P

NEO-M9N

Standard precision GNSS module

Integration manual

Abstract

Integration manual describing the u-blox NEO-M9N GNSS module. NEO-M9N offers ultra-robust meter-level GNSS positioning performance with concurrent reception of up to four GNSS (GPS, GLONASS, BeiDou, Galileo) in a 12.2 x 16.0 mm package.

u-blox

Document information

Title	**NEO-M9N**
Subtitle	Standard precision GNSS module
Document type	Integration manual
Document number	UBX-19014286
Revision and date	R09 09-Feb-2024
Disclosure restriction	C1-Public

This document applies to the following products:

Type number	Firmware version	IN/PCN reference	RN reference
NEO-M9N-00B-00	SPG 4.04	UBX-21029274 UBX-22039049	UBX-20036165

Contents

1 Integration manual overview

This document is an important source of information on all aspects of NEO-M9N system, software and hardware design. The purpose of this document is to provide guidelines for a successful integration of the receiver with the customer's end product.

2 System description

2.1 Overview

The NEO-M9N GNSS receiver features the u-blox M9 standard precision GNSS platform . It provides exceptional sensitivity and acquisition times for all L1 GNSS systems. u-blox M9 receivers are available in different variants to serve automotive and industrial tracking applications, such as navigation, telematics and UAVs.

u-blox M9 receivers support concurrent reception of four GNSS. The high number of visible satellites allows the receiver to select the best signals. This maximizes the position accuracy, in particular under challenging conditions such as deep urban canyons.

u-blox M9 receivers detect jamming and spoofing events and report them to the host, which allows the system to react to such events. Advanced filtering algorithms mitigate the impact of RF interference and jamming, thus enabling the product to operate as intended.

The receiver also provides higher navigation rate and improved security features compared to previous u-blox GNSS generations.

The NEO-M9N module is available in the NEO form factor, which is a 12.2 x 16.0 mm LCC package.

2.2 Architecture

The NEO-M9N receiver provides all the necessary RF and baseband processing to enable multi-constellation operation. The block diagram below shows the key functionality.

2.2.1 Block diagram
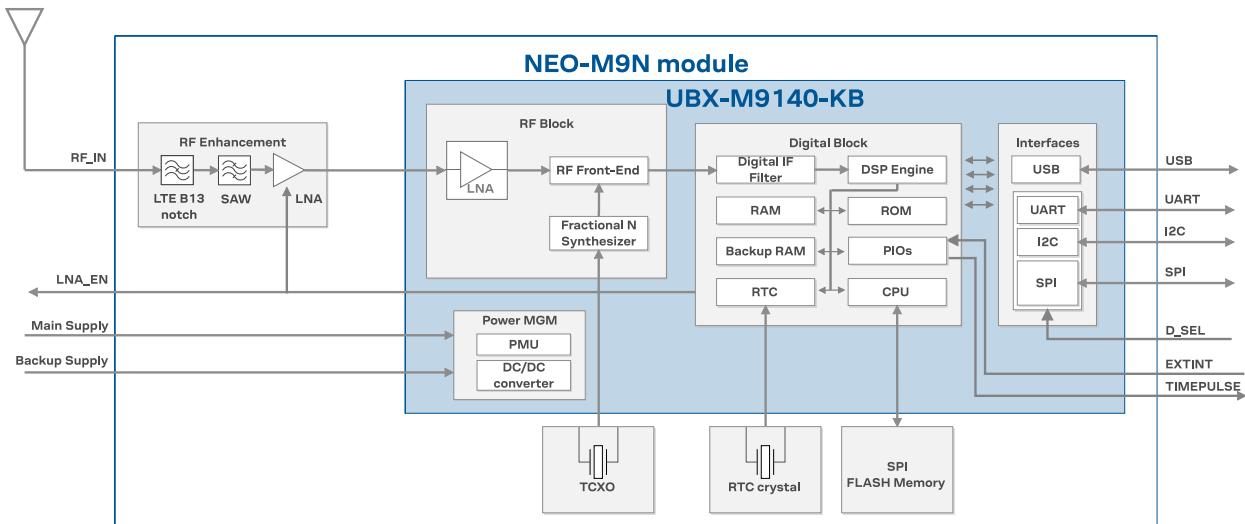
Figure 1: NEO-M9N block diagram

3 Receiver functionality

This chapter describes the NEO-M9N operational features and their configuration.

3.1 Receiver configuration

NEO-M9N 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 startup of the receiver from several sources of configuration. The configuration interface and the available keys are described fully in the applicable 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.

⚠ **CAUTION** The configuration interface has changed from earlier u-blox positioning receivers. Legacy messages have been deprecated and will not be supported in future firmware releases. Users are advised to adopt the configuration interface described in this document. See legacy UBX-CFG message fields reference section in the applicable Interface description [2].

Configuration interface settings are held in a database consisting of separate configuration items. An item is made up of a pair consisting of a key ID and a value. Related items are grouped together and identified under a common group name: CFG-GROUP-*; 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 UBX-CFG-VALSET, UBX-CFG-VALGET, and UBX-CFG-VALDEL. For more information about these messages and the configuration keys, see the configuration interface section in the applicable Interface description [2].

3.1.1 Changing the receiver configuration

All configuration messages, including legacy UBX-CFG messages, will result in a UBX-ACK-ACK or UBX-ACK-NAK response. If several configuration messages are sent without waiting for this response then the receiver may pause processing of input messages until processing of a previous configuration message has been completed. When this happens a warning message "wait for cfg ACK" will be sent to the host.

3.1.2 Default GNSS configuration

The NEO-M9N default GNSS configuration is set as follows:

- GPS: L1C/A
- GLONASS: L1OF
- Galileo: E1B/C
- BeiDou: B1I
- QZSS: L1C/A
- SBAS: L1C/A

For more information about the default configuration, see the applicable Interface description [2].

3.1.3 Default interface settings

Interface	Settings
UART	38400 baud, 8 bits, no parity bit, 1 stop bit.
	Output messages: NMEA **GGA, GLL, GSA, GSV, RMC, VTG, TXT** (no UBX).
	Input protocols: UBX, NMEA and RTCM 3.3.
USB	Output messages activated as in UART. Input protocols available as in UART.
I2C	Output messages activated as in UART. Input protocols available as in UART.
SPI	Output messages activated as in UART. Input protocols available as in UART.

Table 1: Default interface settings

☞ Refer to the applicable interface description [2] for information about further settings.

By default the NEO-M9N outputs NMEA 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 UART 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 Basic hardware configuration

The NEO-M9N receiver has been configured with default setting already during the module production. The receiver starts up and is fully operational as soon as proper power supply, communication interfaces and antenna signal from the application have been connected.

