



ZED-F9R

Getting started with high-precision sensor fusion

Application note



Abstract

This application note leads the user through the steps necessary to evaluate and prototype a sensor fusion-based system using the ZED-F9R. The document cuts across many types of product documentation especially for beginners.

Document information

Title	ZED-F9R	
Subtitle	Getting started with high-precision sensor fusion	
Document type	Application note	
Document number	UBX-22035176	
Revision and date	R01	9-Jan-2023
Disclosure restriction	C1-Public	

This document applies to the following products:

Product name
ZED-F9R

u-blox or third parties may hold intellectual property rights in the products, names, logos, and designs included in this document. Copying, reproduction, or modification of this document or any part thereof is only permitted with the express written permission of u-blox. Disclosure to third parties is permitted for clearly public documents only.

The information contained herein is provided "as is" and u-blox assumes no liability for its use. No warranty, either express or implied, is given, including but not limited to, with respect to the accuracy, correctness, reliability, and fitness for a particular purpose of the information. This document may be revised by u-blox at any time without notice. For the most recent documents, visit www.u-blox.com.

Copyright © u-blox AG.

Contents

Document information	2
Contents	3
1 Introduction	5
1.1 Required items.....	5
1.1.1 Operational ZED-F9R module	5
1.1.2 GNSS antenna requirements	5
1.1.3 Host system.....	6
1.1.4 Odometer data	6
1.1.5 Correction data	6
2 Communicating with the receiver in u-center	7
2.1 Connecting the receiver with u-center	7
2.2 Updating the receiver firmware	7
2.3 Storing configuration settings.....	8
2.4 Modifying the receiver configuration in UBX-CFG-VALSET view	9
2.5 Modifying the receiver configuration in the Generation 9 Advanced Configuration view	9
3 Installation	11
3.1 Mounting the receiver	11
3.2 Mounting the antenna	13
3.3 Connecting cables.....	13
3.4 Setting up UART configuration.....	13
4 GNSS setup	14
4.1 Verifying GNSS signals	14
4.2 Changing enabled constellations	15
5 RTK setup	16
5.1 Setting up NTRIP client in u-center.....	16
5.2 Setting up MQTT client in u-center.....	17
5.3 Monitoring RTK status	18
6 Sensor fusion setup	19
6.1 Providing odometer data to the receiver.....	19
6.1.1 Wheel ticks.....	19
6.1.2 Speed.....	19
6.1.3 Data stream quality.....	19
6.1.4 UBX-ESF-MEAS sample code	20
6.2 Sensor fusion configuration	20
6.2.1 Dynamic platform model.....	21
6.2.2 IMU-mount alignment	21
6.2.3 Setting up odometer configuration	21
6.2.4 Navigation output rate	22
7 Calibration and testing	23
7.1 Initialization and calibration	23

7.2 Preserving calibration	24
7.3 Performing a basic test drive.....	24
7.4 Scenario testing	24
A Appendix.....	25
A.1 Recording logs	25
A.2 Replaying logs.....	25
A.3 Interference of GNSS signals	25
A.4 Communications and protocols.....	26
A.5 Basic alignment & position offset checks.....	27
A.6 Odometry data issues	27
A.7 RTK never reaching fix	27
A.8 Dropping out of dead reckoning periodically	28
A.9 Troubleshooting checklist.....	28
Related documentation	30
Revision history	30
Contact.....	30

1 Introduction

This document acts as a getting started guide for evaluating and using the u-blox ZED-F9R module and high-precision sensor fusion technology. The guide is written with a focus on robotic applications, which is the platform that requires the most comprehensive receiver setup. As such, the guide is also suitable for other platforms where the setup is more simple.

The complete setup process is divided into self-contained chapters focusing on specific parts of the setup. The setup process includes

- preparing the host system
- configuring the receiver
- installing the receiver in a vehicle, and
- running test drives

It is strongly recommended to go through the chapters in order to make the setup process easier.

The document may contain specific information based on the product documentation such as Integration manual [1] or Interface description [2] which are updated regularly. Please refer to the product documentation whenever there are inconsistencies between this document and the official product documentation. This document is intended as a beginner's guide and does not supersede product documentation.

1.1 Required items

The following items are required for completing the setup in this guide:

- An operational ZED-F9R module
- A suitable multiband GNSS antenna
- A host PC running u-center for F9 products
- Serial communication between the receiver and the host
- Means to provide odometer data to the receiver
- Means to provide correction data to the receiver, including access to a correction service or a base station
- Means to properly attach the receiver to the vehicle

1.1.1 Operational ZED-F9R module

At the center of the setup is the ZED-F9R module installed on a printed circuit board (PCB). We recommend the C102-F9R evaluation kit for first-time users, as it provides a tested, ready-to-use device for most purposes. A custom design is also allowed, provided it follows the module's hardware design requirements (see [1]). The design should also provide at least a UART connection to the receiver but having USB support is strongly recommended.

1.1.2 GNSS antenna requirements

During the evaluation phase, it is best to start with known working components and the design can be optimized for cost, space, weight, or other considerations at a later stage. As a high-level summary, the antenna must:

- Be capable of multi-band L1 and L2 reception
- Be an active antenna with at least 17 dB of gain
- Be designed for real-time kinematic (RTK) applications with suitable phase center variation and group delay characteristics
- Include ground plane, if the antenna requires one

1.1.3 Host system

Certain functionality in the setup needs to be implemented by a host system. This includes:

- Configuring the receiver
- Monitoring and debugging the receiver
- Providing odometer data to the receiver
- Providing correction data to the receiver

All of the above can be done in u-center, except for providing odometer data to the receiver. It is recommended to start evaluating and developing the system with a PC running u-center as the main host, and gradually move the functionality to an embedded host for the final product.

It is also strongly recommended to make monitoring and debugging the receiver possible in the final product. To do this, design your system so that it is possible to connect to u-center directly to the receiver. Alternatively, data logging may be implemented by the host to allow data analysis and debugging.

1.1.4 Odometer data

Odometer data refers to information about the distance travelled by a vehicle. This information is usually available from a vehicle either as wheel ticks or speed measurements. The data is typically generated by sensors that can measure the rotation of the vehicle's wheels, such as Hall effect sensors. These sensors are often used in low-cost brushless DC motors utilized in typical robotic applications.

Providing odometer data to the receiver is mandatory for high-precision sensor fusion. If odometer data is not available from the vehicle, an external sensor must be attached to the vehicle for an accurate assessment of achievable accuracy.

Odometer data can be supplied to the receiver either through dedicated hardware pins for wheel ticks and direction, if the application type permits it, or through a serial interface as UBX-ESF-MEAS messages.

1.1.5 Correction data

Correction data is another mandatory part for high-precision sensor fusion. Corrections allow the receiver to measure its position more precisely by providing more information about the available satellites. Correction data is available through dedicated services, such as u-blox PointPerfect, that utilize dedicated software for obtaining the data. This data then is provided to the receiver through a serial port.

2 Communicating with the receiver in u-center

All communication with the receiver is handled through the u-blox proprietary UBX message protocol (see [2]). Through different messages, the user can apply configurations and monitor the status of the receiver. u-center provides a human-machine interface for these operations. This section acts as an introduction into communicating with the receiver in u-center. u-center is available from the [u-blox website](#).

2.1 Connecting the receiver with u-center

Follow these steps to connect u-center to the receiver and verify that the connection works:

1. Power on the receiver.
2. Connect the receiver to the host with a serial or USB cable.
3. Open u-center on your PC.
4. In u-center, select the correct serial port from **Receiver > Connection > COMxx**.
5. If connecting through UART, set the correct baudrate from **Receiver > Baudrate**. The default baudrate of the ZED-F9R is 38400.
6. Open the Messages View from **View > Messages View**.
7. Select the UBX-MON-VER message from the left-side of the list.
8. Click **Poll**. The view should look like the following when the receiver is operating properly:

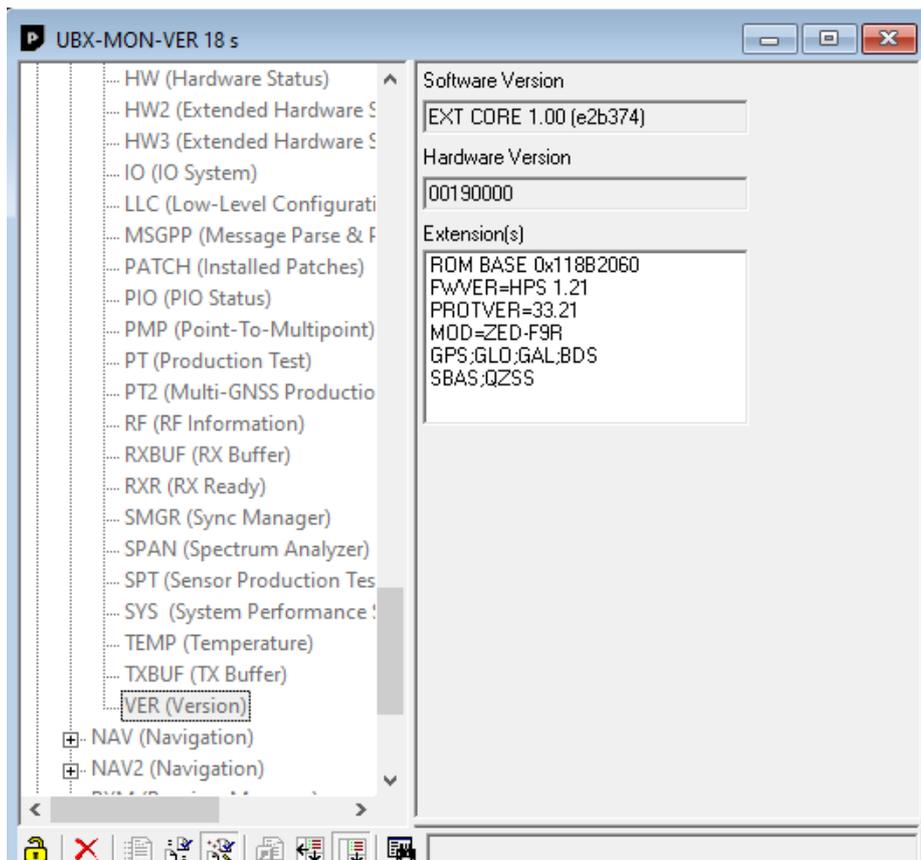


Figure 1: u-center UBX-MON-VER message

2.2 Updating the receiver firmware

Please update to the latest receiver firmware if the shipped ZED-F9R module has an older version. u-blox receiver firmware binaries are available on the [ZED-F9R product website](#).

