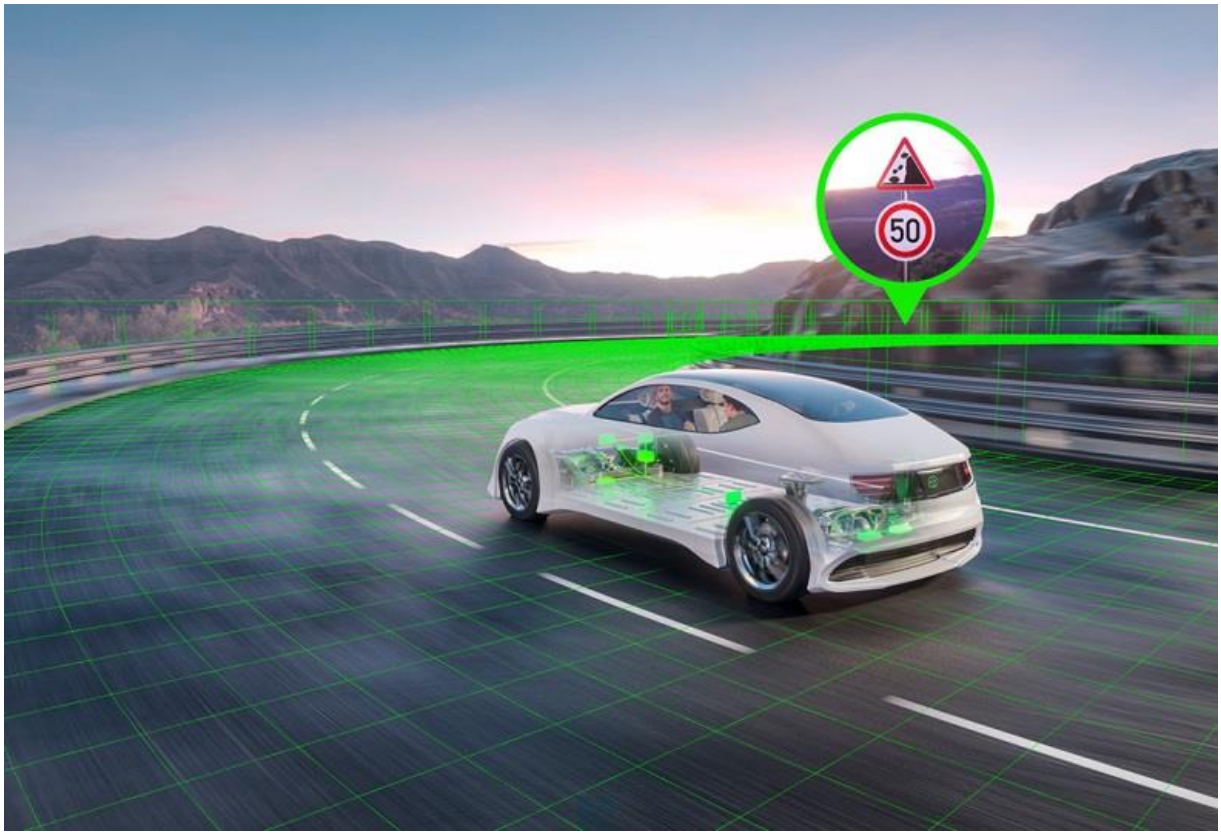
Elektrobit

EB robinos Predictor Eval Kit v1.9.0

Combined ADASISv2 and ADASISv3 Provider solution on Raspberry Pi
User guide



Version 1.9.0, released on 2025-03-28

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
Email: info@elektrobit.com

Technical support

https://www.elektrobit.com/support

Legal disclaimer

Confidential information

Table of contents

1 Overview of EB robinos Predictor Eval Kit

Elektrobit has been a member in the ADASIS AISBL (http://adasis.org/) for several years and actively participates in defining the next electronic horizon protocol version (ADASISv3). Along to this initiative, Elektrobit provides a wide range of products in the field of electronic horizon, both on the Provider and on the Reconstructor side. Elektrobit supports its customers in the field of R&D tooling (ADTF, Matlab Simulink) as well as by providing efficient and scalable solutions for series products with a high maturity grade. Due to many successfully launched OEM SOPs (on the road since 2012), Elektrobit has grown into one of the global market leaders in the field of electronic horizon. See also https://www.elektrobit.com/products/eb-robinos/predictor/ for details.

To showcase Elektrobit's expertise and experience, Elektrobit uses a Raspberry Pi as an R&D platform to demonstrate its electronic horizon products, further called **EB robinos Predictor Eval Kit**. The major advantages of EB robinos Predictor Eval Kit are:

- Small hardware dimensions (form factor)
- High robustness of hardware and software, ready for test drives
- Support for different I/O: CAN bus, Ethernet
- Low costs for customers

The EB robinos Predictor Eval Kit default delivery contains the following items which allow you to run the ADASISv2 and ADASISv3 Provider out of the box:

- Pre-assembled Raspberry Pi-based EB robinos Predictor Eval Kit
- Power supply
- GNSS receiver
- PEAK PCAN USB Connector
- Micro SD card
- Pre-installed software image with ADASISv2 and ADASISv3 Provider + Reconstructor

Additionally, you can configure and control EB robinos Predictor Eval Kit with any Wi-Fi- or Ethernet-capable device using the **WebApp user interface**. As a result, EB robinos Predictor Eval Kit provides an easy-to-use electronic horizon provider you can adjust to your environment.

This document helps you to put EB robinos Predictor Eval Kit into operation. It describes all relevant configuration parameters you can use to adjust the ADASISv2 and ADASISv3 Provider settings.

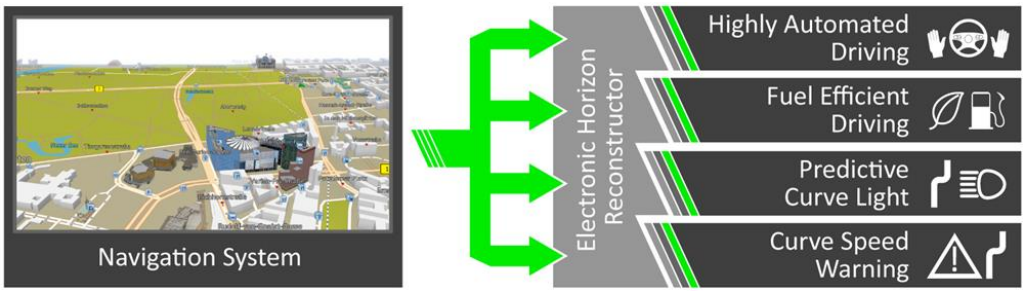


Figure 1: Benefits of an electronic horizon

2 System overview

The following figure shows all system modules of EB robinos Predictor Eval Kit:

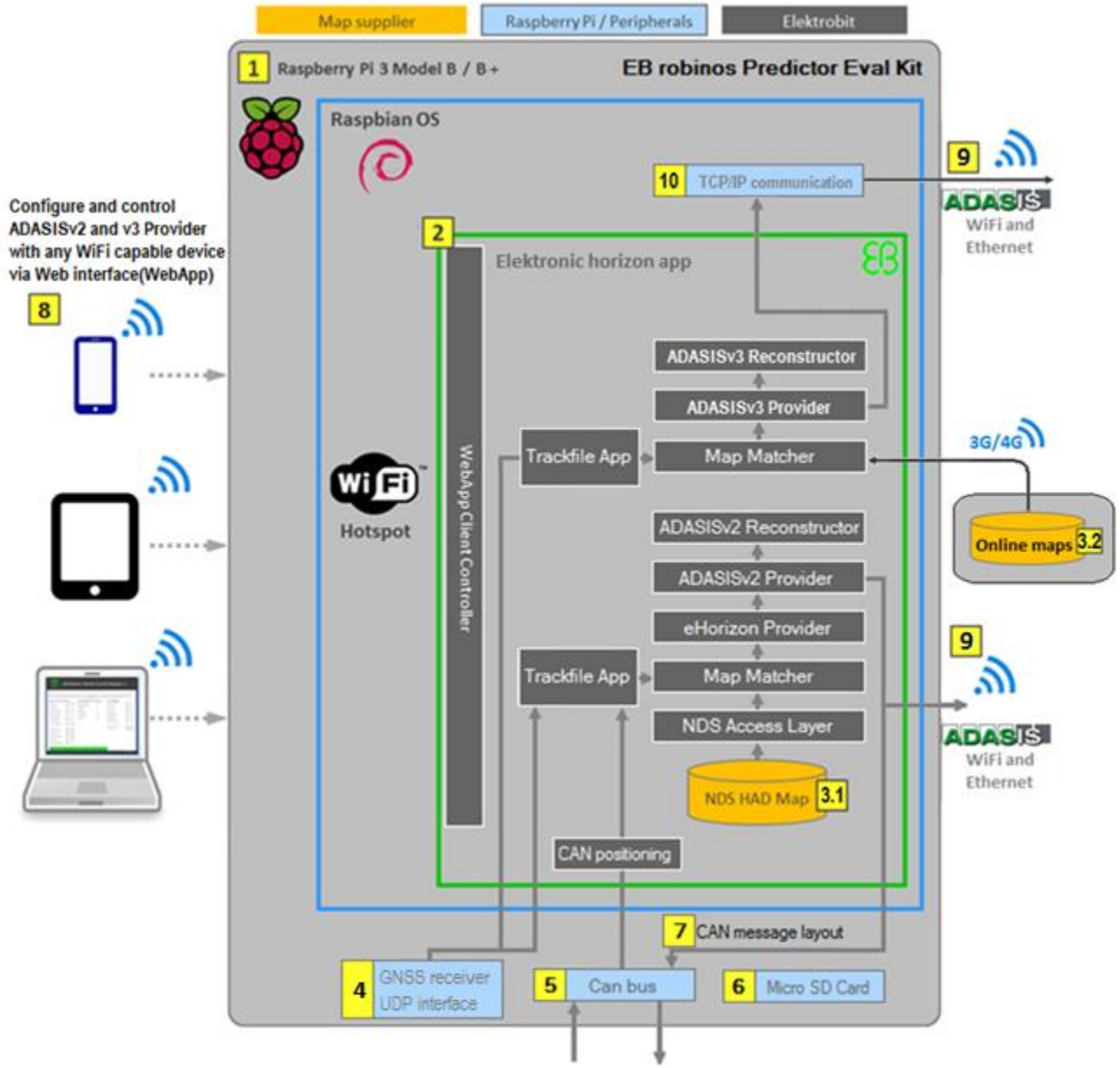


Figure 2: System view of EB robinos Predictor Eval Kit (Raspberry Pi-based Elektrobit Electronic Horizon Provider)

The following chapters describe the system modules as numbered in figure 2.

2.1 Raspberry Pi 3 Model B+ (1)

Elektrobit uses the Raspberry Pi 3 Model B+ which has the following characteristics:

- 1.4 GHz 64-bit quad-core ARMv8 CPU
- 802.11n wireless LAN
- 1 GB RAM

- 4 USB ports
- 40 GPIO pins
- Ethernet port
- Micro SD card slot
- Raspbian Linux (https://www.raspbian.org/)

2.2 Electronic horizon app (ADASISv2 and ADASISv3) (2)

In EB robinos Predictor Eval Kit, Elektrobit integrates its ADASISv2 and ADASISv3 Provider applications which provide up-to-date map information. For more information see also https://www.elektrobit.com/products/automated-driving/eb-robinos/predictor/.

The most relevant software components are the following:

- ADASISv2 Provider, using CAN or UDP interface for data sending
- ADASISv3 Provider, using Wi-Fi or Ethernet interface (TCP/IP) for data sending
- ADASISv2 Reconstructor
- ADASISv3 Reconstructor
- Map Matcher, which uses either the GNSS receiver or trackfile playback data to calculate a map-matched position based on NDS HAD map database (ADASISv2) or online maps (ADASISv3)
- NDS AccessLayer, optimized to run with NDS 2.4.x or NDS 2.5 HAD including a LaneBB v.3.2 map
- WebApp Client Controller, to receive configuration and commands from connected WebApp remote controls and to render the user interface

2.3 ADASISv2 Provider – NDS HAD Map (3.1)

Elektrobit provides a fully compliant NDS access layer core based on the NDS filling specification [1]. For end user convenience, ADASISv2 Provider on EB robinos Predictor Eval Kit comes accompanied by NDS2.5.4 map database from HERE. These maps cover the areas of Europe and North America and can be used for familiarization with electronic horizon behavior. For configuration information refer to chapter 7.1.4.

2.4 ADASISv3 Provider – online maps (3.2)

ADASISv3 Provider on EB robinos Predictor Eval Kit supports the major map suppliers, providing online map data. This data can be accessed over a mobile data connection and conveniently used by vehicle-based applications on the road. Currently EB robinos Predictor Eval Kit uses TomTom Autostream Client Library 6.0.0, HERE HD Live Map (Native) format data (Elektrobit solution), or NDS.Live backend format data to access HD map content and hints from the online server. In addition to the online usage, it is possible to store the online map data cache after the drive and use it in offline mode later (currently available with TomTom and NDS.Live map provider).

The available HD maps contain detailed lane information. Lane borders, lane model, lane groups, lane connections, etc. are used as a basis to form an electronic horizon that is compliant to the ADASISv3 protocol specification. Hints are used to ensure that while the vehicle moves, also relevant map data outside the

vehicle's immediate vicinity gets prefetched and cached in time in order to keep the electronic horizon continuous and consistent throughout the drive.

2.5 Map data limitations

The usage of EB robinos Predictor Eval Kit and the provided electronic horizon is limited in terms of available map data content and geographical coverage. For ADASISv2, the geographical coverage of the pre-installed maps is mentioned in chapter 2.3. There is also an option for the user to add other NDS maps for ADASISv2, see chapter 7.1.4 for details. For ADASISv3, the geographical coverage is dependent on online map suppliers, listed in chapter 2.4. They can be contacted directly for more detailed up-to-date information.

EB robinos Predictor Eval Kit does not support maps that require position shifting. This is currently the situation with China maps. A position-shifting mechanism is not a part of the Eval Kit solution, thus the product will not be able to generate a reliable or any electronic horizon.

2.6 Positioning source (4)

EB robinos Predictor Eval Kit uses a standard GNSS receiver with a USB 2.0 connector to receive NMEA 0183 positioning strings. See chapter 5.2 and 6.2 for further details on how to use ADASISv2 and ADASISv3 Provider in GNSS mode.

A UDP interface can also be used as a positioning source. See chapter 5.3 for instructions for ADASISv2 and chapter 6.2 for ADASISv3.

Every EB robinos Predictor Eval Kit is accompanied with a NAVILOCK NL-8012U GPS USB receiver, which has been preconfigured and tested by the EB robinos Predictor team. Elektrobit cannot promise functionality or give detailed support for other devices. However, if a different GNSS receiver is to be used with EB robinos Predictor Eval Kit, the following is to be noted: The NMEA data sent by the position device needs to contain GPGGA-messages. The data stream can be checked, for example, with Putty (chapter 4.4) with the `cat /dev/ttyACM0` command (assuming the position device is connected to port `ttyACM0`).

For further questions regarding positioning source, contact the Elektrobit support team via www.elektrobit.com/support.

2.7 CAN bus (ADASISv2) (5)

By connecting the Raspberry Pi to a CAN bus (see chapter 3.3 and chapter 11 for more information about recommended CAN setup), it is possible to:

- Send CAN position data to Map Matcher and calculate a map-matched position
- Receive electronic horizon data as ADASISv2 CAN messages from ADASISv2 Provider

2.8 Micro SD card (6)

The delivered micro SD card incorporates the Raspberry Pi's operating system and all Elektrobit software applications. The micro SD card must always be connected to the device and must not be pulled off during runtime.

Warning **Do not disconnect the micro SD card from the device during runtime**

Unexpected disconnection of the micro SD card might cause data loss and problems with the Raspberry Pi file system.

2.9 CAN message layout for receiving data via CAN bus (ADASISv2) (7)

EB robinos Predictor Eval Kit sends ADASISv2 messages according to the official ADASIS protocol specification (Motorola format, Big Endian). The protocol specification can be downloaded free of charge for all ADASIS AISBL members from the respective homepage (http://adasis.org/). Furthermore, a DBC file is available that describes all ADASISv2 CAN messages and signals.

The matching specification [2] for the ADASISv2 protocol sending by EB robinos Predictor Eval Kit is:

- `200v2.0.4-D2.2-ADASIS_v2_Specification.pdf`

The customer has to adapt their CAN environment to this message and signal description in order to make use of the electronic horizon data.

However, the CAN format can also be adjusted to Little Endian (Intel format). Follow the instructions in chapter 7.1.2 to adjust the CAN format properties.

2.10 WebApp user interface (8)

EB robinos Predictor Eval Kit can be configured and controlled via user interface (WebApp). Once the Raspberry Pi is running, a Wi-Fi hotspot is established and any Wi-Fi-capable device (computer, tablet, mobile phone) can connect to the Raspberry Pi. You can access the user interface (WebApp) by using a web browser (Firefox, Google Chrome, and Safari (iOS/MacOS) are fully supported). It is also possible to use the WebApp with Ethernet cable connection. Chapter 4 provides information on how to establish such connections.

2.11 Receiving data via Wi-Fi/Ethernet (9)

When ADASIS Provider is running and receiving valid positions via GNSS receiver, CAN or trackfile, it starts to send electronic horizon data. Depending on the ADASIS version, a different Internet protocol is used.

2.11.1 ADASISv2 – UDP

When ADASISv2 Provider is running and UDP data sending mode is selected, ADASISv2 message data can be received directly from port **3334**. The IP address depends on the connection used as described in chapter 4.1 (Ethernet) or chapter 4.2 (Wi-Fi).

Note that some firewalls are blocking UDP traffic by default. So, in case of any issues with data reception check the firewall settings.

2.11.2 ADASISv3 – TCP/IP

ADASISv3 Provider sends electronic horizon data in msgpack format. See chapter 2.12 for details. This data can be received directly from port **51245**. The IP address depends on the connection used as described in chapter 4.1 (Ethernet) or chapter 4.2 (Wi-Fi).

2.12 ADASISv3 communication scheme (10)

EB robinos Predictor Eval Kit and Elektrobit's ADASISv3 Reconstructor use Franca IDL to generate all ADASISv3 message containers according to the official ADASIS protocol specification. The ADASISv3 messages that are ready to be sent are serialized and deserialized by using msgpack (http://msgpack.org). It is possible to connect any ADASISv3 client application which incorporates Elektrobit's Reconstructor using either Wi-Fi or Ethernet connection. All ADASISv3 messages are transferred using the TCP/IP application layer.

The protocol specification can be downloaded free of charge for all ADASIS AISBL members from the respective homepage (http://adasis.org/).

3 EB robinos Predictor Eval Kit connection instructions

Within this chapter an instruction on how to connect EB robinos Predictor Eval Kit with either a CAN network or a computer client (e.g. EB Assist ADASISv2 Reconstructor toolbox) is provided. Further details on needed hardware to establish a connection between EB robinos Predictor Eval Kit and a computer can be found in chapter 12.

3.1 Power supply

EB robinos Predictor Eval Kit uses a 5 V micro-USB connector as power source.

Note: Some computer USB ports do not provide sufficient power to run EB robinos Predictor Eval Kit without issues. In these cases, unexpected and misleading errors might occur. As a result, Elektrobit strongly recommends using an external power supply (such as a cell phone charger) or USB cables to connect EB robinos Predictor Eval Kit with the USB hubs powered by an external power supply with a minimum of 2 A.

3.2 GNSS receiver connection

In case EB robinos Predictor Eval Kit is configured in GNSS mode (see chapter 7.2 and 8.2), connect the USB receiver in any vacant USB slot of the Raspberry Pi. The application software automatically connects to the GNSS receiver and the electronic horizon provider and streams the NMEA strings.

3.3 CAN interface (ADASISv2 only)

EB robinos Predictor Eval Kit delivered by Elektrobit already provides a RS232 connector that is ready to be used within CAN area networks. Check whether the CAN area network already has a terminator resistor. If so, one can easily leave the delivered terminator resistor away. In case EB robinos Predictor Eval Kit software image is installed on Raspberry Pi by the customer, check the needed hardware components from the list in chapter 11 to enable CAN features.

Via a CAN/USB connector, it is also possible to connect EB robinos Predictor Eval Kit to a computer in order to visualize the electronic horizon tree in an appropriate tooling (e.g. EB Assist ADTF ADASISv2 Reconstructor toolbox). Using EB robinos Predictor Eval Kit in CAN sending mode disables horizon visualization in WebApp.

3.4 Ethernet/Wi-Fi interface

EB robinos Predictor Eval Kit uses the Raspberry Pi's Ethernet interface in order to connect with the local area network. Alternatively, you can use the built-in Wi-Fi access point to connect with the client applications.

3.5 Micro SD card slot

The Raspberry Pi operating system, Elektrobit application software and the ADAS maps are stored on the delivered micro SD card. The software is configured to only run with the delivered Raspberry Pi. If software is not activated, see chapter 10.3 to get valid activation keys. Make sure the SD card is placed properly in the Raspberry Pi and is not locked as trackfiles and configuration data may be written during the runtime.

3.6 Mobile data connection

To access the online map data, Raspberry Pi needs a functioning Internet connection. For convenient use on the road and outside the office, Elektrobit recommends using a 3G/4G/5G USB mobile dongle to establish a reliable Internet connection. In the office, it is also possible to use Ethernet cable to get Internet access. There is an indicating icon visible on WebApp to show the connection status (figure 3).

