ZED-F9K

High Precision ADR with 3D sensors

Integration Manual

Abstract
Lane accurate positioning under challenging conditions. Decimeter level accuracy for automotive mass markets. Ideal for ADAS, V2X and head-up displays. Low risk multi-band RTK turnkey solution with built-in inertial sensors. Lag-free displays with up to 30 Hz real-time position update rate.

www.u-blox.com

UBX-18047567 - R02

Document Information

Title	**ZED-F9K**	
Subtitle	High Precision ADR with 3D sensors	
Document type	Integration Manual	
Document number	UBX-18047567	
Revision and date	R02	06-May-2019
Document status	Objective Specification – Confidential	

This document applies to the following products:

Product name	Type number	Firmware version	PCN reference
ZED-F9K	ZED-F9K-00B-00	LAP B03	N/A

CONFIDENTIAL

u-blox reserves all rights to this document and the information contained herein. Products, names, logos and designs described herein may in whole or in part be subject to intellectual property rights. Reproduction, use, modification or disclosure to third parties of this document or any part thereof without the express permission of u-blox is strictly prohibited.

The information contained herein is provided "as is" and u-blox assumes no liability for the use of the information. No warranty, either express or implied, is given with respect to, including but not limited 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. For most recent documents, please visit www.u blox.com.

Copyright © 2019, u-blox AG.

u-blox is a registered trademark of u-blox Holding AG in the EU and other countries.

Contents

1 Integration manual structure ... 6
2 System description ... 7
2.1 Overview ... 7
2.1.1 Automotive dead reckoning (ADR) ... 7
2.1.2 High navigation rate (HNR) ... 8
2.1.3 Real Time Kinematic ... 8
2.2 Architecture ... 8
2.2.1 Block diagram ... 8
3 Receiver functionality ... 9
3.1 Receiver configuration ... 9
3.1.1 Changing the receiver configuration ... 9
3.1.2 Default GNSS configuration ... 9
3.1.3 Default interface settings ... 10
3.1.4 Basic receiver configuration ... 10
3.1.5 RTCM corrections ... 11
3.1.6 Navigation configuration ... 13
3.2 Automotive dead reckoning (ADR) ... 14
3.2.1 Introduction ... 14
3.2.2 Solution type ... 14
3.2.3 Installation configuration ... 14
3.2.4 Sensor configuration ... 21
3.2.5 ADR system configuration ... 24
3.2.6 Operation ... 24
3.2.7 High navigation rate (HNR) ... 32
3.3 Interfaces ... 33
3.3.1 UART interfaces ... 34
3.3.2 SPI interface ... 34
3.3.3 USB interface ... 34
3.3.4 D_SEL interface ... 35
3.3.5 RESET_N interface ... 35
3.3.6 SAFEBOOT_N interface ... 35
3.3.7 TIMEPULSE interface ... 36
3.3.8 TX_READY interface ... 36
3.3.9 Display data channel (DDC) ... 36
3.3.10 Antenna supervisor ... 36
3.3.11 EXTINT ... 39
3.3.12 WT and DIR inputs ... 39
3.3.13 Communication ports ... 40
3.4 Multiple GNSS assistance (MGA) ... 46
3.4.1 AssistNow Online ... 46
3.4.2 Host software ... 47
3.4.3 AssistNow Online sequence ... 47
3.4.4 Flow control ... 48
3.4.5 Authorization ... 48

CONFIDENTIAL

3.4.6 Service parameters....48
3.4.7 Multiple servers....50
3.5 Clocks and time....50
3.5.1 Receiver local time....50
3.5.2 Navigation epochs....51
3.5.3 iTOW timestamps....51
3.5.4 GNSS times....52
3.5.5 Time validity....52
3.5.6 UTC representation....52
3.5.7 Leap seconds....53
3.5.8 Real time clock....53
3.5.9 Date....53
3.5.10 Time pulse....54
3.5.11 Timemark....58
3.6 Security (operating, monitoring and maintaining)....59
3.6.1 Receiver status monitoring....59
3.6.2 Receiver message authentication....61
3.6.3 Spoofing detection / monitoring....61
3.7 Broadcast navigation data....62
3.7.1 Parsing navigation data subframes....62
3.7.2 GPS....62
3.7.3 GLONASS....63
3.7.4 BeiDou....64
3.7.5 Galileo....64
3.7.6 QZSS....66
3.7.7 Summary....67
3.8 Forcing a receiver reset....67
3.9 Firmware update description....68
3.9.1 Capabilities....68
3.9.2 Firmware update to flash....68
3.9.3 Firmware Update Sample Code....70
3.9.4 Firmware Verification....71
4 Design....72
4.1 Pin assignment....72
4.2 Power supply....73
4.2.1 VCC: Main supply voltage....74
4.2.2 V_BCKP: Backup supply voltage....74
4.2.3 ZED-F9K power supply....74
4.3 ZED-F9K minimal design....75
4.4 WT and DIR interface example....76
4.5 Antenna....77
4.5.1 Antenna bias....78
4.6 EOS/ESD precautions....81
4.6.1 ESD protection measures....81
4.6.2 EOS precautions....81
4.6.3 Safety precautions....82
4.7 Electromagnetic interference on I/O lines....82
4.7.1 General notes on interference issues....82
4.7.2 In-band interference mitigation....83
4.7.3 Out-of-band interference....83

CONFIDENTIAL

4.8 Layout....83
4.8.1 Placement....84
4.8.2 Module footprint and paste mask....84
4.8.3 Layout guidance....86
4.9 Design guidance....88
4.9.1 General considerations....88
4.9.2 backup battery....88
4.9.3 RF front-end circuit options....88
4.9.4 Antenna/ RF input....89
4.9.5 Ground pads....90
4.9.6 Schematic design....90
4.9.7 Layout design-in guideline....90
5 Product handling....91
5.1 ESD handling precautions....91
5.2 Soldering....91
5.3 Tapes....94
5.4 Reels....95
5.5 Moisture sensitivity levels....95
Appendix....96
A Glossary....96
B RTCM ITRF Geodetic models....96
C Virtual reference service....98
D RTK configuration procedures with u-center....99
D.1 Receiver configuration with u-center....100
Related documents....105
Revision history....106

CONFIDENTIAL

1 Integration manual structure

This document provides a wealth of information to enable a successful design with the ZED-F9K module. The manual is structured according to system, software and hardware aspects.

The first section, "System description" outlines the basics of enabling RTK operation with the ZED-F9K. This is essential reading for anyone new to the device to enable them to understand a working RTK implementation.

The following section "Receiver functionality" provides an exhaustive description of the receiver's functionality. Beginning with the new configuration messages, both existing and new users should read this section to understand the new message types employed. Most of the following sub-sections should be familiar to existing users of u-blox positioning products, however some changes are introduced owing to the new configuration messages.

The sections from "Design" onwards address hardware options when designing the ZED-F9K into a new product. This part gives power supply recommendations and provides guidance for circuit design and PCB lay-out assistance. An antenna section provides design information and recommendation for this important component. A final "Design guidance" section helps the designer to check that crucial aspects of the design-in process have been carried out.

The final section addresses the general product handling concerns giving guidance on ESD precautions, production soldering considerations and module delivery tape and reel information.

CONFIDENTIAL

2 System description

2.1 Overview

The ZED-F9K module features the u-blox F9 multi-band L1/L2 GNSS receiver with rapid convergence time within seconds. This mass-market component provides decimeter level positioning with high availability, while making use of all 4 GNSS constellations simultaneously.

It is the first dead-reckoning module with an integrated Inertial Measurement Unit (IMU) capable of high precision positioning. The sophisticated built-in algorithms fuse the IMU data, GNSS measurements, wheel ticks (or speed in m/s), and vehicle dynamics model to provide lane accurate positioning where GNSS alone would fail. The module operates under open-sky motorways, in the wooded countryside, in difficult urban environments, and even in tunnels and underground parking. In modern automotive applications, such as Advanced Driver Assistance System (ADAS) where availability can improve the safety of our roads, ZED-F9K is the ultimate solution.

The device is a turnkey solution eliminating the technical risk of integrating third party libraries, precise positioning engines, and the multi-faceted hardware engineering aspects of radio frequency design and digital design. The u-blox approach provides a transparent evaluation of the positioning solution, provides clear lines of responsibility for design support while reducing supply chain complexity during production.

ZED-F9K is ideal for innovative automotive architecture designs with limited space and power. The module provides accurate location services to the increasing number of intelligent Electronic Control Units (ECU) such as Telematics Control Unit, Navigation system, Infotainment and V2x safety systems.

The module reaches a high navigation output rate of up to 30 Hz. The on-board processor augments fused GNSS position with additional IMU-based position estimates. Drivers experience responsive, lag-free user interfaces. ZED-F9K can output raw IMU and raw GNSS data for advanced applications.

The module supports a range of leading correction services. ZED-F9K comes with built-in support for RTCM formatted OSR-type corrections, and in a future release, SSR-type corrections. SSR-based techniques use data sparingly and are suitable for both IP-based cellular networks and broadcast- based satellite connectivity. The unique combination ensures the reliable distribution of correction services scalable to millions of vehicles using mobile network in populated areas and satellites for remote areas.

ZED-F9K modules are manufactured in ISO/TS 16949 certified sites and are fully tested on a system level. Qualification tests are performed as stipulated in the ISO 16750 standard: “Road vehicles–Environmental conditions and testing for electrical and electronic equipment”.

2.1.1 Automotive dead reckoning (ADR)

Automotive Dead Reckoning (ADR) provides high-accuracy positioning in locations with poor or no GNSS coverage. ADR is based on Sensor Fusion Dead Reckoning (SFDR) technology, which combines multi-constellation GNSS measurements with the ZED-F9K's internal 6-axis IMU and wheel tick or speed.

See the ADR section in this document for more information.

CONFIDENTIAL

2.1.2 High navigation rate (HNR)

Higher Navigation Rate (HNR) provides low-latency position, velocity and attitude solutions at a higher rate than the configured GNSS measurement rate.

Sensor measurements are used to propagate the solution at the higher rate between GNSS epochs.

See the HNR section in this document for more information.

2.1.3 Real Time Kinematic

u-blox ZED-F9K high precision receiver takes GNSS precision to the next level:

- Delivers accuracy down to the decimeter level
- Fast time to first fix and robust performance with multi-band, multi-constellation reception
- Compatible with leading correction services for global coverage and versatility

2.2 Architecture

The ZED-F9K module provides all the necessary RF and baseband processing to enable multi-band, multi-constellation operation. The block diagram below shows the key functionality implemented in the module.

2.2.1 Block diagram

Figure 1: ZED-F9K block diagram

CONFIDENTIAL

3 Receiver functionality

This section describes the ZED-F9K operational features and their configuration.

3.1 Receiver configuration

The ZED-F9K is fully configurable with UBX configuration interface keys. The configuration database in the receiver's RAM holds the current configuration, which 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 fully in the ZED-F9K Interface Description [2].

A configuration setting stored in RAM remains effective until power-down or reset. If stored in BBR (battery backed RAM), the setting will be used as long as the backup battery supply remains. Configuration settings can be saved permanently in flash memory.

☞ The configuration interface has changed from earlier u-blox positioning receivers. There is some backwards compatibility however, users are strongly advised to adopt the configuration interface described in this document. See legacy UBX-CFG message fields reference section in the ZED-F9K Interface Description [2].

Configuration interface settings are held in a database consisting of separate configuration items. An item is made up of a key ID and value pair. Related items are grouped together and identified under a common group name: CFG-GROUP-ITEM; a convention used in u-center and within this document. Within u-center, a configuration group is identified as "Group name" and the configuration item is identified as the "item name" under the "Generation 9 Configuration View" - "Advanced Configuration" view.

The UBX messages available to change or poll the configurations are the UBX-CFG-VALSET, UBX-CFG-VALGET, and UBX-CFG-VALDEL messages. For more information about these messages and the configuration keys see the configuration interface section in the ZED-F9K Interface Description [2].

3.1.1 Changing the receiver configuration

The configuration messages UBX-CFG-VALSET, UBX-CFG-VALGET and UBX-CFG-VALDEL, will result in a UBX-ACK-ACK or a UBX-ACK-NAK response.

3.1.2 Default GNSS configuration

The ZED-F9K default GNSS configuration is set as follows:

- GPS: L1C/A, L2C
- GLONASS: L1OF, L2OF
- Galileo: E1B/C, E5b
- BeiDou: B1I
- QZSS: L1C/A, L2C

BeiDou B2I is also supported but not enabled in the default GNSS configuration.

For more information about default configuration, see the ZED-F9K Interface Description [2].

CONFIDENTIAL

3.1.3 Default interface settings

Interface	Settings
UART1 output	38400 Baud, 8 bits, no parity bit, 1 stop bit. NMEA **GGA, GLL, GSA, GSV, RMC, VTG, TXT** (and no UBX) messages are output by default.
UART1 input	38400 Baud, 8 bits, no parity bit, 1 stop bit. UBX, NMEA and RTCM 3.3 messages are enabled by default.
UART2 output	38400 Baud, 8 bits, no parity bit, 1 stop bit. **No host interface support (UBX).** Output disabled by default.
UART2 input	38400 Baud, 8 bits, no parity bit, 1 stop bit. **No Host interface support (UBX).** RTCM 3.3 protocol enabled by default.
USB output	NMEA **GGA, GLL, GSA, GSV, RMC, VTG, TXT** (and no UBX) messages are output by default.
USB input	UBX, NMEA, RTCM 3.3 protocols enabled by default.
DDC	Fully compatible with the I^2C industry standard, available for communication with an external host CPU or u-blox cellular modules, operated in slave mode only. Default messages activated as in UART1. Input/output protocols available as in UART1. Maximum bit rate 400 kb/s.
SPI	Allow communication to a host CPU, operated in slave mode only. Default messages activated as in UART1. Input/output protocols available as in UART1. SPI is not available unless D_SEL pin is set to low (see section D_SEL interface in ZED-F9K Integration Manual).

Table 1: Default interface settings

UART2 can be configured as an RTCM interface. RTCM 3.3 is the default input protocol. UART2 may also be configured for NMEA output, typically NMEA GGA output for use with Virtual reference service correction services.

By default the ZED-F9K outputs NMEA 4.10 messages that include satellite data for all GNSS bands being received. This results in a higher-than-before NMEA load output for each navigation period. Make sure the UART1 baud rate being used is sufficient for the selected navigation rate and the number of GNSS signals being received.

3.1.4 Basic receiver configuration

This section summarizes the basic receiver configuration most commonly used.

3.1.4.1 Communication interface configuration

Several configuration groups allow operation mode configuration of the various communications interfaces. These include parameters for the data framing, transfer rate and enabled input/output protocols. See Communication ports section for details. The configuration groups available for each interface are:

Interface	Configuration groups
UART1	CFG-UART1-*, CFG-UART1INPROT-*, CFG-UART1OUTPROT-*
UART2	CFG-UART2-*, CFG-UART2INPROT-*, CFG-UART2OUTPROT-*
USB	CFG-USB-*, CFG-USBINPROT-*, CFG-USBOUTPROT-*
I^2C	CFG-I2C-*, CFG-I2CINPROT-*, CFG-I2COUTPROT-*
SPI	CFG-SPI-*, CFG-SPIINPROT-*, CFG-SPIOUTPROT-*

Table 2: Default configurations

3.1.4.2 Message output configuration

The rate of NMEA and UBX protocol output messages are configurable.

If the rate configuration value is zero, then the corresponding message will not be output. Values greater than zero indicate how often the message is output.

CONFIDENTIAL

For periodic output messages the rate relates to the event the message is related to. For example, the UBX-NAV-PVT (navigation position velocity and time solution) is related to the navigation epoch. If the rate of this message is set to one (1), it will be output for every navigation epoch. If the rate is set to two (2), it will be output every other navigation epoch. The rates of the output messages are individually configurable per communication interface. See the CFG-MSGOUT-* configuration group.

Some messages, such as UBX-MON-VER, are not periodic and will only be output as the answer to a poll request.

The UBX-INF-* information messages are non-periodic output messages that do not have a message rate configuration. Instead they can be enabled for each communication interface via the CFG-INFMSG-* configuration group.

☞ All message output is additionally subject to the protocol configuration of the communication interfaces. Messages of a given protocol will not be output until the protocol is enabled for output on the interface (see previous section).

3.1.4.3 GNSS signal configuration

The GNSS constellations and bands are configurable with configuration keys. Each GNSS constellation can be enabled or disabled independently. A GNSS constellation is considered to be enabled when the constellation enable key is set and at least one of the constellation's band keys are enabled.

ZED-F9K only supports certain combinations of constellations and bands. For all constellations, both L1 and L2 bands must either be enabled or disabled. BeiDou B2 is the exception (can either have BeiDou B1+B2 or B1-only). Unsupported combinations will be rejected with a UBX-ACK-NAK and the warning: "invalid sig cfg" will be sent via UBX-INF and NMEA-TXT messages (if enabled).

The following table shows possible configuration key combinations for the GPS constellation.

Constellation key CFG-SIGNAL-GPS_ENA	Band key CFG-SIGNAL-GPS_L1CA_ENA	Band key CFG-SIGNAL-GPS_L2C_ENA	Constellation enabled?
false (0)	false (0)	false (0)	**no**
false (0)	false (0)	true (1)	**no**
false (0)	true (1)	false (0)	**no**
false (0)	true (1)	true (1)	**no**
true (1)	false (0)	false (0)	**no**
true (1)	false (0)	true (1)	**Unsupported combination**
true (1)	true (1)	false (0)	**Unsupported combination**
true (1)	true (1)	true (1)	**yes**

Table 3: Example of possible values of configuration items for the GPS constellation

3.1.5 RTCM corrections

RTCM is a binary data protocol for communication of GNSS correction information. The ZED-F9K high precision receiver supports RTCM as specified by RTCM 10403.3, Differential GNSS (Global Navigation Satellite Systems) Services – Version 3 (October 7, 2016).

The RTCM specification is currently at version 3.3 and RTCM version 2 messages are not supported by this standard. Users can download the standard from the RTCM website here.

CONFIDENTIAL

To modify the RTCM input/output settings, see the configuration section in the u-blox ZED-F9K Interface Description [2].

Users need to be aware of the datum used by the correction source. The receiver position will provide coordinates in the correction source reference frame. This may need to be taken into account when using the RTK-derived position. See the RTCM ITRF Geodetic models section in the Appendix for more information.

3.1.5.1 List of supported RTCM input messages

Message	Description
RTCM 1001	L1-only GPS RTK observables
RTCM 1002	Extended L1-only GPS RTK observables
RTCM 1003	L1/L2 GPS RTK observables
RTCM 1004	Extended L1/L2 GPS RTK observables
RTCM 1005	Stationary RTK reference station ARP
RTCM 1006	Stationary RTK reference station ARP with antenna height
RTCM 1007	Antenna descriptor
RTCM 1009	L1-only GLONASS RTK observables
RTCM 1010	Extended L1-only GLONASS RTK observables
RTCM 1011	L1/L2 GLONASS RTK observables
RTCM 1012	Extended L1/L2 GLONASS RTK observables
RTCM 1033	Receiver and antenna description
RTCM 1074	GPS MSM4
RTCM 1075	GPS MSM5
RTCM 1077	GPS MSM7
RTCM 1084	GLONASS MSM4
RTCM 1085	GLONASS MSM5
RTCM 1087	GLONASS MSM7
RTCM 1094	Galileo MSM4
RTCM 1095	Galileo MSM5
RTCM 1097	Galileo MSM7
RTCM 1124	BeiDou MSM4
RTCM 1125	BeiDou MSM5
RTCM 1127	BeiDou MSM7
RTCM 1230	GLONASS code-phase biases
RTCM 4072.0, subtype 0	Reference station PVT (u-blox proprietary RTCM Message)
RTCM 4072.1, subtype 1	Additional reference station information (u-blox proprietary RTCM Message)

Table 4: ZED-F9K supported input RTCM version 3.3 messages

3.1.5.2 NTRIP - Networked Transport of RTCM via Internet Protocol

Networked Transport of RTCM via Internet Protocol, or NTRIP, is a standard protocol for streaming differential data over the Internet in accordance with specifications published by RTCM.

There are three major parts to the NTRIP system: The NTRIP client, the NTRIP server, and the NTRIP caster:

1. The NTRIP server is a PC or on-board computer running NTRIP server software communicating directly with a GNSS reference station. The NTRIP server serves as the

CONFIDENTIAL

intermediary between the GNSS receiver (NTRIP Source) streaming RTCM data and the NTRIP caster.

2. The NTRIP caster is an HTTP server which receives streaming RTCM data from one or more NTRIP servers and in turn streams the RTCM data to one or more NTRIP clients via the Internet.
3. The NTRIP client receives streaming RTCM data from the NTRIP caster to apply as real-time corrections to a GNSS receiver.

u-center GNSS evaluation software provides an NTRIP client application that can help to evaluate a ZED-F9K design. The u-center NTRIP client connects over the Internet to an NTRIP service provider, using access credentials such as user name and password from the service provider. The u-center NTRIP client then forwards the RTCM 3.3 corrections to a ZED-F9K receiver connected to the local u-center application. RTCM corrections from a virtual reference service are also supported by the u-center NTRIP client.

NTRIP documentation can be downloaded from the RTCM standards website.

3.1.6 Navigation configuration

This section presents various configuration options related to the navigation engine. These options can be configured through various configuration groups, such as CFG-NAVSPG-*, CFG-ODO-*, and CFG-MOT-*.

3.1.6.1 Navigation input filters

The navigation input filters in CFG-NAVSPG-* configuration group provide the input data of the navigation engine.

Configuration item	Description
CFG-NAVSPG-FIXMODE	By default, the receiver calculates a 3D position fix if possible but reverts to 2D position if necessary (**auto 2D/3D**). The receiver can be forced to only calculate 2D (**2D only**) or 3D (**3D only**) positions.
CFG-NAVSPG-CONSTR_ALT, CFG-NAVSPG-CONSTR_ALTVAR	The fixed altitude is used if fixMode is set to 2D only. A variance greater than zero must also be supplied.
CFG-NAVSPG-INFIL_MINELEV	Minimum elevation of a satellite above the horizon in order to be used in the navigation solution. Low elevation satellites may provide degraded accuracy, due to the long signal path through the atmosphere.
CFG-NAVSPG-INFIL_NCNOTHRS, CFG-NAVSPG-INFIL_CNOTHRS	A navigation solution will only be attempted if there are at least the given number of SVs with signals at least as strong as the given threshold.

Table 5: Navigation input filter parameters

3.1.6.2 Navigation output filters

The result of a navigation solution is initially classified by the fix type (as detailed in the `fixType` field of UBX-NAV-PVT message). This distinguishes between failures to obtain a fix at all ("No Fix") and cases where a fix has been achieved, which are further subdivided into specific types of fixes (e.g. 2D, 3D, dead reckoning).

Where a fix has been achieved, a check is made to determine whether the fix should be classified as valid or not. A fix is only valid if it passes the navigation output filters as defined in CFG-NAVSPG-OUTFIL. In particular, both PDOP and accuracy values must lie below the respective limits.

☞ Important: Users are recommended to check the `gnssFixOK` flag in the UBX-NAV-PVT or the NMEA valid flag. Fixes not marked valid should not normally be used.

UBX-NAV-STATUS message also reports whether a fix is valid in the `gpsFixOK` flag. These messages have only been retained for backwards compatibility and users are recommended to use the UBX-NAV-PVT message.

CONFIDENTIAL

3.1.6.3 Degraded navigation

Degraded navigation describes all navigation modes which use less than four satellite vehicles (SV).