The receiver firmware can be updated via the firmware update utility in u-center:

1. Open u-center and connect it to the receiver.
2. Open the firmware update utility from **Tools > Firmware Update**.
3. Select the target firmware image file in the **Firmware image field**.
4. If connected through UART, check the **Use this baudrate for update** box and select a higher baudrate to make the update faster. The receiver does not need to be configured separately for this step.
5. Press the green **Go** button in the bottom left corner to start the update process.
6. The receiver is ready to be used once the console output states that the update completed successfully.
7. The new firmware version can be confirmed with the UBX-MON-VER message.

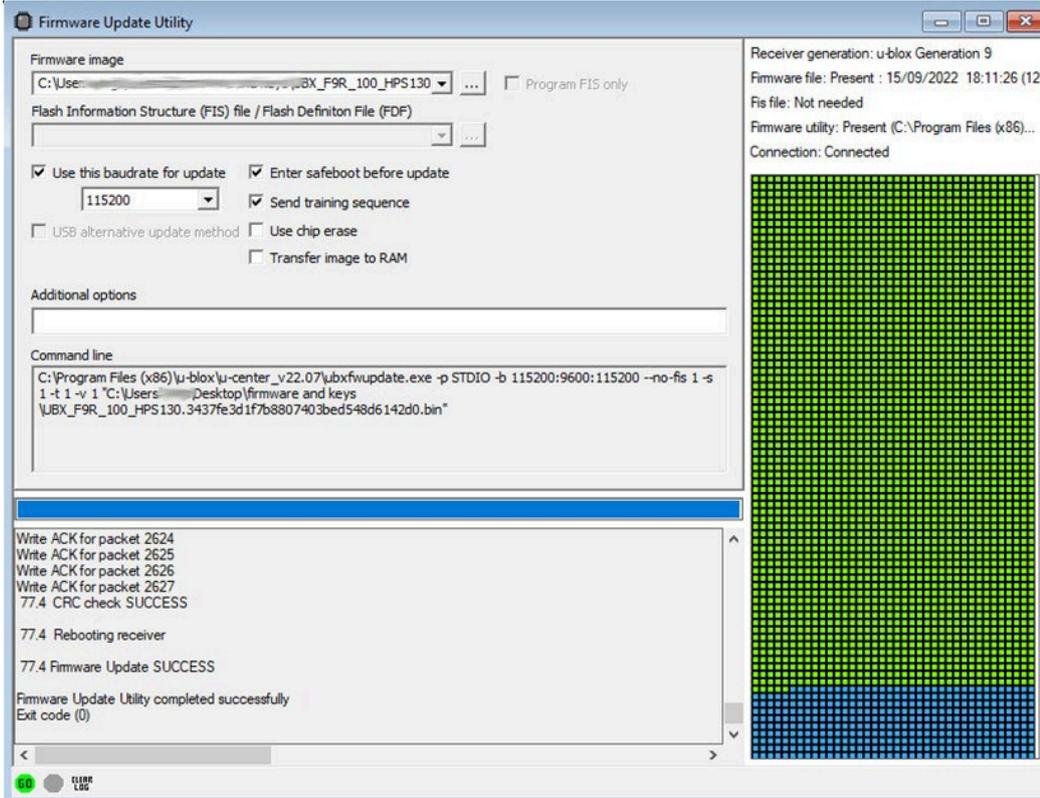


Figure 2: u-center firmware update

2.3 Storing configuration settings

ZED-F9R is fully configurable with UBX configuration interface messages. The configuration database in the receiver's RAM holds the current configuration that is used by the receiver at run-time. It is constructed on start-up of the receiver from several sources of configuration. The configuration interface and the available keys are described in the F9 HPS 1.30 Interface description [2].

Memory layer	Description
RAM	The configuration settings stored in RAM remain effective until power-down or reset
BBR (battery-backed RAM)	The configuration settings are stored as long as the backup battery supply remains
Flash memory	Permanent storage of configuration settings

Table 1 Memory layers for storing configuration settings

2.4 Modifying the receiver configuration in UBX-CFG-VALSET view

Figure 3 shows the UBX-CFG-VALSET message view.

1. Select the configuration items from the **Group** and **Key Name** dropdown menus and add these to the list of configuration items to set by clicking the **Add to list** button.
2. To set the value to apply, select an item from the list and set the value in the **Value field**. To poll the current value of the item, click the **Get current value** button.
3. Select the correct memory layer(s) in the **Set value in layers** field.
4. Click the **Send** button in the bottom left corner of the window.

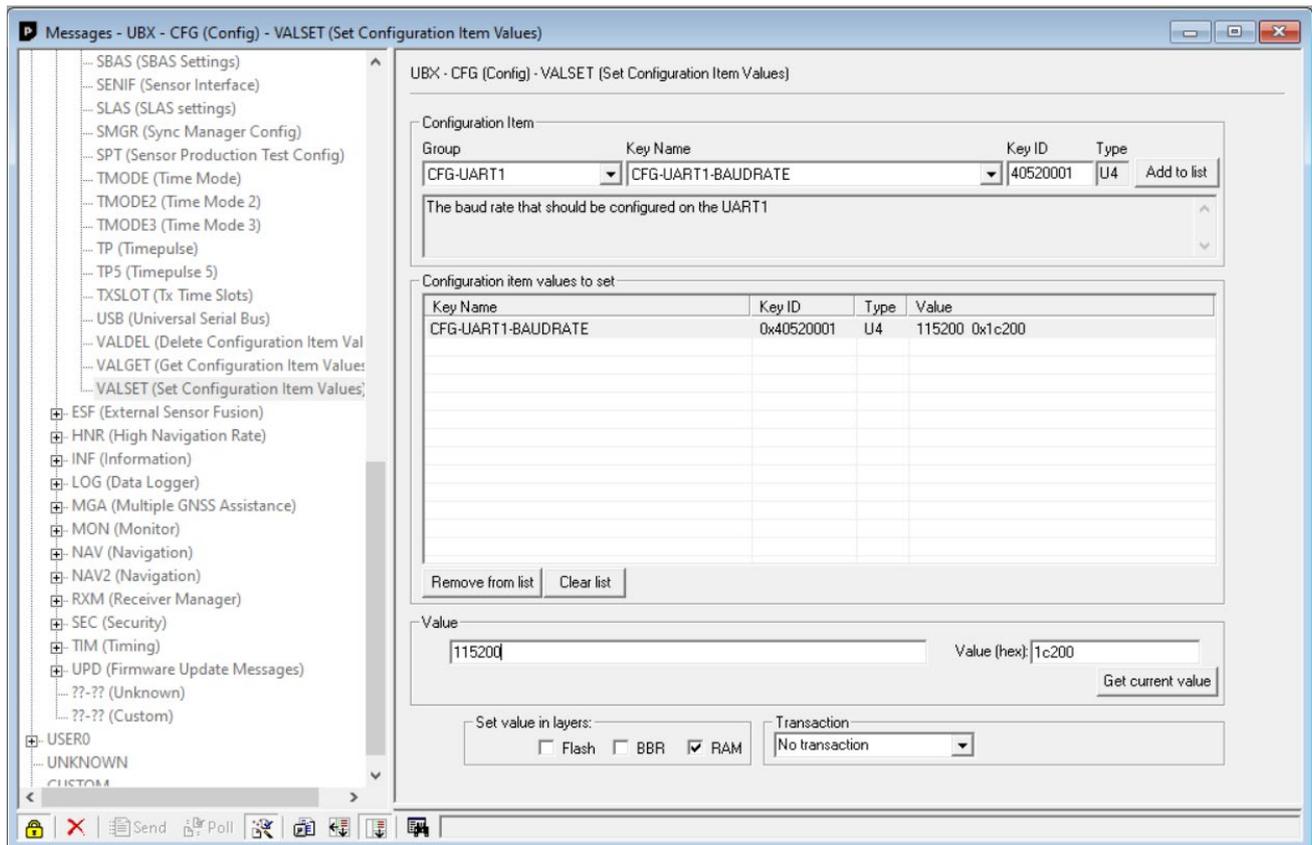


Figure 3: UBX-CFG-VALSET message

During the evaluation phase, it is most simple to write the configuration directly to flash. During development and series production, it is recommended to write the configuration to the RAM or BBR layer at startup. The host should not write to flash at every startup because of the maximum limit on the number flash write cycles.

2.5 Modifying the receiver configuration in the Generation 9 Advanced Configuration view

Figure 4 shows the **Generation 9 Advanced Configuration View**. This view presents all the available configuration items as a grouped list.

1. Add items to the list to be applied by selecting an item from the grouped list.
2. Write the correct value in the **Value** field.
3. Click the **Set in RAM/BBR/Flash** button.
4. To send the configuration to the receiver, click the **Send config changes** button.