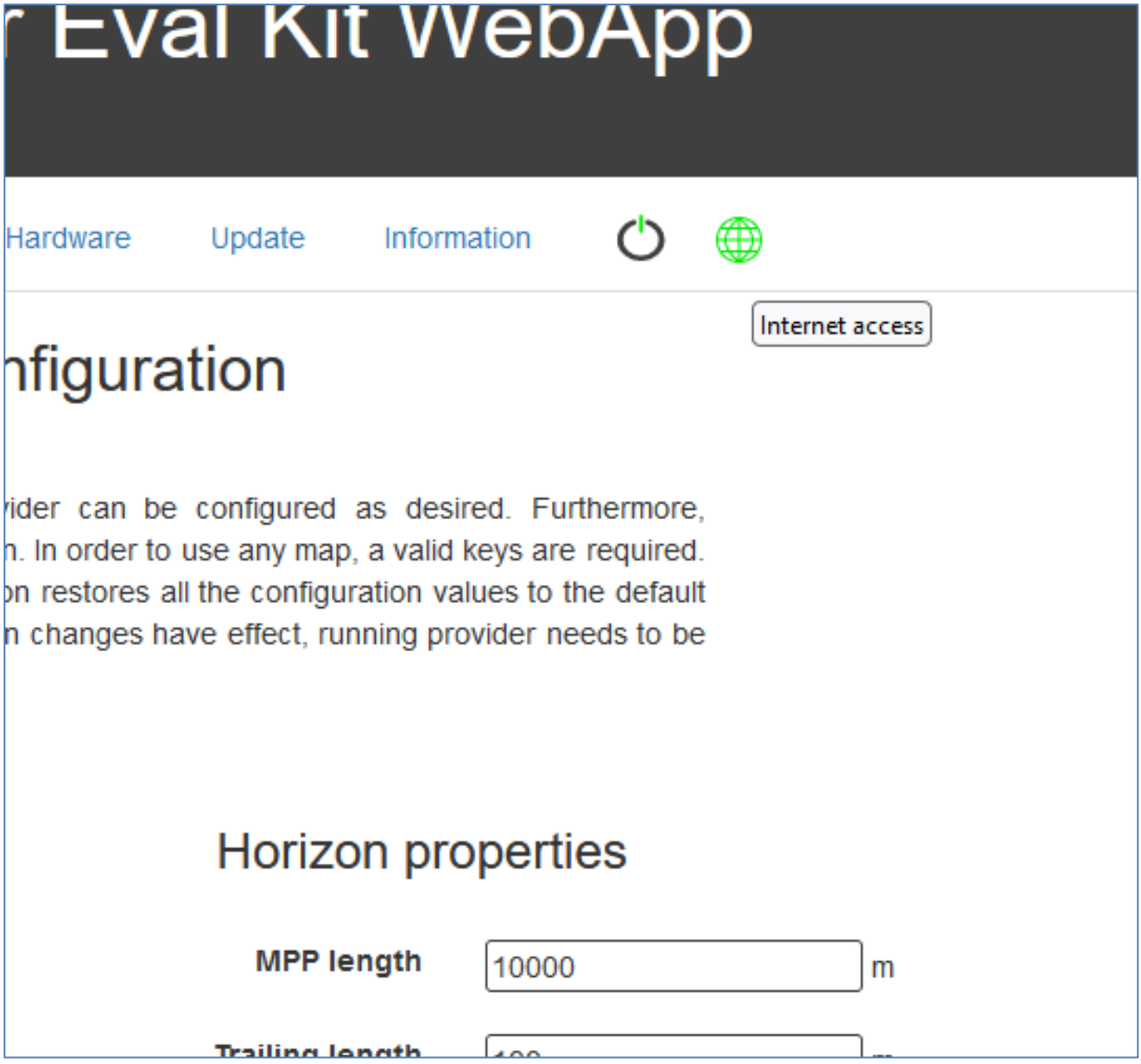


Figure 3: WebApp Internet access icon indicating that Raspberry Pi device is connected to Internet

Figure 4: EB robinos Predictor Eval Kit with peripherals ready for use

4 Connecting to EB robinos Predictor Eval Kit

This chapter provides step by step instructions of how to connect your computer using wired Ethernet or Wi-Fi connection.

4.1 Connecting a computer with a Raspberry Pi using wired Ethernet connection

Every EB robinos Predictor Eval Kit device is configured to hold a static IP address. The DHCP server mode is switched off. Furthermore, Elektrobit recommends using a dedicated Ethernet port (e.g. an USB/Ethernet adapter such as TU2-ET100 or similar) on your computer to establish a connection to the Raspberry Pi:

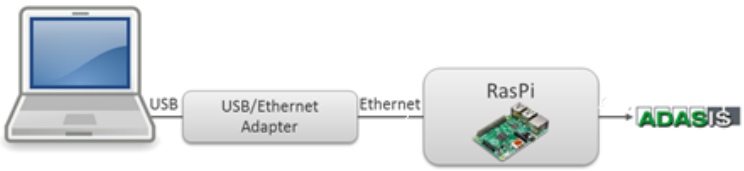


Figure 5: Ethernet connection between computer and Raspberry Pi

Once the hardware connection between the USB/Ethernet adapter and EB robinos Predictor Eval Kit is established and the power supply is connected, you can start the WebApp user interface by using your device's Internet browser and the following IP address/port:

WebApp interface	192.168.0.1:5000

Table 1: IP/port for the WebApp using wired Ethernet connection

4.2 Connecting a computer with a Raspberry Pi using Wi-Fi connection

The WebApp user interface can be accessed via a wireless LAN (Wi-Fi) connection.

Figure 6: Wi-Fi connection between computer and Raspberry Pi

Once the system is booted, EB robinos Predictor Eval Kit opens a visible Wi-Fi access point (DHCP server). Connect your computer to the following EB robinos Predictor Eval Kit's Wi-Fi network and use the following user credentials to establish a connection over Wi-Fi:

Wi-Fi name	EBRaspi
Wi-Fi password	elektrobit
WebApp interface	192.168.42.1:5000

Table 2: User credentials to connect with the Raspberry Pi over Wi-Fi

4.3 Accessing the Raspberry Pi's file system using WinSCP

In some cases, it is necessary to have full file system access. A convenient way to access the Raspberry Pi's file system is to use graphical FTP client software. Elektrobit recommends using the free FTP solution WinSCP (https://winscp.net/eng/index.php). In figure 7, WinSCP login window is presented with host name used for Wi-Fi connection. Save the connection settings in order to easily re-connect to the Raspberry Pi when needed. These user credentials are required for accessing Raspberry Pi's file system:

- User name: **pi**
- Password: **raspberry**

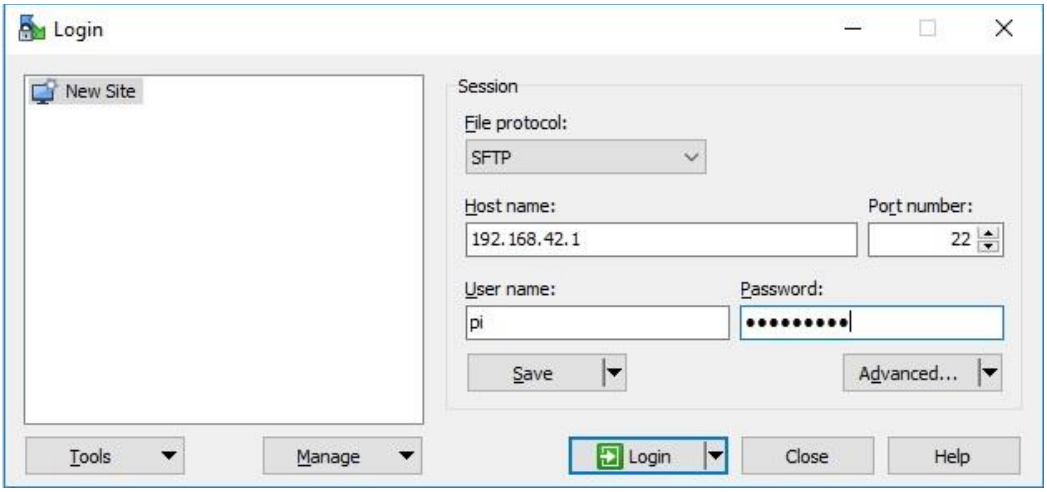


*Figure 7: Use **Login** to add connection information and user credentials in WinSCP*

Once the connection is established (over Ethernet or Wi-Fi), the right-side tab in figure 8 shows the Raspberry Pi's file system (`/home/pi/`).

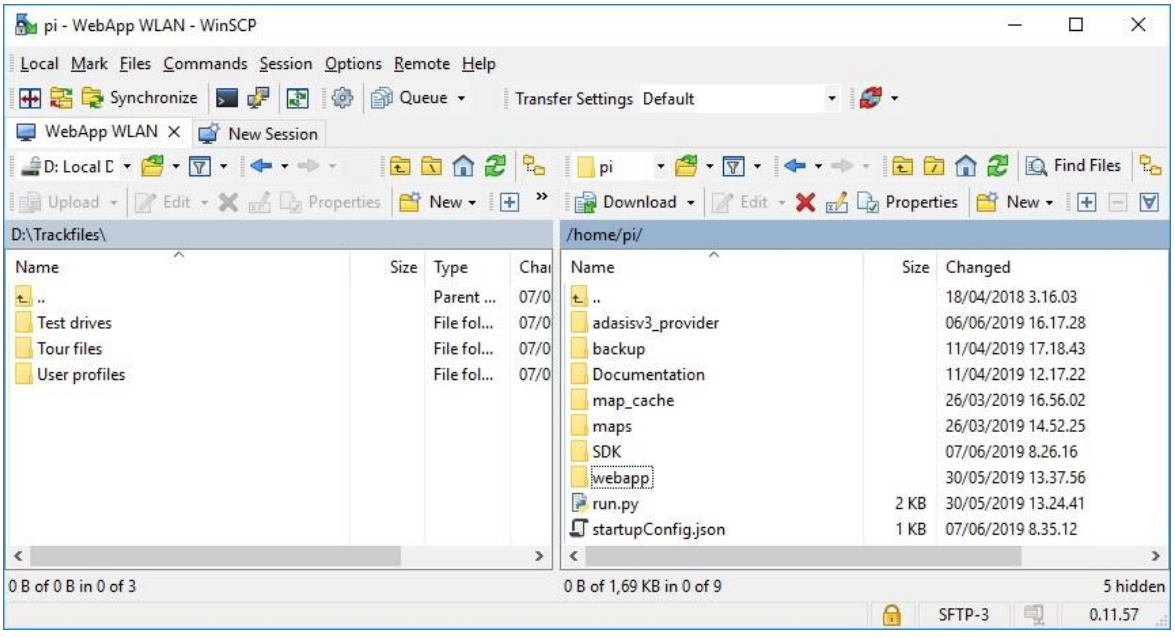


Figure 8: FTP client WinSCP (computer) connected with a Raspberry Pi

4.4 Accessing the Raspberry Pi's file system using PuTTY

PuTTY is a simple SSH client application which provides a powerful command line tooling. Developers and customers who are familiar with using command line scripting will have fast control of the Raspberry Pi.

Once PuTTY is installed (http://www.putty.org/), open it and use the same IP address as described in chapter 4.1 (Ethernet) or 4.2 (Wi-Fi) respectively to connect to the device. Figure 9 shows the PuTTY configuration window.

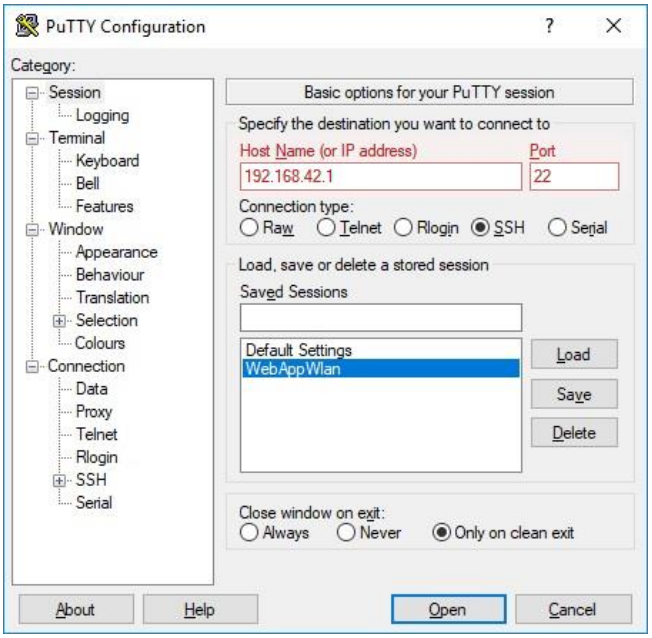


Figure 9: PuTTY Configuration to connect to the Raspberry Pi

After successful connection, the PuTTY command line window is visible, and you can enter these user credentials:

- login as: **pi**
- pi@192.168.42.1's password: **raspberry**

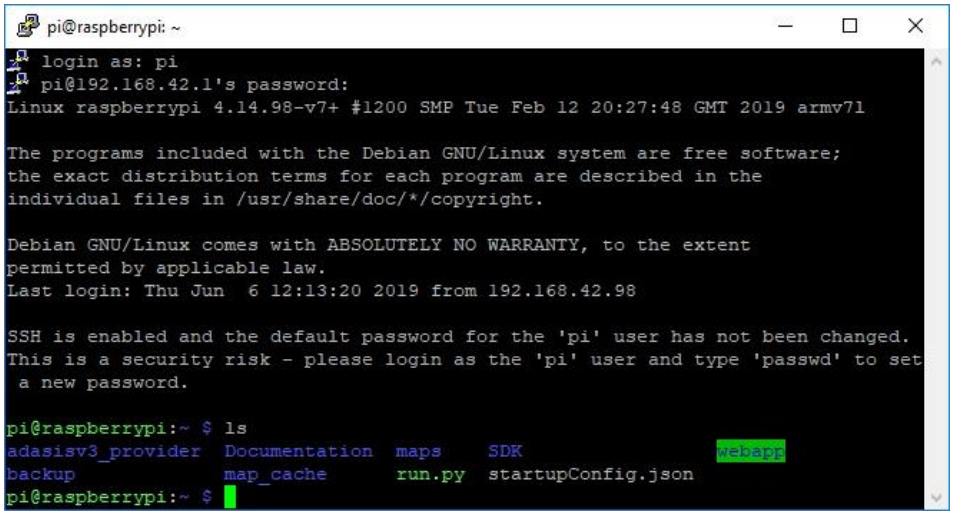


Figure 10: PuTTY's command line window

4.5 Accessing the EB robinos Predictor Eval Kit data folders using Samba

Samba (https://www.samba.org/samba/) network sharing is set up on Raspberry Pi device to enable convenient way to transfer files between Raspberry Pi and Windows device. For ADASISv3 Provider, trackfile and TomTom map cache folders are visible thorough Windows network view (figure 11).

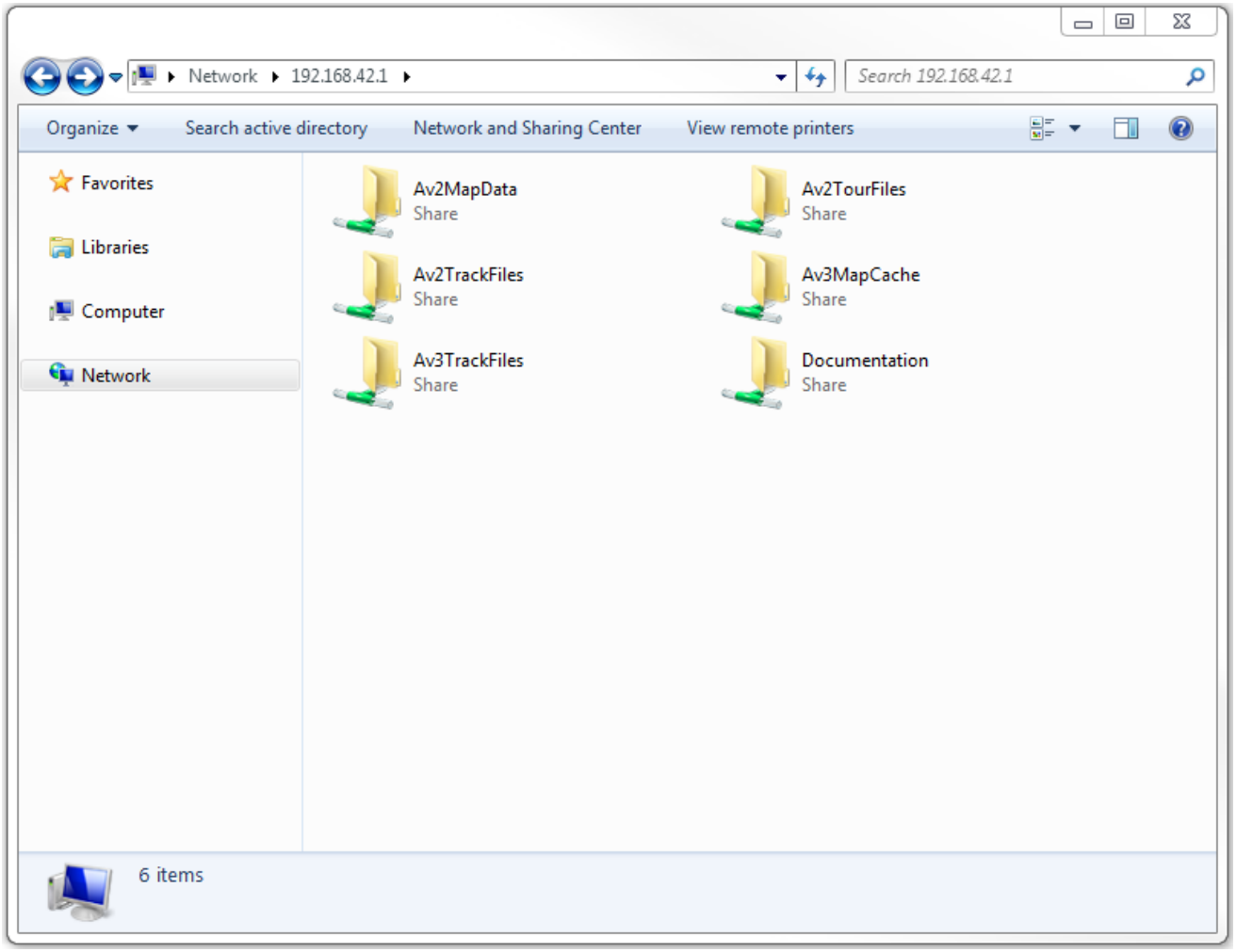


Figure 11: EB robinos Predictor Eval Kit data folders on Raspberry Pi are accessible via Samba network share.

4.6 Receiving ADASIS Provider data

When ADASISv2 or ADASISv3 Provider is running and receives valid positions it starts to send electronic horizon data. See chapter 2.11 for details.

5 Software operational modes (ADASISv2)

In this chapter, different operation modes for ADASISv2 Provider running on EB robinos Predictor Eval Kit are presented. Each of the modes can be used to send data via UDP or CAN. This can be selected via configuration page.

5.1 Playback

For easy verification in customer's environment, EB robinos Predictor Eval Kit can be used in ADASISv2 Provider playback mode. In this mode, ADASISv2 Provider is sending electronic horizon data (ADASISv2 messages) based on the map-matched positions from previously recorded test-drives. After system startup and starting the trackfile playback via WebApp (see chapter 7.2), EB robinos Predictor Eval Kit will send ADASISv2 messages without further configuration. The customer is able to receive ADASISv2 data and interpret the messages according to the format specification as described in chapter 2.9 and 2.11.1.

Note: There is no need for a GNSS receiver or CAN position sending in this mode, thus this mode is established for test purposes within buildings (e.g., labor test environment without or disrupted GNSS reception). Elektrobit will pre-install trackfiles that customers can use for their verification purposes. Trackfiles can also be generated based on the Open Source Routing Machine generated routes (see chapter 9).

5.2 Live GNSS

EB robinos Predictor Eval Kit provides a ADASISv2 Provider live GNSS mode. Within this mode, ADASISv2 Provider will use the connected GNSS receiver to obtain GNSS position data. Once valid GNSS position data is received and the electronic horizon provider is able to match the GNSS position to the available map data, the application will start to build up the electronic horizon and send ADASISv2 messages accordingly.

In case the vehicle moves offroad (= Map Matcher is not able to match the current GNSS position to the map), the electronic horizon provider will not expand any horizon data anymore. Once the vehicle is back on the digitized road, electronic horizon will continue building up and sending valid data.

Note: Depending on the receiver device used and whether the receiver runs through a cold or warm start, it might take several minutes until a usable GNSS fix state is established. Before that no valid ADASISv2 data is sent on CAN.

The received GNSS data can be recorded and saved into a trackfile that can be used later on for verification purposes. It is possible for user to disable/enable the trackfile recording via the **Save trackfile** checkbox in the **Configuration** tab before starting the provider in live GNSS mode. By default, trackfile saving is enabled. The trackfiles are named as `logXXX.trk` (`XXX` presents the increasing numbering) and can be found from the selection list in the WebApp's trackfile playback mode. In the Raspberry Pi, the trackfiles are stored in `/home/pi/adasisv2_provider/trk`.

By default, the GNSS receiver shipped with Eval Kit is configured to send a positions at 1Hz interval. In case you are using a custom configuration with higher position frequency (up to 10Hz has been verified to work), you also need to increase baud rate in the webapp Configuration, see Chapter 7.1.3 for ADASISv2 or Chapter

8.1.2 for ADASISv3. If the baud rate is set too low, messages from the GNSS receiver will get missed which might cause position to stay OFFROAD indefinitely.

5.3 UDP positioning

In ADASISv2 Provider UDP positioning mode the position data is fed via UDP interface to ADASISv2 Provider. Similarly, as with Live GNSS mode, once valid position data is received via UDP interface, the application will start to build up the electronic horizon and send ADASISv2 messages accordingly.

The port used to receive the position data can be configured in the **Configuration** tab. Available port range is **1024–65535**. Note that port **3334** is reserved for the horizon output in UDP sending mode and therefore cannot be set.

The position input is fed to ADASISv2 Provider immediately, i.e., no time simulation is done. Therefore, always provide only one position with one input (although a JSON array is used in the format), and provide the input with a frequency of 1 Hz for optimal map-matching performance. Frequencies up to 10 Hz might still work well. At higher frequencies, the system load might become too high, and no or unreliable output might be generated.

For examples and position data format description with UDP positioning mode, see chapter 7.4.

The received position data can be recorded and saved into a trackfile that can be used later on for verification purposes. It is possible for user to disable/enable the trackfile recording via the **Save trackfile** checkbox in the **Configuration** tab before starting the provider in UDP positioning mode. By default, trackfile saving is enabled.

