Confidential

FIXPOSITION

Vision-RTK 2

Fixposition Positioning Sensor

INTEGRATION MANUAL

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
info@fixposition.com

Precise global positioning everywhere

Table of Contents

Document information **3**
Glossary **4**
1. System description **5**
1.1. Overview 5
1.2. High Precision Sensor Fusion 5
1.3. Supported dynamic models 6
2. Technical information **7**
2.1. System indicators 7
2.2. Physical connectors 7
2.2.1. Connectors 7
2.2.2. Ethernet 8
2.2.3. I/O connector 8
2.2.4. AUX connector 9
2.2.5. Power connector 9
2.2.6. GNSS and Wi-Fi connectors 10
2.2.7. USB 10
2.3. Sensor frame 11
2.4. Enclosure measurements 12
3. Installation guidelines **13**
3.1. Sensor setup requirements 13
4. Sensor configuration **15**
4.1. Web interface overview 15
4.2. Networking configuration 16
4.2.1. Network specifications 16
4.2.2. Data ports 17
4.2.3. Time synchronization 17
4.3. RTK/GNSS configuration 19
4.3.1 RTK/GNSS specifications 19
4.3.2. RTCM3 input messages 20
4.4. Local NTRIP caster 21
4.5. Sensor fusion configuration 21
4.6. Wheel speed configuration 22
4.6.1. Embedded web interface panel 22
4.6.2. Configuring an auxiliary sensor 22
4.6.3. Passenger car OBD-2 23
4.6.4. UART input 24
4.6.5. Raw CAN bus logging 25
4.6.6. Verify the messages 25

4.7. Output messages configuration 26
4.8. RTK/GNSS solution types 27
4.9. Recording data 27
4.10. IMU calibration 28
4.11. ROS driver installation 29
5. Output messages 30
5.1. Fixposition messages 30
5.1.1. Odometry message 30
5.1.2. LLH message 32
5.1.3. RAWIMU message 32
5.1.4. CORRIMU message 33
5.1.5. TF message 33
5.2. NMEA messages 34
5.2.1. NMEA LLH 34
5.2.2. NMEA HDT 34
Appendix 35
A. Software updates 35

Document information

Title	Vision-RTK 2
Subtitle	Fixposition Vision-RTK 2 Positioning Sensor
Document type	Integration manual
Version number	1.3
Published Date	October 2022
Disclosure restriction	Confidential/NDA

Product status	Corresponding content status
Engineering sample	Data based on early testing. Revised and supplementary data will be published later.

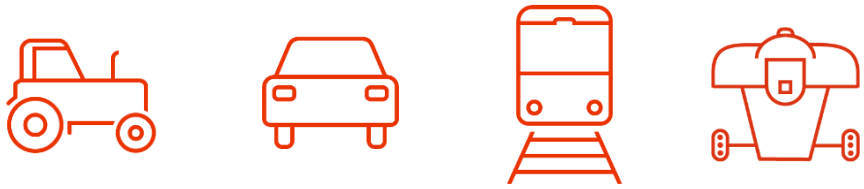
Glossary

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
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 Comission for Maritime Services
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver-Transmitter
UAV	Unmanned Aerial Vehicle
UTC	Universal Time Coordinate
VRS	Virtual Reference Station
VRTK2	Vision-RTK 2
WGS-84	World Geodetic System 1984

1. System description

1.1. Overview

The Vision-RTK 2 is a high-end sensor fusion solution that provides real-time high-accuracy pose information in all scenarios, including in 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 IMU to provide continuous high-accuracy positioning even in extended outages GNSS environments. The system can also receive information from additional auxiliary sensors to increase performance whenever required.



Heavy machinery Automotive Railway Ground robot

The Vision-RTK 2 is the ideal solution for applications such as robotics, precision agriculture, autonomous vehicles, infrastructure mapping, outdoor industrial applications and many others.

1.2. High Precision Sensor Fusion

By fusing visual odometry with several sensor data streams, Vision-RTK 2 provides an accurate and reliable position, velocity and orientation measurements including in extended GNSS outages. The sensor fusion data streams are summarized in the picture below:

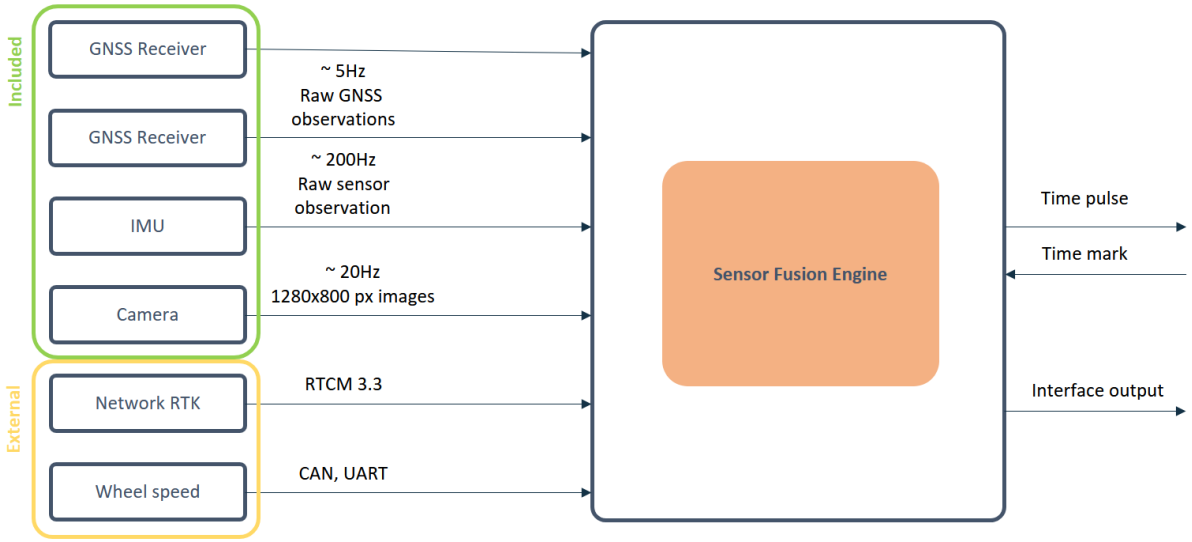


Figure 1 Sensor fusion dataflow diagram

1.3. Supported dynamic models

Vision-RTK 2 supports several navigation modes to adjust to different platforms. These modes are meant to capture the data streams from a platform with specific dynamic conditions and thus apply certain restrictions to them:

Mode	Application	Velocity range	Angular velocity range	Acceleration range
Automotive	With similar dynamics of a passenger car.	±22 m/s	0.5 rad/s	±4 m/s^2
Handheld	With similar dynamics of a human gait.	±3 m/s	1.5 rad/s	±4 m/s^2
Lawnmower	With similar dynamics of a lawnmower robot.	±3 m/s	1.0 rad/s	±4 m/s^2
Slow	With similar dynamics of a slow-moving passenger car.	±3 m/s	0.5 rad/s	±4 m/s^2

Table 1 Supported sensor fusion modes

To reach optimal performance in all conditions, it is recommended to enable an auxiliary wheel speed sensor. All the above modes are compatible with a wheel odometry input data stream.

2. Technical information

2.1. System indicators

The Vision-RTK 2 has three LED’s to indicate the current status of the sensor.

- Green LED: indicates that the sensor is powered.
- Amber LED: indicates CPU activity with an intermittent blinking.
- Red LED: indicates an error.