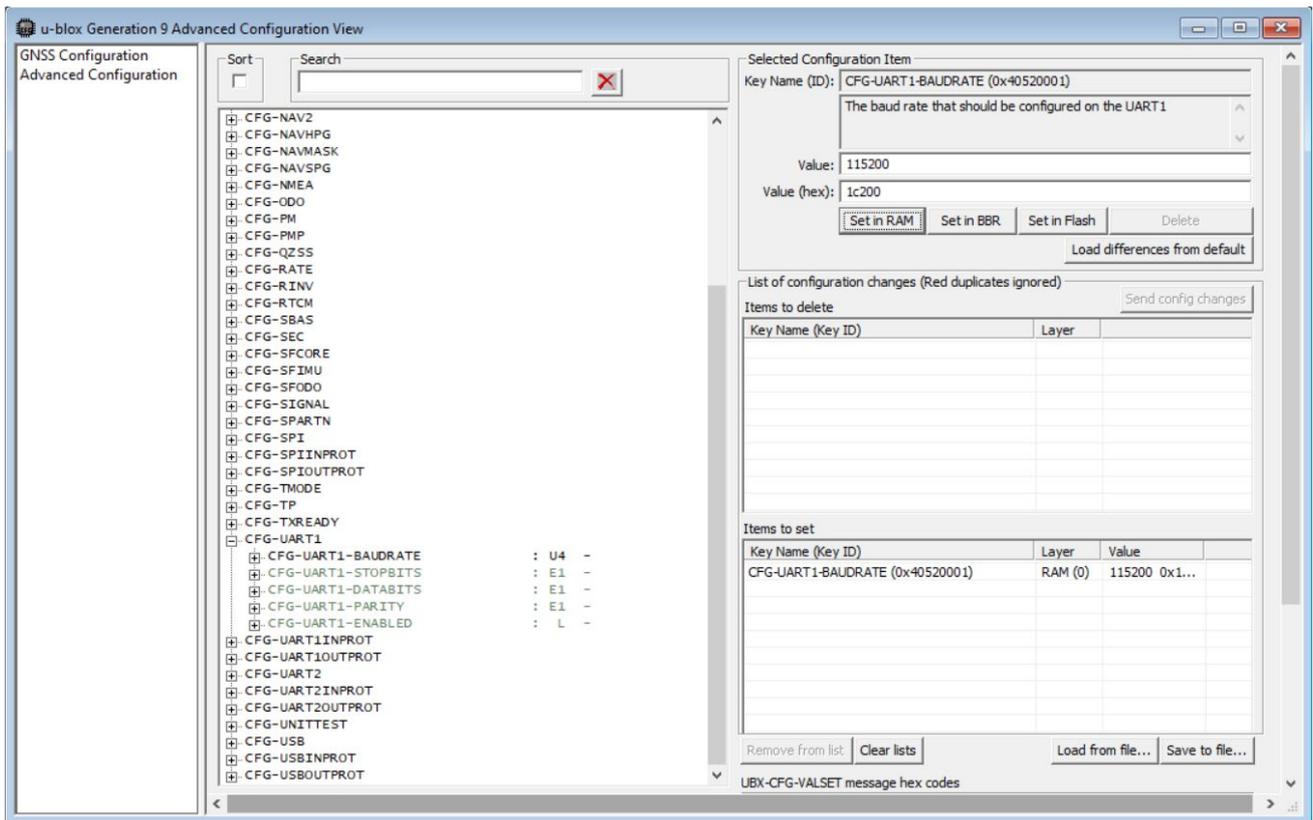


Figure 4: Generation 9 configuration view

During the evaluation phase, it is most simple to write the configuration directly to flash. During development and series production, it is recommended to write the configuration to the RAM or BBR layer at startup. The host should not write to flash at every startup because of the maximum limit on the number flash write cycles.

3 Installation

The first step of setting up the ZED-F9R is installing it into the vehicle. Proper installation is vital for the navigation performance of the receiver, as its internal sensors must be able to measure the dynamics of the vehicle accurately in a consistent way. This means the antenna and receiver must be mounted rigidly in a fixed position and orientation relative to the vehicle's body.

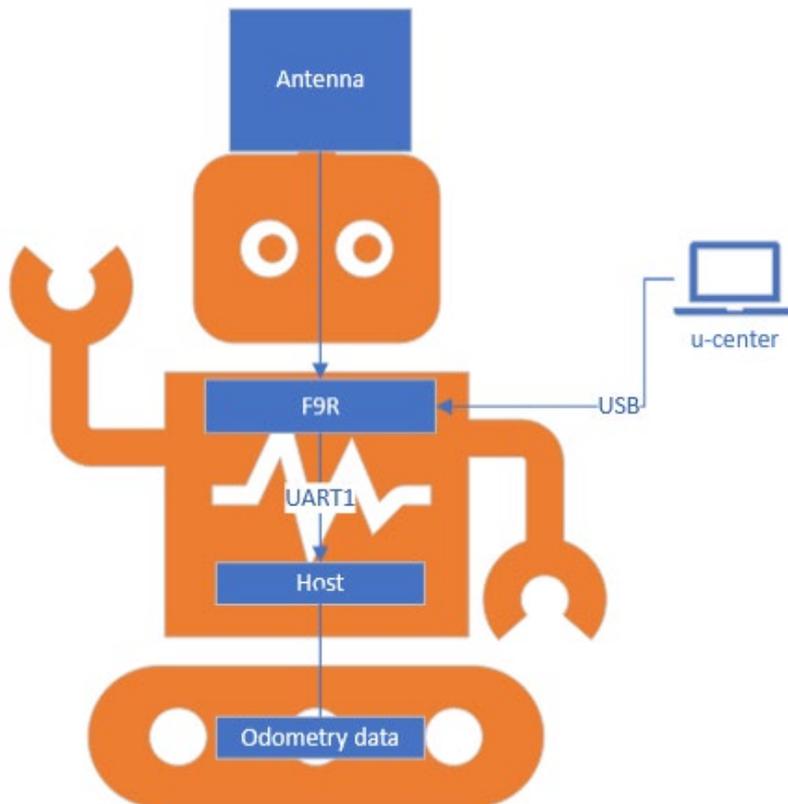


Figure 5: Conceptual system view with good alignment

3.1 Mounting the receiver

Start by finding a suitable location in the vehicle to mount the receiver. The placement of the receiver does not matter in most applications. For best performance, the receiver should be placed as close to the center of the vehicle's rear axle as possible. If the application's accuracy requirements are very demanding, an advanced configuration may be needed.

Next, plan the orientation of the receiver. Figure 6 shows the inertial measurement unit (IMU) frame and the installation frame. The default assumption is that the x, y and z axes of the IMU frame are parallel to the x, y, and z axes of the installation frame. The IMU frame orientation is printed on the label of the C102-F9R evaluation kit; for custom designs refer to Figure 7. For the simplest possible installation, mount the receiver in the orientation described above. If the receiver is mounted in any other orientation, a custom configuration must be set for correct performance.

 Note that the configured misalignment must be within 3 degrees of the truth to prevent significant degradation in navigation performance. A misalignment of tens of degrees will result in failure. Consult section 3.2.3 of the ZED-F9R Integration manual [1] for more information about the IMU alignment. Configuring the misalignment is described in section 6.2.2.