3.1.6.3.1 2D navigation

If the receiver only has three SVs for calculating a position, the navigation algorithm uses a constant altitude to compensate for the missing fourth SV. When a SV is lost after a successful 3D fix (min. four SVs available), the altitude is kept constant at the last known value. This is called a 2D fix.

☞ u-blox receivers do not calculate any navigation solution with less than three SVs.

3.2 Automotive dead reckoning (ADR)

3.2.1 Introduction

u-blox solutions for Automotive Dead Reckoning (ADR) allow high-accuracy positioning in places with poor or no GNSS coverage. ADR is based on Sensor Fusion Dead Reckoning (SFDR) technology, which combines GNSS measurements with those from external sensors. The ZED-F9K computes a solution type called GAWT by combining GNSS measurements with the outputs of a 3-axis accelerometer, a 3-axis gyroscope and wheel tick (sometimes called a speed tick) or speed measurements. The utilization of these sensors ensures a quick recovery of a high precision navigation solution after short GNSS signal outage (going under a bridge, signaling panels, etc...).

The firmware automatically detects and continuously calibrates the sensors.

3.2.2 Solution type

ZED-F9K produces a solution type called GAWT, which is described in the following sections.

The GAWT solution combines raw GNSS signals with data from wheel-tick sensors, accelerometers and gyroscopes to compute a fused navigation solution.

To operate the ZED-F9K in GAWT mode with optimal performance, the following tasks need to be completed:

- It is essential that the IMU-to-Antenna (CFG-SFIMU-IMU2ANT keys) and the IMU-to-VRP (CFG-SFODO-IMU2VRP keys) lever-arms are accurately configured (centimeter level ideally).
- The IMU misalignment angles are also essential. It is recommended to enable automatic alignment with the CFG-SFIMU-AUTO_MNTALG_ENA key. The misalignment angles can also be configured with CFG-SFIMU-IMU_MNTALG keys.
- It is mandatory to perform an initial calibration drive after flashing software, a cold start, or changing sensor configuration keys (CFG-SFCORE-*, CFG-SFIMU-* or CFG-SFODO-*). The calibration drive allows the software to detect and calibrate the sensors. See Accelerated Initialization and Calibration Procedure section for additional information. Performance is likely to be sub-optimal if the calibration drive is not performed correctly.
- If the maximum counter value of a wheel-tick sensor cannot be represented as a 2^N value, it must be configured manually. See Odometer Types section for additional information.
- It is strongly recommended that RTCM stream is available during the initial calibration drive, so that the resulting calibration parameters can be estimated more accurately.

3.2.3 Installation configuration

This section describes 3 reference points used in ZED-F9K ADR implementation.

3.2.3.1 Vehicle reference point (VRP)

The vehicle reference point (VRP) is defined as the center of the vehicle's rear axle. The VRP is the reference point at which the single tick or speed, must be measured.

CONFIDENTIAL

The VRP can be configured with CFG-SFODO-IMU2VRP configuration keys. See the Interface Description for more information.

3.2.3.2 IMU reference point (IRP)

The IRP is defined as the center of the IMU chip within the module. The IRP is the reference point where the 3-dimensional movement is measured.

By default, the ADR navigation engine will calculate the navigation solution at the IRP.

Figure 2: ZED-F9K IRP (measurements to center of the IMU chip within the module)

☞ Z-axis is pointing upwards from the top of the module.

3.2.3.3 Configurable reference point (CRP)

The configurable reference point (CRP) can be configured by the user, such that the navigation engine will calculate the navigation solution at the configured CRP.

The CRP can be configured with CFG-SFCORE-IMU2CRP configuration keys. See the Interface Description for more information.

3.2.3.4 IMU-frame

The IMU-frame corresponds to the IMU instrumental frame within which the accelerometers and gyroscopes are assumed to be orthogonally mounted. The IMU-frame is defined as follows:

- The origin (O in the following pictures) is the IRP. By convention, an IMU contains three orthogonally-mounted accelerometers and three orthogonally-mounted gyroscopes which have the same orientation.
- The x-axis corresponds to the x-axis accelerometer/gyroscope sensing axis;
- The y-axis corresponds to the y-axis accelerometer/gyroscope sensing axis;
- The z-axis corresponds to the z-axis accelerometer/gyroscope sensing axis.

CONFIDENTIAL

3.2.3.5 Installation-frame

The installation-frame is a right-handed 3D Cartesian frame rigidly connected with the vehicle and is used to define the installation setup required for ADR navigation. The installation frame is defined as follows:

- The origin (O in the following pictures) is the IRP;
- The x-axis points towards the front of the vehicle;
- the y-axis points towards the left of the vehicle;
- the z-axis completes the right-handed reference system by pointing up.

3.2.3.6 IMU and GNSS antenna placement

It is important to correctly position the ZED-F9K and the GNSS antenna to achieve a consistent navigation solution. In particular:

- The ZED-F9K module and the GNSS antenna can be placed in any position on the vehicle. To achieve optimal performance, the ADR solution must be correctly configured. See Solution type section for more information.
- For articulated vehicles (e.g. truck and trailer), the ZED-F9K should be placed on the front section of the vehicle.
- In order to achieve good positioning performance, the GNSS antenna should be positioned so that the mean CN0 under good signal conditions is above 35 dBHz. If this is not possible then degraded performance may be observed.

3.2.3.7 IMU-to-antenna lever-arm

The IMU-to-Antenna lever-arm is defined as the offset from the ZED-F9K IRP to the GNSS antenna center.

If the GNSS antenna is placed at a significant distance from the ZED-F9K module, then errors can be introduced in the navigation solution (particularly when the vehicle experiences high heading rate). These errors can be compensated for by configuring the IMU-to-Antenna lever-arm offset in the installation-frame.

The IMU-to-Antenna Lever-arm components can be configured via CFG-SFIMU-IMU2ANT configuration keys. See the Interface Description for more information.

CONFIDENTIAL

Figure 3: Leverarms Configuration

3.2.3.8 IMU-to-VRP lever-arm

The IMU-to-VRP lever-arm is defined as the offset from the ZED-F9K IRP to the VRP in the installation frame.

If the ZED-F9K module is placed at a significant distance from the VRP, then errors can be introduced in the navigation solution (particularly when the vehicle experiences high heading rate). These errors can be compensated for by configuring the IMU-to-VRP lever-arm offset in the installation-frame.

The IMU-to-VRP Lever-arm components can be configured via CFG-SFODO-IMU2VRP configuration keys. See the Interface Description for more information.

3.2.3.9 User-defined lever-arm

The lever arm keys CFG-SFCORE-IMU2CRP, CFG-SFODO-IMU2VRP and CFG-SFIMU-IMU2ANT must be interpreted in the following way:

- **LA_X**: Corresponds to the x-axis component of the lever-arm.
- **LA_Y**: Corresponds to the y-axis component of the lever-arm.
- **LA_Z**: Corresponds to the z-axis component of the lever-arm.

☞ In order to prevent degradation of the positioning solution the lever-arms must be configured with a resolution of at least 0.1 [m].

3.2.3.10 IMU-mount alignment

This section describes how IMU-mount misalignment angles, i.e. the angles which rotate the installation-frame to the IMU-frame, can be configured.

The IMU-mount misalignment angles are defined as follows:

- The transformation from the installation-frame to the IMU-frame is described by three Euler angles about the installation-frame axes denoted as *IMU-mount roll*, *IMU-mount pitch* and *IMU-mount yaw* angles. All three angles are referred as the *IMU-mount misalignment angles*.

The default assumption is that the IMU-frame and the installation-frame have the same orientation (i.e. all axes are parallel). If the IMU-mount misalignment angles are slightly incorrect (typically a few degrees) the navigation solution can be degraded. If there are large (tens of degrees) IMU-mount misalignments the position calculation may fail. Therefore, it is essential to correctly configure the IMU-mount misalignment settings.

☞ It is strongly recommended to use the automatic IMU-mount alignment as described in the following section.

3.2.3.10.1 Automatic IMU-mount alignment

The automatic IMU-mount alignment engine estimates automatically the IMU-mount roll, pitch and yaw angles. It requires an initialization phase during which no INS/GNSS fusion can be achieved (see Filter Modes section for further details). The progress of the automatic alignment initialization can be monitored with the UBX-ESF-STATUS message, and/or with the UBX-ESF-ALG message providing more details. When the vehicle is subject to sufficient dynamics (i.e. left and right turns during a normal drive), the automatic IMU-mount alignment engine will estimate the IMU-mount misalignment angles. Once the automatic IMU-mount alignment engine has sufficient confidence in the estimated angles, the IMU-mount misalignment angles initialization phase is completed. The raw accelerometer and gyroscope data (i.e. the IMU observations) are then compensated for IMU-mount misalignment and sensor fusion can begin. The resulting IMU-mount misalignment angles are output in the UBX-ESF-ALG message.

3.2.3.10.1.1 Enabling/disabling automatic IMU-mount alignment

The user can activate/deactivate the automatic IMU-mount alignment with the CFG-SFIMU-AUTO_MNTALG_ENA configuration key.

☞ If automatic IMU-mount alignment is deactivated while aligning, the estimated misalignment angles that were available at deactivation time are used (only if they were initialized, see next section). If automatic IMU-mount alignment is re-activated, alignment is pursued by starting from the state where deactivation happened.

3.2.3.10.2 User-defined IMU-mount alignment

It is possible to configure the IMU-mount alignment angles using the CFG-SFIMU-IMU_MNTALG configuration keys. The values that should be set in the configuration message are the Euler angles required to rotate the installation-frame to the IMU-frame. The IMU-mount yaw rotation should be performed first, then the IMU-mount pitch and finally the IMU-mount roll. At each stage, the rotation is around the appropriate axis of the transformed installation-frame, meaning that the order of the rotation sequence is important (see figure below).

CONFIDENTIAL

Figure 4: EULER-ANGLES

If there is only a single IMU-mount misalignment angle then it may be measured as shown in the three examples below.

CONFIDENTIAL

Figure 5: Installation Frame

☞ In order to prevent significant degradation of the positioning solution the IMU-mount misalignment angles should be configured with an accuracy of less than 5 degrees.

The following list describes in details how the CFG-SFIMU-IMU_MNTALG keys are to be interpreted with respect to the example illustrated in the figure above:

- **CFG-SFIMU-IMU_MNTALG_YAW**: The IMU-mount yaw angle (`yaw`) corresponds to the rotation around the installation-frame z-axis (vertical) required for aligning the installation-frame to the IMU-frame (`yaw` = 344.0 degrees if the IMU-mount misalignment is composed of a single rotation around the installation-frame z-axis, i.e. with no IMU-mount roll and IMU-mount pitch rotation).
- **CFG-SFIMU-IMU_MNTALG_PITCH**: The IMU-mount pitch angle (`pitch`) corresponds to the rotation around the installation-frame y-axis required for aligning the installation-frame to the IMU-frame (`pitch` = 26.5 degrees if the IMU-mount alignment is composed of a single rotation around the installation-frame y-axis, i.e. with no IMU-mount roll and IMU-mount yaw rotation).
- **CFG-SFIMU-IMU_MNTALG_ROLL**: The IMU-mount roll angle (`roll`) corresponds to the rotation around the installation-frame x-axis required for aligning the installation-frame to the IMU-frame (`roll` = -23.5 degrees if the IMU-mount misalignment is composed of a single rotation around installation-frame x-axis, i.e. with no IMU-mount pitch and IMU-mount yaw rotation).

3.2.4 Sensor configuration

This section describes the external sensor configuration parameters.

3.2.4.1 Odometer configuration

The odometer is a generic term for the wheel-tick or speed sensor.

The odometer can be configured with the CFG-SFODO-* configuration keys.

☞ The ZED-F9K has not been designed to work without an odometer input. Although the ZED-F9K will calculate a position without an odometer input, the accuracy of that position will be compromised.

3.2.4.1.1 Odometer interfaces

Odometer data can be delivered to ZED-F9K via the following interfaces:

- **Hardware interface**: ZED-F9K has a dedicated pin (WT) for analog wheel-tick signal input, and another pin (DIR) dedicated to the wheel-tick direction signal.
- The WT pin is enabled with the `CFG-SFODO-USE_WT_PIN` key.
- The DIR pin polarity is automatically detected by the receiver by default. To manually configure the polarity, automatic detection must be turned-off by setting the `CFG-SFODO-DIS_AUTODIRPINPOL` key and the polarity must be defined in the `CFG-SFODO-DIR_PINPOL` key.

Double edge counting can be enabled via the `CFG-SFODO-CNT_BOTH_EDGES` key. It can increase performance with low resolution wheel ticks, but is not suitable for all types of wheel tick signals. It **must not** be used with signals that are not generated with approximately 50% duty signal as it would impair performance.

- **Software interface**: Odometer data can be delivered to the receiver over one of the communication interfaces. The data shall be contained in UBX-ESF-MEAS messages. UBX-ESF-MEAS (data type 10) shall be used for single tick odometer data and UBX-ESF-MEAS (data type 11) shall be used for speed odometer data. See the Interface Description for more information.
- By default, the receiver automatically ignores the WT pin if wheel-tick/speed data are detected on the software interface. Therefore data coming from the software interface will be prioritized over data coming from the hardware interface. To disable the automatic use of data detected on the software interface, the `CFG-SFODO-DIS_AUTOSW` key must be set.

3.2.4.1.2 Odometer types

ZED-F9K supports sensors delivering the following types of data:

- **Relative wheel-tick data**: If the wheel-tick sensor delivers relative wheel-tick counts (i.e. wheel-tick count since the previous measurement), the `CFG-SFODO-COUNT_MAX` value must be set to `0`.
- **Absolute wheel-tick data**: If the wheel-tick sensor delivers absolute wheel-tick counts (i.e. wheel-tick count since startup at time tag 0) that always increase, regardless of driving forward or backward (with driving direction indicated separately). By default, the maximum absolute wheel-tick counter value is automatically estimated by the receiver for a maximum counter value that can be represented as a 2^N value. Other maximum counter values must be manually configured. For example, a `CFG-SFODO-COUNT_MAX`=1024 roll-over value would be automatically estimated, but a `CFG-SFODO-COUNT_MAX`=1000 must be configured. The maximum counter value is configured by setting the CFG-SFODO-DIS_AUTOCOUNTMAX key and setting the `CFG-SFODO-COUNT_MAX` value to the upper threshold of the absolute wheel-tick sensor count before starting again from zero (roll-over).

CONFIDENTIAL

- **Speed data**: Data coming from this sensor type can only be delivered to the receiver via one of the communication ports within a UBX-ESF-MEAS (data type 11). The speed data shall be delivered in meters per second.

If speed data but no absolute or relative wheel-tick data are detected, the receiver automatically uses the speed data without the need of reconfiguring the `CFG-SFODO-USE_SPEED` key. This behavior can be deactivated by setting the `CFG-SFODO-DIS_AUTOSPEED` key and by manually setting or clearing the `CFG-SFODO-USE_SPEED` key. If wheel-tick data (or both wheel-tick and speed data) are detected on the software interface, the receiver uses the data type (by default wheel-tick data) corresponding to the configured `CFG-SFODO-USE_SPEED` key.

To make the receiver interpret incoming speed data (data type 11 in ESF-MEAS) instead of the single wheel-tick data (data type 10 in ESF-MEAS) on the software interface, the `CFG-SFODO-USE_SPEED` key must be set.

☞ It is strongly recommended to use absolute wheel-tick sensors in order to ensure robust measurement processing even after sensor failures or outages.

3.2.4.1.3 Odometer settings

The following odometer settings can be configured:

- **Sampling Frequency**: The wheel-tick/speed data sampling frequency (`CFG-SFODO-FREQUENCY`) should be provided with an accuracy of approximately 10%. If not provided, it is automatically determined during initialization phase: this requires a consistent data rate and can take several minutes. Once initialized, the sampling frequency will be stored in non-volatile storage. For optimal navigation performance, the standard wheel-tick/speed input at 10 [Hz] is recommended.
- **Latency**: For best positioning performance, the latency of the wheel-tick/speed data (`CFG-SFODO-LATENCY`) should be given as accurately as possible (to within at least 10 ms). If not provided, the wheel-tick/speed data latency is assumed zero. More details about latency can be found in the Sensor Time Tagging section.
- **Quantization error**: If absolute/relative wheel-tick data are used (e.g. if the tick data is a distance), the quantization error can be defined in the `CFG-SFODO-QUANT_ERROR` key. The quantization error can be calculated as `2*Pi*R / T` with `R` the wheel radius, `T` the number of ticks per wheel rotation. If the quantization error is not provided, it is automatically initialized by the receiver.
- **Speed data accuracy (software interface only)**: If speed data are used, the speed data accuracy can be set in the `CFG-SFODO-QUANT_ERROR` key. If not provided, the speed data accuracy is automatically initialized by the receiver.
- **Scale factor**: If the coarse WT scale factor is not configured in the `CFG-SFODO-FACTOR` key, it is estimated automatically during initialization (see Initialization Mode section for more details).
- **Combination of multiple rear wheel-ticks (software interface only)**: If wheel-ticks are being received from both rear wheels, the receiver can be configured with the `CFG-SFODO-COMBINE_TICKS` key to use the combined rear wheel-ticks rather than a single-tick. It is recommended to use combined rear wheel-ticks if available, as they are often of higher quality than the single-ticks.

3.2.4.2 UBX-ESF-MEAS time tagging (software interface only)

In order to achieve optimal performance with the fusion solution it is essential to determine the epoch in the receiver time frame when the UBX-ESF-MEAS sensor data were taken. This may be done in one of the following ways:

- First Byte Reception: reception time of first byte of the UBX-ESF-MEAS message
- Time Mark on External Input: reception time of time mark signal sent to external input

CONFIDENTIAL

The latency of sensor data is the absolute difference between the time at which the sensor measurement is taken and the time at which either the first byte of the UBX-ESF-MEAS message or the pre-processor`s time mark are detected at the receiver.

The latency for the wheel-tick can be set with the `CFG-SFODO-LATENCY` key.

☞ It is essential that the time tags used for all UBX-ESF-MEAS data have the same resolution.

ZED-F9K automatically generates UBX-ESF-MEAS messages containing measurements from the internal IMU using the default time tag resolution of 1 millisecond. If the customer wants to input odometer data with UBX-ESF-MEAS messages, it is essential that the odometer time tag has a resolution of 1 millisecond.

3.2.4.2.1 First byte reception

The easiest way to determine the sensor measurement generation time is to have the GNSS receiver assume the time of reception of the first byte of the UBX-ESF-MEAS message (minus the constant configured latency in CFG-SFODO-LATENCY) to be the time of sensor measurement. This approach is the simplest to implement, but Time Mark on External Input can yield better latency control and compensation.

Figure 6: First Byte Reception

3.2.4.2.2 Time mark on external input

In this case, the preprocessor unit generating the measurements sends a signal to the EXTINT input of the GNSS receiver, marking the moment of measurement generation. The subsequent UBX-ESF-MEAS message is then flagged accordingly, and the measurements in the message will be assumed to have been generated at the time of external signal reception (minus the constant configured latency in CFG-SFODO-LATENCY). This approach is the preferred solution, but it can be difficult to realize an exact analog time signal for the preprocessor unit.

Figure 7: Time Mark on External Input

3.2.5 ADR system configuration

3.2.5.1 Enabling/disabling fusion filter

The ADR feature can be enabled and disabled with the `CFG-SFCORE-USE_SF` key. If ADR is disabled, the receiver outputs a GNSS-only solution. IMU sensor measurements are still available in UBX-ESF-MEAS and UBX-ESF-RAW messages.

3.2.5.2 Recommended configuration

In general, it's recommended to use the default configuration values in order to achieve optimal ADR navigation performance.

By default, the navigation solution update rate is 1 Hz. It can be configured with the CFG-RATE-NAV key.

☞ At higher navigation rates, it is strongly recommended to check (and maybe reduce) the number of enabled output messages. CPU load, memory and interface bandwidth constraints may be a limiting factor.

3.2.6 Operation

This section describes how the ADR receiver operates.

3.2.6.1 Fusion filter modes

The fusion filter operates in different modes which are output in the UBX-ESF-STATUS message.

The table below summarizes the different fusion filter modes with the associated tasks the receiver is doing.

Mode	Performed Tasks / Possible Causes	Published Fix Type
Initialization	Initialization of IMU Initialization of IMU-mount alignment Initialization of INS (position, velocity, attitude) IMU sensor error (e.g. missing data) detected	3D-Fix (GNSS)
Fusion	Fine-calibration of IMU-mount misalignment angles Fine-calibration of IMU sensors Fine-calibration of wheel-tick factors UDR fallback mode when missing WT data detected	GNSS/DR Fix
Suspended Fusion	Sensor error (e.g. missing data) detected Ferry detected	3D-Fix (GNSS)
Disabled Fusion	Fatal fusion filter error occurred Fusion filter turned-off by user	3D-Fix (GNSS)

Table 6: Fusion Modes

More details about each fusion mode are given in the following sections.

3.2.6.1.1 Initialization mode

The purpose of the initialization phase is to estimate all unknown parameters which are required for achieving fusion. The initialization phase is triggered after a receiver coldstart or a filter reset in case of fusion failure. The receiver is in initialization mode if the `fusionMode` field in the UBX-ESF-STATUS message is `0:INITIALIZING`. In this case the required sensor calibration status (`calibStatus`) are flagged as `0: NOT CALIBRATED` and the navigation solution output during initialization is based on GNSS solely.

CONFIDENTIAL

The initialization phase comprises the following internal steps whose status is published in the `initStatus` field of the UBX-ESF-STATUS message:

- **IMU initialization:** Unknown crucial IMU parameters such as sensor sampling frequency are estimated during initialization. As long as all required IMU parameters are not initialized, the status of the IMU initialization (`imuInitStatus`) is flagged as `1:INITIALIZING` in the UBX-ESF-STATUS message. Moreover, the required sensor calibration statuses (`calibStatus`) are flagged as `0:NOT CALIBRATED` in the UBX-ESF-STATUS message. Note that if the user configured all required sensor settings, this step is skipped and IMU initialization is flagged as `2:INITIALIZED`.
- **IMU-mount alignment initialization:** If automatic IMU-mount alignment is enabled (see the Automatic IMU-mount Alignment Configuration section), initial IMU-mount roll, pitch and yaw angles need to be estimated. For that, good GNSS signal reception as well as sufficient vehicle dynamics (i.e. a series of left and right turns during a normal drive) need to be at hand. As long as the IMU-mount alignment is not initialized, the status of the IMU-mount alignment (`mntAlgStatus`) is flagged as `1:INITIALIZING` in the UBX-ESF-STATUS message. Once initialized, the IMU-mount alignment status is flagged as `2:INITIALIZED`. If no IMU-mount alignment is required, the IMU-mount alignment is flagged as `0:OFF`. A detailed description of the automatic IMU-mount alignment operation can be found in the Automatic IMU-mount Alignment Operation section.
- **INS initialization:** Before entering fusion mode, the initial vehicle position, velocity and especially attitude (vehicle roll, pitch and heading angles) need to be known with sufficient accuracy. This is achieved during INS initialization phase using GNSS which comprises an INS coarse alignment step. As long as the fusion filter isn't initialized, the status of the INS initialization (`insInitStatus`) is flagged as `1:INITIALIZING` in the UBX-ESF-STATUS message. Once initialized, the INS initialization is flagged as `2:INITIALIZED`.
- **Wheel-tick sensor initialization:** Solution enters fusion mode (`fusionMode` field in the UBX-ESF-STATUS message is on `1:FUSION`), even when wheel-tick is not yet initialized, following a UDR mode approach. WT sensor parameters, like initial wheel-tick factor, are estimated in parallel and are used once estimated with sufficient accuracy. As long as the wheel-tick parameters are not initialized, the status of the wheel-tick initialization (`wtInitStatus`) is flagged as `1:INITIALIZING` in the UBX-ESF-STATUS message. Once initialized, the wheel-tick sensor initialization is flagged as `2:INITIALIZED`, WT data are used by the filter and the parameters are stored in non-volatile storage. If no wheel-tick data are required (in UDR products), the wheel-tick initialization is flagged as `0:OFF`.
- **Sensor error (e.g. missing data) detected:** Sensor timeout of more than 500ms will trigger an INS re-initialization.

