrections source configuration
1.6	2023-6-8	1.17.1+	● Update images on document for new Atlas ● Updated Company logo ● Added documentation on Wheel Ticks
1.7	2025-2-13	V2.2.0 +	

Table Of Contents

1. Introduction **4**
1.1 Getting Started 4
1.2 Connections 5
1.3 Installation 8
1.3.1 Orientation 8
1.3.2 Measuring Lever Arms 9
1.3.3 Device Calibration 10
2. Operating Atlas **11**
2.1 User Interface Overview 11
2.1.1 Map View 11
2.1.2 Device Status View 14
2.1.3 Graphs View 15
2.1.4 Logs View 16
2.1.5 Settings View 16
2.1.5.1 Orientation Settings 16
2.1.5.2 GNSS Settings 17
2.1.5.3 Vehicle Settings: 18
2.1.5.4 Output Settings: 19
2.1.5.5 Network Settings 20
2.1.5.6 Device Settings 20
3. Data Interfaces **22**
3.1 Output Interfaces 22
3.1.1 Point One FusionEngine 22
3.1.1.1 ROS Message Support 22
3.1.2 NMEA 0183 22
3.2 Data Logging 23
3.3 REST Control API 24
3.3.1 Device Controls 24
3.3.1.5 GET /api/v1/device/status 24
3.3.1.6 POST /api/v1/device/factory_reset 25
3.3.2 Navigation Engine Controls 25
3.3.2.1 POST /api/v1/application/start 25
3.3.2.2 POST /api/v1/application/stop 25
3.3.2.3 POST /api/v1/application/clear_state 26
3.3.2.4 POST /api/v1/application/reset_calibration 26
3.3.2.5 GET /api/v1/application/get_corrections_settings 27
3.3.2.6 POST /api/v1/application/set_corrections_settings 27
3.3.2.7 GET /api/v1/application/export_settings 28
3.3.2.8 POST /api/v1/application/import_settings 29

1. Introduction

This guide will familiarize you with the Point One Atlas inertial navigation system and provide step-by-step instructions for using it.

To use Atlas you will need:

- An Atlas device
- An Ethernet or Wi-Fi network with an internet connection
    - Ethernet required for initial setup
- A computer or tablet device
- A GNSS antenna
    - Triple-frequency (L1/L2/L5) recommended
- A connection to your vehicle's CAN bus, used for wheel speed measurements
    - Recommended for best performance, but not required

Note: CAN message formats vary by manufacturer and vehicle model/year. Please contact Point One to enable support for your vehicle's CAN data.

1.1 Getting Started

Follow these steps to begin using your new Atlas device:

1. Rigidly mount the device and carefully measure the lever arms as described in Section 1.3.
2. Connect the device to a GNSS antenna and an Ethernet network as described in Section 1.2.
    - Note: The Ethernet network must have DHCP and an available internet connection. For normal operation Atlas can use a Wi-Fi connection and/or static IP address, but it does not currently support Wi-Fi or static IP for initial setup.
3. Connect power to the device.
4. Scan the QR code on the top of the device with a computer or tablet on the same network as Atlas, then follow the listed URL. This will automatically detect the IP address assigned to your device and take you to the device's user interface webpage.
    - Note: If you are not on the same network as your Atlas device, the webpage referenced by the QR code will simply display Atlas's assigned local IP address, which you can enter into a browser manually.
    - It may take up to 30 seconds for Atlas to boot before the web page displays the IP address.

5. On the UI menu, select **Settings**, then enter the device orientation and lever arms you measured in step 1 and click **Save**.
    - By default, Atlas outputs the position of the body frame (center of the rear axle). You can configure the output lever arm to relocate the position to anywhere you would like on the vehicle.
6. Navigate to the **Map View** page.

Note: The GNSS antenna must be outside in view of the sky for the device to navigate for the first time.

1.2 Connections

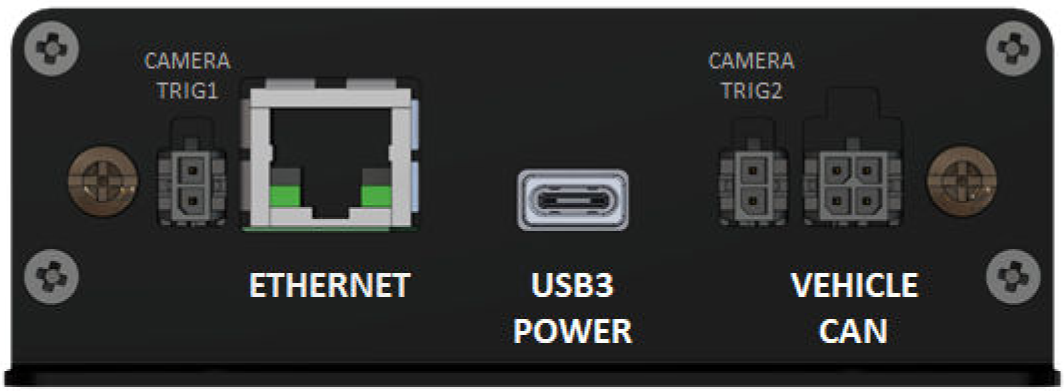


Figure 1: Atlas back panel connections.

Figure 1 shows the available connections on the back panel of your Atlas device. Atlas should be powered through the 4-pin CAN connector for operation in a production environment. The USB3 power is *only for desktop development.*

Note: Powering Atlas via USB C requires a specific, 2A capable power adapter that ships with the unit. **Atlas will not work properly if powered through a hub or from a laptop computer.**

The CAN bus connector is oriented as follows:

CAN H	CAN L
Vin	GND

Table 1: CAN bus pinout.

The CAN H and CAN L connectors are *not* terminated with a 120 ohm resistor.

The Vin pin on the CAN connector may be used to power Atlas. It can accept DC voltage levels from +5.5V to +55V, and is typically connected to the vehicle's +12V auxiliary power bus.

The camera triggers and wheel tick input are +3.3V signals, oriented as follows

Signal
GND

Table 2: Camera trigger/Wheeltick pinout.

Note: The camera trigger port 1 is configured as a wheel tick input for legacy applications. While this will help with dead reckoning substantially, the preferred method for wheel speed input is via the CAN bus. Please contact your sales representative for help in configuration for wheel tick input.

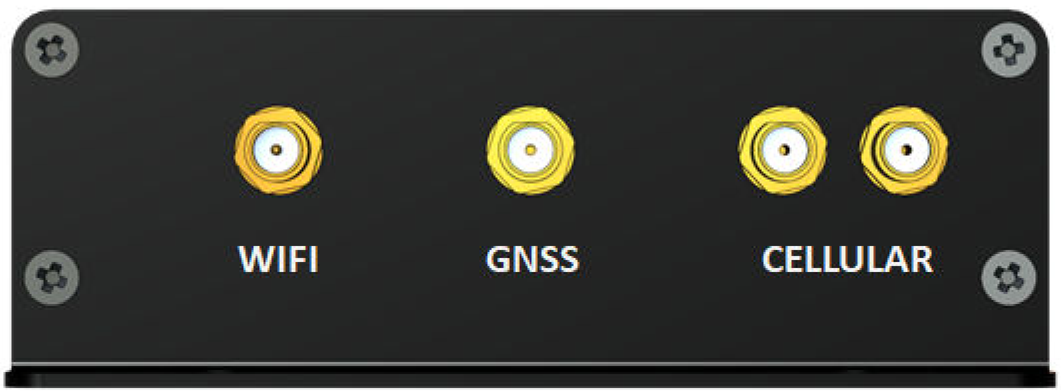


Figure 2: Atlas front panel connections.

The front panel of the device, shown in Figure 2, has four SMA RF connectors. The left-most connector is for an external Wi-Fi antenna (included). Note the Wi-Fi SMA connector is the “reverse polarity” variant.