3.1.4.2 Communication interface configuration

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

Interface	Configuration groups
UART1	CFG-UART1-*, CFG-UART1INPROT-*, CFG-UART1OUTPROT-*
USB	CFG-USB-*, CFG-USBINPROT-*, CFG-USBOUTPROT-*
I2C	CFG-I2C-*, CFG-I2CINPROT-*, CFG-I2COUTPROT-*
SPI	CFG-SPI-*, CFG-SPIINPROT-*, CFG-SPIOUTPROT-*

Table 2: Interface configurations

3.1.4.3 Message output configuration

This product supports two protocols for output messages. One is NMEA and the other one is a u-blox proprietary "UBX" protocol. NMEA is a well-known industry standard, used mainly for providing information about position, time and satellites. UBX messages can be used to configure the receiver and also to periodically provide information about position, time and satellites. With the UBX protocol it is easy to monitor the receiver status and get much deeper information about the receiver status. The rate of NMEA and UBX protocol output messages are configurable and it is possible to enable or disable single NMEA or UBX messages individually.

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.

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 non-periodic and will only be output as an answer to a poll request.

The UBX-INF-* and NMEA-Standard-TXT 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 Communication interface configuration).

3.1.4.4 GNSS signal configuration

The GNSS constellations are configurable with configuration keys. Each GNSS constellation can be enabled or disabled independently.

3.1.5 RTCM corrections

RTCM is a standard-based binary protocol for the communication of GNSS correction information. The NEO-M9N module 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.

To modify the RTCM input/output settings, see the configuration section in the applicable Interface description [2].

3.1.5.1 NTRIP - networked transport of RTCM via internet protocol

Networked Transport of RTCM via internet protocol, or NTRIP, is an open standard protocol for streaming differential data and other kinds of GNSS streaming 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 an on-board computer running NTRIP server software communicating directly with a GNSS reference station. The NTRIP server serves as the intermediary between the GNSS receiver (NTRIP Source) streaming correction data and the NTRIP caster.
2. The NTRIP caster is an HTTP server which receives streaming correction data from one or more NTRIP servers and in turn streams the correction data to one or more NTRIP clients via the internet.
3. The NTRIP client receives streaming correction 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 NEO-M9N 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 NEO-M9N receiver connected to the local

u-center application. RTCM corrections from a virtual reference service are also supported by the u-center NTRIP client.

3.1.6 Legacy configuration interface compatibility

Although there is some backwards compatibility for the legacy UBX-CFG configuration messages, users are strongly advised to adopt the configuration interface described in this document.

See Legacy UBX-CFG message fields reference section in the applicable Interface description [2].

3.1.7 Navigation configuration

This section presents various configuration options related to the navigation engine. These options can be configured through CFG-NAVSPG-* configuration keys.

3.1.7.1 Platform settings

u-blox receivers support different dynamic platform models (see the table below) to adjust the navigation engine to the expected application environment. These platform settings can be changed dynamically without performing a power cycle or reset. The settings improve the receiver's interpretation of the measurements and thus provide a more accurate position output. Setting the receiver to an unsuitable platform model for the given application environment is likely to result in a loss of receiver performance and position accuracy.

The dynamic platform model can be configured through the CFG-NAVSPG-DYNMODEL configuration item. The supported dynamic platform models and their details can be seen in Table 3 and Table 4 below.

Platform	Description
Portable (default)	Applications with low acceleration, e.g. portable devices. Suitable for most situations.
Stationary	Used in timing applications (antenna must be stationary) or other stationary applications. Velocity restricted to 0 m/s. Zero dynamics assumed.
Pedestrian	Applications with low acceleration and speed, e.g. how a pedestrian would move. Low acceleration assumed.
Automotive	Used for applications with equivalent dynamics to those of a passenger car. Low vertical acceleration assumed.
At sea	Recommended for applications at sea, with zero vertical velocity. Zero vertical velocity assumed. Sea level assumed.
Airborne <1g	Used for applications with a higher dynamic range and greater vertical acceleration than a passenger car. No 2D position fixes supported.
Airborne <2g	Recommended for typical airborne environments. No 2D position fixes supported.
Airborne <4g	Only recommended for extremely dynamic environments. No 2D position fixes supported.
Wrist	Only recommended for wrist-worn applications. Receiver will filter out arm motion.

Table 3: Dynamic platform models

Platform	Max altitude [m]	Max horizontal velocity [m/s]	Max vertical velocity [m/s]	Sanity check type	Max position deviation
Portable	12000	310	50	Altitude and velocity	Medium
Stationary	9000	10	6	Altitude and velocity	Small
Pedestrian	9000	30	20	Altitude and velocity	Small
Automotive	6000	100	15	Altitude and velocity	Medium
At sea	500	25	5	Altitude and velocity	Medium
Airborne <1g	80000	100	6400	Altitude	Large

Platform	Max altitude [m]	Max horizontal velocity [m/s]	Max vertical velocity [m/s]	Sanity check type	Max position deviation
Airborne <2g	80000	250	10000	Altitude	Large
Airborne <4g	80000	500	20000	Altitude	Large
Wrist	9000	30	20	Altitude and velocity	Medium

Table 4: Dynamic platform model details

Applying dynamic platform models designed for high acceleration systems (e.g. airborne <2g) can result in a higher standard deviation in the reported position.

If a sanity check against a limit of the dynamic platform model fails, then the position solution is invalidated. Table 4 above shows the types of sanity checks which are applied for a particular dynamic platform model.

3.1.7.2 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

If the receiver only has three satellites for calculating a position, the navigation algorithm uses a constant altitude to compensate for the missing fourth satellite. When a satellite is lost after a successful 3D fix (min four satellites 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 satellites.

3.1.7.3 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).

☞ The NEO-M9N firmware does not support the dead reckoning position fix type.

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 be 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 be used.
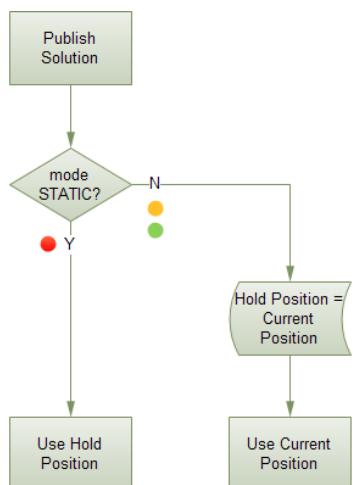
UBX-NAV-STATUS message also reports whether a fix is valid in the gpsFixOK flag. This message has only been retained for backwards compatibility and users are recommended to use the UBX-NAV-PVT message.

3.1.7.4 Static hold

Static hold mode allows the navigation algorithms to decrease the noise in the position output when the velocity is below a predefined "Static Hold Threshold". This reduces the position wander caused by environmental factors such as multi-path and improves position accuracy especially in stationary applications. By default, static hold mode is disabled.

If the speed drops below the defined "Static Hold Threshold", the static hold mode will be activated. Once static hold mode has been entered, the position output is kept static and the velocity is set to zero until there is evidence of movement again. Such evidence can be velocity, acceleration, changes of the valid flag (e.g. position accuracy estimate exceeding the position accuracy mask, see also section Navigation output filters), position displacement, etc.

The CFG-MOT-GNSSDIST_THRS configuration item additionally allows for configuration of distance threshold. If the estimated position is farther away from the static hold position than this threshold, static mode will be quit. The CFG-MOT-GNSSSPEED_THRS configuration item allows you to set a speed that the static hold will release.