2.2. Physical connectors

2.2.1. Connectors

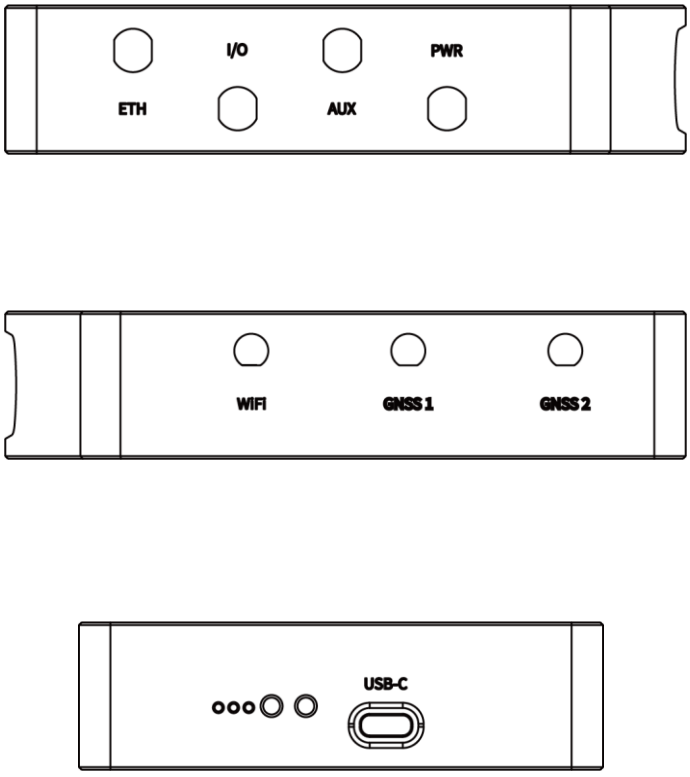


Figure 2 Vision-RTK 2 connectors overview

2.2.2. Ethernet

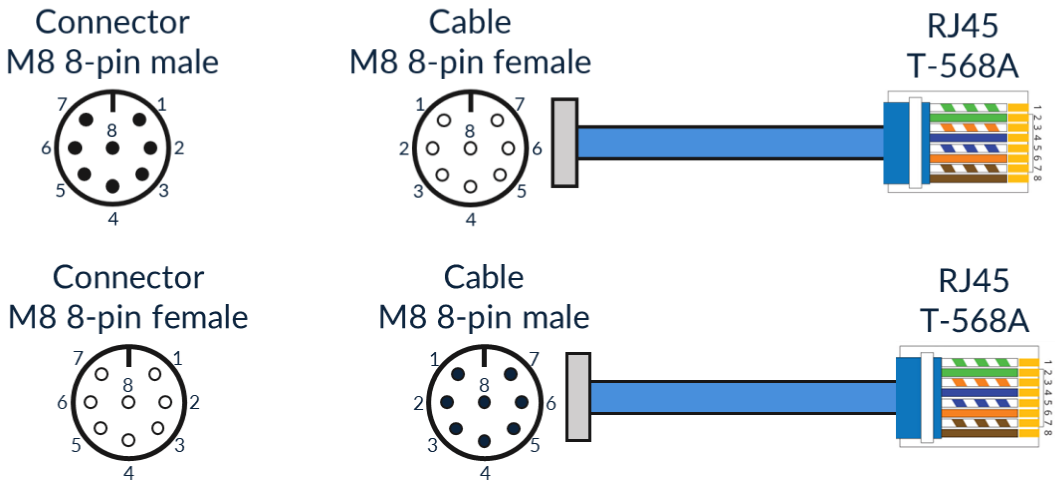


Figure 3 Ethernet connector and cable

The Vision-RTK 2 exists in 2 variants. Variants manufactured before October 2022 are equipped with male connectors on the sensor, future sensors are equipped with female connectors.

2.2.3. I/O connector

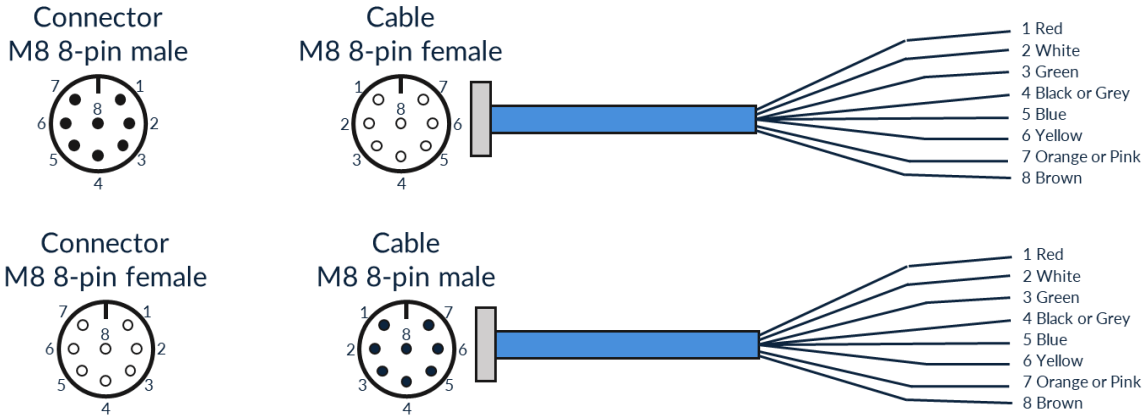


Figure 4 I/O connector pin overview

The Vision-RTK 2 exists in 2 variants. Variants manufactured before October 2022 are equipped with male connectors on the sensor, future sensors are equipped with female connectors.

Pin	Wire color	Symbol	Description
1	Red	CANH	CAN High
2	White	CANL	CAN Low
3	Green	MAIN_RST	Reset pin
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 2 I/O pin definition

2.2.4. AUX connector

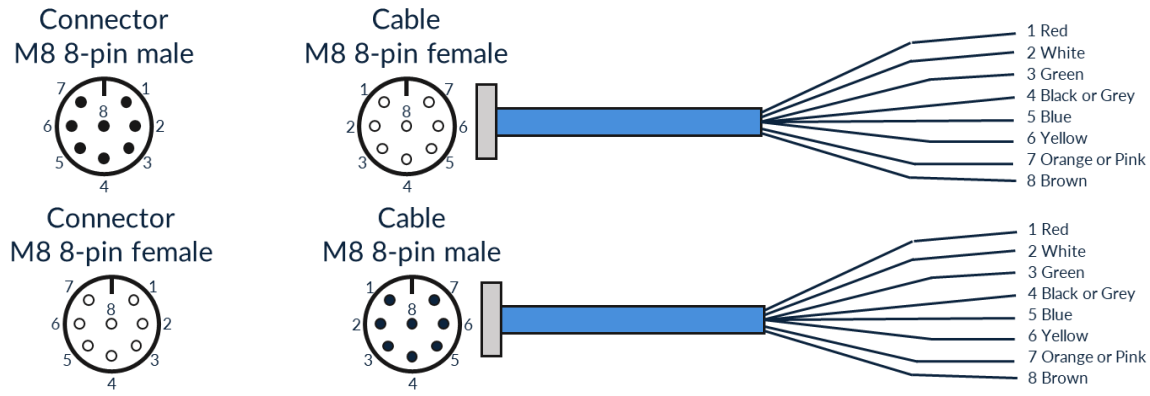


Figure 5 AUX connector pin overview

The Vision-RTK 2 exists in 2 variants. Variants manufactured before October 2022 are equipped with male connectors on the sensor, future sensors are equipped with female connectors.

Pin	Wire color	Symbol	Description
1	Red	+5V1	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 AUX pin definition

2.2.5. Power connector

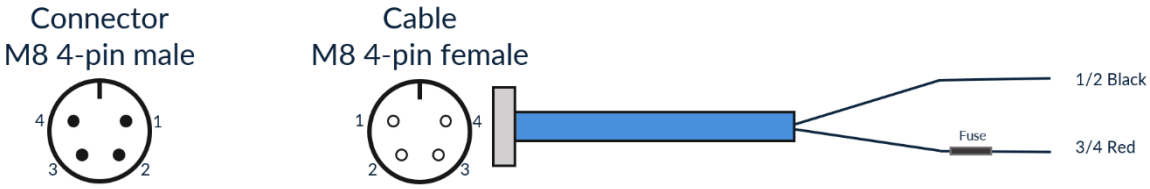


Figure 6 Power connector pin overview

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

Table 4 Power pin definition

2.2.6. GNSS and Wi-Fi connectors

- The Vision-RTK 2 is equipped with two GNSS receivers which can be connected to antennas via the female SMA connectors labelled GNSS 1 and GNSS 2
- The Vision-RTK 2 can have its Wi-Fi range significantly increased by connecting a Wi-Fi antenna to the female RP-SMA connector labelled WiFi

2.2.7. USB

The Vision-RTK 2 is equipped with a USB-C port. This can be used to connect an external drive for data recording (see 4.9. Recording data)

2.3. Sensor frame

The origin of the sensor's reference frame is located in the Fixposition logo. The reference frame components are shown in the picture below. Note that all the functionalities and messages of the sensor use this as their default reference frame.