5.4 CAN positioning

In order to receive position data via CAN bus, EB robinos Predictor Eval Kit can be used in ADASISv2 Provider CAN positioning mode. Within this mode, ADASISv2 Provider will use the connected CAN bus to obtain position data. Similarly, as in GNSS mode, the electronic horizon provider will start to expand the horizon data and send ADASISv2 messages once a valid position (i.e., not offroad) is received and matched to the available map data. Note that CAN positioning works only with UDP data sending mode.

The received position data can be recorded and saved into a trackfile that can be used later on for verification purposes. It is possible for user to disable/enable the trackfile recording via the **Save trackfile** checkbox in the **Configuration** tab before starting the provider in CAN positioning mode. By default, trackfile saving is enabled.

6 Software operational modes (ADASISv3)

In this chapter, different operational modes for ADASISv3 Provider running on EB robinos Predictor Eval Kit are presented. Each of the modes can be used to receive ADASISv3 data via TCP/IP. To use ADASISv3 Provider, a functioning Internet access is needed.

6.1 Playback

For an easy verification in the customer environment, EB robinos Predictor Eval Kit can be used in Position playback mode. In this mode, ADASISv3 Provider sends electronic horizon data (ADASISv3 messages) based on the map-matched positions from the previously recorded test drives. After system startup and starting the nmea or trackfile playback via the WebApp (see chapter 8.2), EB robinos Predictor Eval Kit sends ADASISv3 messages without further configuration. The customer is able to receive ADASISv3 data (see chapter 2.11.2) and interpret the messages according to the format specification described in chapter 2.12.

Note: There is no need for a GNSS receiver in this mode, thus this mode is established for test purposes within buildings (e.g., a labor test environment with disrupted or without any GNSS reception). Elektrobit pre-installs trackfiles that customers can use for their verification purposes. Trackfiles can also be generated based on the Open Source Routing Machine generated routes (see chapter 9).

6.2 UDP Positioning

Within this mode ADASISv3 Provider uses UDP connection to obtain GNSS position data. Once valid GNSS position data is received and the electronic horizon provider is able to match the GNSS position to the available map data, the application starts to build up the electronic horizon and sends ADASISv3 messages accordingly.

In case the vehicle moves offroad, which means that the Map Matcher is not able to match the current GNSS position to the map, the electronic horizon provider does not expand any horizon data anymore. Once the vehicle is back on the digitized road, the electronic horizon provider continues building up and sending valid data.

When using UDP Positioning, the received positioning data can be recorded into .raw/.ts files that can be used later for verification purposes. Recording can be disabled or enabled in **Configuration** tab before starting the UDP Positioning mode. After operation, if recording was enabled, the recorded files can be found under /home/pi/adasisv3_provider/udp on the Raspberry Pi.

6.3 Live GNSS

Within this mode ADASISv3 Provider uses the connected GNSS receiver to obtain GNSS position data. Once valid GNSS position data is received and the electronic horizon provider is able to match the GNSS position to the available map data, the application starts to build up the electronic horizon and sends ADASISv3 messages accordingly.

In case the vehicle moves offroad, which means that the Map Matcher is not able to match the current GNSS position to the map, the electronic horizon provider does not expand any horizon data anymore. Once the vehicle is back on the digitized road, the electronic horizon provider continues building up and sending valid data.

Note: Depending on the GNSS receiver used and whether the receiver runs through a cold or warm start, it might take several minutes until a usable GNSS fix state is established. Until then no valid ADASISv3 data is sent.

When using GNSS receiver, the received GNSS data can be recorded and saved into nmea or trackfile that can be used later for verification purposes. The user can disable or enable the nmea or trackfile recording in **Configuration** tab before starting the Live GNSS mode. By default, nmea and trackfile saving are enabled. The trackfiles are named as `track_mm-dd-yyyy_hh-mm-ss.trk` (mm, dd, yyyy, hh, mm, ss present the month, date, year, hour, minutes, and seconds respectively) and can be found from the selection list in the WebApp's trackfile playback mode. In the Raspberry Pi, the trackfiles are stored in `/home/pi/adasisv3_provider/trk`.

Alternatively, nmea files which are raw GNSS data are named as `track_mm-dd-yyyy_hh-mm-ss.nmea` and `track_mm-dd-yyyy_hh-mm-ss.nmeats`. The nmea files can also be found from the selection list in playback mode and are stored in `/home/pi/adasisv3_provider/nmea`. The nmeats files are stored with an additional timestamp information for replicating the timestamp of the GNSS.

7 Using Elektrobit's Electronic Horizon Provider (ADASISv2)

The most convenient way to configure and control EB robinos Predictor Eval Kit is use the WebApp user interface. Connect your device (computer, cell phone, tablet device) by using Ethernet or Wi-Fi credentials (see chapter 4) and type the respective IP address and port number into your web browser. After this WebApp user interface (figure 12) is visible for the user. There are four operating modes as described in chapter 5.

In case you have been using the combined ADASISv2 and ADASISv3 EB robinos Predictor Eval Kit in ADASISv3 mode, change to ADASISv2 mode via **Change ADASIS version** button in the **Configuration** tab.

7.1 Configuring ADASISv2 Provider

EB robinos Predictor Eval Kit provides a possibility to adjust Elektrobit's ADASISv2 Provider configuration parameters. These are stored in `AdasisProviderV2.json` and `ATFHorizonProvider.conf` files. Configurations are also adjustable for LearningMPP component, for which they are stored in `LearningMPPConfig.json`. The files are in the `adasisv2_provider` directory of the Raspberry Pi (`/home/pi/adasisv2_provider/`). There are two possibilities to adjust these parameters: Use the WebApp user interface (recommended) or access the Raspberry Pi's file system and change the parameters manually. Detailed information about configurable parameters can be found in chapter 7.1.1.

In addition to the ADASISv2 Provider configuration, sending ADASISv2 data can also be configured either via UDP (Wi-Fi/Ethernet) or CAN. Once CAN sending mode is selected, CAN IDs for each message type can also be configured (see chapter 7.1.2).

The user can set preferences for the automatic Live GNSS mode on device startup and decide if saving the trackfile is enabled or disabled. These settings, together with CAN/UDP sending mode, are stored in `startupConfig.json` file in Raspberry Pi's directory (`/home/pi/`).

In the **Configuration** tab of WebApp all relevant configuration parameters are adjustable within their value ranges. Once the desired values are adjusted, save the current settings by pressing the **Save** button. If ADASISv2 Provider is running, it needs to be stopped and started again so that the changes become effective.

To perform a safe shutdown for the Raspberry Pi device, press the power off icon in the rightmost tab of the navigation menu, select **Power Off**, and wait for **EBRaspi** Wi-Fi network to disappear before unplugging the power supply. The device can be rebooted by selecting **Reboot**.

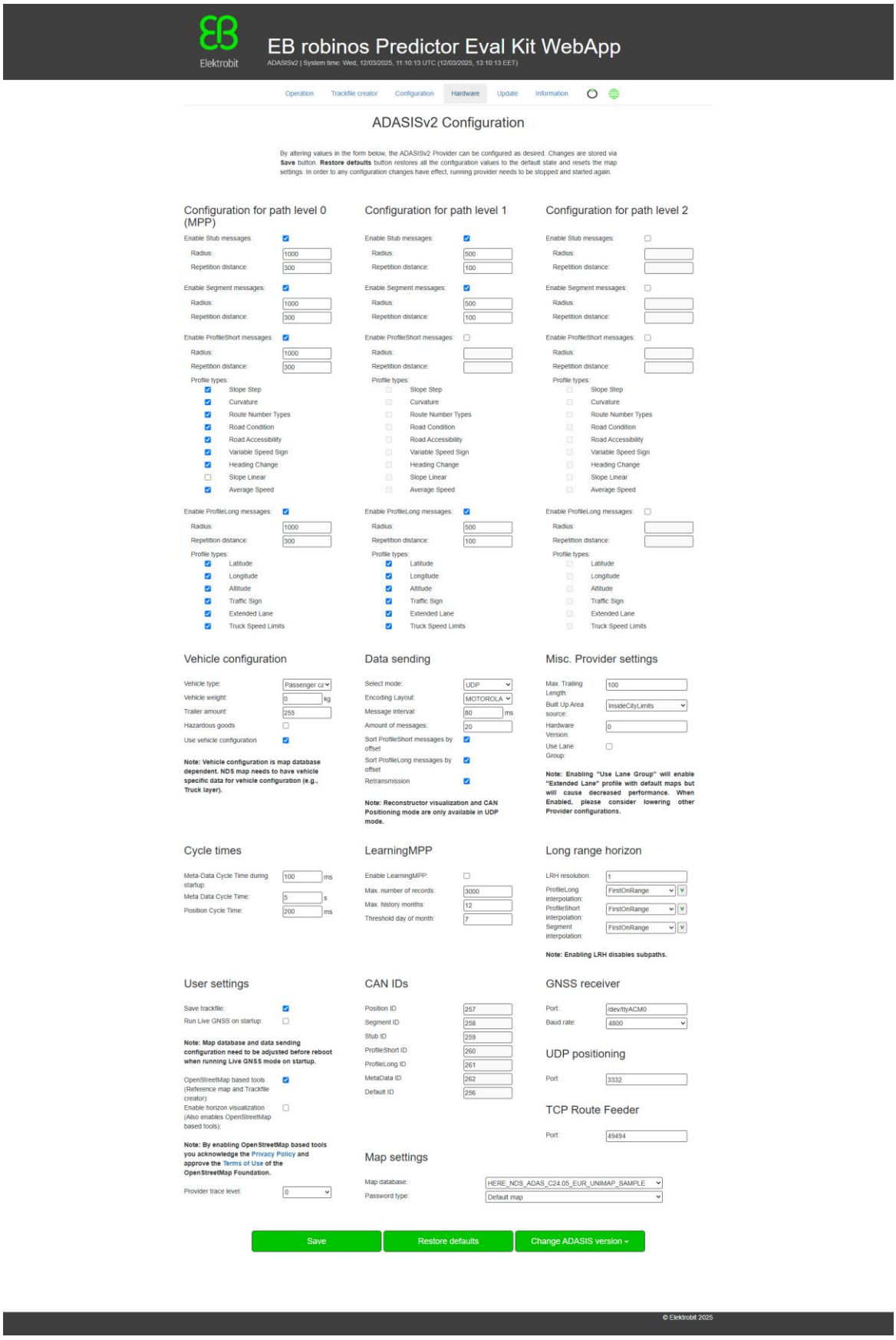


Figure 12: WebApp showing EB robinos Predictor Eval Kit Configuration tab

7.1.1 Configurable ADASISv2 Provider parameters

The following table provides all ADASISv2 Provider parameters you can configure using the WebApp user interface. Click **Restore defaults** in the **Configuration** tab to restore the pre-defined default values for each path level and parameter.

Note: If messages on path level 1 or 2 are enabled, Stub messages on previous path levels must be enabled. For example, if path level 2 Segment messages are enabled, path level 0 and 1 Stub messages must be enabled.

No.	Attribute	Value range	Description	Default value
Configuration for path levels 0 (MPP) to 2				
1	Stub message radius	0–8000 [m]		Level 0: 1000 m Level 1: 300 m Level 2: DISABLED
2	Stub message repetition distance	0–8190 [m]	In case there is no attribute change to transmit, the horizon provider sends intermediate messages after a certain repetition distance to enhance the reliability for the ADAS client.	Level 0: 300 m Level 1: 0 m Level 2: DISABLED
3	Segment message radius	0–8000 [m]		Level 0: 1000 m Level 1: 500 m Level 2: DISABLED
4	Segment message repetition distance	0–8190 [m]	In case there is no attribute change to transmit, the horizon provider sends intermediate messages after a certain repetition distance to enhance the reliability for the ADAS client.	Level 0: 300 m Level 1: 100 m Level 2: DISABLED
5	ProfileShort message radius	0–8000 [m]		Level 0: 1000 m Level 1: DISABLED Level 2: DISABLED

6	ProfileShort message repetition distance	0–8190 [m]	In case there is no attribute change to transmit, the horizon provider sends intermediate messages after a certain repetition distance to enhance the reliability for the ADAS client.	Level 0: 300 m Level 1: DISABLED Level 2: DISABLED
7	ProfileShort types	• Slope Step • Curvature • Route Number Types • Road Condition • Road Accessibility • Variable Speed Sign • Heading Change • Slope Linear	Selection of transmitted ProfileShort types.	Level 0: ALL EXCEPT 'Slope Linear' ENABLED Level 1: ALL DISABLED Level 2: ALL DISABLED
8	ProfileLong message radius	0–8000 [m]		Level 0: 1000 m Level 1: 500 m Level 2: DISABLED
9	ProfileLong message repetition distance	0–8190 [m]	In case there is no attribute change to transmit, the horizon provider sends intermediate messages after a certain repetition distance to enhance the reliability for the ADAS client.	Level 0: 300 m Level 1: 100 m Level 2: DISABLED
10	ProfileLong types	• Latitude • Longitude • Altitude • Traffic Sign • Extended Lane • Truck Speed Limits	Selection of transmitted ProfileLong types.	Level 0: ALL ENABLED Level 1: ALL ENABLED Level 2: ALL DISABLED
Vehicle configuration				
Note that the vehicle configuration is map database-dependent. NDS map needs to have vehicle-specific data for vehicle configuration (e.g., Truck layer).				
11	Vehicle type	• Passenger car • Truck • Bus	Used to configure which vehicle type is to be set.	Passenger car

12	Vehicle weight	0-MAXUINT32 [kg]	Defines the weight of the vehicle in kg. 0 means unknown or not applicable.	0
13	Trailer amount	0-255	Defines the number of trailers used in the vehicle. 255 means the number of trailers is unknown.	255
14	Hazardous goods	True/False	Defines whether the vehicle contains hazardous material.	False
15	Use vehicle configuration	True/False	MPP and subpath generation takes the vehicle configuration into account, i.e., paths are not extended if the vehicle configuration matches the restrictions on the link. Also used for enabling/disabling the usage of configured vehicle type and weight for lane-specific restrictions and in attribute sending of PassageForbidden, RoadSign, and SpeedLimit.	True
Cycle times				
16	Meta-Data Cycle Time during startup	0–20000 [ms]	Rapid initialization of meta data. During system start, MetaDataMessages are sent cyclically.	100
17	Meta Data Cycle Time	0–20 [s]	Defines period time of meta data messages sent in seconds. A value of 0 disables the sending of meta data messages.	5
18	Position Cycle Time	0–2000 [ms]	Defines period time of position messages sent in milliseconds. A	200

			value of 0 disables the sending of position messages.	
Misc. Provider settings				
19	Built Up Area source	• InsideCityLimits • Urban • InsideCityLimitsThenUrban	Source of Built-Up Area (BUA) segment data message. BUA can be derived from Urban or InsideCityLimits attribute of NDS map used. InsideCityLimitsThenUrban first evaluates InsideCityLimits attribute and when it is unknown, then the Urban attribute will be evaluated.	InsideCityLimits
20	Maximum trailing length	10–500 [m]	Defines the trailing length (electronic horizon data behind the current vehicle position).	100 m
21	Use Lane Group	True/False	Depending on the configuration, the lane-specific information (Profile Long message Extended Lane, Segment/Stub message Number of lanes) is fetched either from Guidance/Routing layers (false) or from LaneGroup layer (true). Depending on the NDS map used the lane-specific information can be in different layers. Note: Setting the Use Lane Group option to true will cause decreased performance and thus other Provider configurations should be set to lower values.	False
22	Hardware Version	0–511	This field will be filled into the hardwareVersion field of the MetaDataMessage as is.	0

Learning MPP				
1	Enable LearningMPP	True/False	Choose whether the LearningMPP component is active.	False
2	Max. number of records	0-MAXUINT32	Maximum number of records to store in the learningMPP database.	3000
3	Max. history months	0-MAXUINT16	Data older than the specified number of months will be deleted from the LearningMPP database.	30
4	Threshold day of month	0-MAXUINT16	LearningMPP data from the current month is only used if the current day of the month is >= threshold.	7
Long range Horizon				
1	LRH resolution	0-255	Defines the offset resolution in meters. I.e. When resolution is 64, then data within 64 meter blocks are combined under one offset value.	1 (LRH disabled)
2	ProfileLong interpolation	• FirstOnRange • CenterMostOn-Range • LastOnRange • Average	Sets default interpolation strategy for ProfileLong messages when Long range horizon is enabled.	FirstOnRange
3	ProfileShort interpolation	• FirstOnRange • CenterMostOn-Range • LastOnRange • Average	Sets default interpolation strategy for ProfileShort messages when Long range horizon is enabled.	FirstOnRange
4	Segment interpolation	• FirstOnRange • CenterMostOn-Range • LastOnRange • Average	Sets default interpolation strategy for Segment messages when Long range horizon is enabled.	FirstOnRange
5	Expandable profile/field specific interpolation	• Use default • FirstOnRange • CenterMostOn-Range	“Use default” uses the default interpolation for the corresponding profile. For example if ProfileLong interpolation is set to	Use default

	settings (located under each default strategy selection)	• LastOnRange • Average	"FirstOnRange" and Latitude is set to "Use default", FirstOnRange will be used on it. By setting the configuration to something else, the specified strategy will be used for that profile instead of the default.	

Table 3: Configurable ADASISv2 parameters

7.1.2 Configuration for data sending

The following table provides all parameters related to data sending that are configurable by using the WebApp user interface. Click **Restore defaults** in the **Configuration** tab to restore the pre-defined default values for each parameter. For CAN sending, it is also possible to configure all messages to use the same CAN ID.

No.	Attribute	Value range	Description	Default value
1	Data sending mode	• UDP • CAN	Selection of ADASISv2 message data sending protocol.	UDP
2	Encoding layout	• MOTOROLA (Big Endian) • INTEL (Little Endian)	The endianness of the messages is adjustable: The default message layout (bit order) is according to MOTOROLA format and thus compatible to standard ADASIS forum DBC file.	MOTOROLA
3	Message interval	20–160 [ms]	Cycle time between sent out ADASISv2 messages.	80
4	Amount of messages	1–20	Amount of ADASISv2 messages sent in one cycle.	20
5	Sort ProfileShort messages by offset	True/False	Defines whether the sending order of ProfileShort messages is sorted by offset (True) or by profile type (False).	True
6	Sort ProfileLong messages by offset	True/False	Defines whether the sending order of ProfileLong messages is sorted by offset (True) or by profile type (False).	True

7	Retransmission	True/False	When Retransmission is checked, ADASISv2 Provider sends previously sent messages again with Retransmission flag on if there are no new messages available.	True
CAN IDs (configurable if CAN mode is selected)				
8	Position ID	0–2047	ADASISv2 Position messages are sent with this CAN identifier.	257
9	Segment ID	0–2047	ADASISv2 Segment messages are sent with this CAN identifier.	258
10	Stub ID	0–2047	ADASISv2 Stub messages are sent with this CAN identifier.	259
11	ProfileShort ID	0–2047	ADASISv2 ProfileShort messages are sent with this CAN identifier.	260
12	ProfileLong ID	0–2047	ADASISv2 ProfileLong messages are sent with this CAN identifier.	261
13	MetaData ID	0–2047	ADASISv2 MetaData messages are sent with this CAN identifier.	262
14	Default ID	0–2047	Default CAN identifier.	256

Table 4: Configuration for data sending

7.1.3 Positioning interface configurations

The following table provides all GNSS receiver, UDP positioning and TCP route feeding related parameters that are configurable by using the WebApp user interface. Click **Restore defaults** in the **Configuration** tab to restore the pre-defined default values for each parameter. Default values are set for the NAVILOCK NL-8012U GPS USB receiver and may differ with other devices.

No.	Attribute	Default value	Description
1	Port	`/dev/ttyACM0`	Port to which GNSS receiver is attached. Devices are listed in /dev folder in Raspberry Pi.
2	Baud rate	4800	The speed at which the data is transferred from the GPS receiver to the Raspberry Pi. Different receivers use different speeds.
3	UDP port	3332	Port where positioning data is received when using UDP positioning mode.
4	TCP port	49494	Port for sending route data over TCP connection.

Table 5: Configurable Positioning interface parameters

7.1.4 Configuration for map settings

The following table provides all ADASISv2 Map database settings you can configure using the WebApp user interface. Click **Restore defaults** in the **Configuration** tab to restore the pre-defined default configuration.

No.	Attribute	Value range	Description	Default value
1	Map database	Default maps: • HERE_NDS_ADAS_C23.07_EUR_UNIMAP_SAMPLE (Europe) • HERE_NDS_ADAS_C23.07_NAM_UNIMAP_SAMPLE (North America)	List of map databases stored in `/home/pi/maps/` (Av2MapData in Samba share). User can add maps to this folder and if `ROOT.NDS` is found, the map database will be listed here. NDS versions supported: • 2.4.3_ADE_3.2 • 2.4.4 • 2.5.1 • 2.5.2 • 2.5.3 • 2.5.4 Encryption types supported: • Plain text password (1 or 2 string) • Keystore (`.NKS` file + password) Compression methods supported: • zlib • zstd Disclaimer: Support by Elektrobit for user-added maps is limited.	HERE_NDS_ADAS_C23.07_EUR_UNIMAP_SAMPLE
2	Password type	• Default map • Plain text password • Keystore	If Default map is selected, the hardcoded password for default maps is used. If Plain text password is selected, the fields for setting passwords in string format appear. If Keystore is selected, the fields for selecting KeyStore file and setting keystore password appear	Default map
3	Plain text password 1	string	Password for user added maps that are encrypted with plain text	-

			password. If the map is encrypted with one string password, this field should only be filled. If the map is encrypted with two string passwords, this field should contain the first password.	
4	Plain text password 2	string	Password for user added maps that are encrypted with plain text password. If the map is encrypted with one string password, this field should be empty. If the map is encrypted with two string passwords, this field should contain the second password.	-
5	Keystore	-	If the user added map uses KeyStore encryption, the KeyStore file (`*.NKS`) is to be stored in `/home/pi/keystore/` (`Av2KeyStoreFiles` in Samba share) and will be listed here.	-
6	Keystore password	string	Password for KeyStore selected above	-

Table 6: Configurable map database settings parameters

7.1.5 Configuration for user settings

EB robinos Predictor Eval Kit will start directly on Live GNSS mode if **Run Live GNSS on startup** is selected. All saved settings will be used on device startup. Remember to set the desired map database and data sending mode and enable/disable trackfile saving before using automatic Live GNSS mode. Saved trackfiles can be found in `/home/pi/adasisv2_provider/trk/`.