Figure 2: Position publication in static hold mode
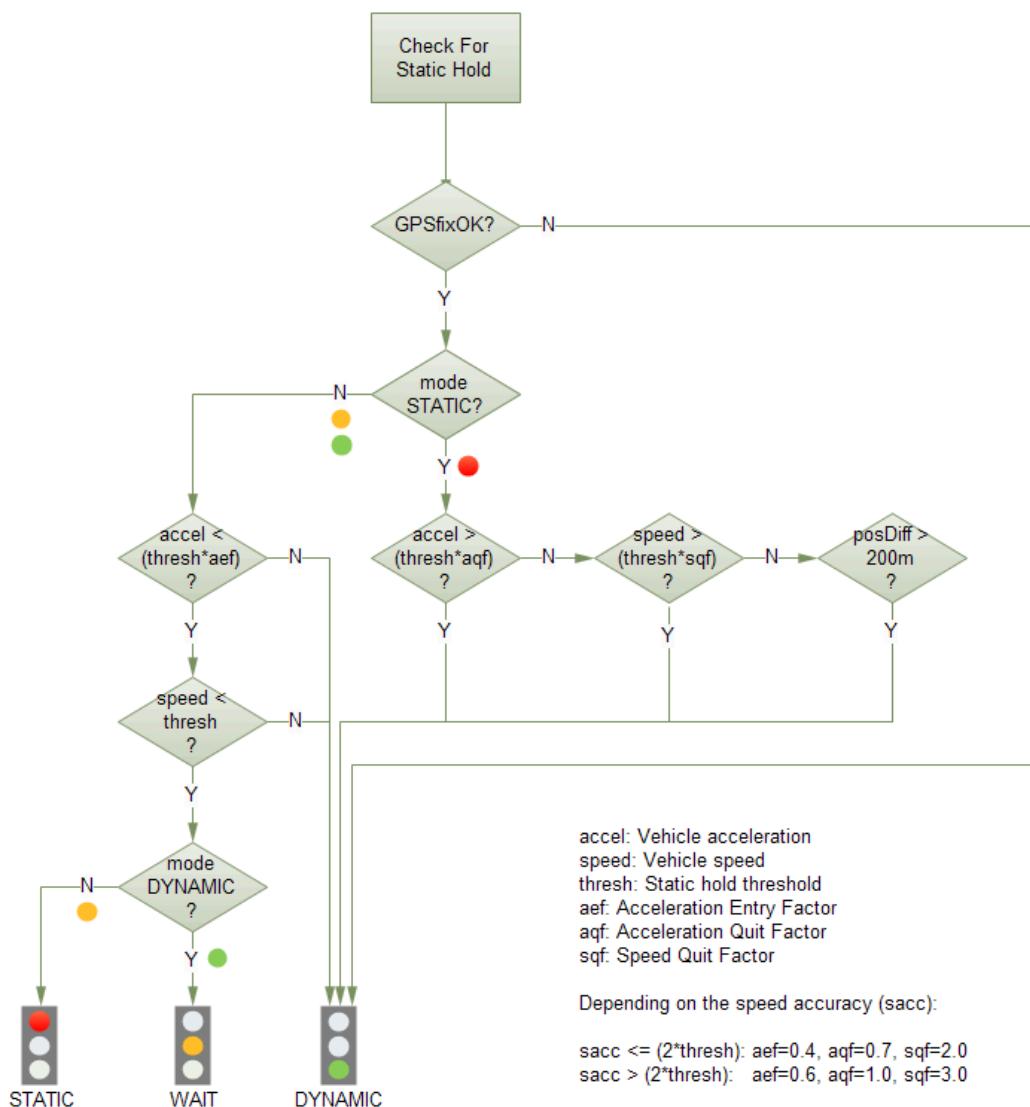
Figure 3: Flowchart of the static hold mode

3.1.7.5 Freezing the course over ground

The receiver derives the course over ground from the GNSS velocity information. If the velocity cannot be calculated with sufficient accuracy (e.g., with bad signals) or if the absolute speed value is very low (under 0.1 m/s) then the course over ground value becomes inaccurate too. In this case the course over ground value is frozen, i.e. the previous value is kept and its accuracy is degraded over time. These frozen values will not be output in the NMEA messages NMEA-RMC and NMEA-VTG unless the NMEA protocol is explicitly configured to do so (see NMEA protocol configuration in the applicable Interface description [2]).
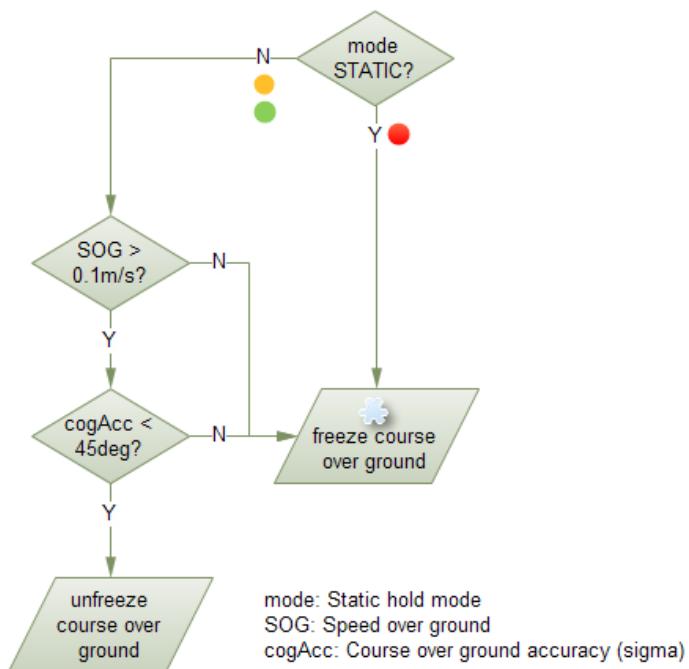
Figure 4: Flowchart of the course over ground freezing

3.1.7.6 Weak signal compensation

In normal operating conditions, low signal strength (i.e. signal attenuation) indicates likely contamination by multi-path. The receiver trusts such signals less in order to preserve the quality of the position solution in poor signal environments. This feature can result in degraded performance in situations where the signals are attenuated for another reason, for example due to antenna placement. In this case, the weak signal compensation feature can be used to restore normal performance. There are three possible modes:

- Disabled: no weak signal compensation is performed
- Automatic: the receiver automatically estimates and compensates for the weak signal
- Configured: the receiver compensates for the weak signal based on a configured value

These modes can be selected using CFG-NAVSPG-SIGATTCOMP. In the case of the "configured" mode, the user should input the maximum C/N0 observed in a clear-sky environment, excluding any outliers or unusually high values. The configured value can have a large impact on the receiver performance, so it should be chosen carefully.

3.2 SBAS

The NEO-M9N is capable of receiving multiple SBAS signals concurrently, even from different SBAS systems (WAAS, EGNOS, MSAS, etc.). They can be tracked and used for navigation simultaneously. Every SBAS satellite that broadcasts ephemeris or almanac information can be used for navigation, just like a normal GNSS satellite.