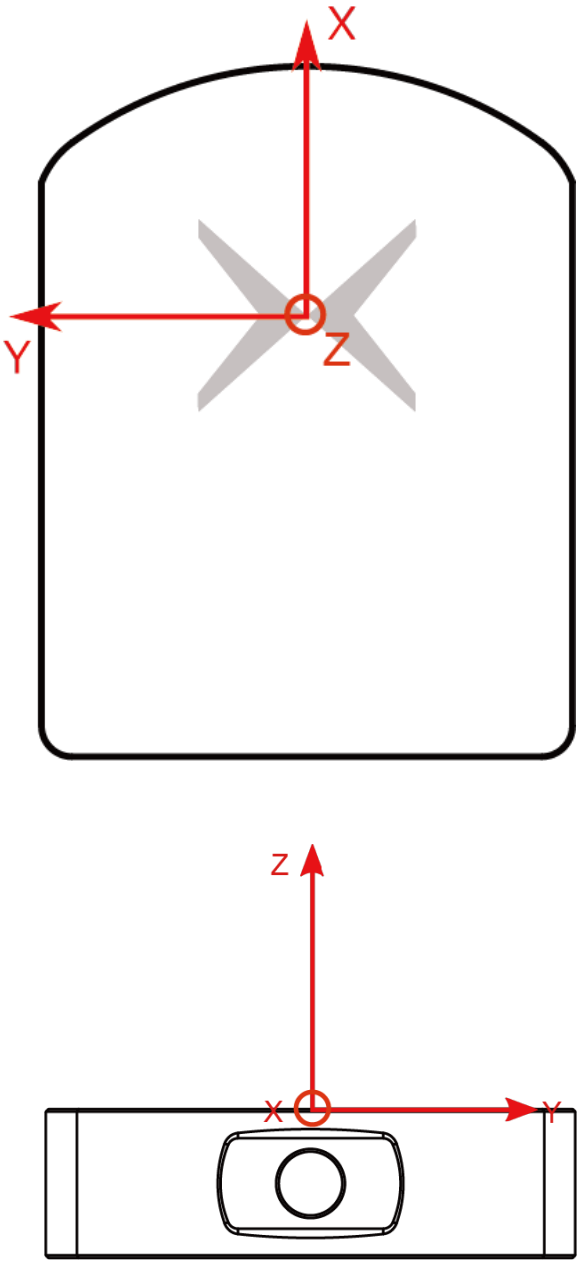


Figure 7 Sensor frame

2.4. Enclosure measurements

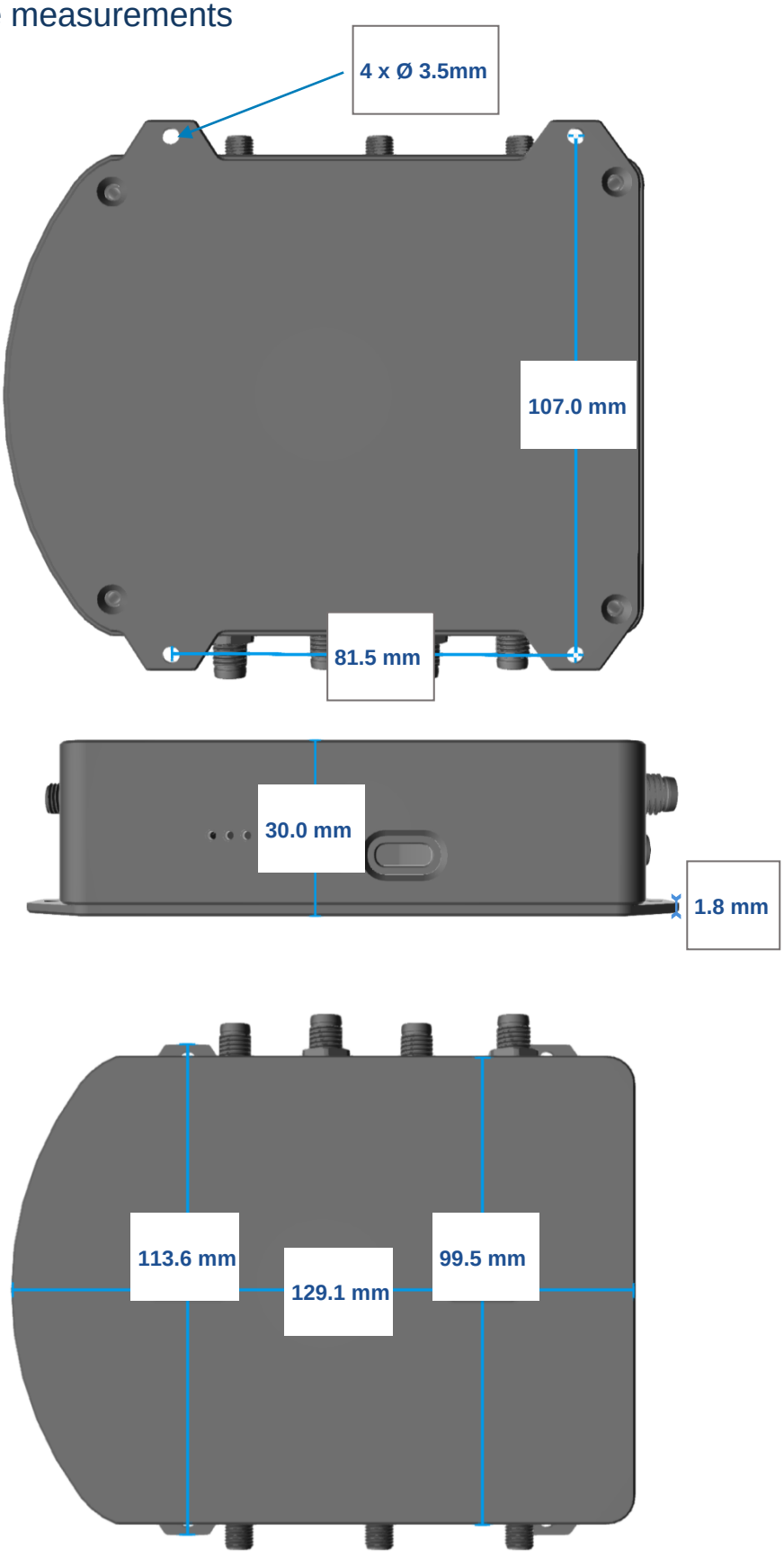


Figure 8 Measurements of Vision-RTK 2

3. Installation guidelines

3.1. Sensor setup requirements

In order for the proper functioning and optimal performance of the Vision-RTK 2 it is essential that certain setup requirements are fulfilled:

Mounting:

- The Vision-RTK 2 must be securely and firmly mounted to the vehicle body
- The Vision-RTK 2 should be mounted as to maximize the number of features from static objects that appear and disappear from the scene while the platform is travelling. Its view should not be dominated by featureless scenes like the sky and obstructions that move with the platform (e.g., a car hood) should be minimized (see Figures 9-1 and 9-2). If some of the obstructions remain in the camera scene, this can be removed in the *Camera Live Stream* section (see Figure 11) using the embedded web interface.

Figure 9-1 Camera view of the Vision-RTK 2 setup in a ground robot. Note that the horizon line lies in the middle of the camera frame.

Figure 9-2 Camera view of the Vision-RTK 2 setup in a tractor. Note that the tractor hood covers almost half of the camera frame, thus preventing correct feature acquisition.

- The antennas must be securely and firmly mounted to the vehicle body (on the same rigid body as the Vision-RTK 2). They must be mounted to the manufacturers specifications and pointing upwards, i.e., perpendicular to the ground.
- During operation the Vision-RTK 2 and antennas must remain in the same relative position to each other, that is, they must be rigidly mounted on the same frame
- The sensor must be mounted to minimize any vibrations that are not due to the vehicle motion, that is, excitation to the IMU that are not due to motion should be minimized.
- The antennas should be mounted high and unobstructed from metal or other signal impeding materials, as well as away from objects that can cause interference (e.g., LiDARs, unfoiled or unshielded ethernet cables, USB 3 devices, etc.)
- The antenna placement must have a full view of the sky to ensure full satellite visibility. Active helix antennas are preferred with full signal frequency tracking (suitable for GNSS L1, L2 and L5).
- The antennas should be mounted at least 20cm apart, but ideally >50cm apart.
- The sensor-to-antenna distances are accurately introduced with respect to the sensor frame (See Figure 10). Note that the antenna reference point corresponds to its antenna phase center or geometric equivalent.

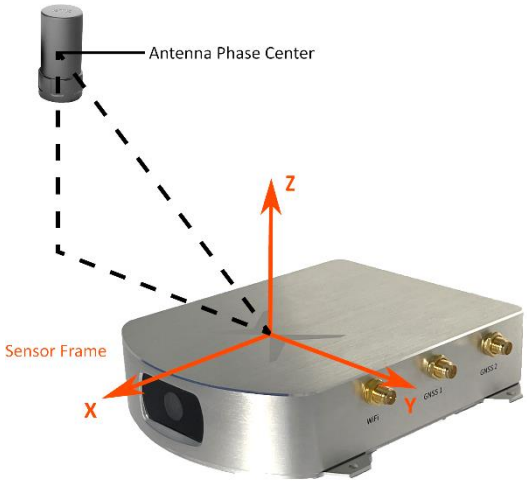


Figure 10 Sensor-to-antenna distance example

Maintenance:

- Ensure the camera lens is clean and unobstructed
- Ensure all cables are securely fastened
- Ensure the sensor and antennas are securely fastened

Auxiliary sensors:

The Vision-RTK 2 is able to incorporate additional auxiliary sensors into the fusion output. Currently, wheelspeed input is available, for details on how to set this up see 4.6. Wheel speed configuration. To enquire about other auxiliary sensors (external IMU, camera, radar, etc. please contact us directly).

Other considerations

- It is recommended that the Vision-RTK 2 be connected to a reliable and steady power source between 10-24 V (min 4.5V and max 40V)
- High quality coaxial cables are required with minimal signal attenuation, delay and a male SMA connector.
- A reliable internet connection is necessary if receiving RTK corrections over NTRIP
- Using antennas as well as a correction stream that have access to as many satellite constellations and signals as possible is preferred (e.g., antennas with only L1 capability, or corrections for only GPS signals will drastically reduce the sensors performance)
- If the chosen platform is expected to frequently travel at speeds above 3 m/s, it is recommended that the Vision-RTK 2 be mounted at more than 40cm above the ground to avoid the unsuccessful tracking of rapidly moving ground features.
- Wheelspeed input can greatly improve the performance of the sensor in outages, however, if you are operating with a platform and in an environment that exhibit excessive slippage, it may be detrimental. We recommend you assess whether the incorporation of this extra information into the fusion engine is beneficial to you or not.

