FI✕POSITION

Vision-RTK 2

Fixposition Positioning Sensor

INTEGRATION MANUAL
VERSION 2.1.0

Abstract: This document explains how to integrate the Vision-RTK 2 positioning sensor into a host system and provides comprehensive details on how to configure it to obtain the maximum positioning accuracy.

www.fixposition.com
support@fixposition.com

precise global positioning everywhere

Document information

Title	Vision-RTK 2
Subtitle	Fixposition Vision-RTK 2 Positioning Sensor
Document type	Integration manual
Version number	2.1.0
Published Date	June 21, 2023
Disclosure restriction	Confidential/NDA
Product status	Engineering sample
Content status	Data based on early testing. Revised and supplementary data will be published later

Glossary

ASCII	American Standard Code for Information Interchange
APC	Antenna Phase Center
ARP	Antenna Reference Point
CAN	Controlled Area Network
DHCP	Dynamic Host Configuration Protocol
ECEF	Earth-Centered Earth-Fixed
ENU	East-North-Up
GLONASS	Global Navigation Satellite System (Russian)
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
DOP	Dilution of Precision
FP	Fixposition
IMU	Inertial Measurement Unit
IP	Internet Protocol
LLH	Latitude Longitude Height
NMEA	National Marine Electronics Association
NTRIP	Networked Transport of RTCM via Internet Protocol
OBD-2	On-board Diagnostic 2
ROS	Robot Operating System
RTK	Real Time Kinematics
RTKLIB	Real Time Kinematics Library
RTCM	Radio Technical Commission for Maritime Services
SMA	SubMiniature version A
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver-Transmitter
UAV	Unmanned Aerial Vehicle
USB	Universal Serial Bus
UTC	Universal Time Coordinate
VRS	Virtual Reference Station
VRTK2	Vision-RTK 2
WGS-84	World Geodetic System 1984

Contents

Vision-RTK 2 Notices

The following notices apply to Vision-RTK 2. Changes or modifications to this equipment, not expressly approved by Fixposition AG., could void the user's authority to operate this equipment.

1.1. European Union

Fixposition AG. hereby declares that Vision-RTK 2 is in compliance with the essential requirements and other relevant provisions of the 2014/30/EU Directive and UN/ECE Regulation 10.

Vision-RTK 2 Wi-Fi
Fixposition AG. declares that Vision-RTK 2 is in compliance with:

1. EU Directive 2014/53/EU.

The full text of the EU Declaration of Conformity may be obtained from the Fixposition website.

Radio Information

- Description of Service: Wi-Fi (802.11b/g/n).

- Operational Frequency: 2400 MHz to 2480 MHz.

- Modulation: OFDM.

- Rated Power: 17.5 dBm e.i.r.p.

The full text of the EU Declaration of Conformity may be obtained from the Fixposition website.

WEEE Notice
If you purchased your Vision-RTK 2 product in Europe, please return it to your dealer or supplier at the end of its life. The objectives of Fixposition's environment policy are, in particular, to preserve, protect and improve the quality of the environment, protect human health, and utilize natural resources prudently and rationally. Sustainable development advocates the reduction of wasteful consumption of natural resources and the prevention of pollution. Waste electrical and electronic equipment (WEEE) is a regulated area. Where waste generation cannot be avoided, it should be reused or recovered for its material or energy. WEEE products may be recognized by their wheeled bin label.

RoHS

The Vision-RTK 2 is in conformity with:

1. Directive 2011/65/EU of the European Parliament and of the Council of 8 June 2011 on restricting the use of certain hazardous substances in electrical and electronic equipment.

2. The Restriction of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment Regulations 2012 (S.I. 2012/3032).

1.2. FCC

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation. Vision-RTK 2 has been tested and found to comply with the radiated and conducted emission limits for a Class B digital device. The Class B limits are designed to provide reasonable protection against harmful interference in a residential installation. The equipment listed generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Re-orient or relocate the Vision-RTK 2.
- Increase the separation between the equipment and the Vision-RTK 2.
- Connect the equipment to an outlet on a circuit different from that to which the Vision-RTK 2 is connected.
- Consult the dealer or an experienced radio/TV technician for help.

To maintain compliance with the limits of a Class B digital device, you must use shielded interface cables.

The Vision-RTK 2 has been authorized for use in Mobile applications. At least 20 cm (8 in) of separation between the Vision-RTK 2 and the User must be maintained at all times.

Wi-Fi

The Vision-RTK 2 contains a Wi-Fi radio with the following approvals:

- FCC ID: TFB-1004

1.3. Japan

Wi-Fi

Vision-RTK 2 contains a Wi-Fi radio with the following approvals:

R 201-180720

Remarks and observations

The following conditions are applicable:

- Antennas for IEEE 802.11a/b/g/n/ac.
- Listed are 2 Dipoles, 1 PIFA, and 1 Chip antenna with a max antenna peak gain of 2.5 dBi at 2.4 GHz and 4.0 dBi at 5.0 GHz.

System description

2.1. Overview

The Vision-RTK 2 is a high-end sensor fusion solution that provides real-time high-accuracy pose information in all scenarios, including GNSS degraded and denied environments.

The system includes two multi-frequency Real-Time Kinematics (RTK) GNSS receivers for instant heading calculation after initialization, an embedded camera, and an Inertial Motion Unit (IMU) to provide continuous high-accuracy positioning even in extended GNSS outages.

The system can also receive information from additional auxiliary sensors (such as wheelspeed) to increase performance whenever required.

(a) Agriculture (b) Urban (c) Lawn mowing (d) Mining
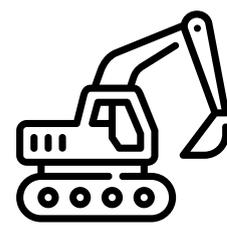
The Vision-RTK 2 is the ideal solution for applications in robotics, precision agriculture, autonomous driving, infrastructure mapping, outdoor industrial applications, and many others.

2.2. High-precision sensor fusion

Vision-RTK 2 provides accurate and reliable position, velocity, and orientation measurements by fusing visual odometry with several sensor data streams, including in extended GNSS outages.

2.3. Supported dynamics models

Vision-RTK 2 supports several navigation modes to adjust to different platforms' specific dynamics. These modes capture the data streams from the selected vehicle type and tune the sensor fusion algorithm to improve performance based on the platform restrictions. Table 2.1 summarizes the characteristics of each tuning mode.

Mode	Application	v range	ω range
Car	Dynamics similar to that of a passenger car.	±22 m/s	0.5 rad/s
Handheld	Dynamics similar to that of a human gait.	±3 m/s	1.5 rad/s
Lawnmower	Dynamics similar to that of a lawnmower robot.	±3 m/s	1.0 rad/s
Slow robot	Dynamics similar to that of a slow-moving utility robot.	±3 m/s	0.5 rad/s

Table 2.1.: Supported sensor fusion modes

·v: Velocity
·ω: Angular velocity

Enabling a wheelspeed sensor is recommended for optimal performance. All tuning modes except handheld support wheelspeed. Note that the above v and ω values correspond to the expected dynamics for these vehicles. However, even though the sensor does not strictly enforce these ranges, exceeding these values can degrade performance.

Technical Information

3.1. System indicators

The Vision-RTK 2 has three LEDs to indicate the current status of the sensor.

- **Green:** System running
- **Amber:** System activity (intermittent blinking)
- **Red:** Not in use

3.2. Physical connectors

3.2.1. Connectors overview
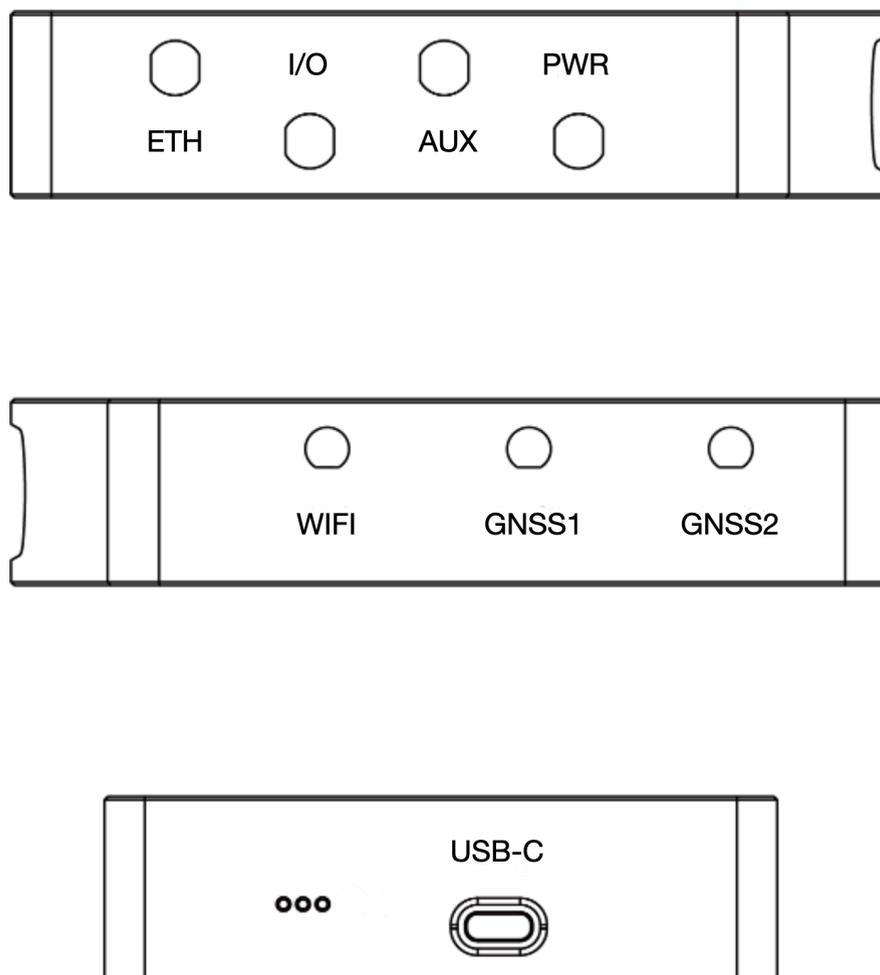
Figure 3.1.: Vision-RTK 2 connectors overview

3.2.2. Ethernet
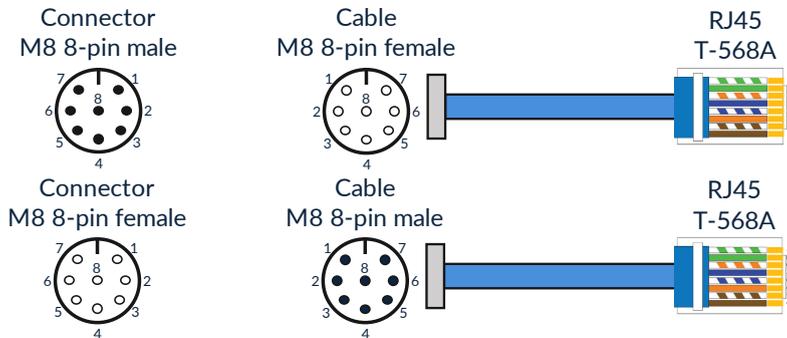
Figure 3.2.: Ethernet connector standard

Note: *The Vision-RTK 2 exists in two variants. Variants manufactured before October 2022 were equipped with male connectors on the sensor, and future sensors will contain female connectors.*

3.2.3. I/O connector
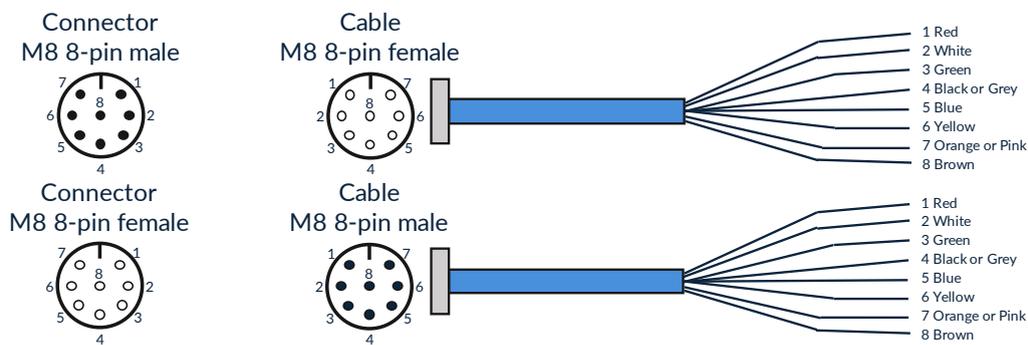
Figure 3.3.: I/O connector pin overview

Note: *The Vision-RTK 2 exists in two variants. Variants manufactured before October 2022 were equipped with male connectors on the sensor, and future sensors will contain female connectors.*

Pin	Wire color	Symbol	Description
1	Red	CANH	CAN High[1]
2	White	CANL	CAN Low
3	Green	PWR_SHDN	Power shutdown[2]
4	Black or Grey	GND	Signal ground
5	Blue	TM_PLS	GNSS1 receiver time pulse
6	Yellow	TM_MRK	GNSS1 receiver time mark
7	Orange or Pink	UART1_RX	UART receiver input 1
8	Brown	UART1_TX	UART transmitter output 1

Table 3.1.: I/O connector pin definition

---

[1] The user must ensure a termination resistor is on the CAN bus. The cable must be at most 30 cm. If you require a longer cable, you must provide a setup that ensures the integrity of the communication.

[2] The PWR_SHDN pin should be driven low when using the I/O connector. To shut down the sensor, drive this pin high (~3V). The sensor will only restart when the pin is no longer high. This behavior applies to all sensors with version number 1B or later (Refer to the label on the sensor). For sensors with no version number or version number 1A, the pin must be high during operation (~3V) and low to shut down the sensor.

3.2.4. AUX connector
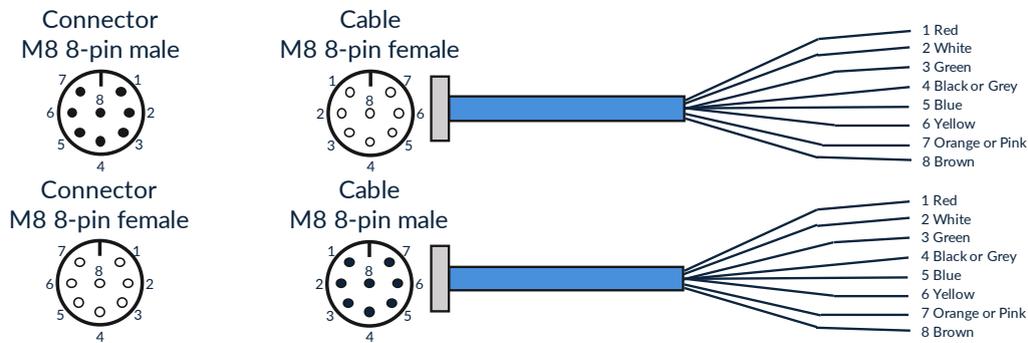
Figure 3.4.: AUX connector pin overview

Note: *The Vision-RTK 2 exists in two variants. Variants manufactured before October 2022 were equipped with male connectors on the sensor, and future sensors will contain female connectors.*

Pin	Wire color	Symbol	Description
1	Red	+5V	Voltage supply output of 5V
2	White	UART2_RX	UART receiver input 2
3	Green	UART2_TX	UART transmitter output 2
4	Black or Grey	GND	Signal ground
5	Blue	Reserved	Reserved
6	Yellow	Reserved	Reserved
7	Orange or Pink	Reserved	Reserved
8	Brown	Reserved	Reserved

Table 3.2.: Aux connector pin definition

3.2.5. Power connector
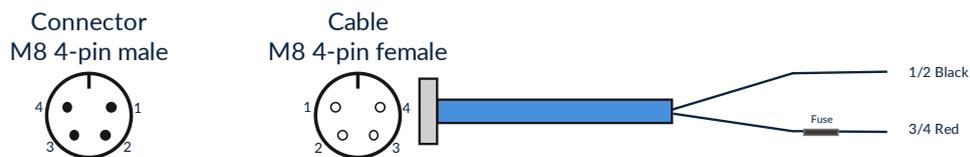
Figure 3.5.: Power connector pin overview

Pin	Wire color	Symbol	Description
1/2	Black	GND	Power ground
3/4	Red	VCC	Main power input

Table 3.3.: Power connector pin definition

3.2.6. GNSS connector

The Vision-RTK 2 contains two GNSS receivers that can connect to an antenna via the female SMA connectors labeled GNSS 1 and GNSS 2.

3.2.7. Wi-Fi connector

The Vision-RTK 2 can significantly increase its Wi-Fi range by connecting a Wi-Fi antenna to the female RP-SMA connector labeled Wi-Fi.

3.2.8. USB (Type-C)

The Vision-RTK 2 contains a USB-C port to connect an external drive for data recording (see 5.10). After FW 2.63, the user can also use this port to access the recovery mode. (see 5.2.5).

3.2.9. Sensor frame

The origin of the sensor's reference frame is on top of the Fixposition logo. Figure. 3.6 presents the sensor's reference frame. Note that all the functionalities and messages of the Vision-RTK 2 employ this reference frame by default.
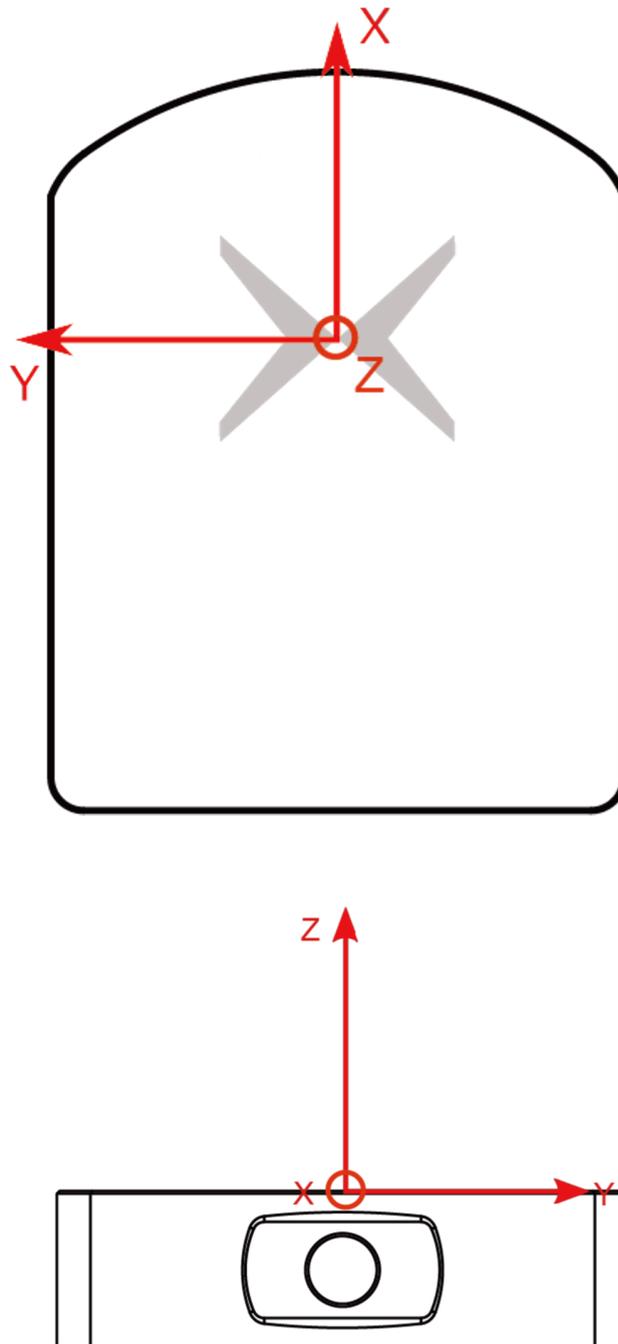
Figure 3.6.: Reference frame of the Vision-RTK 2

3.2.10. Fixposition logo position

Figure 3.7 provides a visual representation of the Fixposition logo's position relative to the boundaries of the sensor's housing. This reference can help measure the extrinsic parameters between Vision-RTK 2 and other components (e.g., GNSS antennas, wheel-speed sensor), especially in cases where the logo is not accessible.