The following table provides descriptions for all user setting parameters that are configurable by using the WebApp user interface. Click **Restore defaults** in the **Configuration** tab to restore the pre-defined default values for each parameter.

No.	Attribute	Default value	Description
1	Save trackfile	`Enabled`	Selects whether or not trackfile will be saved under `/home/pi/adasisv2_provider/trk/` when ADASISv2 Provider is used in UDP/GNSS positioning modes.

2	Run Live GNSS on startup	`Disabled`	When enabled, ADASISv2 Provider will be started in Live GNSS mode after booting.
3	OpenStreetMap based tools	`Enabled`	Enables OpenStreetMap based tools: Trackfile creator and reference map on **Operation** tab.
4	Enable Horizon visualization	`Enabled`	Enables or disables dynamic horizon data visualization on **Operation** tab.
5	Provider traces enabled	`Disabled`	Select if provider traces are enabled. Should be kept off to prevent SD card from filling up but when there is e.g. needs to share traces with Elektrobit, they can be enabled.

Table 7: Configurable user setting parameters

7.1.6 Configuration for hardware settings

ADASISv2-related hardware settings can be changed via the **Hardware** tab.

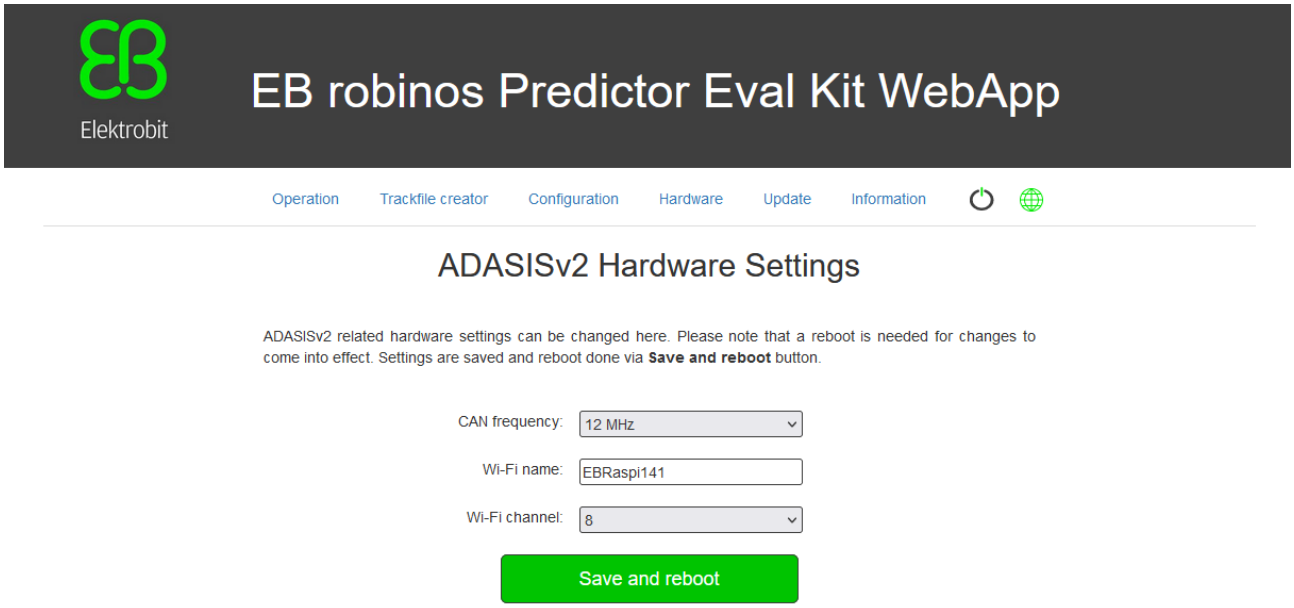
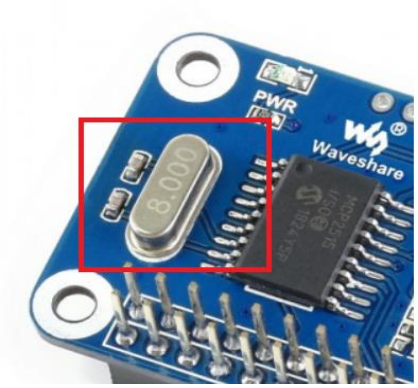


Figure 13: WebApp showing EB robinos Predictor Eval Kit Hardware tab

Note that a reboot is needed for changes to become effective. Settings are saved and a reboot is started via the **Save and reboot** button.

The following table provides all ADASISv2 hardware settings:

No.	Attribute	Value range	Description	Default value
1	CAN frequency	• 8 MHz • 12 MHz	Eval Kit uses CAN board with 8 MHz or 12 MHz oscillator. Oscillator frequency is preset when Eval Kit is assembled but it might need to be reset after update. Eval Kits that use 12 MHz oscillators have a respective label on the bottom of the case. The oscillator used can be checked also from the CAN board:  *Figure 14: 8 MHz oscillator*	Preset when Eval Kit is assembled.
N/A	CAN baud rate		Currently no configuration is available for CAN baud rate in WebApp. If CAN baud rate needs to be modified, it can be done via editing the file `/etc/network/interfaces`. Find and replace the value in the following string: `bitrate 500000`	500 kb/s.
2	Wi-Fi name	Text string that can contain letters, numbers and _ characters. Maximum length is 32 characters.	Wi-Fi name that is used when computer is connected to the EB robinos Predictor Eval Kit's Wi-Fi network. See chapter 4.2 for more details.	Preset when Eval Kit is assembled.

3	Wi-Fi channel	1-11	Wi-Fi channel that is used when computer is connected to the EB robinos Predictor Eval Kit's Wi-Fi network. If several Eval Kits are used close to each other, different Wi-Fi channels should be used. Recommended channels to use are 1, 6, and 11. On these channels the frequencies used do not overlap.	Preset when Eval Kit is assembled.

Table 8: Configurable hardware settings parameters

7.2 Operating ADASISv2 Provider

To operate the ADASISv2 Provider, select the **Operation** tab on the EB robinos Predictor Eval Kit WebApp. The content of the **Operation** tab is dependent on whether or not horizon visualization is enabled in **Configuration** tab chapter 7.1.5. If visualization is enabled, see chapter 7.2.1 for detailed instructions. Chapter 7.2.2 details usage when visualization is disabled.

7.2.1 ADASISv2 Operation (visualization enabled)

When "Enable horizon visualization" parameter is enabled on the **Configuration** tab, the **Operation** page consists of a reference map which visualizes the generated ADASISv2 horizon and optionally selected trackfile and/or current input position, and a 3D widget which visualizes the data contained in the ADASISv2 Profile Long Extended Lane messages. For further details about data presented in the 3D view, click the highlighted button (figure 15). Note: 3D view currently expects traffic to be right-handed and thus using it with left-handed traffic might cause unexpected overlapping of paths.

With heavy ADASISv2 Provider configuration, there might be performance issues with the horizon visualization. Functionality has been verified with default configuration. Please also note that while the horizon visualization can work in offline mode, in order for the reference map to be able to load map data, the device used to operate WebApp also needs to be connected to internet.

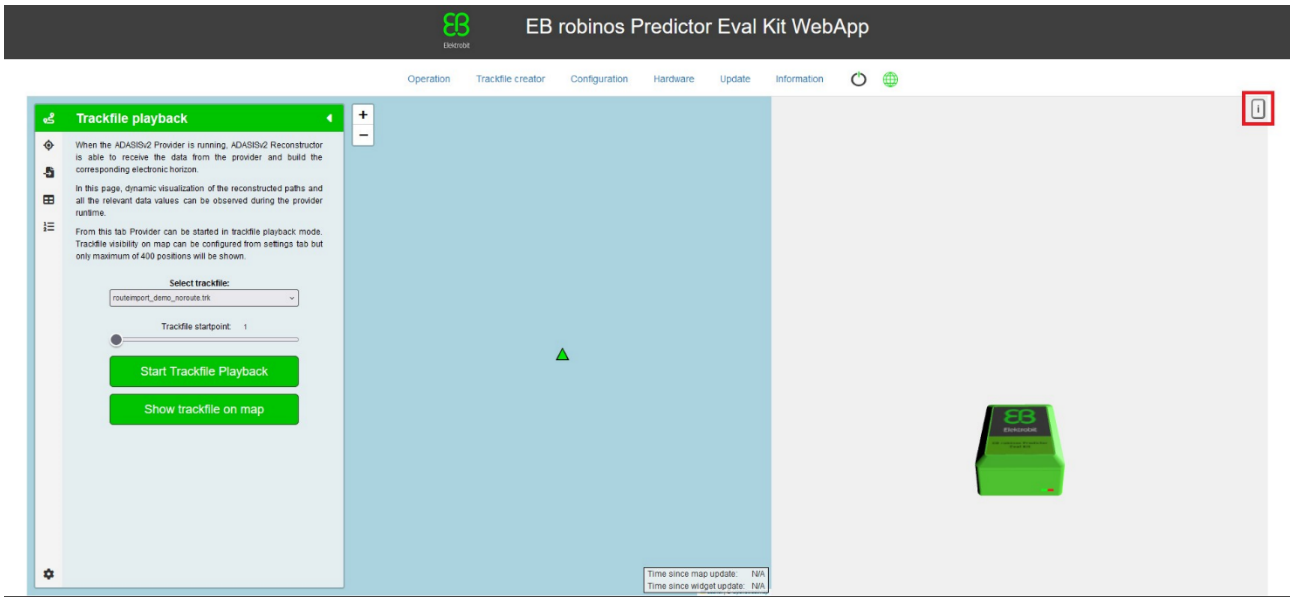


Figure 15: ADASISv2 operation tab with visualizations enabled

7.2.1.1 Trackfile Playback

To start ADASISv2 Provider in trackfile playback mode, select the first icon from the sidebar located on the left side of the reference map (Figure 15). From there, trackfile can be selected from the dropdown menu and the row of trackfile from which playback should start, can be set with the slider located below the dropdown. To start playback, click **Start Trackfile Playback.** To preview trackfile on the reference map before starting playback, click **Show trackfile on map**. During playback, trackfile visibility and current input position visibility can be toggled on the settings panel (Figure 16). From there, other settings related to horizon visualization can also be adjusted.

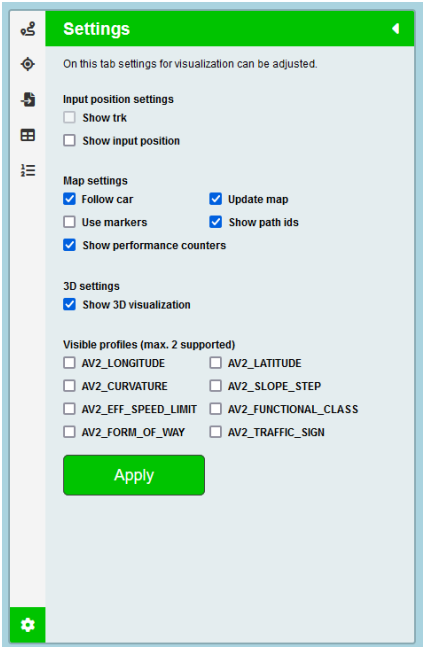


Figure 16: ADASISv2 visualization settings

7.2.1.2 Live positioning

To start ADASISv2 Provider in UDP, GNSS or CAN positioning mode, select the second icon from the sidebar located on the left side of the reference map (Figure 17). On this panel, the desired positioning mode can be selected from the dropdown and operation can be started by clicking **Start recording.** To stop operation, click **Stop recording** from the same panel.

Further details about the available live positioning modes can be found from chapter 5.2 (GNSS), chapter 5.3 (UDP), and chapter 5.4 (CAN).

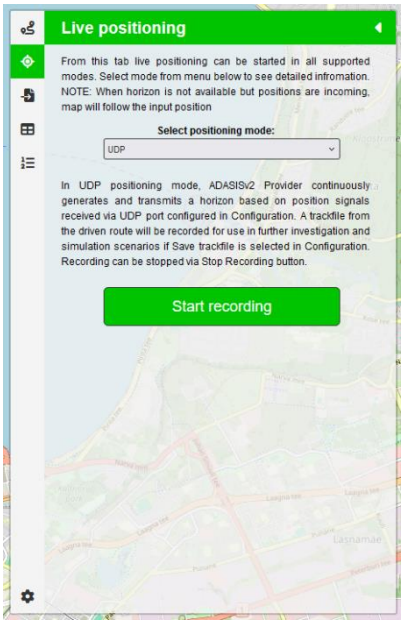


Figure 17: ADASISv2 Live Positioning panel

7.2.2 ADASISv2 Operation (visualization disabled)

When **Enable horizon visualization** parameter is disabled on the **Configuration** tab, the **Operation** page consists of controls for starting operation and a position input reference map (if **OpenStreetMap based tools** is enabled on the **Configuration tab**)**.** When reference map is enabled, the button located under zoom controls can be used to toggle current input positioning following on or off. Operation between trackfile playback and Live positioning can be chosen from **Positioning mode** selector located on top of the page.

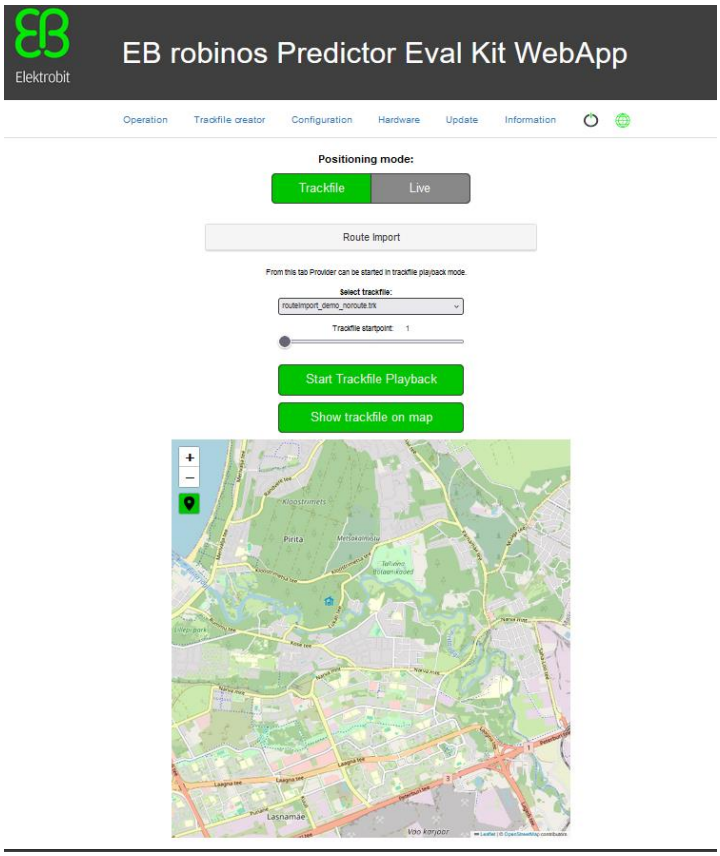


Figure 18: ADASISv2 Operation page without horizon visualization

7.2.2.1 Trackfile Playback

When Trackfile **Positioning mode** is selected on the operation tab (Figure 18), trackfile can be selected from the dropdown and slider located below it can be used to set row of trackfile where playback should start. To start playback, click **Start trackfile playback**. To preview trackfile on the reference map (if enabled) before starting playback, click **Show trackfile on map**.

7.2.2.2 Live Positioning

When Live **Positioning mode** is selected on the operation tab (Figure 19), the structure of page is otherwise the same as in trackfile playback mode but now the dropdown can be used to select between UDP, GNSS and CAN positioning modes. To start operation, click **Start recording.** Further details about the available live positioning modes can be found from chapter 5.2 (GNSS), chapter 5.3 (UDP), and chapter 5.4 (CAN).

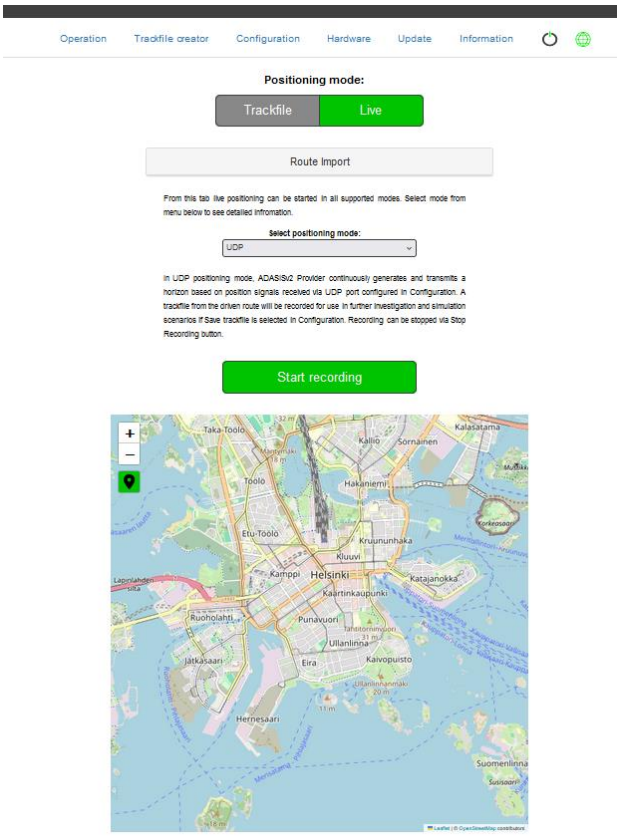


Figure 19: ADASISv2 Operation page without horizon visualization in Live positioning mode

7.3 Route Import

When using EB robinos Eval Kit in ADASISv2 mode, Route Import functionality can be used to control where MPP is generated. This can be used in conjunction with e.g. navigation systems to generate horizon along the calculated route. On Eval Kit, there are three main ways how Route Import can be used. The route data can be fed to Eval Kit over TCP connection in all operation modes, it can be fed through the webapp GUI in form of json files, or it can be contained within a trackfile when using Playback mode.

When route is imported and MPP follows the imported route, the ADASISv2 segment messages corresponding to the imported part have their “Part of Calculated Route” flag set to 1. When horizon visualizations are enabled on the Operation tab, the part of MPP corresponding to imported route is drawn in red instead of blue based on information provided by this flag.

This chapter is structured in the following way: chapter 7.3.1 describes the process for generating route data, chapter 7.3.2 describes how the data can be used through Webapp GUI, chapter 7.3.3 describes how the data can be used with TCP connection, and finally chapter 7.3.4 describes how the route data can be embedded into trackfiles.

7.3.1 Route data format

4 encoding formats are supported for route data:

- 0: OpenLr (https://download.tomtom.com/open/banners/openlr-whitepaper_v1.5.pdf)
- 1: Eblr (Elektrobit proprietary format, schema available in /home/pi/Documentation/ADASISv2)
- 2: Eblr-gz (GZipped Eblr)
- 3: Encoded polyline (precision 6, https://developers.google.com/maps/documentation/utilities/polylinealgorithm)

Each of the data formats can be generated with included python script called openlr_converter.py which is located under /home/pi/adasisv2_provider/tools (see chapter 4.3 for how to access the Eval Kit file system). This script is meant as a starting point to generate some valid data, and the documentation listed above can be used to implement your own systems for generating route data.

For route data to work with Eval Kit, it needs to be converted to so called “proute” format which is a string consisting of hexadecimal values. For binary formats, such as OpenLr and Eblr-gz, this is generated by reading the binary file byte-by-byte and concatenating these two-digit hexadecimal values into a string. For text formats the hexadecimal values are the hexadecimals corresponding to each character in the original input (polyline encoding uses ASCII encoding and Eblr uses UTF-8 encoding).

To use openlr_converter.py, minimum of three arguments need to be specified: input, output type and output format. Full list of arguments and their explanations can be seen by running “python openlr_converter.py -h”. For normal use the most useful arguments that are covered by the examples below are:

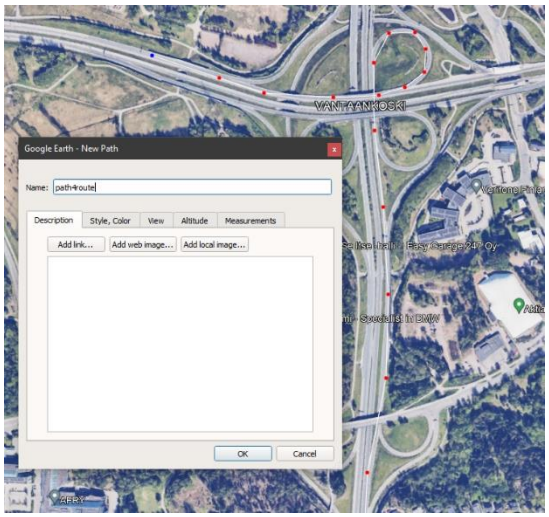
- Input: **--from-gpx-file** and **--from-kml-file**
- Output type: **--to-proute** and **--to-json-file**
- Output format: **--output-olb**, **--output-eblr**, **--output-eblr-gz** and **--output-polyline**

When output type “to-proute” is used, script simply prints the data which can then be pasted into e.g. trackfile. Using “to-json-file” creates a .json file that can directly be used through webapp or the contents can be fed over TCP interface.

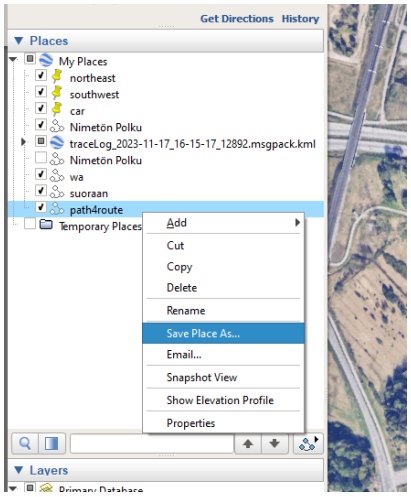
Note: the openlr_converter.py script used in the examples currently generates all formats through OpenLr which effectively limits the accuracy of route data to the precision supported by OpenLr.