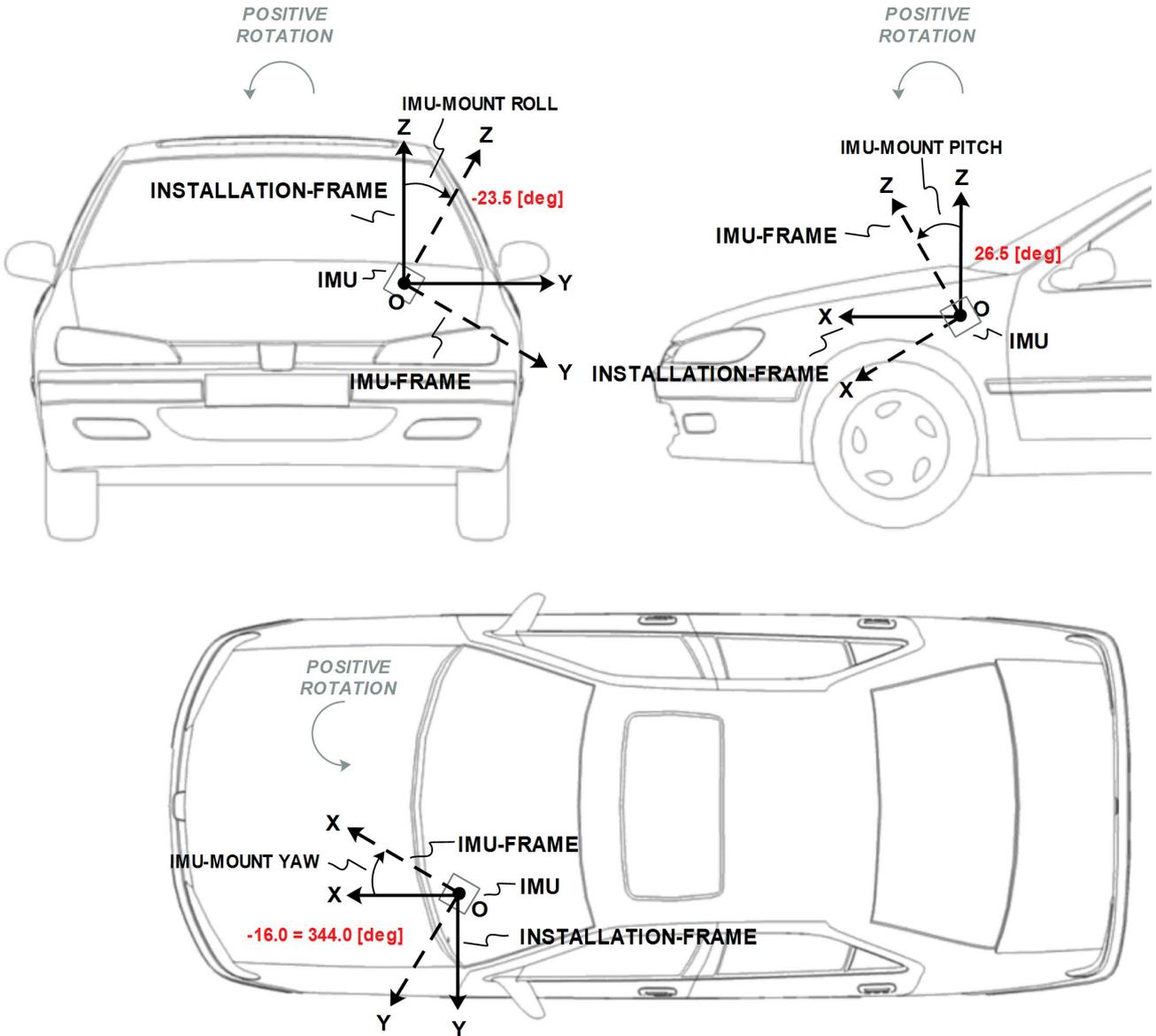


Figure 6: IMU frame and installation frame



Figure 7: Orientation information. Z-axis points upwards.

Finally, attach the receiver rigidly to the vehicle's frame. meaning that it must not be able to move relative to the vehicle. This also means that the receiver must not be attached to any part of the vehicle that is able to move. Any movement and excessive vibration will be detrimental to the navigation performance.

3.2 Mounting the antenna

Find a suitable location for the antenna. The antenna must be placed so that it has a clear view of the sky. For best performance, place the antenna as close to the receiver as possible.

If the antenna is an active patch antenna such as the u-blox ANN-MB, it needs a sufficient ground plane, e.g., a metallic disc with a ~15 cm diameter.

3.3 Connecting cables

Connect all the necessary cables, i.e., power, antenna, serial communication etc. to the receiver. Verify that the receiver is operational by repeating the procedure in 2.1

Once the serial connection to the receiver is established and verified, the installation part is complete.

3.4 Setting up UART configuration

If you are using UART as the main interface for communication, increase the receiver's baudrate. The default baudrate of 38400 may not be fast enough if the receiver is outputting a high number of messages. This can lead to data loss and even a complete failure in communication.

The recommended baudrate is 115200. This can be configured with CFG-UART1-BAUDRATE. Some applications may require even higher rates, especially if a higher navigation output rate is enabled. The utilization of the receiver's communication ports can be monitored with the UBX-MON-COMMS, UBX-MON-TXBUF and UBX-MON-RXBUF messages.

4 GNSS setup

Setting up the F9R for good GNSS performance is simple: just make sure the receiver has good quality signals. The default configuration of the receiver firmware is suitable for most use cases and does not need to be modified for most applications. Nevertheless, this chapter shows how to change the GNSS constellation configuration settings in case it is needed.

4.1 Verifying GNSS signals

1. Go to a location with an open sky view.
2. Connect the receiver to u-center.
3. Wait for the receiver to reach a 3D fix.
4. Check the available satellite signals in the Satellite Level view by clicking **View > Docking Windows > Satellite Level**:

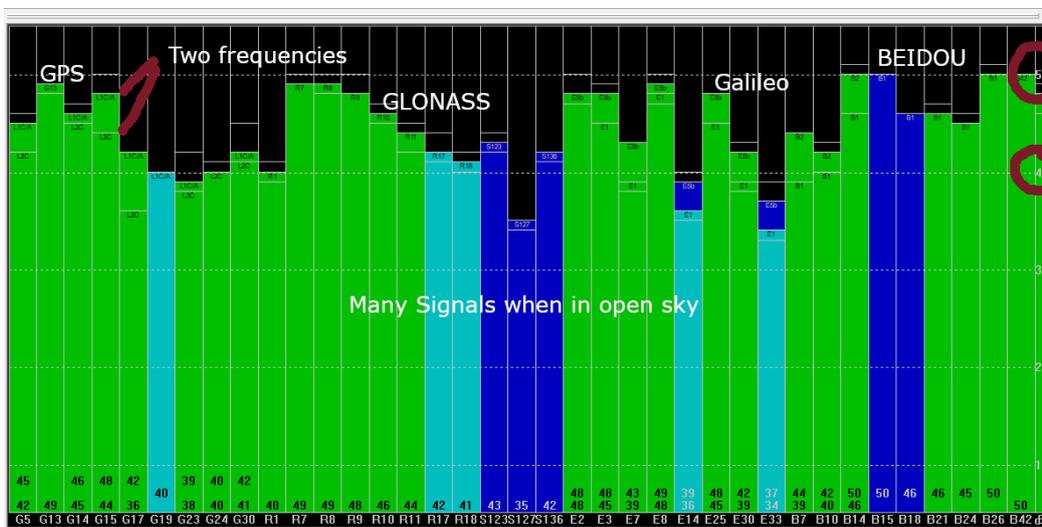


Figure 8: u-center Satellite Level view

Under open sky conditions, you should have the following result:

- At least 20 signals are available
- Average C/N0 value of used signals is at least 40
- Expected constellations are used
- Signals from multiple bands are available, e.g., L1 and L2 for GPS

If the signal quality is bad, improve it before continuing. Bad GNSS signals will result in bad navigation performance in all conditions. Likely reasons for poor signal quality are bad antenna, missing ground plane and obstructed sky view.

In u-center, verify that the position is reported in the Data view by clicking **View > Docking Windows > Data**.

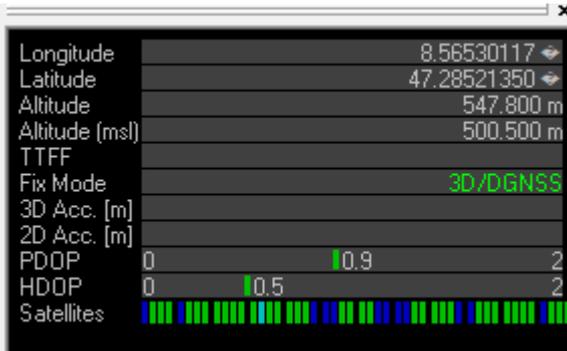


Figure 9: u-center Data view

4.2 Changing enabled constellations

The enabled satellite constellations and signals are controlled by configuration items in the CFG-SIGNAL group. A constellation is enabled when the constellation's enable key and both the L1 and L2 band keys are set to 1. Disabling a constellation can be done by setting the constellation's enable key to 0.

For example:

- To disable the BeiDou constellation, set the configuration item CFG-SIGNAL-BDS_ENA to 0.
- To enable the BeiDou constellation, set the configuration items CFG-SIGNAL-BDS_ENA, CFG-SIGNAL-BDS_B1_ENA and CFG-SIGNAL-BDS_B2_ENA to 1.

5 RTK setup

For proper RTK performance, the receiver requires a continuous feed of correction data. It is available from correction data providers through either NTRIP or MQTT protocols, each requiring application software for fetching the data from the provider's server and feeding it to the receiver through serial ports. This chapter shows how to use both the NTRIP and MQTT client on u-center, and how to monitor the RTK status of the receiver.

5.1 Setting up NTRIP client in u-center

There are both commercial and free NTRIP services available. For hobbyists and early prototyping, a good option is [RTK2go](#), a free community-driven NTRIP service where correction data streams from other users are available for use. For production-grade applications, more robust commercial services are recommended.

Start using the u-center NTRIP client with the following steps:

1. Open u-center and connect to the receiver via **Receiver > Connection**.
2. Open the NTRIP client settings from **Receiver > NTRIP Client**.
3. Fill in the NTRIP caster settings fields.
4. To fetch the available mount points from the service, click the **Update source table** button.
5. Select the correct mount point from the dropdown menu.
6. Press **OK** to start the NTRIP client.

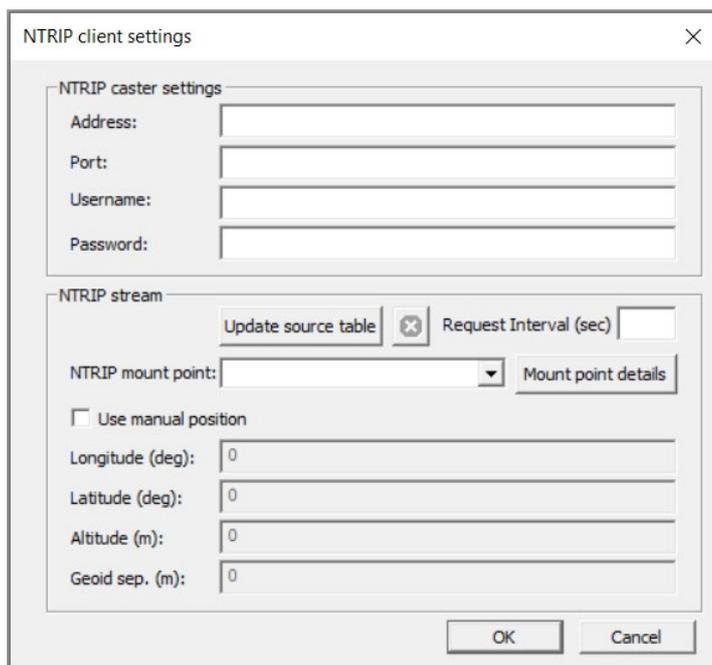


Figure 10: u-center NTRIP client

The status bar at the bottom of the u-center window provides information on the status of the service for monitoring and debugging purposes. When the authentication is successful and the service is connected, the connection symbol turns green.