Users can request a STEP file of the sensor's housing to enhance the integration of the Vision-RTK 2 into their platform. This file provides a detailed and comprehensive representation of the sensor's physical design, allowing for seamless integration.
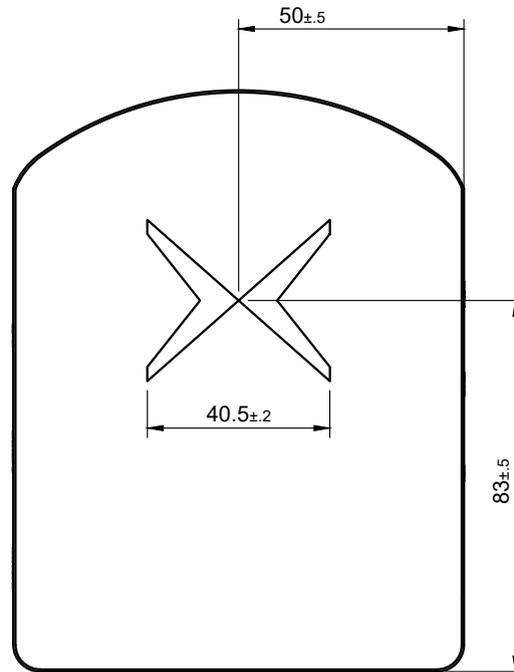
Figure 3.7.: Fixposition logo position on the Vision-RTK 2

3.2.11. Enclosure measurement
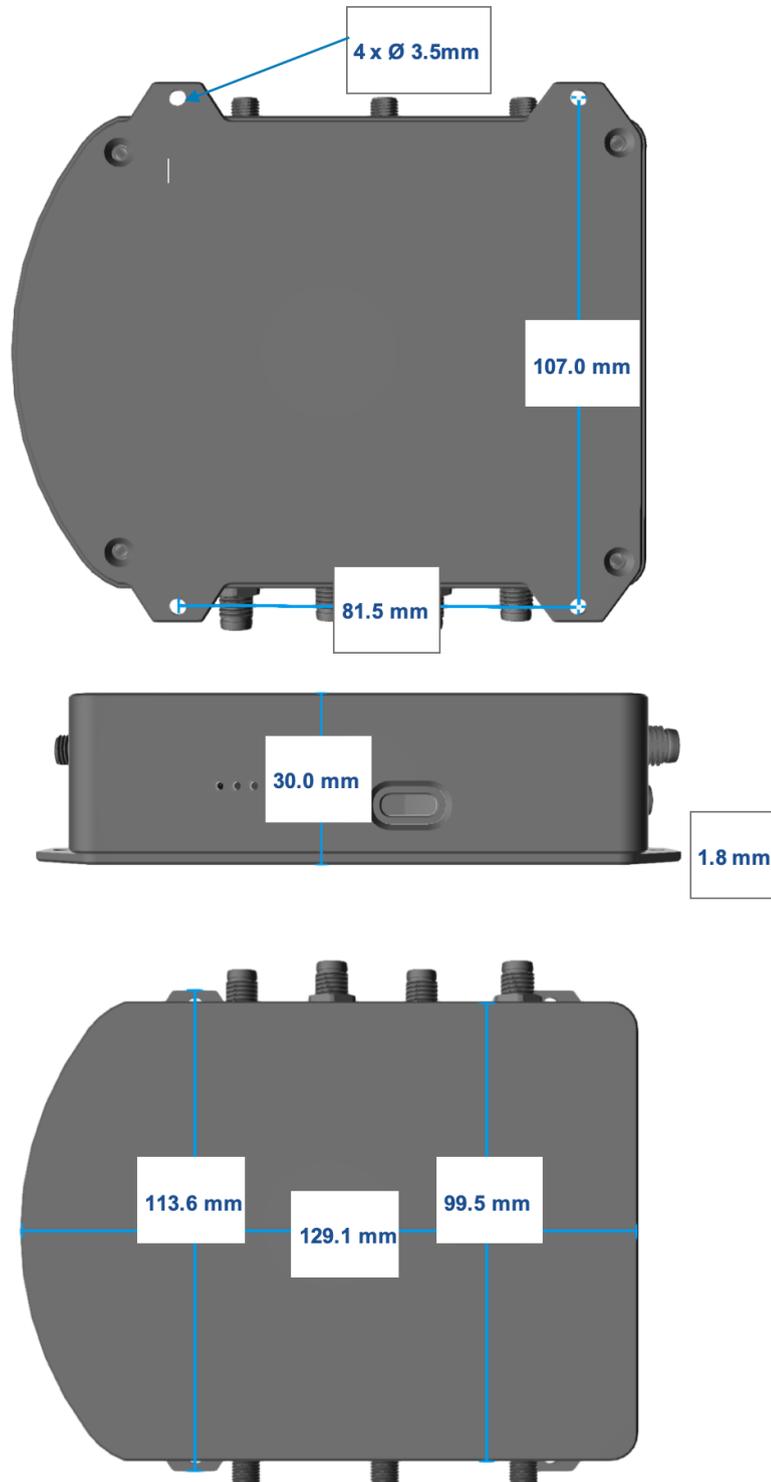
Figure 3.8.: Dimensions of the Vision-RTK 2's enclosure

Installation Guidelines

4.1. Sensor setup requirements

For the proper functioning and optimal performance of the Vision-RTK 2, the user must fulfill the following setup requirements:

- Mount the Vision-RTK 2 firmly/rigidly to the vehicle's body.

- Attach the GNSS antennas to the same rigid body as the VRTK, their relative positions must remain unchanged. Please carefully follow the antenna manufacturer's installation guidelines and comply with their requirements to minimize interference.

- Mount the GNSS antennas at least 20cm apart, ideally, more than 50cm apart.

- Measure the sensor-to-antenna distances with millimeter accuracy (See Figure 4.1). Note that the antenna reference point corresponds to its phase center, described in the corresponding antenna datasheet. The default extrinsic parameters correspond to the measurements of the starter kit.
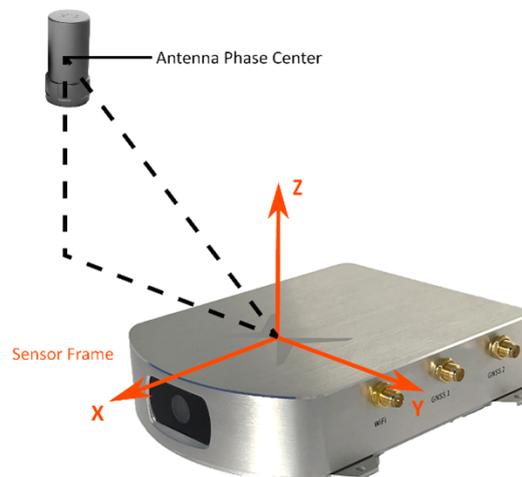
Figure 4.1.: Sensor-to-antenna distance example

- Minimize vibrations experienced by the sensor (e.g., engine, gravel).

- Use shielded cables only and avoid sources of electromagnetic interference near the antennas (e.g., LiDAR, USB3).

- Ensure that the GNSS antennas have an unoccluded view of the sky.

- Employ an external lighting source if working in dim scenarios.

- Reduce static obstructions (e.g., vehicle's structure) and featureless scenes (e.g., sky, mud) in the camera view (See Figure 4.2).

(a) Ground robot: The horizon line lies in the middle of the camera frame.

(b) Car: The hood covers almost half of the frame, preventing feature acquisition.

Figure 4.2.: Obstructions and featureless scenes in the camera view

4.2. Other considerations

- Use a power source between 10-24 V (min 4.5V and max 40V).

- Employ high-quality coaxial cables with minimal signal attenuation and delay with a male SMA connector.

- Select antennas and a correction service that supports L1 and L2 bands for as many satellite constellations as possible (See Appendix E).

- wheelspeed can improve performance in GNSS outages; however, excessive slippage may be detrimental. Assess whether incorporating this data benefits you.

- Consider the camera FOV data when integrating the sensor (See Appendix D).

- The Vision-RTK 2's performance is not affected by whether the sensor is facing backward or forward in the direction of movement.

- Under ideal conditions, the high-precision GNSS receivers employed by the Vision-RTK2 can deliver accuracy down to the centimeter level: 0.01 m + 1 ppm circular error probable (CEP) - measured using a 1 km baseline and patch antennas with good ground planes. Thus, if the sensor connects to a base station located 20 km away, the receiver can provide an accuracy of approximately 3 cm. It is worth noting that the degradation rate increases significantly for distances longer than 20 km.

4.3. Maintenance procedure

To ensure the long-lasting adequate performance of the sensor, the user must periodically perform the following steps:

- Clean the camera lens from any obstructions.

- Verify the integrity of all cables.

- Tighten all connections to the sensor.

- Ensure the sensor and the GNSS antennas are firmly attached to the structure and rigid with respect to each other.

Sensor Configuration

5.1. Web interface overview

To configure the Vision-RTK 2 and visualize the current trajectory, the user can access the web interface through a browser using the IP 10.0.1.1 with a direct Wi-Fi connection or 10.0.2.1 with a direct Ethernet connection. These IP addresses only apply when no intermediary device, such as a router, is in use. Figure 5.1 presents an overview of the web interface content. In the Wi-Fi case, the SSID is the UID of the Vision-RTK 2, and the default password is "1234567890".

Status

Fusion
• Fusion status
• State estimate

GNSS
• Receivers
• Correction data

Configuration

Fusion
• Settings

GNSS
• Correction data
• Advanced

Network
• Wi-Fi client
• Wi-Fi access point
• Ethernet

I/O
• CAN interface
• I/O ports
• Output generators
• Output messages

Camera
• Camera image
• Record calibration sequence
• Camera calibration

Wheelspeed
• Wheelspeed sensors

System

Logs
• Record logs
• Calibration sequences

Update
• Software update
• Message

Info
• General
• Status
• I/O status

Tools
• Reset configuration
• Factory reset
• Webinteface password
• Map access token

Figure 5.1.: Web interface structure overview

The home page of the web interface is shown in Figure 5.2. The user can access the different configuration options through the navigation bar at the top of the screen, which also contains status indicators to monitor the sensor. Please refer to Subsection 5.14 for more information about the status indicators.

Figure 5.2.: Home page of the web interface

5.2. Network configuration

To access the network configuration page in the web interface, the user must press "Configuration" on the navigation bar and select "Network". As shown in Figure 5.3, this page presents the configuration for the Wi-Fi interface, client, and access point, as well as the Ethernet connection configuration.

Figure 5.3.: Networking configuration page in the web interface

5.2.1. Network specifications

To connect to a Wi-Fi network and access the Internet, the user must specify the configuration of the Wi-Fi interface, which includes the desired Wi-Fi band and whether the access point of the sensor is enabled. Currently, the sensor supports the 2.4 and 5.0 GHz bands or disabling the Wi-Fi module altogether (see Figure 5.4). After updating the configuration of the Wi-Fi interface, it is necessary to reboot the sensor to apply them. Currently, we only support one band at a time, meaning the access point and the hotspot should operate at the same frequency.
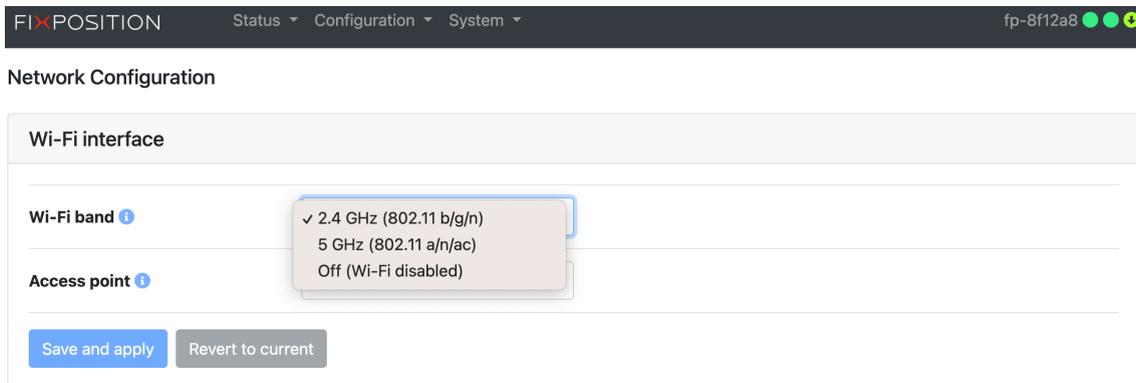
Figure 5.4.: Wi-Fi interface configuration options

The table below summarizes all available network configurations for the Vision-RTK 2:

Network	Sensor IP	Netmask	Client IP	Configuration
Wi-Fi client	Dynamic	Dynamic	n/a	Web interface
Wi-Fi access point	10.0.1.1	255.255.255.0	10.0.1.10–254	Enabled by default
Ethernet DHCP server	10.0.2.1	255.255.255.0	10.0.1.10–254	Enabled by default
Ethernet DHCP client	Dynamic	Dynamic	n/a	Web interface
Ethernet static	Variable	Variable	n/a	Web interface
USB	10.0.3.1	255.255.255.0	10.0.3.10-254	For recovery only

Table 5.1.: Network configuration details

Some details, recommendations, and known limitations to take into consideration:

- Even though wireless connections are supported, Ethernet is recommended for use in the field, as the Wi-Fi protocol does not guarantee stability.

- 2.4 GHz is preferred over 5 GHz for Wi-Fi connections due to its range and reliability.

- An Ethernet connection is always prioritized over Wi-Fi for Internet access.

- If the Ethernet is configured as static-ip and it doesn't provide Internet access, make sure to leave the gateway field empty. Only then the Wi-Fi connection will be used for Internet access.

- The only supported Wi-Fi security configuration is "wpa-psk" (WPA2). Currently, we do not support "SAE" (WPA3) nor "open" Wi-Fi connections (no password).

- The Vision-RTK2 shares Internet access with devices connected to its network.

- When employing an intermediary device, such as a router or an access point, the DHCP server assigns a dynamic IP address to the sensor. The 10.0.1.1 and 10.0.2.1 IPs do not apply when accessing the sensor via the intermediary device.

- Use distinct SSIDs for each Wi-Fi band (2.4 and 5 GHz) to facilitate connectivity.

- The Vision-RTK2 will only connect automatically to the network marked as "default" in the web interface. However, marking a new Wi-Fi connection as "default" does not activate this connection automatically; the user must press the "connect" button.

- The IP address, netmask, gateway, and DNS of each connection can be configured.

- Wi-Fi client connections are only partially reliable; they sometimes require multiple attempts to succeed in connecting.

- If the Wi-Fi client connection is lost, reconnecting can take up to two minutes.

- The network SSID must match the regular expression ^[-_a-zA-Z0-9]{2,32}$. In other words, the network SSID must contain only alphanumeric characters.

- The Wi-Fi chip regulatory domain is configured as "passive automatic" (i.e., the chip monitors the existing Wi-Fi channels and uses only those valid in the region).

5.2.2. Network data ports

The following network data ports are available over Ethernet and Wi-Fi:

Port	Protocol	Clients	Function
21000-21004	Raw TCP/IP socket	Multiple	VRTK data output ($FP messages) and input (wheelspeed, RTCM3)
23010	Raw TCP/IP socket	Multiple	NTRIP data stream (RTCM3 messsages from the NTRIP caster)
80	HTTP	Multiple	Web interface (incl. sw update)
123	UDP	Multiple	NTP time server (Network Time Protocol)
n/a	ICMP	n/a	ICMP traffic (ping, etc.)
53	UDP	Multiple	Domain (DNS) service for internet sharing

Table 5.2.: Network data ports

Notes:

- The NTP server only provides timing information if the VRTK2 sensor can access precise time from Internet servers or GNSS.

- Only one simultaneous input stream can be handled correctly on TCP port numbers 21000 – 21004. For example, to input RTCM3 and wheelspeed via the network, use one port (e.g., 21000) for wheelspeed and another (e.g., 21001) for corrections.

- The Vision-RTK2 rejects connection attempts to any other port (either blocked by the firewall or there is no service on that port).

5.2.3. Outbound connections

The Vision-RTK2 makes outbound connections to the Internet to the following servers:

- 0.openembedded.pool.ntp.org – NTP time sync.

- NTRIP caster server.

While the web interface is open in a client browser:

- 8.8.8.8 – Checking internet connectivity.

The web interface (i.e., the client browser, not the VRTK2 itself) connects to:

- api.mapbox.com - Map data used on the Fusion status page.

5.2.4. Camera streaming

It is possible to stream the camera image from the following URL:

- `http://x.x.x.x/api/v2/camera/stream` (ca. 2 MiB/s).

Some notes and warnings:

- Each running stream costs CPU resources and may affect the sensor's performance; use with consideration.

- The stream format is compatible with commercial software like Mozilla Firefox, Google Chrome, ffmpeg, and VLC. The transport is HTTP, and the stream format is MJEPG.

- The frame rate is limited (approx. 4 fps).

- The camera stream is only for debugging and development (e.g., checking camera alignment). **Operational and continuous use is not supported.**

- The camera's intrinsic data is available to users upon request. The reference camera model is based on the OpenCV documentation **OpenCV Fish-eye Camera Calibration** `https://docs.opencv.org/3.4/db/d58/group__calib3d__fisheye.html#details`.

5.2.5. USB recovery network

When connected to a PC, the USB port on the Vision-RTK2 acts as a "USB Ethernet gadget." The PC sees a network interface similar to a USB-to-ethernet dongle.

The PC should automatically detect the network interface and configure it. This way, the user can access the web interface via http://10.0.3.1 to change the configuration. The configured password does not protect this access mode to the web interface as with other interfaces (Ethernet and Wi-Fi).

The user cannot configure this network access mode as it is only for recovery (e.g., misconfigured Ethernet and Wi-Fi) and should not use it for data transfer.

The USB network interface should automatically work on Linux (e.g., Ubuntu 22.04). Windows should recognize the sensor as a "USB Ethernet/RNDIS Gadget," not as a (non-functional) COM port.

5.2.6. Time synchronization

To visualize which time source the Vision-RTK2 is employing, head to the "System->Info" page of the web interface and check the 'Time synchronisation' field. An example is shown in Fig. 5.5, where the Vision-RTK2 is synchronised to the time information provided by the GNSS receivers.

The time information can come from one of the following sources:

- Internal clock.

- Internet Time Service (ITS).

- GNSS time information.

On initialization, the time information provided by the Vision-RTK2 is based on its internal clock. If an Internet connection is provided, the sensor will attempt to communicate with an Internet Time Service (ITS) to synchronise its clock. Once the sensor receives GNSS signals, its internal clock will synchronise with the atomic clock of the GNSS satellites, which have the highest accuracy. Note that the Vision-RTK2 has an internal battery to keep its internal clock running without the sensor's power source connected.

Internet connectivity	Active
Time synchronisation	Time synced to GNSS

Figure 5.5.: Time information displayed in the GNSS Status dashboard

To synchronize another system with the very accurate clock of the Vision-RTK 2, there are two options:

1. Use NTP protocol over the network. The Vision-RTK 2 has a built-in NTP server (port 123). You can use that to synchronize your system clock to the very accurate system clock of the Vision-RTK 2.

2. The Vision-RTK 2 outputs the PPS signal from the GNSS1 receiver.

   - Time pulse over the pin 5 of the I/O connector.

   - Time mark over the pin 6 of the I/O connector.
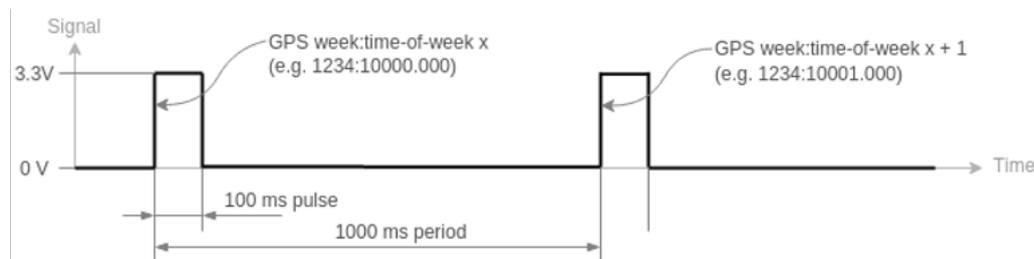
Figure 5.6.: An illustration of time pulse function

Note: The NTP server of the Vision-RTK2 can provide a clock signal accurate up to the sub-millisecond using a direct Ethernet connection, as the time information provided will be based on GNSS data.

Time pulse: The time pulse function provides a one pulse per second (pps) signal. Once the sensor has received any GNSS signals information, the 1 pps signal will adjust to each second, and the rising edge of each pulse will align to the top of GPS time seconds. The duty cycle is 10%, meaning the pulse width is 100 ms. The timestamp message is available in the raw GNSS1 output on port 20010. Some devices require an NMEA-GP-RMC message along with the time pulse. If this is the case, please get in touch with Fixposition directly.

Time mark: The time mark accurately measures the time a pulse is detected on pin 6 and outputs a timestamp message on port 20010. At most, the maximum time mark frequency is 5 Hz.

To read the PPS signal from a Vision-RTK 2, the user must use a microcontroller or a similar device to read these signals. This process involves connecting the PPS output from the Vision-RTK 2 to a GPIO (General Purpose Input/Output) pin on the user's microcontroller and programming it to read the state of this pin once every second.