Note that initialization phase requires good GNSS signal conditions as well as periods during which vehicle is stationary and moving (including left and right turns). Once all required initialization steps are achieved, fusion mode is triggered and continuous calibration begins.

3.2.6.1.2 Fusion mode

Once initialization phase is achieved, the receiver enters fusion mode. The receiver is in fusion mode if the `UBX-ESF-STATUS.fusionMode` field is set to `1:FUSION`. The fusion filter then starts to compute combined GNSS/Dead-reckoning fixes (fused solutions) and to calibrate the sensors required for computing the fused navigation solution. This is the case when the sensor calibration status (`UBX-ESF-STATUS.calibStatus`) is set to `1:CALIBRATING`. As soon as the calibration reaches a status where optimal fusion performance can be expected, (`UBX-ESF-STATUS.calibStatus` is set to `2/3:CALIBRATED`.

CONFIDENTIAL

3.2.6.1.2.1 UDR fallback mode

In case of WT sensor error / timeout (500ms), or normal WT sensor initialization, the solution falls back to UDR (Untethered Dead Reckoning) mode. The `UBX-ESF-STATUS.fusionMode` field still shows `1:FUSION` when UDR fallback mode has been entered. However, the following flags can be used to determine when the receiver has entered UDR mode:

- The `UBX-ESF-STATUS.wtInitStatus` shows `1:INITIALIZING`.
- The WT fault flag, `UBX-ESF-STATUS.missingMeas` field, is set to `1`, indicating a WT timeout has been detected.

3.2.6.1.3 Suspended fusion mode

Sensor fusion is temporarily suspended in cases where no fused solution should/can be computed. In this case, the receiver produces a GNSS-only solution. The receiver is considered to be in temporarily-disabled fusion mode when the `UBX-ESF-STATUS.fusionMode` field is set to `2:SUSPENDED`.

Fusion is suspended in the following scenarios:

- If one or several sensors deliver erroneous data or no data at all. Fusion is suspended during the sensor failure period. The receiver automatically recovers once the affected sensor(s) is/are back to normal operation.
- If the vehicle is detected to be on a ferry or other moving platform, where odometer data does not indicate any displacement.

3.2.6.1.4 Disabled fusion mode

Sensor fusion can be permanently disabled if recurrent fusion failures occur. The receiver is considered to be in permanently disabled fusion mode if the `UBX-ESF-STATUS.fusionMode` field is set to `3:DISABLED`. In this case, the receiver produces a GNSS-only solution if possible.

Fusion is permanently disabled in the following cases:

- If the fusion filter was manually turned-off by the user with the `CFG-SFCORE-USE_SF` key.
- If the filter diverges due to significantly wrong installation or filter parameters.

3.2.6.2 Accelerated initialization and calibration procedure

This section describes how to perform fast initialization and calibration of the ADR receiver for the purpose of evaluation.

The duration of the initialization phases depend on the quality of the GNSS signals and the dynamics encountered by the vehicle.

The initialization time required for reaching fused navigation mode can be shortened by following the procedure in the order described in the table below.

Phase	Procedure	Indicator of Success
IMU initialization	After receiver coldstart or first receiver use, turn-on car engine and stay stationary under good GNSS signal reception conditions for at least 3 minutes.	IMU initialization status `UBX-ESF-STATUS.imuInitStatus` shows `2:INITIALIZED`.
INS initialization (position and velocity)	Once IMU is initialized, stay stationary under good GNSS signal reception conditions until a reliable GNSS fix could be achieved.	GNSS 3D fix achieved, good 3D position accuracy (at least 5 m), high number of used SVs (check UBX-NAV-PVT message).

CONFIDENTIAL

Phase	Procedure	Indicator of Success
IMU-mount alignment initialization	Start driving with a minimum speed of 30 km/h and do a series of approximately 10 left and right turns (at least 90 degrees turns). *This step can be skipped if automatic IMU-mount alignment is turned-off.*	IMU-mount alignment status (`UBX-ESF-STATUS.mntAlgStatus`) shows `2:INITIALIZED`, the IMU-mount alignment status `UBX-ESF-ALG.status` shows `3:COARSE ALIGNED`.
Wheel-tick sensor initialization	Drive for at least 500 meters at a minimum speed of 20 km/h. To shorten this calibration step, the car should be driven at higher speed (around 50 km/h) for at least 10 seconds under good GNSS visibility.	Wheel-tick sensor initialization status `UBX-ESF-STATUS.wtInitStatus` shows `2:INITIALIZED`.
INS initialization (attitude)	Drive straight for at least 100 meters at a minimum speed of 40 km/h.	INS initialization status `UBX-ESF-STATUS.insInitStatus` shows `2:INITIALIZED`.

Table 7: Accelerated initialization procedure

Once initialization is completed, the `UBX-ESF-STATUS.fusionMode` field shows `1:FUSION`, combined GNSS/Dead-reckoning fixes (fused solutions) are output and the sensors used in the navigation filter start to get calibrated. Calibration is a continuous process running in the background and directly impacting the navigation solution quality.

The calibration time required for reaching optimal ADR navigation performance can be shortened by following the procedure described in the table below.

Phase	Procedure	Indicator of Success
IMU-mount alignment calibration	Keep driving with a minimum speed of 30 km/h and do a series of left and right turns (at least 90 degrees). At each turn the estimated IMU-mount misalignment angles are refined and their accuracy increased. *This step can be skipped if automatic IMU-mount alignment is turned-off.*	Once the IMU-mount alignment engine has high confidence in its misalignment angle estimates, the IMU-mount alignment status `UBX-ESF-ALG.status` is set to `4:FINE ALIGNED`.
IMU calibration (gyroscope and accelerometer)	Drive curves and straight segments for a few minutes by including a few stops lasting at least 30 seconds each. This drive should also include some periods with higher speed (at least 50 km/h) and can typically be carried out on normal open-sky roads with good GNSS signal reception conditions.	The calibration status of the used sensors `UBX-ESF-STATUS.calibStatus` shall show `2/3:CALIBRATED`.

Table 8: Accelerated calibration procedure

Note that the calibration status (`UBX-ESF-STATUS.calibStatus`) of some used sensors might fall back to `1:CALIBRATING` if the receiver is operated in challenging conditions. In such a case, fused navigation solution uncertainty increases until optimal conditions are observed again for re-calibrating the sensors.

☞ In the presence of significant temperature gradient affecting the gyroscopes, the fused navigation performance might also depend on how well the temperature compensation table is populated. The table is gradually filled-in while the vehicle is stationary and by observing gyroscope biases at different temperatures. Therefore the quality of the gyroscope temperature compensation depends on how many temperature bins could be observed while the vehicle was stationary and on the duration of observation for each bin.

3.2.6.3 Automatic IMU-mount alignment

3.2.6.3.1 Alignment solution output

The IMU-mount misalignment angles are shown in UBX-ESF-ALG messages.

CONFIDENTIAL

- **IMU-mount angle initialization**: During IMU-mount angle initialization (`UBX-ESF-ALG.status` field is equal to `2`), the published angles (`yaw, pich and roll`) correspond to the current estimated values but are not yet applied for rotating the IMU observations.
- **IMU-mount angle initialization complete**: After initialization (`UBX-ESF-ALG.status` field is equal or higher than `3`), the published angles correspond to the estimated value and are applied for rotating the IMU observations.
- **Automatic IMU-mount Alignment disabled**: If automatic IMU-mount alignment is disabled, the published angles correspond to the IMU-mount angles configured by the user (see User-defined IMU-mount Alignment section) and are applied for rotating the IMU observations.

CAUTION If user-defined IMU-mount misalignment angles were configured by the user using CFG-SFIMU-IMU_MNTALG keys (see User-defined Configuration section) and automatic IMU-mount alignment is active, the angles output in the UBX-ESF-ALG message still correspond to the definition given above: they represent the full rotation required for transforming IMU data from installation-frame to IMU-frame. This means that the output misalignment angles are computed from the composed rotation of the user-defined rotation and the internally-estimated rotation.

3.2.6.3.2 Alignment progress

The progress of the automatic IMU-mount alignment can be monitored by checking the `UBX-ESF-ALG.status` field.

- **IMU-mount roll/pitch angle initialization ongoing**: The alignment engine is initializing the IMU-mount roll and pitch angles (`UBX-ESF-ALG.status` is 1). Both angles can only be initialized if vehicle encounters left and right turns (as occurring during a normal drive).
- **IMU-mount yaw angle initialization ongoing**: The alignment engine is initializing the IMU-mount yaw angle (`UBX-ESF-ALG.status` is 2). IMU-mount yaw angle can only be initialized once IMU-mount roll and pitch angles are initialized and if vehicle encounters left and right turns (as occurring during a normal drive).
- **IMU-mount alignment coarse calibration ongoing**: Once initialized (`UBX-ESF-ALG.status` is 3), the automatic IMU-mount alignment engine has sufficient confidence in all IMU-mount misalignment angles and validates their use for compensating the accelerometer and gyroscope data (fused navigation solutions can be computed). The IMU-mount misalignment angles are filtered each time the observed vehicle dynamics are measurable.
- **IMU-mount alignment fine calibration ongoing**: Once the IMU-mount misalignment angles are estimated with a good accuracy, the automatic IMU-mount alignment engine becomes more conservative in updating the IMU-mount misalignment angles (`UBX-ESF-ALG.status` is 4).

3.2.6.3.3 Alignment reset

If for some reasons the IMU-mount misalignment angles estimated by the automatic IMU-mount alignment engine are no longer valid (for example due to a change in the physical mounting of the device), the IMU-mount misalignment angles can be reset by sending an UBX-ESF-RESETALG message. In addition, the misalignment angles are also reset in the following cases:

- If a cold-start command is sent.
- If the user-defined IMU-mount misalignment angles are changed by sending CFG-SFIMU-IMU_MNTALG keys (see User-defined Alignment section for more details).

The IMU-mount alignment engine then falls back into initialization mode.

3.2.6.3.4 Alignment errors

The following errors might be output in the `UBX-ESF-ALG.error` bit field:

- **IMU-mount roll/pitch angle error**: If the automatic IMU-mount alignment engine suspects wrong IMU-mount roll and/or IMU-mount pitch misalignment angles (either due to a wrong

CONFIDENTIAL

initialization or a change in the physical mounting of the device), the `UBX-ESF-ALG.error` bit 0 is set to 1.

- **IMU-mount yaw angle error**: If the automatic IMU-mount alignment engine suspects wrong IMU-mount yaw misalignment angle (either due to a wrong initialization or a change in the physical mounting of the device), the `UBX-ESF-ALG.error` bit 1 is set to 1.
- **Euler Angle singularity ('gimbal-lock') error**: The Euler angle singularity `UBX-ESF-ALG.error` bit 2 is set when the automatic IMU-mount alignment engine detects an installation where the IMU-frame is misaligned in such a way that a degree of freedom is lost when two IMU-mount misalignment (Euler) angles begin to describe the same rotations (or axes). This happens for example with an IMU-mount misalignment of +/- 90 degrees around the IMU-mount pitch axis, where IMU-mount roll and IMU-mount yaw cannot be distinguished from each other. In such a case, these IMU-mount misalignment angles start to heavily fluctuate with time due to the mathematical singularity occurring at these points, meaning that the IMU-mount misalignment angles output in the UBX-ESF-ALG are not stable in time. Note however that each individual set of IMU-mount misalignment angles output in such a case still describes the correct rotation. Moreover, the internal rotation applied for aligning the IMU readings doesn't suffer from this singularity issue and optimal fusion can still be achieved.

3.2.6.4 Navigation output

3.2.6.4.1 Local-level north-east-down (NED) frame

The local-level frame is a geodetic frame with following features:

- The origin (O) is a point on the Earth's surface;
- The x-axis points to North;
- the y-axis points to East;
- the z-axis completes the right-handed reference system by pointing down.

The frame is referred to as North-East-Down (NED) since its axes are aligned with the North, East and Down directions.

3.2.6.4.2 Vehicle-frame

The vehicle-frame is a right-handed 3D Cartesian frame rigidly connected with the vehicle and is used to determine the attitude of the vehicle with respect to the local-level frame. It has the following features:

- The origin (O) is the IRP;
- The x-axis points towards the front of the vehicle;
- the y-axis points towards the right of the vehicle;
- the z-axis completes the right-handed reference system by pointing down.

3.2.6.4.3 Vehicle position and velocity output

The position and velocity information is output in several messages like UBX-NAV-PVT for example. Position and velocity computed by the ADR navigation filter are referenced to the IRP.

3.2.6.4.4 Vehicle position and velocity output in CRP

When setting the IMU-to-CRP lever-arm, position and velocity computed by the ADR navigation filter are referenced to the CRP.

3.2.6.4.5 Vehicle attitude output

The transformation between the vehicle-frame and the local-level frame is described by three attitude angles about the local-level axes denoted as *vehicle roll*, *vehicle pitch* and *vehicle heading*. All three angles are referred as *vehicle attitude* and are illustrated in the figure below:

CONFIDENTIAL

Figure 8: Vehicle Attitude Output

The order of the sequence of rotations around the navigation axes defining the vehicle attitude matrix in terms of vehicle attitude angles is illustrated below:

VEHICLE ATTITUDE DEFINITION

ϕ : Vehicle roll angle

θ : Vehicle pitch angle

ψ : Vehicle heading angle

$\mathbf{C}_b^n$: Rotation between body-frame (*b*) and local-level NED navigation-frame (*n*)

$$\mathbf{C}_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \quad \mathbf{C}_Y = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad \mathbf{C}_Z = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{C}_b^n = \mathbf{C}_Z^T \cdot \mathbf{C}_Y^T \cdot \mathbf{C}_X^T$$
$$= \begin{bmatrix} \cos(\theta)\cos(\psi) & \sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi) & \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ \cos(\theta)\sin(\psi) & \sin(\phi)\sin(\theta)\sin(\psi) + \cos(\phi)\cos(\psi) & \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix}$$

Figure 9: Vehicle Attitude Definition

Note that in this picture the body-frame corresponds to the vehicle-frame.

The vehicle attitude is output in the UBX-NAV-ATT message. The message provides all three angles together with their accuracy estimates.

3.2.6.4.6 Vehicle dynamics output

The UBX-ESF-INS message outputs information about vehicle dynamics provided by the INS: compensated vehicle angular rates and compensated vehicle accelerations. The acceleration data is free of any gravitational acceleration. Its accuracy is directly dependent on the filter attitude estimation accuracy.

Compensated vehicle dynamics information is output with respect to the vehicle-frame.

☞ The message outputs only dynamics information that is directly compensated by the fusion filter. This implies that depending on the solution type and the sensor availability, dynamics along some axes of the vehicle-frame might not be available.

3.2.6.5 Sensor data types

The UBX-ESF-MEAS messages can be used as input and/or output messages. They can provide the following functionality:

- Output the ZED-F9K internal IMU measurements;
- Input external wheel tick or speed measurements to ZED-F9K from a host;

A different number of data fields may be used, and these can contain different types of measurements. The type of each measurement is specified in the `UBX-ESF-MEAS.dataType` field.

The supported sensor data types are:

Type	Description	Unit	Format of the 24 data bits
0	none, data field contains no data		
1..4	reserved		
5	z-axis gyroscope angular rate	deg/s *2^-12	signed
6..7	reserved		

CONFIDENTIAL

Type	Description	Unit	Format of the 24 data bits
8	rear-left wheel ticks		Bits 0-22: unsigned tick value. Bit 23: direction indicator (0=forward, 1=backward)
9	rear-right wheel ticks		Bits 0-22: unsigned tick value. Bit 23: direction indicator (0=forward, 1=backward)
10	single tick (speed tick)		Bits 0-22: unsigned tick value. Bit 23: direction indicator (0=forward, 1=backward)
11	speed	m/s * 1e-3	signed
12	gyroscope temperature	deg Celsius * 1e-2	signed
13	y-axis gyroscope angular rate	deg/s *2^-12	signed
14	x-axis gyroscope angular rate	deg/s *2^-12	signed
15	reserved		
16	x-axis accelerometer specific force	m/s^2 *2^-10	signed
17	y-axis accelerometer specific force	m/s^2 *2^-10	signed
18	z-axis accelerometer specific force	m/s^2 *2^-10	signed
19..63	reserved		

Table 9: Definition of data types

3.2.6.6 Raw sensor data output

The UBX-ESF-RAW message can be used to access raw sensor measurements. A variable number of data fields may be used in a single message and these can contain different types of measurements. The type of each measurement is specified in the `UBX-ESF-RAW.dataType` field.

The possible data types are the same as described in the Sensor Data Types section. One UBX-ESF-RAW message can contain multiple samples from the same sensor. The user can separate and order these using the time tags attached to each of the measurements.

The measurements are made at a fixed rate. The sampling rate or other sensor configuration options cannot be changed.

To enable this feature the UBX-ESF-RAW message must be enabled using CFG-MSGOUT-UBX_ESF_RAW keys. If any non-zero rate is selected the message will be output at the rate at which the sensors are sampled.

☞ Turning on this feature does not disable sensor fusion in the receiver.

3.2.6.7 Receiver startup and shutdown

Continuous dead reckoning is possible over receiver restarts if the following conditions are true:

- Non-volatile storage is available
- A real-time clock is available or time assistance is provided on startup
- The sensor data stream is only stopped when the vehicle is stationary
- The vehicle is not moved while the receiver is off

3.2.7 High navigation rate (HNR)

3.2.7.1 Introduction

ZED-F9K provides a low latency position, velocity and vehicle attitude to be output between GNSS epochs. Sensor measurements are used to propagate the solution at the higher rate between GNSS epochs.

The high navigation rate solution is output using the UBX-HNR-PVT and UBX-HNR-ATT messages.

CONFIDENTIAL

3.2.7.2 Configuration

The high navigation rate output can be configured using the CFG-SFCORE-HNR_RATE configuration key.

☞ If a high navigation rate has been configured with CFG-SFCORE-HNR_RATE, then it is strongly recommended to check (and maybe reduce) the number of enabled output messages. CPU load, memory and interface bandwidth constraints may be a limiting factor.

3.3 Interfaces

ZED-F9K provides UART1, SPI, DDC (I^2C compatible) and USB interfaces for communication with a host CPU. The interfaces are configured via the configuration interface which is described in the ZED-F9K Interface Description [2].

☞ It is important to isolate interface pins when VCC is removed. They can be allowed to float or connected to a high impedance.

Some example isolation circuits are shown below.

Figure 10: ZED-F9K output isolation

Figure 11: ZED-F9K input isolation

CONFIDENTIAL

Figure 12: ZED-F9K interface level translation

3.3.1 UART interfaces

ZED-F9K includes 2 UART ports.

UART1 can be used for host interface. It supports a configurable baud rate and protocol selection.

UART2 is available as an optional stand-alone RTCM interface. It should not be used as a host interface.

☞ The default baud rate is 38400 baud. To prevent buffering problems it is recommended not to run at a lower baud rate than the default.

3.3.2 SPI interface

The ZED-F9K high precision receiver has an SPI slave interface that can be selected by setting D_SEL = 0. The SPI slave interface is shared with UART1. The SPI pins available are: SPI_MISO (TXD), SPI_MOSI (RXD), SPI_CS_N, SPI_CLK. The SPI interface is designed to allow communication to a host CPU. The interface can be operated in slave mode only. The maximum transfer rate using SPI is 125 kB/s and the maximum SPI clock frequency is 5.5 MHz.

3.3.3 USB interface

The USB interface is compatible with a USB version 2.0 FS (full speed, 12 Mb/s) interface.

☞ USB suspend mode is not supported.

☞ USB bus powered mode is not supported.

CONFIDENTIAL

☞ It is important to connect V_USB to ground when the USB interface is not used in an application.

There are additional hardware requirements if USB is designed in to be used:

In *self powered mode* the receiver is powered by its own power supply. V_USB is used to detect the availability of the USB port, i.e. whether the receiver is connected to a USB host.

- Pin 38, V_USB needs to be powered by a separate LDO enabled by module VCC and supplied by the USB host.
- A pull down resistor is required on the output of this V_USB LDO
- V_USB (pin 38) requires 1 µF and 100 nF capacitors mounted adjacent to the pin to ensure correct V_USB voltage detection
- Apply USB_DM and USB_DP series resistors; typically 27 Ω

Figure 13: ZED-F9K V_USB supply

R11 = 100 k Ω is recommended

R4, R5 = 27 Ω is recommended

3.3.4 D_SEL interface

The D_SEL pin can be used to configure the functionality of pins 42 to 45. It is possible to configure the pins as UART1 + I^2C, or as SPI. See Table 10 below.

Pin No	D_SEL == 0	D_SEL == 1
42	SPI_MISO	UART1 TXD
43	SPI_MOSI	UART1 RXD
44	SPI_CS_N	DDC/I^2C SDA
45	SPI_CLK	DDC/I^2C SCL

Table 10: D_SEL configuration

3.3.5 RESET_N interface

The ZED-F9K high precision receiver provides the ability to reset the receiver. The RESET_N pin is an input-only pin with an internal pull-up resistor. Driving RESET_N low for at least 100 ms will trigger a cold start.

☞ The RESET_N pin will trigger a cold start and therefore should only be used as a recovery option and not a Power On Reset.

3.3.6 SAFEBOOT_N interface

The ZED-F9K high precision receiver provides a SAFEBOOT_N pin that is used to command the receiver into safe boot mode.

If this pin is low at power up, the receiver starts in safe boot mode and GNSS operation is disabled.

CONFIDENTIAL

It can be used to recover from situations where the Flash has become corrupted and needs to be restored.

In safe boot mode the receiver runs from a passive oscillator circuit with less accurate timing and hence the receiver is unable to communicate via USB.

In this mode only UART1 and DDC communication is possible. For communication via UART1 in safe boot mode, a training sequence (0x 55 55 at 9600 baud) must be sent by the host to the receiver in order to begin communication. After this the host must wait at least 2 ms before sending any data.

Safe boot mode is used in production to program the Flash and to set the low level configuration in the eFuse.

It is recommended to have the possibility to pull the SAFEBOOT_N pin low in the application. This can be provided using an externally connected test point or a host I/O port.

3.3.7 TIMEPULSE interface

The ZED-F9K high precision receiver provides a time pulse on the TIMEPULSE pin.

3.3.8 TX_READY interface

The TX_READY function is used to indicate when the receiver has data to transmit. A listener can wait on the TX_READY signal instead of polling the DDC or SPI interfaces. The CFG-TXREADY message lets you configure the polarity and the number of bytes in the buffer before the TX_READY signal goes active. The TX_READY function is disabled by default.