4. Sensor configuration

4.1. Web interface overview

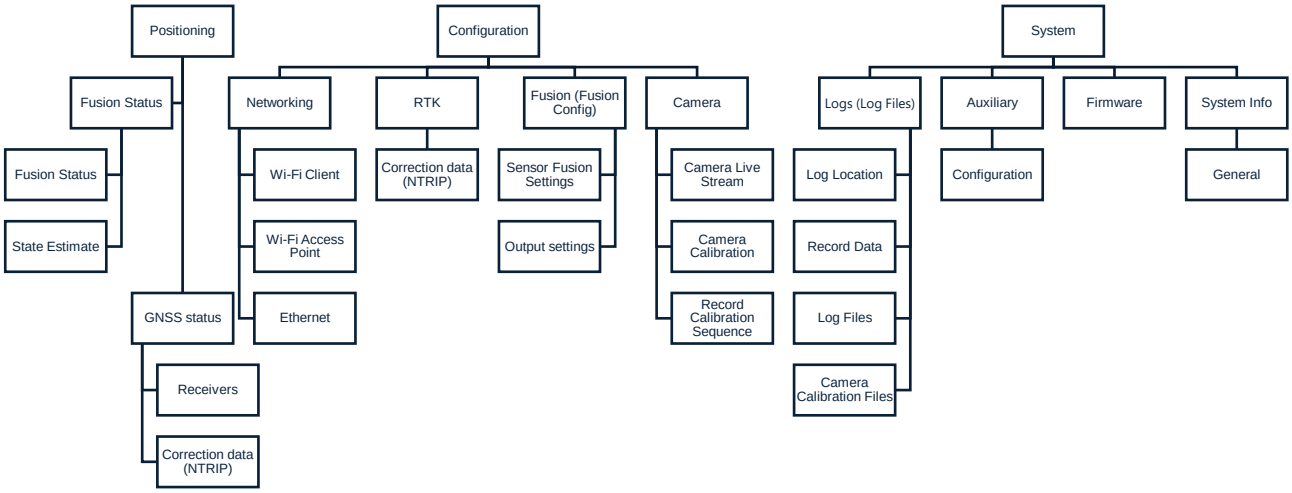


Figure 11 Embedded web interface structure

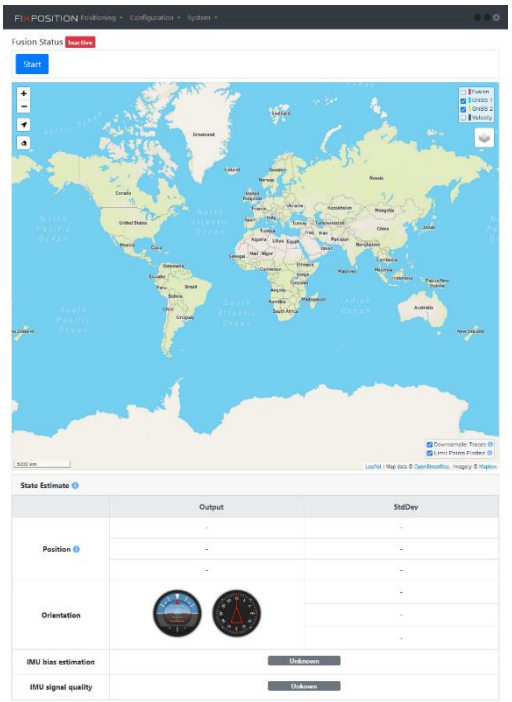


Figure 12 Main page of the embedded web interface

4.2. Networking configuration

4.2.1. Network specifications

The embedded web interface can be accessed via Wi-Fi and via Ethernet. In the Wi-Fi case, the SSID is the UID of the Vision-RTK 2 and its default password is “1234567890”. Additional networking configurations can be found in the section *Configuration → Networking*. The table below summarizes all the Networking configuration:

Network	Sensor IP	Netmask	Client IP	Configuration
Wi-Fi client	Dynamic	Dynamic	N/A	Configurable in the embedded 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	Configurable in the embedded web interface
Ethernet static	Variable	Variable	N/A	Configurable in the embedded web interface

Table 5 Networking specifications

Note that the Vision-RTK 2 will prioritize an ethernet connection for internet access over a Wi-Fi client.

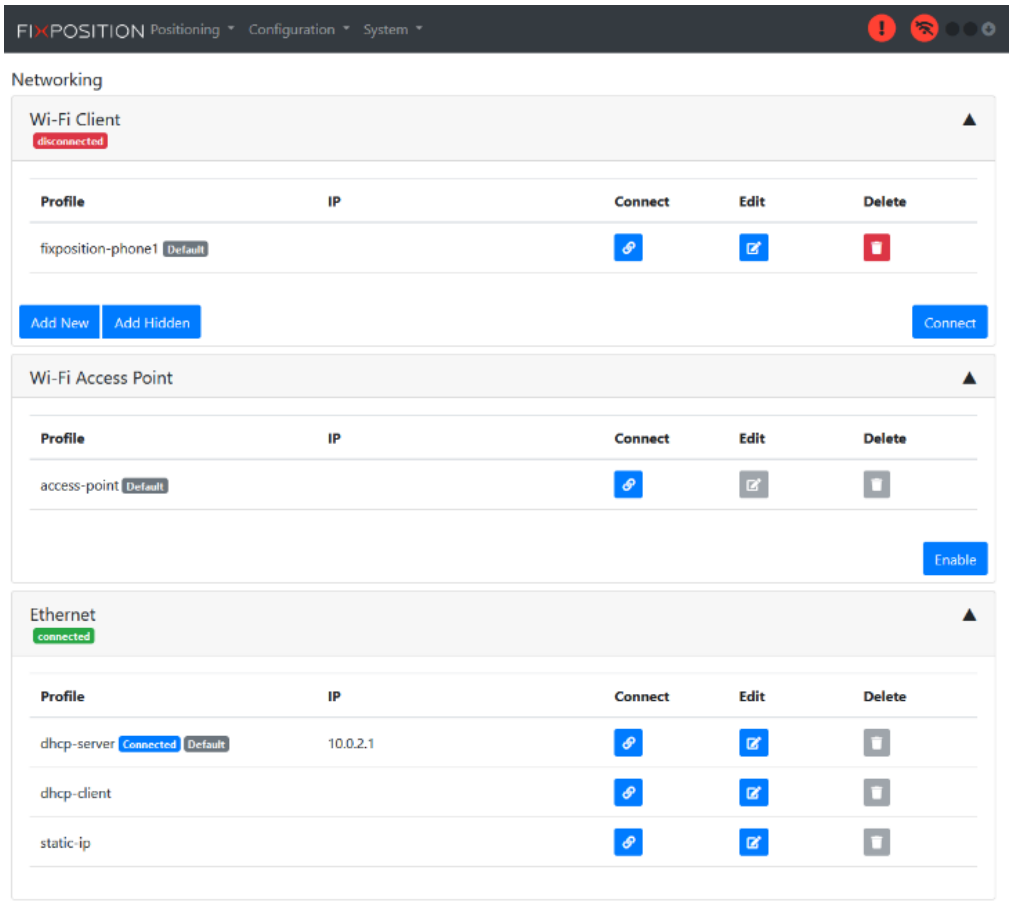


Figure 13 Networking page of the embedded web interface

4.2.2. Data ports

The following TCP/IP ports are available for connection:

Port	Protocol	Clients	Function
21000	Raw TCP/IP socket	Multiple	Output messages (see section 4.7)
20010	Raw TCP/IP socket	Multiple	GNSS1 receiver output
20020	Raw TCP/IP socket	Multiple	GNSS2 receiver output
23010	Raw TCP/IP socket	Multiple	NTRIP data stream received from the NTRIP caster
80	HTTP	Multiple	Web interface
8080	HTTP	Multiple	Firmware update interface
8888	HTTP/WebSocket	Multiple	Camera streaming (special uses only, see below)
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 6 List of available TCP/IP data ports

Port 8888 Camera streaming

It is possible to stream the camera image from TCP/IP port 8888. It works as follows:

Stream (MJPEG format):
- Full resolution: http://x.x.x.x:8888/stream?topic=/camera/image (ca. 9 MiB/s)
- Low resolution: http://x.x.x.x:8888/stream?topic=/camera/lowres/image (ca. 2 MiB/s)

Still image (JPEG format):
- Full resolution: http://x.x.x.x:8888/snapshot?topic=/camera/image (ca. 350 KiB)
- Low resolution: http://x.x.x.x:8888/snapshot?topic=/camera/lowres/image (ca. 90 KiB)

Some notes and warnings:

- The stream format is compatible with common software, such as Mozilla Firefox, Google Chrome, ffmpeg, VLC, etc. The transport is HTTP and the format is MJEPG.
- The frame rate is limited (approx. 4 fps)
- Each running stream costs some CPU and may affect the sensor performance (one stream should be fine, but multiple streams will cost too much CPU, which will affect fusion performance)
- The streams (or stills) should **only be used for debugging and development**, for example for checking the camera alignment. Operational and continuous use is not recommended.
- The web streaming service can be a bit buggy. It may stop working when too many streams and/or connects/disconnects happen.

4.2.3. Time synchronization

There are two options:

1. Use NTP protocol over 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 precise system clock of the Vision-RTK 2.

2. Use the PPS signal. The VRTK2 outputs the PPS signal from the GNSS1 receiver.

   The pin for the **time pulse** on the VRTK2 is **pin 5 on the I/O connector**
   The pin for the **time mark** on the VRTK2 is **pin 6 on the I/O connector**

   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 itself to each second in time. The raised edge

of each pulse will be aligned to the top of GPS time seconds. The duty cycle is 10 %, meaning that the pulse width is 100 ms. The timestamp message is available in the raw GNSS1 output on port 20010.