The middle connector is for the GNSS antenna. It is designed to be used with a powered antenna/LNA, and provides a 3.3V DC supply voltage. Proper antennas are available from the Point One online store (https://store.pointonenav.com).

> Warning: If your antenna is already powered by another source (another GNSS receiver, powered bias-T, etc.), you must attach a DC block between the power source and the Atlas antenna connection to avoid damage to the Atlas GNSS receiver.

The two connectors on the right are for an optional cellular antenna, used to connect to the internet by Atlas devices configured with an internal cellular modem in order to receive GNSS corrections data.

Note that the Atlas cellular connection cannot currently be used to serve map data or provide internet access to other connected devices such as a laptop.

1.3 Installation

1.3.1 Orientation

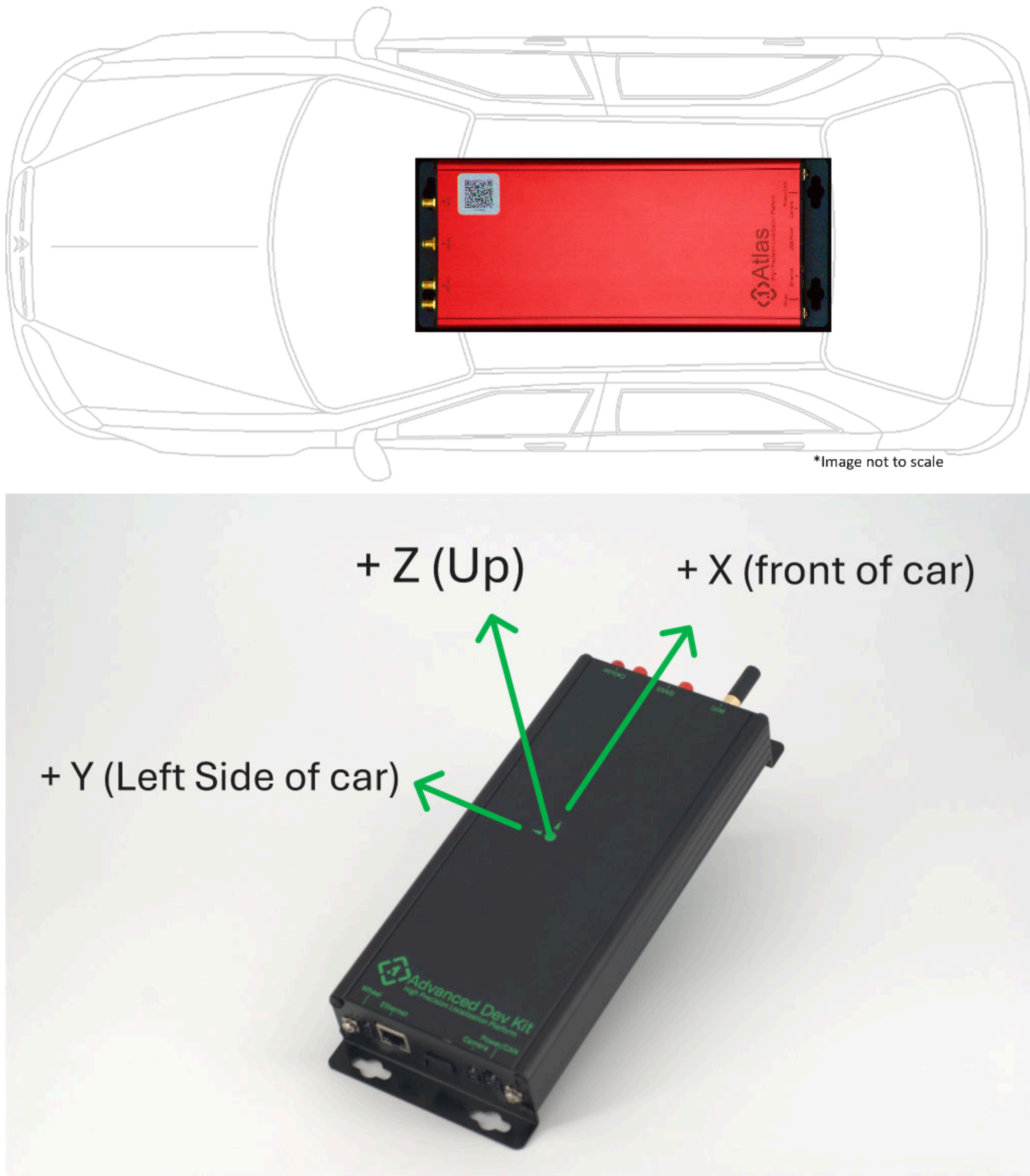


Figure 3: Recommended device orientation.

Both Atlas and the GNSS antenna must be mounted rigidly to your vehicle. The device calibration procedure will estimate the angle between the device (IMU) axes and the vehicle body axes (see Section 1.3.3). For best performance, you should try to install it so that the device axes are aligned as closely with the vehicle axes as possible.

Note: If the device lever arm or orientation changes (i.e., Atlas is moved), the existing calibration will no longer be valid and the device will need to be recalibrated.

We recommend installing Atlas with its x, y, and z axes matching the vehicle body frame (see Section 1.3.3). In this orientation, the top panel (the panel with the QR code sticker) will face up towards the roof of the vehicle and the front panel (the panel with the SMA antenna connectors) will face forward toward the front of the vehicle.

If you orient it differently, make sure to specify the correct orientation when configuring the device settings (see Section 2.1.4).

1.3.2 Measuring Lever Arms

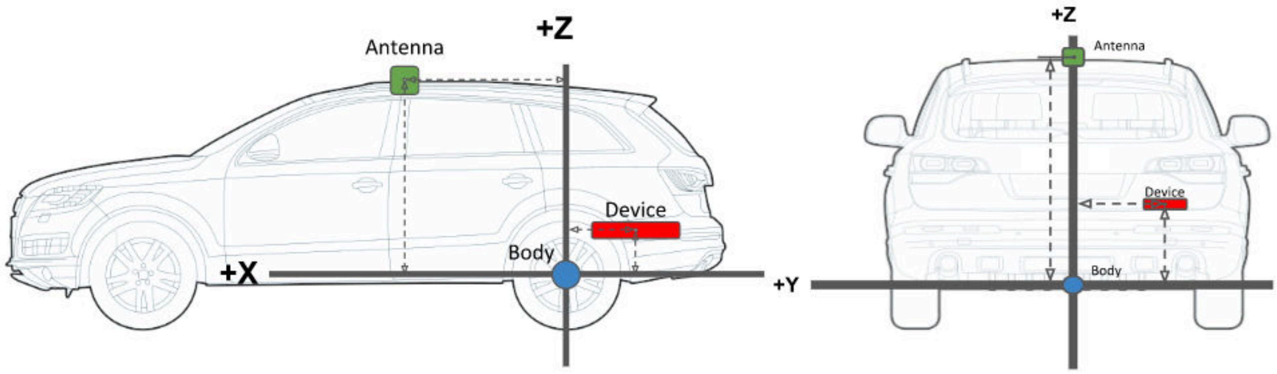


Figure 4: Body frame and lever arm definition

Once installed, you must measure the lever arms from the vehicle body frame to both Atlas and the GNSS antenna. For best performance, it is important to measure the lever arms as accurately as possible.

The Atlas body frame is a right-handed coordinate system centered at the middle of the rear axle of the vehicle, and oriented as follows:

+x - Toward the front of the vehicle
+y - Toward the left side of the vehicle
+z - Up

The lever arms are defined as the vector to the sensor from the vehicle body frame origin, resolved in the body frame. For instance, if the Atlas device is located behind and above the rear axle, toward the right-hand side of the vehicle (as in the diagram above), then the Atlas lever arm will have a negative x component, a negative y component, and positive z component. Ideally these measurements will be accurate to better than 5 cm.