3.3.9 Display data channel (DDC)

An I^2C compliant DDC interface is available for communication with an external host CPU or u-blox cellular modules. The interface can be operated in slave mode only. The DDC protocol and electrical interface are fully compatible with fast-mode of the I^2C industry standard. Since the maximum SCL clock frequency is 400 kHz, the maximum transfer rate is 400 kb/s. The SCL and SDA pins have internal pull-up resistors which should be sufficient for most applications. However, depending on the speed of the host and the load on the DDC lines additional external pull-up resistors may be necessary.

☞ To use the DDC/I^2C interface DSEL pin must be left open.

3.3.10 Antenna supervisor

An active antenna supervisor provides the means to check the antenna for open and short circuits and to shut off the antenna supply if a short circuit is detected. Once enabled, the active antenna supervisor produces status messages, reporting in NMEA and/or UBX protocol.

The antenna supervisor can be configured through the CFG-HW-ANT_* configuration items. The current configuration of the active antenna supervisor can also be checked by polling the related CFG-HW_ANT_* configuration items.

The current active antenna status can be determined by polling the UBX-MON-RF message. If an antenna is connected, the initial state after power-up is “Active Antenna OK" in the u-center UBX-MON-RF view.

An active antenna supervisor circuit is connected to the ANT_DET, ANT_OFF, ANT_SHORT_N pins. For an example the open circuit detection circuit using ANT_DET, "high" = Antenna detected (antenna consumes current); "low" = Antenna not detected (no current drawn).

CONFIDENTIAL

The following schematic details the required circuit and the sections following it detail how to enable and monitor each feature:

Figure 14: ZED-F9K antenna supervisor

☞ The bias-t inductor must be chosen for multi-band operation; a value of 120 nH 5% is required for our recommended Murata part if current is limited below its 110 mA rating. See antenna bias section for additional information.

☞ Circuit shows buffer [U4]. Buffer is not strictly necessary when supplied from VCC. It is only required when supplying antenna voltage that is not obtained from or controlled by module VCC or VCC_RF .

L1: 300 mA and >500 Ω at L-band frequencies

C2: MURATA GRM033R71C103KE14 CER X7R 0402 10N 10% 16V

ESD protection diode on RF trace TYCO, 0.25PF, PESD0402-140 -55/+125C

3.3.10.1 Antenna voltage control - ANT_OFF

☞ Antenna status (as reported in UBX-MON-RF and UBX-INF-NOTICE messages) is not reported unless the antenna voltage control has been enabled.

Enable the antenna voltage control by setting the configuration item CFG-HW-ANT_CFG_VOLTCTRL to true (1).

Result:

- UBX-MON-RF in u-center: Antenna status = OK. Antenna power status = ON
- ANT_OFF pin = active high to turn antenna off therefore the pin is low to enable an external antenna.

Start-up message at power up if configuration stored:

```
$GNTXT,01,01,02,ANTSUPERV=AC *00

$GNTXT,01,01,02,ANTSTATUS=INIT*3B
```

```
$GNTXT,01,01,02,ANTSTATUS=OK*25
```

`ANTSUPERV=AC` indicates antenna control is activated

3.3.10.2 Antenna short detection - ANT_SHORT_N

Enable the antenna short detection by setting the configuration item CFG-HW-ANT_CFG_SHORTDET to true (1).

Result:

- UBX-MON-RF in u-center: Antenna status = OK. Antenna power status = ON
- ANT_OFF = active high to disable an external antenna therefore the pin is low to enable an external antenna.
- ANT_SHORT_N = active low to detect a short therefore the pin is high (PIO pull up enabled to be pulled low if shorted)

Start-up message at power up if configuration is stored:

```
$GNTXT,01,01,02,ANTSUPERV=AC SD *37
```

```
$GNTXT,01,01,02,ANTSTATUS=INIT*3B
```

```
$GNTXT,01,01,02,ANTSTATUS=OK*25
```

`ANTSUPERV=AC SD` (Antenna control and short detection activated)

Then if shorted (ANT_SHORT_N pulled low):

- UBX-MON-RF in u-center: Antenna status = SHORT. Antenna power status = ON (auto power down is not enabled = off by default)

```
$GNTXT,01,01,02,ANTSTATUS=SHORT*73
```

- ANT_OFF = active high therefore still low (still enabled as auto power down is not enabled)

☞ After a detected antenna short, the reported antenna status will keep on being reported as shorted. If the antenna short detection auto recovery is enabled, then the antenna status can recover after a timeout. To recover the antenna status immediately, a power cycle is required or configuring off and on again the antenna short detection functionality.

3.3.10.3 Antenna short detection auto recovery

Enable the antenna short detection auto recovery by setting the configuration item CFG-HW-ANT_CFG_RECOVER to true (1).

Result:

- UBX-MON-RF in u-center: Antenna status = OK. Antenna power status = ON
- ANT_OFF = active high there for the PIO is low to enable an external antenna
- ANT_SHORT_N = high (PIO pull up enabled to be pulled low if shorted)

Start-up message at power up if configuration is stored:

```
$GNTXT,01,01,02,ANTSUPERV=AC SD PDoS SR*3E
```

```
$GNTXT,01,01,02,ANTSTATUS=INIT*3B
```

```
$GNTXT,01,01,02,ANTSTATUS=OK*25
```

`ANTSUPERV=AC SD PDoS SR` (indicates short circuit recovery added - SR)

Then if antenna is shorted (ANT_SHORT_N pulled low):

CONFIDENTIAL

- `$GNTXT,01,01,02,ANTSTATUS=SHORT*73`
- UBX-MON-RF in u-center: Antenna status = SHORT. Antenna power status = OFF
- ANT_OFF = high (to disable - active high)

After a time out period receiver will re-test the short condition by enabling ANT_OFF = LOW

If a short is not present it will report antenna condition is ok:

`$GNTXT,01,01,02,ANTSTATUS=OK*25`

MON-RF in u-center: Antenna status = OK. Antenna power status = ON

3.3.10.4 Antenna open circuit detection - ANT_DETECT

Enable the antenna open circuit detection by setting the configuration item CFG-HW-ANT_CFG_OPENDET to true (1).

Result:

- UBX-MON-RF in u-center: Antenna status = OK. Antenna power status = ON
- ANT_OFF = active high therefore PIO is low to enable external antenna
- ANT_SHORT_N = active low therefore PIO is high (PIO pull up enabled to be pulled low if shorted)
- ANT_DETECT = active high therefore PIO is high (PIO pull up enabled to be pulled low if antenna not detected)

Start-up message at power up if configuration is stored:

`$GNTXT,01,01,02,ANTSUPERV=AC SD OD PDoS SR*15`

`$GNTXT,01,01,02,ANTSTATUS=INIT*3B`

`$GNTXT,01,01,02,ANTSTATUS=OK*25`

`ANTSUPERV=AC SD OD PDoS SR` (indicates open circuit detection added - OD)

Then if ANT_DETECT is pulled low to indicate no antenna:

`$GNTXT,01,01,02,ANTSTATUS=OPEN*35`

Then if ANT_DETECT is left floating or it is pulled high to indicate antenna connected:

`$GNTXT,01,01,02,ANTSTATUS=OK*25`

3.3.11 EXTINT

EXTINT is an external interrupt pin with fixed input voltage thresholds with respect to VCC. It can be used for functions such as accurate external frequency aiding and ON/OFF control. Leave open if unused, this function is disabled by default.

3.3.12 WT and DIR inputs

ZED-F9K pin 22 (WT) is available as a wheel-tick input. Pin 23 (DIR) is available as a direction input (forward/reverse indication).

By default the wheel-tick count is derived from the rising edges of the WT input.

The WT input may be configured to use both the rising and falling edges of the wheel-tick signal. To use both edges, the wheel-tick pulses shall have approximately 1:1 mark:space ratio regardless of speed. The minimum recommended pulse width is 10 us.

For optimal performance the wheel-tick resolution should be less than 5 cm.

CONFIDENTIAL

The wheel-tick input shall change frequency linearly with respect to the change in vehicle speed.

When the vehicle is stationary, there shall be no wheel-tick pulses at the WT input. This is especially important at system shutdown and startup.

☞ Performance may be affected if there are wheel-ticks on the WT pin while the vehicle is stationary.

The DIR input shall indicate whether the vehicle is moving forwards or backwards.

By default, the DIR pin-polarity is automatically initialized once the vehicle has reached required minimum speed of **30 km/h**. The DIR pin polarity can also be configured with the CFG-SFODO-DIR_PINPOL key.

☞ Incorrect operation may occur if WT input is used without a matching DIR input.

Alternatively, the vehicle WT (or speed) and DIR inputs can be provided via one of the communication interfaces with UBX-ESF-MEAS messages. In this use case, the WT pin can be configured as EXTINT1 to provide a time-mark for the UBX-ESF-MEAS messages. The DIR pin can be left open.

☞ Do not exceed the maximum voltage of 3.6 V at the WT and DIR inputs.

☞ For more information, see the ZED-F9K Interface Description[2].

3.3.13 Communication ports

u-blox receivers are enabled with a highly flexible communication interface. It supports a variety of protocols, and is truly multi-port and multi-protocol capable. See Interface Description for the supported protocols. Each protocol can be enabled on several ports at the same time (multi-port capability) with individual settings (e.g. baud rate, message rates, etc.) for each port. Furthermore, more than one protocols can be enabled on a single port at the same time (multi-protocol capability).

Port #	Electrical interface
0	DDC (I^2C compatible)
1	UART1
2	UART2
3	USB
4	SPI

Table 11: Port number assignment

The following table shows the port numbers reported in the UBX-MON-COMMS message.

Port #	Electrical interface
0x0000	DDC (I^2C compatible)
0x0001	UART1
0x0102	UART2
0x0003	USB
0x0004	SPI

Table 12: Port number assignment reported in the UBX-MON-COMMS message.

3.3.13.1 UART ports

The serial ports consist of an RX and a TX line. Neither handshaking signals nor hardware flow control signals are available. These serial ports operate in asynchronous mode. The baud rates can be configured individually for each serial port. However, there is no support for setting different baud rates for reception and transmission.

CONFIDENTIAL

As of UBX protocol version 18 and beyond, the UART RX interface will be disabled when more than 100 frame errors are detected during a one-second period. This can happen if the wrong baud rate is used or the UART RX pin is grounded. An error message appears when the UART RX interface is re-enabled at the end of the one-second period.

Baud rate	Data bits	Parity	Stop bits
4800	8	none	1
9600	8	none	1
19200	8	none	1
38400	8	none	1
57600	8	none	1
115200	8	none	1
230400	8	none	1
460800	8	none	1
921600	8	none	1

Table 13: Possible UART interface configurations

Note that for protocols such as NMEA or UBX, it does not make sense to change the default word length values (data bits) since these properties are defined by the protocol and not by the electrical interface.

If the amount of data configured is too much for a certain port's bandwidth (e.g. all UBX messages output on a UART port with a baud rate of 9600), the buffer will fill up. Once the buffer space is exceeded, new messages to be sent will be dropped. To prevent message loss, the baud rate and communication speed or the number of enabled messages should be carefully selected so that the expected number of bytes can be transmitted in less than one second.

3.3.13.2 SPI port

SPI is a four-wire synchronous communication interface. In contrast to UART, the master provides the clock signal, which therefore doesn't need to be specified for the slave in advance. Moreover, a baud rate setting is not applicable for the slave.

CAUTION The SPI clock speed is limited depending on hardware and firmware versions!

3.3.13.2.1 Maximum SPI clock speed

The receiver supports a maximum SPI clock speed of 5.5 MHz.

3.3.13.2.2 Read access

As the register mode is not implemented for the SPI port, only the UBX/NMEA message stream is provided. This stream is accessed using the back-to-back read and write access (see section back-to-back read and write access). When no data is available to be written to the receiver, `MOSI` should be held logic high, i.e. all bytes written to the receiver are set to 0xFF.

To prevent the receiver from being busy parsing incoming data, the parsing process is stopped after 50 subsequent bytes containing 0xFF. The parsing process is re-enabled with the first byte not equal to 0xFF.

If the receiver has no more data to send, it sets `MISO` to logic high, i.e. all bytes transmitted decode to 0xFF. An efficient parser in the host will ignore all 0xFF bytes which are not part of a message and will resume data processing as soon as the first byte not equal to 0xFF is received.

3.3.13.2.3 Back-to-back read and write access

The receiver does not provide any write access except for writing UBX and NMEA messages to the receiver, such as configuration or aiding data. For every byte written to the receiver, a byte will

CONFIDENTIAL

simultaneously be read from the receiver. While the master writes to `MOSI`, at the same time it needs to read from `MISO`, as any pending data will be output by the receiver with this access. The data on `MISO` represents the results from a current address read, returning 0xFF when no more data is available.

Figure 15: SPI back-to-back read/write access

3.3.13.3 USB port

A single USB port is provided for host communication purposes. See the ZED-F9K Data sheet [1] for availability. This port can be used for communication purposes and to power the positioning chip or module.

The ZED-F9K module supports only self-powered mode operation in which the receiver is supplied from its own power supply. The V_USB pin is used to detect the availability of the USB port, i.e. whether the receiver is connected to a USB host.

☞ USB bus powered mode is not supported.

☞ The voltage range for V_USB is specified from 3.0 V to 3.6 V, which differs slightly from the specification for VCC.

☞ The boot screen is retransmitted on the USB port after the enumeration. However, messages generated between boot-up of the receiver and USB enumeration are not visible on the USB port.

3.3.13.4 TX-READY indication

This feature enables each port to define a corresponding pin, which indicates if bytes are ready to be transmitted. By default, this feature is disabled. For USB, this feature is configurable but might not behave as described below due to a different internal transmission mechanism. If the number of pending bytes reaches the threshold configured for this port, the corresponding pin will become active (configurable active-low or active-high), and stay active until the last bytes have been transferred from software to hardware.

☞ This is not necessarily equal to all bytes transmitted, i.e. after the pin has become inactive, up to 16 bytes can still need to be transferred to the host.

The TX-READY pin can be selected from all PIOs which are not in use (see UBX-MON-HW3 for a list of the PIOs and their mapping), each TX-READY pin is exclusively for one port and cannot be shared. If the PIO is invalid or already in use, only the configuration for the TX-READY pin is ignored, the rest of the port configuration is applied if valid. The acknowledge message does not indicate if the TX-READY configuration is successfully set, it only indicates the successful configuration of the port. To

CONFIDENTIAL

validate successful configuration of the TX-READY pin, the port configuration should be polled and the settings of TX-READY feature verified (will be set to disabled/all zero if the settings are invalid).

The threshold should not be set above 2 kB, as the internal message buffer limit can be reached before this, resulting in the TX-READY pin never being set as messages are discarded before the threshold is reached.

3.3.13.5 Extended TX timeout

If the host does not communicate over SPI or DDC for more than approximately 2 seconds, the device assumes that the host is no longer using this interface and no more packets are scheduled for this port. This mechanism can be changed by enabling "extended TX timeouts", in which case the receiver delays idling the port until the allocated and undelivered bytes for this port reach 4 kB. This feature is especially useful when using the TX-ready feature with a message output rate of less than once per second, and polling data only when data is available, determined by the TX-READY pin becoming active.

3.3.13.6 DDC (I^2C) port

The display data channel (DDC) bus is a two-wire communication interface compatible with the I^2C standard.

Unlike all other interfaces, the DDC is not able to communicate in full-duplex mode, i.e. TX and RX are mutually exclusive. u-blox receivers act as a slave in the communication setup, therefore they cannot initiate data transfers on their own. The host, which is always master, provides the data clock (SCL), and the clock frequency is therefore not configurable on the slave.

The receiver's DDC address is set to 0x42 by default.

As the receiver will be run in slave mode and the DDC physical layer lacks a handshake mechanism to inform the master about data availability, a layer has been inserted between the physical layer and the UBX and NMEA layer. The receiver DDC interface implements a simple streaming interface that allows the constant polling of data, discarding everything that is not parse-able. The receiver returns 0xFF if no data is available. The TX-ready feature can be used to inform the master about data availability and can be used as a trigger for data transmission.

3.3.13.6.1 Read access

The DDC interface allows 256 slave registers to be addressed. As shown in Figure 16 only three of these are currently implemented. The data registers 0 to 252, at addresses 0x00 to 0xFC, each 1 byte in size, contain information to be defined later - the result of reading them is undefined. The currently available number of bytes in the message stream can be read at addresses 0xFD and 0xFE. The register at address 0xFF allows the data stream to be read. If there is no data awaiting transmission from the receiver, then this register will deliver the value 0xFF, which cannot be the first byte of a valid message. If message data is ready for transmission, then successive reads of register 0xff will deliver the waiting message data.

☞ The registers 0x00 to 0xFC are reserved for future use and may be defined in a later firmware release. Do not use them, as they don't provide any meaningful data.

CONFIDENTIAL

Figure 16: DDC register layout

3.3.13.6.1.1 Read access forms

There are two forms of DDC read transfer. The "random access" form includes a slave register address and thus allows any register to be read. The second "current address" form omits the register address. If this second form is used, then an address pointer in the receiver is used to determine which register to read. This address pointer will increment after each read unless it is already pointing at register 0xFF, the highest addressable register, in which case it remains unaltered. The initial value of this address pointer at start-up is 0xFF, so by default all current address reads will repeatedly read register 0xFF and receive the next byte of message data (or 0xFF if no message data is waiting). Figure 17 shows the format of the random access form of the request. Following the start condition from the master, the 7-bit device address and the `RW` bit (which is a logic low for write access) are clocked onto the bus by the master transmitter. The receiver answers with an acknowledge (logic low) to indicate that it recognizes the address. Next, the 8-bit address of the register to be read must be written to the bus. Following the receiver's acknowledgement, the master again triggers a start condition and writes the device address, but this time the `RW` bit is a logic high to initiate the read access. Now, the master can read 1 to `N` bytes from the receiver, generating a not-acknowledge and a stop condition after the last byte being read.

CONFIDENTIAL

Figure 17: DDC random read access

The format of the current address read request is:

Figure 18: DDC current address read access

3.3.13.6.2 Write access

The receiver does not provide any write access except for writing UBX and NMEA messages to the receiver, such as configuration or aiding data. Therefore, the register set mentioned in section Read Access is not writeable. Following the start condition from the master, the 7-bit device address and the `RW` bit (which is a logic low for write access) are clocked onto the bus by the master transmitter. The receiver answers with an acknowledge (logic low) to indicate that it is responsible for the given address. Now, the master can write 2 to `N` bytes to the receiver, generating a stop condition after the last byte being written. The number of data bytes must be at least 2 to properly distinguish from the write access to set the address counter in random read accesses.

CONFIDENTIAL

Figure 19: DDC write access

3.4 Multiple GNSS assistance (MGA)

The u-blox MGA services provide a proprietary implementation of an A-GNSS protocol compatible with u-blox GNSS receivers. When a client device makes an MGA request, the service responds with the requested data using UBX protocol messages. These messages are ready for direct transmission to the receiver communication port without requiring any modification by the client.

AssistNow Online optionally provides satellite ephemerides, health information and time aiding data suitable for GNSS receiver systems with direct internet access.

The data downloaded from the service is organized by date and encoded into a sequence of UBX MGA messages.

The ZED-F9K supports AssistNow Online only.

3.4.1 AssistNow Online

AssistNow Online is u-blox's end-to-end Assisted GNSS (A-GNSS) solution for receivers that have access to the Internet. Data supplied by the AssistNow Online Service can be directly uploaded to a u-blox receiver in order to substantially reduce time to first fix (TTFF), even under poor signal conditions. The system works by collecting data such as ephemeris and almanac from the satellites through u-blox's "Global Reference Network" of receivers and providing this data to customers in a convenient form that can be forwarded directly to u-blox receivers.

The AssistNow Online Service uses a simple, stateless, HTTP interface. Therefore, it works on all standard mobile communication networks that support Internet access, including GPRS, UMTS and Wireless LAN. No special arrangements need to be made with mobile network operators to enable AssistNow Online.

CONFIDENTIAL

Figure 20: MGA architecture

The data returned by the AssistNow Online Service is a sequence of UBX-MGA messages, starting with an estimate of the current time in the form of a UBX-MGA-INI-TIME_UTC message.

☞ AssistNow Online currently supports GPS, GLONASS, BeiDou, Galileo, and QZSS.

☞ Customers may choose to use third party sources of assistance data instead of using the AssistNow Online Service. Customers choosing this option will need to ensure that the data is converted from the format used by the third party source to the appropriate MGA messages. However, it is important to ensure that the receiver has an estimate of the current time before it processes any other assistance data. For this reason, it is strongly recommended to send a UBX-MGA-INITIME_UTC or UBX-MGA-INI-TIME_GNSS as the first message of any assistance.

3.4.2 Host software

As u-blox receivers have no means to connect directly with the Internet, the AssistNow Online system can only work if the host system that contains the receiver can connect to the Internet, download the data from the AssistNow Online Service and forward it on to the receiver. In the simplest case that may involve fetching the data from the AssistNow Online Service (by means of a single HTTP or HTTPS GET request), and sending the resulting data to the receiver.

Depending on the circumstances, it may be beneficial for the host software to include:

- Creating an appropriate UBX-MGA-INI-TIME_UTC message to deliver a better estimation of the current time to the receiver, especially if the host system has a very good estimation of the current time and can deliver a time pulse to one of the receiver's EXTINT pins.
- Enable and use flow control to prevent loss of data due to buffer overflow in the receiver.

☞ u-blox provides the source code for an example library, called libMGA, that provides all of the functionality we expect in most host software.

3.4.3 AssistNow Online sequence

A typical sequence of use of the AssistNow Online Service comprises the following steps:

- Power-up the u-blox receiver.

- Request data from the AssistNow Online Service.
- Optionally send UBX-MGA-INI-TIME_UTC followed by hardware time synchronization pulse if hardware time synchronization is required.
- Send the UBX messages obtained from the AssistNow Online Service to the receiver.

3.4.4 Flow control

u-blox receivers aim to process incoming messages as quickly as possible, but there will always be a small delay in processing each message. Uploading assistance data to the receiver can involve sending as many as one hundred of individual messages to the receiver, one after the other. If the communication link is fast, and/or the receiver is busy (trying to acquire new signals), it is possible that the internal buffers will overflow and some messages will be lost. In order to combat this, u-blox receivers support an optional flow control mechanism for assistance.

Flow control is activated by using the CFG-NAVSPG-ACKAIDING configuration item. As a result the receiver will issue an acknowledgment message (ACK) for each assistance message it successfully receives. The host software can examine these acknowledgments to establish whether there were any problems with the data sent to the receiver and deduce (by the lack of acknowledgment) if any messages have been lost. It may then be appropriate to resend some of the assistance messages.

The simplest way to implement flow control would be to send one UBX-MGA message at a time, waiting for the acknowledgment, before sending the next. However, such a strategy is likely to introduce significant delays into the whole assistance process. The best strategy will depend on the amount of assistance data being sent and the nature of the communications link (e.g. baud rate of serial link). u-blox recommends that when customers are developing their host software they start by sending all assistance messages and then analyze the resulting acknowledgments to see if there are any loses of messages. Adding small delays during the transmission may be a simple but effective way to avoid loss of data.

3.4.5 Authorization

The AssistNow Online Service is only available for use by u-blox customers. In order to use the services, customers will need to obtain an authorization token from u-blox. This token must be supplied as a parameter whenever a request is made to either service.

3.4.6 Service parameters

The information exchange with the AssistNow Online Service is based on the HTTPS protocol. Upon reception of an HTTPS GET request, the server will respond with the required messages in binary format or with an error string in text format. After delivery of all data, the server will terminate the connection.