For receiving correction data, the NEO-M9N automatically chooses the best SBAS satellite as its primary source. It will select only one since the information received from other SBAS satellites is redundant and could be inconsistent. The selection strategy is determined by the proximity of the satellites, the services offered by the satellite, the configuration of the receiver (test mode allowed/ disallowed, integrity enabled/disabled) and the signal link quality to the satellite.

If corrections are available from the chosen SBAS satellite and used in the navigation calculation, the differential status will be indicated in several output messages such as UBX-NAV-PVT, UBX-NAV-STATUS, UBX-NAV-SAT, NMEA-GGA, NMEA-GLL, NMEA-RMC and NMEA-GNS (see the applicable Interface description [2]). The message UBX-NAV-SBAS provides detailed information about which corrections are available and applied.

The most important SBAS feature for accuracy improvement is ionosphere correction. The measured data from regional Ranging and Integrity Monitoring Stations (RIMS) are combined to make a Total Electron Content (TEC) map. This map is transferred to the receiver via SBAS satellites to allow a correction of the ionosphere error on each received signal.

Message type	Message content	Source
0(0/2)	Test mode	All
1	PRN mask assignment	Primary
2, 3, 4, 5	Fast corrections	Primary
6	Integrity	Primary
7	Fast correction degradation	Primary
9	Satellite navigation (ephemeris)	All
10	Degradation	Primary
12	Time offset	Primary
17	Satellite almanac	All
18	Ionosphere grid point assignment	Primary
24	Mixed fast / long-term corrections	Primary
25	Long-term corrections	Primary
26	Ionosphere delays	Primary

Table 6: Supported SBAS messages

Each satellite services a specific region and its correction signal is only useful within that region. Planning is crucial to determine the best possible configuration, especially in areas where signals from different SBAS systems can be received:

- **Example 1 - SBAS receiver in North America:** In the eastern parts of North America, make sure that EGNOS satellites do not take preference over WAAS satellites. The satellite signals from the EGNOS system should be disallowed by using the PRN mask.
- **Example 2 - SBAS receiver in Europe:** Some WAAS satellite signals can be received in the western parts of Europe, therefore it is recommended that the satellites from all but the EGNOS system should be disallowed using the PRN mask.

☞ Although u-blox receivers try to select the best available SBAS correction data, it is recommended to configure them to exclude unwanted SBAS satellites.

To configure the SBAS functionalities use the CFG-SBAS-* configuration group.

Parameter	Description
CFG-SIGNAL-SBAS_ENA	Enabled/disabled status of the SBAS subsystem
CFG-SBAS-USE_TESTMODE	Allow/disallow SBAS usage from satellites in test mode
CFG-SBAS-USE_RANGING	Use the SBAS satellites for navigation (ranging)
CFG-SBAS-USE_DIFFCORR	Combined enable/disable switch for fast, long-term, and ionosphere corrections
CFG-SBAS-USE_INTEGRITY	Apply integrity information data
CFG-SBAS-PRNSCANMASK	Allows selectively enabling/disabling SBAS satellites

Table 7: SBAS configuration parameters

☞ When SBAS integrity data is applied, the navigation engine stops using all signals for which no integrity data are available (including all non-GPS signals). It is not recommended to enable SBAS integrity on borders of SBAS service regions in order not to inadvertently restrict the number of available signals.

☞ SBAS integrity information is required for at least 5 GPS satellites. If this condition is not met, SBAS integrity data will not be applied.

☞ SBAS is only used if no correction service is available. If the connection stream is lost during the operation, the receiver will switch to using the SBAS corrections after the time set in CFG-NAVSPG-CONSTR_DGNSSTO (60 s by default) has elapsed.

☞ When the receiver switches from a solution using correction data to a standard position solution, the reference frame of the output position will switch from the one of the correction data to that of the standard position solution. For an SBAS solution this reference frame will be aligned within a few cm of WGS84 (and modern ITRF realizations).

3.3 Geofencing

3.3.1 Introduction

● Geofence
× User position
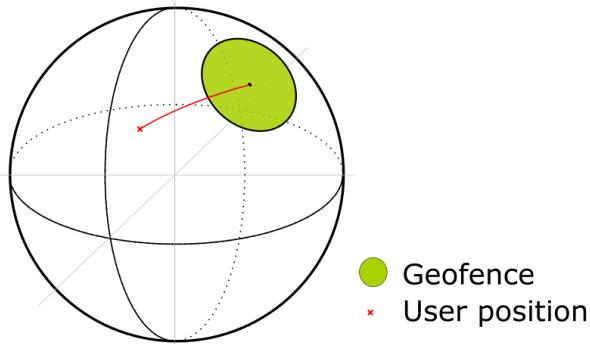
Figure 5: Geofence

The geofencing feature allows for the configuration of up to four circular areas (geofences) on the Earth's surface. The receiver will then evaluate for each of these areas whether the current position lies within the area or not and signal the state via UBX messaging and PIO toggling.

3.3.2 Interface

Geofencing can be configured using the CFG-GEOFENCE-* configuration group. The geofence evaluation is active whenever there is at least one geofence configured.
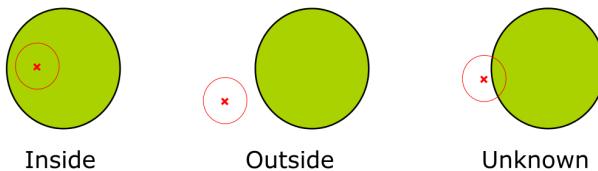
The current state of each geofence plus the combined state is output in UBX-NAV-GEOFENCE with every navigation epoch.

3.3.3 Geofence state evaluation

With every navigation epoch the receiver will evaluate the current solution's position versus the configured geofences. There are three possible outcomes for each geofence:

• *Inside* - The position is inside the geofence with the configured confidence level
• *Outside* - The position lies outside of the geofence with the configured confidence level
• *Unknown* - There is no valid position solution or the position uncertainty does not allow for unambiguous state evaluation

The position solution uncertainty (standard deviation) is multiplied with the configured confidence sigma level and taken into account when evaluating the geofence state (red circle in Figure 6).

Inside   Outside   Unknown

Figure 6: Geofence states

The combined state for all geofences is evaluated as the combination (Union) of all geofences:

- *Inside* - The position lies inside of at least one geofence
- *Outside* - The position lies outside of all geofences
- *Unknown* - All remaining states

3.3.4 Using a PIO for geofence state output

This feature can be used, for example, for waking up a sleeping host when a defined geofence condition is reached. The receiver will toggle the assigned pin according to the combined geofence state. Due to hardware restrictions, the geofence unknown state is not configurable and is always represented as HIGH. If the receiver is in the software backup mode or in the reset state, the pin will go to HIGH accordingly. The meaning of the LOW state can be configured using the CFG-GEOFENCE-PINPOL configuration item.

☞ The CFG-GEOFENCE-PIN configuration item refers to a PIO and not a physical device pin. The PIO number must be set so that it is mapped to the assigned geofence state device pin.