The lever arms should be measured at the nominal phase center of the antenna, and at the nominal center of the IMU within the Atlas device, which is approximately the center of the chassis. For an exact measurement, measure to the dot on the chassis. You should measure to this point when configuring the device location.

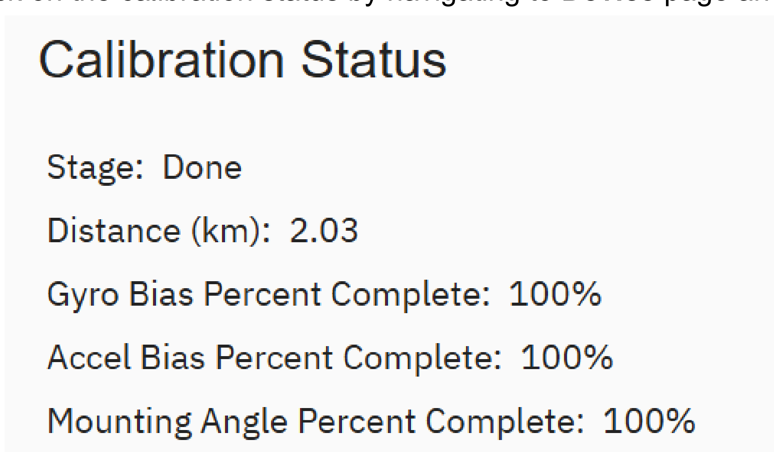
1.3.3 Device Calibration

When you first use your Atlas device, it will need to be calibrated. The calibration procedure accurately measures the orientation of the device with respect to the vehicle axes by analyzing navigation performance over time. For best calibration results, it is important to measure the lever arms as accurately as possible (see Section 1.3.2).

To calibrate your device, first go to the **Settings** page and specify the device orientation and the lever arms you measured previously. Then, simply navigate to the **Map** page and drive normally. Depending on your driving and the route you take, the calibration procedure may take some time to converge (typically 20 km or less). Note that turns and other maneuvers can help the calibration converge faster.

When finished, the calibration indicator on the UI will show as above (also see Section 2.1.1) to indicate that the calibration was successful and the system is ready for precision operation.

You can also check on the calibration status by navigating to **Device** page and see status:

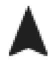


2. Operating Atlas

This section includes detailed information for using your Atlas device.

2.1 User Interface Overview

2.1.1 Map View

The map view is accessed by clicking the  icon on the left toolbar.

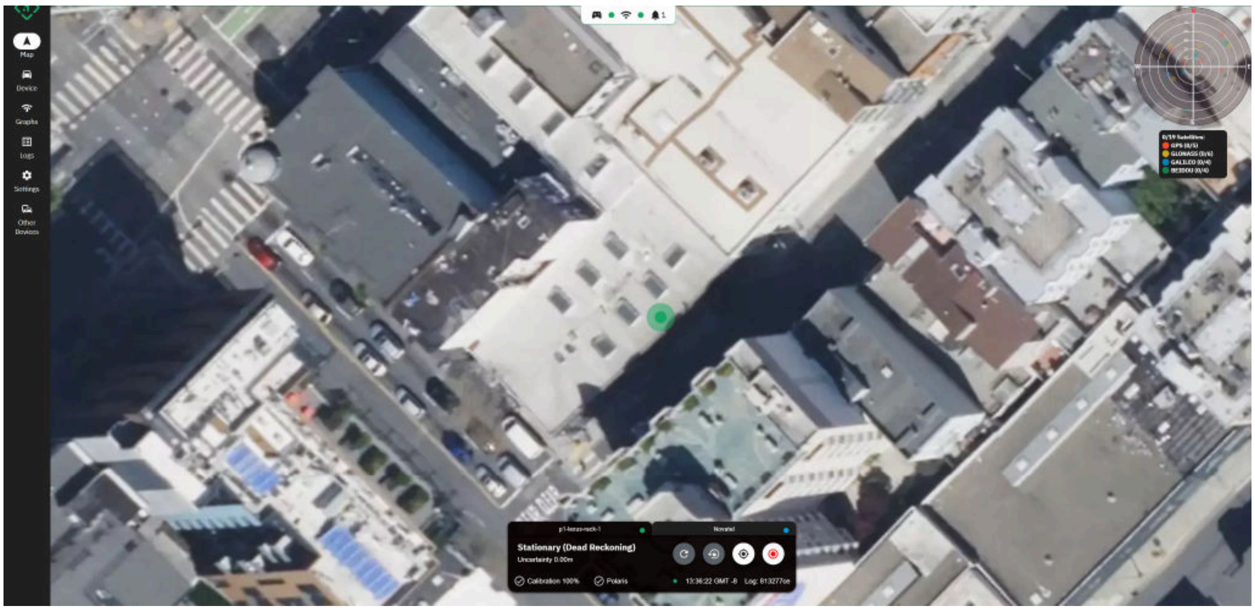


Figure 5: UI map view page.

The map view displays your vehicle's position and status in real time as the system navigates.

Figure 6: Log Start/Stop button.

Use the button in the bottom right corner of the display to start and stop logging

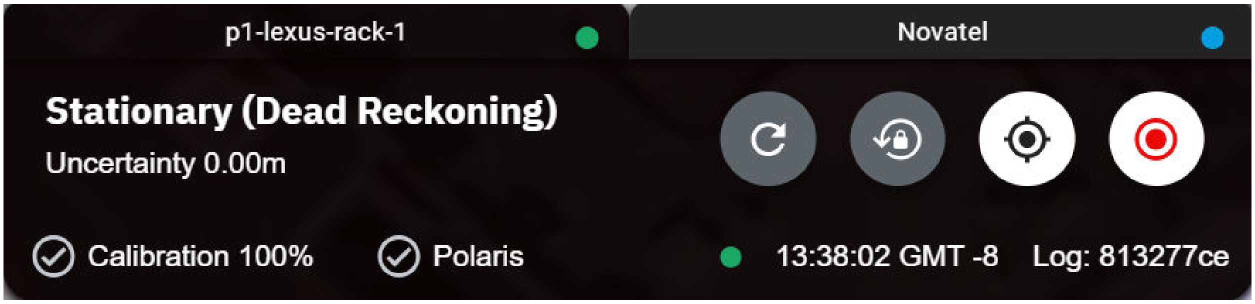


Figure 7: Map view dash panel.

The dash panel shown in Figure 7 provides a detailed view of the current solution in real time when the system is operating.

The **Solution Type** in the top left indicates the current navigation mode.

Dead Reckoning	The device is using the IMU, wheel speed (if available) and dynamical constraints for navigation and not using GNSS. This generally happens indoors, tunnels or when the vehicle is detected to be stationary.
Fixed INS	The highest positional accuracy where the GNSS has resolved integer ambiguities and coupled the solution with an IMU and other sensors.
Float INS	The device is using correction data in the GNSS solution with carrier phase data and coupling the IMU into the solution and other sensors.
Differential INS	The device is using correction data in the GNSS solution but unable to use carrier phase (likely due to a poor sky environment). It is still coupling the IMU and other sensors. into the solution.
Standalone INS	The device is using GNSS without any correction data and coupling the IMU and other sensors into the solution. This usually indicates the device has corrections disabled, has poor connectivity to the internet or is traveling outside the coverage area of the correction service.
System Initializing	The navigation system is not yet started.

The two fields on the bottom left indicate the current calibration status and the status of the connection to the Point One Polaris RTK network (or 3rd party NTRIP correction network if so configured).