7.3.1.1 Example 1: KML file to Eblr-gz proute string

1. Draw route on Google Earth pro using “path” tool



2. Save path as .kml file



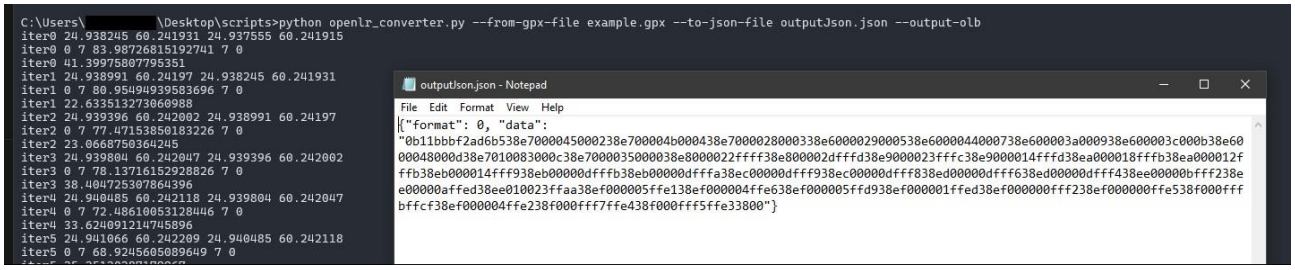
3. Run “python openlr_converter.py --from-kml-file <file.kml> --to-proute --output-eblr-gz”. This will print the route data in proute format. Format can be changed by replacing the argument --output-eblr-gz with other options listed in chapter 7.3.1.

```
C:\Users\          \Desktop\scripts>python openlr_converter.py --from-kml-file path4route.kml --to-proute --output-eblr-gz
1f8b08009f5b8e6602ffad97e16e53310c855fa5ea7f82edd88e2d6de351aab25e50a5d24ea51bf0f69c4c6ceca2de8b40b7da34654aa2cff6f1717af3e1fb97c3ea69387fdd9f8eb76b2eb45e0dc7fbd36e7ffc7cbb7ebc7c7a17eb0
f7737e7d3e36558ed77d8b15e3d9c4f4ffbdd70def435adef6e0ee787d5e174dcec8f9bdd70bf1b7052b484d754f756adbae3d8617b79bbc3a9489887530bc64eecf874bebf5d37fc3d7deb17af3e0edbf3334755ac76c7877efe0b18
920a5390d768c6f8a9ebd5f670d95f1e77c3af2d3d8cc3f0b4bd20accde5c7c3f0fc2f80be2ede4f734b5336ab9497e9dbb49043157e188596e1f737be1705792c870b665b1c152c5b27aa37a1d3b89a892a648ca1c768d11b659315
ccc1aa824b1b565b909d7a6e3b7da55ec24c99444ad4dea2cf65825de8a546621ab8ca4b765b19168d43391719950493292d5ccddcd66b179842db540d9de9a2b84e2eccb6287e0939c961ad7b1c5dc5cd3d8e6b33dd676d5a29503c5
54845d7559ea262865cf8ac9752b81ec1dc11145ce62737b4b6d590887504f48458cd8593dd4c094ec5441312510824b531b79a33d46d4c5d8b19a1db59041eb4b03e3402b737d8c944a655823291eee4d94ceb085aa3c02e9918d88
4965c5ad6701016168d6e4f57059230326fdc3ad734758e53dd0a2943eb026155dcbdb0ae310d480dc3a0f63a5ec5b6d1cc86399a73dc32363f881b71425d8a968996b9703f4625516998821 3caaed4903837aab3f9963f463b694904f
4dd8a0615c692ecb0d55630c1a3e34a16ef12ece6c4d7acf4e73774ffffcd1d514c2b399ab23659dab315f2b0083411deb04355e5062cae239376bc4753c6bbc60d0881b04d2bd68612f91aa6698bd78054ea4424141ed630ac79965bd
e7227d4ddc41aa67ba23d69e1c94e80867c71714c78092a5c6194 9038fd0336de22c5b993f7e6c0c3e77fc5bdfd7ad94cc0c3ff04ef6946149913f0247873d7fe74861af6c78fa7c7e36ef312c4ebfa57302febd7a0820b0ca50fae5e
60b93e7f5e4efd3d98f7cfdf2dee7e0264be2dcc890c0000

C:\Users\          \Desktop\scripts>
```

7.3.1.2 Example 2: GPX to OpenLr json file

1. Obtain GPX file (for this example data was generated with https://gpx.studio/)
2. Run “python openlr_converter.py --from-gpx-file <inputFile.gpx> --to-json-file <outputFile.json> --output-olb”. Json file with format and data in proute format is generated. Format can be changed by replacing the argument --output-olb with other options listed in chapter 7.3.1.

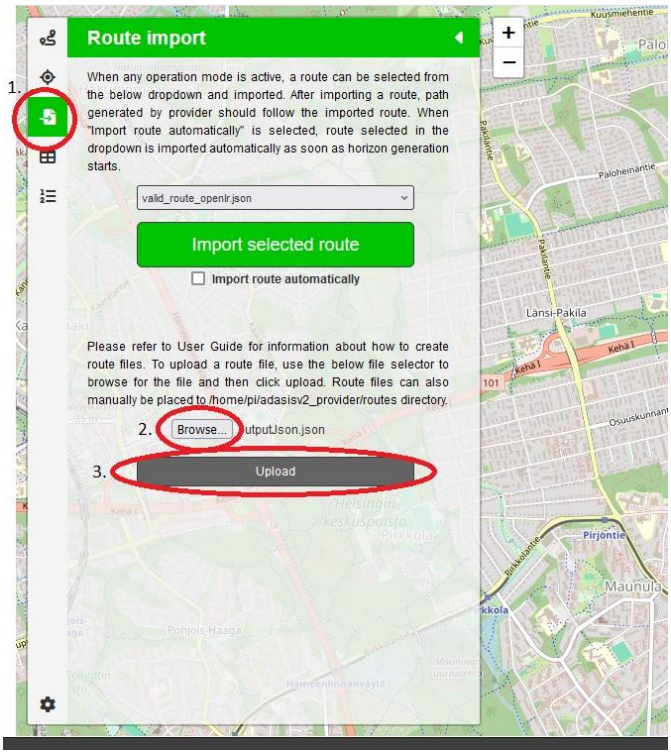


7.3.2 Route import through GUI

The easiest way to import a route is to use the webapp GUI. The example in chapter 7.3.1.2 directly generates compatible json files with the argument **--to-json-file**. In the pictures of this chapter the horizon visualization has been enabled but the process for importing routes with them disabled is the same except the menus look a little different.

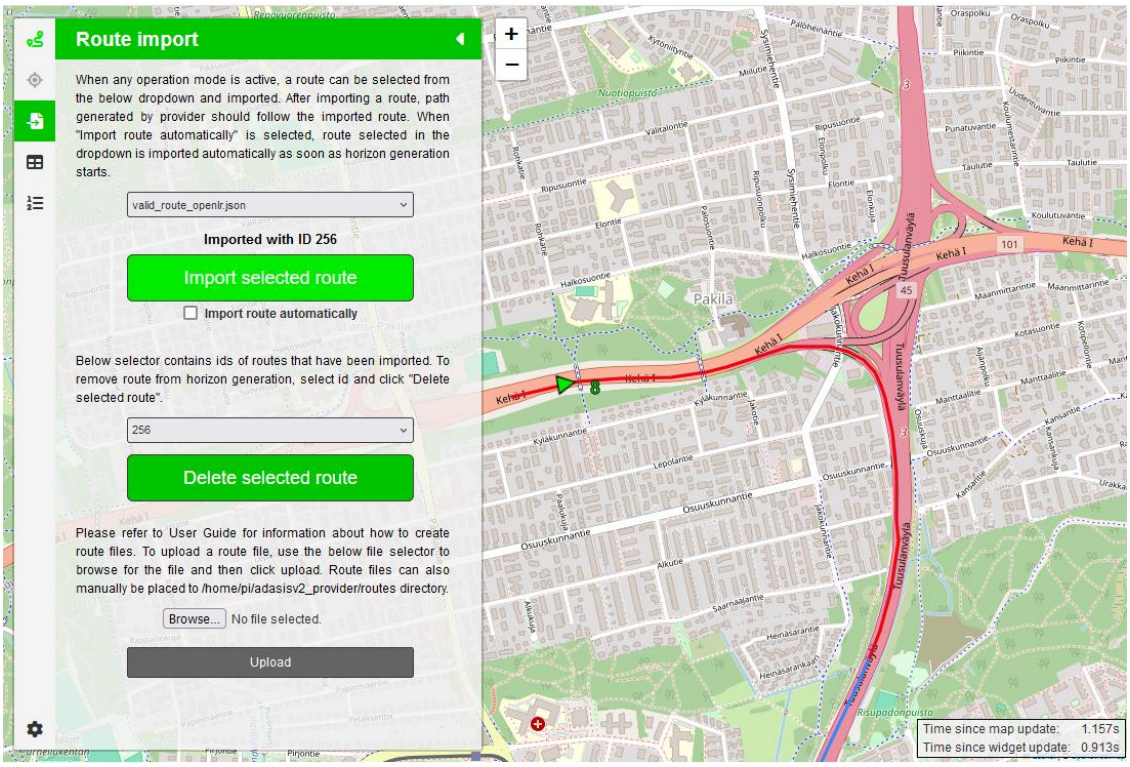
Once you have generated a json file with route data with e.g. openlr_converter.py, you can follow the following steps to upload and import the file:

1. Upload file (Open Route import panel, click “Browse” to select the file and then click upload)



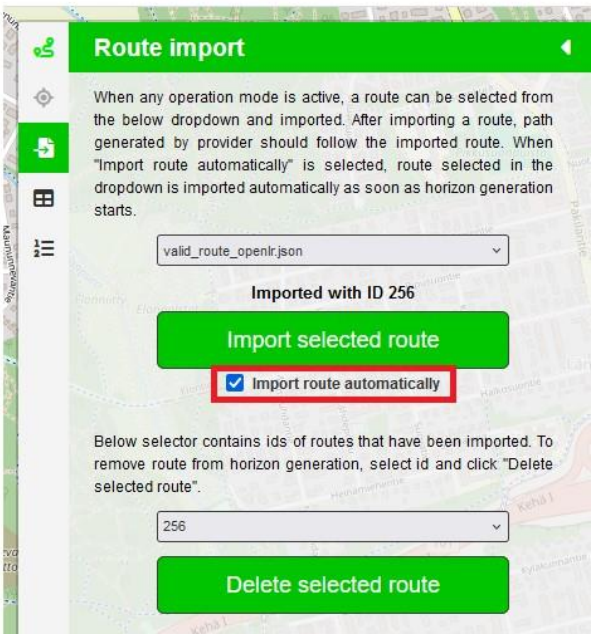
2. File should now appear in the dropdown selector. In case file doesn’t automatically appear, refresh the page.

3. Start operation (playback or live positioning)
4. When operation is running, select correct file from dropdown and click “Import selected route”
5. If import was successful, you should see a message with ID of the recently imported route and a new dropdown menu should appear which can be used to delete the route from horizon generation



6. If you want to remove route from horizon generation, select correct id from the lower dropdown and click “Delete selected route”

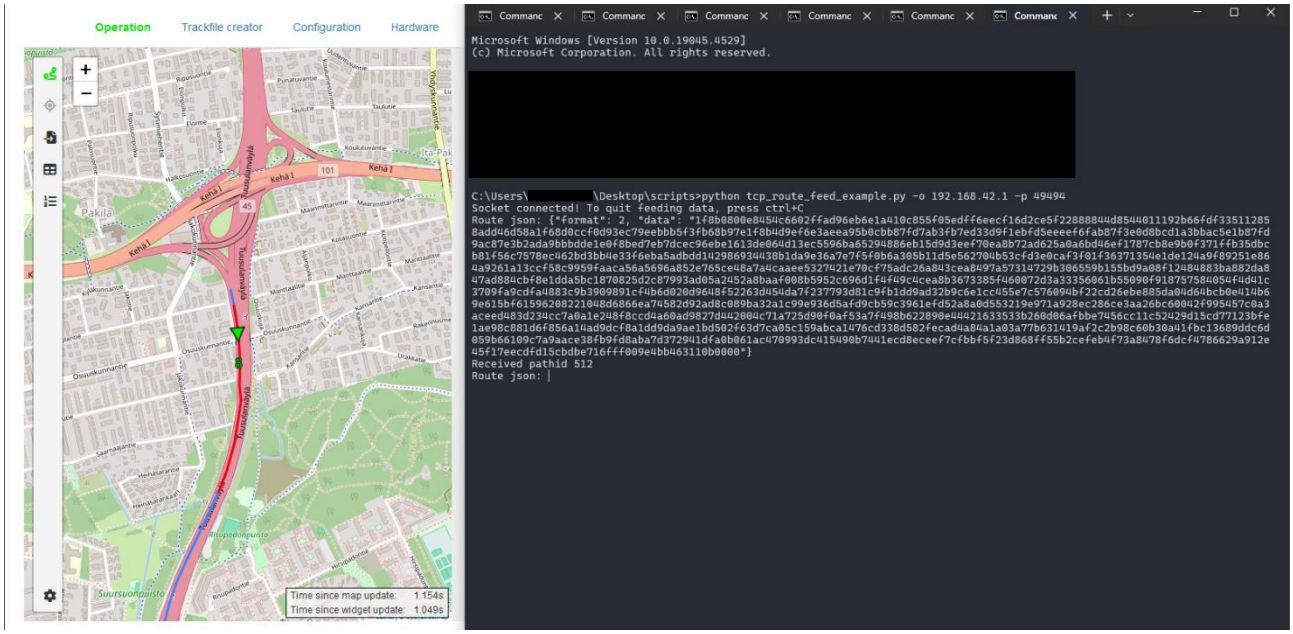
Route import can also be set to happen automatically when operation is started. To do this, check “Import route automatically” and start operation. Route selected in the dropdown at the time of starting operation will now automatically be imported as soon as vehicle position gets onroad. Once import has happened successfully, the menu will update in same manner as when doing manual import and thus the automatically imported route can be removed from horizon generation similar to manually imported routes.



7.3.3 Feeding routes over TCP Connection

An example python script for feeding route data over TCP from stdin is included on the Eval Kit under /home/pi/adasisv2_provider/tools. The script takes two arguments -o/--host which is the IP address of Eval Kit (192.168.42.1 (wifi) or 192.168.0.1 (wired)), and -p/--port which is the port for TCP route receiving. By default, the port is 49494, but it can be adjusted on the Configuration tab.

The TCP route receiver takes route data in the following JSON format {“format”: <data type>, “data”: <route data in PROUTE format string>} which is exactly the same as what is contained in the json files generated by openlr_converter.py with **--to-json-file** argument. Once you have generated a valid json (e.g. as in Chapter 7.3.1.2), start operation on Eval Kit and then start the tcp_route_feed_example.py. After this you can paste in the data and press Enter to send it. If import is successful, Eval Kit will respond with the route ID and it will be printed by the example script.



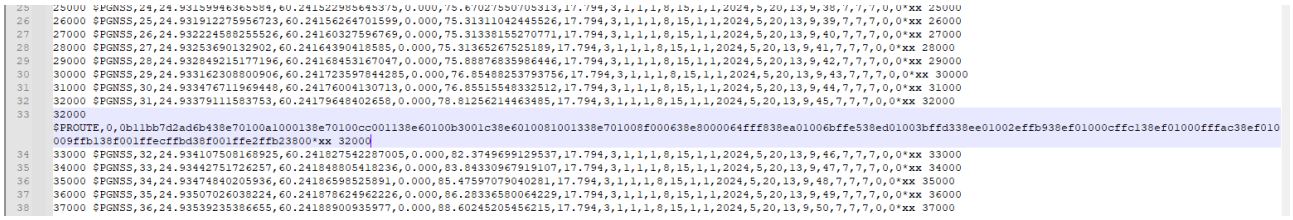
To delete a route through the TCP connection, send message with {“delete”: <ID>}. For this there will not be any response. Please note that importing can be done through webapp and TCP interface simultaneously, but imports done through TCP interface are not reflected in the GUI (e.g. route imported through the TCP interface won’t appear as option to delete in the webapp).

7.3.4 Route data in Trackfiles

Route data can also be imported from a trackfile. To do this, generate route data in proute string format (see Chapter 7.3.1.1) and add a line in the following format to a trackfile:

<timestamp> $PROUTE,<data-format>,<data-in-proute-format>*xx <timestamp>

In the line, <data-format> is an integer 0-3 depending on the format of route data (see list in the beginning of chapter 7.3) and <data-in-proute-format> is the data printed by script in the example of chapter 7.3.1.1. In the below example, trackfile contains a route at timestemp 3200 in format 0 (=OpenLr).



When trackfile with route data is played, the route is imported once playback reaches the line where data is located. This way, by adjusting the line where data is located, you can make sure route is imported at desired time.

7.4 Positioning format for UDP positioning (ADASISv2)

UDP positioning receives a list of position observation in JSON format with the "`positions`" key. The parameter description and constraints can be seen in the following subchapters. The examples are the following:

```
{"positions":[{"tick":2585849,"latitude":65.063807,"longitude":25.439697,"positi
on_error":10.0,"altitude":35.8,"altitude_error":10.0,"heading":2.914874383755729
6,"heading_error":0.08720872664625998,"velocity":4.76,"velocity_error":0.5,"angu
lar_velocity":0.02617993878,"angular_velocity_error":0.00872664626,"pitch":0.017
45329252,"pitch_error":0.005235987756}]}
```

```
{"positions":[{"tick":2586849,"latitude":65.063758,"longitude":25.439705,"positi
on_error":10.0,"altitude":35.6,"altitude_error":10.0,"heading":3.125361091546246
,"heading_error":0.08720872664625998,"velocity":5.44,"velocity_error":0.5,"angul
ar_velocity":0.02617993878,"angular_velocity_error":0.00872664626,"pitch":0.0174
5329252,"pitch_error":0.005235987756}]}
```

```
{"positions":[{"tick":2587849,"latitude":65.063703,"longitude":25.4397,"position
_error":10.0,"altitude":35.5,"altitude_error":10.0,"heading":3.205297171287586,"
heading_error":0.08720872664625998,"velocity":6.28,"velocity_error":0.5,"angular
_velocity":0.02617993878,"angular_velocity_error":0.00872664626,"pitch":0.017453
29252,"pitch_error":0.005235987756}]}
```

7.4.1 Parameter descriptions

`<*_error>` values are standard deviations.

tick: int Position observation timestamp. Currently this value is not in use. For future purposes.

latitude: double Estimated input position, WGS84 latitude in [deg].

longitude: double Estimated input position, WGS84 longitude in [deg].

position_error: double Estimated error circle (standard deviation) of horizontal input position [m].

altitude: double Height above the WGS84 reference ellipsoid [m].

altitude_error: double Estimated error ellipse (standard deviation) of input altitude [m].

heading: double Estimated absolute heading aligned to geographic north (0=North, pi/2=East, pi=South, 3/2pi=West). The values must be normalized to the range [0, 2pi) [rad].

heading_error: double Estimated error (standard deviation) of heading [rad].

velocity: double Estimated input movement velocity relative to heading (negative when moving backwards) [m/s].
velocity_error: double Estimated error (standard deviation) of input velocity [m/s].
angular_velocity: double Estimated turn rate of heading (positive in clockwise direction). Allowed range is [-pi, pi] [rad/s].
angular_velocity_error: double Estimated error (standard deviation) of input angular velocity [rad/s].
pitch: double Estimated pitch grade (positive uphill) [rad].
The pitch angle is the angle between the vehicle's attitude relative to level ground (or the horizontal plane perpendicular to gravity). Pitch is positive while driving uphill and negative while driving downhill. Pitch is 0.0 while driving on level surface.
pitch_error: double Estimated error ellipse (standard deviation) of pitch grade [rad].

7.4.2 Parameter constraints

All fields must be available in the input. In case the value is not available, it can be set to NaN (case-sensitive), e.g., heading_rate_error: NaN. In general, all parameters need to be filled with valid values, limited functionality is already available with tick, latitude, longitude, position_error, heading, and heading_error.

8 Using Elektrobit's Electronic Horizon Provider (ADASISv3)

The most convenient way to configure and control EB robinos Predictor Eval Kit and ADASISv3 Provider is to use the WebApp user interface. Connect your device (computer, cell phone, tablet device) by using the Ethernet or Wi-Fi credentials as described in chapter 4. You see the WebApp user interface (figure 20). In the navigation bar you can select between the two operating modes **Playback** and **Live Stream** as described in chapter 6.

In case you have been using the combined ADASISv2 and ADASISv3 EB robinos Predictor Eval Kit in ADASISv2 mode, change to ADASISv3 mode via the **Change ADASIS version** button in the **Configuration** tab.

8.1 Configuring ADASISv3 Provider

EB robinos Predictor Eval Kit provides a possibility to adjust the Elektrobit's ADASISv3 Provider configuration parameters. The configuration parameters are stored in the `adasisv3_Provider_conf.json` file. The file is located in the `adasisv3_provider` directory on the Raspberry Pi (`/home/pi/adasisv3_Provider/`). There are two possibilities to adjust these parameters: You can use the WebApp user interface (recommended) or access the Raspberry Pi's file system to change the parameters manually. You can find detailed information about the configurable parameters in chapter 8.1.1.

User preferences of automatic Live stream mode on device startup and trackfile saving enabled/disabled when using Live stream mode are also possible to be set. These settings are stored in the `startupConfig.json` file in Raspberry Pi's directory (`/home/pi/`).

To configure all relevant ADASISv3 configuration parameters within their value ranges, select the **Configuration** tab of EB robinos Predictor Eval Kit WebApp. Once you entered the desired values, click **Save** to save the current settings. In case you want to use the default values again, you can restore them by clicking the **Restore defaults** button.