Figure 11: u-center status bar

To view more details of the NTRIP client's operation, click the connection symbol. The **NTRIP Log** window is displayed.

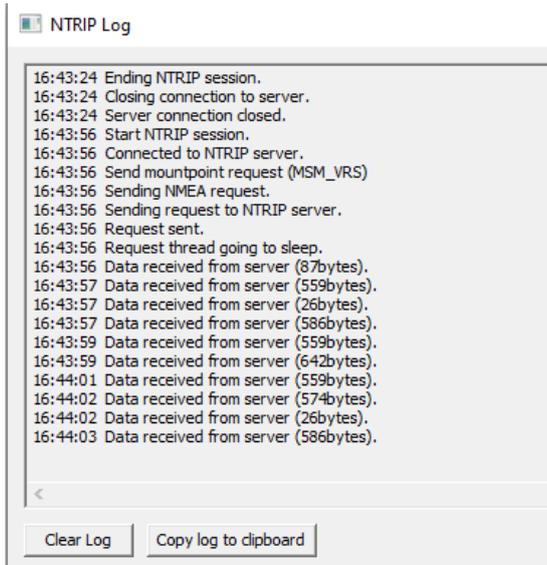


Figure 12: NTRIP client log

5.2 Setting up MQTT client in u-center

IoT devices and applications require a reliable, robust, and secure messaging protocol. That is where MQTT comes in. MQTT is an OASIS standard messaging protocol for the Internet of Things (IoT). It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth. (Source: <https://mqtt.org/>)

PointPerfect is a high performance GNSS augmentation service that enables high accuracy with high precision GNSS receivers. PointPerfect adopts the industry-driven SPARTN messaging format. SPARTN is an industry-led format that enables the highly efficient transfer of GNSS correction data.

The service uses MQTT as the basic transport mechanism for the different elements of the service. This includes authentication, ancillary services like AssistNow, service key delivery and the actual service. u-center implements a MQTT client.

Start using the u-center MQTT client with the following steps:

1. Open u-center and connect to the receiver via **Receiver > Connection**.
2. Open the MQTT client settings from **Receiver > MQTT Client**.
3. Download a u-center config file from the Thingstream portal associated with your device and region of operation and add it as the "JSON config file".
4. Check the box to "Subscribe to key topic".
5. Check the box to "Subscribe to AssistNow topic".
6. Check the box to "Subscribe to data topic".
7. Select the data topic using the drop-down menu suitable for the region of operation.
8. Press **OK** to start the MQTT client.

For more information on the PointPerfect service, please refer to: [PointPerfect getting started guide](#)

The status bar at the bottom of the u-center window provides information on the status of the service for monitoring and debugging purposes. When the authentication is successful and the service is connected, the connection symbol turns green.



Figure 13: Bottom Status bar of u-center

To view more details of the MQTT client's operation, click the connection symbol. The **MQTT Log** window is displayed.

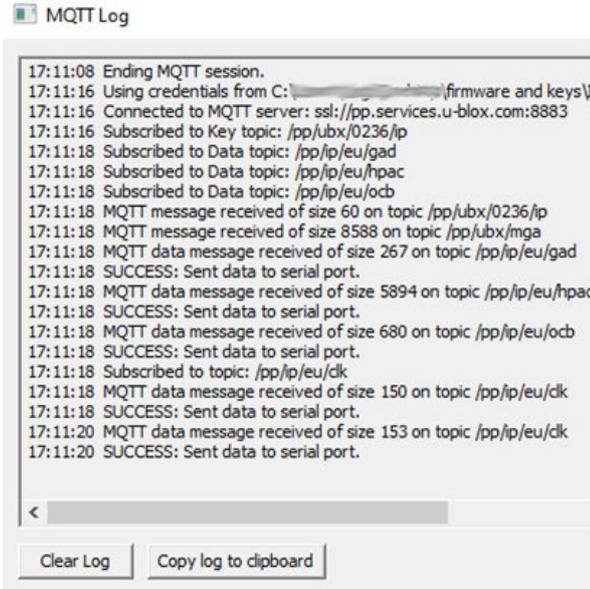


Figure 14: MQTT client log

5.3 Monitoring RTK status

Make sure the receiver is getting correction data by monitoring the RTK status in u-center (see Figure 15):

- In the general information view (**View > Docking Windows > Data**), the RTK status is shown in the **Fix status** field as “Float” or “Fixed” if RTK is used.
- Alternatively, in the UBX-NAV-PVT message view, the RTK status is shown in the Carrier Range Status field as “Float” or “Fixed” if RTK is used.

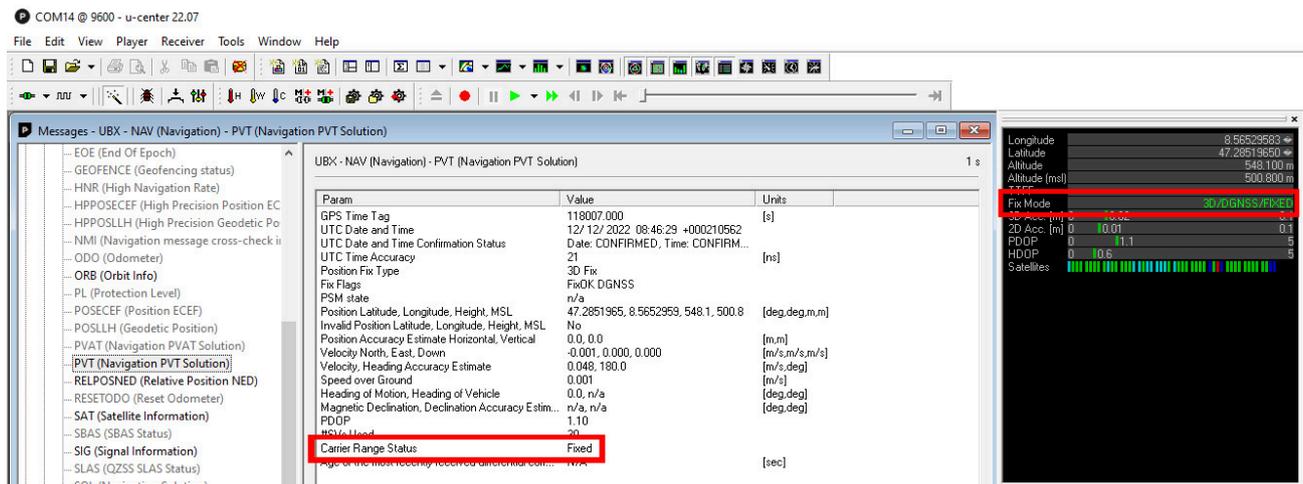


Figure 15: RTK status in u-center

Under open sky scenarios, if the receiver gets to a fixed state in less than two minutes, the receiver, antenna and correction service are compatible and working well.

6 Sensor fusion setup

The final and most important part of the setup is setting up the receiver for sensor fusion. The receiver depends heavily on the internal IMU data, odometer data from the vehicle, and proper configuration.

6.1 Providing odometer data to the receiver

As mentioned in 1.1.4, the receiver requires either wheel ticks or speed data to be supplied to it. Wheel ticks may be sent through either the dedicated pins for wheel ticks and direction, or through a serial interface as UBX-ESF-MEAS messages. Speed data may be sent through a serial interface only. The rest of the chapter covers cases where the serial interface is used. Consult chapter 3.2.5 of the ZED-F9R Integration manual [1] for a detailed description of wheel ticks and speed.

6.1.1 Wheel ticks

Wheel ticks represent the rotation of a wheel: every rotation generates a constant number of ticks. Through the wheel tick sensor resolution (ticks per revolution) and wheel diameter, the information can be converted into the distance travelled by the wheel.

The wheel ticks should be absolute, meaning the value starts from zero and increments regardless of the direction of travel. The direction is indicated with its own bit in the input message.

Wheel ticks can be input from either a single source (single tick) or from both the rear-left and rear-right wheel of a vehicle. When providing a single tick, the value should represent the vehicle's movement at the center of the rear axle. This can be achieved by averaging the wheel tick measurement of both rear-wheels. Providing rear-wheel ticks is preferred whenever available. The UBX-ESF-MEAS data types for wheel ticks are:

- single tick = 10
- rear-left wheel tick = 8
- rear-right wheel tick = 9

For optimal performance, the wheel tick resolution should be less than 5 cm, i.e., one tick should correspond to a displacement of 5 cm.

6.1.2 Speed

Speed is input as a signed value in millimeters per second. The UBX-ESF-MEAS data type for speed is 11.

6.1.3 Data stream quality

When sending odometer data with UBX-ESF-MEAS messages, ensure that the data stream is of good quality. For optimal performance:

- Supply odometer data at a rate of 10 Hz.
- Send data at even intervals, e.g., every 100 ms for 10 Hz input.
- Data must be sent with delay less than 10 % of the sampling interval.
- Data loss must be less than 1 %.
- Give odometer data priority over other communication with the receiver. If the odometer data stream is being hampered by other data, a separate serial port may be used.
- Use the highest baudrate available to minimize the latency caused by transmission.

6.1.4 UBX-ESF-MEAS sample code

u-blox has collaborated with the [OpenMower project](#) to provide an example of integrating the ZED-F9R in a robotic application. The u-blox driver source code can be found [here](#).

In the OpenMower project, wheel ticks from two individual wheels are provided to the receiver. The data is sent to the receiver with the `GpsInterface::send_wheel_ticks` function. The function takes in the sensor timestamp, and the direction and wheel tick value from the two wheels. These are put into an array, together with the UBX-ESF-MEAS message class and identifier bytes. Next, the `GpsInterface::send_packet` function completes the message with the two-byte UBX protocol header in the beginning and the two-byte checksum in the end. The complete message is sent to the receiver with the `GpsInterface::send_raw` function.