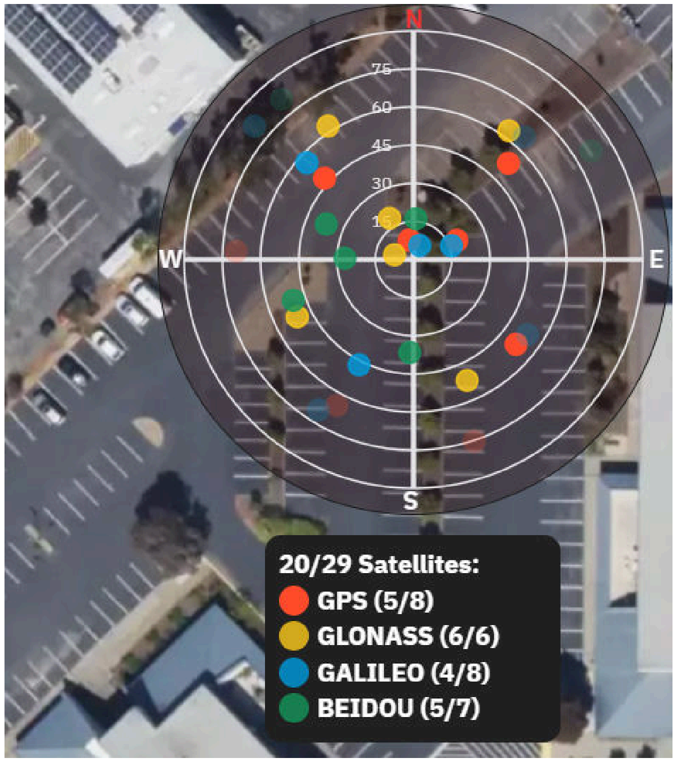


Figure 8: Map view Satellite Sky View

The top left of the Map View shows the Satellite Sky View, which shows first, the amount of satellites being used in the solution (20 in this case), and how many satellites are being tracked (28 in this case).

2.1.2 Device Status View

The device status view is accessed by clicking the ⓘ icon on the left toolbar.

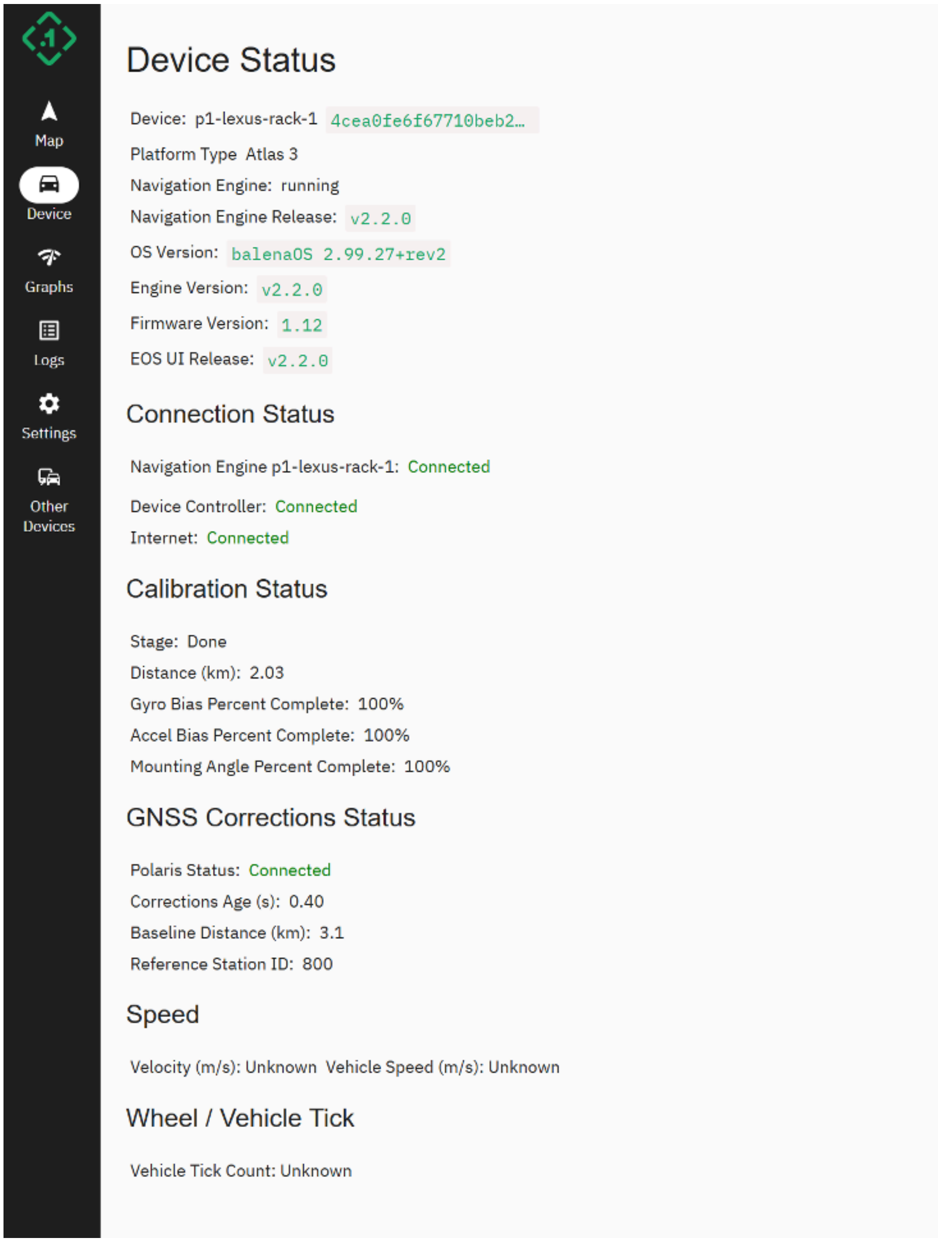


Figure 9: Device status view panel.

The device status page shows the device name, Nav Engine Release, Internet status, Calibration Status, GNSS Correction Status, along with any speed information available.

2.1.3 Graphs View

The detailed status view is accessed by clicking the icon on the left toolbar.

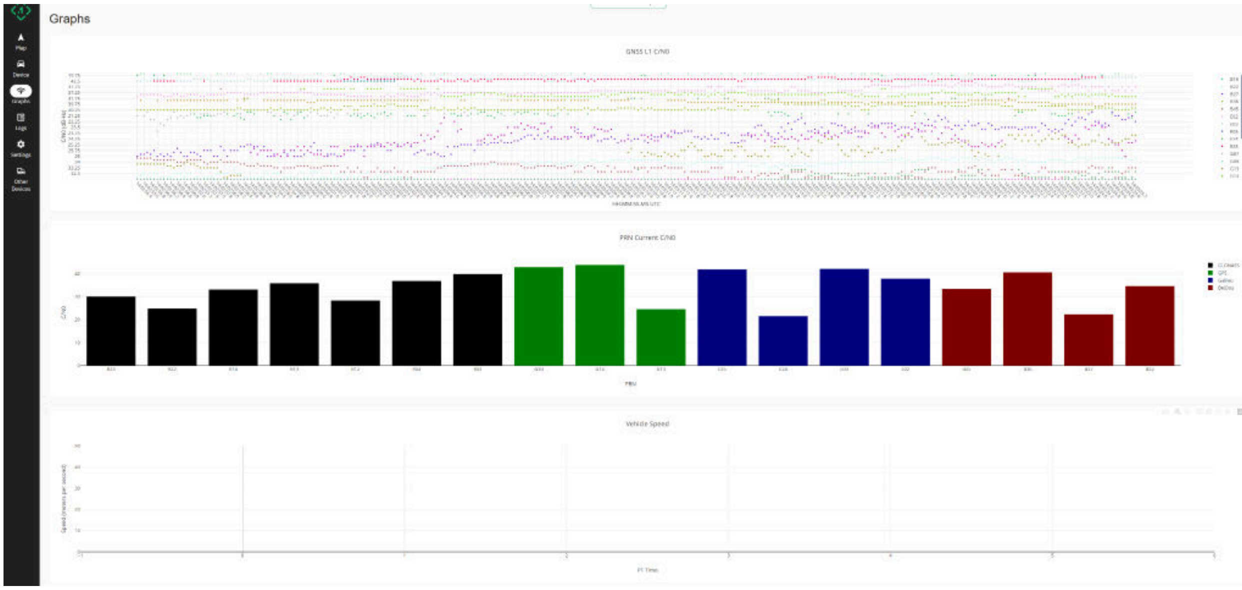
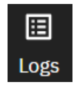


Figure 10: Example Graph Status display.