The HTTPS GET request from the client to the server should contain a standard HTTPS query string in the request URL. The query string consists of a set of "key=value" parameters in the following form:

key=value;key=value;key=value;

The following rules apply:

- The order of keys is important.
- Keys and values are case sensitive.
- Keys and values must be separated by an "equal" character ("=").
- Key/value pairs must be separated by semicolons (";").
- If a value contains a list, each item in the list must be separated by a comma (",").

CONFIDENTIAL

The following table describes the keys that are supported:

Key name	Unit/range	Optional	Description
token	String	Mandatory	The authorization token supplied by u-blox when a client registers to use the service.
gnss	String	Mandatory	A comma separated list of the GNSS for which data should be returned. Valid GNSS are: gps, gal, glo, bds and qzss (case sensitive).
datatype	String	Mandatory	A comma separated list of the data types required by the client. Valid data types are: eph, alm, aux and pos. Time data is always returned for each request. If the value of this parameter is an empty string, only time data will be returned.
lat	Numeric [degrees]	Optional	Approximate user latitude in WGS 84 expressed in degrees and fractional degrees. Must be in range -90 to 90. Example: lat=47.2.
lon	Numeric [degrees]	Optional	Approximate user longitude in WGS 84 expressed in degrees and fractional degrees. Must be in range -180 to 180. Example: lon=8.55.
alt	Numeric [meters]	Optional	Approximate user altitude above WGS 84 Ellipsoid. If this value is not provided, the server assumes an altitude of 0 meters. Must be in range -1000 to 50000.
pacc	Numeric [meters]	Optional	Approximate accuracy of submitted position (see position parameters note below). If this value is not provided, the server assumes an accuracy of 300km. Must be in range 0 to 6000000.
tacc	Numeric [seconds]	Optional	The timing accuracy (see time parameters note below). If this value is not provided, the server assumes an accuracy of 10 seconds. Must be in range 0 to 3600.
latency	Numeric [seconds]	Optional	Typical latency between the time the server receives the request, and the time when the assistance data arrives at the u-blox receiver. The server can use this value to correct the time being transmitted to the client. If this value is not provided, the server assumes a latency of 0. Must be in range 0 to 3600.
filteronpos	(no value required)	Optional	If present, the ephemeris data returned to the client will only contain data for the satellites which are likely to be visible from the approximate position provided by the lat, lon, alt and pacc parameters. If the lat and lon parameters are not provided the service will return an error.
filteronsv	String	Optional	A comma separated list of u-blox gnssId:svId pairs. The ephemeris data returned to the client will only contain data for the listed satellites.

Table 14: AssistNow Online parameter keys

Thus, as an example, a valid parameter string would be:

token=XXXXXXXXXXXXXXXXXXXXXX;gnss=gps,qzss;datatype=eph,pos,aux;lat=47.28;lon=8.56;pacc=1000

3.4.6.1 Position parameters (lat, lon, alt and pacc)

The position parameters (lat, lon, alt and pacc) are used by the server for two purposes:

- If the filteronpos parameter is provided, the server determines the currently visible satellites at the user position, and only sends the ephemeris data of those satellites which should be in view at the location of the user. This reduces bandwidth requirements. In this case the "pacc" value is taken into account, meaning that the server will return all SVs visible in the given uncertainty region.
- If the datatype "pos" is requested, the server will return the position and accuracy in the response data. When this data is supplied to the u-blox receiver, depending on the accuracy of the provided data, the receiver can then choose to select a better startup strategy. For example, if the position is accurate to 100km or better, the u-blox receiver will choose to go for a more optimistic startup strategy. This will result in quicker startup time. The receiver will decide which strategy to choose, depending on the "pacc" parameter. If the submitted user position is less accurate than what is being specified with the "pacc" parameter, then the user will experience prolonged or even failed startups.

3.4.6.2 Time parameters (tacc and latency)

Time data is always returned with each request. The time data refers to the time at which the response leaves the server, corrected by an optional latency value. This time data provided by the

CONFIDENTIAL

service is accurate to approximately 10 ms but by default the time accuracy is indicated to be +/-10 seconds in order to account for network latency and any time between the client receiving the data and it being provided to the receiver.

If both the network latency and the client latency can safely be assumed to be very low (or are known), the client can choose to set the accuracy of the time message (tacc) to a much smaller value (e.g. 0.5 s). This will result in a faster TTFF. The latency can also be adjusted as appropriate. However, these fields should be used with caution: if the time accuracy is not correct when the time data reaches the receiver, the receiver may experience prolonged or even failed start-ups.

For optimal results, the client should establish an accurate sense of time itself (e.g. by calibrating its system clock using a local NTP service) and then modify the time data received from the service as appropriate.

3.4.7 Multiple servers

u-blox has designed and implemented the AssistNow Online Service in a way that should provide very high reliability. Nonetheless, there will be rare occasions when a server is not available (e.g. due to failure or some form of maintenance activity). In order to protect customers against the impact of such outages, u-blox runs at least two instances of the AssistNow Online Service on independent machines. Customers have a choice of requesting assistance data from any of these servers, as all will provide the same information. However, should one fail for whatever reason, it is highly unlikely that the other server(s) will also be unavailable. Therefore customers requiring the best possible availability are recommended to implement a scheme where they direct their requests to a chosen server, but, if that server fails to respond, have a fallback mechanism to use another server instead.

3.5 Clocks and time

This section introduces and explains the concepts of receiver clocks and time bases.

3.5.1 Receiver local time

The receiver is dependent on a local oscillator for both the operation of its radio parts and also for timing within its signal processing. No matter what nominal frequency the local oscillator has, u-blox receivers subdivide the oscillator signal to provide a 1 kHz reference clock signal, which is used to drive many of the receiver's processes. In particular, the measurement of satellite signals is arranged to be synchronized with the "ticking" of this 1 kHz clock signal.

When the receiver first starts, it has no information about how these clock ticks relate to other time systems; it can only count time in 1 millisecond steps. However, as the receiver derives information from the satellites it is tracking or from aiding messages, it estimates the time that each 1 kHz clock tick takes in the time-base of the relevant GNSS system. In previous generations of u-blox receivers this was always the GPS time-base, but for this generation it could be GPS, GLONASS, Galileo, or BeiDou. This estimate of GNSS time based on the local 1 kHz clock is called receiver local time.

As receiver local time is a mapping of the local 1 kHz reference onto a GNSS time-base, it may experience occasional discontinuities, especially when the receiver first starts up and the information it has about the time-base is changing. Indeed, after a cold start, the receiver local time will initially indicate the length of time that the receiver has been running. However, when the receiver obtains some credible timing information from a satellite or aiding message, it will jump to an estimate of GNSS time.

CONFIDENTIAL

3.5.2 Navigation epochs

Each navigation solution is triggered by the tick of the 1 kHz clock nearest to the desired navigation solution time. This tick is referred to as a navigation epoch. If the navigation solution attempt is successful, one of the results is an accurate measurement of time in the time-base of the chosen GNSS system, called GNSS system time. The difference between the calculated GNSS system time and receiver local time is called the clock bias (and the clock drift is the rate at which this bias is changing).

In practice the receiver's local oscillator will not be as stable as the atomic clocks to which GNSS systems are referenced and consequently clock bias will tend to accumulate. However, when selecting the next navigation epoch, the receiver will always try to use the 1 kHz clock tick which it estimates to be closest to the desired fix period as measured in GNSS system time. Consequently the number of 1 kHz clock ticks between fixes will occasionally vary (so when producing one fix per second, there will normally be 1000 clock ticks between fixes, but sometimes, to correct drift away from GNSS system time, there will be 999 or 1001).

The GNSS system time calculated in the navigation solution is always converted to a time in both the GPS and UTC time-bases for output.

Clearly when the receiver has chosen to use the GPS time-base for its GNSS system time, conversion to GPS time requires no work at all, but conversion to UTC requires knowledge of the number of leap seconds since GPS time started (and other minor correction terms). The relevant GPS to UTC conversion parameters are transmitted periodically (every 12.5 minutes) by GPS satellites, but can also be supplied to the receiver via the UBX-MGA-GPS-UTC aiding message. By contrast when the receiver has chosen to use the GLONASS time-base as its GNSS system time, conversion to GPS time is more difficult as it requires knowledge of the difference between the two time-bases, but conversion to UTC is easier (as GLONASS time is closely linked to UTC).

Where insufficient information is available for the receiver to perform any of these time-base conversions precisely, pre-defined default offsets are used. Consequently plausible times are nearly always generated, but they may be wrong by a few seconds (especially shortly after receiver start). Depending on the configuration of the receiver, such "invalid" times may well be output, but with flags indicating their state (e.g. the "valid" flags in UBX-NAV-PVT).

☞ u-blox receivers employ multiple GNSS system times and/or receiver local times (in order to support multiple GNSS systems concurrently), so users should not rely on UBX messages that report GNSS system time or receiver local time being supported in future. It is therefore recommended to give preference to those messages that report UTC time.

3.5.3 iTOW timestamps

All the main UBX-NAV messages (and some other messages) contain an iTOW field which indicates the GPS time at which the navigation epoch occurred. Messages with the same iTOW value can be assumed to have come from the same navigation solution.

Note that iTOW values may not be valid (i.e. they may have been generated with insufficient conversion data) and therefore it is not recommended to use the iTOW field for any other purpose.

☞ The original designers of GPS chose to express time/date as an integer week number (starting with the first full week in January 1980) and a time of week (often abbreviated to TOW) expressed in seconds. Manipulating time/date in this form is far easier for digital systems than the more "conventional" year/month/day, hour/minute/second representation. Consequently, most GNSS receivers use this representation internally, only converting to a more "conventional form" at external interfaces. The iTOW field is the most obvious externally visible consequence of this internal representation.

CONFIDENTIAL

If reliable absolute time information is required, users are recommended to use the UBX-NAV-PVT navigation solution message which also contains additional fields that indicate the validity (and accuracy in UBX-NAV-PVT) of the calculated times (see also the GNSS times section below for further messages containing time information).

3.5.4 GNSS times

Each GNSS has its own time reference for which detailed and reliable information is provided in the messages listed in the table below.

Time reference	Message
GPS time	UBX-NAV-TIMEGPS
BeiDou time	UBX-NAV-TIMEBDS
GLONASS time	UBX-NAV-TIMEGLO
Galileo time	UBX-NAV-TIMEGAL
UTC time	UBX-NAV-TIMEUTC

Table 15: GNSS times

3.5.5 Time validity

Information about the validity of the time solution is given in the following form:

- Time validity: Information about time validity is provided in the `valid` flags (e.g. `validDate` and `validTime` flags in the UBX-NAV-PVT message). If these flags are set, the time is known and considered as valid for being used. These flags can be found in the GNSS times table in the GNSS times section above as well as in the UBX-NAV-PVT message.
- Time validity confirmation: Information about confirmed validity is provided in the `confirmedDate` and `confirmedTime` flags in the UBX-NAV-PVT message. If these flags are set, the time validity could be confirmed by using an additional independent source, meaning that the probability of the time to be correct is very high. Note that information about time validity confirmation is only available if the `confirmedAvai` bit in the UBX-NAV-PVT message is set.

☞ `validDate` means that the receiver has knowledge of the current date. However, it must be noted that this date might be wrong for various reasons. Only when the `confirmedDate` flag is set, the probability of the incorrect date information drops significantly.

☞ `validTime` means that the receiver has knowledge of the current time. However, it must be noted that this time might be wrong for various reasons. Only when the `confirmedTime` flag is set, the probability of incorrect time information drops significantly.

☞ `fullyResolved` means that the UTC time is known without full seconds ambiguity. When deriving UTC time from GNSS time the number of leap seconds must be known, with the exception of GLONASS. It might take several minutes to obtain such information from the GNSS payload. When the one second ambiguity has not been resolved, the time accuracy is usually in the range of ~20s.

3.5.6 UTC representation

UTC time is used in many NMEA and UBX messages. In NMEA messages it is always reported rounded to the nearest hundredth of a second. Consequently, it is normally reported with two decimal places (e.g. 124923.52). What is more, although compatibility mode (selected using CFG-NMEA-COMPAT) requires three decimal places, rounding to the nearest hundredth of a second remains, so the extra digit is always 0.

CONFIDENTIAL

UTC time is also reported within some UBX messages, such as UBX-NAV-TIMEUTC and UBX-NAV-PVT. In these messages date and time are separated into seven distinct integer fields. Six of these (year, month, day, hour, min and sec) have fairly obvious meanings and are all guaranteed to match the corresponding values in NMEA messages generated by the same navigation epoch. This facilitates simple synchronization between associated UBX and NMEA messages.

The seventh field is called nano and it contains the number of nanoseconds by which the rest of the time and date fields need to be corrected to get the precise time. So, for example, the UTC time 12:49:23.521 would be reported as: hour: 12, min: 49, sec: 23, nano: 521000000.

It is however important to note that the first six fields are the result of rounding to the nearest hundredth of a second. Consequently the nano value can range from -5000000 (i.e. -5 ms) to +994999999 (i.e. nearly 995 ms).

When the nano field is negative, the number of seconds (and maybe minutes, hours, days, months or even years) will have been rounded up. Therefore, some or all of them will need to be adjusted in order to get the correct time and date. Thus in an extreme example, the UTC time 23:59:59.9993 on 31st December 2011 would be reported as: year: 2012, month: 1, day: 1, hour: 0, min: 0, sec: 0, nano: -700000.

Of course, if a resolution of one hundredth of a second is adequate, negative nano values can simply be rounded up to 0 and effectively ignored.

Which master clock the UTC time is referenced to is output in the message UBX-NAV-TIMEUTC.

The preferred variant of UTC time can be specified using CFG-NAVSPG-UTCSTANDARD configuration item.

3.5.7 Leap seconds

Occasionally it is decided (by one of the international time keeping bodies) that, due to the slightly uneven spin rate of the Earth, UTC has moved sufficiently out of alignment with mean solar time (i.e. the Sun no longer appears directly overhead at 0 longitude at midday). A "leap second" is therefore announced to bring UTC back into close alignment. This normally involves adding an extra second to the last minute of the year, but it can also happen on 30th June. When this happens UTC clocks are expected to go from 23:59:59 to 23:59:60 and only then on to 00:00:00.

It is also theoretically possible to have a negative leap second, in which case there will only be 59 seconds in a minute and 23:59:58 will be followed by 00:00:00.

u-blox receivers are designed to handle leap seconds in their UTC output and consequently users processing UTC times from either NMEA and UBX messages should be prepared to handle minutes that are either 59 or 61 seconds long.

Leap second information can be polled from the u-blox receiver with the message UBX-NAV-TIMELS.

3.5.8 Real time clock

u-blox receivers contain circuitry to support a real time clock, which (if correctly fitted and powered) keeps time while the receiver is otherwise powered off. When the receiver powers up, it attempts to use the real time clock to initialize receiver local time and in most cases this leads to appreciably faster first fixes.

3.5.9 Date

All GNSS frequently transmit information about the current time within their data message. In most cases, this is a time of week (often abbreviated to TOW), which indicates the elapsed number of

CONFIDENTIAL

seconds since the start of the week (midnight Saturday/Sunday). In order to map this to a full date, it is necessary to know which week and so the GNSS also transmit a week number, typically every 30 seconds. Unfortunately the GPS L1C/A data message was designed in a way that only allows the bottom 10 bits of the week number to be transmitted. This is not sufficient to yield a completely unambiguous date as every 1024 weeks (a bit less than 20 years), the transmitted week number value "rolls over" back to zero. Consequently, GPS L1 receivers can't tell the difference between, for example, 1980, 1999 or 2019 etc.

Fortunately, although BeiDou and Galileo have similar representations of time, they transmit sufficient bits for the week number to be unambiguous for the foreseeable future (the first ambiguity will be in 2078 for Galileo and not until 2163 for BeiDou). GLONASS has a different structure, based on a time of day, but again transmits sufficient information to avoid any ambiguity during the expected lifetime of the system (the first ambiguous date will be in 2124). Therefore, u-blox 9 receivers using Protocol Version 24 and above regard the date information transmitted by GLONASS, BeiDou and Galileo to be unambiguous and, where necessary, use this to resolve any ambiguity in the GPS date.

☞ Customers attaching u-blox receivers to simulators should be aware that GPS time is referenced to 6th January 1980, GLONASS to 1st January 1996, Galileo to 22nd August 1999 and BeiDou to 1st January 2006; the receiver cannot be expected to work reliably with signals that appear to come from before these dates.

3.5.9.1 GPS-only date resolution

In circumstances where only GPS L1C/A signals are available and for receivers with earlier firmware versions, the receiver establishes the date by assuming that all week numbers must be at least as large as a reference rollover week number. This reference rollover week number is hard-coded at compile time and is normally set a few weeks before the software is completed, but it can be overridden by CFG-NAVSPG-WKNROLLOVER configuration item to any value the user wishes.

The following example illustrates how this works: Assume that the reference rollover week number set in the firmware at compile time is 1524 (which corresponds to a week in calendar year 2009, but would be transmitted by the satellites as 500). In this case, if the receiver sees transmissions containing week numbers in the range 500 ... 1023, these will be interpreted as week numbers 1524 ... 2047 (calendar year 2009 ... 2019), whereas transmissions with week numbers from 0 to 499 are interpreted as week numbers 2048 ... 2547 (calendar year 2019 ... 2028).

☞ It is important to set the reference rollover week number appropriately when supplying u-blox receivers with simulated signals, especially when the scenarios are in the past.

3.5.10 Time pulse

3.5.10.1 Introduction

u-blox receivers include a time pulse function providing clock pulses with configurable duration and frequency. The time pulse function can be configured using the CFG-TP-* configuration group. The UBX-TIM-TP message provides time information for the next pulse, time source and the quantization error of the output pin.

CONFIDENTIAL

Figure 21: Timepulse

3.5.10.2 Recommendations

- The time pulse can be aligned to a wide variety of GNSS times or to variants of UTC derived from them (see the chapter on time bases). However, it is strongly recommended that the choice of time base is aligned with the available GNSS signals (so to produce GPS time or UTC(USNO), ensure GPS signals are available, and for GLONASS time or UTC(SU) ensure the presence GLONASS signals). This will involve coordinating that the setting of CFG-SIGNAL-* configuration group with the choice of time pulse time base.
- When using time pulse for precision timing applications it is recommended to calibrate the antenna cable delay against a reference timing source.
- In order to get the best timing accuracy with the antenna, a fixed and *accurate* position is needed.
- If relative time accuracy between multiple receivers is required, do not mix receivers of different product families. If this is required, the receivers must be calibrated accordingly, by setting cable delay and user delay.
- The recommended configuration when using the UBX-TIM-TP message is to set both the measurement rate (CFG-RATE-MEAS) and the time pulse frequency (CFG-TP-*) to 1Hz.

☞ Since the rate of UBX-TIM-TP is bound to 1 Hz, more than one UBX-TIM-TP message can appear between two pulses if the time pulse frequency is set larger than 1 Hz. In this case all UBX-TIM-TP messages in between a time pulse T1 and T2 belong to T2 and the last UBX-TIM-TP before T2 reports the most accurate quantization error. In general, if the time pulse rate is not configured to 1 Hz, there will not be a single UBX-TIM-TP message for each time pulse.

The sequential order of the signal present at the TIMEPULSE pin and the respective output message for the simple case of 1 pulse per second (1PPS) is shown in the following figure.

CONFIDENTIAL

Figure 22: Timepulse and TIM-TP

3.5.10.3 GNSS time bases

GNSS receivers must handle a variety of different time bases as each GNSS has its own reference system time. What is more, although each GNSS provides a model for converting their system time into UTC, they all support a slightly different variant of UTC. So, for example, GPS supports a variant of UTC as defined by the US National Observatory, while BeiDou uses UTC from the National Time Service Center, China (NTSC). While the different UTC variants are normally closely aligned, they can differ by as much as a few hundreds of nanoseconds.

Although u-blox receivers can combine a variety of different GNSS times internally, the user must choose a single type of GNSS time and, separately, a single type of UTC for input (on EXTINTs) and output (via the Time Pulse) and the parameters reported in corresponding messages.

The CFG-TP-* configuration group allows the user to choose between any of the supported GNSS (GPS, GLONASS, BeiDou, etc) times and UTC. Also, the CFG-NAVSPG-* configuration group allows the user to select which variant of UTC the receiver should use. This includes an "automatic" option which causes the receiver to select an appropriate UTC version itself, based on the GNSS configuration, using, in order of preference, USNO if GPS is enabled, SU if GLONASS is enabled, NTSC if BeiDou is enabled and, finally, European if Galileo is enabled.

The receiver will assume that the input time pulse uses the same GNSS time base as specified for the output using CFG-TP-*. So if the user selects GLONASS time for time pulse output, any time pulse input must also be aligned to GLONASS time (or to the separately chosen variant of UTC). Where UTC is selected for time pulse output, any GNSS time pulse input will be assumed to be aligned to GPS time.

☞ u-blox receivers allow users to choose independently GNSS signals used in the receiver (using CFG-SIGNAL-*) and the input/output time base (using CFG-TP-*). For example it is possible to instruct the receiver to use GPS and GLONASS satellite signals to generate BeiDou time. This practice will compromise time-pulse accuracy if the receiver cannot measure the timing difference between the constellations directly and is therefore not recommended.

☞ The information that allows GNSS times to be converted to the associated UTC times is only transmitted by the GNSS at relatively infrequent periods. For example GPS transmits UTC(USNO) information only once every 12.5 minutes. Therefore, if a time pulse is configured to use a variant of UTC time, after a cold start, substantial delays before the receiver has sufficient information to start outputting the time pulse can be expected.

3.5.10.4 Time pulse configuration

u-blox ZED-F9K receivers provide a time pulse (TIMEPULSE) signal with a configurable pulse period, pulse length and polarity (rising or falling edge).

CONFIDENTIAL

It is possible to define different signal behavior (i.e. output frequency and pulse length) depending on whether or not the receiver is locked to a reliable time source. Time pulse signals can be configured using the configuration group CFG-TP-*.

3.5.10.5 Configuring time pulse with CFG-TP-*

The configuration group CFG-TP-* can be used to change the time pulse settings, and includes the following parameters defining the pulse:

- **timepulse enable** - time pulse is active if this bit is set.
- **time pulse index** - Index of time pulse output pin to be configured. A device with one time pulse output is configured with index 0.
- **antenna cable delay** - Signal delay due to the cable between antenna and receiver.
- **RF group delay** - Signal delay in the RF module of the receiver (read-only).
- **pulse frequency/period** - Frequency or period time of the pulse when locked mode is not configured or active.
- **pulse frequency/period lock** - Frequency or period time of the pulse, as soon as receiver has calculated a valid time from a received signal. Only used if the corresponding flag is set to use another setting in locked mode.
- **pulse length/ratio** - Length or duty cycle of the generated pulse, either specifies a time or ratio for the pulse to be on/off.
- **pulse length/ratio lock** - Length or duty cycle of the generated pulse, as soon as receiver has calculated a valid time from a received signal. Only used if the corresponding flag is set to use another setting in locked mode.
- **user delay** - The cable delay from the receiver to the user device plus signal delay of any user application.
- **lock to gps freq** - Use frequency gained from GPS signal information rather than local oscillator's frequency if flag is set.
- **lock to gnss freq** - Use frequency gained from GNSS signal information rather than local oscillator's frequency if flag is set.
- **locked other setting** - If this bit is set, as soon as the receiver can calculate a valid time, the alternative setting is used. This mode can be used for example to disable time pulse if time is not locked, or indicate lock with different duty cycles.
- **is frequency** - Interpret the "Frequency/Period" field as frequency rather than period if flag is set.
- **is length** - Interpret the "Length/Ratio" field as length rather than ratio if flag is set.
- **align to TOW** - If this bit is set, pulses are aligned to the top of a second.
- **polarity** - If set, the first edge of the pulse is a rising edge (Pulse Mode: Rising).
- **grid UTC/GPS** - Selection between UTC (0) or GPS (1) timegrid. Also effects the time output by UBX-TIM-TP message.
- **grid UTC/GNSS** - Selection between UTC (0), GPS (1), GLONASS (2) and Beidou (3) timegrid. Also effects the time output by UBX-TIM-TP message.