To perform a safe shutdown for the Raspberry Pi device, press the power off icon in the rightmost tab of the navigation menu, select **Power Off**, and wait for **EBRaspi** Wi-Fi network to disappear before unplugging the power supply. The device can be rebooted by selecting **Reboot**.

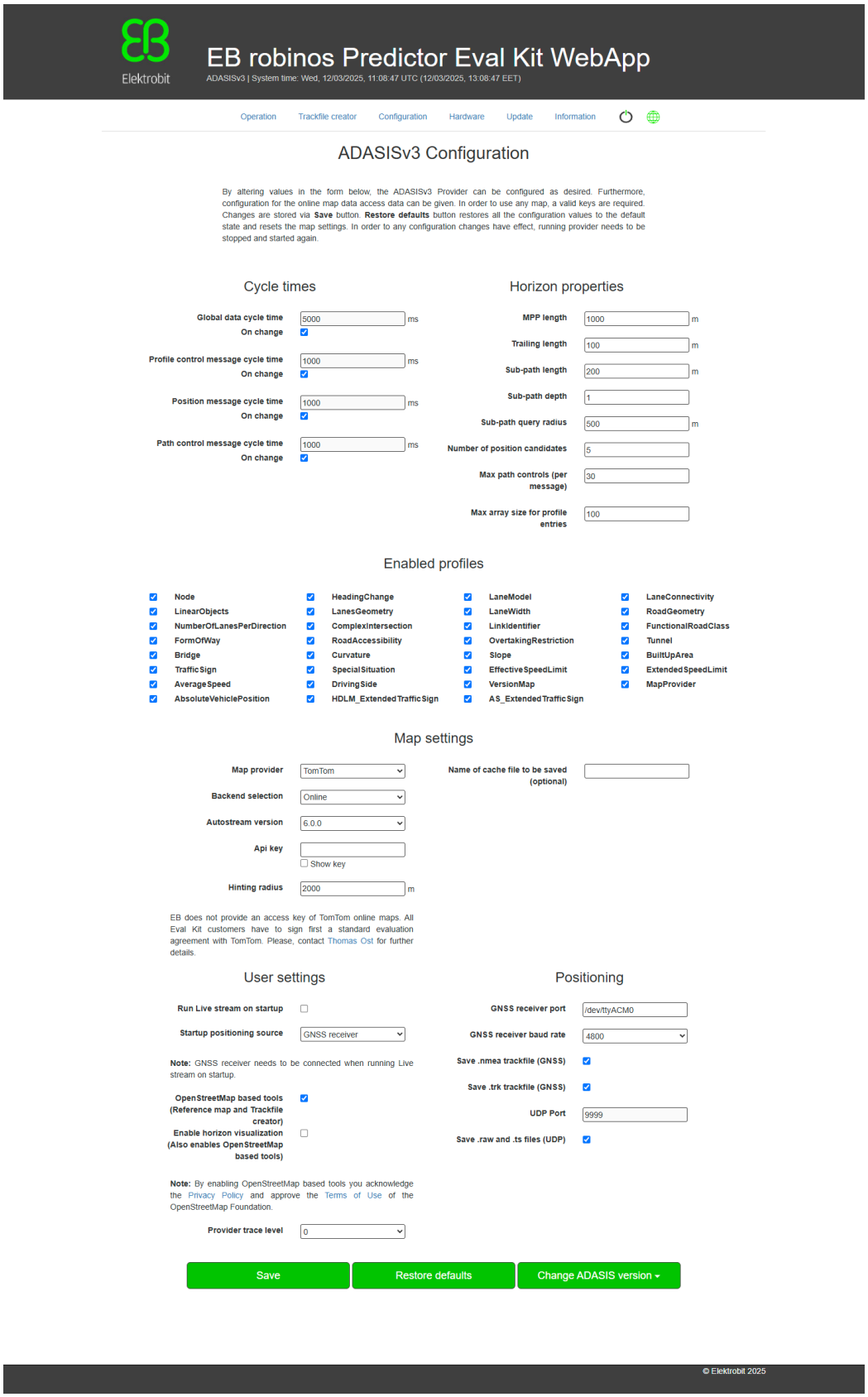


Figure 20: WebApp showing EB robinos Predictor Eval Kit Configuration tab

8.1.1 Configurable ADASISv3 parameters

The following table provides all parameters you can configure using the WebApp user interface:

No.	Attribute	Default value	Value range	Description
	Cycle times			
1-4	On change	True	True/False	When true, corresponding ADASISv3 messages are sent on change. If false, cycle times defined are used.
1	Global data	5000	0–[ms]	Defines period time of ADASISv3 global data message in milliseconds.
2	Profile control message	1000	0–[ms]	Defines period time of ADASISv3 profile control message in milliseconds.
3	Position message	1000	0–[ms]	Defines period time of ADASISv3 position message in milliseconds. Presents the positions of the vehicle on the path network and describes movement of the vehicle.
4	Path control message	1000	0–[ms]	Defines period time of ADASISv3 path control message in milliseconds.
	Horizon properties			
5	MPP length	1000	1–[m]	Defines up to which length the MPP shall be expanded and kept.
6	Trailing length	100	0–500 [m]	Defines the trailing length (electronic horizon data behind the current vehicle position that is kept before deleting).
7	Sub-path length	200	1–[m]	Defines up to which length sub-paths shall be expanded and kept.
8	Sub-path depth	1	0–	Maximum depth (nesting level) of sub-paths in the electronic horizon path tree.
9	Sub-path query radius	500	1–2000 [m]	Search distance for expanding sub-paths along the electronic horizon tree from the current vehicle position.
10	Number of position candidates	5	1–5	Maximum number of map-matched position candidates taken into account when building up electronic horizon.

11	Max path controls	30	0–	Maximum number of path controls per path control message.
12	Max array size for profile entries	100	0–	Defines how large the arrays containing horizon profile entries can be. Zero for unlimited array size.
	Profile configuration			
13	• Node • HeadingChange • LaneModel • LaneConnectivity • LinearObjects • LanesGeometry • LaneWidth • RoadGeometry • NumberOfLanesPer Direction • ComplexIntersection • LinkIdentifier • FunctionalRoadClass • FormOfWay • RoadAccessibility • OvertakingRestriction • Tunnel • Bridge • Curvature • Slope • BuiltUpArea • TrafficSign • SpecialSituation • EffectiveSpeedLimit • ExtendedSpeedLimit • AverageSpeed • DrivingSide • VersionMap • MapProvider • AbsoluteVehiclePosition • HDLM_ExtendedTrafficSign • AS_ExtendedTrafficSign	ALL ENABLED	True/False	Select which profiles are transmitted.

	Map settings			
14	Map provider	TomTom	TomTom/ HERE/ NDS.Live	Selection of supported map providers.
15	Backend selection	Online		Map provider backend connectivity mode. **TomTom:** Online, TestServer, and Offline connectivity modes are available. Online and TestServer backends require a functioning Internet connection. To use offline backend, pre-loaded map data cache file needs to be available. **HERE:** Online connectivity mode available for maps of Europe (EUR) and North America (NAR), requires a functioning Internet connection. **NDS.Live:** Online and offline connectivity modes are available. Online backend requires a functioning Internet connection. To use offline backend, pre-loaded map data cache file needs to be available.
16	AutoStream version	6.0.0		**TomTom** AutoStream client library version. Currently only version 6.0.0 is supported. This item is TomTom-specific.
17	TomTom: Api key HERE: Key ID/Key secret NDS.Live: Api key	-	-	User credentials for map provider application. **TomTom:** Elektrobit does not provide an access key of TomTom online maps. All Eval Kit customers must sign first a standard evaluation agreement with TomTom. For further details, contact Thomas Ost (Thomas.Ost@tomtom.com). **HERE:** Elektrobit does not provide an access key of HERE online maps. All Eval Kit customers have to contact HERE and apply for the key directly from the map's vendor. For further

				details, contact Pierre Lewandowski (Pierre.Lewandowski@here.com). **NDS.Live**: Elektrobit does not provide an access key by providers of NDS.Live backend maps. All Eval Kit customers have to contact an NDS.Live map provider and apply for the key directly from the map's vendor.
18	Hinting radius	2000	0–[m]	Determines the radius for preloading data from the map. Every time a link is loaded, a position hint is provided to the map backend to load further data in advance. The center point of the radius is the last coordinate of the center line geometry of the link in driving direction. This item is currently **TomTom**-specific.
19	Url	-	-	NDS.Live URL to backend. This item is **NDS.Live** map provider-specific.
20	Port	65535	0–65535	NDS.Live network port to backend. This item is **NDS.Live** map provider-specific.
21	Name of cache file to be saved	-	-	Optional. To use ADASISv3 Provider in offline mode without a functioning Internet connection, pre-loaded map data cache needs to be saved beforehand in online mode. After playback or live stream mode have been used in online mode, the saved map data cache file from the respective area can be selected for use in offline backend mode. This item is currently **TomTom**- and **NDS.Live**-specific.
22	Catalog version	0		Map provider backend version. Value 0 means the latest version available will be used. This item is **HERE**-specific.

				To get more information about available map versions, etc., contact the respective map supplier directly. **HERE:** Pierre Lewandowski (Pierre.Lewandowski@here.com)
23	Certificate	-	-	SSL certificate file for online map backend. Files in `.pem` format uploaded to Samba Share folder `Map certificates` or `/home/pi/map_cert` are listed here. This item is currently **NDS.Live** map provider-specific.
24	Node	service	service	NDS.Live Network node type to backend. Currently only **service** is supported. This item is **NDS.Live** map provider-specific.
25	Authentication type	basic	basic	NDS.Live Network authentication type to backend. Currently only **basic** is supported. This item is **NDS.Live** map provider-specific.
26	Protocol type	http_rest	http_rest	NDS.Live Network protocol type to backend. Currently only **http rest** is supported. This item is **NDS.Live** map provider-specific.

Table 9: Configurable ADASISv3 parameters

8.1.2 Configuration for positioning

The following table covers all positioning related parameters which are configurable by using the WebApp user interface. Click **Restore defaults** in the **Configuration** tab to restore the pre-defined default values for each parameter. By default, trackfiles are always recorded and saved when GNSS positioning is used. Similarly, .raw and .ts files are recorded from UDP positioning input by default. Saving of these files can be enabled or disabled in the **Configuration** tab under **Positioning**.

No.	Attribute	Default value	Description
1	GNSS receiver port	`/dev/ttyACM0`	Port to which device is attached. Devices listed in `/dev` folder in Raspberry Pi.
2	GNSS Baud rate	4800	The speed at which the data is transferred from the GPS receiver to the Raspberry Pi. Different receivers use different speeds.
3	Save .nmea trackfile (GNSS)	Enabled	When enabled, nmea trackfile will be recorded when Live GNSS operation mode is used. Generated files can be found under /home/pi/adasisv3_provider/nmea.
4	Save .trk trackfile (GNSS)	Enabled	When enabled, trk trackfile will be recorded when Live GNSS operation mode is used. Generated files can be found under /home/pi/adasisv3_provider/trk.
5	UDP port	9999	Port for UDP positioning.
6	Save .raw and .ts files (UDP)	Enabled	When enabled, .raw and .ts files will be recorded when UDP positioning is used. Generated files can be found under /home/pi/adasisv3_provider/udp.

Table 10: Configurable GNSS receiver parameters

When GNSS positioning is used, ADASISv3 Provider expects to receive GNSS position data which is used to generate an electronic horizon. Any USB GNSS receiver that feeds positions in NMEA 0183 format (RMC, GGA and GSA messages are at least required) can be used.

When UDP positioning is used, ADASISv3 Provider expects to receive a list of position observations in JSON format which can be sent as UDP packets (recommended frequency of 1 Hz) by using EB robinos Predictor Eval Kit's IP (192.168.42.1 or 192.168.0.1 depending on used connection) and port 9999. Input fields have a string type with units specified in chapter 8.4. Note that an invalid JSON string will be ignored. Currently there is no support for saving trackfiles based on positions received via **UDP interface**. However, for investigation purposes, there is an option to save a binary file and a timestamp file that are coming to the UDP interface under the Positioning configuration in the **Configuration** tab. That raw binary data file and timestamp file can be saved in `/home/pi/adasisv3_provider/udp` as extensions of `.raw` and `.ts` with the same filename convention as defined in chapter 6.2. These raw data files can be sent to the Elektrobit team for investigation or used for your own purposes.

Position observation description can be found in chapter 8.3.

8.1.3 Configuration for user settings

Via the User settings on the **Configuration** tab, it is possible to select default Live positioning source. This affects the behavior of **Run Live stream on startup.** If **Run Live stream on startup** is enabled, upon booting EB robinos Predictor Eval Kit will start directly on the configured Live positioning mode. All saved settings will

be used on device startup, including the desired option for enable/disable trackfile saving when using GNSS receiver/UDP interface. Remember that the GNSS receiver needs to be connected on startup for Live stream to start. Saved trackfiles can be found in `/home/pi/adasisv3_provider/trk` and `/home/pi/adasisv3_provider/nmea`.

Additionally, user settings contain settings related to OpenStreetMap tools. **OpenStreetMap based tools** enables reference map for input positions on **Operation** tab and trackfile creator functionality. **Enable horizon visualization** enables visualization of generated horizon on the **Operation** tab. Note: horizon visualization is partly based on OpenStreetMap and thus enabling it also enables other **OpenStreetMap based tools.**

Provider traces enabled controls tracing level for AdasisV3 Provider. This should be kept as OFF to prevent SD card from filling up but if there is for example needs to share logs with Elektrobit, then the traces can be enabled. Tracelogs are generated to `/home/pi/adasisv3_provider/tracelogs`.

8.1.4 ADASISv3 hardware settings

ADASISv3-related hardware settings can be changed via the **Hardware** tab.

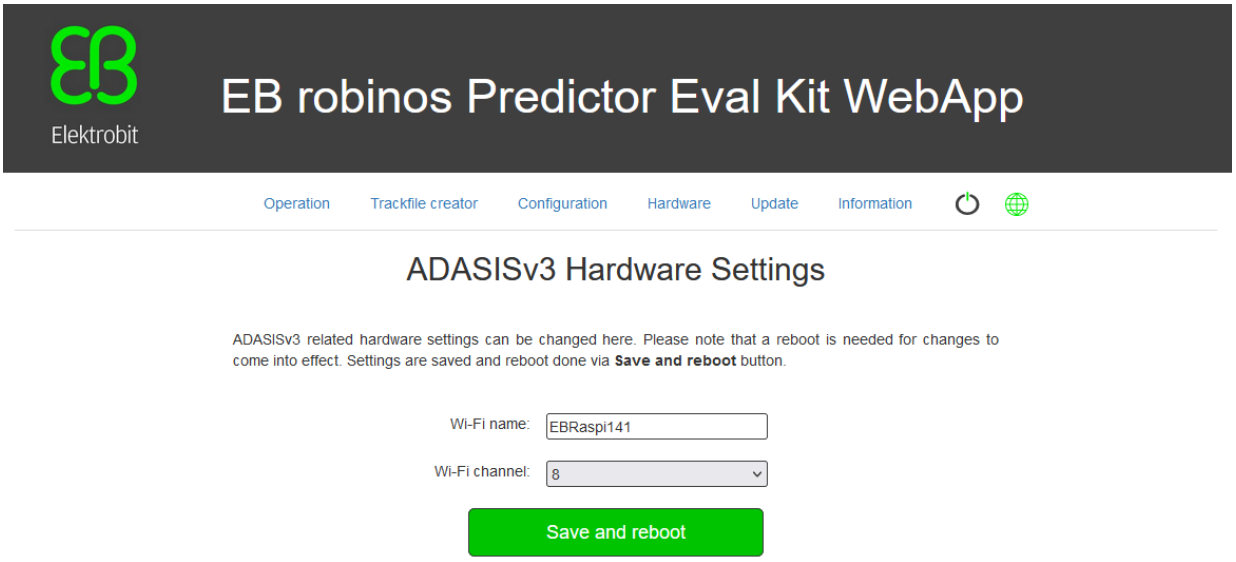


Figure 21: WebApp showing EB robinos Predictor Eval Kit ADASISv3 hardware.

Note that a reboot is needed for changes to become effective. Settings are saved and a reboot is started via the **Save and reboot** button.

The following table provides all ADASISv3 hardware settings:

No.	Attribute	Value range	Description	Default value

1	Wi-Fi name	Text string that can contain letters, numbers and _ characters. Maximum length is 32 characters.	Wi-Fi name that is used when computer is connected to the EB robinos Predictor Eval Kit's Wi-Fi network. See chapter 4.2 for more details.	Preset when Eval Kit is assembled.
2	Wi-Fi channel	1–11	Wi-Fi channel that is used when computer is connected to the EB robinos Predictor Eval Kit's Wi-Fi network. If several Eval Kits are used close to each other, different Wi-Fi channels should be used. Recommended channels to use are 1, 6, and 11. On these channels used frequencies do not overlap.	Preset when Eval Kit is assembled.

Table 11: Configurable hardware settings parameters

8.2 Operating ADASISv3 Provider

To operate the ADASISv3 Provider, select the **Operation** tab on the EB robinos Predictor Eval Kit WebApp when ADASIS version 3 has been selected on the **Configuration** tab. The content of the **Operation** tab is dependent on whether or not horizon visualization is enabled in **Configuration** tab chapter 8.1.3. if visualization is enabled, see chapter 8.2.1 for detailed instructions. Chapter 8.2.2 details usage when visualization is disabled.

8.2.1 ADASISv3 Operation (visualization enabled)

When **Enable horizon visualization** parameter is enabled on the **Configuration** tab, the **Operation** page consists of a reference map which visualizes the generated ADASISv3 horizon and optionally the selected trackfile and/or current input position, and a 3D widget which visualizes the lane level details contained in ADASISv3 data. For further details about data presented in the 3D view, click the highlighted button (Figure 22).

With heavy ADASISv3 Provider configuration, there might be performance issues with the horizon visualization. Functionality has been verified with default configuration. Please also note that while the horizon visualization can work in offline mode, in order for the reference map to be able to load map data, the device used to operate WebApp also needs to be connected to internet.

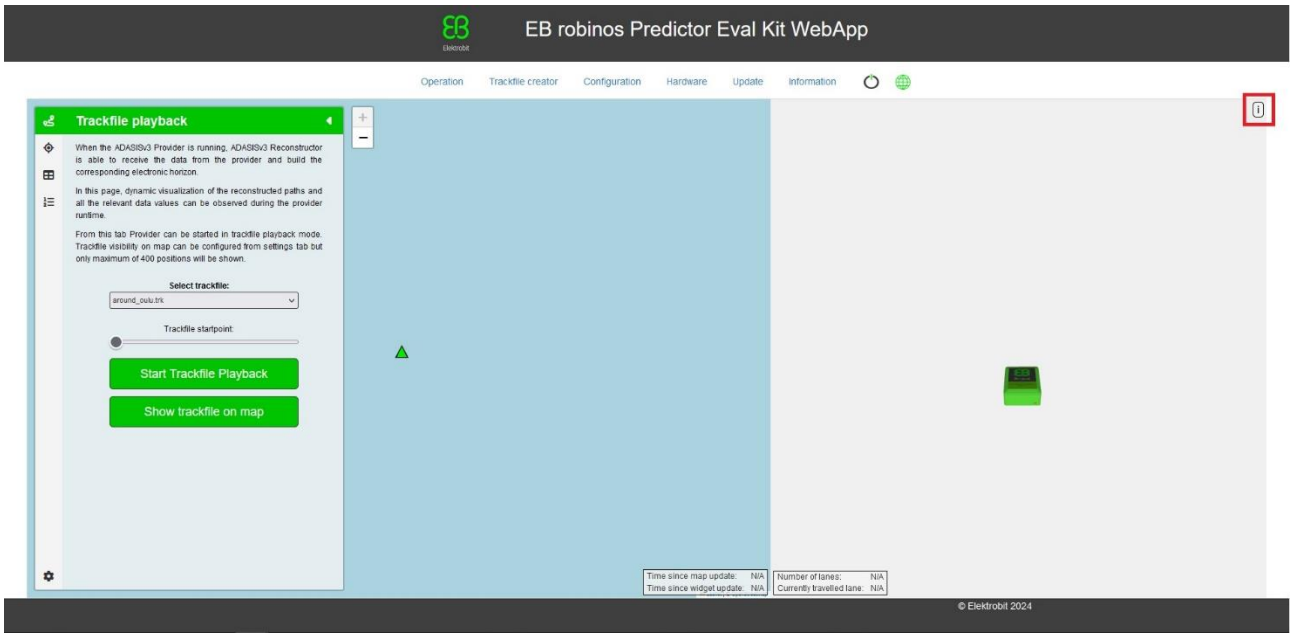


Figure 22: ADASISv3 Operation page when horizon visualization is enabled

8.2.1.1 Trackfile Playback

To start ADASISv3 Provider in trackfile playback mode, select the first icon from the sidebar located on the left side of the reference map (Figure 22). From there, trackfile can be selected from the dropdown menu and the row of trackfile from which playback should start, can be set with the slider located below the dropdown. To start playback, click **Start Trackfile Playback.** To preview trackfile on the reference map before starting playback, click **Show trackfile on map**. During playback, trackfile visibility and current input position visibility can be toggled on the settings panel (Figure 23). From there, other settings related to horizon visualization can also be adjusted.

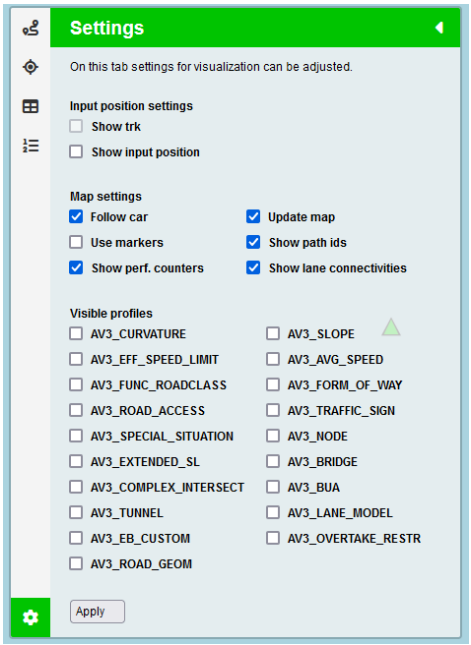


Figure 2315: ADASISv3 horizon visualization settings panel

8.2.1.2 Live Positioning

To start ADASISv2 Provider in UDP, GNSS or CAN positioning mode, select the second icon from the sidebar located on the left side of the reference map (Figure 24). On this panel, the desired positioning mode can be selected from the dropdown and operation can be started by clicking **Start recording.** To stop operation, click **Stop recording** from the same panel.