The Graphs page shows selected detailed information about the GNSS signal quality (C/N0) and vehicle speed in real time.

2.1.4 Logs View

The detailed status view is accessed by clicking the  icon on the left toolbar.

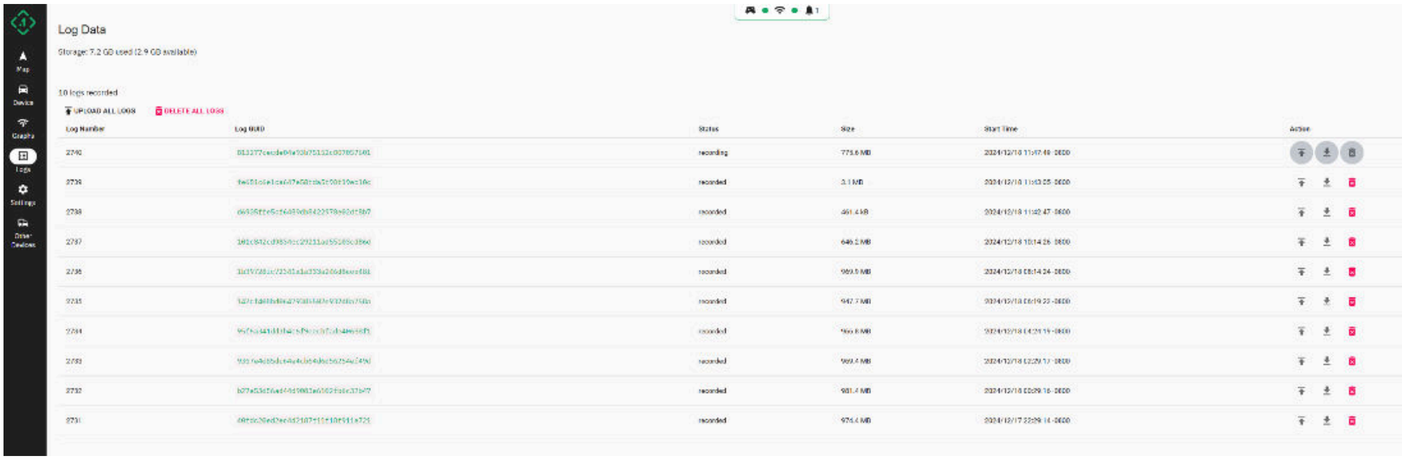


Figure 11: Log display

The Display shows the previously and currently recorded logs and allows users to store up to ~13GB of logs on the Atlas device locally. Those logs can be downloaded on any computer on the network and/or uploaded to Point One directly for troubleshooting via the UI.

2.1.5 Settings View

The settings view is accessed by clicking the  icon on the left toolbar.

The settings page includes settings for specifying the device orientation, sensor lever arms, controlling desired output data, network settings, resetting the device calibration, and more.

2.1.5.1 Orientation Settings

Before using your Atlas device, you must specify its orientation and location within the vehicle. See [Section 1.3] for details.

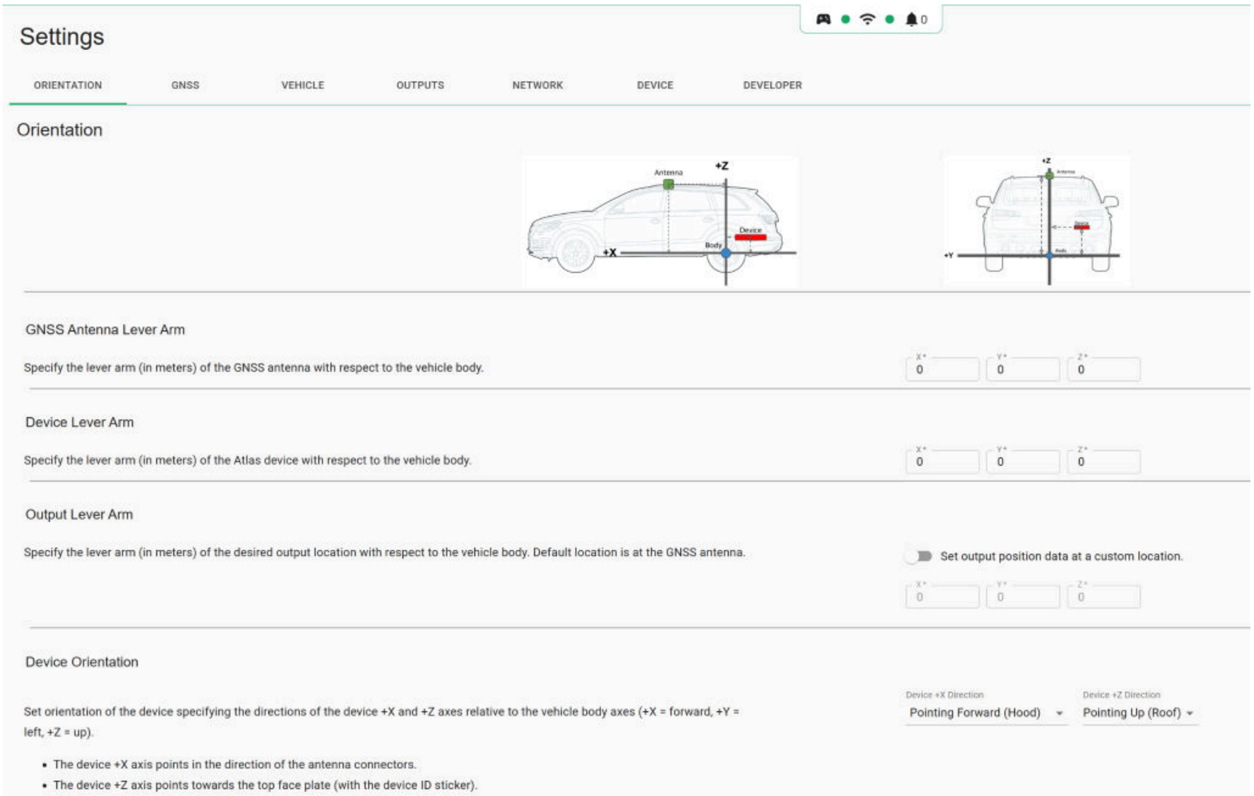


Figure 12: Orientation Settings display

The available settings include:

- **GNSS Antenna Lever Arm** - The location of the GNSS antenna
- **Device Lever Arm** - The location of the IMU internal to the Atlas device (see Section 1.3.2)
- **Output Lever Arm** - The location where the generated vehicle position solution will be located
- **Device Orientation** - The coarse (90 degree) rotation of the device with respect to the vehicle body axes (see Section 1.3)

All lever arms are measured with respect to the center of the vehicle's rear axle.

Note: If the device lever arm or orientation changes (i.e., Atlas is moved), the existing calibration will no longer be valid and the device will need to be recalibrated.

2.1.5.2 GNSS Settings

The **GNSS Settings** panel provides a way to change the Corrections Source. This will allow the user to change over from Polaris protocol to NTRIP if desired.