☞ The maximum pulse length can't exceed the pulse period.

☞ Time pulse settings shall be chosen in such a way, that neither the high nor the low period of the output is less than 50 ns (except when disabling it completely), otherwise pulses can be lost.

3.5.10.5.1 Example

The example below shows the 1PPS TIMEPULSE signal generated on the time pulse output according to the specific parameters of the CFG-TP-* configuration group:

- **CFG-TP-TP1_ENA** = 1
- **CFG-TP-PERIOD_TP1** = 100 000 µs

CONFIDENTIAL

- **CFG-TP-LEN_TP1** = 100 000 µs
- **CFG-TP-TIMEGRID_TP1** = 1 (GPS)
- **CFG-TP-PULSE_LENGTH_DEF** = 0 (Period)
- **CFG-TP-ALIGN_TO_TOW_TP1** = 1
- **CFG-TP-USE_LOCKED_TP1** = 1
- **CFG-TP-POL_TP1** = 1
- **CFG-TP-PERIOD_LOCK_TP1** = 100 000 µs
- **CFG-TP-LEN_LOCK_TP1** = 100 000 µs

The 1 Hz output is maintained whether or not the receiver is locked to GPS time. The alignment to TOW can only be maintained when GPS time is locked.

Figure 23: Time pulse signal with the example parameters

3.5.11 Timemark

The receiver can be used to provide an accurate measurement of the time at which a pulse was detected on the external interrupt pin. The reference time can be chosen by setting the time source parameter to UTC, GPS, GLONASS, BeiDou, Galileo or local time in the CFG-TP-* configuration group. The UTC standard can be set in the CFG-NAVSPG-* configuration group. The delay figures defined with CFG-TP-* are also applied to the results output in the UBX-TIM-TM2 message.

A UBX-TIM-TM2 message is output at the next epoch if

- The UBX-TIM-TM2 message is enabled, or
- A rising or falling edge was triggered since last epoch on one of the EXTINT channels.

The UBX-TIM-TM2 messages includes the time of the last timemark, new rising/falling edge indicator, time source, validity, number of marks and a quantization error. The timemark is triggered continuously.

☞ Only the last rising and falling edge detected between two epochs is reported since the output rate of the UBX-TIM-TM2 message corresponds to the measurement rate configured with CFG-RATE-MEAS (see Figure 24 below).

CONFIDENTIAL

Figure 24: Timemark

3.6 Security (operating, monitoring and maintaining)

A host application should be designed to monitor the various security indicators and provide the information to the user.

3.6.1 Receiver status monitoring

Messages in the UBX class UBX-MON are used to report the status of the parts of the embedded computer sub-system that are not GNSS specific.

The main purposes are:

- Hardware and software versions, using UBX-MON-VER.
- Status of the communications input/output system, using UBX-MON-COMMS.
- Status of various hardware sections with UBX-MON-HW3 and UBX-MON-RF.

3.6.1.1 Input/output system monitoring

The I/O system is a GNSS-internal layer where all data input- and output capabilities (such as UART, DDC, SPI, USB) of the GNSS receiver are combined. Each communications task has buffers assigned, where data are queued. For data originating at the receiver, to be communicated over one or multiple communications queues, the UBX-MON-COMMS message can be used to monitor the status of the queues. This message shows the current and maximum buffer usage, as well as error conditions.

☞ If the amount of data configured is too much for a certain port's bandwidth (e.g. all UBX messages output on a UART port with a baud rate of 9600), the buffer will fill up. Once the buffer space is exceeded, new messages to be sent will be dropped.

Inbound data to the GNSS receiver are placed in buffers. Usage of these buffers can be monitored through the UBX-MON-COMMS message. Further, as data are then parsed within the receiver (e.g. to separate UBX and NMEA data), the UBX-MON-COMMS message can be used. This message shows (for each port and protocol) how many messages were successfully received. It also shows (for each port) how many bytes were discarded because they were not in any of the supported protocol framings.

3.6.1.2 Jamming/interference indicator / cw jamming monitoring

The field `jamInd` of the UBX-MON-RF message can be used as an indicator for continuous wave (narrow-band) jammers/interference only. The interpretation of the value depends on the application. It is necessary to run the receiver in an unjammed environment to determine an appropriate value for the unjammed case. If the value rises significantly above this threshold, this indicates that a continuous wave jammer is present.

This indicator is always enabled.

The indicator is reporting any currently detected narrow-band interference over all currently configured signal bands.

3.6.1.3 Jamming/interference monitor (ITFM) / broadband interference monitoring

The field `flags` of the UBX-MON-RF message can be used as an indicator for both broadband and continuous wave (CW) jammers/interference. It is independent of the (CW only) jamming indicator described in Jamming/interference indicator above.

This monitor reports whether jamming has been detected or suspected by the receiver. The receiver monitors the background noise and looks for significant changes. Normally, with no interference detected, it will report "OK". If the receiver detects that the noise has risen above a preset threshold, the receiver reports "Warning". If in addition, there is no current valid fix, the receiver reports "Critical".

The monitor has four states as shown in the following table:

Value	Reported state	Description
0	Unknown	Jamming/interference monitor not enabled, uninitialized or antenna disconnected
1	OK	no interference detected
2	Warning	position ok but interference is visible (above the thresholds)
3	Critical	no reliable position fix and interference is visible (above the thresholds); interference is probable reason why there is no fix

Table 16: Jamming/interference monitor reported states

The monitor is disabled by default. The monitor is enabled by sending an appropriate CFG-ITFM-* configuration group with the CFG-ITFM-ENABLE bit set. In this message it is also possible to specify the thresholds at which broadband and CW jamming are reported. These thresholds should be interpreted as the dB level above "normal". It is also possible to specify whether the receiver expects an active or passive antenna.

CONFIDENTIAL

☞ The monitor algorithm relies on comparing the currently measured spectrum with a reference from when a good fix was obtained. Thus the monitor will only function when the receiver has had at least one (good) first fix, and will report "Unknown" before this time.

The monitor is reporting any currently detected interference over all currently configured signal bands.

3.6.2 Receiver message authentication

3.6.2.1 Asymmetric host interface signature description introduction

The asymmetric host interface signature feature is designed to provide a way to authenticate the host communication channels.

The feature uses ECDSA on a curve secp192r1 and assumes a pair of private and public keys.

3.6.2.2 Configuring the session id

The session id is a 24-byte number that can be configured via the CFG-SEC-ECCFGSESSIONID0, CFG-SEC-ECCFGSESSIONID1 and CFG-SEC-ECCFGSESSIONID2 configuration items in the beginning of the communication session with the receiver. Configuring the session id is necessary to enable the output of the UBX-SEC-ECSIGN message.

3.6.2.3 Enabling

To enable fully the signature feature the configuration items in the CFG-SEC need to be configured.

The UBX-SEC-ECSIGN message will only be output after enabling it on the desired ports. This can be done per port via the CFG-MSGOUT-UBX_SYC_SEC_* configuration items.

3.6.2.4 Signing

All output messages are combined together every second into a 32 bytes number, using sha-256 hash function. The resulting hash, together with a timestamp-based nonce and a session id, is signed using ECDSA and the private key. The produced signature is 48 bytes long. The hash, nonce, session id and the signature are included in the UBX-SEC-ECSIGN message, which is output after the last combined message.

3.6.2.5 Verification

In order to check authenticity of the output data, the host should combine received messages between two consecutive UBX-SEC-ECSIGN messages with sha-256 hash and run ECDSA verification over the hash, nonce and the session id using the corresponding public key.

3.6.3 Spoofing detection / monitoring

3.6.3.1 Introduction

Spoofing is the process whereby someone tries to forge a GNSS signal with the intention of fooling the receiver into calculating a different user position than the true one.

The spoofing detection feature monitors the GNSS signals for suspicious patterns indicating that the receiver is being spoofed. A flag in UBX-NAV-STATUS message (flags2 - `spoofDetState`) alerts the user to potential spoofing.

The spoofing detection feature monitors suspicious changes in the GNSS signal indicating external manipulation. Therefore the detection is only successful when the signal is genuine first and when the transition to the spoofed signal is being observed directly. When a receiver is started up to a spoofed signal the detection algorithms will be unable to recognize the spoofing. Also, the

CONFIDENTIAL

algorithms rely on availability of signals from multiple GNSS constellations; the detection does not work in single GNSS mode.

3.7 Broadcast navigation data

This section describes the data reported via UBX-RXM-SFRBX.

The UBX-RXM-SFRBX reports the broadcast navigation data message collected by the receiver from each tracked signal. When enabled, a separate message is generated every time the receiver decodes a complete subframe of data from a tracked signal. The data bits are reported, as received, including preambles and error checking bits as appropriate. However because there is considerable variation in the data structure of the different GNSS signals, the form of the reported data also varies. Indeed, although this document uses the term "subframe" generically, it is not strictly the correct term for all GNSS (e.g. GLONASS has "strings" and Galileo has "pages").

3.7.1 Parsing navigation data subframes

Each UBX-RXM-SFRBX message contains a subframe of data bits appropriate for the relevant GNSS, delivered in a number of 32 bit words, as indicated by numWords field.

Due to the variation in data structure between different GNSS, the most important step in parsing a UBX-RXMSFRBX message is to identify the form of the data. This should be done by reading the gnssId field, which indicates which GNSS the data was decoded from. In almost all cases, this is sufficient to indicate the structure and the following sections are organized by GNSS for that reason. However, in some cases the identity of the GNSS is not sufficient, and this is described, where appropriate, in the following sections.

In most cases, the data does not map perfectly into a number of 32 bit words and, consequently, some of the words reported in UBX-RXM-SFRBX messages contain fields marked as "Pad". These fields should be ignored and no assumption should be made about their contents.

UBX-RXM-SFRBX messages are only generated when complete subframes are detected by the receiver and all appropriate parity checks have passed.

Where the parity checking algorithm requires data to be inverted before it is decoded (e.g. GPS L1C/A), the receiver carries this out before the message output. Therefore, users can process data directly and do not need to worry about repeating any parity processing.

The meaning of the content of each subframe depends on the sending GNSS and is described in the relevant Interface Control Documents (ICD).

3.7.2 GPS

The data structure in the GPS L1C/A and L2C signals is dissimilar and thus the UBX-RXM-SFRBX message structure differs as well. For the GPS L1C/A and L2C signals it is as follows:

3.7.2.1 GPS L1C/A

For GPS L1C/A signals, there is a fairly straightforward mapping between the reported subframe and the structure of subframe and words described in the GPS ICD. Each subframe comprises ten data words, which are reported in the same order they are received.

Each word is arranged as follows:

CONFIDENTIAL

Figure 25: GPS L1C/A subframe word

3.7.2.2 GPS L2C

For GPS L2C signals each reported subframe contains the CNAV message as described in the GPS ICD. The ten words are arranged as follows:

Figure 26: GPS L2C subframe words

3.7.3 GLONASS

For GLONASS L1OF and L2OF signals, each reported subframe contains a string as described in the GLONASS ICD. This string comprises 85 data bits which are reported over three 32 bit words in the UBX-RXM-SFRBX message. Data bits 1 to 8 are always a hamming code, whilst bits 81 to 84 are a string number and bit 85 is the idle chip, which should always have a value of zero. The meaning of other bits vary with string and frame number.

The fourth and final 32 bit word in the UBX-RXM-SFRBX message contains frame and superframe numbers (where available). These values aren't actually transmitted by the SVs, but are deduced by the receiver and are included to aid decoding of the transmitted data. However, the receiver does not always know these values, in which case a value of zero is reported.

The four words are arranged as follows:

Figure 27: GLONASS subframe words

In some circumstances, (especially on startup) the receiver may be able to decode data from a GLONASS SV before it can identify the SV. When this occurs UBX-RXM-SFRBX messages will be issued with an svId of 255 to indicate "unknown".

3.7.4 BeiDou

For BeiDou B1I D1, B1I D2, B2I D1, B2I D2 signals, there is a fairly straightforward mapping between the reported subframe and the structure of subframe and words described in the BeiDou ICD. Each subframe comprises ten data words, which are reported in the same order they are received.

Each word is arranged as follows:

Figure 28: BeiDou subframe word

Note that as the BeiDou data words only comprise 30 bits, the 2 most significant bits in each word reported by UBX-RXM-SFRBX are padding and should be ignored.

3.7.5 Galileo

The Galileo E1 C/B and E5 bI/bQ signals both transmit the I/NAV message but in different configurations. The UBX-RXM-SFRBX structures for them are as follows.

3.7.5.1 Galileo E1 C/B

For Galileo E1 C/B signals, each reported subframe contains a pair of I/NAV pages as described in the Galileo ICD.

Galileo pages can either be "Nominal" or "Alert" pages. For Nominal pages the eight words are arranged as follows:

Figure 29: Galileo E1 C/B subframe words

3.7.5.2 Galileo E5 b1/bQ

For Galileo E5 b1/bQ signals, each reported subframe contains a pair of I/NAV pages as described in the Galileo ICD. Galileo pages can either be "Nominal" or "Alert" pages. For Nominal pages the eight words are arranged as follows:

Figure 30: Galileo E5 b1/bQ subframe words

Alert pages are reported in very similar manner, but the page type bits will have value 1 and the structure of the eight words will be slightly different (as indicated by the Galileo ICD).

3.7.6 QZSS

The structure of the data delivered by QZSS L1C/A signals is effectively identical to that for GPS (L1C/A).

Similarly the QZSS L2C signal is effectively identical to the GPS (L2C).

3.7.7 Summary

The following table gives a summary of the different data message formats reported by the UBX-RXM-SFRBX message:

GNSS	Signal	gnssId	sigId	numWords	period
GPS	L1C/A	0	0	10	6s
GPS	L2CL	0	3	10	12s
GPS	L2CM	0	4	10	12s
Galileo	E1 C	2	0	8	2s
Galileo	E1 B	2	1	8	2s
Galileo	E5 bI	2	5	8	2s
Galileo	E5 bQ	2	6	8	2s
BeiDou	B1I D1	3	0	10	6s
BeiDou	B1I D2	3	1	10	0.6s
BeiDou	B2I D1	3	2	10	0.6s
BeiDou	B2I D2	3	3	10	0.6s
QZSS	L1C/A	5	0	10	6s
QZSS	L2CM	5	4	10	12s
QZSS	L2CL	5	5	10	12s
GLONASS	L1OF	6	0	4	2s
GLONASS	L2OF	6	2	4	2s

Table 17: Data message formats reported by UBX-RXM-SFRBX

3.8 Forcing a receiver reset

Typically, in GNSS receivers, one distinguishes between cold, warm, and hot starts, depending on the type of valid information the receiver has at the time of the restart.

- **Cold start:** In cold start mode, the receiver has no information from the last position (e.g. time, velocity, frequency etc.) at startup. Therefore, the receiver must search the full time and frequency space, and all possible satellite numbers. If a satellite signal is found, it is tracked to decode the ephemeris (18-36 seconds under strong signal conditions), whereas the other channels continue to search satellites. Once there is a sufficient number of satellites with valid ephemeris, the receiver can calculate position and velocity data. Other GNSS receiver manufacturers call this startup mode **Factory startup**.
- **Warm start:** In warm start mode, the receiver has approximate information for time, position, and coarse satellite position data (Almanac). In this mode, after power-up, the receiver normally needs to download ephemeris before it can calculate position and velocity data. As the ephemeris data usually is outdated after 4 hours, the receiver will typically start with a warm start if it has been powered down for more than 4 hours. In this scenario, several augmentations are possible. See the section on Multi-GNSS Assistance.
- **Hot start:** In hot start mode, the receiver was powered down only for a short time (4 hours or less), so that its ephemeris is still valid. Since the receiver does not need to download ephemeris again, this is the fastest startup method.

In the UBX-CFG-RST message, one can force the receiver to reset and clear data, in order to see the effects of maintaining/losing such data between restarts. For this, the UBX-CFG-RST message offers the `navBbrMask` field, where hot, warm and cold starts can be initiated, and also other combinations thereof.

CONFIDENTIAL

☞ Data stored in flash memory is not cleared by any of the options provided by UBX-CFG-RST. So, for example, if valid AssistNow Offline data is stored in the flash it is likely to have an impact on a "cold start".

The reset type can also be specified. This is not related to GNSS, but to the way the software restarts the system.

- **Hardware reset** uses the on-chip watchdog, in order to electrically reset the chip. This is an immediate, asynchronous reset. No Stop events are generated.
- **Controlled software reset** terminates all running processes in an orderly manner and, once the system is idle, restarts operation, reloads its configuration and starts to acquire and track GNSS satellites.
- **Controlled software reset (GNSS only)** only restarts the GNSS tasks, without reinitializing the full system or reloading any stored configuration.
- **Controlled GNSS stop** stops all GNSS tasks. The receiver will not be restarted, but will stop any GNSS related processing.
- **Controlled GNSS start** starts all GNSS tasks.

3.9 Firmware update description

ZED-F9K is supplied with a fully functional firmware which has been rigorously tested and meets our stringent release criteria. u-blox may release updated images containing for example, security fixes, enhancements, bug fixes etc. Therefore it is important that customers implement a firmware update mechanism in their system.

u-blox can provide a platform-independent sample code for updating the firmware.

3.9.1 Capabilities

The update class UBX-UPD provides an interface to transfer memory data from the host to the receiver and vice-versa. Additionally, there are messages to perform specific sequences such as the flash memory erase and write routines, reading the flash organization, or calculating a checksum over a memory region. This class also includes some special messages to reboot the receiver in a special mode, or to control queue sizes at the receiver.

3.9.2 Firmware update to flash

The following sequence diagram describes the procedure before the flashing can start:

CONFIDENTIAL

Figure 31: Procedure before the flashing can start

A firmware image is a binary file containing the software to be run by the GNSS receiver. A firmware update is the process of transferring a firmware image to the receiver and storing it in non-volatile flash memory.

During the time that the erase and write procedures are running, the flash memory (including processor instructions) cannot be read. For this reason, the receiver must be booted from internal ROM memory prior to erasing or writing flash. The message UBX-UPD-SAFE prompts the receiver to perform a reset, executing code from ROM. As the receiver starts from ROM, stored configurations will not be applied, including the baud rate for the serial ports.

As the receiver is running out of the internal ring oscillator, a training sequence must be sent to synchronize the baud rate (which is expected to be at 9600 baud). This consists of the following sequence: 0x5555. Note that this message is not encapsulated in a UBX message, but consists of only these two bytes. After sending the training sequence the host has to wait for at least 2 ms before sending messages to the receiver. To make sure that communication still functions, the version can be polled by sending the UBX-MON-VER message. See *[1.Safeboot]* from Figure 31.

Since different flash types are supported, the flash organization (block sizes and number of blocks) can be determined by reading the unique manufacturer and device IDs from the flash. The organization structure can then be looked up at the host. The IDs are read by processing the receiver's response to the UBX-UPD-FLDET message. See *[2.Detect the flash]* from Figure 31.

The following sequence diagram describes the erasing and flashing:

Figure 32: Erasing and flashing sequence

With the flash organization available, the main firmware update procedure can begin. Flash memory needs to be erased before it can be written.

The simplest procedure is to first delete all sectors necessary for the firmware to fit into by using the UBX-UPD-CERASE. See *[4.Erase the flash]* from Figure 32.

Then, beginning with the first packet, each packet is transferred to the receiver by using the UBX-UPD-FLWRI command until the whole image is written to flash. See *[5. Send the image...]* from Figure 32. A faster alternative (as can be also seen in the sample code) is to transfer write-packets into the receiver's queue while the receiver waits for the flash sectors to be written.

See the verification description from the section Firmware Verification below. See *[6.Verify...]* from Figure 32.

If unsuccessful, the process can be repeated. If the update has completed, then the receiver can be rebooted by sending the UBX-UPD-RBOOT command.

3.9.3 Firmware Update Sample Code

Sample code for the firmware update written in C++ is available from u-blox. Contact your nearest u-blox Field Application Engineer to get a copy. It is also available as part of the Chipset Development Kit. The sample shows the whole firmware update process implementation. Platform dependent code was added for Windows based and Linux based operating systems and should be easily

adaptable for any other platform. The code was written for little-endian systems, but can be adapted to also work on big-endian systems. The sample code includes code to use the following communication interfaces:

- Serial (RS232)
- USB
- DDC via Aardvark hardware (a USB to DDC converter board)
- SPI via Aardvark hardware (a USB to SPI converter board)

3.9.4 Firmware Verification

Before trying to start the new firmware, the content can be verified against the image by using the UBX-UPD-CRC (version 1) message. This message checks the validity of the downloaded image without any knowledge about the image structure. This is done by computing a fletcher32 checksum of the binary on the host, and then sending that computed checksum to the receiver, which will do the same and compare it to the received checksum. See the table below for details about the field values. The result will then be returned in UBX-UPD-CRC message.

☞ Note that the CRC algorithm in the ROM is only prepared to check 4 byte aligned buffers with a size that is a multiple of 4. If the image to download doesn't have a size which is a multiple of 4 bytes, the image will need to be padded before starting the download process. Any value can be used as padding. However, we recommend to use 0xFF.

Field	Value
version	0x01
region	0x00(RAM) or 0x01(flash)
addr	0x00800000(RAM) or 0x48(flash)
size	actual size of binary
crcA	First word of fletcher32 checksum of binary
crcB	Second word of fletcher32 checksum of binary

Table 18: Fields of UBX-UPD-CRC (version 1)

CONFIDENTIAL

4 Design

This section provides information to help carry out a successful schematic and PCB design.

⚠ Do not load Pin 4 (ANT_DETECT) with a capacitance more than 1 nF.

4.1 Pin assignment

The pin assignment of the ZED-F9K module is shown in Figure 33. The defined configuration of the PIOs is listed in Table 19.

☞ The ZED-F9K is an LGA package with the I/O on the outside edge and central ground pads.