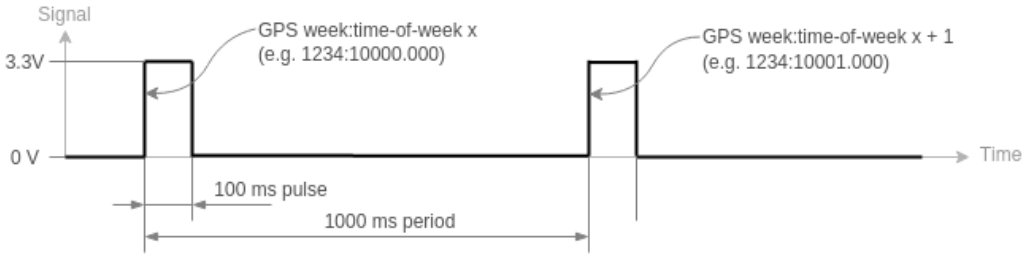


Figure 13 Networking page of the embedded web interface

Time mark

The time mark provides an accurate measurement of the time at which a pulse is detected on pin 6 and outputs a timestamp message on the port 20010. The maximum time mark frequency is 5 Hz, that is at most one mark per 200 ms interval.

For method 2. some devices require a NMEA-GP-RMC message along the timepulse. If this is the case, please contact Fixposition directly.

4.3. RTK/GNSS configuration

4.3.1 RTK/GNSS specifications

The Vision-RTK 2 RTK/GNSS configuration must fullfil the following specifications:

- Coverage: the operational area of the sensor must match the geographical area of the correction data service.
- Transport: it must be Networked Transport of RTCM via Internet Protocol (NTRIP) version 1 only.
- Data format: a minimum of specific RTCM3 messages for proper operation (see section 4.3.2).
- GNSS constellation: the data must include all four GNSS constellations to ensure the maximum GNSS performance.

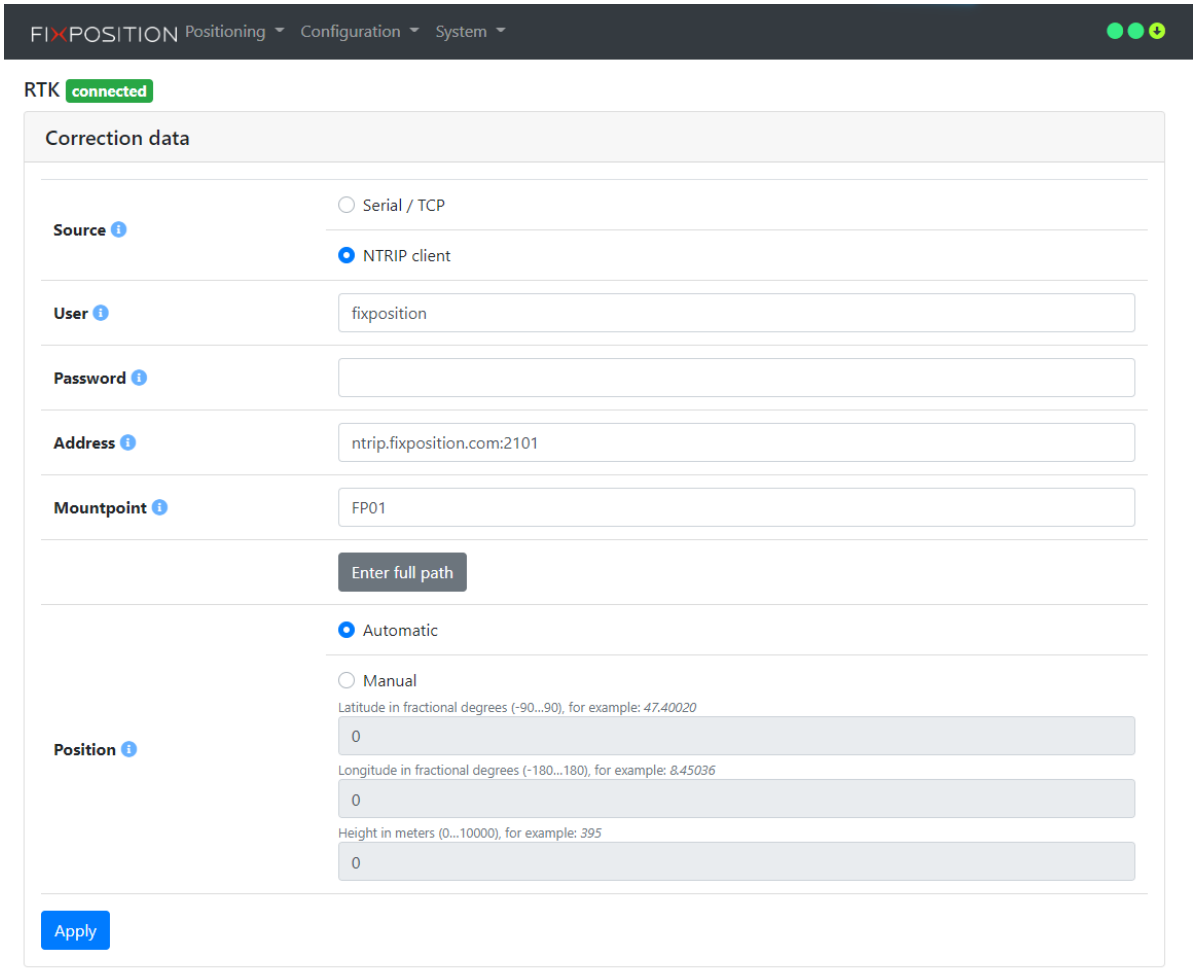


Figure 14 RTK configuration page in the embedded web interface

The sensor requires RTK correction data to provide an accurate satellite positioning. The correction data can be configured via the following methods in the section *Configuration → RTK*:

- Built-in NTRIP client: this method can connect to a selected NTRIP caster either from an automatic positioning from GNSS signals or manually from a configured reference point. It requires that the sensor has access to the caster server by enabling an internet connection. The user, password, address and mountpoint fields must be entered.

- Serial port: under “Source”, tick the “Serial / TCP” field. This method requires that you stream RTCM3 messages via a serial port on the Vision-RTK 2. In the section *System → System info*, the serial port field *rx*

indicates the number of bytes received on the serial port and the filed rtcm3 indicates the number of RTCM3 messages received on the serial port. The baudrate is currently not configurable and must be at 115200.

- TCP/IP port 21000: under “Source”, tick the “Serial / TCP” field. This method requires that you stream RTCM3 messages over port 21000 on the Vision-RTK 2. In the section *System → System info*, the serial port field *rx* indicates the number of bytes received on the serial port and the filed rtcm3 indicates the number of RTCM3 messages received.

The requirements on the correction data (the contents) for input of RTK corrections via serial port or TCP/IP are the same as for obtaining them via NTRIP.

- The **coverage** (geographic operation area) of the service must match the operation area of the sensor
- The **data format** must be RTCM3 (implied by NTRIP)
- The **data content** must be OSR style messages (“MSM”, multi-signal messages)
- The data must include data for all **GNSS** (i.e., GPS, GLONASS, Galileo **and** BeiDou)Missing systems will degrade performance significantly.

4.3.2. RTCM3 input messages

The Vision-RTK 2 requires a minimum of RTCM3 message for proper RTK/GNSS operation. The usage of less than four constellations is disadvised.

Type	Message
Reference station position - **Update rate: every 10 s 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 5 s or less**	- RTCM type 1230

Table 7 List of required RTCM3 input messages

4.4. Local NTRIP caster

Alternatively, a local NTRIP caster can be setup to stream correction data to the Vision-RTK 2. In the example below, a host system receives a serial input and streams it to the Vision-RTK 2.

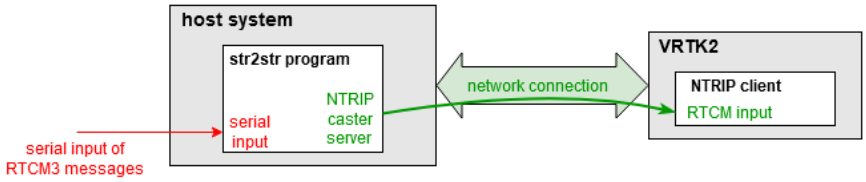


Figure 15 Workflow diagram of the str2str program combined with the Vision-RTK 2

The RTKLIB str2str program can divide an input stream and divide into multiple streams with different formats. For additional information, refer to https://rtkexplorer.com/pdfs/manual_demo5.pdf.

4.5. Sensor fusion configuration

Access the web interface and navigate to the section *Configuration → Fusion*. In *Sensor Fusion Settings* you are able to configure the following:

1. Autostart: it will start sensor fusion immediately after the system boots.
2. Mode (see Section 1.3) at which the sensor will adapt to the specific platform.
3. Sensor to antenna translation: specifies the position of the antenna 1 and antenna 2 with respect to the Vision-RTK 2 sensor frame (see Section 2.3) in meters.