Here is a high-level procedure:

- **Hardware Setup:** Connect the PPS output pin from the user's Vision-RTK 2 to a GPIO pin on the microcontroller. Depending on the voltage levels used by the Vision-RTK 2, the user may also need to include a voltage level shifter or voltage divider in the circuit to prevent damage to the microcontroller.

- **Software Setup:** Write a program that uses the microcontroller's interrupt capabilities to respond to the rising or falling edge of the PPS signal.

5.3. Input/output system overview
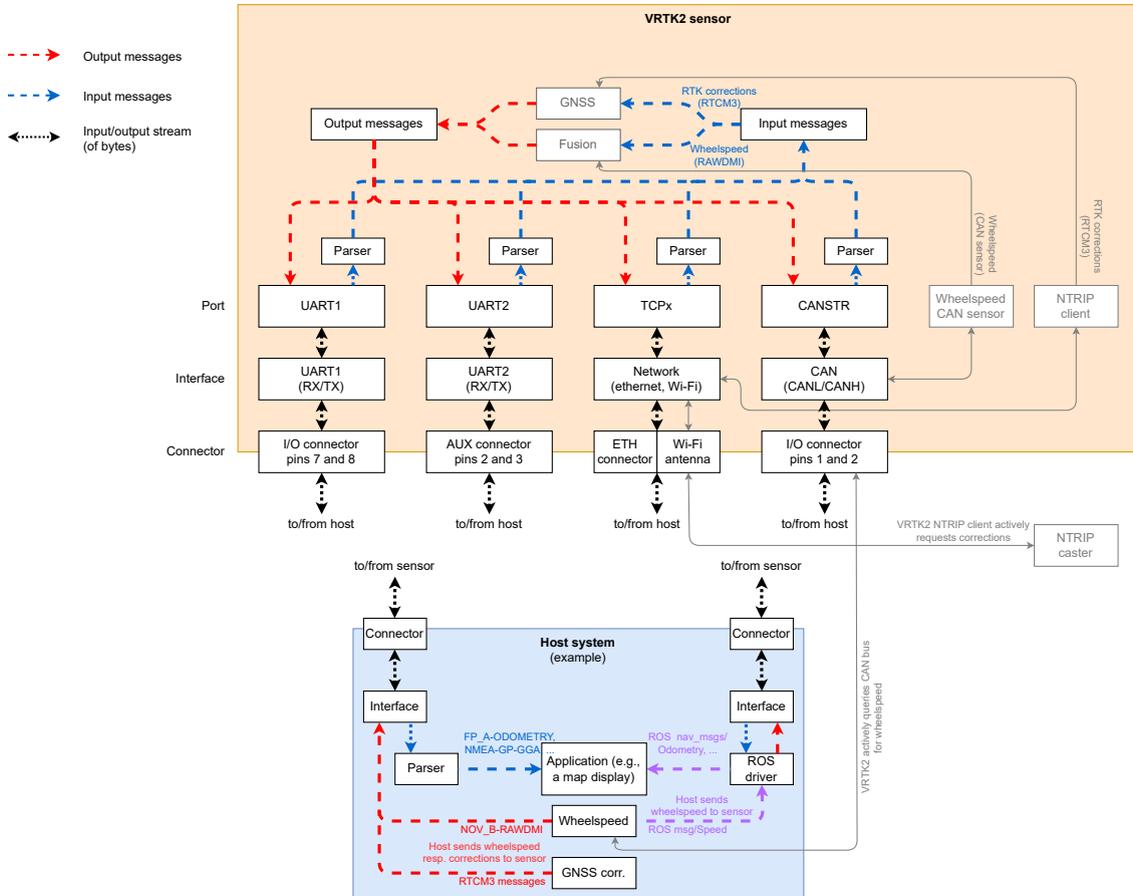
Figure 5.7.: Input/output system overview

The Vision-RTK 2 provides multiple input and output data stream options. This section provides an overview of the I/O system. Note the distinction between:

- **Port**: Physical (e.g., UART) or logical (e.g., network socket) connection endpoint. Via a *connector* and an *interface* this is what the customer ultimately connects to.

- **Interface**: Hardware peripheral which connects a *port* with a *connector*.

- **Connector**: Provides the electrical connection to the *interface* (i.e., the connector names are written on the sensor).

The network interfaces of the sensor comprise Ethernet, Wi-Fi station (client), Wi-Fi access point, and USB network. See Section 5.2 for more details on network configuration.

Port	Interface	Connector	Description
UART1	UART1	I/O	(Asynchronous) serial port for speed up to 1Mbit/s, via a UART1 interface
UART2	UART2	AUX	(Asynchronous) serial port for speed up to 1Mbit/s, via a UART2 interface
TCP0	Network	Eth, Wi-Fi, USB	TCP/IP raw socket server on port number 21000, can handle multiple clients
TCP1	Network	Eth, Wi-Fi, USB	TCP/IP raw socket server on port number 21001, can handle multiple clients
TCP2	Network	Eth, Wi-Fi, USB	TCP/IP raw socket server on port number 21002, can handle multiple clients
TCP3	Network	Eth, Wi-Fi, USB	TCP/IP raw socket server on port number 21003, can handle multiple clients
TCP4	Network	Eth, Wi-Fi, USB	TCP/IP raw socket server on port number 21004, can handle multiple clients
CANSTR	CAN	I/O	CAN "streaming" port. Broadcasts output messages and listens to incoming messages

Table 5.3.: Connector, Interface, Port overview

5.3.1. UART port configuration

The UART1 and UART2 ports correspond to the pins 7 (UART1_RX), 8 (UART1_TX), 4 (GND) in the I/O connector, and pins 2 (UART2_RX), 3 (UART_TX), and 4 (GND) in the AUX connector (see Subsection 3.2 for more information). The user can configure the baud rate of the UART ports inside the "Configuration → I/O" panel of the ieb interface. The available options are 9'600, 19'200, 38'400, 57'600, 115'200, 230'400, 460'800, or 921'600. A baud rate of 115'200 or higher is recommended.

The user can input the wheelspeed sensor and RTCM3 correction data streams into the Vision-RTK 2 via UART (see Section 5.8 and Section 5.4, respectively). The Vision-RTK can stream the output messages (e.g., FP odometry) via UART (see Subsection 5.3.4).

Figure 5.8.: UART baud rate configuration

5.3.2. CAN streaming port configuration

All input and output ports are streams of bytes (uint8_t). The CANSTR port packs the output stream of bytes into as many CAN frames as necessary. Depending on the CANSTR configuration, this can be classical CAN frames with up to 8 bytes or CAN FD frames with up to 64 bytes of payload. Figure 5.9 shows the CANSTR message format generation. The CANSTR port corresponds to the pins 1 (CANH), 2 (CANL), and 4 (GND) of the I/O connector (see Table 3.1).
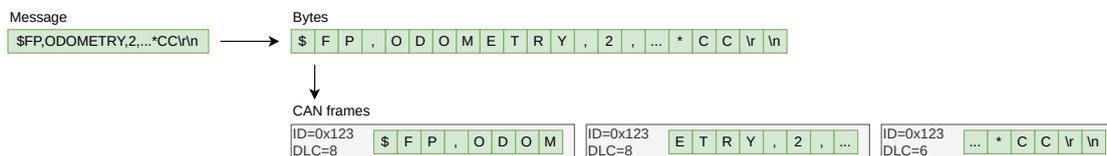
Figure 5.9.: CANSTR message format

The user can configure the CAN interface inside the "Configuration → I/O" panel of the web interface, as shown in Figure 5.10. The following configuration options are available:

- **CAN interface:** The CAN interface can be enabled or disabled.

- **Classical CAN bitrate:** The bitrates [Hz] can be: 10'000, 20'000, 50'000, 125'000, 250'000, 500'000, 800'000 or 1'000'000.

- **CAN FD (BRS) bitrate:** The CAN FD bitrate must be greater or equal to the classical CAN bitrate.

The user must reboot the sensor after updating the CAN interface configuration.

I/O Configuration
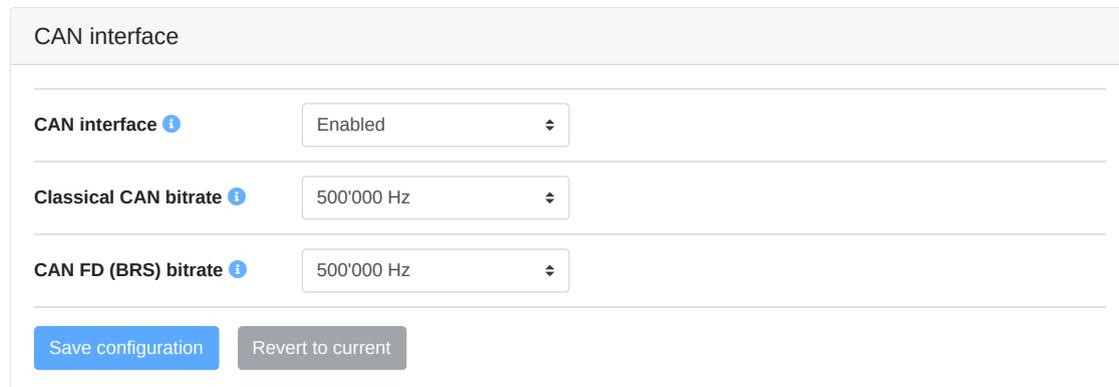
Figure 5.10.: CAN interface configuration

The CANSTR module configuration can be modified in the CANSTR panel inside the "Configuration → I/O" panel of the web interface, as shown in Figure 5.11. The following configuration options are available:

- **CAN ID output:** CAN ID for outgoing CAN frames.

  - Standard frame format (SFF): 11-bit value in the range 0x001 - 0x1ff.

  - Extended frame format (EFF): 29-bit value in the range 0x00000001 - 0x1fffffff.

- **CAN ID input:** CAN ID for incoming CAN frames.

  - Standard frame format (SFF): 11-bit value in the range 0x001 - 0x1ff.

  - Extended frame format (EFF): 29-bit value in the range 0x00000001 - 0x1fffffff.

- **CAN frame format:** CAN frame format used by the connector.

  - Standard frame format (SFF): Use 11-bit CAN IDs for input and output.

  - Extended frame format (EFF): Use 29-bit CAN IDs for input and output.

- **CAN FD:** Classical CAN or CAN flexible data rate (FD)

  - Classical CAN: CAN frames of up to 8 bytes payload.

  - CAN FD: CAN frames of up to 64 bytes of payload.

- **CAN FD BRS:** Enable bitrate switching (BRS) when using CAN FD.

  - Enabled: Use CAN FD BRS.

  - Disabled: Do not use CAN FD BRS.

The CAN streaming only inputs the wheelspeed signals in the current software version. Future releases will introduce more functionalities. Please see 5.8.1 for further configuration of the wheelspeed sensor signals over the CAN bus.

CANSTR	
CAN ID output ⓘ	0x123
CAN ID input ⓘ	0x124
CAN frame format ⓘ	SFF ⌄
CAN FD ⓘ	Classical CAN ⌄
CAN FD BRS ⓘ	Disabled ⌄

Save and apply   Revert to current

Figure 5.11.: CANSTR module configuration

5.3.3. Output generator

The user can configure the Fusion output at the Output generators module inside the "Configuration → I/O" panel of the web interface, as shown in Figure 5.12. The following configuration options are available:

- **Fusion output frequency:** Rate in [Hz] at which the Fusion generates its output. Range: 1-100.

- **Fusion output offset:** Specifies how much time ahead Fusion calculates its output [s], compensating for output or processing latency (at the cost of some accuracy). Range: 0.0-0.2.

- **TF output frequency:** Output frequency in [Hz] for the TF output. Range: 1-10.

- **NMEA output format:** NMEA format used by the Fusion output.

    - Strict NMEA: Strictly follow the NMEA specifications.

    - High-Precision: Enable output of non-standard data fields with more precision.

- **Output Point of Interest configuration:** Parameters to transform the output from the sensor frame to the Point of Interest (POI); see Section 5.13 for more details.

    - Output translation: Translation vector in [m] from the sensor frame to the POI. Range: -100.0-100.0.

    - Output rotation: Rotation from the sensor frame to the POI using ZYX Euler angles (yaw-pitch-roll) in degrees. Range: -180.0-180.0".

Figure 5.12.: Output generators configuration

Note: With High-Precision enabled, the output is no longer NMEA compliant. See the comparison of the two modes in Table 5.4. If you are currently parsing standard NMEA formatted messages, you must update your parser to support the added digits for the "High-Precision" NMEA output provided.

Field	Strict NMEA	High-precision	Notes
Latitude, Longitude (degrees)	ddmm.mmmmm [~ 20 mm]	ddmm.mmmmmmm [~ 0.2 mm]	Approximate, worst case (at the equator)
Height, Altitude (meters)	float(.1) [100 mm]	float(.4) [0.1 mm]	
Time (seconds)	ss.ss [10 ms]	ss.ssss [0.1 ms]	
Speed (knots)	float(.1) [~ 100 mm/s]	float(.5) [~ 0.1 mm/s]	
Heading, Course (degrees)	float(.1) [0.1 deg]	float(.4) [0.0001 deg]	
Number of satellites	00...12	00...99	

Table 5.4.: Strict(standard) and High-precision NMEA format comparison

5.3.4. Output messages configuration

Figure 5.13.: Output message configuration

The user can select which messages are assigned to each port at the Output messages module inside the "Configuration → I/O" panel of the web interface, as shown in Figure 5.13. In this example, the ports UART1 and TCP0 stream out the **FP_A-ODOMETRY** message, and the TCP1 port outputs the **NOV_B-INSPVAX** message. Please refer to Chapter 7 for available output message formats.

In the advanced view, each available message shows its output rate. Zero (0) means the message is disabled and is not output. One (1) means the message is enabled, and its output frequency is 1 Hz. Values greater than one indicate that one of every n-th message is output (e.g., Five (5) means only one of every five messages is output).

While values other than 0 or 1 may not make sense for most messages or scenarios, there are use cases for these values. For example, we can configure the Fusion output to 50 Hz and enable the **FP_A-ODOMETRY** output on TCP0. Enabling this message implies a rate of 1 by default; thus, its output is also at 50 Hz. This high output rate would not be possible on a low-bandwidth port, such as UART. However, the user can select a rate of 10, meaning the sensor would output only every tenth message (i.e., 5 Hz), for which the bandwidth is sufficient.

Note that there is no guarantee for output rates greater than one to be aligned to the top of the second. For example, for a message generated at 10 Hz, with a rate of 5, the sensor cannot guarantee the time stamps to end in x.0 and x.5, as a slight delay could change the timings to x.1 and x.6 or x.4 and x.9.

In Figure 5.14, the output rate is 10 for the **FP_A-ODOMETRY** message on the UART1 port. If the desired output frequency is 200 Hz, the actual output frequency is equal to or smaller than 20 Hz.

$$\text{real output frequency} \leq \text{theoretical output frequency} = \frac{\text{Fusion output frequency}}{\text{output rate}}$$

Output messages								
Fusion output ⓘ	UART1	UART2	TCP0	TCP1	TCP2	TCP3	TCP4	CANSTR
FP_A-ODOMETRY	**10**	0	1	0	0	0	0	0

Figure 5.14.: An example of setting up the output rate

5.4. Correction service (RTK) configuration

The Vision-RTK 2 requires RTCM3 data to provide real-time corrections for accurate localization. The user can configure the correction service in the "Configuration → GNSS" panel of the web interface. The correction service configuration must fulfill the following specifications:

- **Coverage:** The area of operation of the sensor must match the geographic operation area of the correction service provider.

- **Transport:** Networked Transport of RTCM via Internet Protocol (NTRIP), version 1 only. This method implies that the sensor must have access to the Internet and the chosen service/server.

- **Data format:** The supported RCTM3 messages are listed in Subsection 7.2.1.

- **Data content:** OSR-style multi-signal messages (MSM).

- **GNSS constellations:** The data must include all four major GNSS constellations (i.e., GPS, GLONASS, Galileo, and BeiDou). Missing systems will degrade performance significantly.

The user must select the desired method to input the correction service data in the "Source" parameter of the "Correction data" field inside the "Configuration → GNSS" panel of the web interface (see Figure 5.15 for reference). The following methods are available:

- **Built-in NTRIP client:** Connects to the selected NTRIP caster through the Internet. The sensor automatically sends its location to the caster either from (a) the automatic position estimate from GNSS or (b) a manually configured reference position. The user must fill in all NTRIP fields to establish a connection (e.g., user, password, address, and mountpoint). If the NTRIP caster does not use a login, put dummy or none for the user and password fields. The mountpoint field is mandatory.

- **I/O port:** The user streams the corresponding RTCM3 messages at appropriate rates over one of the five TCP ports or one of the two serial ports available on the Vision-RTK 2. Note that the user can visualize the number of bytes, messages, and errors received through any port in the "I/O status" field found in the "System → Info" panel of the web interface. Please refer to Subsection 6.3 for the details.

Note: Any VRS or closest-basestation RTK correction service requires an initial position estimate (single-3D GNSS position) to select or generate the corresponding physical/virtual basestation to be used by the sensor/rover. Thus, the user must head outdoors to establish the connection to the RTK service provider.

FIXPOSITION Status ▾ Configuration ▾ System ▾ fp-8f12a8 ●●⊙

GNSS Configuration

Correction data `connected`

Source ⓘ
- ○ I/O port
- ● NTRIP client
  - [Enter NTRIP path]

User ⓘ   [fixposition]

Password ⓘ   [••••••••] [👁]

Host ⓘ   [ntrip.fixposition.com]

Port ⓘ   [2101]

Mountpoint ⓘ   [FP01]

Position ⓘ
- ● Automatic
- ○ Manual

  Latitude in fractional degrees (-90…90), for example: *47.40020*
  [0]
  Longitude in fractional degrees (-180…180), for example: *8.45036*
  [0]
  Height in meters (0…10000), for example: *395*
  [0]

[Save and apply] [Revert to current]   [Restart NTRIP client]

Advanced

Options ⓘ   [Advanced GNSS options]

Note that changes to the CAN interface configuration are only activated on boot. Reboot the sensor after saving a changed configuration.

[Save] [Revert to current]

Figure 5.15.: RTK configuration page in the web interface

5.4.1. Supported RTCM3 messages

Besides meeting the general requirements for correction service data, the Vision-RTK2 requires that the latency (age) of the data should be kept as low as possible (ideally better than 1 second) and at least the following RTCM3 messages for proper operation:

Type	Message
Reference station position **- Update rate: every 10s or less**	One of the following: - RTCM type 1005 (Stationary RTK reference station ARP) - RTCM type 1006 (Stationary RTK reference station ARP with antenna height)
GPS observables **- Update rate: 1Hz**	One of the following: - RTCM type 1074 (GPS MSM4) - RTCM type 1075 (GPS MSM5) - RTCM type 1077 (GPS MSM7)
Galileo observables **- Update rate: 1Hz**	One of the following: - RTCM type 1094 (Galileo MSM4) - RTCM type 1095 (Galileo MSM5) - RTCM type 1097 (Galileo MSM7)
BeiDou observables **- Update rate: 1Hz**	One of the following: - RTCM type 1124 (BeiDou MSM4) - RTCM type 1125 (BeiDou MSM5) - RTCM type 1127 (BeiDou MSM7)
GLONASS observables **- Update rate: 1Hz**	One of the following: - RTCM type 1084 (GLONASS MSM4) - RTCM type 1085 (GLONASS MSM5) - RTCM type 1087 (GLONASS MSM7)
GLONASS code-phase biases **- Update rate: every 5s or less**	- RTCM type 1230

Table 5.5.: List of required RTCM3 input messages

5.5. Local NTRIP caster

Alternatively, the user can set up a local NTRIP caster to stream correction data to the Vision-RTK 2. In the example below, a host system receives RTCM3 as a serial input and streams them to the Vision-RTK 2 via a network connection using its NTRIP client.
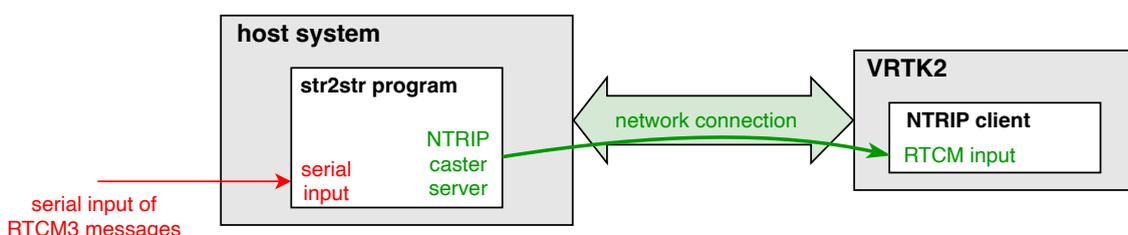
Figure 5.16.: Workflow diagram of the str2str program combined with the Vision-RTK 2

5.6. RTKLIB/str2str

RTKLIB's str2str application has the capacity to divide an incoming data stream into several streams, each with different formats. Users interested in leveraging this functionality on a host system to connect to the NTRIP service, retrieve RTCM3 data, and subsequently relay this correction information to VRTK2 via I/O (UART/TCP), can follow the steps below. These guidelines will cover RTKLIB installation, establishing a connection with the NTRIP service, and streaming data to a designated IP address and port.