Figure 33: ZED-F9K pin assignment

Pin No	Name	I/O	Description
1	GND	-	Ground
2	RF_IN	I	RF input
3	GND	-	Ground
4	ANT_DETECT	I	Active Antenna Detect
5	ANT_OFF	O	External LNA disable
6	ANT_SHORT_N	O	Active antenna short detect
7	VCC_RF	O	Voltage for external LNA
8	Reserved	-	Reserved
9	Reserved	-	Reserved
10	Reserved	-	Reserved
11	Reserved	-	Reserved
12	GND	-	Ground
13	Reserved	-	Reserved
14	GND	-	Ground
15	Reserved	-	Reserved

CONFIDENTIAL

Pin No	Name	I/O	Description
16	Reserved	-	Reserved
17	Reserved	-	Reserved
18	Reserved	-	Reserved
19	Reserved	-	Reserved
20	RTK_STAT	O	RTK status 0 – Fixed, blinking – receiving RTCM data, 1 – no corrections
21	Reserved	-	Reserved
22	WT	I	Wheel Ticks
23	DIR	I	Direction
24	Reserved	-	Reserved
25	Reserved	-	Reserved
26	RXD2	I	Correction UART input
27	TXD2	O	Correction UART output
28	Reserved	-	Reserved
29	Reserved	-	Reserved
30	Reserved	-	Reserved
31	Reserved	-	Reserved
32	GND	-	Ground
33	VCC	I	Voltage supply
34	VCC	I	Voltage supply
35	Reserved	-	Reserved
36	V_BCKP	I	Backup supply voltage
37	GND	-	Ground
38	V_USB	I	USB Power input
39	USB_DM	I/O	USB Data
40	USB_DP	I/O	USB Data
41	GND	-	Ground
42	TXD / SPI_MISO	O	Serial Port if D_SEL =1(or open). SPI MISO if D_SEL = 0
43	RXD / SPI_MOSI	I	Serial Port if D_SEL =1(or open). SPI MOSI if D_SEL = 0
44	SDA / SPI_CS_N	I/O	DDC Data if D_SEL =1 (or open). SPI Chip Select if D_SEL = 0
45	SCL / SPI_CLK	I/O	DDC Clock if D_SEL =1(or open). SPI Clock if D_SEL = 0
46	TX_READY	O	TX_Buffer full and ready for TX of data
47	D_SEL	I	Interface select
48	GND	-	Ground
49	RESET_N	I	RESET_N
50	SAFEBOOT_N	I	SAFEBOOT_N (for future service, updates and reconfiguration, leave OPEN)
51	EXT_INT	I	External Interrupt Pin
52	Reserved	-	Reserved
53	TIMEPULSE	O	Time Pulse
54	Reserved	-	Reserved

Table 19: ZED-F9K pin assigment

4.2 Power supply

The u-blox ZED-F9K module has three power supply pins: VCC, V_BCKP and V_USB.

CONFIDENTIAL

4.2.1 VCC: Main supply voltage

The **VCC** pin is connected to the main supply voltage. During operation, the current drawn by the module can vary by some orders of magnitude. For this reason, it is important that the supply circuitry be able to support the peak power for a short time (see the ZED-F9K Data sheet [1] for specification).

The module integrates a DC/DC converter, which allows reduced power consumption.

- When switching from backup mode to normal operation or at start-up, u-blox ZED-F9K modules must charge the internal capacitors in the core domain. In certain situations, this can result in a significant current draw. For low power applications using backup mode, it is important that the power supply or low ESR capacitors at the module input can deliver this current/charge.
- To reduce peak current during power on, users can employ an LDO that has an in-built current limiter
- Do not add any series resistance greater than 0.2 Ω to the VCC supply as it will generate input voltage noise due to dynamic current conditions.
- For the ZED-F9K module the equipment must be supplied by an external limited power source in compliance with the clause 2.5 of the standard IEC 60950-1.

4.2.2 V_BCKP: Backup supply voltage

The V_BCKP pin can be used to provide power to maintain the real-time clock (RTC) and battery backed RAM (BBR) when VCC is removed. V_BCKP can be provided with a battery. If V_BCKP is provided during a VCC outage, the receiver may be able to perform a hot or warm start. Especially if the RTC and BBR contents are still valid, i.e. a short VCC outage. The fusion filter data is also stored in BBR and ZED-F9K can restart in fusion mode.

If V_BCKP is not provided, the module will perform a cold start at power up, and the sensors will need to be recalibrated.

If a host is connected to ZED-F9K, V_BCKP can be partially emulated by using UBX-UPD-SOS functionality. BBR data can saved to the host and restored at startup. See Interface Description for more information.

- Avoid high resistance on the **V_BCKP** line: During the switch from main supply to backup supply, a short current adjustment peak can cause a high voltage drop on the pin with possible malfunctions.
- If no backup supply voltage is available, connect the **V_BCKP** pin to **VCC**.
- Allow all I/O including UART and other interfaces to Float/High impedance in HW backup mode (Battery Back-up connected with VCC removed). See the Interfaces section.

Real-Time Clock (RTC)

The Real-Time Clock (RTC) is driven by a 32 kHz oscillator using an RTC crystal. If VCC is removed whilst a battery is connected to **V_BCKP**, most of the receiver is switched off leaving the RTC and BBR powered. This operating mode is called Hardware Backup Mode which enables time keeping and all relevant data to be saved to allow a hot or warm start.

4.2.3 ZED-F9K power supply

The ZED-F9K high precision receiver requires a low noise, low dropout voltage, very low source impedance power supply of 3.3V typically. No inductors or ferrite beads should be used from LDO

CONFIDENTIAL

to the module VCC pin. The peak currents need to be taken into account for the source supplying the LDO for the module.

A power supply fed by 5V is shown in the figure below. This example circuit is intended only for the module supply.

Figure 34: ZED-F9K power supply

4.3 ZED-F9K minimal design

The minimal electrical circuit for ZED-F9K operation using the UART1 interface is shown in Figure 35 below.

Figure 35: Minimal ZED-F9K design

CONFIDENTIAL

For a minimal design with the ZED-F9K GNSS modules, the following functions and pins should be considered:

- Connect the power supply to VCC and V_BCKP.
- If hot or warm start operations are needed, connect a Backup Battery to V_BCKP.
- If USB is not used connect V_USB to ground.
- Ensure an optimal ground connection to all ground pins of the ZED-F9K GNSS modules.
- If antenna bias is required, see ZED-F9K antenna bias section.

The host interface typically supplies the RTCM messages required for RTK operation.

4.4 WT and DIR interface example

This section shows an example design for interfacing the WT and DIR pins.

Figure 36: Example WT and DIR design

ID	Description
R24	RES THICK FILM CHIP 0603 2K2 5% 0.1W
R25	RES THICK FILM CHIP 0603 2K2 5% 0.1W
R26	RES THICK FILM CHIP 0603 2K2 5% 0.1W
R27	RES THICK FILM CHIP 0603 2K2 5% 0.1W
R28	RES THICK FILM CHIP 0603 1K0 5% 0.1W
R29	RES THICK FILM CHIP 0603 1K0 5% 0.1W
C41	CAP CER X7R 0603 10N 10% 25V
C42	CAP CER X7R 0603 100N 10% 10V
C43	CAP CER X7R 0603 100N 10% 10V
C25	CAP CER X7R 0603 100N 10% 10V

ID	Description
C44	CAP CER X7R 0603 10N 10% 25V
C45	CAP CER X7R 0603 100N 10% 10V
C46	CAP CER X7R 0603 100N 10% 10V
C47	CAP CER X7R 0603 100N 10% 10V
D17	VOLTAGE REGULATOR DIODE FAIRCHILD BZX84 SOT23 6V2 0.2A
D18	VOLTAGE REGULATOR DIODE FAIRCHILD BZX84 SOT23 6V2 0.2A
U15	OPTOCOUPLER LVTTL/LVCMOS COMPATIBLE AVAGO HCPL-070L-000E SO8
U16	OPTOCOUPLER LVTTL/LVCMOS COMPATIBLE AVAGO HCPL-070L-000E SO8
U17	TINY LOGIC UHS INVERTER WITH SCHMITT TRIGGER FAIRCHILD NC7SZ14 SOT23-5
U18	TINY LOGIC UHS INVERTER WITH SCHMITT TRIGGER FAIRCHILD NC7SZ14 SOT23-5

Table 20: WT and DIR example

4.5 Antenna

u-blox recommends using an active antenna with ZED-F9K.

If an active antenna needs to be implemented in an application case, it is recommended that an OEM active antenna module be used that meets our specification. To implement the required RF circuitry and source the required components to meet group delay specification is not a simple process compared to previous L1 only implementation.

Figure 37: u-blox low cost dual-band antenna internal structure

☞ A suitable ground plane is required for the antenna to achieve good performance.

☞ Location of the antenna is critical to reach the stated performance. Unsuitable locations could include, under vehicle dash, rear-view mirror location, etc.

L1 + L2/E5b active antenna required specifications

Parameter	Specification	
Antenna Type		Active antenna
Active antenna recommendations	Typical gain	30 dB
	Maximum gain	40 dB
	Maximum noise figure	2 dB
L1 band antenna gain[1]	1559 - 1606 MHz: 3 dBic typ.	
L2/E5b band antenna gain[2]	1197 - 1249 MHz: 2 dBic typ.	

1 Measured with a ground plane d=150 mm

2 Measured with a ground plane d=150 mm

CONFIDENTIAL

Parameter	Specification
Polarization	RHCP
Axial ratio	2 dB max at Zenith
Phase center variation	<10 mm over elevation/azimuth
Group delay variation in-band[3]	10 ns max @ each GNSS system bandwidth. Note: Inter-signal requirement 50 ns max.
EMI immunity out-of-band[4]	30 V/m
Out-of-band [5] Rejection	40 dB typ
ESD circuit protection	15 kV human body model air discharge

Table 21: Antenna specifications for ZED-F9K modules

The antenna system should include filtering to ensure adequate protection from nearby transmitters. Care should be taken in the selection of antennas placed close to cellular or Wi-Fi transmitting antennas.

4.5.1 Antenna bias

Active antennas have an integrated low-noise amplifier that contributes an additional current of typically 5 to 20 mA to the system's power consumption budget. If customers do not want to make use of the antenna supervisor function the filtered VCC_RF supply voltage output can supply the antenna if the supply voltage of the ZED-F9K module matches the antenna working voltage (e.g. 3.0 V).

☞ A series current limiting resistor is required to prevent short circuits destroying the bias-t inductor.

☞ The bias-t inductor must be chosen for multi-band operation, a value of 120 nH 5% is recommended for the recommended Murata L part. It has a self-resonance frequency of 1 GHz and a high impedance (> 500 Ω) at L-band frequencies.

The recommended bias-t inductor (Murata LQW15ANR12J00) has a maximum current capacity of 110 mA. Hence the current is limited to 70 mA at 3.3V using an active limiter in the recommended circuit shown in Figure 39 below. A 10 Ω resistor (R2) is provided to measure the current. This resistor power rating must be chosen to ensure reliability in the chosen circuit design.

CONFIDENTIAL

3 GNSS system bandwidths: 1559... 1563 MHz; 1573... 1578 MHz; 1598... 1606 MHz; 1192... 1212 MHz; 1197... 1217 MHz; 1223... 1231 MHz; 1242... 1249 MHz

4 Exception L1 and L2 bands +/- 200 MHz, emphasis on cellular bands

5 GNSS system bandwidths: 1559... 1563 MHz; 1573... 1578 MHz; 1598... 1606 MHz; 1192... 1212 MHz; 1197... 1217 MHz; 1223... 1231 MHz; 1242... 1249 MHz

■LQW15ANR12J00,

Figure 38: ZED-F9K antenna bias inductor impedance

A recommended circuit design for an active antenna bias is shown below. This example shows an external voltage of 3.3V with current limiting as described above. An ESD protection diode should also be connected to the input as shown.

Figure 39: ZED-F9K reference design for antenna bias

L1: MURATA LQW15A LQW15ANR12J00 0402 120N 5% 0.11A -55/+125C

D1: TYCO, 0.25PF, PESD0402-140 -55/+125C

C2: MURATA GRM0335C1H470JA01 47pF ±5% 50Vdc C0G(EIA) 0±30ppm -55 to 125C

C3: MURATA GRM155R61A104KA01 CER X5R 0402 100N 10% 10V

R2: RES THICK FILM CHIP 1206 10R 5% 0.25W

It is recommended to use active current limiting. If active current limiting is not used, the following important points covered below need to be taken into account:

Figure 40: ZED-F9K VCC_RF antenna bias

☞ The bias-t inductor and current limiting resistor must be selected to be reliable with a short on the antenna feed if no active current limiter is used. Our recommended part has a limit of 110 mA, a part with a higher current capability will be needed if the short circuit current is as described here. This will also be affected by the voltage level of the antenna bias supply due to power dissipation. The current limit capability of the antenna bias supply needs to be considered. In the case where the module supplies the voltage via VCC_RF, a higher value resistor will be needed to ensure the module supply inductor is protected. The current should be limited to below 150 mA at the module supply voltage under shorting conditions. In this case a value of 17 Ω or more is required at a module supply of 3V to limit short circuit current to 150 mA. The DC resistance of the bias-t inductor is assumed to be 1-2 Ω and the module internal feed inductor is assumed to be 1.2 Ω.

If the VCC_RF voltage does not match with the supply voltage of the active antenna, use a filtered external supply.

☞ The power dissipation in the resistor and inductor needs to be taken into account based on the supply voltage and short circuit current. The bias-t inductor current capability and the bias resistor value need to be calculated as shown above. The supply voltage for the bias-t and its current capability is part of the calculation.

Figure 41: ZED-F9K external voltage antenna bias

4.6 EOS/ESD precautions

When integrating GNSS receivers into wireless systems, careful consideration must be given to electromagnetic and voltage susceptibility issues. Wireless systems include components which can produce Electrostatic Discharge (ESD), Electrical Overstress (EOS) and Electro-Magnetic Interference (EMI). CMOS devices are more sensitive to such influences because their failure mechanism is defined by the applied voltage, whereas bipolar semiconductors are more susceptible to thermal overstress. The following design guidelines are provided to help in designing robust yet cost effective solutions.

⚠ To avoid overstress damage during production or in the field it is essential to observe strict EOS/ESD/EMI handling and protection measures.

⚠ To prevent overstress damage at the RF_IN of your receiver, never exceed the maximum input power as specified in the u-blox ZED-F9K Data sheet [1].

4.6.1 ESD protection measures

⚠ GNSS receivers are sensitive to Electrostatic Discharge (ESD). Special precautions are required when handling. Most defects caused by ESD can be prevented by following strict ESD protection rules for production and handling. When implementing passive antenna patches or external antenna connection points, then additional ESD measures as shown in the figure below can also avoid failures in the field.

Figure 42: RF ESD precautions

4.6.2 EOS precautions

Electrical Overstress (EOS) usually describes situations when the maximum input power exceeds the maximum specified ratings. EOS failure can happen if RF emitters are close to a GNSS receiver or its antenna. EOS causes damage to the chip structures. If the RF_IN is damaged by EOS, it's hard to determine whether the chip structures have been damaged by ESD or EOS.

EOS protection measures as shown in the figure below are recommended for any designs combining wireless communication transceivers (e.g. GSM, GPRS) and GNSS in the same design or in close proximity.

CONFIDENTIAL

Figure 43: Active antenna EOS protection

4.6.3 Safety precautions

The ZED-F9K must be supplied by an external limited power source in compliance with the clause 2.5 of the standard IEC 60950-1. In addition to external limited power source, only Separated or Safety Extra-Low Voltage (SELV) circuits are to be connected to the module including interfaces and antennas.

☞ For more information about SELV circuits see section 2.2 in Safety standard IEC 60950-1

4.7 Electromagnetic interference on I/O lines

Any I/O signal line with a length greater than approximately 3 mm can act as an antenna and may pick up arbitrary RF signals transferring them as noise into the GNSS receiver. This specifically applies to unshielded lines, in which the corresponding GND layer is remote or missing entirely, and lines close to the edges of the printed circuit board.

If, for example, a cellular signal radiates into an unshielded high-impedance line, it is possible to generate noise in the order of volts and not only distort receiver operation but also damage it permanently. Another type of interference can be caused by noise generated at the PIO pins that emits from unshielded I/O lines. Receiver performance may be degraded when this noise is coupled into the GNSS antenna.

EMI protection measures are particularly useful when RF emitting devices are placed next to the GNSS receiver and/or to minimize the risk of EMI degradation due to self-jamming. An adequate layout with a robust grounding concept is essential in order to protect against EMI.

☞ Intended Use: In order to mitigate any performance degradation of a radio equipment under EMC disturbance, system integration shall adopt appropriate EMC design practice and not contain cables over three meters on signal and supply ports.

4.7.1 General notes on interference issues

Received GNSS signal power at the antenna are very low. At the nominal received signal strength (-128 dBm) it is below the thermal noise floor of -111 dBm. Due to this fact, a GNSS receiver is susceptible to interference from nearby RF sources of any kind. Two cases can be distinguished:

- Out-of-band interference: Typically any kind of wireless communications system (e.g. LTE, GSM, CDMA, 3G, WLAN, Bluetooth, etc.) may emit its specified maximum transmit power in close proximity to the GNSS receiving antenna, especially if such a system is integrated with the GNSS receiver. Even at reasonable antenna selectivity, destructive power levels may reach the RF input of the GNSS receiver. Also, larger signal interferers may generate intermodulation products inside the GNSS receiver front-end that fall into the GNSS band and contribute to in-band interference.
- In-band interference: Although the GNSS band is kept free from intentional RF signal sources by radio-communications standards, many devices emit RF power into the GNSS band at levels

CONFIDENTIAL

much higher than the GNSS signal itself. One reason is that the frequency band above 1 GHz is not well regulated with regards to EMI, and even if permitted, signal levels are much higher than GNSS signal power. Notably, all types of digital equipment, like PCs, digital cameras, LCD screens, etc. tend to emit a broad frequency spectrum up to several GHz of frequency. Also wireless transmitters may generate spurious emissions that fall into GNSS band.

As an example, GSM uses power levels of up to 2W (+33 dBm). The absolute maximum power input at the RF input of the GNSS receiver can be +15 dBm. The GSM specification allows spurious emissions for GSM transmitters of up to 36 dBm, while the GNSS signal is less than -128 dBm. By simply comparing these numbers it is obvious that interference issues must be seriously considered in any design of a GNSS receiver. Different design goals may be achieved through different implementations:

- The primary focus is prevention of destruction of the receiver from large input signals. Here the GNSS performance under interference conditions is not important and suppression of the GNSS signal is permitted. It is sufficient to just observe the maximum RF power ratings of all the components in the RF input path.
- GNSS performance must be guaranteed even under interference conditions. In that case, not only the maximum power ratings of the components in the receive patch must be observed. Further, non-linear effects like gain compression, NF degradation (desensitization) and intermodulation must be analyzed.

☞ Pulsed interference with a low duty cycle like e.g. GSM may be destructive due to the high peak power levels.

4.7.2 In-band interference mitigation

With in-band interference, the signal frequency is very close to the GNSS frequency. Such interference signals are typically caused by harmonics from displays, micro-controller operation, bus systems, etc. Measures against in-band interference include:

- Maintaining a good grounding concept in the design
- Shielding
- Layout optimization
- Low-pass filtering of noise sources, e.g. digital signal lines
- Remote placement of the GNSS antenna, far away from noise sources
- Adding a LTE, CDMA, GSM, WCDMA, BT band-pass filter before antenna

4.7.3 Out-of-band interference

Out-of-band interference is caused by signal frequencies that are different from the GNSS carrier frequency. The main sources are wireless communication systems such as LTE, GSM, CDMA, WCDMA, Wi-Fi, BT, etc.

Measures against out-of-band interference include maintaining a good grounding concept in the design and adding a GNSS band-pass filter into the antenna input line to the receiver.

For GSM applications, like a typical handset design, an isolation of approximately 20 dB can be reached with careful placement of the antennas. If this is insufficient, an additional SAW filter is required on the GNSS receiver input to block the remaining GSM transmitter energy.

4.8 Layout

This section details layout and placement requirements of the ZED-F9K high precision receiver.

CONFIDENTIAL

4.8.1 Placement

GNSS signals at the surface of the Earth are about 19 dB below the thermal noise floor. A very important factor in achieving maximum GNSS performance is the placement of the receiver on the PCB. The placement used may affect RF signal loss from antenna to receiver input and enable interference into the sensitive parts of the receiver chain, including the antenna itself. When defining a GNSS receiver layout, the placement of the antenna with respect to the receiver, as well as grounding, shielding and interference from other digital devices are crucial issues and need to be considered very carefully.

Signal loss on the RF connection from antenna to receiver input must be minimized as much as possible. Hence, the connection to the antenna must be kept as short as possible.

Ensure that RF critical circuits are clearly separated from any other digital circuits on the system board. To achieve this, position the receiver digital part closer to the digital section of the system PCB and have the RF section and antenna placed as far as possible away from the other digital circuits on the board.

A proper GND concept shall be followed: The RF section shall not be subject to noisy digital supply currents running through its GND plane.

Care must also be exercised with placing the receiver in proximity to circuitry that can emit heat. The RF part of the receiver is very sensitive to temperature and sudden changes can have an adverse impact on performance.

⚠ The TCXO of a GNSS receiver is a temperature sensitive device. Avoid high temperature drift and air convection.

4.8.2 Module footprint and paste mask

☞ These are recommendations only and not specifications. The exact copper, solder and paste mask geometries, distances, stencil thicknesses and solder paste volumes must be adapted to the specific production processes (e.g. soldering etc.) of the customer.

Refer to the ZED-F9K Data Sheet [1] for the mechanical dimensions.

CONFIDENTIAL

4.8.2.1 Footprint

Symbol	Min. (mm)	Typ.(mm)	Max. (mm)
H'		0.80	
K'		1.50	
R'		1.10	

Figure 44: ZED-F9K suggested footprint (i.e. copper mask)

CONFIDENTIAL

4.8.2.2 Paste mask

Symbol	Min. (mm)	Typ.(mm)	Max. (mm)
H'		0.75	
K'		1.55	
R'		1.05	

Figure 45: ZED-F9K suggested paste mask

4.8.3 Layout guidance

Presented layout guidance reduces the risk of performance issues at design level with the ZED-F9K high precision receiver.

4.8.3.1 RF In trace

The RF In trace has to work in the combined GNSS L1 + L2 signal band.

For FR-4 PCB material with a dielectric permittivity of for example 4.7 we can calculate the trace width at 1575 MHz for 50 Ω impedance.

A grounded co-planar RF trace is recommended as it provides the maximum shielding from noise with adequate vias to the ground layer.

The RF trace must be shielded by vias to Ground along the entire length of the trace and the ZED-F9K high precision receiver RF_IN pad should be surrounded by vias as shown in the figure below.

CONFIDENTIAL

Figure 46: RF input trace

The RF_IN trace on the Top layer should be referenced to a suitable ground layer.

4.8.3.2 Vias for the ground pads

The ground pads under the ZED-F9K high precision receiver need to be grounded with vias to the lower ground layer of the PCB. A solid ground layer fill on the top layer of the PCB is recommended. This is shown in the figure below.

Figure 47: Top layer fill and vias

4.8.3.3 VCC pads

The VCC pads for the ZED-F9K high precision receiver need to be as low an impedance as possible with large vias to the lower power layer of the PCB. The VCC pads need a large combined pad and the de-coupling capacitors must be placed as close as possible. This is shown in the figure below.

CONFIDENTIAL

Figure 48: VCC pads

4.9 Design guidance

4.9.1 General considerations

⚠ Do not load Pin 4 (ANT_DETECT) with a capacitance more than 1 nF.

Check power supply requirements and schematic:

- Is the power supply voltage within the specified range?
- If USB is not used, connect the V_USB pin to ground.
- If USB is used, is there a 1 µF and a 100 nF capacitor right near the V_USB pin? This is just for the V_USB pin.
- Is there a 1 µF cap right next to the module VCC pin?
- Compare the peak current consumption of the ZED-F9K GNSS module with the specification of your power supply.
- GNSS receivers require a stable power supply. Avoid series resistance (less than 0.2 Ω) in your power supply line (the line to VCC) to minimize the voltage ripple on VCC.
- Allow all I/O to Float/High impedance when VCC is not applied. See "Interfaces" section.

4.9.2 backup battery

Check backup supply requirements and schematic:

- For achieving a minimal Time To First Fix (TTFF) after a power down (warm starts, hot starts), make sure to connect a backup battery to V_BCKP.
- Verify your battery backup supply can provide the battery backup current specified in the ZED-F9K datasheet.
- Allow all I/O including UART and other interfaces to Float/High impedance in HW backup mode (Battery Back-up connected with VCC removed).