When wheel tick messages are being delivered to the receiver, use the UBX-ESF-MEAS graph view in u-center to monitor the wheel tick data. The graphs below show two systems. In the top picture the wheel ticks are combined into a single tick with `CFG-SFODO-COMBINE_TICKS = 1`. In the bottom picture both left and right wheel ticks are being used. Both pictures show the vehicle moving at a constant velocity.

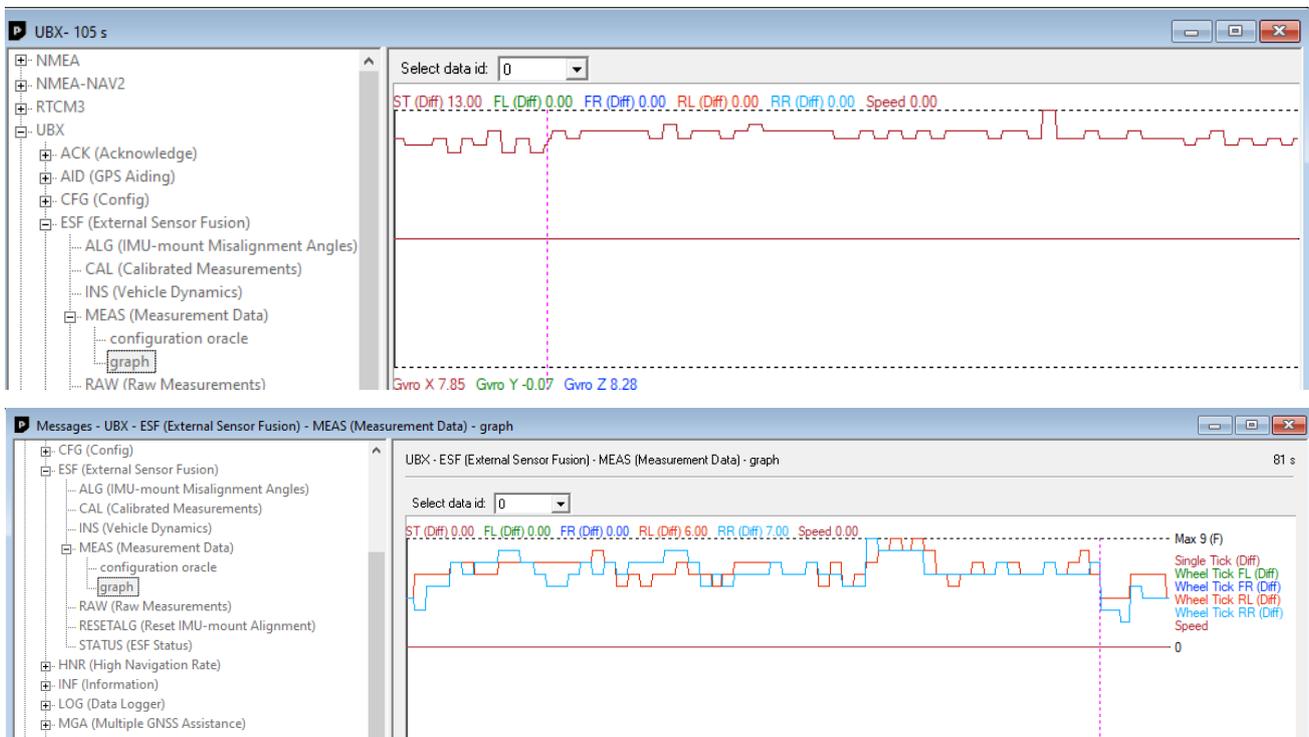


Figure 16: u-center UBX-ESF-MEAS graph view of single tick (top) and left/right (bottom) wheel tick systems

6.2 Sensor fusion configuration

The sensor fusion configuration of the receiver depends on several different aspects of the entire setup, including the used vehicle, how the receiver is mounted on the vehicle, and what type of odometer data is used. Thus, the configuration needs to be modified if the setup is changed.

6.2.1 Dynamic platform model

ZED-F9R supports different dynamic platform models to adjust the high-precision sensor fusion navigation engine to the expected application, that is, a vehicle and its expected operation environment. It is important to select the correct model to ensure that the sensor fusion works as expected. Refer to chapter 3.2.2 of the ZED-F9R Integration manual [1] for more information about the dynamic models.

The following dynamic models are available:

- Automotive
 - For applications with equivalent dynamics to those of a passenger car
 - Configured with CFG-NAVSPG-DYNMODEL = 4
- Robotic lawn mower (RLM)
 - For applications with equivalent dynamics to those of a robotic lawn mower
 - Configured with CFG-NAVSPG-DYNMODEL = 11
- E-scooter
 - For applications with equivalent dynamics to those of an e-scooter or a bicycle
 - Configured with CFG-NAVSPG-DYNMODEL = 12

Select the correct model for the application and apply the configuration in u-center.

6.2.2 IMU-mount alignment

The receiver needs to know its orientation relative to the vehicle to perform sensor fusion. There are two ways for the receiver to have this information:

- Automatic IMU-mount alignment
- User-defined IMU-mount alignment

Automatic IMU-mount alignment allows the receiver to estimate the roll, pitch and yaw angles based on the vehicle's movements. This mode is **only available when using the automotive dynamic model**. To enable automatic alignment, set the configuration item CFG-SFIMU-AUTO_MNTALG_ENA to 1.

Using user-defined IMU-mount alignment is **mandatory when using the robotic lawn mower and e-scooter dynamic models**. It may also be used when using the automotive dynamic model. To configure the receiver to use the user-defined IMU-mount alignment, set the following configuration items:

- Disable automatic alignment by setting CFG-SFIMU-AUTO_MNTALG_ENA to 0
- Set the yaw angle with CFG-SFIMU-IMU_MNTALG_YAW
- Set the pitch angle with CFG-SFIMU-IMU_MNTALG_PITCH
- Set the roll angle with CFG-SFIMU-IMU_MNTALG_ROLL

Consult chapter 3.2.3 of the ZED-F9R Integration manual [1] on how to determine the individual angles. It can also be beneficial to use tools like [3] to help determine and visualize the Euler angles in the setup. Determining the angles can be complicated, but it needs to be done carefully to ensure proper navigation performance.

If the installation is changed, the user-defined IMU-mount alignment angles must be reconfigured to match the new installation.

6.2.3 Setting up odometer configuration

The odometer configuration depends on both

- The type of odometer data used in the setup, and
- The dynamic platform model.

Follow these steps to properly configure the receiver:

1. Ensure the receiver only reads odometer data from the input UBX-ESF-MEAS messages by disabling the hardware wheel tick pin. Set CFG-SFODO-USE_WT_PIN = 0.
2. If using either the RLM or e-scooter dynamic platform model, disable automatic wheel tick polarity detection by setting CFG-SFODO-DIS_AUTODIRPINPOL = 1. **Do this regardless of whether wheel ticks or speed data is used in the setup.**
3. If using speed data, set CFG-SFODO-USE_SPEED = 1 and CFG-SFODO-COMBINE_TICKS = 0. This ensures that the receiver uses speed data.
4. If using rear-wheel ticks, set CFG-SFODO-COMBINE_TICKS = 1.
5. If using single tick, set CFG-SFODO-COMBINE_TICKS = 0.
6. **If using wheel ticks and the RLM model**, set the wheel tick scaling factor with CFG-SFODO-FACTOR. The factor is the distance per one tick in micrometers.

 The wheel tick scaling factor can be determined by dividing the distance travelled by the increase in wheel ticks. Alternatively, the value can be calculated directly from the wheel circumference and the number of ticks per revolution. As a numerical example, for a wheel with a diameter of 20 cm and 600 ticks per revolution, the factor is $3.14159 * \frac{0.2}{600} * 10^6 \mu m \approx 1047 \mu m$, i.e., CFG-SFODO-FACTOR = 1047.

6.2.4 Navigation output rate

By default, the receiver outputs its navigation solution at 1 Hz. For applications that require a higher output rate, the priority navigation mode feature can be used. With priority navigation mode, the receiver can output at 1-30 Hz. Priority navigation mode rate can be configured with the CFG-RATE-NAV_PRI0 configuration item. Consult the ZED-F9R Integration manual [\[1\]](#) for more information.

7 Calibration and testing

Before being able to perform sensor fusion, the receiver must calibrate itself with vehicle dynamics and odometer data. This is done automatically over time by the receiver while the vehicle is being driven. Consult chapter 3.2.7 of the ZED-F9R Integration manual for more details about the calibration process.

This section expects that all the previous sections have been completed successfully. The following must be true:

- The receiver has been properly installed in the vehicle
- The receiver has been properly configured
- The receiver has achieved a good 3D GNSS fix
- The receiver has achieved an RTK fix
- Odometer data is supplied to the receiver

7.1 Initialization and calibration

The receiver starts in an initialization phase, during which it estimates all unknown parameters required for sensor fusion. The phase is triggered after a receiver cold start or a filter reset. The receiver is in initialization phase if the `UBX-ESF-STATUS.fusionMode` field is set to 0: `INITIALIZING`. In this phase, the receiver produces a GNSS-only output.

During the initialization phase the receiver does the following steps:

- IMU initialization: Unknown crucial IMU parameters such as sensor sampling frequency are estimated.
- IMU-mount alignment initialization: Done only when automatic alignment is enabled.
- INS initialization: Initial vehicle position, velocity and attitude need to be known with sufficient accuracy.
- Wheel tick & speed sensor initialization: WT/speed sensor parameters are estimated.