1. Update the system packages:

```
sudo apt-get update
```

2. Install necessary tools for building RTKLIB:

```
sudo apt-get install build-essential gcc g++ git
```

3. Clone the RTKLIB repository:

```
git clone git@github.com:rinex20/RTKLIB-demo5.git
```

4. Navigate to the RTKLIB build directory:

```
cd RTKLIB-demo5/app/consapp/str2str/gcc
```

5. Compile RTKLIB:

```
make
```

This process will generate an executable named *str2str* in the RTKLIB build directory.

Within the Vision-RTK 2 web interface, users can navigate to "Configuration → GNSS", then select the "I/O port" option. This will enable users to stream RTCM3 data to any accessible TCP/UART port on the Vision-RTK 2 device.

Some NTRIP services (including certain VRS services) require receivers to reciprocate with NMEA-GP-GGA_GNSS data. This enables the service provider to assign the user a nearby real or virtual base station. Given these circumstances, users also need to configure Vision-RTK 2 to output the NMEA-GP-GGA_GNSS data. For instance, a TCP port (like TCP4) can be set to exclusively output NMEA-GP-GGA_GNSS data at a rate of 10 or less, as depicted in Figure 5.17.

GNSS data ⓘ	UART1	UART2	TCP0	TCP1	TCP2	TCP3	TCP4	CANSTR
NMEA-GP-GGA_GNSS	0	0	0	0	0	0	**10**	0

Figure 5.17.: The set up of the output of NMEA-GP-GGA_GNSS in the I/O section

To establish a connection with the NTRIP service and stream data to a preferred IP address and port, the following command can be implemented:

```
str2str -in ntrip://<username>:<password>@<ntrip\_service\_url>:<port
>/<mountpoint> -out tcpcli://<IP\_address>:<port> -b 1
```

In this command:

- Replace *<username>:<password>* with your NTRIP service username and password.

- Replace *<ntrip_service_url>:<port>/<mountpoint>* with the URL, port, and mountpoint for your NTRIP service.

- The *tcpcli://<IP_address>:<port>* argument instructs str2str to output the RTCM data as a TCP server at the specified IP address and port.

This command initiates data streaming. To halt the stream, press Ctrl+C in the terminal running str2str. Depending on your requirements, these steps may need adjustments. For instance, if you desire the stream to initiate automatically at system startup, consider creating a systemd service.

Note: The str2str utility has the capability to multiplex a single RTK correction source to multiple clients. The correction source can be either a serial port, a raw TCP/IP connection, or an NTRIP client. The str2str caster is designed to handle multiple clients efficiently. However, it is important to note that a VRS-type correction source is only compatible with this version here: https://github.com/rinex20/RTKLIB-demo5. Other forks or the str2str installed from Ubuntu's apt-get might not work with VRS correction.

For additional information, refer to https://rtkexplorer.com/pdfs/manual_demo5.pdf.

5.7. Camera configuration

The user can visualize the camera stream in the "Configuration → Camera" panel of the web interface. A distorted and downsampled (both in resolution and frame rate) stream of the camera is presented here. As Chapter 4.1 mentions, no static parts or featureless scenes must be present in the image view, as these would affect the sensor's performance. The user can use the cropping tools on the page to eliminate these featureless areas.

Also, note that the auto-exposure calibration of the camera is affected by the cropped area. Thus, cropping this section will help with exposure if there is a bright object at all times in the camera view. Figure 5.18 presents an example of how to crop these undesired features from the camera view.
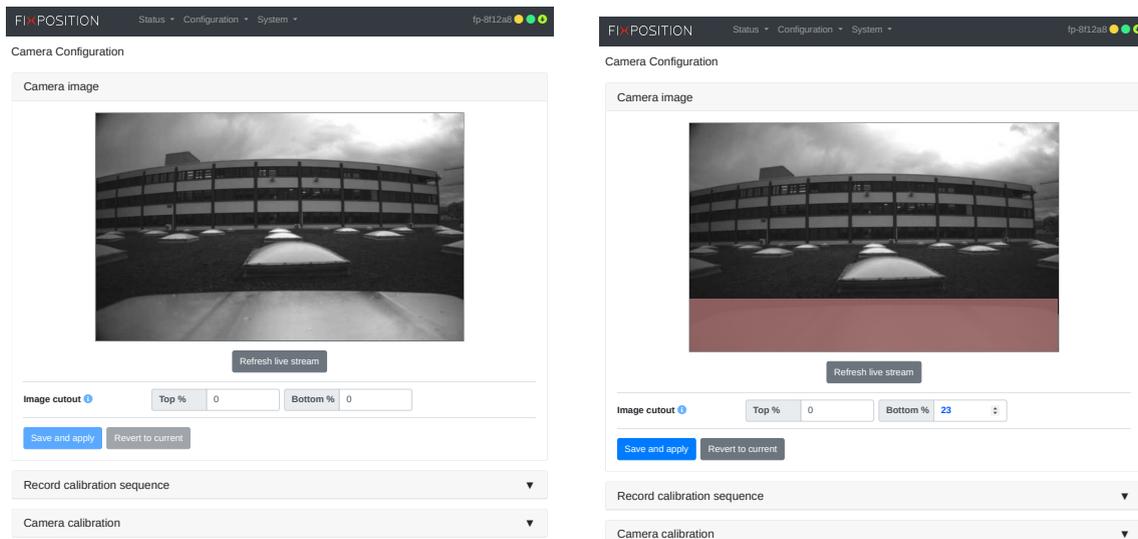
(a) Static features on the bottom of the image

(b) Cropping 23% on the bottom

Figure 5.18.: An example of the image view's cutout

5.8. Wheelspeed sensor configuration

The user can configure the wheelspeed sensor input by accessing the "Configuration → Wheelspeed" panel of the web interface. The wheelspeed sensor can be connected to the UART/TCP or CAN ports. However, before setting up the wheelspeed sensor, the corresponding ports should be configured according to Section 5.3.

Figure 5.19.: Wheelspeed sensor configuration on the web interface

The wheelspeed sensor configuration supports up to four external sensors. Each sensor requires a specific definition to be an input to the sensor fusion engine. Note that the user must type the corresponding strings strictly for each field (i.e., all fields are quote, space, and case-sensitive). The order of the wheelspeed sensor definition does not matter. The low-level sensor configuration required by any of the ports involves the following fields:

- **Enable** – Enables the wheelspeed sensor. If unchecked, the Vision-RTK 2 will not use any other parameters. It can be left unchecked to keep the configuration saved.

- **Name** – Reference name for the wheelspeed sensor representing the type of measurement that it will generate. The value must be unique among all enabled sensors. Depending on the setup, it should be one of the following:

    - RC (rear centre) – Rear Centre wheel (Most common case).

    - RL (rear left) – Rear Left wheel.

    - RR (rear right) – Rear Right wheel.

    - FL (front left) – Front Left wheel.

Vision-RTK 2 | Fixposition Positioning Sensor

- FR (front right) – Front Right wheel.

- **Type** - Unique identifier that specifies how to read this particular sensor type. Enter the string as-is, do not add quotes, spaces, or anything else.

- **Device** - Specifies the port in use:

  - CAN bus – for sensors connected via the CAN interface (see 5.3.2).

  - I/O port – for sensors connected via any of the I/O ports (see 5.3.1).

- **Reverse** – When enabled, the low-level sensor driver inverts the sign of the measurement (Only supported by some sensors).

- **Use sensor** – When enabled, the Fusion engine will employ the measurements from this wheelspeed sensor. This setting does not affect the sensor's operation.

- **Translation** – Translation – Translation vector from the center of the Vision-RTK 2's reference frame to the wheelspeed sensor axis (see Figure 3.6 for reference).

The general configuration shared by all sensors includes the following fields:

- **Wheelspeed sign** – Indicates if the wheelspeed values are signed (i.e., negative for backward movement and positive for forward). If unchecked, the Fusion engine interprets only the magnitude of the wheelspeed values and considers the unknown direction of motion. Keep this setting unchecked when mixing sensors with signed and unsigned values. We advise enabling the 'wheelspeed sign' option if supported by the wheel odometry sensor, as it causes the Vision-RTK2 to react faster to wheelspeed measurements after being stationary.

- **Record raw CAN** – When enabled, the sensor records the CAN frames in the bus. Only use when advised, as this option will impact performance.

Figure 5.20.: Preset list of the CAN and I/O wheelspeed configurations

Vision-RTK 2 | Fixposition Positioning Sensor

The Fixposition CAN message can support wheelspeed measurements from one or four wheels. However, if two or more wheels are used, the user can combine their vectors and input them as a single rear center wheelspeed measurement. All available presets have been listed in the configuration panel of the wheelspeed module in the web interface (see Figure 5.20). For customizing the Fixposition CAN message interface of the Vision-RTK 2, please consult the Fixposition team.

5.8.1. Fixposition CAN wheelspeed sensor

The CAN message must be formatted as described in Appendix B for a generic speed input. The user should send the corresponding CAN frames at regular intervals, where the input rate must be at most 50 Hz. The message latency, measured from computing the wheelspeed to broadcasting it over the CAN bus, must be as low as possible. Increased and, in particular, irregular latency degrades the Vision-RTK 2's performance. The low-level sensor parameters are automatically filled in when one of the preset settings is selected. For example, Figure 5.21 presents the configuration when the "Fixposition CAN message (vehicle speed, CAN)" preset is selected.

Figure 5.21.: Fixposition CAN message (vehicle speed) preset configuration

Based on the Fixposition CAN frame structure, when the Rear Centre (RC) wheelspeed sensor is selected, the corresponding vehicle wheelspeed value will be written in the Front Right (FR) data field. Thus, the Fixposition CAN message will be structured as follows:

- FR: vehicle speed value [mm/s]

- FL: 0xffff

- RR: 0xffff

- RL: 0xffff

Else, if two or more wheelspeed sensors are selected, the message structure will be:

- FR: vehicle speed value [mm/s]

- FL: vehicle speed value [mm/s]

- RR: vehicle speed value [mm/s]

- RL: vehicle speed value [mm/s]

Note that for this configuration to work, the CAN bus must be connected to CANL (CAN low), CANH (CAN high), and GND (Ground).

5.8.2. Fixposition I/O wheelspeed sensor

For streaming the wheelspeed values via the UART or TCP ports, the user must employ the NOV_B-RAWDMI message format detailed in Subsection 7.1.1. The binary message must be input on a UART/TCP port at a regular interval with a maximum input rate of 50 Hz. The message latency, measured from computing the wheelspeed to broadcasting it over the I/O port, must be as low as possible. The low-level sensor parameters are automatically filled in when one of the preset settings is selected. For example, Figure 5.22 presents the configuration when the "I/O message (vehicle speed, I/O port)" preset is selected.

Figure 5.22.: I/O message (vehicle speed) preset configuration

The dmi1..4 values are speed values in an arbitrary unit. Its resolution must be enough to produce meaningful measurements from small movements (e.g., mm/s, 0.01km/h, 0.02m/s). Coarse resolutions like kilometers per hour will not work well, particularly at slow speeds. Thus, we recommend employing millimeters per second as the standard measurement unit. The dmiX fields can have the following assignments:

- dmi1 is for RC wheel or FR wheel.

- dmi2 is for FL wheel or YW sensor.

- dmi3 is for RR wheel.

- dmi4 is for RL wheel.

The wheelspeed mask is divided into eight fields and determines which of the dmi1..4 values contain valid data and the type of data the value represents. The dmiX_mask fields can be either 0 or 1, depending on whether the dmiX value is invalid or valid. The dmiX_type is a 7-bit unsigned integer representing the value type in the dmiX field. Currently, only two values are supported: 0 for a linear speed or 1 for an angular velocity. Some example masks:

- 0x00000001 = dmi1 value is valid and represents a linear velocity.

- 0x00000802 = dmi2 value is valid and represents an angular velocity.

- 0x00000004 = dmi3 value is valid and represents a linear velocity.

- 0x02000008 = dmi4 value is valid and represents an angular velocity.

An example message with hexdump of the binary data:

- dmi1 = 111 = 0x0000006f = 6f 00 00 00 (at offset 12).

- dmi2 = -22222 = 0xffffff752 = 32 a9 ff ff (at offset 16).

- dmi3 = 333333 = 0x00051615 = 15 16 05 00 (at offset 20).

- dmi4 = -44 = 0xffffffd4 = d4 ff ff ff (at offset 24).

- mask = 0x00000001 | 0x00000002 | 0x00000004 | 0x00000008 = 0x0000000f = 0f 00 00 00 (at offset 28).

An example of a NOV_B-RAWDMI message is shown below:

```
0x0000 00000   aa 44 13 14  dd 08 00 00  00 00 00 00   6f 00 00 00
              ^^^^^^^^^^^^constant header^^^^^^^^^^^^  ^^^^dmi1^^^
0x0010 00016   32 a9 ff ff     15 16 05 00  d4 ff ff ff     0f 00 00 00
              ^^^^dmi2^^^  ^^^^dmi3^^^  ^^^^dmi4^^^  ^^^^mask^^^
0x0020 00032   69 9d 53 7b
              ^^checksum^
```

Figure 5.23.: NOV_B-RAWDMI example message

5.9. Sensor fusion configuration

The user can configure the sensor fusion engine by accessing the web interface and navigating to the "Configuration → Fusion" panel. In this section, the following configuration options are available:

- **Autostart:** Once enabled, the Fusion engine will be launched automatically on system bootup. Note: Enabling this will not start the Fusion engine directly.

- **Housing:** Prototype (i.e., 3D-printed) or Standard (i.e., aluminum).

- **Tuning mode:** Expected platform dynamics (see Section 2.3).

- **GNSS extrinsics:** Position of the GNSS antennas with respect to the Vision-RTK 2 sensor frame in meters. These values should be accurate to the mm level.

Figure 5.24.: Fusion engine configuration on the web interface

5.10. Log sensor data

The user can access the web interface and navigate to the "System → Logs" panel to generate a recording of the sensor. The user must press the "Record" button in the "Record logs" module to start recording. To stop a recording, press the "Stop" button. The generated recording is encrypted and meant for debugging purposes for the Fixposition customer support team.

The user can only use it to visualize the performed trajectory on the Customer Dashboard and download the corresponding KML data. If the user wants to record the customer outputs of the sensor, they can record the corresponding ROS topics or log the messages received.

The user can connect an external drive (minimum write speed 100 MB/s) to the Vision-RTK 2 through the USB-C port for additional storage. To record the bags in this device, the user must select "External" in the "Log location" field. Once the drive reaches total capacity, the logger will overwrite the beginning of the recording.

Figure 5.25.: Recording data module in the web interface

Customers can only see/use/delete their data and download the related KML data. If the customer wants to use this service, please get in touch with the Fixposition team.

Vision-RTK 2 | Fixposition Positioning Sensor

There are three recording levels available:

- **Minimal:**
    - Time and system information.
    - Fusion engine status, information, and output.
    - User I/O.
    - External sensors (wheel odometry).
    - Internal sensors (processed GNSS, IMU).
    - ROS information (clock, TF).
- **Medium:**
    - Same as minimal.
    - Raw GNSS and NTRIP data, latency, and status.
- **Maximal:**
    - Same as medium.
    - Camera image.

5.11. IMU calibration

The Vision-RTK 2 requires a start-up procedure before being fully operational. The user must ensure the following requirements are fulfilled to start the calibration procedure:

- Both receivers have an RTK-fixed status.
- The Fusion engine is initialized, and the extrinsics are correct.

Once fulfilled, drive approximately 2 minutes under RTK-fixed with some dynamic movement to converge the IMU biases. Perform eight figures and move back-and-forth within a 10 meters area as shown in Figure 5.27. The gyroscope and accelerometer bias will converge to steady values with this procedure. The sensor will raise a flag once it is fully calibrated (see #N°22 in FP A-ODOMETRY described in Subsection 7.3.1).

On the web interface, the user can visualize the status of the calibration procedure and the IMU noise by heading to the bottom of the "Status → Fusion" page. The corresponding indicators are presented in Figure 5.26. The IMU bias estimation can be either **Unknown** (sensor not initialized), **Not converged**, or **Converged**. Similarly, the IMU signal quality indicators can be either **Unknown**, **Low IMU noise**, **Medium IMU noise**, or **Excessive IMU noise**. Keep in mind that this indicator is calculated based on the variance of the IMU measurements over a period of time. Some platforms might experience excessive IMU noise without a significant impact on performance. This indicator only informs the user of their system dynamics.

IMU bias estimation	Unknown	Not converged	Converged
IMU signal quality	Unknown	Excessive IMU noise	Medium IMU noise	Low IMU noise

Figure 5.26.: IMU bias estimation and signal quality indicators
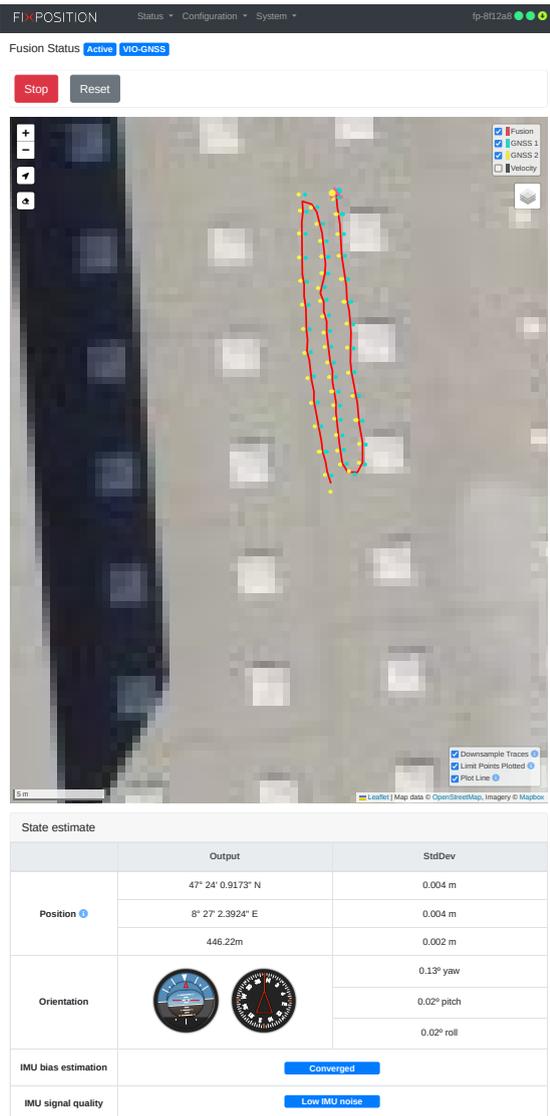
(a) Drive backward and forwards
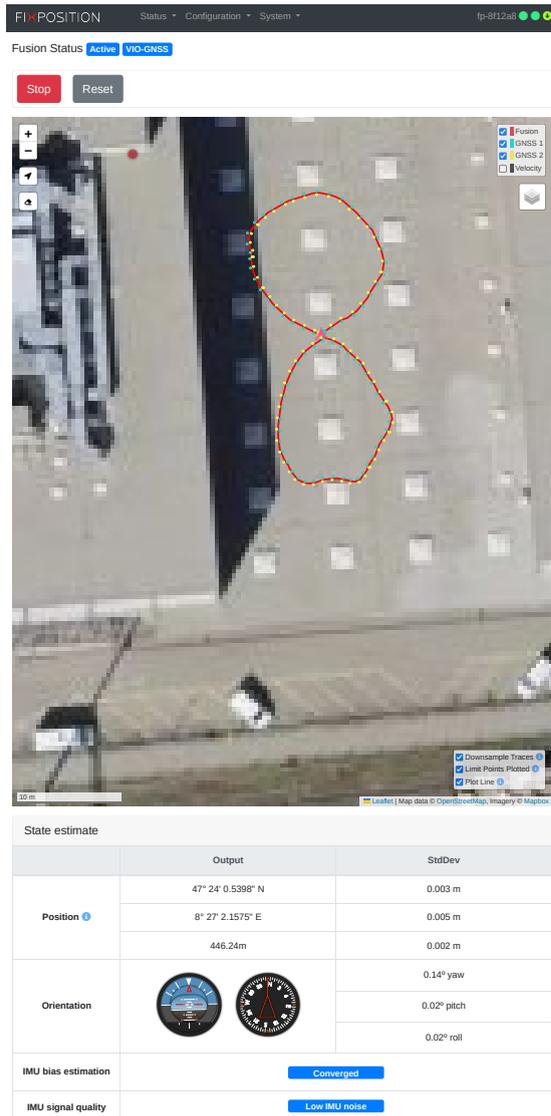
(b) Drive eight figures

Figure 5.27.: Example trajectory for the IMU calibration procedure
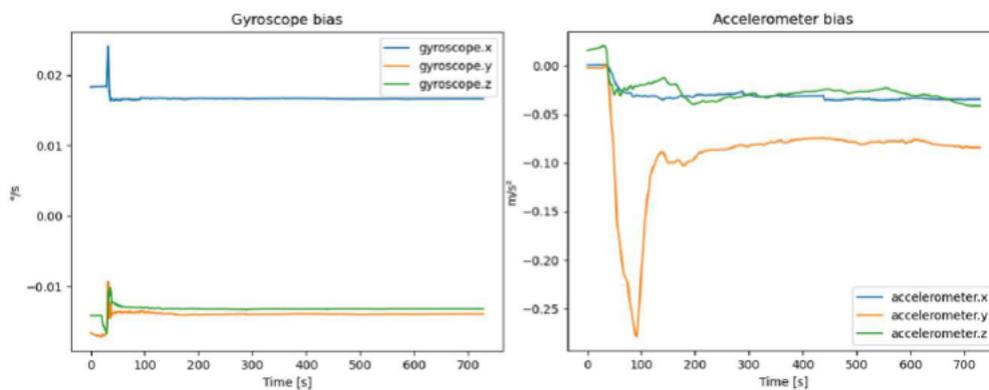
Figure 5.28.: Gyroscope and accelerometer biases over time

5.12. ROS driver installation

Please visit `https://github.com/fixposition/fixposition_driver` for more information. From FW 2.63, both ROS1 and ROS2 are supported. The ROS driver version should be compatible with its corresponding software release version.