After starting the sensor fusion output, it will be enabled once both GNSS/RTK receivers acquire an RTK fix solution and the sensor has captured any kind of translation.

4.6. Wheel speed configuration

4.6.1. Embedded web interface panel

Access the embedded web interface and navigate to the section *System → Auxiliary*.

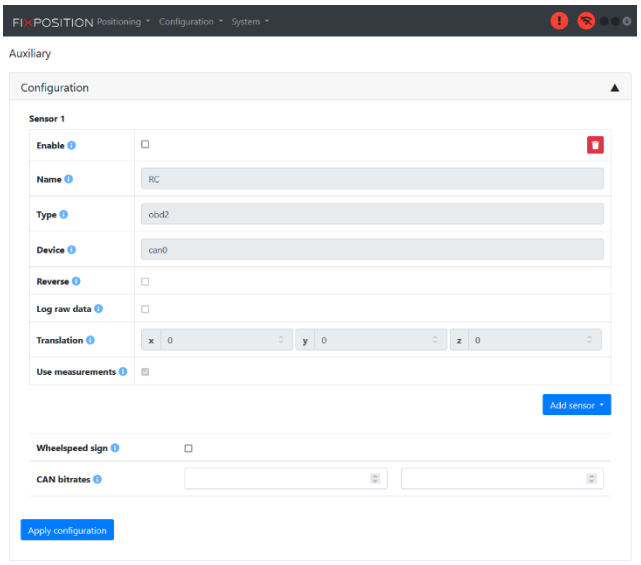


Figure 16 Auxiliary sensor configuration of the embedded web interface

4.6.2. Configuring an auxiliary sensor

The auxiliary sensor configurator supports up to four external sensors. Each individual sensor requires a correct definition for it be used as an input to the sensor fusion engine. Note that for each field, the string must be typed exactly, because all fields are quote, space and case sensitive. The order of the auxiliary sensor definition does not matter. The configuration is the following:

- Low-level sensor configuration
  - **Enable** – Check to enable this sensor. If not checked, none of the other parameters will be used. It can be left unchecked, for example, to keep the configuration saved.
  - **Name** – This is the reference name for the sensor and represents the type of measurement that it will generate. The value must be unique among all the configured and enabled sensors, no two active sensors may use the same value. Depending on the setup it should be one of the following (See Figure 17).
    - RC – Rear Centre wheel. Most common case.
    - RL – Rear Left wheel.
    - RR – Rear Right wheel.
    - FL – Front Left wheel.
    - FR – Front Right wheel.
    - NA – Special value (see 4.6.5 *Raw CAN bus logging* below).
  - **Type** – This is the type of the sensor, as its unique identifier.
  - **Device** – This is the Linux device name.
  - **Reverse** – When enabled the auxiliary sensor inverts the sign of the measurement. For example, if the sensor measures “12.3”, the measurement message contains “12.3” as the speed value if this is unchecked. If checked, the measurement message contains the value “-12.3”.
  - **Log raw data** – This enables raw data logging. Currently only the “*canlog*” fake sensor supports this. See also *Raw CAN bus logging* below. Note that if you enable this on a sensor that cannot do raw logging, the system will automatically uncheck this.
- Fusion configuration
  - **Translation** – This is the translation vector from the origin of the Vision-RTK 2 sensor frame (see section 2.3) to the origin of the wheel speed sensor in meters.

- **Use measurement** – If checked, the measurement is used by the sensor fusion engine.

Separately from the sensor configuration there is the configuration for the CAN interface:

- **CAN bitrate** – This is the CAN bus bitrate in hertz. This value is used for all sensors that use the "*can0" device*. See below for an indication of the right value for different sensors.

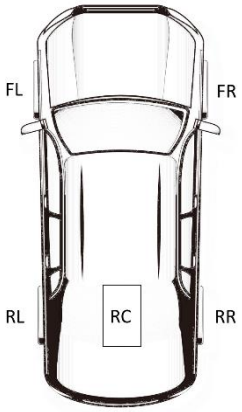


Figure 17 Wheel reference name

4.6.3. Passenger car OBD-2

The low-level sensor parameters are:

- *Enable*: *checked*
- *Name*: *RC*
- *Type*: *obd2*
- *Device*: *can0*
- *Reverse*: *unchecked*
- *Log raw data* (logging raw data not supported): *unchecked*
- *Use measurement: checked*
- *Wheelspeed sign: unchecked*
- *CAN bitrate*: *500000, 500000*

The OBD-2 CAN bus pins must be connected to the CANH, CANL and GND. Note that not all cars will support this interface, the CAN bus interface can differ from each manufacturer.

4.6.4. UART input

For streaming wheel speed information via the UART serial port, the message format must be the following:

Offset	Type	Value	Description	
0	uint8_t	0xaa	Header	Sync byte 1
1	uint8_t	0x44		Sync byte 2
2	uint8_t	0x13		Sync byte 3
3	uint8_t	20		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	Payload	Wheel speed value 1, for RC or FR wheel
16	int32_t	dmi2		Wheel speed value 2, for FL wheel
20	int32_t	dmi3		Wheel speed value 3, for RR wheel
24	int32_t	dmi4		Wheel speed value 4, for RL wheel
28	int32_t	mask		Wheel speed value mask
32	uint32_t	cksum	Checksum	CRC2 checksum (seed 0x00000000, polynomial 0xEDB88320)

Table 8 UART input message format

This binary message must be input on UART1 (pin 2 and 3 form the Host connector) at a regular interval and with a maximum input rate of 50 Hz. The latency from the wheel measurement to the input in the serial port must be reduced as much as possible.

The dmi1..4 values are speed values in an arbitrary unit. Its resolution must be enough to produce meaningful measurements out of small movements. Note that coarse resolutions like kilometer per hour will not work well, meter per second is recommended as the standard measurement unit. Subsequently, the wheel speed mask value determines which of the dmi1..4 values contains valid data with [0x00000001, 0x00000002, 0x00000004, 0x00000008] respectively.

The low-level sensor parameters are:

- *Enable*: *checked*
- *Name*: *RC*
- *Type*: *serial*
- *Device*: *uart1*
- *Reverse*: *checked or unchecked, depending on the input data*
- *Log raw data*: un*checked*
- *Use measurement*: checked
- *Wheel speed sign*: checked or unchecked
- *CAN bitrate*: not used

The *mask* determines which of the *dmi1..4* values contain valid data:

- 0x00000001 = *dmi1* value is valid
- 0x00000002 = *dmi2* value is valid
- 0x00000004 = *dmi3* value is valid
- 0x00000008 = *dmi4* value is valid

The *dmi1…4* values are speed values in an arbitrary unit. The resolution should be chosen such that small movements produce a useful signal. For example, [mm/s], [0.01km/h] or [2^{-10}m/s] should work. Coarse resolution, such as [km/h], may not work well, in particular at slow speeds.

- *dmi1* is for RC wheel or FR wheel
- *dmi2* is for FL wheel
- *dmi3* is for RR wheel
- *dmi4* is for RL wheel

An example message with hexdump of the binary data:

- *dmi1* = 111 = 0x0000006f = 6f 00 00 00 (at offset 12)
- *dmi2* = -22222 = 0xffffd752 = 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 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^
```

4.6.5. Raw CAN bus logging

The Vision-RTK 2 can also record an available CAN bus. The low-level sensor parameters are:

- *Enable*: *checked*
- *Name*: *NA*
- *Type*: *canlog*
- *Device*: *can0*
- *Reverse*: *unchecked*
- *Log raw data*: *checked*
- *CAN bitrate*: any value within the CAN bus protocol

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

4.6.6. Verify the messages

In the section *System → System info*, verify that the field *auxiliary sensors* receives the messages correctly. For example, a message of:

```
RC(obd2): 1234 meas (345 bad)
```

Means that the sensor name "RC", type "odb2" has produced 1234 measurement of which 345 were bad (i.e., ignition in car off and CAN bus therefore remains silent). When using the serial input, additional information will be displayed in the serial port field.

4.7. Output messages configuration

The Vision-RTK 2 can output several types of message formats. These are listed below:

Message type	Format	Rate	Description
ODOMETRY	ASCII	Configurable	Main message in ECEF WGS-84 coordinates. Consult the additional documentation included in the FP message converter.
LLH	ASCII	Configurable	Additional message in geodetic coordinates. Consult the additional documentation in FP message converter.
RAWIMU	ASCII	200 Hz	Additional message with time, accelerations and angular velocities. Raw values without bias correction, only coordinate transformation is applied.
CORRIMU	ASCII	200 Hz	Additional message with time, accelerations and angular velocities. Corrected values with bias, coordinate transformation is applied.
TF	ASCII	Configurable	Additional message with static coordinate transformations.
NMEA	ASCII	Configurable	Standard NMEA GGA (http://aprs.gids.nl/nmea/#gga)
INSPAVXB	Binary	Configurable	Novatel format message (https://docs.novatel.com/OEM7/Content/SPAN_Logs/INSPVAX.htm)
BESTGNSSPOS	Binary	5 Hz	Novatel format message with the best available GNSS position (https://docs.novatel.com/OEM7/Content/SPAN_Logs/BESTGNSSPOS.htm)
HEADING	Binary	5 Hz	Novatel format message with the heading (https://docs.novatel.com/OEM7/Content/Logs/HEADING2.htm)
RAWIMU	Binary	200 Hz	Novatel format message with the raw IMU measurements (https://docs.novatel.com/OEM7/Content/SPAN_Logs/RAWIMU.htm)

Table 9 List of available output messages

The output messages can be configured in *Configuration → Fusion*, in the *Output Settings* section.

The following options are available:

- Output port: configurable TCP/IP output port (see 4.2.2. Data ports). The ports 2007, 2008, 20010, 20011, 22000 and 23000 are restricted.
- Output frequency: frequency in hertz at which the messages will be delivered. It is limited to a maximum of 200 Hz.
- Translation sensor to output: translation of the positioning output with respect to the sensor frame in meters.
- Rotation sensor to output: rotation of the positioning output with respect to the sensor frame in Euler angles.

For additional information on the Vision-RTK 2 message setup, consult the documentation from the FP Message converter (see section 5) and the GNSS Transformation Library.

4.8. RTK/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 10 List of RTK/GNSS signal qualities.

Note that the acquired fix types are neither a measure of the signal quality nor a measure of the positioning accuracy.

4.9. Recording data

To obtain the sensor's data, go to *Configuration → Fusion* and navigate to the *Record Data* section. To start recording, press the Record button and to stop recording, press the same button.

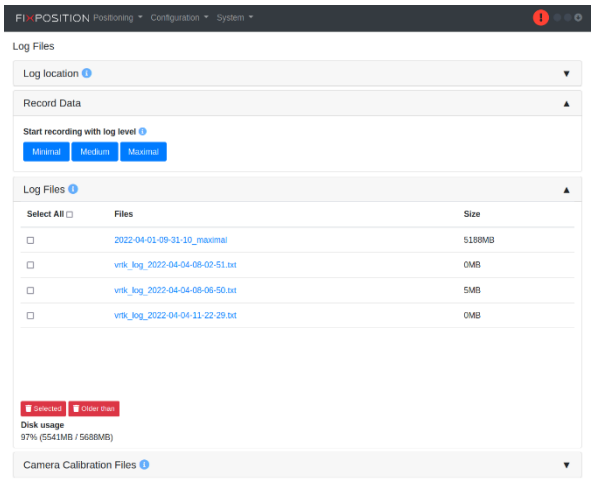


Figure 18 Recording data section in the embedded web interface

ROSbag type	Topics	Description
Minimal	/fusion_optim/odometry /fusion_optim/status	Sensor fusion odometry data and status
	/customer_output/poi/geodetic /customer_output/poi/odometry /customer_output/serial_input	Navigation data output
	/wheels/raw /wheels/data	External sensors data
	/(.*)/epoch /imu/data	Internal sensors data
	/clock	Miscellaneous ROS data

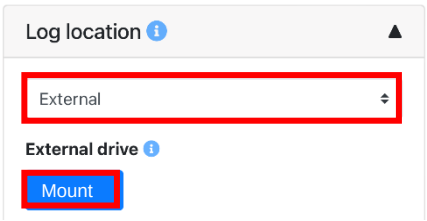
	/tf_static	
Medium	All the minimal topics	
	/ntrip/raw '/(.*)/raw'	RAW GNSS data
	/ntrip/latency '/(.*)/latency' '/(.*)/status'	NTRIP and GNSS latency
Maximal	All the minimal and medium topics	
	/camera/lowres/image /camera/lowres/camera_info	Camera images

The following table describes all the ROS topics included in the minimal, medium and maximal ROSbags:

Table 11 ROSbags topic structure

In addition, the FP message converter provides an alternative recording data (see 5.Output messages).

For additional storage, you can also configure an external drive. Simply connect an external drive (minimum write speed 100 MB/s) to the USB-C port and in the web-interface, in the menu select *system -> Logs -> Log location,* select *External* and click *Mount.*



4.10. IMU calibration

The Vision-RTK requires a start-up procedure before being fully operational. To start the calibration procedure, the following requisites must be fulfilled:

- Both receivers require an RTK fix resolution.
- The sensor fusion engine must be initialized and have a stable solution.

Once those requisistes are fulfilled, drive aproximately 2 minutes in RTK fix with some dynamic movement to converge the IMU biases. For example, driving eight figures and driving backwards and forwards within a 10 meters area as shown below.

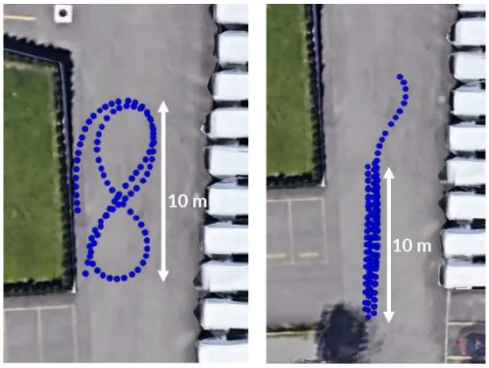


Figure 19 Example trajectory for the IMU bias trajectory.

With this procedure, the gyroscope and accelerometer bias will converge to steady values. The sensor will raise a flag once it is fully calibrated (see section 5.1).

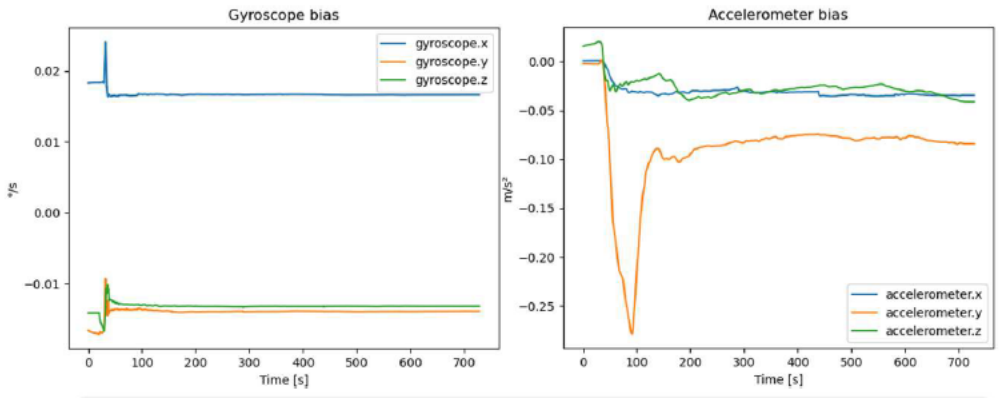


Figure 20 Gyroscope and accelerometer biases over time

4.11. ROS driver installation

Please visit https://github.com/fixposition/fixposition_driver for more information.

5. Output messages

5.1. Fixposition messages

5.1.1. Odometry message

The odometry messages contain full sensor fusion odometry output and additional status information. The following example showcases the odometry message (wrapped in multiple lines for readability):