The initialization phase requires good GNSS signal conditions and periods when the vehicle is stationary and moving. Depending on the used dynamic platform model, additional criteria must also be met (see chapter 3.2.2 of the ZED-F9R Integration manual [\[1\]](#)).

Once the initialization is complete, the receiver enters fusion mode and continuous calibration begins. The receiver is in fusion mode if the `UBX-ESF-STATUS.fusionMode` field is set to 1: `FUSION`. The fusion filter starts to calibrate the sensors required for computing the fused navigation solution. As soon as the calibration reaches a state where optimal fusion performance can be expected, `UBX-ESF-STATUS.calibStatus` is set to 2 or 3: `CALIBRATED`. Figure 17 shows the `UBX-ESF-STATUS` message output on a fully calibrated receiver.

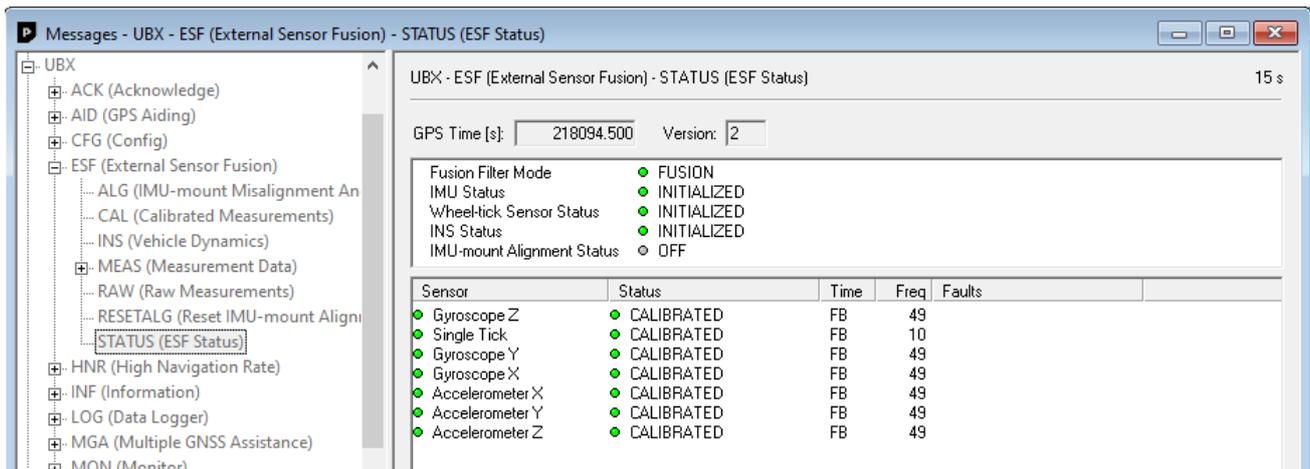


Figure 17: u-center UBX-ESF-STATUS showing calibrated and initialized system

While the receiver will get calibrated eventually during normal operation of the vehicle, an accelerated initialization and calibration procedure can be performed during evaluation. See section 3.2.7.2 of the ZED-F9R Integration manual [1].

7.2 Preserving calibration

Without preventive measures, the receiver calibration only persists while the receiver is powered on, i.e., the calibration is lost and the receiver needs to be re-calibrated after every power cycle. To preserve calibration, the following options are available:

- Backup supply voltage: Calibration is stored in the battery-backed RAM
- Save-on-shutdown feature
- Advanced calibration handling feature

Consult sections 3.9.3 (save-on-shutdown), 3.9.4 (advanced calibration handling) and 4.2.2 (backup supply voltage) of the ZED-F9R Integration manual [1] for more information.

7.3 Performing a basic test drive

The best way to confirm that the system is properly configured is to perform a recorded drive after configuration in open sky conditions. To see every stage of the receiver's operations, the test drive should start with a cold start. This way it is possible to analyze the time to first fix (TTFF), time to calibration and time to RTK solution.

In a proper setup, the receiver will consistently operate in a 3D+DR+RTK mode. Dropping out of sensor fusion and dropping out of RTK are evidence of something failing in the setup. It is recommended to validate correct functionality in open sky conditions before moving on to more difficult scenarios, such as urban canyon and DR-only scenarios. The appendix section contains some troubleshooting hints.

7.4 Scenario testing

Once the system is working reliably with sensor fusion and RTK, the entire system can be tested under different scenarios. The scenarios may include driving in open sky areas, areas with no sky visibility, wooded areas, areas near small and large buildings, and with different type of antennas or correction services.

A rigorous analysis will involve comparing the receiver's output to that of a truth system. If a truth system is not available, it is also possible to perform repeatability tests over different periods of time. If the system performance has high repeatability in different logs collected in different locations, it is very likely that the system is also quite accurate.

A Appendix

A.1 Recording logs

Follow these steps to record a logfile in u-center:

1. Open u-center and connect to the receiver
2. Enable the messages that need to be monitored
3. (Optional) Enable debug messages with the debug button (Figure 18, 1). This enables messages that are required for u-blox support to assist in debugging any receiver issues.



Figure 18 Logfile recording controls in u-center

4. Press the record button (Figure 18, 2).
5. When prompted to poll the receiver configuration, make sure u-blox Generation 9 is selected and click Yes.

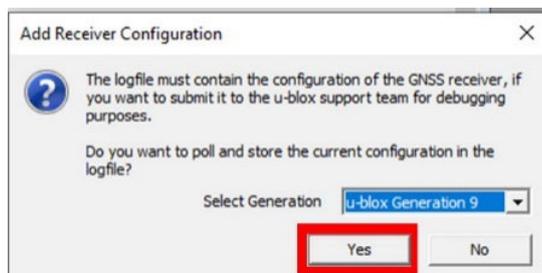


Figure 19 Polling the receiver configuration in u-center

A.2 Replaying logs

Once a log has been recorded, u-center allows the log to be loaded and various aspects of the drive scenario to be inspected in an offline mode.

Basics of replay include:

1. To open a log file, select **File > Open**
2. Set up the Views relevant to the issue at hand. For example, the Map view can help identify the relevant portion of the recording during a particular movement.
3. Set up the Message view with the message to monitor during the replay. For example, UBX-RXMCOR shows messages associated with correction messages being received while UBX-ESFSTATUS shows the status of the external sensor fusion which shows various aspects of the calibration process.
4. Press the **Player > Play > <message play rate>**

A.3 Interference of GNSS signals

The receiver's GNSS performance can be negatively affected by different sources of interference in the GNSS signals. This interference can be caused by:

- Cellular, Wi-Fi, Bluetooth and other wireless products
- Electric motors
- Power supplies

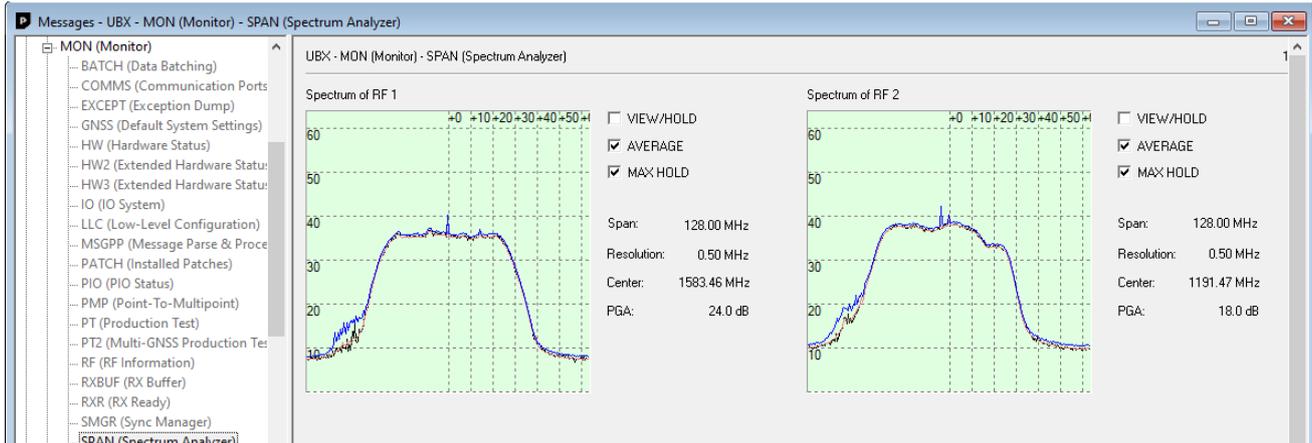


Figure 20: u-center UBX-MON-SPAN showing the spectrum at the L1 and L2 bands

A.4 Communications and protocols

One reason that a fixed ambiguity is not achieved may be the corrections not being received or used. Figure 21 shows the UBX-MON-COMMS message used for monitoring the traffic at the communication ports. When the receiver is outputting and getting corrections on UART1, the byte counter should be increasing for both the transmit and receive direction. If the protocol byte counters are not incrementing on the interface as expected, this could mean that either the protocol is disabled on the interface or the baudrate is incorrect.