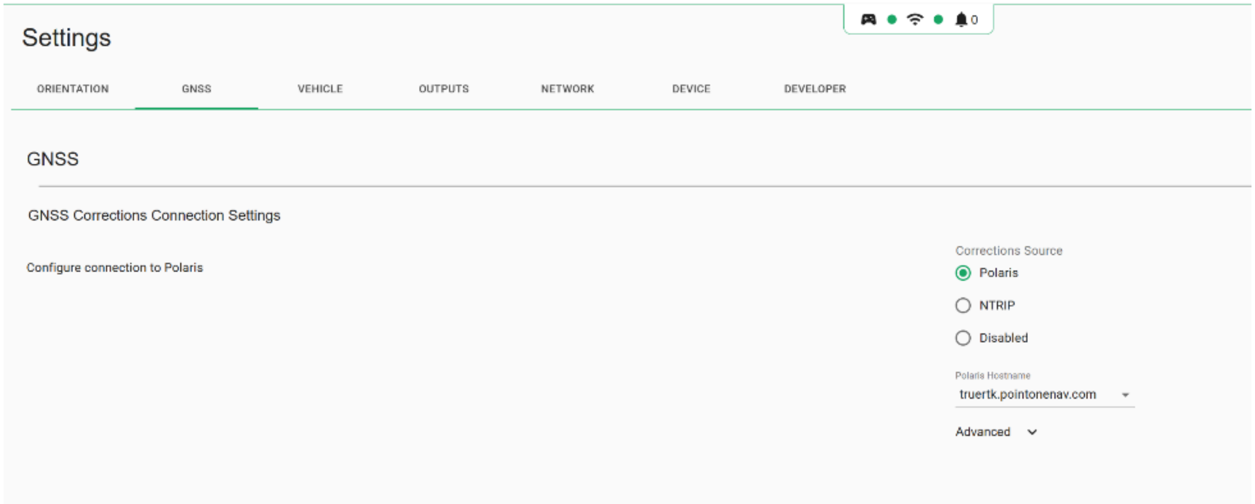


Figure 13: GNSS Settings display

2.1.5.3 Vehicle Settings:

The **Vehicle Settings** panel provides controls for configuring wheel speed input for an Atlas device. If planning on using a CAN profile that is already defined, select the correct vehicle model in the options menu, and everything else will auto populate.

If enabling Hardware Ticks, navigate to the Vehicle Settings panel, and for "Vehicle Select", select the drop-down menu and choose "Unknown Vehicle". After, enable Hardware ticks and input the necessary information about the wheel tick data.

Note: If trying to input hardware Wheel Ticks to the device, please connect your input to Camera Trig 1 shown in Figure 1.

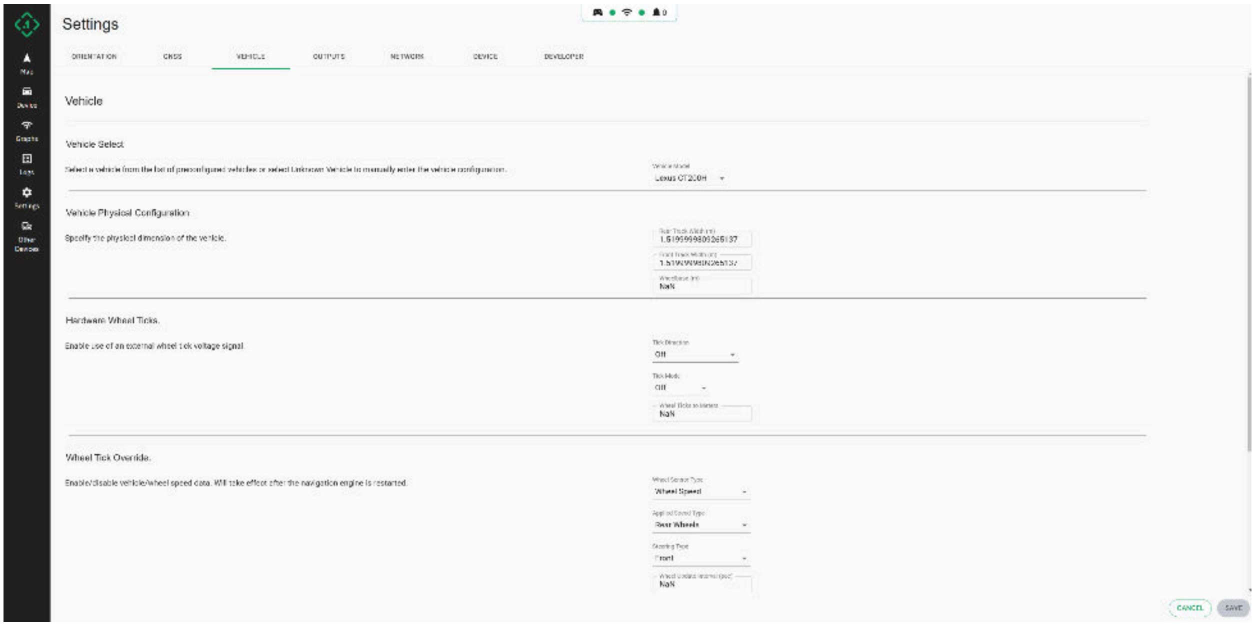


Figure 14: Vehicle Settings display

2.1.5.4 Output Settings:

Navigate to the **Output Settings** panel, where this allows the users to setup the output messages over TCP, UDP or the data logged on the device.

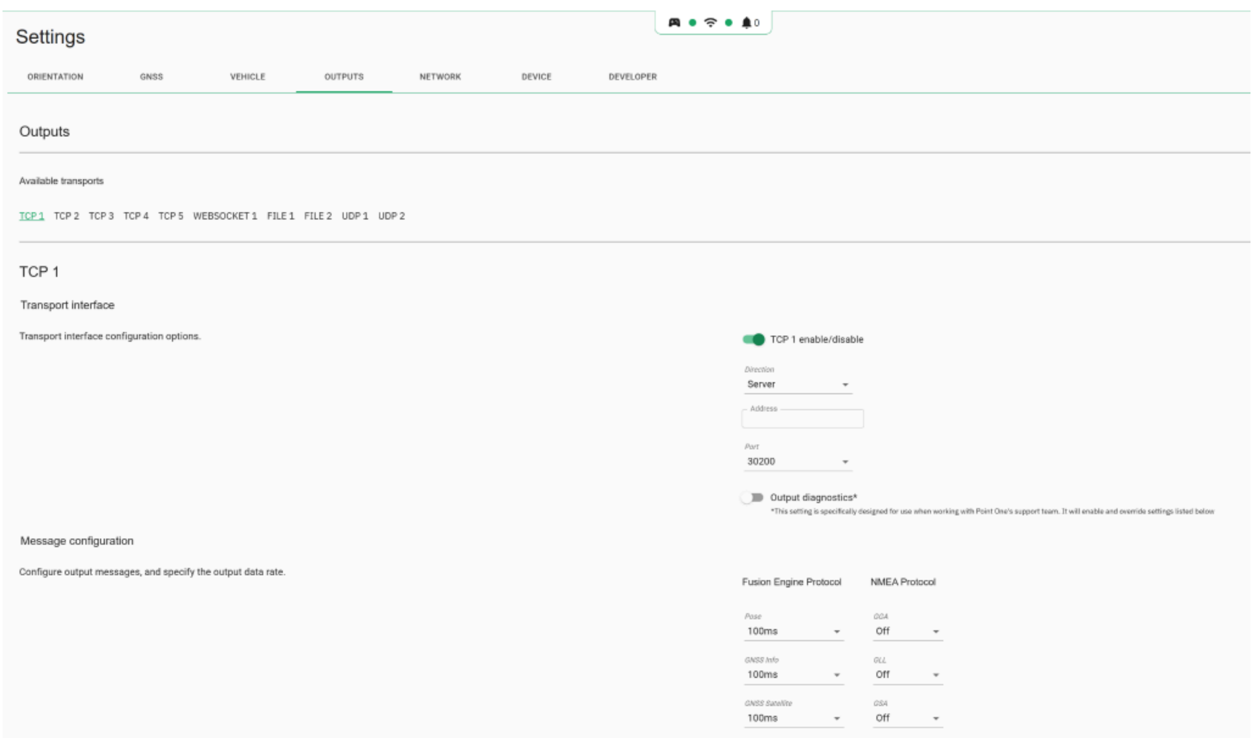


Figure 15: Output Settings display

> Note: If any analysis is ever needed to be done by Point One, please enable Output Diagnostics.