```
$FP,ODOMETRY,1,2197,126191.765,4278415.1169,636245.1942,4672227.8942,-0.921035,-0.001266,-0.365401,
-0.134863,0.6169,-0.0140,-0.0068,0.01857,-0.01427,-0.00746,-0.1185,-0.0795,9.7791,4,1,1,1,0.55214,
0.33578,0.50777,0.08625,-0.13062,-0.45209,0.00227,0.00020,0.00270,0.00027,0.00031,0.00232,
0.03314,0.03828,0.03199,-0.00290,0.00246,-0.00119,fp_release_vr2_2.36.1_67*47
```

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	ODOMETRY	Message type, ODOMETRY for this message
2	msg_version	Numeric	-	1	Message version
3	gps_week	Numeric	-	2197	GPS week number, range 0–9999
4	gps_tow	Float (.6)	s	126191.765	GPS time of week, range 0.000–604799.999
5	pos_x	Float (.4)	m	4278415.1169	Position in ECEF WGS-84, X component
6	pos_y	Float (.4)	m	636245.1942	Position in ECEF WGS-84, Y component
7	pos_z	Float (.4)	m	4672227.8942	Position in ECEF WGS-84, Z component
8	orientation_w	Float (.6)	-	-0.921035	Quaternion with respect to ECEF, W component
9	orientation_x	Float (.6)	-	-0.001266	Quaternion with respect to ECEF, X component
10	orientation_y	Float (.6)	-	-0.365401	Quaternion with respect to ECEF, Y component
11	orientation_z	Float (.6)	-	-0.134863	Quaternion with respect to ECEF, Z component
12	vel_x	Float (.4)	m/s	0.6169	Velocity in sensor frame, X component
13	vel_y	Float (.4)	m/s	-0.0140	Velocity in sensor frame, Y component
14	vel_z	Float (.4)	m/s	-0.0068	Velocity in sensor frame, Z component
15	rot_x	Float (.5)	rad/s	0.01857	Bias corrected angular velocity in sensor frame, X component
16	rot_y	Float (.5)	rad/s	-0.01427	Bias corrected angular velocity in sensor frame, Y component
17	rot_z	Float (.5)	rad/s	-0.00746	Bias corrected angular velocity in sensor frame, Z component
18	acc_x	Float (.4)	m/s^2	-0.1185	Bias corrected acceleration in sensor frame, X component
19	acc_y	Float (.4)	m/s^2	-0.0795	Bias corrected acceleration in sensor frame, Y component
20	acc_z	Float (.4)	m/s^2	9.7791	Bias corrected acceleration in sensor frame, Z component
21	fusion_status	Numeric	-	4	Sensor fusion status, see table 12
22	imu_bias_status	Numeric	-	1	IMU bias status, see Table 13
23	gnss_fix_type	Numeric	-	1	GNSS fix type, see Table 9
24	wheelspeed_status	Numeric	-	1	Wheel speed status, see Table 14
25	pos_cov_xx	Float (5)	m^2	0.55214	Position covariance, element XX

26	pos_cov_yy	Float (5)	m^2	0.33578	Position covariance, element YY
27	pos_cov_zz	Float (5)	m^2	0.50777	Position covariance, element ZZ
28	pos_cov_xy	Float (5)	m^2	0.08625	Position covariance, element XY
29	pos_cov_yz	Float (5)	m^2	-0.13062	Position covariance, element YZ
30	pos_cov_xz	Float (5)	m^2	-0.45209	Position covariance, element XZ
31	orientation_cov_xx	Float (5)	rad^2	0.00227	Velocity covariance, element XX
32	orientation_cov_yy	Float (5)	rad^2	0.00020	Velocity covariance, element YY
33	orientation_cov_zz	Float (5)	rad^2	0.00270	Velocity covariance, element ZZ
34	orientation_cov_xy	Float (5)	rad^2	0.00027	Velocity covariance, element XY
35	orientation_cov_yz	Float (5)	rad^2	0.00031	Velocity covariance, element YZ
36	orientation_cov_xz	Float (5)	rad^2	0.00232	Velocity covariance, element XZ
37	vel_cov_xx	Float (5)	m^2/s^2	0.03314	Velocity covariance, element XX
38	vel_cov_yy	Float (5)	m^2/s^2	0.03828	Velocity covariance, element YY
39	vel_cov_zz	Float (5)	m^2/s^2	0.03199	Velocity covariance, element ZZ
40	vel_cov_xy	Float (5)	m^2/s^2	-0.00290	Velocity covariance, element XY
41	vel_cov_yz	Float (5)	m^2/s^2	0.00246	Velocity covariance, element YZ
42	vel_cov_xz	Float (5)	m^2/s^2	-0.00119	Velocity covariance, element XZ
43	sw_version	String	-	fp_release_v.0	Software version

Table 12 Odometry message structure

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

Table 13 Sensor fusion status (see field 21)

Value	Description
0	Not converged
1	IMU bias converged

Table 14 IMU bias status (see field 22)

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

Table 15 Wheel speed status (see field 24)

5.1.2. LLH message

The LLH messages contain time, geographic coordinates and position covariances of the sensor output frame in East-North-Up coordinates. The coordinates are transformed from ECEF using WGS-84 parameters. The following example showcases the LLH message (wrapped in multiple lines for readability):

```
$FP,LLH,1,2197,126191.765,47.398826818,8.458494107,457.518,0.31537,1.0076,0.072696,-0.080012,0.0067274,
-0.011602*4E\r\n
```

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	LLH	Message type, LLH for this message
2	msg_version	Numeric	-	1	Message version
3	gps_week	Numeric	-	2197	GPS week number, range 0–9999
4	gps_tow	Float (.6)	s	126191.765	GPS time of week, range 0.000–604799.999
5	latitude	Float (.9)	deg	47.398826818	Latitude, range -90.000000000–90.000000000, > 0 for North, < 0 for South
6	longitude	Float (.9)	deg	8.458494107	Longitude, range -180.000000000–180.000000000, > 0 for East, < 0 for West
7	height	Float (.4)	m	457.518	Ellipsoidal height
8	pos_cov_ee	Float (5)	m^2	0.31537	Position covariance in ENU, element EE
9	pos_cov_nn	Float (5)	m^2	1.0076	Position covariance in ENU, element NN
10	pos_cov_uu	Float (5)	m^2	0.072696	Position covariance in ENU, element UU
11	pos_cov_en	Float (5)	m^2	-0.080012	Position covariance in ENU, element EN
12	pos_cov_nu	Float (5)	m^2	0.0067274	Position covariance in ENU, element NU
13	pos_cov_eu	Float (5)	m^2	-0.011602	Position covariance in ENU, element EU

Table 16 LLH message structure

5.1.3. RAWIMU message

The RAWIMU messages contain time, linear acceleration and angular velocity with respect to the sensor frame. The values are not bias-corrected and the output frequency is fixed to 200 Hz. The following example showcases the RAWIMU message:

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

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	RAWIMU	Message type, always RAWIMU for this message
2	msg_version	Numeric	-	1	Message version
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/s^2	-0.199914	Raw acceleration in sensor frame, X component
6	acc_y	Float (.6)	m/s^2	0.472851	Raw acceleration in sensor frame, Y component
7	acc_z	Float (.6)	m/s^2	9.917973	Raw acceleration in sensor frame, Z component
8	rot_x	Float (.6)	rad/s	0.023436	Raw angular velocity in sensor frame, X component
9	rot_y	Float (.6)	rad/s	0.007723	Raw angular velocity in sensor frame, Y component
10	rot_z	Float (.6)	rad/s	0.002131	Raw angular velocity in sensor frame, Z component

Table 17 RAWIMU message structure

5.1.4. CORRIMU message

The CORRIMU messages contain time, linear acceleration and angular velocity with respect to the sensor frame. The values are bias-corrected and the output frequency is fixed to 200 Hz. The following example showcases the CORRIMU message:

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

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	CORRIMU	Message type, always CORRIMU for this message
2	msg_version	Numeric	-	1	Message version
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/s^2	-0.195224	Bias corrected acceleration in sensor frame, X component
6	acc_y	Float (.6)	m/s^2	0.393969	Bias corrected acceleration in sensor frame, Y component
7	acc_z	Float (.6)	m/s^2	9.869998	Bias corrected acceleration in sensor frame, Z component
8	rot_x	Float (.6)	rad/s	0.013342	Bias corrected angular velocity in sensor frame, X component
9	rot_y	Float (.6)	rad/s	-0.004620	Bias corrected angular velocity in sensor frame, Y component
10	rot_z	Float (.6)	rad/s	-0.000728	Bias corrected angular velocity in sensor frame, Z component

Table 18 CORRIMU message structure

5.1.5. TF message

The TF messages contain static coordinate transformations. The following example showcases the TF message:

```
$FP,TF,1,VRTK,CAM,0.01795,0.00044,-0.01103,0.485049,-0.508955,0.511098,-0.494440*4A
```

#	Field	Format	Unit	Example	Description
1	msg_type	String	-	TF	Message type, always TF for this message
2	msg_version	Numeric	-	1	Message version
3	from_frame	String	-	CAM	From frame
4	to_frame	String	-	VRTK	To frame
5	translation_x	Float (.5)	m	0.01795	Translation, X component
6	translation_y	Float (.5)	m	0.00044	Translation, Y component
7	translation_z	Float (.5)	m	-0.01103	Translation, Z component
8	orientation_w	Float (.6)	-	0.485049	Rotation in quaternion, W component
9	orientation_x	Float (.6)	-	-0.508955	Rotation in quaternion, X component
10	orientation_y	Float (.6)	-	0.511098	Rotation in quaternion, Y component
11	orientation_z	Float (.6)	-	-0.494440	Rotation in quaternion, Z component

Table 19 TF message structure

5.2. NMEA messages

5.2.1. NMEA LLH

The NMEA LLH messages contain the following information:

#	Field	Format	Unit	Example	Description
1	talker_formatter	String	-	GPGGA	Sentence talker ID and formatter, always GPGGA for this message
2	utc_time	hhmmss.sss	-	105842.002	UTC time in NMEA format
3	latitude	DDmm.mm	-	4723.510710	Latitude in NMEA format
4	latitude_ns	Character	-	N	Latitude North (N) or South (S) indicator
5	longitude	DDmm.mm	-	00830.684893	Longitude in NMEA format
6	longitude_ew	Character	-	E	Longitude East (E) or West (W) indicator
7	quality	Digit	-	1	Fix quality indicator
8	num_sv	Numeric	-	14	Number of satellites in use
9	hdop	Numeric	-	1	Horizontal dilution of precision
10	height	Float (.2)	m	456.76	Orthometric height
11	height_unit	Character	-	M	Height unit, always M for meters
12	undulation	Float (.1)	m	0.0	Geoid undulation
13	undulation_unit	Character	-	M	Undulation unit, always M for meters
14	age	Numeric	s	n/a	Age of correction data, always *null*
15	station	Numeric	-	n/a	Differential base station ID, always *null*

Table 20 NMEA LLH message structure

5.2.2. NMEA HDT

The NMEA HDT messages contain the following information:

#	Field	Format	Unit	Example	Description
1	talker_formatter	String	-	GPHDT	Sentence talker ID and formatter, always GPHDT for this message
2	heading	Float (.5)	deg	8.06135	Heading
3	deg_true	Character	-	T	Heading degrees true, always T

Table 21 NMEA HDT message structure

Appendix

A. Software updates

To update the software version of the Vision-RTK 2 go to the section *System* → *Firmware* in the Embedded Web Interface and drag and drop a SWU file inside the grey area.

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