3.4 Logging

3.4.1 Introduction

The logging feature allows position fixes and arbitrary byte strings from the host to be logged in the receiver's flash memory. Logging of position fixes happens independently of the host system, and can continue while the host is powered down.

The following table lists all the logging-related messages:

Message	Description
UBX-LOG-CREATE	Creates a log file and activates the logging system
UBX-LOG-ERASE	Erases a log file and deactivates the logging subsystem
UBX-LOG-INFO	Provides information about the logging system
UBX-LOG-STRING	Enables a host process to write a string of bytes to the log file

Table 8: Logging control and configuration messages

Message	Description
UBX-LOG-RETRIEVE	Starts the log retrieval process
UBX-LOG-RETRIEVEPOS	A position log entry returned by the receiver
UBX-LOG-RETRIEVEPOSEXTRA	Odometer position data
UBX-LOG-RETRIEVESTRING	A byte string log entry returned by the receiver

Message	Description
UBX-LOG-FINDTIME	Finds the index of the first entry (given time)

Table 9: Logging retrieval messages

3.4.2 Setting the logging system up

An empty log can be created using the UBX-LOG-CREATE message and a log can be deleted with the UBX-LOG-ERASE message. The logging system will only run if a log is in existence, so most logging messages will be rejected with a UBX-ACK-NAK message if there is no log present. Only one log can be created at any one time so a UBX-ACK-NAK message will be returned if a log already exists. The message specifies the maximum size of the log in bytes (with some preset values provided). Both the logging subsystem and the receiver file-store have implementation overheads, so the total space available for log entries will be somewhat smaller than the size specified.

UBX-LOG-CREATE also allows the log to be specified as a circular log. If the log is circular, a set of older log entries will be deleted when it fills up, and the space freed up is used for new log entries. By contrast, if a non-circular log becomes full then new entries which do not fit will be rejected. UBX-LOG-CREATE also causes the logging system to start up so that further logging messages can be processed. The logging system will start up automatically on power-up if there is a log in existence. The log will remain in the receiver until specifically erased using the UBX-LOG-ERASE message.

The CFG-LOGFILTER-* configuration group controls whether logging of entries is currently enabled and selects position fix messages for logging.
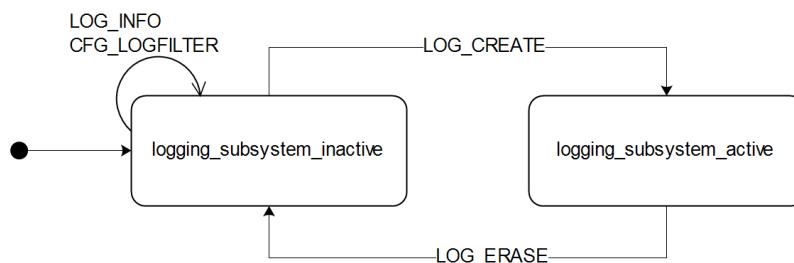
Figure 7: The top level active/inactive states of the logging subsystem

3.4.3 Information about the log

The receiver can be polled for a UBX-LOG-INFO message which will give information about the log. This will include the maximum size that the log can grow to (which, due to overheads, will be smaller than that requested in UBX-LOG-CREATE) and the amount of log space currently occupied. It will also report the number of entries currently in the log together with the time and date of the newest and oldest messages that have a valid time stamp.

Log entries are compressed and have housekeeping information associated with them, so the actual space occupied by log messages may be difficult to predict. The minimum size for a position fix entry is 9 bytes and the maximum 24 bytes, the typical size is 10 or 11 bytes. If the odometer is enabled then this will use at least another three bytes per fix.

Each log also has a fixed overhead which is dependent on the log type. The approximate size of this overhead is shown in the following table.

Log type	Overhead
circular	Up to 40 kB
non-circular	Up to 8 kB

Table 10: Log overhead size

The number of entries that can be logged in any given flash size can be estimated as follows:

Approx. number of entries = (flash size available for logging - log overhead)/typical entry size

For example, if 1500 kB of flash is available for logging (after other flash usage such as the firmware image is taken into account) a non-circular log would be able to contain approximately 139000 entries: ((1500*1024)- (8*1024))/11 = 138891.

3.4.4 Recording

The CFG-LOGFILTER-RECORD_ENA configuration item must be set to *true* to enable recording into the log. Nothing will be recorded if recording is disabled, otherwise position fix and UBX-LOG_STRING entries can be recorded. When recording is enabled an entry will also be created from each UBX-LOG-STRING message. These will be timestamped if the receiver has current knowledge of time.

The CFG-LOGFILTER-* configuration group has several values which can be used to select position fix entries for logging. If CFG-LOGFILTER-APPLY_ALL_FILTERS is *false*, then all position fixes will be logged (subject to a maximum rate of 1 Hz). Otherwise, a position is logged if any of the or if all of MIN_INTERVAL, TIME_THRS, SPEED_THRS or POSITION_THRS thresholds are exceeded. If a threshold is set to zero it is ignored.

Position fixes are only recorded if a valid fix is obtained. Failed and invalid fixes are not recorded. Position fixes are compressed to reduce the amount of flash space used. In order to improve the compression, the fix values are rounded. This means that the values returned by the logging system may differ slightly from those that are gathered in real time.

The recorded data for a fix comprises:

- The time and date of the fix recorded to a precision of one second.
- Latitude and longitude to a precision of one millionth of a degree. Depending on position on Earth this is a precision in the order of 0.1 m.
- Altitude (height above mean sea level) to a precision of 0.1 m. Entries with an altitude lower than -470 m (lower than the lowest point on earth) or higher than 20,000 m may not be recorded in the log.
- Ground speed to a precision of 1 cm/s.
- The fix type (only successful fix types, since these are the only ones recorded).
- The number of satellites used in the fix is recorded, but there is a maximum count which can be recorded. If the actual count exceeds this maximum count then the maximum count will be recorded. If a log entry is retrieved with a satellite count equal to the maximum this means that value or more. The maximum count is 51.
- A horizontal accuracy estimate is recorded to give an indication of fix quality. This is an approximate compressed representation of the accuracy as determined by the fix process. Any accuracy less than 0.7 m will be recorded as 0.7 m and any value above 1 km will be recorded as 1 km. Within these limits, the recorded accuracy will always be greater than the fix accuracy number (by up to 40%).
- Heading to a precision of one degree.
- Odometer distance data (if odometer is enabled).
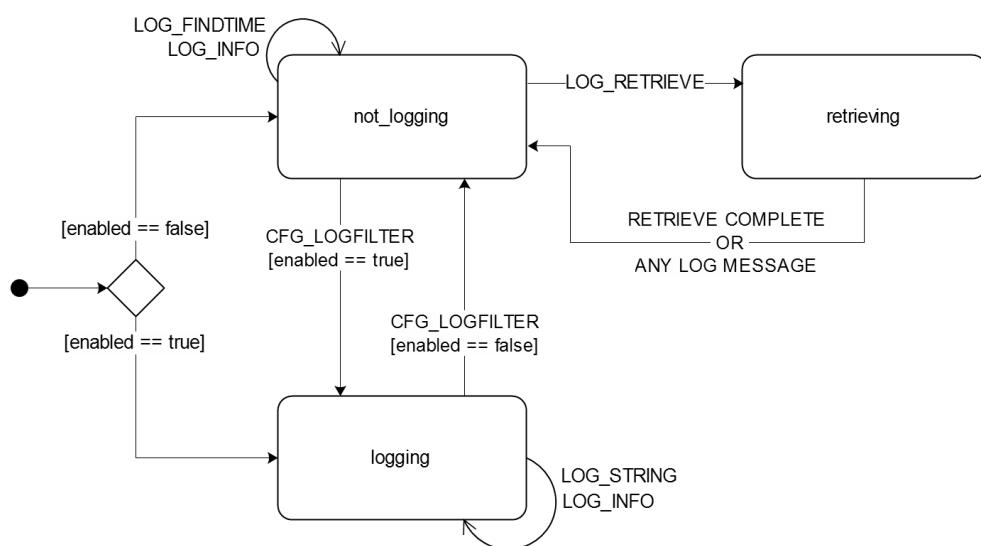
Figure 8: The states of the active logging subsystem