2.1.5.5 Network Settings

The **Network Settings** panel allows for the user to configure the device's internet connectivity. Through this menu, you can:

- Set Static IP via the ethernet menu
- Set up Wifi

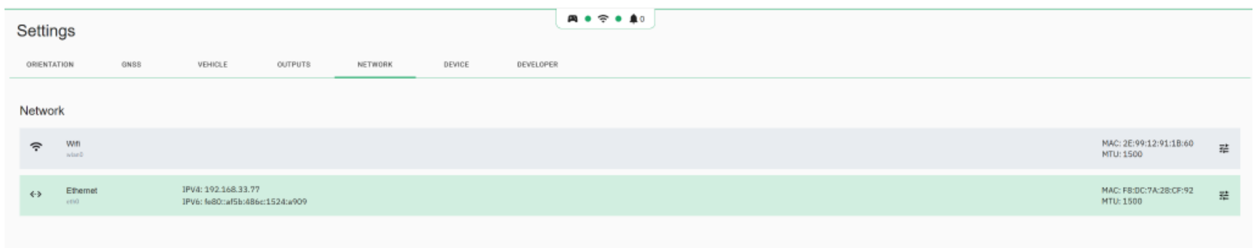


Figure 16: Network Settings display

2.1.5.6 Device Settings

The **Device Settings** panel enables the user to send device resets via the UI, including:

- Device Reset (Reset the device to apply config updates)
- Position Data Reset (Reset Current Position, Attitude, and Ephemeris Data - **Cold Start**)
- Navigation Engine Reset (Reset Position and Training Data)
- Calibration Reset (Reset the Navigation Engine and Calibration Settings)
- Factory Reset (Resets Device to original Factory settings)

Additionally, this menu has an option to log all data on device at start-up, if that is desired

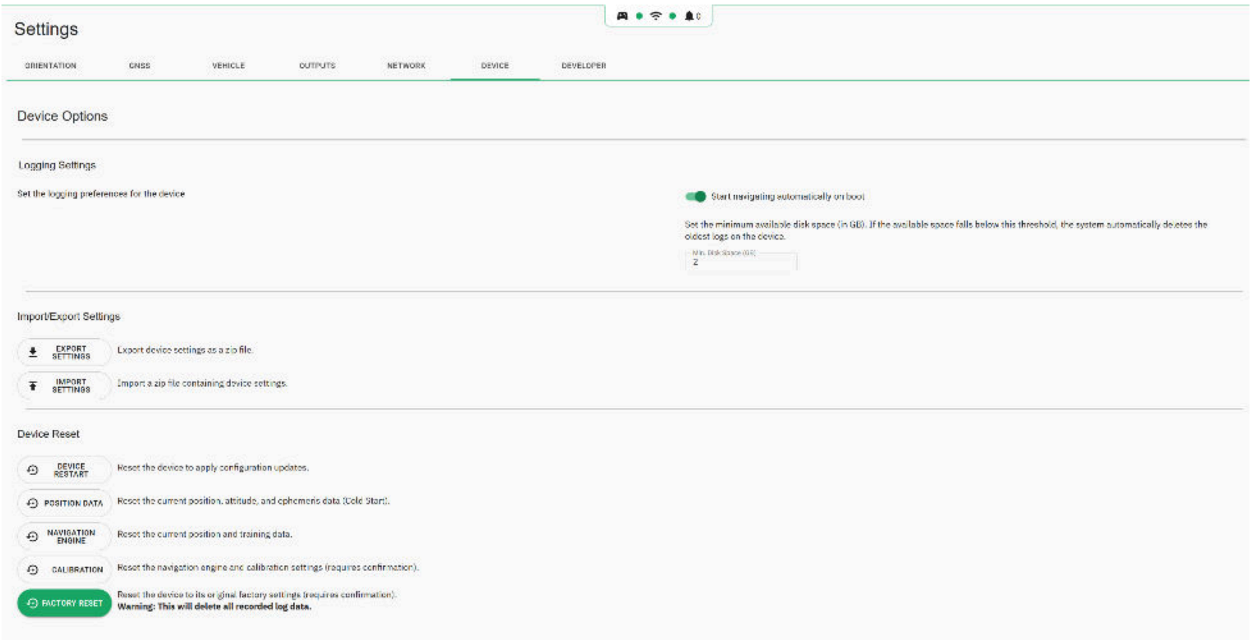


Figure 17: Device Settings display

3. Data Interfaces

3.1 Output Interfaces

3.1.1 Point One FusionEngine

Atlas uses the Point One FusionEngine protocol to send and receive data and commands in real time, as well as log data for post-processing. See https://github.com/PointOneNav/fusion-engine-client for message definitions, support source code, and tools for post-processing and data analysis. The latest specification is documented here: http://pointonenav.com/files/fusion-engine-message-spec

To configure the FusionEngine output, go to the **Settings -> Output Settings** panel. From there you can configure which output messages you would like to receive, and your desired output data rate.

The FusionEngine interface can be used in real time by connecting to TCP port 30201, or by sending data to a specified UDP destination hostname/IP address and port. Once connected, you will automatically receive configured data messages as the system operates.

In addition to real-time communication, you can configure the device to record FusionEngine messages to disk, generating a `fusion_engine.p1log` binary file. The recorded data can be downloaded later for post-processing.

3.1.1.1 ROS Message Support

FusionEngine supports a selection of messages that are designed to be content-compatible (but not byte-compatible) with corresponding ROS messages for easy integration into ROS. These include the ROS Pose, GPSFix, and IMU messages.

ROS messages can be enabled to be sent along with all other FusionEngine messages on the **Settings -> Output Settings** panel.

3.1.2 NMEA 0183

Atlas also includes support for generating selected messages from the NMEA 0183 standard in real time. To receive NMEA data, connect to TCP port 30200, or configure a UDP destination on the **Settings -> Output Settings** control panel.

Atlas currently supports the following NMEA messages:

- GPGGA
- GPRMC
- GPGSA
- GPGSV

3.2 Data Logging

Atlas creates a new log with an automatically generated ID each time you start navigating. Logs contain all incoming sensor data and internal information in a proprietary Point One format, as well as a binary file containing all FusionEngine messages that were generated in real time (if enabled). Data logs are designed to be used for:

- Post-processing data analysis (see Section 3.1.1)
- Customer assistance from Point One

Logs can be downloaded to your local machine by clicking the **Download** button on the **Device Status** page. This will generate a `*.tar.gz` file that you can extract locally. The extracted log directory will contain a `fusion_engine.p1log` file containing FusionEngine message content that you can use for offline analysis. See Section 3.1.1. All other files are meant for Point One internal use only.

If you encounter an issue and require further assistance, you may either download the log and send it, or click the **Upload to Point One** button to send the log to Point One automatically for processing (the device must remain on and connected to the internet while uploading the log). When uploading a log for assistance, please send an email to support@pointonenav.com, detailing any issues you encountered.

3.3 REST Control API

Atlas includes an HTTP REST API, which can be used to programmatically control the device. This is the same interface that is used by the UI to interact with the device. This section outlines the various supported commands and expected responses.

The HTTP interface is hosted on port 80. To use the REST API, send HTTP GET or POST requests to `http://DEVICE_IP/PATH`, where `DEVICE_IP` is the IP address of the Atlas device and `PATH` is the query path for a command listed in the sections below. For example, to start the navigation engine for a device with IP address `1.2.3.4`, issue a POST request to:

```
http://1.2.3.4/api/v1/application/start
```

3.3.1 Device Controls

3.3.1.5 GET /api/v1/device/status

Get the current device status.