The Fixposition ROS driver operating as a ROS node can listen to any I/O port to get outputs from the Vision-RTK 2 and then publish them in the ROS network. The user can directly stream the wheelspeed information via CAN/TCP/UART into the Vision-RTK 2. They can also fill in the X-directional velocity information from any resources into a ROS message and then publish it inside the ROS network so that the Fixposition ROS driver can subscribe to this information and convert it to NOV-B_RAWDMI (binary data).

Finally, this binary information will be streamed into the Fusion engine to improve localization performance. An extra node called Fixposition Odometry Converter is provided to help integrate the wheelspeed on your vehicle. This node is intended to be used as middleware if you already have a topic with the wheelspeed values running on your system. Currently, messages of the type Twist, TwistWithCov and Odometry are accepted.

Figure 5.29.: ROS driver of Vision-RTK 2 from Fixposition in Github

5.13. Point of interest configuration

The default reference frame of the VRTK2 is located at the X shape on the sensor housing (see section 2.3). If the user requires the odometry output in a different reference frame, the "**Output translation**" meaning "Translation from the sensor to output represented by x-y-z Cartesian coordinates" and "**Output rotation**" meaning "Rotation from the sensor to output represented by Roll-Pitch-Yaw (RPY) in the Cartesian coordinates" fields in the "Output generators" module of the web interface can be used (see Figure 5.12).

The XYZ displacements from the center of the VRTK2's reference frame to the desired point are expressed in meters. In contrast, the rotation from the sensor's frame to the output's body frame is expressed in degrees using Roll-Pitch-Yaw (RPY) rotations, also known as Euler Angles. For reference, the arrow on the web interface's map page points towards the output frame's positive X direction (see Figure 5.31).

For reference, Figure 5.30 shows an example of a transformation between the VRTK's body frame and the output's body frame using the previously described convention. In this example, the point of interest (POI) is located at an arbitrary point inside the vehicle. Therefore, the user must calculate the XYZ translations from the sensor's frame of reference to the POI and then rotate the frame using RPY angles.
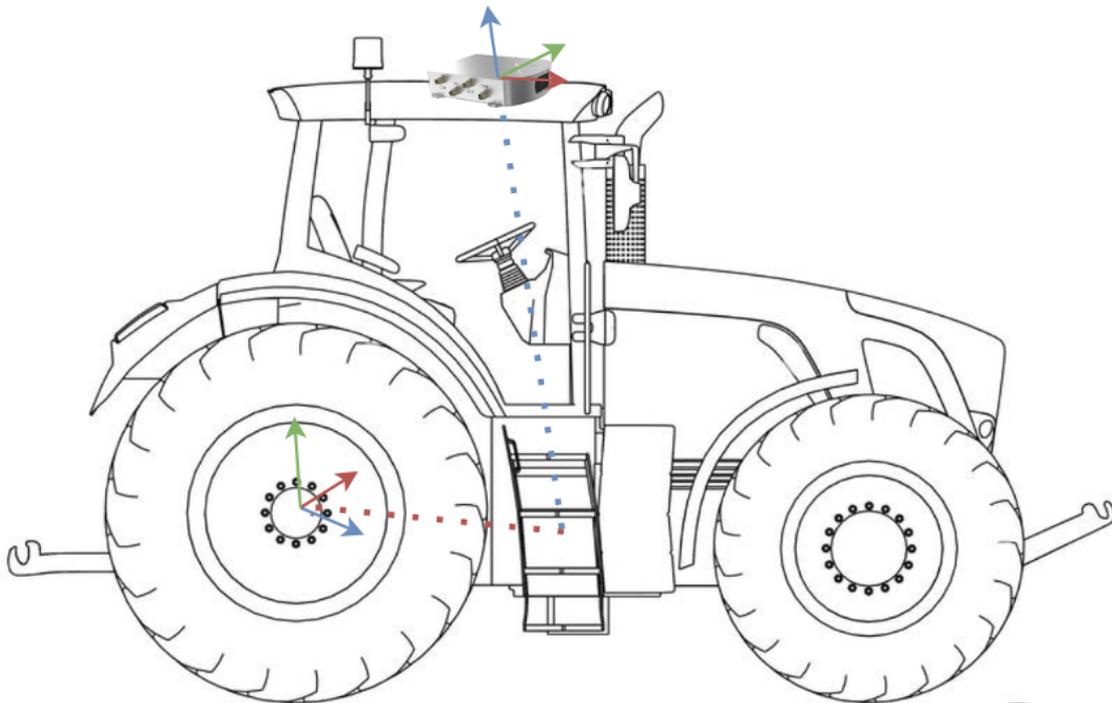
Figure 5.30.: Transform from the VRTK2's body to the output reference frame
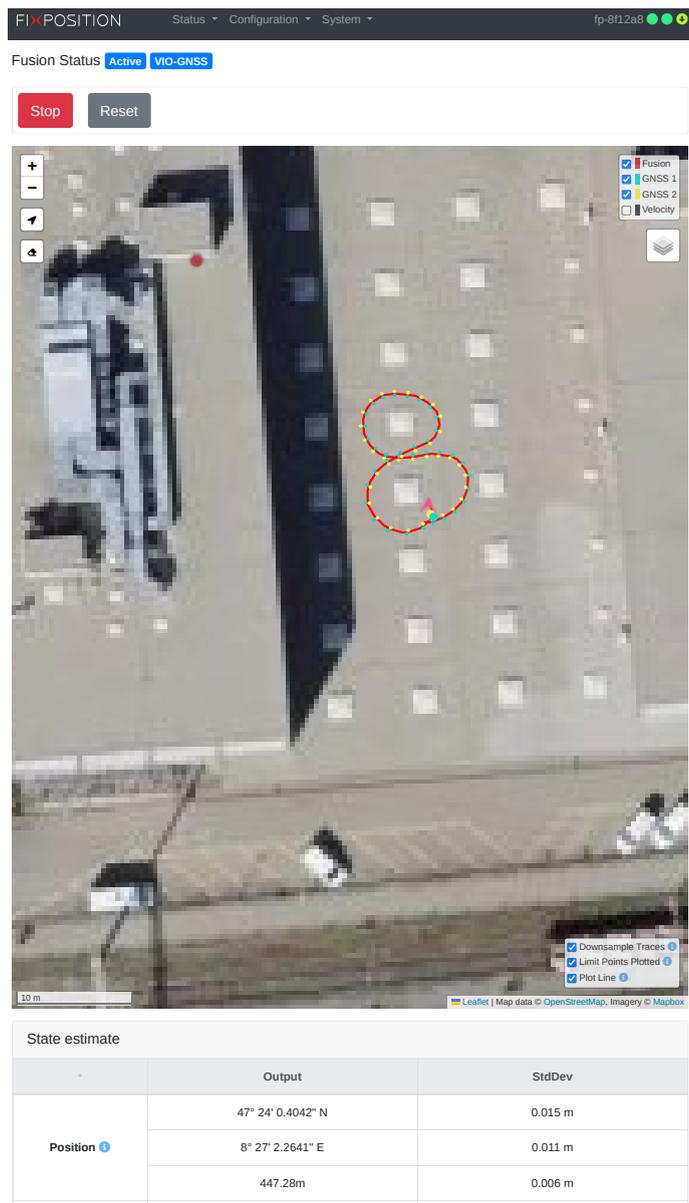
Figure 5.31.: Arrow pointing towards the positive X direction of the output's body frame

5.14. Web interface indicators

The navigation bar of the web interface contains useful indicators to signal the status of the Vision-RTK 2 and its related processes. Fig. 5.32 presents all the available indicators.

Figure 5.32.: Indicators on the web interface

The available indicators are:

1. **System:** This indicator can arise when the system state is not "running," e.g., during boot-up, shut-down, or when a component fails to initialize.

2. **Internet:** The system is not online. This indicator will appear if the sensor cannot ping the 8.8.8.8 address.

3. **Camera:** Missing camera calibration or failure during camera initialization.

4. **Time synchronization:** The system time is not synchronized with GNSS or NTP time info resources.

5. **Correction data stream:** The sensor is not receiving RTCM3 data. The tooltip will show details (e.g., "warning, connected but no data").

6. **External disk connected:** An external hard disk is connected to the sensor. Note that the user still needs to mount/unmount the disk in "System → Logs" manually.

7. **Recording calibration:** A calibration sequence is being recorded.

8. **Recording log:** A log file is being recorded.

9. **IO stream:** One or more error counters are increasing (I/O of any port). Additionally, a more detailed alert is shown on the I/O status tab of the "System → Info" page (see Fig. 5.33). The alert stays visible as long as the error counter(s) keep incrementing and will disappear after a timeout (5s for the indicator, 30s for the alert in the I/O status page). Typical error sources include:

   UART ports:

   a) Too many output messages are configured for the selected bandwidth (determined by the baud rate).

   b) Too long, faulty, and/or low-quality cables.

   c) Occasional errors on a UART connection are expected, albeit rare.

   CANSTR port:

   a) Too many output messages are configured for the selected bandwidth (determined by the bitrate and other devices on the bus).

   b) Bad CAN termination (missing when required, present when not).

   c) Too long CAN cable (e.g., for OBD2, it must be shorter than 30cm).

   d) A misbehaving device on the CAN bus.

   e) Too much load on the CAN bus.

Vision-RTK 2 | Fixposition Positioning Sensor

f) Occasional errors on a CAN connection are expected. The frequency depends on various factors (e.g., number of devices, bitrate, cable length).

TCP ports:

a) Congested network and/or too little bandwidth available. Recommended: use direct one-on-one Ethernet connections between the sensor and host.

Note that on the initial load of the web interface, this warning might appear if the error counters are not zero, even if the last error was a long time ago.

10. **Other:** Unspecified warning - reserved for future use.

Figure 5.33.: Error message on the I/O status tab of the web interface

Additionally, when the GNSS baseline check fails, the following warning appears on the web interface. This indicator does not necessarily suggest an error, as leaving the sensor on a prolonged GNSS outage can trigger this warning due to the low quality of the RTK correction data. Nonetheless, the user should verify their GNSS extrinsics if this error often appears under good GNSS conditions or RTK-fixed at the mission's beginning.
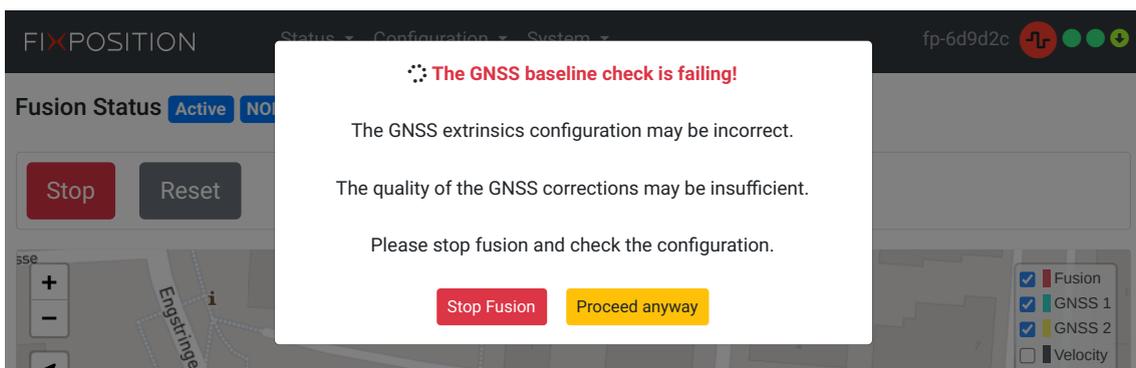
Figure 5.34.: GNSS baseline check error message on the web interface

Vision-RTK 2 | Fixposition Positioning Sensor

Status Dashboard

6.1. GNSS solution types

The following RTK/GNSS fix convention is used:

Fix type	Value	Color code	Description
Unknown	0	Dark	The receiver has no satellite signals.
No fix	1	Blue	The receiver has not enough satellite signals.
Reserved	2	-	-
Reserved	3	-	-
Single 2D	4	Red	Autonomous GNSS fix with very few available satellite signals.
Single 3D	5	Orange	Autonomous GNSS fix, typical in situation where no correction data is received or in outages.
Reserved	6	-	-
RTK float	7	Yellow	RTK with ambiguities not fully solved.
RTK fixed	8	Green	RTK with ambiguities fully solved.

Table 6.1.: List of RTK/GNSS signal qualities

6.2. GNSS status dashboard

Figure 6.1.: GNSS status dashboard

1. **Left column:** status of first GNSS receiver.

2. **Right column:** status of second GNSS receiver.

3. **Status flags:** these 3 signal light indicate the status of the GNSS and NTRIP messages

   a) GNSS 1: **Green** - Fixed, **Yellow** - Float, **Orange** - Single, **Gray** - No data.

b) GNSS 2: **Green** - Fixed, **Yellow** - Float, **Orange** - Single, **Gray** - No data.

c) NTRIP stability: **Green** - Good, **Yellow** - Warn, **Orange** - Poor, **Gray** - No data.

4. **Signal levels:** histogram of the number of GNSS signals and their corresponding signal levels (measured as the carrier to noise ratio in dB Hz). The coloured bars represent the number of signals used in navigation and the gray ones represent the total number of signals tracked. The color scheme used for this histogram corresponds to:

   a) Reds: low signal levels, typically unusable or imprecise

   b) Greens: good signal levels (40+ dBHz)

5. **NTRIP correction stream stability and latency statistics:**

   a) Latency: measured by comparing the timestamp of the RTCM3 messages to the system time.

   b) Update rate: indicates how many sets of correction measurements are received per second. Different RTK services have different update rates. Typically it is 1 Hz nominal, but the sensor supports from 0.5 up to 2 Hz.

   c) Data rate: indicates how much data is received per second. Typically, between 0.5 to 1.5 KB/s.

   d) Stability indicators: **Green** - Good, **Yellow** - Warn, **Orange** - Poor, **Gray** - No data. These values are based on thresholds. For "good" stability, the update rate should be higher than 0.25Hz, the latency lower than 5s, and the maximum latency lower than 10s.

   e) The rows will only become available when the stream has run long enough. Changing the NTRIP configuration, internet outages, web interface disconnection, etc., can lead to artifacts.

6. **Expert mode toggle:** Display additional information in the tables

7. **GNSS fix types:** they can be one of the following:

   a) **No fix** – the GNSS receiver has no fix at all.

   b) **Dead-reckoning** – The GNSS receiver has lost the fix and is extrapolating the movement.

   c) **Single 2D** – autonomous GNSS fix, 2D as only a few signals are available.

   d) **Single 3D** – autonomous GNSS fix. Typical situation if no RTK correction data is available.

   e) **RTK float** – real-time kinematic fix with ambiguities not fully solved.

   f) **RTK fixed** – real-time kinematic fix with ambiguities fully solved.

8. **Update indicators:** these indicators blink whenever the data is updated.

   a) For GNSS 1 and GNSS2, the indicator blinks at 1 Hz; more or less in sync.

   b) The NTRIP indicator blinks at 0.2 Hz (every 5 seconds).

9. **Antenna state and power:** displays antenna supervisor state (left badge) and the antenna power supply status (right badge).

    a) Antenna states:

        i. **Initialising** – During startup of the GNSS receiver.

        ii. **OK** – Antenna detected and OK, power will be On.

        iii. **Open** – No antenna detected, power will be On.

        iv. **Short** – Antenna short circuit detected, power will be Off.

    b) Power states:

        i. **On** – Antenna power supply enabled.

        ii. **Off** – Antenna power supply disabled. Note that after removing the electrical short it can take the GNSS receiver up to one minute to detect this and return to power On.

Note: The 'Time and date' fields display the current estimate using the solutions from each u-blox F9P GNSS receiver independently. The user should not trust these measurements when the values are in gray.

6.3. Input/output dashboard

Output ⓘ	UART1	UART2	TCP0	TCP1	TCP2	TCP3	TCP4	CANSTR
Bytes	28'107'815	149	...9'823'373	...9'823'373	...9'823'373	...9'823'373	...9'823'373	149
Messages	92'915	3	6'995'924	6'995'924	6'995'924	6'995'924	6'995'924	3
Errors	0	0	0	0	0	0	0	0
FP_A-CORRIMU	0	0	1'552'332	1'552'332	1'552'332	1'552'332	1'552'332	0
FP_A-GNSSANT	0	0	7'021	7'021	7'021	7'021	7'021	0
FP_A-GNSSCORR	0	0	7'020	7'020	7'020	7'020	7'020	0
FP_A-LLH	0	0	77'421	77'421	77'421	77'421	77'421	0
FP_A-ODOMETRY	77'428	0	77'428	77'428	77'428	77'428	77'428	0
FP_A-RAWIMU	0	0	1'563'441	1'563'441	1'563'441	1'563'441	1'563'441	0
FP_A-TEXT_INFO	3	3	3	3	3	3	3	3
FP_A-TF_POIIMUH	0	0	1'563'119	1'563'119	1'563'119	1'563'119	1'563'119	0
FP_A-TF_POIVRTK	7'742	0	7'742	7'742	7'742	7'742	7'742	0
FP_A-TF_VRTKCAM	7'742	0	7'742	7'742	7'742	7'742	7'742	0
NMEA-GP-GGA_FUSION	0	0	77'421	77'421	77'421	77'421	77'421	0
NMEA-GP-GGA_GNSS	0	0	35'077	35'077	35'077	35'077	35'077	0
NMEA-GP-GGA_GNSS1	0	0	38'641	38'641	38'641	38'641	38'641	0
NMEA-GP-GGA_GNSS2	0	0	38'635	38'635	38'635	38'635	38'635	0
NMEA-GP-HDT_FUSION	0	0	77'424	77'424	77'424	77'424	77'424	0
NMEA-GP-RMC_GNSS	0	0	35'078	35'078	35'078	35'078	35'078	0
NMEA-GP-RMC_GNSS1	0	0	38'647	38'647	38'647	38'647	38'647	0
NMEA-GP-RMC_GNSS2	0	0	38'633	38'633	38'633	38'633	38'633	0
NOV_B-BESTGNSSPOS_GNSS1	0	0	38'644	38'644	38'644	38'644	38'644	0
NOV_B-BESTGNSSPOS_GNSS2	0	0	38'640	38'640	38'640	38'640	38'640	0
NOV_B-HEADING2	0	0	35'077	35'077	35'077	35'077	35'077	0
NOV_B-INSPVAX	0	0	77'409	77'409	77'409	77'409	77'409	0
NOV_B-RAWIMU	0	0	1'563'329	1'563'329	1'563'329	1'563'329	1'563'329	0
Input ⓘ	UART1	UART2	TCP0	TCP1	TCP2	TCP3	TCP4	CANSTR
Bytes	0	0	0	0	0	0	0	0
Messages	0	0	0	0	0	0	0	0
Errors	0	0	0	0	0	0	0	0

Figure 6.2.: Input/Output Status dashboard

From FW 2.63, the user can navigate to "System → Info" to supervise the I/O status.

- **Bytes:** Accumulated data size of the total messages in a specified port after booting.

- **Messages:** Accumulated total number of the messages in a specified port after booting.

- **Errors:** Accumulated number of the bad/error ones among the total messages after booting.