3.4.5 Retrieval

UBX-LOG-RETRIEVE starts the process which allows the receiver to output log entries. UBX-LOG-INFO may be helpful to a host system in order to understand the current log status before retrieval is started.

Once retrieval has started, one message will be output from the receiver for each log entry requested. Sending any logging message to the receiver during retrieval will cause the retrieval to stop before the message is processed.

To maximize the speed of transfer it is recommended that a high communications data rate is used and GNSS processing is stopped during the transfer (see UBX-CFG-RST).

UBX-LOG-RETRIEVE can specify a start entry index and entry count. The maximum number of entries that can be returned in response to a single UBX-LOG-RETRIEVE message is 256. If more entries than this are required the message will need to be sent multiple times with different startEntry indices. It might be useful to stop recording via CFG_LOGFILTER-RECORD_ENA while retrieving log entries from a circular log to avoid deletion of the requested entries between the request and transmission.

The receiver will send a UBX-LOG-RETRIEVEPOS message for each position fix log entry and a UBX-LOG-RETRIEVESTRING message for each string log entry. If the odometer was enabled at the time a position was logged, then a UBX-LOG-RETRIEVEPOSEXTRA will also be sent. Messages will be sent in the order in which they were logged, so UBX-LOG-RETRIEVEPOS and UBX-LOG-RETRIEVESTRING messages may be interspersed in the message stream.

The UBX-LOG-FINDTIME message can be used to search a log for the index of the first entry less than or equal to the given time. This index can then be used with the UBX-LOG-RETRIEVE message to provide time-based retrieval of log entries.

3.4.6 Command message acknowledgment

Some log operations may take a long time to execute because of the time taken to write to flash memory. The time for some operations may be unpredictable since the number and timing of flash operations may vary. In order to allow host software to synchronize to these delays logging

messages will always produce a response. This will be UBX-ACK-NAK in case of error, otherwise UBX-ACK-ACK unless there is some other defined response to the message.

It is possible to send a small number of logging commands without waiting for acknowledgment, since there is a command queue, but this risks confusion between the acknowledgments for the commands. Also a command queue overflow would result in commands being lost.

3.5 Data batching

3.5.1 Introduction

The data batching feature allows position fixes to be stored in the RAM of the receiver to be retrieved later in one batch. Batching of position fixes happens independently of the host system, and can continue while the host is powered down.

Table 11 lists all the batching-related messages:

Message	Description
UBX-MON-BATCH	Provides information about the buffer fill level and dropped data due to overrun
UBX-LOG-RETRIEVEBATCH	Starts the batch retrieval process
UBX-LOG-BATCH	A batch entry returned by the receiver

Table 11: Batching-related messages

3.5.2 Setting up the data batching

Data batching is disabled per default and it has to be configured before use via the CFG-BATCH-* configuration group.

The feature must be enabled and the buffer size must be set to greater than 0. It is possible to set up a PIO as a flag that indicates when the buffer is close to filling up. The fill level when this PIO is asserted can be set by the user separately from the buffer size. The notification fill level must not be larger than the buffer size.

If the host does not retrieve the batched fixes before the buffer fills up the oldest fix will be dropped and replaced with the newest.

The RAM available in the chip limits the size of the buffer. To make the best use of the available space users can select what data they want to batch. When batching is enabled a basic set of data is stored and the configuration flags EXTRAPVT and EXTRAODO can be used to store more detailed information about the position fixes. Doing so reduces the number of fixes that can be batched.

The receiver will reject configuration if it cannot allocate the required buffer memory. To ensure robust operation of the receiver the limits in Table 12 are enforced:

EXTRAPVT	EXTRAODO	Maximum number of epochs
0	0	300
0	1	221
1	0	156
1	1	132

Table 12: Maximum number of batched epochs

☞ It is recommended to disable all periodic output messages when using batching. This improves system robustness and also helps ensure that the output of batched data is not delayed by other messages.

☞ The buffer size is set up in terms of navigation epochs. This means that the time that can be covered with a certain buffer depends on the navigation rate. This rate can be set via the "CFG-RATE-" configuration group.

3.5.3 Retrieval

UBX-LOG-RETRIEVEBATCH starts the process which allows the receiver to output batch entries. Batching must not be stopped for readout; all batched data is lost when the feature is disabled.

Batched fixes are always retrieved starting with the oldest fix in the buffer and progressing towards newer ones. There is no way to skip certain fixes during retrieval.

When a UBX-LOG-RETRIEVEBATCH message is sent the receiver transmits all batched fixes. It is recommended to send a retrieval request with `sendMonFirst` set. This way the receiver will send a UBX-MON-BATCH message first that contains the number of fixes in the batching buffer. This information can be used to detect when the u-blox receiver finishes sending data.

Once retrieval has started, the receiver will first send UBX-MON-BATCH if `sendMonFirst` option was selected in the UBX-LOG-RETRIEVEBATCH. After that, it will send UBX-LOG-BATCH messages with the batched fixes.

To maximize the speed of transfer it is recommended that a high communications data rate is used.

☞ The receiver will discard retrieval request while processing a previous UBX-LOG-RETRIEVEBATCH message.

☞ The receiver does **not** acknowledge the reception of UBX-LOG-RETRIEVEBATCH; the response that the host should expect are the reply messages.

3.6 Spectrum analyzer

The UBX-MON-SPAN message can be enabled in u-center to provide a low-resolution spectrum analyzer sufficient to identify noise or jammers in the reception band. Once enabled, u-center includes a real-time chart that is updated once per second with the message data. See Figure 9 for an example. The main use for the spectrum analyzer tool is to help idenfity in-band interference sources during prototyping, but it can also be used during normal operation.