Responds with a 200 response and a JSON object as follows:

- `platform_type` - A platform type name (i.e., `Atlas`)
- `platform_version` - The platform version string
- `device_uuid` - The unique ID assigned to the device
- `device_name` - The human-friendly device name
- `sw_release_uuid` - The unique ID of the software release running on the device
- `performing_reset` - `true` if a factory reset is in progress (optional)
- `nav_engine` - Navigation engine (Nautilus) status
    - `version` - The version of the Nautilus software
    - `status` - The current status of the navigation engine: `running`, `stopped`, `unknown`
- `developer_features_enabled` - `true` if developer mode is currently enabled
- `notices` - A list containing the most recent notifications from the device (errors, warnings, etc.). Cleared automatically on return. Each entry is an object containing:
    - `time` - The POSIX timestamp (in seconds) when the error occurred
    - `message` - The message string
    - `type` - The type of message (`info`, `warning`, `error`, `critical`)
    - `id` - A string ID providing context for this message, or `None` if not applicable

Example:

```
$ curl -X GET \
    "http://<IP_ADDRESS>/api/v1/device/status"
```

Response:


```
{"device_name":"bob_loblaw","device_uuid":"40fa62f7a85eb88f5548acf3a13812c0","nav_engine":{"developer_mode_enabled":false,"log":null,"notices":[],"status":"stopped","version":"v1.12.0"},"performing_reset":false,"platform_type":"Atlas","platform_version":"2","sw_release_id":"449f7dbe613b656b20a8e7e196fb4b32"}
```


3.3.1.6 POST /api/v1/device/factory_reset

Perform a factory reset. The navigation engine will be stopped automatically if it is currently running.

When successful, responds with an empty 200 response.

This is the same as `/api/v1/application/clear_state` with `"type": "factory_reset"`.

Example:

```
$ curl -X POST \
    "http://<IP_ADDRESS>/api/v1/device/factory_reset"
```

3.3.2 Navigation Engine Controls

3.3.2.1 POST /api/v1/application/start

Start the navigation engine if not running.

When successful, responds with an empty 200 response.

Example:

```
$ curl -X POST \
    "http://<IP_ADDRESS>/api/v1/application/start"
```

3.3.2.2 POST /api/v1/application/stop

Stop the navigation engine if running.

When successful, responds with an empty 200 response.

Example:

```
$ curl -X POST \
    "http://<IP_ADDRESS>/api/v1/application/stop"
```

3.3.2.3 POST /api/v1/application/clear_state

Perform one of the reset operations described in Section 2.3. The navigation engine will be stopped automatically if it is currently running.

When successful, responds with an empty 200 response.

Request body:

A JSON object containing a `type` parameter with one of the following values:

- `cold` - Reset the position data
- `nav_engine` - Reset the navigation engine
- `calibration` - Reset the navigation engine and calibration data
- `factory_reset` - Perform a full factory reset

Example:

Reset the device calibration:

```
$ curl -X POST \
    "http://<IP_ADDRESS>/api/v1/application/clear_state" \
    --header "Content-Type: application/json" \
    --data '{
        "type": "calibration"
    }'
```

3.3.2.4 POST /api/v1/application/reset_calibration

Reset the device calibration and previous position state. The navigation engine will be stopped automatically if it is currently running.

When successful, responds with an empty 200 response.

This is the same as `/api/v1/application/clear_state` with `"type": "calibration"`.

Example:

```
$ curl -X POST \
    "http://<IP_ADDRESS>/api/v1/application/reset_calibration"
```

3.3.2.5 GET /api/v1/application/get_corrections_settings

Get the current GNSS corrections settings.

Responds with a 200 response and a JSON object containing the keys described in Section 3.3.2.6.

Example:

```
$ curl -X GET \
  "http://<IP_ADDRESS>/api/v1/application/get_corrections_settings"
```

Response:

```
{"protocol": "ntrip", "ntrip_hostname":
"ntrip.polaris.pointonenav.com", "ntrip_port": 2101,
"ntrip_mountpoint": "polaris", "ntrip_password": "password1234",
"ntrip_username": "p1"}
```

3.3.2.6 POST /api/v1/application/set_corrections_settings

Set the GNSS corrections settings on the device.

When successful, responds with an empty 200 response.

Request body:

A JSON object containing the following values:

- `protocol` - The corrections source to be used: `polaris` or `ntrip` [required]

NTRIP parameters (ignored if `protocol` is `polaris`)

- `ntrip_hostname` - The hostname of the NTRIP server [required]

- `ntrip_port` - The TCP port of the NTRIP server [required]
- `ntrip_mountpoint` - The name of the desired NTRIP mountpoint [required]
- `ntrip_username` - A username used to authenticate with the NTRIP server [optional]
- `ntrip_password` - A password used to authenticate with the NTRIP server [optional]

Example:

Connect to the Point One Polaris network using the native Polaris protocol:

```
$ curl -X POST \
  "http://<IP_ADDRESS>/api/v1/application/set_corrections_settings" \
  --header "Content-Type: application/json" \
  --data '{
     "protocol": "polaris"
  }'
```

Connect to the Point One Polaris network using NTRIP:

```
$ curl -X POST \
  "http://<IP_ADDRESS>/api/v1/application/set_corrections_settings" \
  --header "Content-Type: application/json" \
  --data '{
     "protocol": "ntrip",
     "ntrip_hostname": "ntrip.polaris.pointonenav.com",
     "ntrip_port": 2101,
     "ntrip_mountpoint": "polaris",
     "ntrip_username": "p1",
     "ntrip_password": "password1234"
  }'
```

3.3.2.7 GET /api/v1/application/export_settings

Export all device settings as a Zip file.

Responds with a 200 response with MIME type `application/zip` and a Zip file containing the exported configuration settings.

Example:

```
$ curl -o settings.zip -X GET \
  "http://<IP_ADDRESS>/api/v1/application/export_settings"
```

Response:
Downloads a file called `config.zip` in the current directory.

3.3.2.8 POST /api/v1/application/import_settings

Import a zip folder containing exported settings files as returned by an `export_settings` request (see Section 3.3.2.7). All current configuration files will be deleted and replaced with the imported files.

When successful, responds with an empty 200 response.

Example:

Import a file called `config.zip` from the current directory.

```
$ curl -X POST \
  "http://<IP_ADDRESS>/api/v1/application/import_settings" \
  --form "file=@config.zip;type=application/zip"
```

Appendix A: Recommended Calibration Procedure

- [ ] Drive to an open sky area such as a parking lot
- [ ] Come to a complete stop and stay stationary for about 15 seconds
- [ ] Drive in a figure 8 pattern for about 3-4 minutes, making sure to alternate left and right turns.
- [ ] Come to a complete stop again and wait about 30 seconds.
- [ ] Perform a U-Turn
- [ ] Drive in a figure 8 pattern for about 3-4 minutes, making sure to alternate left and right turns, following your previous route approximately but going in the opposite direction.
- [ ] Come to a complete stop again and wait about 30 seconds.
- [ ] Perform normal driving until the “Calibration” light changes to green. This can take an extended period of time (45 minutes) depending on the distance traveled and the dynamics experienced by the vehicle.

Calibration is valid as long as the device is not moved in the vehicle. Even slight movements (for instance removing and putting it back in the same place) *require full recalibration*.

Appendix B: Disclosure

Users are solely responsible for ensuring that the INS is integrated with appropriate redundant navigation sources, safeguards, and validation mechanisms. These may include, but are not limited to, external sensors like cameras, radars, lidars, or maps, and/or human oversight, as applicable to the use case.

Under no circumstances should this device be relied upon as the exclusive means of navigation for safety-critical or autonomous operations. Failure to implement additional navigation references and fail-safe systems may result in property damage, personal injury, or other adverse outcomes.