Counters start at 0 at the start, unaffected by fusion or anything. they will roll over at 2^{32}. For example, in the output demonstrated in Figure 6.2, FP_A-ODOMETRY has accumulated 1760 messages to output in port UART 1 and 5278 messages to output in port UART 2. In UART 1, the total number of accumulated messages to output in UART 1 is 2982 and among them, no error happened. The total data size of these messages is 309056. In CAN bus (CANSTR) port, 37573 messages are reported as error among the output 55299 messages. These data can be further analyzed if necessary for some troubleshooting.

CHAPTER 7

Input Output Messages

7.1. Wheelspeed Input

7.1.1. NOV-B_RAWDMI

For streaming wheelspeed information via the UART serial port or TCP port, the message format called NOV_B-RAWDMI is introduced by Fixposition, the message format should be as follows:

Offset	Type	Value	Description	
0	uint8_t	0xaa		Sync byte 1
1	uint8_t	0x44		Sync byte 2
2	uint8_t	0x13		Sync byte 3
3	uint8_t	20	Header	Payload length
4	uint16_t	2269		Message ID
6	uint16_t	0		Week number
8	int32_t	0		Time of the week [ms]
12	int32_t	dmi1		Wheelspeed value 1, for RC or FR wheel
16	int32_t	dmi2		Wheelspeed value 2, for FL wheel
20	int32_t	dmi3	Payload	Wheelspeed value 3, for RR wheel
24	int32_t	dmi4		Wheelspeed value 4, for RL wheel
28	int32_t	mask		Wheelspeed value mask
32	uint32_t	cksum	Checksum	CRC2 checksum (seed 0x00000000, polynomial 0xEDB88320)

Table 7.1.: NOV-B_RAWDMI message format

56

7.2. GNSS correction input

7.2.1. RTCM3

The Vision-RTK 2 needs at least the following RTCM3 messages for proper operation: Reference station position (rate: every 10 seconds, or more often), one of:

- RTCM type 1005 (Stationary RTK reference station ARP)
- RTCM type 1006 (Stationary RTK reference station ARP with antenna height)

GPS observables (rate: 1 Hz), one of the:

- RTCM type 1074 (GPS MSM4)
- RTCM type 1075 (GPS MSM5)
- RTCM type 1077 (GPS MSM7)

Galileo observables (rate: 1 Hz), one of:

- RTCM type 1094 (Galileo MSM4)
- RTCM type 1095 (Galileo MSM5)
- RTCM type 1097 (Galileo MSM7)

BeiDou observables (rate: 1 Hz), one of:

- RTCM type 1124 (BeiDou MSM4)
- RTCM type 1125 (BeiDou MSM5)
- RTCM type 1127 (BeiDou MSM7)

GLONASS observables (rate: 1 Hz), one of:

- RTCM type 1084 (GLONASS MSM4)
- RTCM type 1085 (GLONASS MSM5)
- RTCM type 1087 (GLONASS MSM7)

GLONASS code-phase biases (rate: every 5 seconds or more often):

- RTCM type 1230 (GLONASS code-phase biases)

7.3. Fusion output

The required output port bandwidth depends on the configuration. For example, on a serial port, the baud rate must be set high enough for the necessary bandwidth. The exact number depends on what messages are enabled for the port and the configured fusion output frequency.

7.3.1. FP_A-ODOMETRY

This message contains full fusion odometry output and additional status information. It is output at the configured rate (see above).

Example message (wrapped on multiple lines for readability):

```
$FP,ODOMETRY,2,2231,227610.750000,4279243.1641,635824.2171,4671589.8683,
    -0.412792,0.290804,-0.123898,0.854216,-17.1078,-0.0526,-0.3252,0.02245,
     0.00275,0.10369,-1.0385,-1.3707,9.8249,4,1,8,8,1,0.01761,0.02274,
    0.01713,-0.00818,0.00235,0.00129,0.00013,0.00015,0.00014,-0.00001,
    0.00001,0.00002,0.03482,0.06244,0.05480,0.00096,0.00509,0.00054,
    fp_release_vr2_2.54.0_160*4F\r\n
```

Message fields:

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	ODOMETRY	Message type, always ODOMETRY for this message
2	msg_version	Numeric	-	2	Message version, always 2 for this version of the ODOMETRY message
3	gps_week	Numeric	-	2231	GPS week number, range 0–9999
4	gps_tow	Float (.6)	s	227610.750000	GPS time of week, range 0.000–604799.999999
5	pos_x	Float (.4)	m	4279243.1641	Position in ECEF, X component
6	pos_y	Float (.4)	m	635824.2171	Position in ECEF, Y component
7	pos_z	Float (.4)	m	4671589.8683	Position in ECEF, Z component
8	orientation_w	Float (.6)	-	-0.412792	Quaternion with respect to ECEF, W component
9	orientation_x	Float (.6)	-	0.290804	Quaternion with respect to ECEF, X component
10	orientation_y	Float (.6)	-	-0.123898	Quaternion with respect to ECEF, Y component
11	orientation_z	Float (.6)	-	0.854216	Quaternion with respect to ECEF, Z component
12	vel_x	Float (.4)	m/s	-17.1078	Velocity in output frame, X component
13	vel_y	Float (.4)	m/s	-0.0526	Velocity in output frame, Y component

#	Field	Format	Unit	Example	Description
14	vel_z	Float (.4)	m/s	-0.3252	Velocity in output frame, Z component
15	rot_x	Float (.5)	rad/s	0.02245	Bias corrected angular velocity in output frame, X component
16	rot_y	Float (.5)	rad/s	0.00275	Bias corrected angular velocity in output frame, Y component
17	rot_z	Float (.5)	rad/s	0.10369	Bias corrected angular velocity in output frame, Z component
18	acc_x	Float (.4)	m/s2	-1.0385	Bias corrected acceleration in output frame, X component
19	acc_y	Float (.4)	m/s2	-1.3707	Bias corrected acceleration in output frame, Y component
20	acc_z	Float (.4)	m/s2	9.8249	Bias corrected acceleration in output frame, Z component
21	fusion_status	Numeric	-	4	Fusion status, see below
22	imu_bias_status	Numeric	-	1	IMU bias status, see below
23	gnss1_fix	Numeric	-	8	Fix status of GNSS1 receiver, see below
24	gnss2_fix	Numeric	-	8	Fix status of GNSS2 receiver, see below
25	wheelspeed_status	Numeric	-	1	Wheelspeed status, see below
26	pos_cov_xx	Float (5)	m2	0.01761	Position covariance, element XX
27	pos_cov_yy	Float (5)	m2	0.02274	Position covariance, element YY
28	pos_cov_zz	Float (5)	m2	0.01713	Position covariance, element ZZ
29	pos_cov_xy	Float (5)	m2	-0.00818	Position covariance, element XY
30	pos_cov_yz	Float (5)	m2	0.00235	Position covariance, element YZ
31	pos_cov_xz	Float (5)	m2	0.00129	Position covariance, element XZ
32	orientation_cov_xx	Float (5)	rad2	0.00013	Velocity covariance, element XX
33	orientation_cov_yy	Float (5)	rad2	0.00015	Velocity covariance, element YY
34	orientation_cov_zz	Float (5)	rad2	0.00014	Velocity covariance, element ZZ
35	orientation_cov_xy	Float (5)	rad2	-0.00001	Velocity covariance, element XY
36	orientation_cov_yz	Float (5)	rad2	0.00001	Velocity covariance, element YZ
37	orientation_cov_xz	Float (5)	rad2	0.00002	Velocity covariance, element XZ
38	vel_cov_xx	Float (5)	m2/s2	0.03482	Velocity covariance, element XX
39	vel_cov_yy	Float (5)	m2/s2	0.06244	Velocity covariance, element YY
40	vel_cov_zz	Float (5)	m2/s2	0.05480	Velocity covariance, element ZZ
41	vel_cov_xy	Float (5)	m2/s2	0.00096	Velocity covariance, element XY
42	vel_cov_yz	Float (5)	m2/s2	0.00509	Velocity covariance, element YZ
43	vel_cov_xz	Float (5)	m2/s2	0.00054	Velocity covariance, element XZ
44	sw_version	String	-	fp_release_ vr2_2.54.0_160	Software version

Table 7.2.: FP_A-ODOMETRY message fields

Value	Description
0	Not started
1	Vision only
2	Visual inertial fusion
3	Inertial-GNSS fusion
4	Visual-inertial-GNSS fusion

Table 7.3.: Fusion status values

Value	Description
0	Not converged
1	IMU bias converged

Table 7.4.: IMU bias status values

Value	Description
0	Unknown
1	No fix
2	Dead-reckoning only
3	Time-only fix
4	Single 2D fix
5	Single 3D fix
6	Single 3D fix with dead-reckoning
7	RTK float fix
8	RTK fixed fix

Table 7.5.: GNSS fix type values

Value	Description
-1	No wheelspeed enabled
0	At least one wheelspeed enabled, no wheelspeed converged
1	At least one wheelspeed enabled and converged

Table 7.6.: Wheelspeed status values

7.3.2. FP_A-LLH

This message contains time, geographic coordinates and the position covariance of the output frame in East-North-up (ENU). The coordinates are transformed from ECEF using the WGS-84 parameters. It is output at the configured rate.

Example message (wrapped on multiple lines for readability):

```
$FP,LLH,1,2231,227563.250000,47.392357470,8.448121451,473.5857,0.04533,
0.03363,0.02884,0.00417,0.00086,-0.00136*62\r\n
```

Message fields:

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	LLH	Message type, always LLH for this message
2	msg_version	Numeric	-	1	Message version, always 1 for this version of the LLH message
3	gps_week	Numeric	-	2231	GPS week number, range 0–9999
4	gps_tow	Float (.6)	s	227563.250000	GPS time of week, range 0.000–604799.999999
5	latitude	Float (.9)	deg	47.392357470	Latitude, range -90.000000000 –90.000000000, > 0 for North, < 0 for South
6	longitude	Float (.9)	deg	8.448121451	Longitude, range -180.000000000 –180.000000000, > 0 for East, < 0 for West
7	height	Float (.4)	m	473.5857	Ellipsoidal height, range -1000.0000-50000.0000
8	pos_cov_ee	Float (5)	m2	0.04533	Position covariance in ENU, element EE
9	pos_cov_nn	Float (5)	m2	0.03363	Position covariance in ENU, element NN
10	pos_cov_uu	Float (5)	m2	0.02884	Position covariance in ENU, element UU
11	pos_cov_en	Float (5)	m2	0.00417	Position covariance in ENU, element EN
12	pos_cov_nu	Float (5)	m2	0.00086	Position covariance in ENU, element NU
13	pos_cov_eu	Float (5)	m2	-0.00136	Position covariance in ENU, element EU

Table 7.7.: FP_A-LLH message fields

7.3.3. NOV_B-INSPVAX

This message contains the Fusion position (w.r.t. geodetic frame assuming WGS-84 transformation parameters), velocity (w.r.t. local tangential plane), and attitude (w.r.t. output/POI frame), as well as information about the position standard deviation. The position type and solution status fields indicate whether or not the corresponding data is valid. This message contains the same information as the FP_A-ODOMETRY message, only presented in another commercial format.

Message fields:

#	Offset	Field	Type	Unit	Description
-	0	sync1	uint8_t	-	Sync byte 1 (always 0xaa)
-	1	sync2	uint8_t	-	Sync byte 2 (always 0x44)
-	2	sync3	uint8_t	-	Sync byte 3 (always 0x12)
-	3	header_len	uint8_t	bytes	Length of the header (always 28)
-	4	msg_id	uint16_t	-	Message ID, always 1465 for this message
-	6	msg_type	uint8_t	-	See protocol documentation
-	7	reserved1	uint8_t	-	Reserved, ignore
-	8	payload_len	uint8_t	bytes	Payload size, always 126 for this message
-	10	reserved2	uint16_t	-	Reserved, ignore
-	12	reserved3	uint8_t	-	Reserved, ignore
1	13	time_status	uint8_t	-	See protocol documentation
2	14	gps_wno	uint16_t	-	GPS week number
3	16	gps_tow	int32_t	ms	GPS time of week
-	20	reserved4	uint32_t	-	Reserved, ignore
-	24	reserved5	uint16_t	-	Reserved, ignore
-	26	reserved6	uint16_t	-	Reserved, ignore
4	28	ins_status	uint32_t	-	Solution status, see below
5	32	pos_type	uint32_t	-	Positioning mode, see below
6	36	lat	double	deg	Latitude
7	44	lon	double	deg	Longitude
8	52	height	double	m	Ellipsoidal height
-	60	reserved7	uint32_t	-	Reserved, ignore
9	64	vel_n	double	m/s	Velocity North
10	72	vel_e	double	m/s	Velocity East
11	80	vel_u	double	m/s	Velocity up
12	88	roll	double	deg	Roll in local level
13	96	pitch	double	deg	Pitch in local level
14	104	azim	double	deg	Azimuth in local level
15	112	std_lat	float	m	Latitude standard deviation
16	116	std_lon	float	m	Longitude standard deviation
17	120	std_height	float	m	Height standard deviation
18	124	std_vel_n	float	m/s	Velocity North standard deviation
19	128	std_vel_e	float	m/s	Velocity East standard deviation
20	132	std_vel_u	float	m/s	Velocity up standard deviation
21	136	std_roll	float	deg	Roll standard deviation
22	140	std_pitch	float	deg	Pitch standard deviation
23	144	std_azim	float	deg	Azimuth standard deviation
24	148	ext_status	uint32_t	-	Extended status, see below
-	152	reserved8	uint16_t	-	Reserved, ignore
-	154	checksum	uint32_t	-	CRC32 checksum (see protocol documentation)

Table 7.8.: NOV B-INSPVAX message fields

Value	Description
0	Inactive
1	Aligning
3	Visual-inertial-GNSS fusion
6	Visual inertial fusion

Table 7.9.: Solution status values

Value	Description
0	No fix
50	RTK fixed fix
34	RTK float fix
16	Single 3D fix
56	INS and RTK fixed fix
55	INS and RTK float fix
53	INS and single 3D fix
19	INS only

Table 7.10.: Positioning mode values

Bit	Description
bit 0 (0x00000001)	Position update
bit 2 (0x00000004)	Zero velocity update
bit 6 (0x00000040)	INS solution convergence
bit 9 (0x00000200)	Velocity update

Table 7.11.: Extended status values

7.3.4. NMEA-GP-GGA_FUSION

This message contains time, date, position (in LLH coordinates), fix quality, number of satellites, and horizontal dilution of precision (HDOP) data provided by the Fusion output.

Example message:

```
$GPGGA,151229.40,4723.54108,N,00826.88485,E,4,12,00.98,473.5,M,,,,*3A\r\n
```

Example message with non-standard high-precision fields:

```
$GPGGA,151229.3999,4723.5410795,N,00826.8848463,E,4,31,00.98,473.5368,
   M,,,,*0F\r\n
```

Message fields:

#	Field	Format	Unit	Example	Description
1	time	hhmmss.ss(ss)	-	151229.40	UTC time (hours, minutes and seconds)
2	lat	ddmm.mmmmm(mm)	-	4723.54108	Latitude
3	lat_ns	Character	-	N	Latitude north (N) or south (S) indicator
4	lon	dddmm.mmmmm(mm)	-	00826.88485	Longitude
5	lon_ew	Character	-	E	Longitude east (E) or west (W) indicator
6	quality	Digit	-	4	Quality indicator, 4 (RTK fixed), 5 (RTK float), 1 (no RTK), 6, 0
7	num_sv	Decimal	-	12	Number of satellites, range 00-12
8	hdop	Float (.2)	-	00.98	Horizontal dilution of precision, range 00.10-99.99
9	alt	Float (.1/.4)	m	473.5	Altitude (above ellipsoid)
10	alt_unit	Character	-	M	Altitude unit, always M (metres)
11	sep	-	-		Geoid separation, always null
12	sep_unit	-	-		Geoid separation unit, always null
13	diff_age	-	-		Age of differential data, always null
14	diff_sta	-	-		DGPS station ID, always null

Table 7.12.: NMEA-GP-GGA_FUSION message format

7.3.5. NMEA-GP-HDT_FUSION

This message contains the actual vessel heading (from the Fusion solution) in degrees. "True" means the geometric North (not the magnetic North). Note that this is not the heading of motion (direction of travel) but the orientation of the sensor. We recommend using the FP_A-ODOMETRY.orientation_* and FP_A-ODOMETRY.vel_* fields to calculate the heading of motion. See Appendix C to understand how to extract the heading from the FP_A-ODOMETRY message.

#	Field	Format	Unit	Example	Description
1	heading	Float (.1.4)	deg	61.7	True heading
2	true_ind	Character	-	T	Always T

Table 7.13.: NMEA-GP-HDT_FUSION message format

7.4. TF output

7.4.1. FP_A-TF

This message contains the transformation (translation in Cartesian coordinates and rotation as a quaternion) from frame A to B.

Example messages:

```
$FP,TF,2,2233,315835.000000,VRTK,CAM,-0.00000,-0.00000,-0.00000,1.000000,
    0.000000,0.000000,0.000000*6B\r\n
```

```
$FP,TF,2,2233,315835.000000,POI,VRTK,-0.99301,-2.01395,-2.99298,0.999995,
    -0.002616,-0.001748,-0.000868*52\r\n
```

Message fields:

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	TF	Message type, always TF for this message
2	msg_version	Numeric	-	2	Message version, always 2 for this version of the TF message
3	gps_week	Numeric	-	2233	GPS week number, range 0–9999
4	gps_tow	Float (.6)	s	315835.000000	GPS time of week, range 0.000–604799.999999
5	frame_a	String	-	POI	Target frame (maximum 8 characters: A-Z and 0-9)
6	frame_b	String	-	VRTK	Initial frame (maximum 8 characters: A-Z and 0-9)
7	translation_x	Float (.5)	m	-0.99301	Translation, X component
8	translation_y	Float (.5)	m	-2.01395	Translation, Y component
9	translation_z	Float (.5)	m	-2.99298	Translation, Z component
10	orientation_w	Float (.6)	-	0.999995	Rotation in quaternion, W component
11	orientation_x	Float (.6)	-	-0.002616	Rotation in quaternion, X component
12	orientation_y	Float (.6)	-	-0.001748	Rotation in quaternion, Y component
13	orientation_z	Float (.6)	-	-0.000868	Rotation in quaternion, Z component

Table 7.14.: FP_A-TF message format

7.5. IMU Output

All IMU output is generated at the IMU frequency of approximately 200 Hz. Note that the IMU output requires a lot of output bandwidth. For example, on a serial port, the baud rate must be set high enough for the necessary bandwidth. The exact number depends on what messages are enabled for the port.

7.5.1. FP_A-RAWIMU

This message contains time and the IMU's acceleration and angular velocity (raw value, no bias correction, only coordinate transformation applied) in the sensor frame.

Example message:

```
$FP,RAWIMU,1,2197,126191.777855,-0.199914,0.472851,9.917973,0.023436,
    0.007723,0.002131*34\r\n
```

Message fields:

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	RAWIMU	Message type, always RAWIMU for this message
2	msg_version	Numeric	-	1	Message version, always 1 for this version of the RAWIMU message
3	gps_week	Numeric	-	2197	GPS week number, range 0–9999
4	gps_tow	Float (.6)	s	126191.777855	GPS time of week, range 0.000–604799.999999
5	acc_x	Float (.6)	m/s2	-0.199914	Raw acceleration in output frame, X component
6	acc_y	Float (.6)	m/s2	0.472851	Raw acceleration in output frame, Y component
7	acc_z	Float (.6)	m/s2	9.917973	Raw acceleration in output frame, Z component
8	rot_x	Float (.6)	rad/s	0.023436	Raw angular velocity in output frame, X component
9	rot_y	Float (.6)	rad/s	0.007723	Raw angular velocity in output frame, Y component
10	rot_z	Float (.6)	rad/s	0.002131	Raw angular velocity in output frame, Z component

Table 7.15.: FP_A-RAWIMU message format

7.5.2. FP_A-CORRIMU

This message contains time and the IMU's acceleration and angular velocity (coordinate transformation and bias correction applied) in the sensor frame.

Example message:

```
$FP,CORRIMU,1,2197,126191.777855,-0.195224,0.393969,9.869998,0.013342,
    -0.004620,-0.000728*7D\r\n
```

Message fields:

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	CORRIMU	Message type, always RAWIMU for this message
2	msg_version	Numeric	-	1	Message version, always 1 for this version of the RAWIMU message
3	gps_week	Numeric	-	2197	GPS week number, range 0–9999
4	gps_tow	Float (.6)	s	126191.777855	GPS time of week, range 0.000–604799.999
5	acc_x	Float (.6)	m/s2	-0.195224	Raw acceleration in output frame, X component
6	acc_y	Float (.6)	m/s2	0.393969	Raw acceleration in output frame, Y component
7	acc_z	Float (.6)	m/s2	9.869998	Raw acceleration in output frame, Z component
8	rot_x	Float (.6)	rad/s	0.013342	Raw angular velocity in output frame, X component
9	rot_y	Float (.6)	rad/s	-0.004620	Raw angular velocity in output frame, Y component
10	rot_z	Float (.6)	rad/s	-0.000728	Raw angular velocity in output frame, Z component

Table 7.16.: FP_A-CORRIMU message format

7.5.3. FP_A-TF_POIIMUH

This is the transformation from the IMU horizontal frame to the sensor's current attitude (**roll, pitch, yaw**).