The shape of the spectrum as well as any RF interference in the form of a spur or intermodulation product can be easily analyzed with this graphical tool. The vertical axis compares the amplitude difference in dB for each frequency. A good spectrum shape is characterized by an even noise floor along with the GNSS band. For example, if any unwanted spur stands out, the vertical axis gives a rough approximation of the power level in dB compared to the noise floor.

Next to the chart, the center frequency, span, and resolution values set for the spectrum, and the PGA value are also displayed. The PGA value represents the internal gain set by the receiver, which depends on the external amplification of the GNSS input signal.
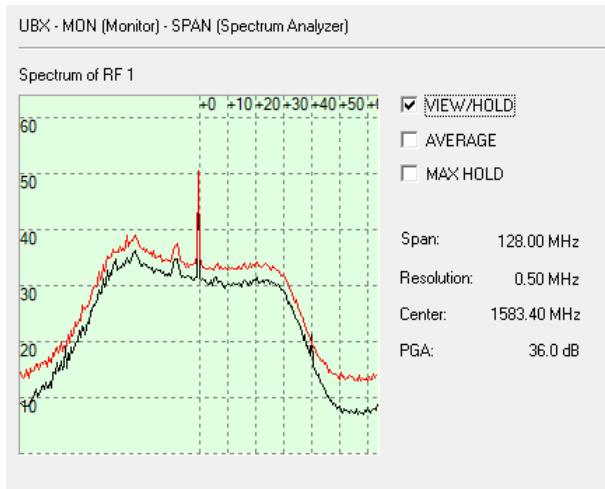
The vertical discontinuous lines in the chart area represent the offset to the center frequency in MHz. This helps to estimate the frequency of any spurious emission seen.

In addition, u-center includes three functions commonly found in any spectrum analyzer. These features support the RF front-end design and help to identify any jammer presence during the application operation.

- View / hold: if selected, the current spectrum shape freezes in a colored line. This allows for a comparison between the time the spectrum was frozen and the real-time spectrum. This is particularly helpful in assessing the impact of running other onboard components.

- Average: if selected, a colored line shows the averaged spectrum for each frequency. This is useful for analysis over time and to obtain a less noisy shape.
- Max hold: if selected, a colored line shows the maximum amplitude measured at each frequency. This option helps to identify any jammer over a period of time.

Figure 9 shows the spectrum view in u-center with the view/hold option selected. The redish line represents the frozen spectrum before modifying the external gain, while the black line represents the current measurement.

Figure 9: Spectrum analyzer view in u-center with the option view/hold selected

☞ By changing the number of constellations enabled, the span widens or narrows. This has a direct impact on the spectrum resolution, as the number of frequencies measured is fixed at 256. For further details about this message and how to calculate each frequency, see the Interface description [2].

☞ A big spur may be visible in the center frequency. The signal comes internally from the receiver and it does not cause any degradation in the performance.

3.7 Communication interfaces

u-blox receivers are equipped with a communication interface[1] which is multi-protocol capable. The interface ports can be used to transmit GNSS measurements, monitor status information and configure the receiver.

A protocol (e.g. UBX, NMEA) can be assigned to several ports simultaneously, each configured with individual settings (e.g. baud rate, message rates, etc.). More than one protocol (e.g. UBX protocol and NMEA) can be assigned to a single port (multi-protocol capability), which is particularly useful for debugging purposes.

The NEO-M9N provides UART1, SPI, I2C and USB interfaces for communication with a host CPU. The interfaces are configured via the configuration methods described in the applicable interface description [2].

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

Port no.	UBX-MON-COMMS portId	Electrical interface
1	0x0100	UART1

---
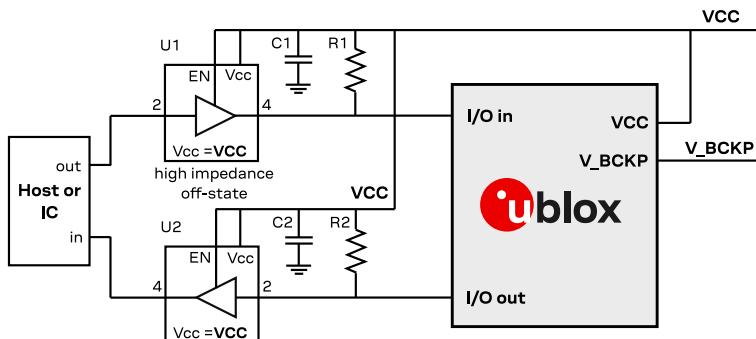
[1] The signal names and related terms have been replaced with new terminology in this document.

Port no.	UBX-MON-COMMS portId	Electrical interface
3	0x0300	USB

Table 13: Port number assignment

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

Example isolation circuit is shown below. The buffer (U1 and U2) type can be either push-pull or open collector. The buffer needs a capacitor nearby (C1 and C2, e.g. 100 nH). A pull-up resistor (R1 and R2, e.g. 10 kΩ) is needed if an open-collector buffer is used in a high-speed data line. The buffer must isolate the line when the VCC is turned off, so it must be either supplied or enabled by the VCC. For the I2C line a dual bidirectional bus buffer is needed.

Figure 10: NEO-M9N interface isolation

3.7.1 UART

A Universal Asynchronous Receiver/Transmitter (UART) port consists of an RX and a TX line. Neither handshaking signals nor hardware flow control signals are available. The UART interface protocol and baud rate can be configured but there is no support for setting different baud rates for reception and transmission.

The NEO-M9N includes two UART serial ports. UART1 can be used as a host interface for configuration, monitoring and control.

☞ 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
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 14: Possible UART interface configurations

☞ 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.

Allow a short time delay of typically 100 ms between sending a baud rate change message and providing input data at the new rate. Otherwise some input characters may be ignored or the port could be disabled until the interface is able to process the new baud rate.

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.

3.7.2 I2C interface

An I2C interface is available for communication with an external host CPU or u-blox cellular modules in peripheral mode only. The I2C protocol and electrical interface supports the Fast-mode of the I2C industry standard with a maximum SCL clock frequency of 400 kHz. Backwards compatibility with Standard-mode I2C bus operation is not supported.

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 I2C lines additional external pull-up resistors may be necessary.

☞ To use the I2C interface D_SEL pin must be left open.

☞ In designs where the host uses the same I2C bus to communicate with more than one u-blox receiver, the I2C peripheral address for each receiver must be configured to a different value. Typically most u-blox receivers are configured to the same default I2C peripheral address value. To poll or set the I2C peripheral address, use the CFG-I2C-ADDRESS configuration item (see the applicable interface description [2]).

The CFG-I2C-ADDRESS configuration item is an 8-bit value containing the I2C peripheral address in 7 most significant bits, and the read/write flag in the least significant bit.