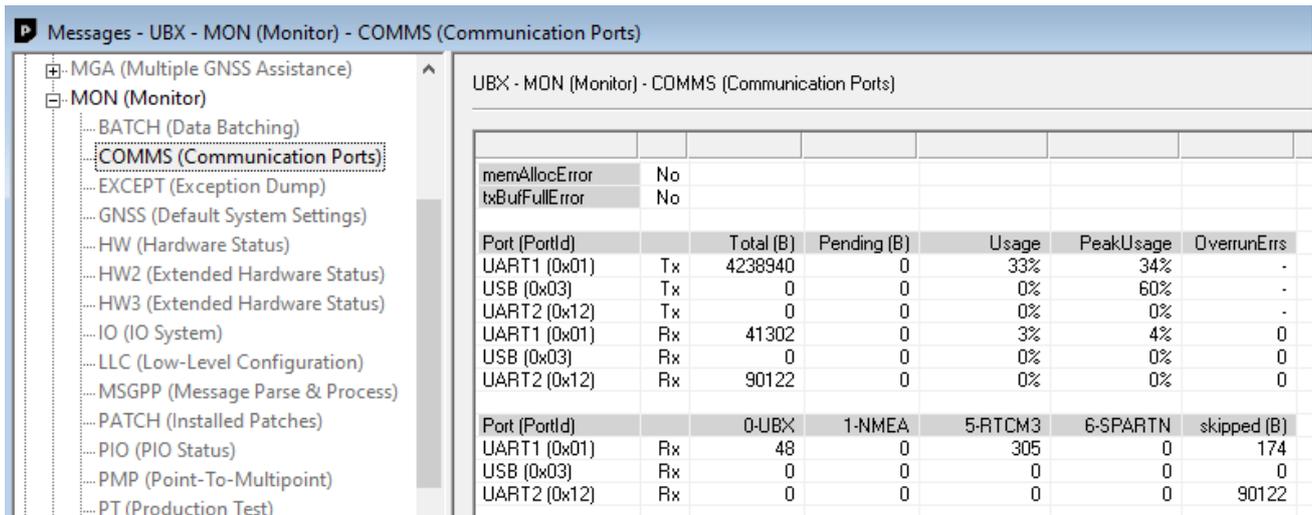


Figure 21: u-center UBX-MON-COMMS showing traffic count by interface and protocol

If the data is arriving, the messages that are being used are shown in UBX-RXM-COR. Below is an example of SPARTN corrections being received. The “can handle”, “Used” and “Decrypted” shows that a software handler exists, whether the message can be used and whether the message was successfully decrypted.

Protocol	Type - Subtype	Can handle	Used	Error Status	Correction ID	Type - Subtype	Encrypted	Decrypted	Eb/N0 [2°-3 dB]
SPARTN (2)	0-2	Yes (1)	Used (2)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	0-1	Yes (1)	Used (2)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	0-0	Yes (1)	Used (2)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	0-2	Yes (1)	Used (2)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	0-1	Yes (1)	Used (2)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	0-0	Yes (1)	Used (2)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	1-2	Yes (1)	Not used (1)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	1-2	Yes (1)	Used (2)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	1-2	Yes (1)	Not used (1)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	1-2	Yes (1)	Not used (1)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	1-1	Yes (1)	Not used (1)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	1-1	Yes (1)	Used (2)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	1-1	Yes (1)	Not used (1)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	1-1	Yes (1)	Not used (1)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	1-0	Yes (1)	Not used (1)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	1-0	Yes (1)	Used (2)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	1-0	Yes (1)	Not used (1)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0
SPARTN (2)	1-0	Yes (1)	Not used (1)	Error-free (1)	Unknown (0xFFFF)	Valid (1) - Valid (1)	Yes (2)	Yes (2)	0

Figure 22: u-center UBX-RXM-COR showing SPARTN messages being used and decrypted

A.5 Basic alignment & position offset checks

There are few drive tests to see whether lateral and longitudinal offsets are an issue.

- On a flat surface, drive in a straight line, turn around and return to the same spot. If there is an unacceptable offset between the out and back track, advanced configuration is needed.
- On a flat surface, stand still for one minute, drive in a straight line and turn around and return to the same spot where the rear wheels are in the same location, standstill for one minute. If there is an unacceptable offset between the large group of readings resulting from the standstill at the beginning and the large group of readings at the end, advanced configuration is needed.

There are few drive tests to see whether there are misalignment issues.

- On a flat surface, drive in a straight line for one minute and then turn left and drive in a straight line. If the mounting of up versus down, this will trigger a sensor fusion reset.
- On a flat surface, at constant speed, drive in a 2 m straight line, perform a smooth turn of arc length of 1 m to the right and continue straight. The recordings during the turn should be smooth without sawtooth like patterns. Repeat for a left turn. If the sawtooth patterns are always on the side of the arc for both turn or are always on the outside for both turns, there is a misalignment along the horizontal axis.

A.6 Odometry data issues

The following are some more rare issues with odometer data:

- Does the rollover value of the odometer data not fall at the powers of two (2^n-1)? E.g., 1023, 2047, 4095, etc. The rollover point needs to be configured with CFG-SFODO-COUNT_MAX.
- Does the vehicle travel at such a high speed that the values rollover more than once between two samples? The values need to be scaled to a smaller value while meeting the minimum resolution of the odometer data.

A.7 RTK never reaching fix

What are some of the common causes for not getting an RTK fix in open sky even though corrections are being received by the receiver?

- For a VRS service, are the RTCM messages compatible with the receiver? This can be checked by inspect the documentation of supported messages or by reviewing the message UBX-MON-COR
- For a Nearest station RTCM service, is the reference station close enough?
- Is the quality of the service good?
- For an SSR service, are the messages being decrypted?
- Is the service having some kind of outage?
- Does the service work with an ANN-MB antenna?

Protocol	Type - Subtype	Can handle	Used	Error Status	Correction ID	Type - Subtype	Encrypted	Decrypted
RTCM3 (1)	1094 - 0	Yes (1)	Used (2)	Error-free (1)	76	Valid (1) - Invalid (0)	N/A (0)	N/A (0)
RTCM3 (1)	1084 - 0	Yes (1)	Used (2)	Error-free (1)	76	Valid (1) - Invalid (0)	N/A (0)	N/A (0)
RTCM3 (1)	1074 - 0	Yes (1)	Used (2)	Error-free (1)	76	Valid (1) - Invalid (0)	N/A (0)	N/A (0)
RTCM3 (1)	1006 - 0	Yes (1)	Used (2)	Error-free (1)	76	Valid (1) - Invalid (0)	N/A (0)	N/A (0)
RTCM3 (1)	1032 - 0	No (0)	Not used (1)	Error-free (1)	76	Valid (1) - Invalid (0)	N/A (0)	N/A (0)
RTCM3 (1)	1094 - 0	Yes (1)	Used (2)	Error-free (1)	76	Valid (1) - Invalid (0)	N/A (0)	N/A (0)
RTCM3 (1)	1084 - 0	Yes (1)	Used (2)	Error-free (1)	76	Valid (1) - Invalid (0)	N/A (0)	N/A (0)
RTCM3 (1)	1074 - 0	Yes (1)	Used (2)	Error-free (1)	76	Valid (1) - Invalid (0)	N/A (0)	N/A (0)
RTCM3 (1)	1013 - 0	No (0)	Not used (1)	Error-free (1)	76	Valid (1) - Invalid (0)	N/A (0)	N/A (0)
RTCM3 (1)	1094 - 0	Yes (1)	Used (2)	Error-free (1)	76	Valid (1) - Invalid (0)	N/A (0)	N/A (0)
RTCM3 (1)	1084 - 0	Yes (1)	Used (2)	Error-free (1)	76	Valid (1) - Invalid (0)	N/A (0)	N/A (0)

Figure 23: u-center UBX-RXM-COR showing RTCM message types being received

A.8 Dropping out of dead reckoning periodically

- Have the physical dimensions and orientation of the system changed? Has the equipment been installed in the same position as before when it was working?
- Is the right dynamic model for the vehicle been selected?
- Is the kind of motion of your vehicle performs differ in some way from the dynamic model? Is the speed too high? Is it banking on turns when it is not allowed?
- Are there INFORMATION messages in the log with warnings?

A.9 Troubleshooting checklist

Installation related:

- Is receiver mounted in a fixed position relative to the vehicle frame?
- Is the antenna mounted directly above the receiver's IMU?
- Is the receiver's IMU directly above the center of the rear axle?
- Is u-center able to read the version of firmware?

GNSS related:

- Are there more than 20 satellites visible under open sky?
- Are there many signals with a CNO level between 40 and 50 dB?
- Are all four constellations reporting signals?
- Are there two signals on many of the signals?
- Did the receiver get a position with 3D/DGNSS position fix?

RTK related:

- When the NTRIP/MQTT client is enabled, is there a constant stream of data being received?
- Does the receiver report a 3D/DGNSS/Fixed solution under open sky with a correction service in less than 2 minutes?

Odometry data integration related:

- Is the odometry data meeting the specification for delay, loss and jitter?
- Is the highest possible baudrate being used to send odometry data?
- Does the u-center graph for UBX-ESF-MEAS graph show a relatively flat line when the vehicle is travelling at a constant speed in a straight line other than when the counter overflows?

Sensor Fusion related:

- Is the dynamic model appropriate for the vehicle type?
- Are the geometries of the vehicle up such that the antenna, the receiver and the middle of rear axle are relatively inline and within a few centimeters from each other?
- Have the various odometer configurations been double checked?

Calibration related:

- Is the calibration drive suitable for the vehicle type?
- Did the calibration drive result in a calibrated and initialized system?

Final test:

- During the test drive after calibration, does the carrier range status stay consistently in 3D/DR/Fixed mode?

Related documentation

- [1] ZED-F9R Integration Manual, [UBX-20039643](#)
- [2] F9 HPS 1.30 Interface Description, [UBX-22010984](#)
- [3] Euler angles visualization tool, available in <https://compsci290-s2016.github.io/CoursePage/Materials/EulerAnglesViz/>

 For product change notifications and regular updates of u-blox documentation, register on our website, www.u-blox.com.

Revision history

Revision	Date	Name	Comments
R01	9-Jan-2023	jilm	Initial release

Contact

u-blox AG

Address: Zürcherstrasse 68
8800 Thalwil
Switzerland

For further support and contact information, visit us at www.u-blox.com/support.