4.9.3 RF front-end circuit options

☞ It is mandatory that the RF input is fed by an active antenna meeting the requirements for the ZED-F9K.

The first stages of the signal processing chain are crucial to the overall receiver performance.

CONFIDENTIAL

When an RF input connector is employed this can provide a conduction path for harmful or destructive electrical signals. If this is a likely factor the RF input should be protected accordingly.

Additional points on the RF input

- What is the expected quality of the signal source (antenna)?
- What is the external active antenna signal power?
- What is the bandwidth and filtering of the external active antenna?
- Does the external antenna and filtering components meet the group delay variation requirements? This is critical for RTK.

Are destructive RF power levels expected to reach the RF-input? Is interference from wireless transmitters expected?

- What are the characteristics of these signals (duty cycle, frequency range, power range, spectral purity)?
- What is the expected GNSS performance under interference conditions?

Is there a risk of RF-input exposure to excessive ESD stress?

- In the field: Is the antenna connector accessible by the user?
- PCB / system assembly: Is there risk that statically charged parts (e.g. patch antennas) may be discharged through the RF-input?

The following subsections provide several options addressing the various questions above:

☞ In some applications, such as GSM transceivers, interference signals may exceed the maximum power rating of the RF_IN_1 input. To avoid device destruction use of external input protection is mandatory.

☞ During assembly of end-user devices which contain passive patch antennas, an ESD discharge may occur during production when pre-charged antennas are soldered to the GNSS receiver board. In such cases, use of external protection in front of RF_IN_1 is mandatory to avoid device destruction.

ESD discharge cannot be avoided during assembly and / or field use. Note also that SAW filters are susceptible to ESD damage. To provide additional robustness an ESD protection diode can be placed at RF_IN_1 to GND.

4.9.4 Antenna/ RF input

Check RF input requirements and schematic:

- An OEM active antenna module that meets our requirements should be used if there is a need to integrate the antenna.
- The total maximum noise figure including external LNA (or the LNA in the active antenna) should be around 2 dB.
- Ensure active antenna gain is ideally between 30 - 40 dB gain.
- Make sure the antenna is not placed close to noisy parts of the circuitry and not facing noisy parts (e.g. micro-controller or display).
- Signal levels above 40 C/No average are required for optimal RTK performance.
- Ensure antenna supports both L1 and L2 bands.
- Ensure antenna element gain is between 2 - 3 dBic typical for L1 and L2 bands.
- Ensure the group delay variation including active antenna is 10 ns max @ each GNSS system bandwidth. Note: Inter-signal requirement 50 ns max.
- ESD protection on the RF input is mandatory.
- Bias-t inductor must be L1 and L2 band frequency selected. 120 nH typical.
- Ensure RF trace is tuned for 50 Ω at 1575 MHz to ensure L1 and L2 bandwidth.

CONFIDENTIAL

4.9.5 Ground pads

Ensure the ground pads under the module are connected to ground.

If a patch type antenna is used, an antenna ground plane with minimum 100 - 150 mm diameter is required.

4.9.6 Schematic design

For a **minimal design** with the ZED-F9K GNSS modules, the following functions and pins need to be considered:

- Connect the power supply to VCC and V_BCKP.
- V_USB: If USB is used connect V_USB to receiver VCC.
- If USB is not used connect V_USB to ground.
- Ensure an optimal ground connection to all ground pins of the ZED-F9K GNSS modules.
- Choose the required serial communication interfaces (UART, USB, SPI or DDC) and connect the appropriate pins to your application.
- If you need hot or warm start in your application, connect a backup battery to V_BCKP.
- Antenna bias is required, see ZED-F9K antenna bias section.

4.9.7 Layout design-in guideline

- Is the ZED-F9K placed away from any heat source?
- Is the ZED-F9K placed away from any air cooling source?
- Is the ZED-F9K shielded by a cover/case to prevent the effects of air currents and rapid environmental temperature changes?
- Is the ZED-F9K placed as recommended in the Layout and Layout Guidance section?
- Assure a low serial resistance on the VCC power supply line (choose a line width > 400 um).
- Keep the power supply line as short as possible.
- Add a ground plane underneath the GNSS module to reduce interference. This is especially important for the RF input line.
- For improved shielding, add as many vias as possible around the micro strip/coplanar waveguide, around the serial communication lines, underneath the GNSS module, etc.

CONFIDENTIAL

5 Product handling

5.1 ESD handling precautions

⚠ ZED-F9K high precision receivers contain highly sensitive electronic circuitry and are Electrostatic Sensitive Devices (ESD). Observe precautions for handling! Failure to observe these precautions can result in severe damage to the GNSS receiver!

- Unless there is a galvanic coupling between the local GND (i.e. the work table) and the PCB GND, then the first point of contact when handling the PCB must always be between the local GND and PCB GND.
- Before mounting an antenna patch, connect ground of the device.

- When handling the RF pin, do not come into contact with any charged capacitors and be careful when contacting materials that can develop charges (e.g. patch antenna ~10 pF, coax cable ~50-80 pF/m or soldering iron).

- To prevent electrostatic discharge through the RF input, do not touch any exposed antenna area. If there is any risk that such exposed antenna area is touched in non ESD protected work area, implement proper ESD protection measures in the design.

- When soldering RF connectors and patch antennas to the receiver's RF pin, make sure to use an ESD safe soldering iron (tip).

5.2 Soldering

Soldering paste

Use of "no clean" soldering paste is highly recommended, as it does not require cleaning after the soldering process has taken place. The paste listed in the example below meets these criteria.

- Soldering paste: OM338 SAC405 / Nr.143714 (Cookson Electronics)
- Alloy specification: Sn 95.5/ Ag 4/ Cu 0.5 (95.5% Tin/ 4% Silver/ 0.5% Copper)
- Melting temperature: 217 °C
- Stencil thickness: The exact geometry, distances, stencil thicknesses and solder paste volumes must be adapted to the specific production processes (e.g. soldering) of the customer.

Reflow soldering

A convection type-soldering oven is highly recommended over the infrared type radiation oven. Convection heated ovens allow precise control of the temperature, and all parts will heat up evenly, regardless of material properties, thickness of components and surface color.

CONFIDENTIAL

As a reference, see the "IPC-7530 Guidelines for temperature profiling for mass soldering (reflow and wave) processes", published in 2001.

Preheat phase

During the initial heating of component leads and balls, residual humidity will be dried out. Note that this preheat phase will not replace prior baking procedures.

- Temperature rise rate: max. 3 °C/s. If the temperature rise is too rapid in the preheat phase it may cause excessive slumping.
- Time: 60 – 120 s. If the preheat is insufficient, rather large solder balls tend to be generated. Conversely, if performed excessively, fine balls and large balls will be generated in clusters.
- End temperature: 150 – 200 °C. If the temperature is too low, non-melting tends to be caused in areas containing large heat capacity.

Heating - reflow phase

The temperature rises above the liquidus temperature of 217°C. Avoid a sudden rise in temperature as the slump of the paste could become worse.

- Limit time above 217 °C liquidus temperature: 40 – 60 s
- Peak reflow temperature: 245 °C

Cooling phase

A controlled cooling avoids negative metallurgical effects (solder becomes more brittle) of the solder and possible mechanical tensions in the products. Controlled cooling helps to achieve bright solder fillets with a good shape and low contact angle.

- Temperature fall rate: max 4 °C/s

☞ To avoid falling off, the modules should be placed on the topside of the motherboard during soldering

The final soldering temperature chosen at the factory depends on additional external factors like choice of soldering paste, size, thickness and properties of the base board, etc. Exceeding the maximum soldering temperature in the recommended soldering profile may permanently damage the module.

Figure 49: Soldering Profile

☞ Modules **must not** be soldered with a damp heat process.

Optical inspection

After soldering the module, consider an optical inspection step.

Cleaning

☞ No cleaning with water, solvent, ultrasonic cleaner should be carried out:

- Cleaning with water will lead to capillary effects where water is absorbed in the gap between the baseboard and the module. The combination of residues of soldering flux and encapsulated water leads to short circuits or resistor-like interconnections between neighboring pads.
- Cleaning with alcohol or other organic solvents can result in soldering flux residues flooding into the two housings, areas that are not accessible for post-wash inspections. The solvent will also damage the sticker and the ink-jet printed text.
- Ultrasonic cleaning will permanently damage the module, in particular the quartz oscillators.

The best approach is to use a "No Clean" soldering paste and eliminate the cleaning step after the soldering.

Repeated reflow soldering

Only single reflow soldering processes are recommended for boards populated with modules. Modules should not be submitted to two reflow cycles on a board populated with components on both sides in order to avoid upside down orientation during the second reflow cycle. In this case, the module should always be placed on that side of the board, which is submitted into the last reflow cycle. The reason for this (besides others) is the risk of the module falling off due to the significantly higher weight in relation to other components.

Two reflow cycles can be considered by excluding the above described upside down scenario and taking into account the rework conditions described in this section.

☞ Repeated reflow soldering processes and soldering the module upside down are not recommended.

Wave soldering

Base boards with combined through-hole technology (THT) components and surface-mount technology (SMT) devices require wave soldering to solder the THT components. Only a single wave soldering process is encouraged for boards populated with modules.

Rework

We do not recommend using a hot air gun because this is an uncontrolled process and might damage the module.

⚠ Use of a hot air gun can lead to overheating and severely damage the module. Always avoid overheating the module.

After the module is removed, clean the pads before re-applying solder paste, placing and reflow soldering a new module.

⚠ Never attempt a rework on the module itself, e.g. replacing individual components. Such actions immediately terminate the warranty.

Conformal coating

Certain applications employ a conformal coating of the PCB using HumiSeal® or other related coating products. These materials affect the RF properties of the GNSS module and it is important

CONFIDENTIAL

to prevent them from flowing into the module. The RF shields do not provide 100% protection for the module from coating liquids with low viscosity; therefore, care is required in applying the coating.

☞ Conformal coating of the module will void the warranty.

Casting

If casting is required, use viscose or another type of silicon pottant. The OEM is strongly advised to qualify such processes in combination with the module before implementing this in the production.

☞ Casting will void the warranty.

Grounding metal covers

Attempts to improve grounding by soldering ground cables, wick or other forms of metal strips directly onto the EMI covers is done at the customer's own risk. The numerous ground pins should be sufficient to provide optimum immunity to interferences and noise.

☞ u-blox makes no warranty for damages to the module caused by soldering metal cables or any other forms of metal strips directly onto the EMI covers.

Use of ultrasonic processes

Some components on the module are sensitive to ultrasonic waves. Use of any ultrasonic processes (cleaning, welding etc.) may cause damage to the GNSS receiver.

☞ u-blox offers no warranty against damages to the module caused by any Ultrasonic Processes.

5.3 Tapes

The Figure 50 shows the feed direction and illustrates the orientation of the ZED-F9K high precision receivers on the tape: (F9P reel shown for illustrative purposes).

Figure 50: Orientation of ZED-F9K high precision receivers on the tape

The dimensions of the tapes for ZED-F9K high precision receivers are specified in Figure 51 (measurements in mm).

	DIM	±
Ao	17.80	0.2
Bo	22.90	0.2
Ko	3.10	0.2

Figure 51: ZED-F9K tape dimensions (mm)

5.4 Reels

The ZED-F9K high precision receiver GNSS modules are deliverable in quantities of 250 pieces on a reel. The ZED-F9K high precision receiver receivers are shipped on Reel Type B, as specified in the u-blox Package Information Guide. See the u-blox Package Information Guide [3].

5.5 Moisture sensitivity levels

The Moisture sensitivity level (MSL) for ZED-F9K high precision receivers is specified in the table below.

Package	MSL level
LGA	4

Table 22: MSL level

☞ For MSL standard see IPC/JEDEC J-STD-020, which can be downloaded from www.jedec.org.

☞ For more information regarding moisture sensitivity levels, labeling, storage and drying, see the u-blox Package Information Guide [3].

CONFIDENTIAL

Appendix

A Glossary

Abbreviation	Definition
ADR	Automotive Dead Reckoning
ANSI	American National Standards Institute
ARP	Antenna Reference Point
BeiDou	Chinese navigation satellite system
BBR	Battery Backed RAM
CDMA	Code Division Multiple Access
DDC	Display Data Channel
ECDSA	Elliptic Curve Digital Signature Algorithm
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EOS	Electrical Overstress
EPA	Electrostatic Protective Area
ESD	Electrostatic Discharge
Galileo	European navigation satellite system
GLONASS	Russian navigation satellite system
GND	Ground
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSM	Global System for Mobile Communications
IEC	International Electrotechnical Commission
ITRF	International Terrestrial Reference Frame
MSM	Multiple Signal Messages
NTRIP	Networked transport of RTCM via internet protocol
PCB	Printed Circuit Board
RTCM	Radio Technical Commission for Maritime
RTK	Real Time Kinematic
SV	Space Vehicle, a satellite
TDOP	Time Dilution Of Precision
UBX	u-blox
UDR	Untethered Dead Reckoning
QZSS	Quasi-Zenith Satellite System

B RTCM ITRF Geodetic models

RTK is a differential system where the receiver uses the reference datum of the reference station. The International Terrestrial Reference Frame (ITRF) must be obtained from the reference network and then used to transform the receiver position output to match the required reference frame. The user application will need to do the transformation for use in a mapping application if it does not use the same reference frame. An offset can occur if this is not done.

The list of ITRF reference frame years are below:

CONFIDENTIAL

- ITRF94
- ITRF96
- ITRF97
- ITRF2000
- ITRF2005
- ITRF2008
- ITRF2014

There are other similar reference systems used by GNSS correction services such as the ETRS89.

For example the EUREF network uses the ETRS89 reference frame and this info can be found on the homepage: EUREF.

See the ITRF website for information and an online transform calculator: ITRF.

Another online tool can be found here: Transformation.

Figure 52: Transforming between two datums

The user may need to transform the receiver output position data to match the datum being used by their application and mapping system. If comparing the receiver position data with a reference system, the same RTCM stream should be used to ensure that the reference and receiver are using the same reference-frame/datum. If doing post processing of a reference system its output must be transformed to same ITRF and datum being used by the receiver.

For example, when viewing RTK accurate positions on Google Earth: The heights on Google Earth refer to EGM96 and are, therefore, Geoidal heights.

The lat/long are referred to the WGS 84 (ITRF2008) ellipsoid. The result is a visible inaccuracy, the RTK receivers position output will need to be transformed to the Google Earth transform system before it can be realistically used. This is ignoring any local map tile inaccuracies.

Geodetic Coordinate Systems and Ellipsoids:

In practice, the relevant organizations choose to keep their respective frames very close to the International Terrestrial Reference Frame (ITRF), defined and managed by the International Earth Rotation and Reference Systems Service (IERS).

However, because the Earth's tectonic plates and even parts of the Earth's core move, new versions of ITRF are defined every few years, generally with changes of the order of a few millimeters.

Consequently, the major GNSS occasionally decide that they need to update their reference frames to be better aligned to the latest ITRF.

So, for example, GPS switched to WGS84 (G1150) in GPS week 1150 (early 2002) based on ITRF2000, while GLONASS switched from PZ90.02 to PZ90.11 at the end of 2013, based on ITRF2008. The net effect of this, is that all the major GNSS use almost the same reference frame, but there are some small (generally sub-cm) differences between them and these differences occasionally change.

In order to produce positions that can be shown on a map, it is necessary to translate between raw coordinates (e.g. x, y, z) and a position relative to the Earth's surface (e.g. latitude, longitude and altitude) and that requires defining the form of ellipsoid that best matches the shape of the Earth.

Historically many different ellipsoid definitions have been used for maps, many of which predate the existence of GNSS and show quite significant differences, leading to discrepancies of as much as 100m in places. Fortunately, most digital maps now use the WGS84 ellipsoid, which is distinct from the WGS84 coordinate system, but defined by the same body.

C Virtual reference service

Reaching decimeter-level accuracy of positioning typically requires use of precise dual-frequency carrier phase observations. Furthermore, these observations are usually processed using a differential GNSS (DGNSS) algorithm, such as real time kinematic (RTK) or post-processing (PP).

Regardless of the specific differential algorithm, however, implicit in the process is an assumption that the quality of the reference station data is consistent with the desired level of positioning accuracy. The virtual reference service (VRS) concept can help to satisfy this requirement using a network of reference stations.

In contrast, GNSS reference network architectures often make use of multiple reference stations. This approach allows a more precise modeling of distance-dependent systematic errors principally caused by ionospheric and tropospheric refractions, and satellite orbit errors. More specifically, a GNSS reference network decreases the dependence of the error budget on the distance of nearest antenna.

The picture below shows the network of reference stations used to build up a service covering the whole of Switzerland.

CONFIDENTIAL

Figure 53: Virtual reference service network in Switzerland

The network of receivers is linked to a computation center, and each station contributes its raw data to help create network-wide models of the distance-dependent errors. The computation of errors based on the full network's carrier phase measurements involves, first of all, the resolution of carrier phase ambiguities and requires knowledge of the reference station positions.

The latter is usually determined as part of the network setup.

This generated a virtual reference service data is then sent to the user through a wireless connection, often using the Networked Transport of RTCM via Internet Protocol (NTRIP). Finally, just as if a virtual reference service data had come from a physical reference station, the receiver uses standard single-baseline algorithms to determine the coordinates of the user's receiver, in real-time kinematic or post-processed modes.

The receiver must send its own actual position back to the correction server, typically using the NMEA GGA message. Usually if a virtual reference service system does not receive a GGA message it will not provide the RTCM data to the receiver. The virtual reference service concept allows a less dense reference network without accuracy degradation because the multiple reference station network is a better model of the spatially correlated GNSS errors over longer baselines.

As a result the maximum distance between the receiver and the nearest reference station can be extended beyond the typical 10 ~15 kilometers without accuracy degradation of the single reference station case. Another benefit of a virtual reference service is that the reference data are free of site-specific errors such as multipath, because a virtual reference service computation assumes that the virtual station is situated at an ideal location.

D RTK configuration procedures with u-center

This section provides some guidance for using u-center to evaluate ZED-F9K RTK operation

D.1 Receiver configuration with u-center

In this example, UART1 is configured to use a suitable baud-rate for communicating with a host PC running u-center. A set of output messages are enabled so that the receiver status can be monitored.

Using the VALSET configuration window set the UART1 interface for the correct host baud rate:

Select Group: CFG-UART1, Key name: CFG-UART1-BAUDRATE. See Figure 54.

Figure 54: u-center UBX-CFG-VALSET message view

Add it to the list and highlight it. It will now give the option of setting or reading the current value. See Figure 55.

CONFIDENTIAL

Figure 55: Example u-center UBX-CFG-VALSET message view when selecting a configuration item

Next add the value, e.g. 230400 into the Value window that appears below the list. See Figure 56.

Then set the configuration by clicking the Send button at the bottom of the message tree view. Remember to set the u-center baud rate to match the value set in the receiver.

CONFIDENTIAL

Figure 56: u-center UBX-CFG-VALSET message view for setting the CFG-UART1-BAUDRATE configuration item that controls the baudrate of UART1

Next, some UBX example messages are configured to enable viewing the receiver status.

Select Group CFG-MSGOUT, Key name: CFG-MSGOUT-UBX and select the UART1 required messages. Add each message to the list, setting the value of each to 1, then click Send. See Figure 57.

CONFIDENTIAL

Figure 57: u-center UBX-CFG-VALSET message view for setting the CFG-MSGOUT-* configuration items for enabling the output of some recommended UBX messages

☞ UBX-NAV-SOL is shown, however it is not supported on this platform any longer.

To ensure all the required RTCM messages, including most importantly RTCM 1005 are being received regularly, examine the UBX-RXM-RTCM output in u-center. See Figure 58.

CONFIDENTIAL

UBX - RXM (Receiver Manager) - RTCM (RTCM input status)

Statistics:

Message Type	Total messages	CRC passed messages	CRC failed messages	(Last) Reference Station ID
0	0	0	0	n/a
1002	0	0	0	n/a
1005	33	33	0	0
1009	0	0	0	n/a
1010	0	0	0	n/a
1074	0	0	0	n/a
1077	33	33	0	0
1084	0	0	0	n/a
1087	33	33	0	0
1097	33	33	0	0
1124	0	0	0	n/a
1127	33	33	0	0
1230	33	33	0	0
1001	0	0	0	n/a

* Messages not supported by the receiver

Current messages:

ix	Reference Station ID	Message ID	CRC check
198	0	1005	Passed
197	0	1230	Passed
196	0	1127	Passed
195	0	1097	Passed
194	0	1087	Passed
193	0	1077	Passed
192	0	1005	Passed
191	0	1230	Passed
190	0	1127	Passed
189	0	1097	Passed
188	0	1087	Passed
187	0	1077	Passed
186	0	1005	Passed
185	0	1230	Passed
184	0	1127	Passed

Figure 58: u-center UBX-RXM-RTCM view

Once the receiver is receiving a valid set of RTCM messages, it should transition through 3D Fix to 3D/DGNSS to Float and ultimately into Fixed mode. See Figure 59.

Figure 59: u-center data view with RTK Fixed

☞ If using a virtual reference service the receiver must be configured to output NMEA GGA messages to the u-center NTRIP client (UART1 in this example).

Related documents

[1] ZED-F9K Data Sheet, Docu. No. UBX-17061422
[2] ZED-F9K Interface Description, Docu. No. UBX-19000811
[3] u-blox Package Information Guide, Docu. No. UBX-14001652

☞ For regular updates to u-blox documentation and to receive product change notifications please register on our homepage (http://www.u-blox.com).

CONFIDENTIAL

Revision history

Revision	Date	Name	Status / Comments
R01	15-Mar-2019	ssid	Objective Specification
R02	06-May-2019	jhak	Objective Specification Footprint and paste mask images added to section Layout.

CONFIDENTIAL

Contact

For complete contact information visit us at www.u-blox.com.

u-blox Offices

North, Central and South America

u-blox America, Inc.
Phone: +1 703 483 3180
E-mail: info_us@u-blox.com

Regional Office West Coast
Phone: +1 408 573 3640
E-mail: info_us@u-blox.com

Technical Support
Phone: +1 703 483 3185
E-mail: support_us@u-blox.com

Headquarters

Europe, Middle East, Africa

u-blox AG
Phone: +41 44 722 74 44
E-mail: info@u-blox.com
Support: support@u-blox.com

Asia, Australia, Pacific

u-blox Singapore Pte. Ltd.
Phone: +65 6734 3811
E-mail: info_ap@u-blox.com
Support: support_ap@u-blox.com

Regional Office Australia
Phone: +61 2 8448 2016
E-mail: info_anz@u-blox.com
Support: support_ap@u-blox.com

Regional Office China (Beijing)
Phone: +86 10 68 133 545
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Chongqing)
Phone: +86 23 6815 1588
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Shanghai)
Phone: +86 21 6090 4832
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office China (Shenzhen)
Phone: +86 755 8627 1083
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office India
Phone: +91 80 4050 9200
E-mail: info_in@u-blox.com
Support: support_in@u-blox.com

Regional Office Japan (Osaka)
Phone: +81 6 6941 3660
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Japan (Tokyo)
Phone: +81 3 5775 3850
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Korea
Phone: +82 2 542 0861
E-mail: info_kr@u-blox.com
Support: support_kr@u-blox.com

Regional Office Taiwan
Phone: +886 2 2657 1090
E-mail: info_tw@u-blox.com
Support: support_tw@u-blox.com

CONFIDENTIAL