Example message:

```
$FP,TF,2,2233,315835.000000,VRTK,CAM,-0.00000,-0.00000,-0.00000,1.000000,
    0.000000,0.000000,0.000000*6B\r\n
```

```
$FP,TF,2,2233,315835.000000,POI,VRTK,-0.99301,-2.01395,-2.99298,0.999995,
    -0.002616,-0.001748,-0.000868*52\r\n
```

Message fields:

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	TF	Message type, always TF for this message
2	msg_version	Numeric	-	2	Message version, always 2 for this version of the TF message
3	gps_week	Numeric	-	2233	GPS week number, range 0–9999
4	gps_tow	Float (.6)	s	315835.000000	GPS time of week, range 0.000–604799.999999
5	frame_a	String	-	POI	Target frame (maximum 8 characters: A-Z and 0-9)
6	frame_b	String	-	VRTK	Initial frame (maximum 8 characters: A-Z and 0-9)
7	translation_x	Float (.5)	m	-0.99301	Translation, X component
8	translation_y	Float (.5)	m	-2.01395	Translation, Y component
9	translation_z	Float (.5)	m	-2.99298	Translation, Z component
10	orientation_w	Float (.6)	-	0.999995	Rotation in quaternion, W component
11	orientation_x	Float (.6)	-	-0.002616	Rotation in quaternion, X component
12	orientation_y	Float (.6)	-	-0.001748	Rotation in quaternion, Y component
13	orientation_z	Float (.6)	-	-0.000868	Rotation in quaternion, Z component

Table 7.17.: FP_A-TF_POIIMUH message format

7.5.4. NOV_B-RAWIMU

This message contains time and the IMU's acceleration and angular velocity (raw value, no bias correction, only coordinate transformation applied) in the sensor frame. This message conveys similar information to the FP_A-RAWIMU message, only in another format.

Message fields:

#	Offset	Field	Type	Unit	Description
-	0	sync1	uint8_t	-	Sync byte 1 (always 0xaa)
-	1	sync2	uint8_t	-	Sync byte 2 (always 0x44)
-	2	sync3	uint8_t	-	Sync byte 3 (always 0x12)
-	3	header_len	uint8_t	bytes	Length of the header (always 28)
-	4	msg_id	uint16_t	-	Message ID, always 268 for this message
-	6	msg_type	uint8_t	-	See documentation
-	7	reserved1	uint8_t	-	Reserved, ignore
-	8	payload_len	uint8_t	bytes	Payload size, always 40 for this message
-	10	reserved2	uint16_t	-	Reserved, ignore
-	12	reserved3	uint8_t	-	Reserved, ignore
1	13	time_status	uint8_t	-	See documentation
2	14	gps_wno	uint16_t	-	GPS week number
3	16	gps_tow	int32_t	ms	GPS time of week
-	20	reserved4	uint32_t	-	Reserved, ignore
-	24	reserved5	uint16_t	-	Reserved, ignore
-	26	reserved6	uint16_t	-	Reserved, ignore
4	28	gnss_week	uint32_t	-	GNSS (GPS) week number (= gps_week)
5	32	gnss_tow	double	s	GNSS (GPS) time of week (= gps_tow ast 10-3)
6	40	reserved7	uint32_t	-	Reserved, ignore
7	44	x_accel	int32_t	m/s	Change in velocity along x-axis
8	48	y_accel	int32_t	m/s	Change in velocity along y-axis
9	52	z_accel	int32_t	m/s	Change in velocity along z-axis
7	56	z_gyro	int32_t	m/s	Change in angle count around z-axis
8	60	y_gyro	int32_t	m/s	Change in angle count around y-axis
9	64	x_gyro	int32_t	m/s	Change in angle count around x-axis
-	68	checksum	uint32_t	-	CRC32 checksum (see protocol documentation)

Table 7.18.: NOV_B-RAWIMU message format

7.6. GNSS output

7.6.1. NMEA-GP-GGA

This message contains time, date, position (in LLH coordinates), fix quality, number of satellites, and horizontal dilution of precision (HDOP) data provided by the GNSS receiver. This message can be one of the following:

- NMEA-GP-GGA_GNSS for combined (average) output from both receivers.
    - The time and position data are the average between GNSS1 and GNSS2.
    - The fix type, number of satellites, and PDOP values are from the 'best' receiver.
    - The message is only valid if both receivers have a valid fix.
- NMEA-GP-GGA_GNSS1 for output from GNSS 1 receiver.
    - This is the standard NMEA GGA for GNSS1.
- NMEA-GP-GGA_GNSS2 for output from GNSS 2 receiver.
    - This is the standard NMEA GGA for GNSS2.
- NMEA-GP-GGA_FUSION for output from Fusion.
    - The time and position data is from the Fusion solution.
    - The fix type, number of satellites, and PDOP values are from the 'best' GNSS receiver.
    - The message is only valid if Fusion data is valid (Fusion initialized and propagation successful).

Note that these messages do not contain any information that would let one tell them apart. Do not enable more than one of these in the same port.

Example message:

```
$GPGGA,151229.40,4723.54108,N,00826.88485,E,4,12,00.98,473.5,M,,,,*3A\r\n
```

Example message with non-standard high-precision fields:

```
$GPGGA,151229.3999,4723.5410795,N,00826.8848463,E,4,31,00.98,473.5368,
    M,,,,*0F\r\n
```

Message fields:

#	Field	Format	Unit	Example	Description
1	time	hhmmss.ss(ss)	-	151229.40	UTC time (hours, minutes and seconds)
2	lat	ddmm.mmmmm(mm)	-	4723.54108	Latitude
3	lat_ns	Character	-	N	Latitude north (N) or south (S) indicator
4	lon	dddmm.mmmmm(mm)	-	00826.88485	Longitude
5	lon_ew	Character	-	E	Longitude east (E) or west (W) indicator
6	quality	Digit	-	4	Quality indicator, 4 (RTK fixed), 5 (RTK float), 1 (no RTK), 6, 0
7	num_sv	Decimal	-	12	Number of satellites, range 00-12
8	hdop	Float (.2)	-	00.98	Horizontal dilution of precision, range 00.10-99.99
9	alt	Float (.1/.4)	m	473.5	Altitude (above ellipsoid)
10	alt_unit	Character	-	M	Altitude unit, always M (metres)
11	sep	-	-		Geoid separation, always null
12	sep_unit	-	-		Geoid separation unit, always null
13	diff_age	-	-		Age of differential data, always null
14	diff_sta	-	-		DGPS station ID, always null

Table 7.19.: NMEA-GP-GGA message format

7.6.2. NMEA-GP-RMC

This message contains time, date, position (in LLH coordinates), positioning mode, course over ground (COG), and speed (SOG) data provided by the GNSS receiver. RMC is the recommended minimum navigation data to be provided by a GNSS receiver. This message can be one of the following:

- NMEA-GP-RMC_GNSS for combined (average) output from both receivers.
    - The time and position data are the average between GNSS1 and GNSS2.
    - The fix type, number of satellites, and PDOP values are from the 'best' receiver.
    - The course over ground and speed over ground are computed from the average position and velocity between GNSS1 and GNSS2. Note that this is not the same as the average of the individual COG and SOG.
    - The message is only valid if both receivers have a valid fix.
- NMEA-GP-RMC_GNSS1 for output from GNSS 1 receiver.
    - This is the standard NMEA RMC for GNSS1.
- NMEA-GP-RMC_GNSS2 for output from GNSS 2 receiver.
    - This is the standard NMEA RMC for GNSS2.
- NMEA-GP-RMC_FUSION for output from Fusion.
    - The time, position, course over ground, and speed over ground data are from the Fusion solution.
    - The fix type is from the 'best' GNSS receiver.
    - The message is only valid if Fusion data is valid (Fusion initialized and propagation successful).

Note that these messages do not contain any information that would let one tell them apart. Do not enable more than one of these in the same port.

Example message:

```
$GPRMC,151227.40,A,4723.54036,N,00826.88672,E,0.0,81.6,111022,,,R*7C\r\n
```

Example message with non-standard high-precision fields:

```
$GPRMC,151227.3997,A,4723.5403567,N,00826.8867153,E,0.00000,81.6172,
    111022,,,R*4F\r\n
```

Message fields:

#	Field	Format	Unit	Example	Description
1	time	hhmmss.ss(ss)	-	151227.40	UTC time (hours, minutes and seconds)
2	status	Character	-	A	Data validity status, A (data valid) or V (not valid)
3	lat	ddmm.mmmmm(mm)	-	4723.54036	Latitude
4	lat_ns	Character	-	N	Latitude north (N) or south (S) indicator
5	lon	dddmm.mmmmm(mm)	-	00826.88672	Longitude
6	lon_ew	Character	-	E	Longitude east (E) or west (W) indicator
7	speed	Float (.1/.4)	knots	0.0	Speed over ground
8	course	Float (.1/.4)	deg	81.6	Course over ground
9	date	ddmmyy	-	111022	UTC date (day, month and year)
10	magvar	-	-		Magnetic variation, always null
11	magvar_ew	-	-		Magnetic variation east or west indicator, always null
12	mode	Character	-	R	Positioning system mode indicator, R (RTK fixed), F (RTK float), A (no RTK), E, N

Table 7.20.: NMEA-GP-RMC message format

7.6.3. FP_A-GNSSANT

This message contains information regarding the antennas' state and power supply. Under normal conditions, the GNSS state should be OK, and the power should be ON.

Example message:

```
$FP,GNSSANT,1,2234,305129.200151,short,off,0,open,on,28*65\r\n
```

Message fields:

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	GNSSANT	Message type, always GNSSANT for this message
2	msg_version	Numeric	-	1	Message version, always 1 for this version of the GNSSANT message
3	gps_week	Numeric	-	2234	GPS week number, range 0–9999
4	gps_tow	Float (.6)	s	305129.200151	GPS time of week, range 0.000–604799.999999
5	gnss1_state	String	-	short	GNSS1 antenna state, see below
6	gnss1_power	String	-	off	GNSS1 antenna power, see below
7	gnss1_age	Numeric	s	0	Time since gnss1_state or gnss1_power information last changed, 0–604800, or null
8	gnss2_state	String	-	open	GNSS1 antenna state, see below
9	gnss2_power	String	-	on	GNSS1 antenna power, see below
10	gnss2_age	Numeric	s	28	Time since gnss2_state or gnss2_power information last changed, 0–604800, or null

Table 7.21.: FP_A-GNSSANT message format

Values for gnss1_state and gnss2_state:

Value	Description
null	State not available, for example during initialization of the receiver
ok	Antenna detected and good
open	No antenna detected (e.g. no antenna connected, or connected via DC-blocking, such as a power-splitter)
short	Antenna short circuit detected

Table 7.22.: GNSS state values

Values for gnss1_power and gnss2_power:

Value	Description
null	Power status not available, for example during initialization of the receiver
on	Antenna power supply is on
off	Antenna power supply is off

Table 7.23.: GNSS power values

7.6.4. FP_A-GNSSCORR

This message contains statistics regarding the correction data tracked by the GNSS receivers, such as the number of L1 and L2 signals, the fix type, the average correction data latency, update rate, data rate, and message rate, as well as the correction station position and baseline.

Example messages:

```
$FP,GNSSCORR,1,2237,250548.999744,8,25,18,8,25,18,,,,,,,,,*24
```

```
$FP,GNSSCORR,1,2237,250035.999865,8,25,18,8,25,18,0.2,1.0,0.8,5.3,2,
    47.366986804,8.532965023,481.1094,7254*3F
```

Message fields:

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	GNSSANT	Message type, always GNSS-CORR for this message
2	msg_version	Numeric	-	1	Message version, always 1 for this version of the GNSSCORR message
3	gps_week	Numeric	-	2237	GPS week number, range 0–9999
4	gps_tow	Float (.6)	s	250035.999865	GPS time of week, range 0.000–604799.999999
5	gnss1_fix	Numeric	-	8	Fix status of GNSS1 receiver, see below
6	gnss1_nsig_l1	Numeric	-	25	Number of L1 signals with correction data tracked by GNSS1 receiver, range 0–100
7	gnss1_nsig_l2	Numeric	-	18	Number of L2 signals with correction data tracked by GNSS1 receiver, range 0–100
8	gnss2_fix	Numeric	-	8	Fix status of GNSS2 receiver, see below
9	gnss2_nsig_l1	Numeric	-	25	Number of L1 signals with correction data tracked by GNSS2 receiver, range 0–100
10	gnss2_nsig_l2	Numeric	-	18	Number of L2 signals with correction data tracked by GNSS2 receiver, range 0–100
11	corr_latency	Float (.1)	s	0.2	Average correction data latency (10s window), range 0.0–99.9 or null
12	corr_update_rate	Float (.1)	Hz	1.0	Average correction update rate (10s window), range 0.0–10.0 or null
13	corr_data_rate	Float (.1)	KiB/s	0.8	Average correction data rate (10s window), range 0.0–50.0 or null
14	corr_msg_rate	Float (.1)	msgs/s	5.3	Average correction message rate (10s window), range 0-50 - or null
15	sta_id	Numeric	-	2	Correction station ID, range 0–4095 or null
16	sta_lat	Float (.9)	deg	47.366986804	Correction station latitude, range -90.000000000–90.000000000, > 0 for North, < 0 for South, or null

#	Field	Format	Unit	Example	Description
17	sta_lon	Float (.9)	deg	8.532965023	Correction station longitude, range -180.000000000–180.000000000, > 0 for East, < 0 for West, or null
18	sta_height	Float (.4)	m	481.1094	Correction station ellipsoidal height, range -1000.0000-50000.0000 or null
19	sta_dist	Numeric	m	7254	Correction station baseline length (approximate 3d distance), range 0–100000 or null

Table 7.24.: FP_A-GNSSCORR message format

Values for gnss1_fix and gnss2_fix:

Value	Description
0	Unknown
1	No fix
2	Dead-reckoning only
3	Time-only fix
4	Single 2D fix
5	Single 3D fix
6	Single 3D fix with dead-reckoning
7	RTK float fix
8	RTK fixed fix

Table 7.25.: GNSS fix values

7.6.5. NOV_B-HEADING2

This message contains only the heading from GNSS measurements; no Fusion data is used here. The heading is defined as the angle from the true North of GNSS1 to the GNSS2 vector in a clockwise direction.

Message fields:

#	Offset	Field	Type	Unit	Description
-	0	sync1	uint8_t	-	Sync byte 1 (always 0xaa)
-	1	sync2	uint8_t	-	Sync byte 2 (always 0x44)
-	2	sync3	uint8_t	-	Sync byte 3 (always 0x12)
-	3	header_len	uint8_t	bytes	Length of the header (always 28)
-	4	msg_id	uint16_t	-	Message ID, always 1335 for this message
-	6	msg_type	uint8_t	-	See protocol documentation
-	7	reserved1	uint8_t	-	Reserved, ignore
-	8	payload_len	uint8_t	bytes	Payload size, always 48 for this message
-	10	reserved2	uint16_t	-	Reserved, ignore
-	12	reserved3	uint8_t	-	Reserved, ignore
1	13	time_status	uint8_t	-	See protocol documentation
2	14	gps_wno	uint16_t	-	GPS week number
3	16	gps_tow	int32_t	ms	GPS time of week
-	20	reserved4	uint32_t	-	Reserved, ignore
-	24	reserved5	uint16_t	-	Reserved, ignore
-	26	reserved6	uint16_t	-	Reserved, ignore
4	28	sol_status	uint32_t	-	Solution status, see below
5	32	pos_type	uint32_t	-	Positioning mode, see below
6	36	length	float	m	Baseline length
7	40	heading	float	deg	Heading
8	44	pitch	float	deg	Pitch
-	48	reserved7	uint32_t	-	Reserved, ignore
-	52	reserved8	uint32_t	-	Reserved, ignore
-	56	reserved9	uint32_t	-	Reserved, ignore
-	60	reserved10	uint32_t	-	Reserved, ignore
-	64	reserved11	uint32_t	-	Reserved, ignore
9	68	num_svs	uint8_t	-	Number of satellites tracked
10	69	num_sol_svs	uint8_t	-	Number of satellites used in solution
-	70	reserved12	uint16_t	-	Reserved, ignore
11	72	sol_source_msk	uint8_t	-	Bitfield:
-		sol_source	- bits 3…2	-	Solution source, see below
12	73	ext_sol_stat	uint8_t	-	Extended solution status, see below
13	74	gal_bds_sig_msk	uint8_t	-	Galileo and BeiDou signal mask, see below
14	75	gps_glo_sig_msk	uint8_t	-	GPS and GLONASS signal mask, see below

Table 7.26.: NOV_B-HEADING2 message format

Value	Description
0	Solution computed
1	Insufficient observations

Table 7.27.: Solution status

Value	Description
0	No fix
50	RTK fixed fix
34	RTK float fix
16	Single 3D fix
56	INS and RTK fixed fix
55	INS and RTK float fix
53	INS and single 3D fix
19	INS only

Table 7.28.: Positioning mode

Bit	Description
bit 0 (0x01)	Solution verified

Table 7.29.: Extended solution status

Bit	Description
bit 0 (0x01)	Galileo E1
bit 2 (0x04)	Galileo E5B
bit 4 (0x10)	BeiDou B1I
bit 5 (0x20)	BeiDou B2I

Table 7.30.: Galileo and BeiDou signal mask

Bit	Description
bit 0 (0x01)	GPS L1CA
bit 1 (0x02)	GPS L2C
bit 4 (0x10)	GLONASS L1OF
bit 5 (0x20)	GLONASS L2OF

Table 7.31.: GPS and GLONASS signal mask

7.6.6. NOV_B-BESTGNSSPOS

This message contains the best available GNSS position (without Fusion) computed by the GNSS receiver. In addition, it reports several status indicators, including differential age, which helps predict anomalous behavior brought about by outages in differential corrections. A differential age of 0 indicates that no differential correction was used. With the system operating in an RTK mode, this log reflects the latest low-latency solution for up to 60 seconds after the reception of the last base station observations. After these 60 seconds, the position reverts to the best solution available, and the standard deviation fields reflect the accuracy degradation. This message can be either:

- NOV_B-BESTGNSSPOS_GNSS1 for output from GNSS 1 receiver

- NOV_B-BESTGNSSPOS_GNSS2 for output from GNSS 2 receiver

Message field:

#	Offset	Field	Type	Unit	Description
-	0	sync1	uint8_t	-	Sync byte 1 (always 0xaa)
-	1	sync2	uint8_t	-	Sync byte 2 (always 0x44)
-	2	sync3	uint8_t	-	Sync byte 3 (always 0x12)
-	3	header_len	uint8_t	bytes	Length of the header (always 28)
-	4	msg_id	uint16_t	-	Message ID, always 1429 for this message
-	6	msg_type	uint8_t	-	See protocol documentation
-	7	reserved1	uint8_t	-	Reserved, ignore
-	8	payload_len	uint8_t	bytes	Payload size, always 72 for this message
-	10	reserved2	uint16_t	-	Reserved, ignore
-	12	reserved3	uint8_t	-	Reserved, ignore
1	13	time_status	uint8_t	-	See protocol documentation
2	14	gps_wno	uint16_t	-	GPS week number
3	16	gps_tow	int32_t	ms	GPS time of week
-	20	reserved4	uint32_t	-	Reserved, ignore
-	24	reserved5	uint16_t	-	Reserved, ignore
-	26	reserved6	uint16_t	-	Reserved, ignore
4	28	sol_status	uint32_t	-	Solution status, see below
5	32	pos_type	uint32_t	-	Positioning mode, see below
6	36	lat	double	deg	Latitude
7	44	lon	double	deg	Longitude
8	52	height	double	m	Ellipsoidal height
-	60	reserved7	uint32_t	-	Reserved, ignore
9	64	datum	unit32_t	-	Datum, always 61 (= WGS84)
10	68	std_lat	float	m	Latitude standard deviation
11	72	std_lon	float	m	Longitude standard deviation
12	76	std_height	float	m	Height standard deviation
-	80	reserved8	uint32_t	-	Reserved, ignore
-	84	reserved9	uint32_t	-	Reserved, ignore
13	88	sol_age	float	s	Approximate age of solution
14	92	num_svs	uint8_t	-	Number of satellites tracked
15	93	num_sol_svs	uint8_t	-	Number of satellites used in solution
15	94	num_sol_l1_svs	uint8_t	-	Number of satellites with L1 signals used in solution
16	95	num_sol_l2_svs	uint8_t	-	Number of satellites with L2 signals used in solution
-	96	reserved10	uint8_t	-	Reserved, ignore
17	97	ext_sol_stat	uint8_t	-	Extended solution status, see below
18	98	gal_bds_sig_msk	uint8_t	-	Galileo and BeiDou signal mask, see below
19	99	gps_glo_sig_msk	uint8_t	-	GPS and GLONASS signal mask, see below
-	100	checksum	uint32_t	-	CRC32 checksum (see protocol documentation)

Table 7.32.: NOV B-BESTGNSSPOS message format

Value	Description
0	Solution computed
1	Insufficient observations

Table 7.33.: Solution status

Value	Description
0	No fix
50	RTK fixed fix
34	RTK float fix
16	Single 3D fix
56	INS and RTK fixed fix
55	INS and RTK float fix
53	INS and single 3D fix
19	INS only

Table 7.34.: Positioning mode

Bit	Description
bit 0 (0x01)	Solution verified

Table 7.35.: Extended solution status

Bit	Description
bit 0 (0x01)	Galileo E1
bit 2 (0x04)	Galileo E5B
bit 4 (0x10)	BeiDou B1I
bit 5 (0x20)	BeiDou B2I

Table 7.36.: Galileo and BeiDou signal mask

Bit	Description
bit 0 (0x01)	GPS L1CA
bit 1 (0x02)	GPS L2C
bit 4 (0x10)	GLONASS L1OF
bit 5 (0x20)	GLONASS L2OF

Table 7.37.: GPS and GLONASS signal mask

7.7. Text output

Text output is the irregular output of text messages (strings). There are four levels:

- **ERROR** – Errors. An error is something from which the sensor could not recover or something that renders the sensor unusable in normal circumstances.