Further details about the available live positioning modes can be found from chapter 6.2 (UDP) and chapter 6.3 (GNSS).

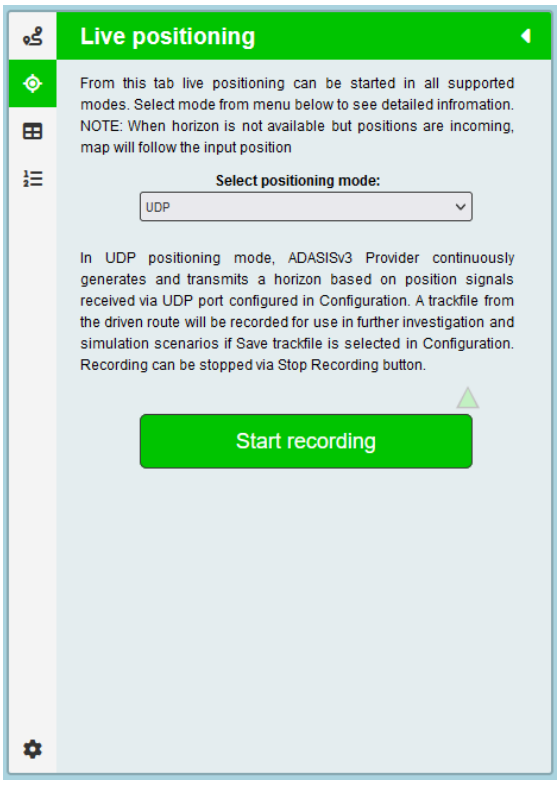


Figure 2416: ADASISv3 Live Positioning panel

8.2.2 ADASISv3 Operation (visualization disabled)

When **Enable horizon visualization** parameter is disabled on the **Configuration** tab, the **Operation** page consists of controls for starting operation and a position input reference map (if **OpenStreetMap based tools** is enabled on the **Configuration tab**). When reference map is enabled, the button located under zoom controls can be used to toggle current input positioning following on or off. Operation between trackfile playback and Live positioning can be chosen from **Positioning mode** selector located on top of the page.

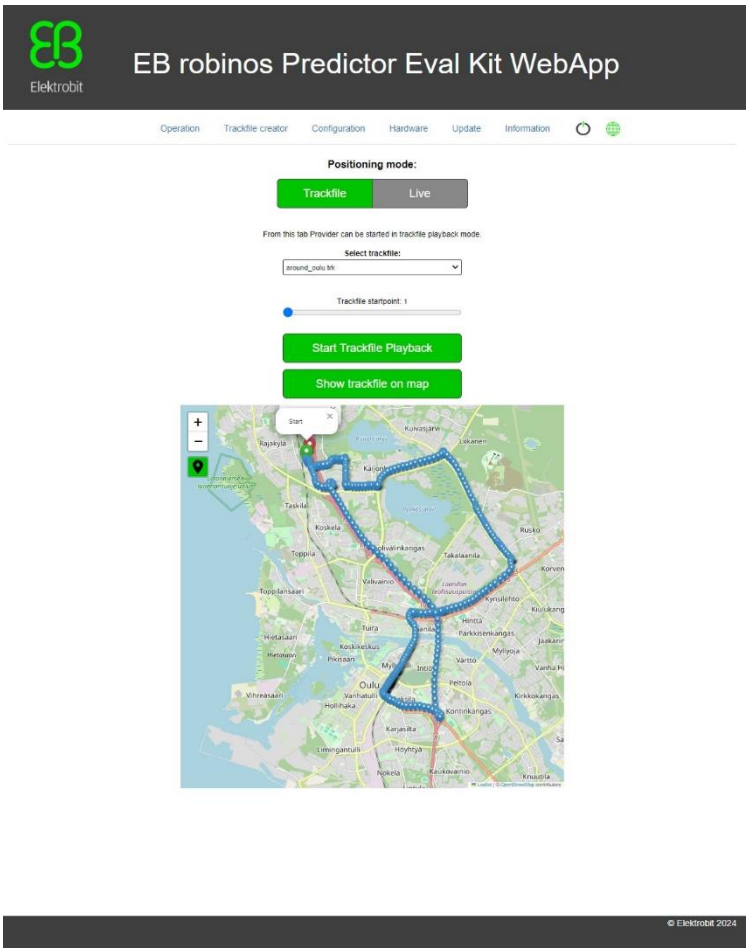


Figure 25: ADASISv3 Operation page without horizon visualization

8.2.2.1 Trackfile Playback

When Trackfile **Positioning mode** is selected on the operation tab (Figure 25), trackfile can be selected from the dropdown and slider located below it can be used to set row of trackfile where playback should start. To start playback, click **Start trackfile playback**. To preview trackfile on the reference map (if enabled) before starting playback, click **Show trackfile on map**.

8.2.2.2 Live Positioning

When Live **Positioning mode** is selected on the operation tab (Figure 26), the structure of page is otherwise the same as in trackfile playback mode but now the dropdown can be used to select between UDP and GNSS positioning modes. To start operation, click **Start recording.** Further details about the available live positioning modes can be found from chapter 6.2 (UDP) and chapter 6.3 (GNSS).

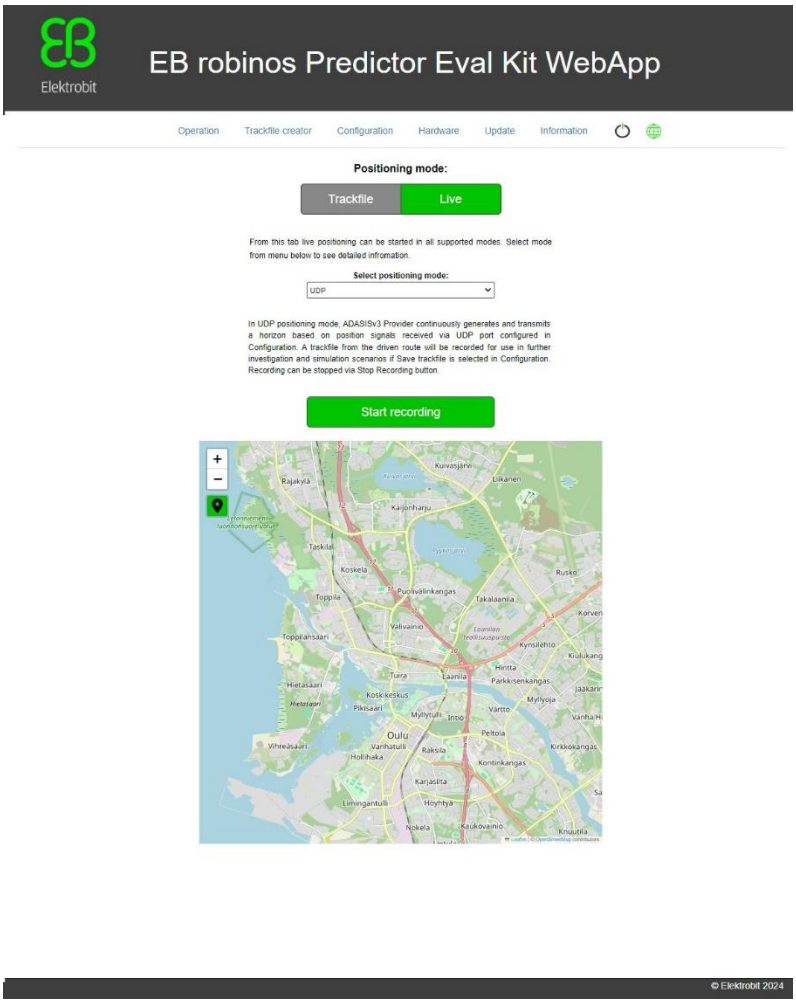


Figure 26: ADASISv3 Operation page without horizon visualization in Live positioning mode

8.3 Positioning format for UDP positioning (ADASISv3)

UDP positioning receives a list of position observation in JSON format with the "`positions`" key. The parameter description and constraints can be seen in the following subchapters. The examples are the following:

```
{"positions":[{"tick":2585849,"latitude":65.063807,"longitude":25.439697,"positi
on_error":10.0,"altitude":35.8,"altitude_error":10.0,"heading":2.914874383755729
6,"heading_error":0.08720872664625998,"velocity":4.76,"velocity_error":0.5,"angu
lar_velocity":0.02617993878,"angular_velocity_error":0.00872664626,"pitch":0.017
45329252,"pitch_error":0.005235987756}]}
```

```
{"positions":[{"tick":2586849,"latitude":65.063758,"longitude":25.439705,"positi
on_error":10.0,"altitude":35.6,"altitude_error":10.0,"heading":3.125361091546246
,"heading_error":0.08720872664625998,"velocity":5.44,"velocity_error":0.5,"angul
ar_velocity":0.02617993878,"angular_velocity_error":0.00872664626,"pitch":0.0174
5329252,"pitch_error":0.005235987756}]}
```

{"positions":[{"tick":2587849,"latitude":65.063703,"longitude":25.4397,"position_error":10.0,"altitude":35.5,"altitude_error":10.0,"heading":3.205297171287586,"heading_error":0.08720872664625998,"velocity":6.28,"velocity_error":0.5,"angular_velocity":0.02617993878,"angular_velocity_error":0.00872664626,"pitch":0.01745329252,"pitch_error":0.005235987756}]}

The non-existing value will be treated as `NaN` and it will not be used for the map-matching calculation.

8.3.1 Parameter descriptions

`<*_error>` values are standard deviations.
tick: double Sensor tick that is source of this observation in [ms]. Note that the position observation must come in with positive increasing ticks, since the system cannot handle backward time jumps.
latitude: double Estimated input position, WGS84 latitude in [deg].
longitude: double Estimated input position, WGS84 longitude in [deg].
position_error: double Estimated error ellipse (standard deviation) of input position [m].
altitude: double Height above the WGS84 reference ellipsoid [m].
altitude_error: double Estimated error ellipse (standard deviation) of input altitude [m].
heading: double Estimated heading aligned to geographic north in radians [rad] (0=North, pi/2=East, pi=South, 3/2pi=West).
heading_error: double Estimated error ellipse (standard deviation) of heading in radians [rad].
velocity: double Estimated input movement velocity relative to heading (negative when moving backwards) [m/s].
velocity_error: double Estimated error (standard deviation) of input velocity [m/s].
angular_velocity: double Estimated turn rate of heading (positive in clockwise direction). Allowed range is [-pi, pi] [rad/s].
angular_velocity_error: double Estimated error (standard deviation) of input angular velocity [rad/s].
pitch: double Estimated pitch grade (positive uphill) [rad].
The pitch angle is the angle between the vehicle's attitude relative to level ground (or the horizontal plane perpendicular to gravity). Pitch is positive while driving uphill and negative while driving downhill. Pitch is 0.0 while driving on level surface.
pitch_error: double Estimated error ellipse (standard deviation) of pitch grade [rad].

8.3.2 Parameter constraints

The following fields in `position_observation` are mandatory to be filled for ADASISv3 Provider's map-matching to work properly.

`tick`, `latitude`, `longitude`, `position_error`, `heading`, `heading_error`, `velocity`, `velocity_error`

If some of the mandatory fields are provided as `NaN`, map-matching cannot be performed and will produce an offroad position.
The following fields are optional but will improve map-matching precision:

`altitude`, `altitude_error`, `angular_velocity`, `angular_velocity_error`, `pitch`, `pitch_error`

9 OpenStreetMap-based tools

Visualizing and generating input position data for ADASISv2 Provider and ADASISv3 Provider is available via OpenStreetMap-based tools. To enable Reference map and Trackfile creation, select the **OpenStreetMap-based tools** checkbox in the **Configuration** tab.

9.1 Trackfile creator

In the Trackfile creator tab, a trackfile can be created between given start and end coordinates. The Open Source Routing Machine-based solution is used to execute routing with OpenStreetMap data for trackfile creation. Map data (`.osm` format) can be obtained, for example, from https://www.openstreetmap.org/export and uploaded to Eval Kit. Note that the start and end coordinates must be within the map area for successful trackfile creation. Also note that the map data used to generate horizon may differ with OpenStreetMap data.

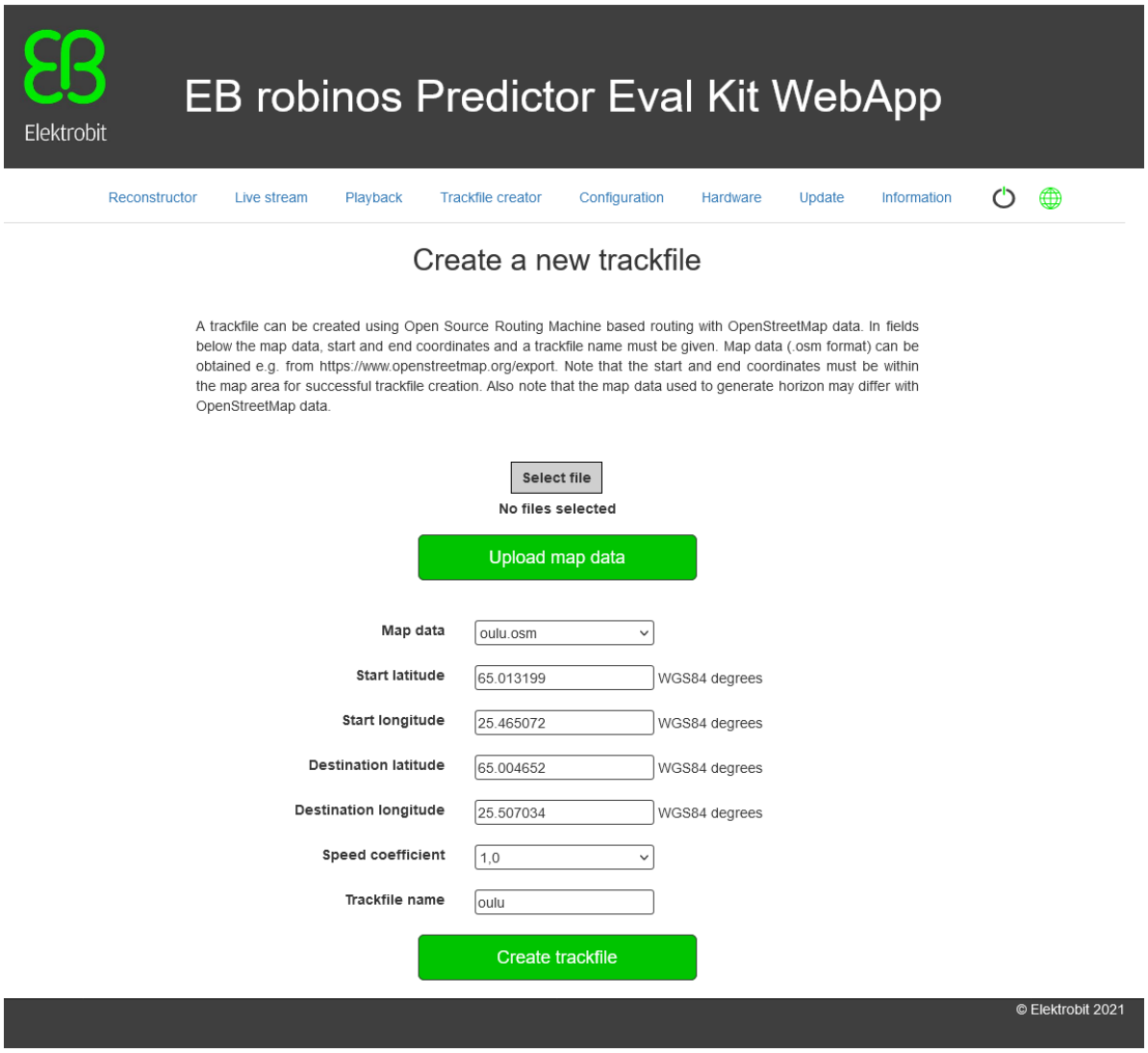


Figure 177: Trackfile creator

10 Updating EB robinos Predictor Eval Kit

There are two methods to update EB robinos Predictor Eval Kit, partially and fully. Every software update contains detailed release notes about the update procedure.

Note that the provided software updates run only on EB robinos Predictor Eval Kits that were shipped by Elektrobit or Raspberry Pi devices with valid activation keys provided by Elektrobit. For further information about activation keys refer to chapter 10.3.

10.1 Performing a partial software update

A partial software update consists of a zipped file (i.e. `update.7z`). You must extract this zip file to your local PC. After that, install the extracted update file (i.e. `update.tar.gz`) to the Raspberry Pi using the WebApp's **Update** tab. In the **Update** tab, press the **Select file** button and browse for the update file from your local PC. Click the **Update** button and wait until the update process finishes. After the update process is finished, you can use the Eval Kit and all of its features again.

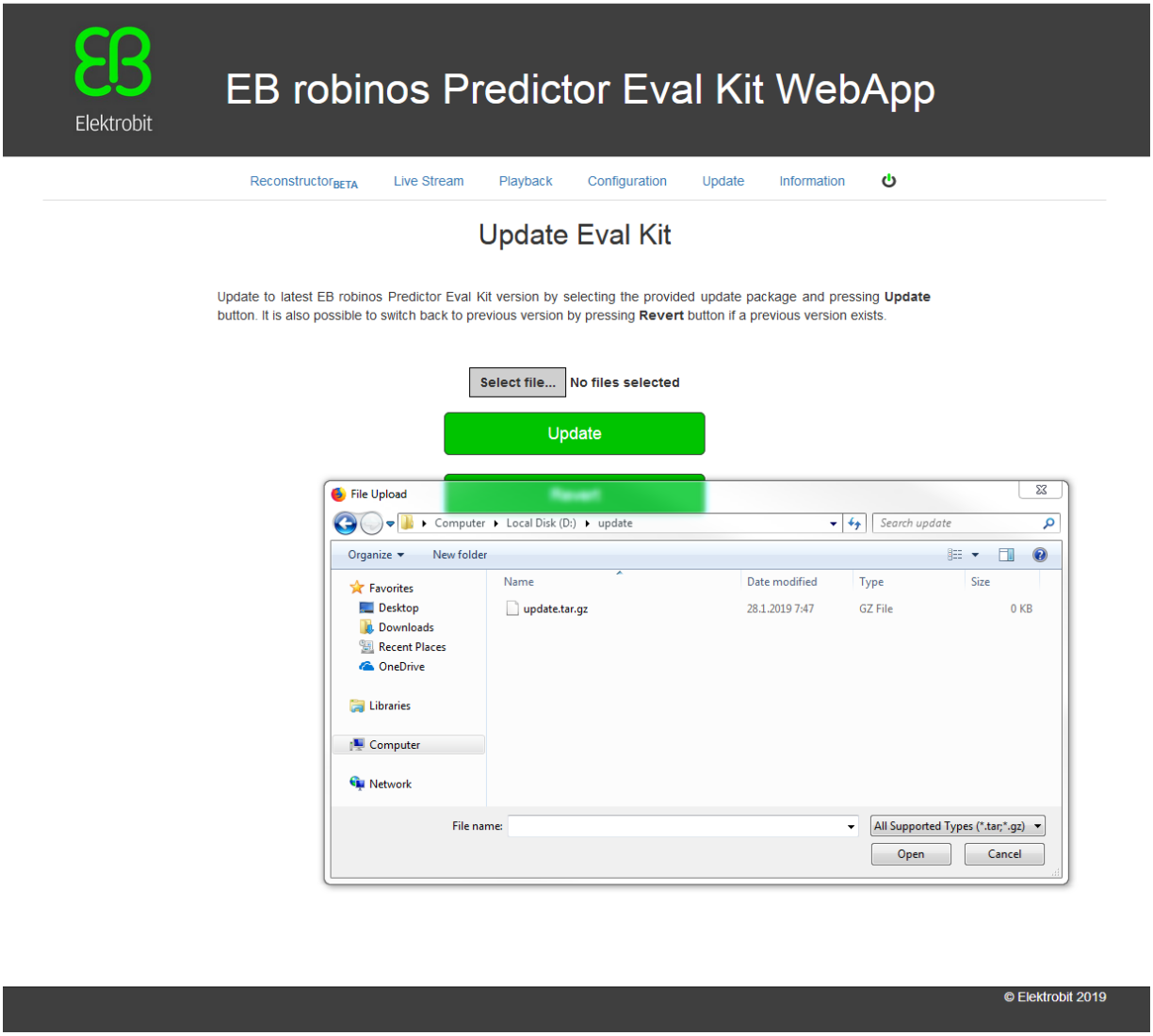


Figure 188: WebApp showing EB robinos Predictor Eval Kit update process

It is also possible to switch back to the previous version of EB robinos Predictor Eval Kit in case there are any issues with the updated version. Click **Revert** in the **Update** tab to switch back to the previous version.