- **WARNING** – Warnings. A warning is something from which the sensor could recover or something that does not render the sensor unusable in normal circumstances (i.e., the sensor operated to specifications). But it can for example hint at degraded performance.

- **INFO** – General information.

- **DEBUG** – Debug information, only useful to Fixposition.

For example, on boot, the sensor outputs a "boot screen". While it does output it to any port where the **FP_A-TEXT_INFO** message is enabled, it is only practical to observe this output on UART and CANSTR ports. It is not possible to connect to the TCP ports in time to observe this output. The boot screen is as follows:

```
$FP,TEXT,1,INFO,Fixposition AG - www.fixposition.com*09\r\n
$FP,TEXT,1,INFO,SW=fp_release_vr2_2.61.0_191*78\r\n
$FP,TEXT,1,INFO,HW=nav-vr2 1.2a 6d9d18*3E\r\n
```

7.7.1. FP_A-TEXT

Example messages:

```
$FP,TEXT,1,INFO,Fixposition AG - www.fixposition.com*09\r\n
$FP,TEXT,1,WARNING,high imu noise*71\r\n
$FP,TEXT,1,ERROR,cfg fail*22\r\n
$FP,TEXT,1,DEBUG,hello world*4B\r\n
```

Message fields:

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	CORRIMU	Message type, always TEXT for this message
2	msg_version	Numeric	-	1	Message version, always 1 for this version of the TEXT message
3	level	String	-	DEBUG	Level, one of: INFO, WARNING, ERROR, DEBUG
4	text	String	-	hello world	Text

Table 7.38.: FP_A-TEXT message format

Software Updates

To update the software version of the Vision-RTK 2 go to the "System → Update" panel in the web interface and drag/drop the corresponding SWU file inside the marked area (see Figure A.1).
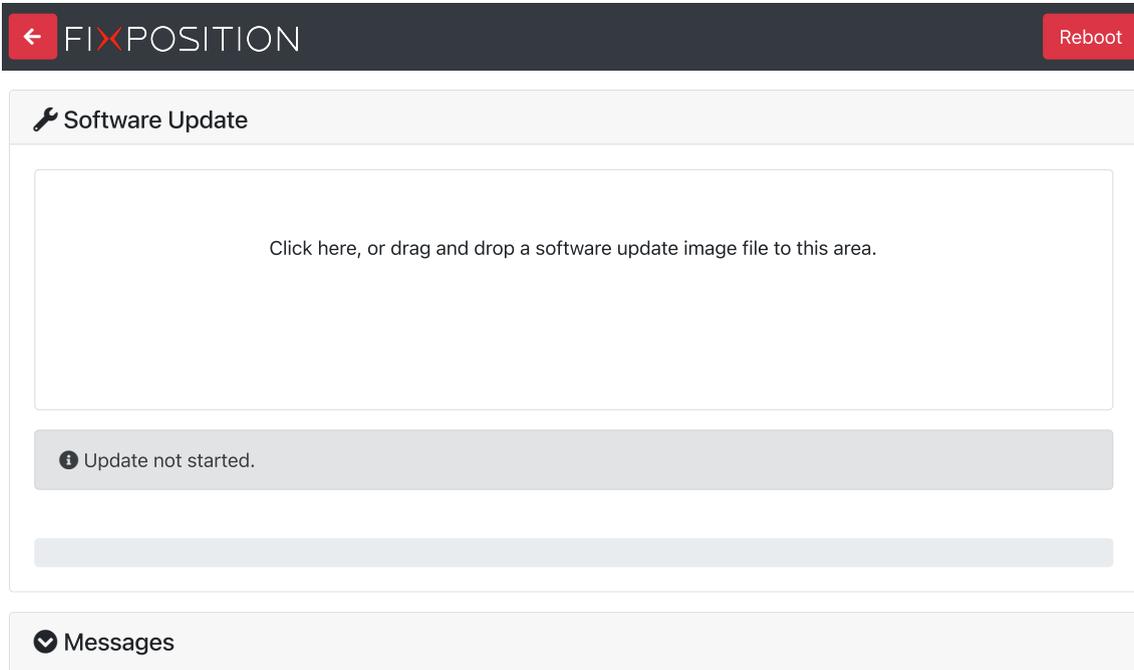
Figure A.1.: Software update in the web interface

Software updates are released every 6-10 weeks. Please email support@fixposition.com to be added to the distribution list.

We also provide Bash and Python scripts to upload the software image to the system and then wait for the notification of success or failure of the flash process. These tools can be found in the Fixposition Software Update library (`https://github.com/fixposition/fixposition_utility/tree/main/software_update`).

Note: When upgrading from 2.58.0 or earlier, the update page will get stuck at "The system will restart. Please be patient, as restarting takes about one minute." Please wait a minute and then manually open http://10.0.1.1 to return to the main interface. This issue was caused by the software update page being moved from "http://x.x.x.x:8080" to "http://x.x.x.x/update".

Fixposition CAN Frame Definition

Figure B.1.: Fixposition CAN frame definition

Coordinate Transformations
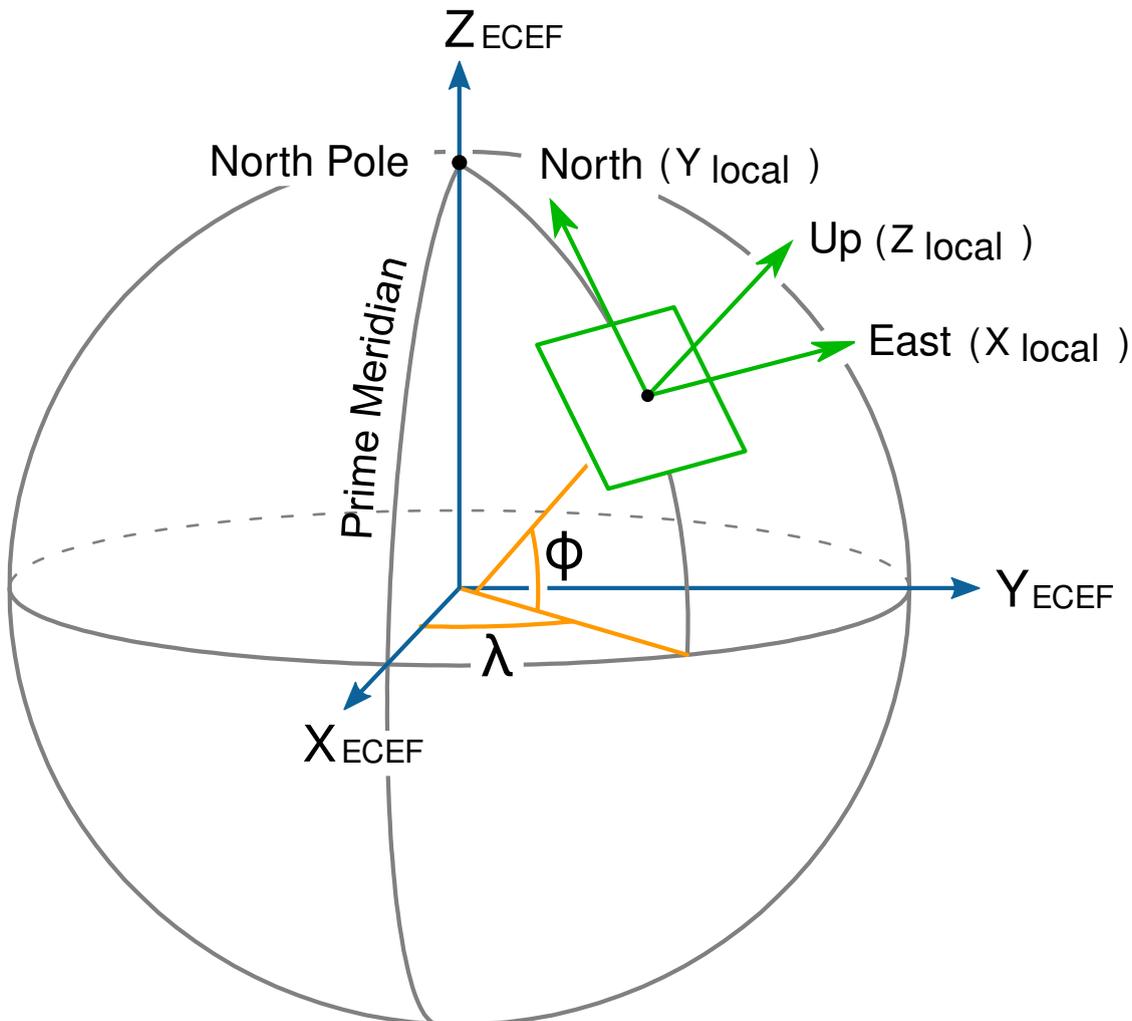
Figure C.1.: ECEF (global) and ENU (local) coordinate systems

Many different coordinate systems exist and are used to represent the pose of an object in space. Most of them can be classified into two groups: global systems, which approximate the entire Earth (e.g., WGS-84), and local systems, which use the best approximation of the local true geometrical shape of the Earth. ECEF coordinates (Earth-Centered, Earth-Fixed) are part of a global Cartesian system in which the center of the Earth is placed at the origin **<0,0,0>**. The odometry output of the VRTK2 sensor is given in ECEF coordinates (see Subsection 7.3.1).

Given that ECEF coordinates represent the pose of the sensor on a sphere, it is common to draw a tangential plane to represent the local pose of the robot. This representation is only accurate up to a certain distance from the origin (83 km on average) but makes tracking the odometry of the robot easier. For a local frame of reference, some outputs are given in ENU coordinates (East-North-Up). See Figure C.1 for a graphical explanation.

To convert the ECEF orientation of the VRTK2 into this local frame (ENU) coordinates, let's assume $\mathbf{q}_{\text{ecef} \to \text{body}}$ represents the orientation of the sensor in ECEF coordinates, which can be extracted from the pose output of the **FP_A-ODOMETRY** message. Then, the rotation from ECEF coordinates to the local frame of reference (ENU) can be computed using the current position of the sensor on the sphere **<x,y,z>**.

The Fixposition GNSS Transformation Lib(**fixposition_gnss_tf**) contains several useful functions for these space operations. For example, the function **TfEnuEcef()** takes an ECEF position coordinate and returns a rotation matrix that transforms from the ECEF plane to the ENU plane. Let's call this rotation matrix $\mathbf{R}_{\text{enu} \to \text{ecef}}$. As the orientation output of the VRTK2 sensor is represented using a quaternion, we need to first convert the rotation matrix to a quaternion $\mathbf{q}_{\text{enu} \to \text{ecef}}$. Thus, the rotation of the VRTK2 sensor in ENU coordinates can be computed as the multiplication of $\mathbf{q}_{\text{enu} \to \text{ecef}}$ times $\mathbf{q}_{\text{ecef} \to \text{body}}$. For a more in-depth explanation, please refer to the following:

- Transformations/ between/ ECEF/ and/ ENU/ coordinates

- Geographic/ coordinate/ conversion

Extract heading:
The **FP_A-ODOMETRY** message contains the position and orientation of the VRTK2 sensor in the ECEF coordinate system. For mathematical stability, the sensor employs quaternions to represent these rotations. Nonetheless, in case the user requires an Euler angles representation using Roll-Pitch-Yaw angles, it is necessary to first convert the orientation of the sensor into a local tangential coordinate system such as ENU or NED. To convert the reference frame from ECEF to ENU, we apply the following transformation:

$$\mathbf{R}_{\text{body} \to \text{enu}} = \mathbf{R}_{\text{ecef} \to \text{enu}} \cdot \mathbf{R}_{\text{body} \to \text{ecef}} \tag{C.1}$$

where $\mathbf{R}_{\text{body} \to \text{ecef}}$ is the orientation of the sensor in the ECEF frame and $\mathbf{R}_{\text{ecef} \to \text{enu}}$ is the orientation of the ECEF frame in the ENU coordinate system (a local tangential plane). To convert $\mathbf{q}_{\text{body} \to \text{enu}}$ into Euler angles, it is possible to use either a rotation matrix or quaternions. For a rotation matrix, we apply the following equations:

$$\alpha = \tan^{-1}\left(\frac{R_{21}}{R_{11}}\right), \tag{C.2}$$

$$\beta = \tan^{-1}\left(\frac{-R_{31}}{\sqrt{1 - R_{31}^2}}\right), \tag{C.3}$$

$$\gamma = \tan^{-1}\left(\frac{R_{32}}{R_{33}}\right), \tag{C.4}$$

where

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{C.5}$$

To convert a quaternion into yaw, pitch, and roll angles (ZYX order), the following equation can be used:

$$\alpha = \tan^{-1}\left(\frac{2q_x q_y + 2q_w q_z}{q_w q_w + q_x q_x - q_y q_y - q_z q_z}\right), \tag{C.6}$$

$$\beta = \sin^{-1}\left(-2q_x q_z + 2q_w q_y\right), \tag{C.7}$$

$$\gamma = \tan^{-1}\left(\frac{2q_y q_z + 2q_w q_x}{q_w q_w - q_x q_x - q_y q_y + q_z q_z}\right), \tag{C.8}$$

where $\mathbf{q} = <q_w, q_x, q_y, q_z>$. In this context, the Yaw angle with respect to the ENU (East-North-Up) frame represents the heading from East to North. To get the heading of the sensor from the North in a clockwise direction, we need to compute 90∘-yaw. Alternatively, the user can calculate the rotation in NED (North-East-Down) coordinates and then extract the Roll-Pitch-Yaw angles with respect to NED, where yaw would be directly the heading in common sense. The function **EcefPoseToEnuEul()**, in the Fixposition GNSS Transformation Lib, receives the pose of the sensor in ECEF coordinates and returns the orientation of the robot in Yaw-Pitch-Roll angles using the equations described above.

VRTK output coordinate system:
The receiver output is always in the coordinate reference system (geodetic datum) used by the correction data service. In RTK mode, the receiver does relative positioning with respect to the base coordinates provided by the correction data service. The system in which those base coordinates are is entirely up to the correction data service provider. The receiver does not need or use this information at all. This is true for all ECEF (XYZ) output. For lat/long/height output, the WGS84 parameters are used to transform ECEF XYZ to lat/lon/height. In non-RTK mode, the output position is WGS84.

Camera FOV Data and Model

- Diagonal Field of View DFOV=106.8°

- Horizontal Field of View HFOV=100°

- Vertical Field of View VFOV=64°
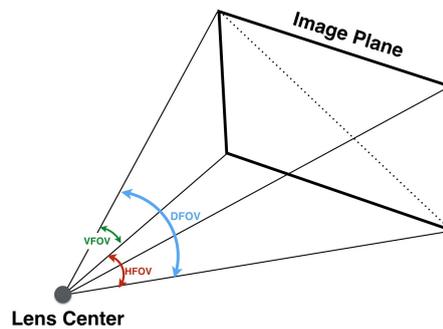
Figure D.1.: An illustration of the definition of D H V FOV
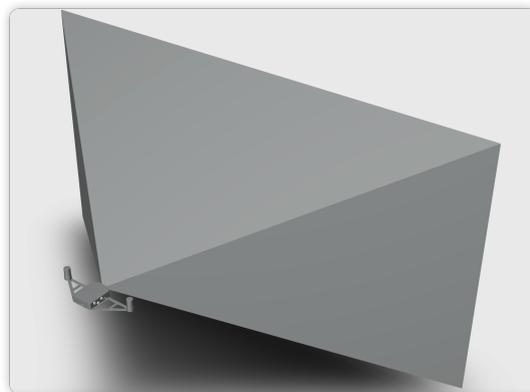
Figure D.2.: A schematic of VRTK2 FOV

The STEP file of the VRTK2 FOV model is available under user's request.

∗ The DFOV (see figure below) is the angle subtended by the diagonal of the camera sensor onto the center of the lens.

Antenna Selection

The Vision-RTK2's GNSS receivers require signals located at the L1 and L2 bands for adequate operation. Based on our internal testing, we recommend using helical antennas with a gain of around 35 dB. For reference, our evaluation kit ships with two Hi-Target AH-3232 antennas. The frequency response of the GNSS antennas should be approximately located at the [1195 - 1280] MHz and [1560 - 1610] MHz bands. On top of that, for helical antennas, we recommend a noise figure lower than 1.5 dB.

Other antenna types (e.g., patch, short helical) should be evaluated carefully depending on their placement with respect to other electrical components and the shape of the ground plane provided for them. Thus, besides using a suitable and good antenna (and appropriate cabling), the placement of the antenna is important. See below and for example the following u-blox document: `https://content.u-blox.com/sites/default/files/ZED-F9P_IntegrationManual_UBX-18010802.pdf`

Antenna assemblies that combine multiple antennas (such as, GNSS, Wi-Fi and cellular) in once casing are not recommended at all. While such antennas work fine for autonomous standard (C/A only) GNSS, they are likely to have bad performance for high-precision GNSS that require phase measurements, such as RTK. Also, the GNSS performance may depend on activity of the other antennas (Wi-Fi signal strength, cellular band used, etc.).

For further analysis, besides analyzing the position estimate performance of the sensor, one can look at the "Receiver RF AGC" value located at the advanced "GNSS status" tab in the web interface. The expected values should ideally be located between 20 and 80 percent. For additional information, see also the following U-Blox document: `https://www.u-blox.com/sites/default/files/products/documents/GNSS-Antennas_AppNote_%28UBX-15030289%29.pdf`.

Only active antennas (with a built-in LNA) are suitable for the VRTK2 sensor.

Rostopic output rate accuracy

The output rate of the rostopics published by the Vision-RTK2 is accurate. However, when looking at the recordings or plotting the received messages, the user might observe the following behavior:
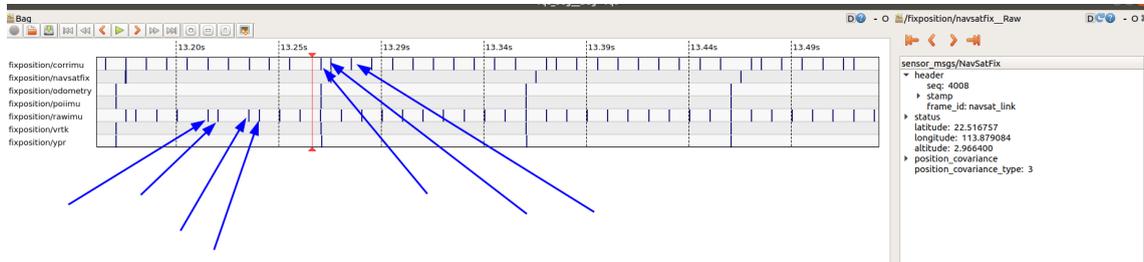

Figure F.1.: Delays in the time of arrival of ROS messages

As observed in the figure, the rate of the output messages is not constant when looking at the time of arrival. However, this is the expected behavior of the sensor, as:

1. We are not a real-time system.

2. All the communications are not guaranteed to be real-time: ROS→TCP→Customer ROS→Bag recording. There are too many steps to then look at the time of arrival of the ROS messages.

3. The user should rely on the timestamp inside the message, and not on the time of arrival.