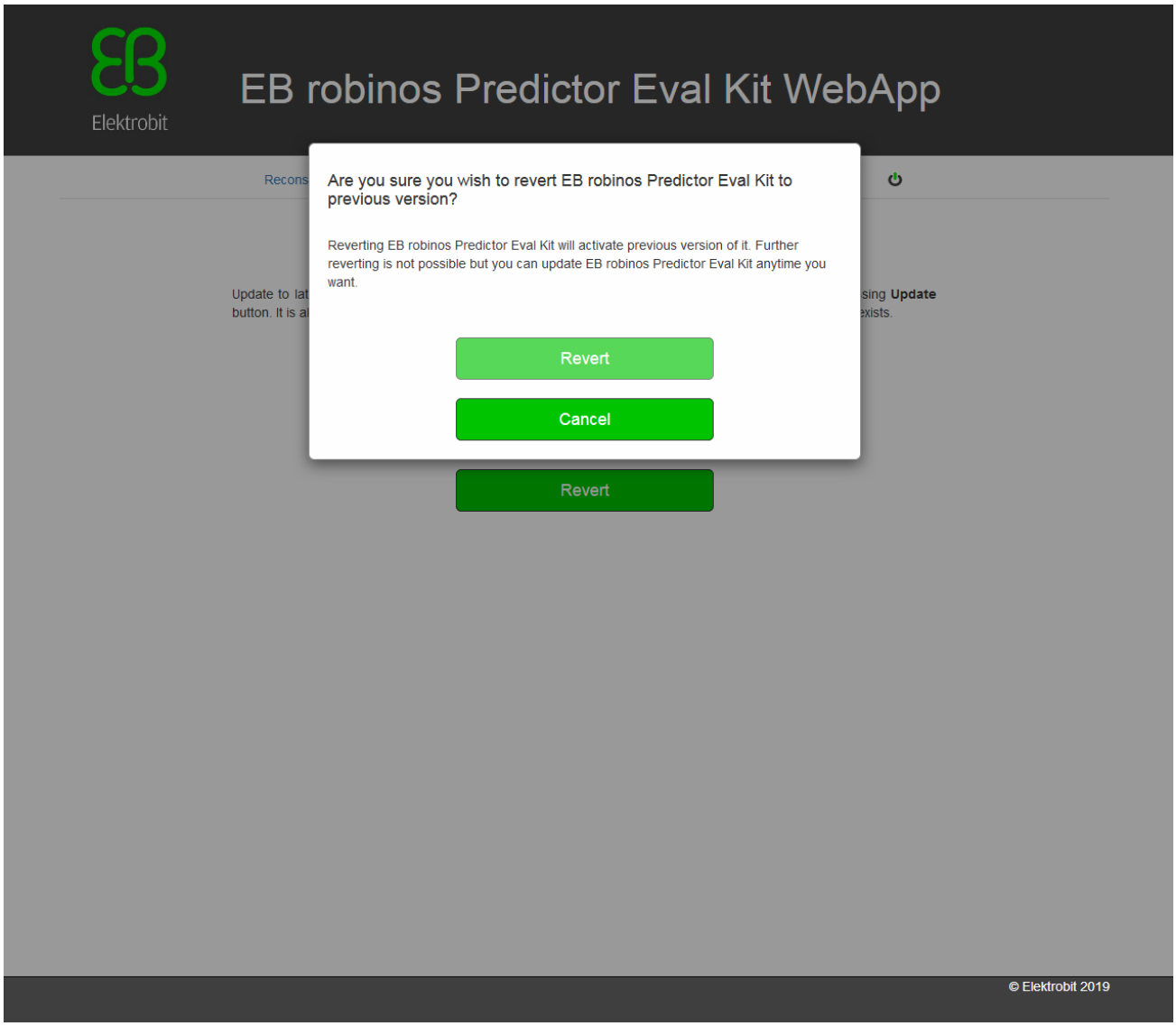


Figure 199: WebApp showing EB robinos Predictor Eval Kit ADASISv3 Provider revert version process

10.2 Full software update

A full software update consists of a ready-to-use SD card image which needs to be written to the Raspberry Pi's micro SD card. In this chapter, the steps for updating the software are described.

10.2.1 Preparing a full software update

ATTENTION: Once a new SD card image file is written to a SD card, all customer-specific content that is stored on the SD card will be irrevocably lost and the ADASISv2 and ADASISv3 Provider configurations are set to default values.

To store possibly modified configuration and trackfiles, copy the following files to your local PC before you start the update process:

ADASISv2-related items:

- `/home/pi/adasisv2_provider/trk/*`
- `/home/pi/adasisv2_provider/*.json`
- `/home/pi/adasisv2_provider/*.conf`
- `/home/pi/adasisv2_provider/*.cfg`

ADASISv3-related items:

- `/home/pi/adasisv3_provider/nmea/*`
- `/home/pi/adasisv3_provider/trk/*`
- `/home/pi/adasisv3_provider/udp/*`
- `/home/pi/map_cache/*`
- `/home/pi/map_cert/*`
- `/home/pi/adasisv3_provider/adasisv3_provider_conf.json` – Keys for online maps

A full software update will also replace all user-added maps on ADASISv2, so if you have added map database or keystore files, remember to back up them.

Refer to chapter 4 on how to access the Raspberry Pi file system.

10.2.2 Performing a full software update

To perform a full software update, flash the EB robinos Predictor Eval Kit image to the Raspberry Pi device. In figure 30, Win32 Disk Imager is opened, and the image file and correct memory drive are selected. Once **Write** is pressed, the selected image file will be written to the card. After that, the SD card can be placed into the Raspberry Pi card slot and after restarting the device, the updated software will start to run and is ready to use. In case you saved individual files before the full software update, copy these files back to the Raspberry Pi's file system. Note that some updates might introduce changes to configuration files. In that case, direct replacing is not possible, but the stored configuration can be referred to when starting any operating mode after update.

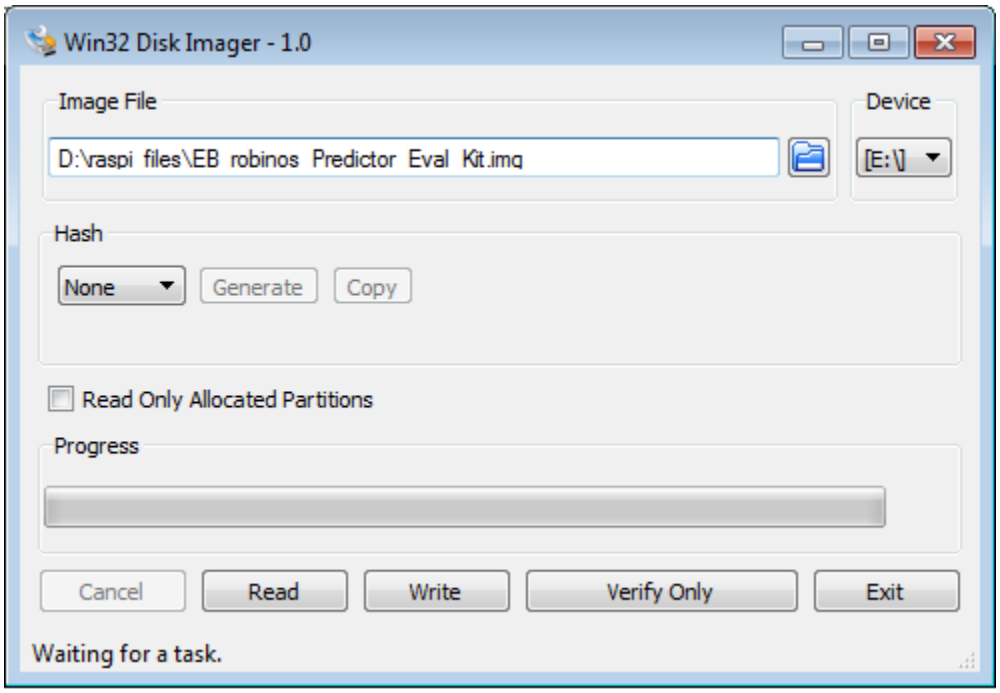


Figure 3020: Win32 Disk Imager

10.3 Activation keys

The ADASISv2/ADASISv3 Provider and ADASISv2/ADASISv3 Reconstructor software is configured to run only with a Raspberry Pi whose serial number is provided to Elektrobit and based on it, unique activation keys are generated. If the software is not yet activated, the view in figure 31 is visible on WebApp. Once valid activation keys have been input, the full functionality of EB robinos Predictor Eval Kit can be utilized. Activation keys need to be input only once.

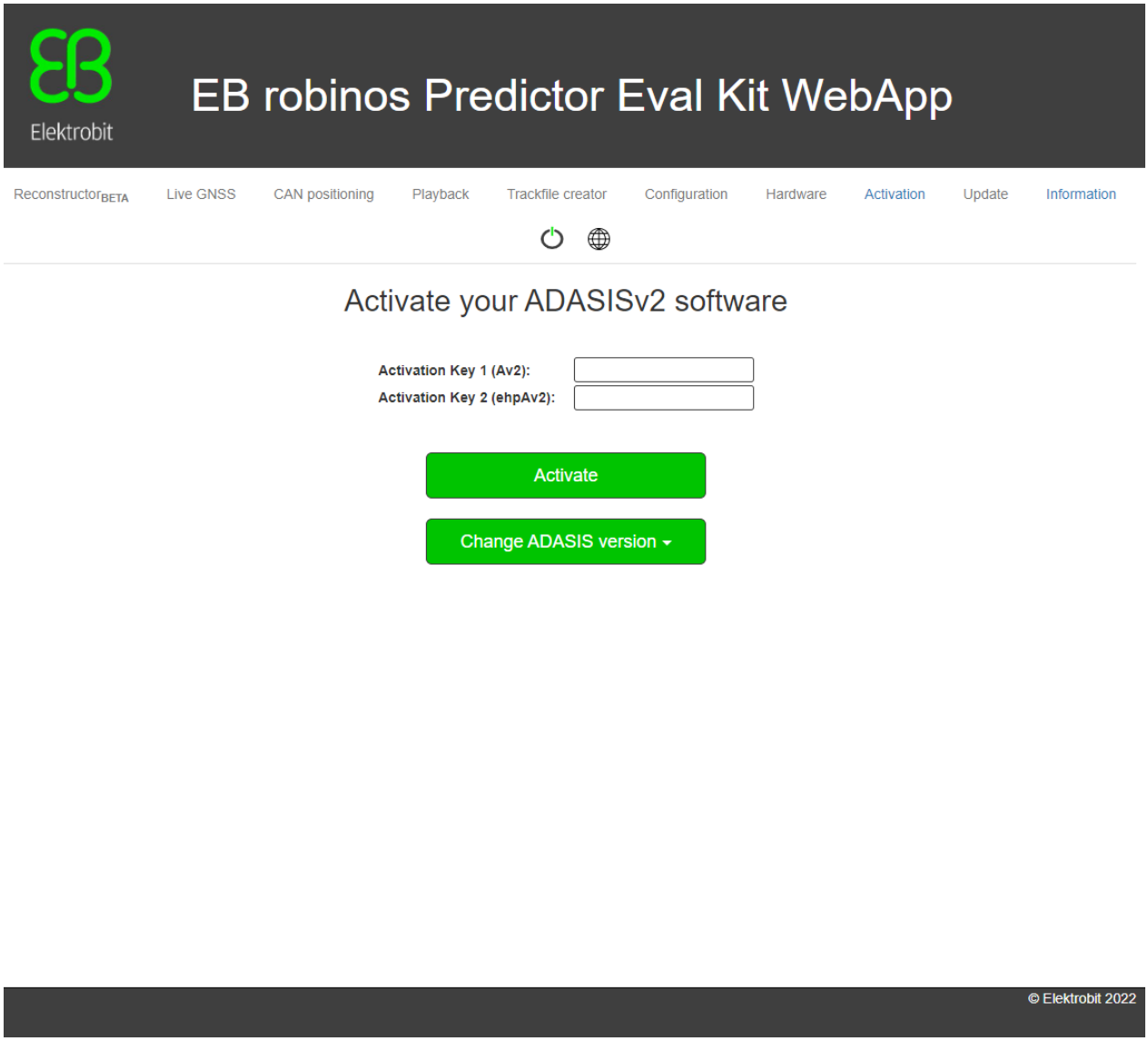


Figure 31: WebApp activation tab

In case you want to receive activation keys for your Raspberry Pi, contact the Elektrobit support team via www.elektrobit.com/support to order them. To find out the Raspberry Pi hardware id and full terms and conditions regarding the usage of EB robinos Predictor Eval Kit, check the **Information** tab (figure 32) of EB robinos Predictor Eval Kit WebApp.

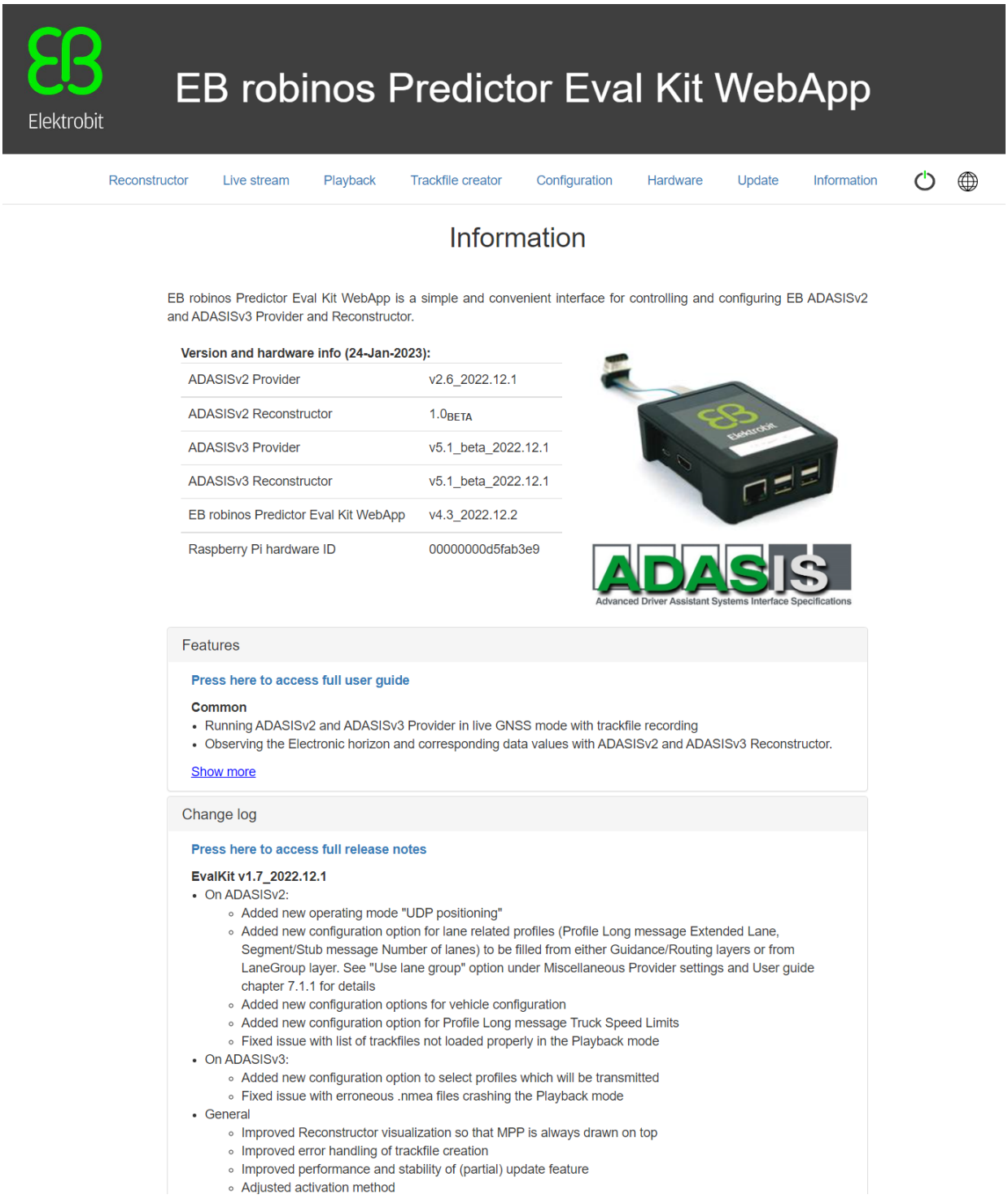


Figure 212: WebApp showing EB robinos Predictor Eval Kit Information tab

11 Support

In case of any questions or issues, you can reach out to our support team through www.elektrobit.com/support. Below chapters have some additional instructions on what you can do already when creating a ticket to receive answers as quickly as possible.

11.1 ADASISv2

In case your problem is related to horizon generation (no horizon output or unexpected data), set "**Provider traces enabled**" ON in configuration tab.

After doing this, rerun the scenario where the issue is seen. After stopping operation, you should see a new nonempty .trc and .trv files generated under /home/pi/adasisv2_provider/tracelogs. Attach the corresponding log files to the ticket.

Additionally, if issue is seen when using any live positioning mode (GPS, CAN, UDP), make sure you have "Save trackfile" option selected on Configuration page. This way, after running the scenario, you should see a new logXXX.trk generated under /home/pi/adasisv2_provider/trk. Also attach this to the ticket. In case you're seeing the problem when doing trackfile playback, attach the trackfile in question to the ticket. All .trk files available for playback in webapp gui can be found in /home/pi/adasisv2_provider/trk.

If the trackfile and/or logs are long, please also write down rough coordinates of the location where you observed the issue at. This ensures that support personnel can find and analyze the correct location. Also, please mention what maps you are using. Currently selected map can be found at the bottom of Configuration page.

11.2 ADASISv3

Similar to ADAISISv2, whenever you're experiencing issues related to horizon generation, first set "**Provider traces enabled**" ON in configuration tab.

After doing this, rerun the scenario where the issue is seen. After stopping operation, you should see a new nonempty .trc and .trv files generated under /home/pi/adasisv2_provider/tracelogs. Attach the corresponding log files to the ticket.

For sharing the input positions, the process of sharing these differs depending on what operation mode is used on ADASISv3. When using GNSS positioning, make sure you have "Save .trk trackfile (GNSS)" enabled on configuration tab. This way .trk file should be generated under /home/pi/adasisv3_provider/trk. Similarly, when using trk playback, all files listed in webapp can be found under that same directory. When using UDP positioning, make sure you have "Save .raw and .ts files (UDP)" enabled on the Configuration page. When this option is enabled, using UDP positioning will generate recordings under /home/pi/adasisv3_provider/udp. When sharing UDP recording, make sure to attach both, .raw and .ts files to the ticket.

As on ADASISv2, in case the attached recordings and logs are long, please also write down the rough coordinates of the problematic location to the support ticket. Please also make sure to mention which map provider you are using: HERE, TomTom, or NDS.Live (also which backend).

12 Glossary

Term/Abbreviation	Description
ADASIS	**A**dvanced **D**river **A**ssistance **S**ystem **I**nterface **S**pecification A consortium of vehicle manufacturers and suppliers that specify electronic horizon protocols which aim to transmit navigation-related information such as map data and car position through vehicle's bus systems. Modern ADAS functions are able to use the electronic horizon to enhance their functionality and availability. See also http://adasis.org/.
ADASISv2	**A**dvanced **D**river **A**ssistance **S**ystem **I**nterface **S**pecification in **V**ersion **2.** Protocol developed by ADASIS Forum to provide electronic horizon data on vehicle's CAN bus.
ADASISv3	**A**dvanced **D**river **A**ssistance **S**ystem **I**nterface **S**pecification in **V**ersion **3.** Protocol developed by ADASIS AISBL to provide electronic horizon data on vehicle's high-speed bus systems (TCP).
CAN	**C**ontroller **A**rea **N**etwork Is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer, see also https://en.wikipedia.org/wiki/CAN_bus.
DBC	**D**ata **B**ase **C**AN Data format to describe/interpret CAN messages and signals exchanged via the CAN bus.
ECU	**E**lectronic **C**ontrol **U**nit
Electronic horizon	An electronic horizon describes the road geometry and attributes ahead of a vehicle based on the vehicle's position and digital navigation map. It changes with the car movement and always represents the full street network the car could take in the intermediate future.
GGA	NMEA 0183 message type. Includes time-, position-, and fix-related data.
GSA	NMEA 0183 message type. Includes data about GPS DOP and active satellites.
HAD	**H**ighly **A**utomated **D**riving
MPP	**M**ost **P**robable **P**ath Defines the main path of the electronic horizon. First-level sub-paths always start at the main path (electronic horizon tree).

NMEA 0183	Data and electrical specification by **N**ational **M**arine **E**lectronics **A**ssociation for communication between different electronics such as GPS receivers, autopilot, gyrocompass, etc.
PSF	**P**hysical **S**torage **F**ormat (map database) Elektrobit's compiled map format that holds all basic and ADAS attributes.
Raspberry Pi	The Raspberry Pi is a credit-card-sized computer (ECU) which can be used in electronics projects. For more information on the ECU, refer to https://www.raspberrypi.org/.
RMC	NMEA 0183 message type. Includes position, velocity, and time data.
TCP	**T**ransmission **C**ontrol **P**rotocol
UDP	**U**ser **D**atagram **P**rotocol

13 Hardware component list

No.	Part	Details
1	Micro SD card	Class 10 micro SD card (minimum 16 GB for the map database and Elektrobit application software)
2a	Power supply minimum 2 A current for office usage	
2b	Power supply minimum 2 A current for in-vehicle usage	
3	PEAK PCAN USB Connector	http://www.peak-system.com/PCAN-USB.199.0.html?&L=1
4	Micro USB cable 2 m	**Optional**
5	GNSS receiver	NAVILOCK NL-8012U GPS USB receiver **Device:** `/dev/ttyACM` **Port:** 0 **Baud rate:** 4800 Other GNSS receivers might work as well, functionality verified with this setup.
6	Raspberry Pi 3, Model B/B+	
7	SUB-D connector (male, 9-pin)	
8	SUB-D connector, 120 Ohm CAN terminator resistor (9-pin)	
9	SUB-D gender changer (female-female, 9-pin)	

10	USB/Ethernet connector to establish an FTP connection to EB robinos Predictor Eval Kit (relevant for application setting changes)	**Optional, only needed in case Wi-Fi not possible**, TU2-ET100 or similar.
11	Ethernet cable to connect the computer via the USB/Ethernet connector with the Raspberry Pi	**Optional, only needed in case Wi-Fi not possible.**

Figure 33: Dimensions of EB robinos Predictor Eval Kit and pin layout of the SUB-D9 connector

Referenced documents

No.	Document	Version	Description
[1]	NDS filling specification	V2.0	Describes mandatory filling of an NDS database that conforms with EB robinos Predictor Eval Kit ADASISv2.
[2]	`200v2.0.4-D2.2-ADASIS_v2_Specification.pdf`	200v2.0.4-D2.2	Describes all ADASISv2 CAN messages and signals.

Table 12: Referenced documents