3.7.2.1 I2C register layout
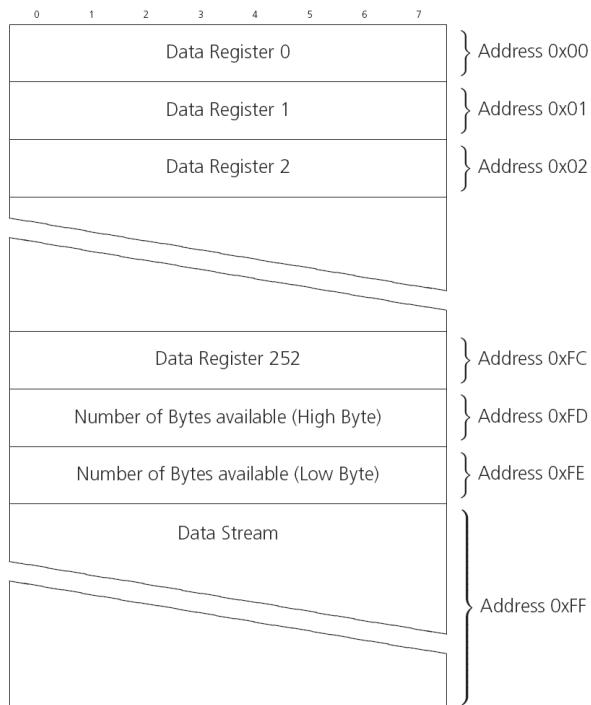
The I2C interface allows 256 registers to be addressed. As shown in Figure 11, only three of these are currently implemented.

The data registers 0 to 252 at addresses 0x00 to 0xFC contain reserved information, the result from their reading is currently undefined. The data registers 0 to 252 are 1 byte wide.

At addresses 0xFD and 0xFE it is possible to read the currently available number of bytes.

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 delivers value 0xFF, which cannot be the first byte of a valid message. If the message data is ready for transmission, the successive reads of register 0xFF will deliver the waiting message data.

☞ Do not use registers 0x00 to 0xFC. They are reserved for future use and they do not currently provide any meaningful data.

Figure 11: I2C register layout

3.7.2.2 Read access types

There are two I2C read transfer forms:

- The "random access" form: includes a peripheral register address and allows any register to be read.
- The "current address" form: omits the register address.

Figure 12 shows the format of the first one, the "random access" form of the request. Following the start condition from the controller, the 7-bit device address and the RW bit (which is a logic low for write access) are clocked onto the bus by the controller 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 acknowledgment, the controller 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 controller can read 1 to N bytes from the receiver, generating a not-acknowledge and a stop condition after the last byte being read.
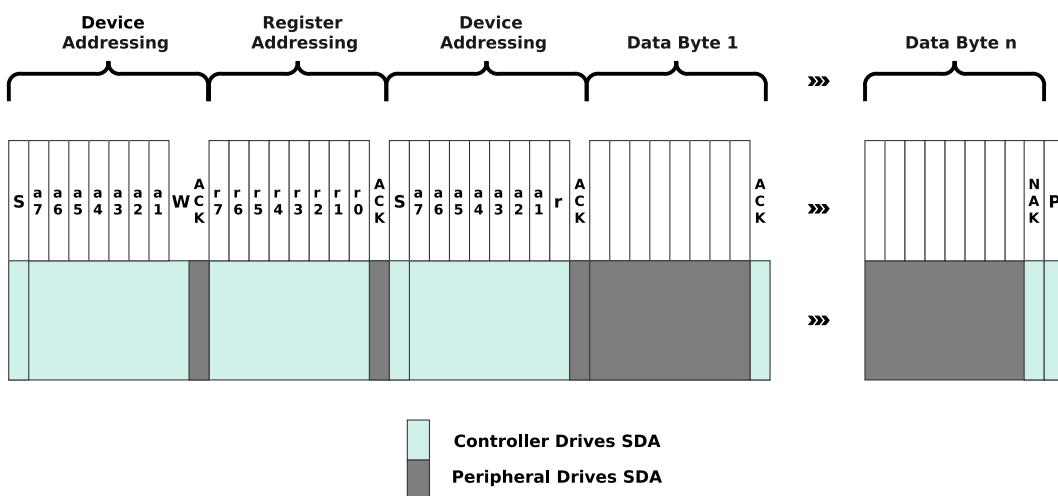
Figure 12: I2C random read access

If the second form, "current address" is used, 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 startup 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).
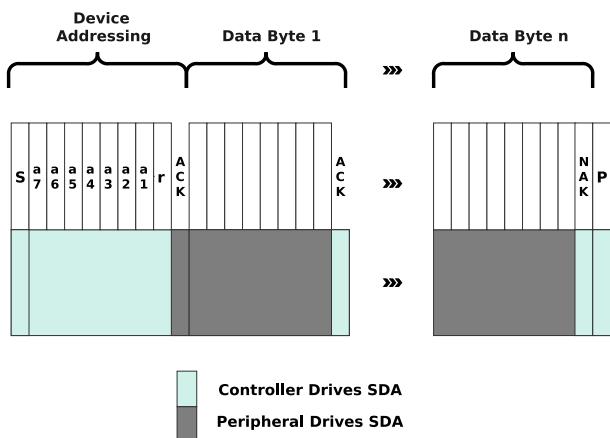
Figure 13: I2C current address read access

3.7.2.3 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 the section Read access is not writeable.

Following the start condition from the controller, the 7-bit device address and the RW bit (which is a logic low for write access) are clocked onto the bus by the controller transmitter. The receiver answers with an acknowledge (logic low) to indicate that it is responsible for the given address.

The controller 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.

Controller Drives SDA
Peripheral Drives SDA
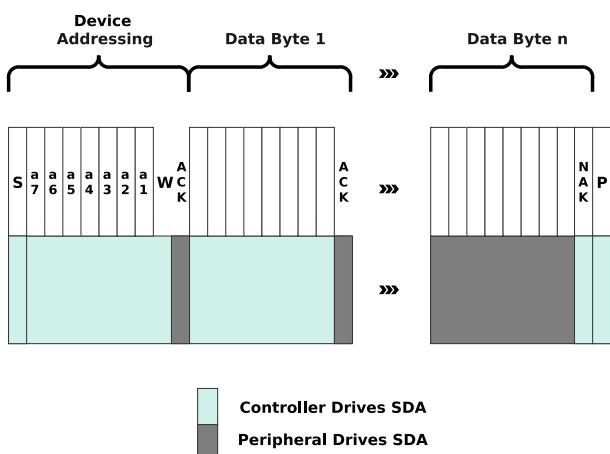
Figure 14: I2C write access

3.7.3 SPI interface

NEO-M9N has an SPI peripheral interface that can be selected by setting D_SEL = 0. The SPI peripheral interface is shared with UART1 and I2C port, the physical pins are same. The SPI pins available are:

- SPI_SDO (TXD)
- SPI_SDI (RXD)
- SPI_CS_N
- SPI_CLK

See more information about communication interface selection from the D_SEL section.

The SPI interface is designed to allow communication to a host CPU. The interface can be operated in peripheral mode only.

3.7.3.1 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 below). When no data is available to be written to the receiver, SDI 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 SDO 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.

3.7.3.2 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 simultaneously be read from the receiver. While the controller writes to SDI of the peripheral, at the same time it needs to read from SDO of the peripheral, as any pending data will be output by the receiver with this access. The data on SDO represents the results from a current address read, returning 0xFF when no more data is available.
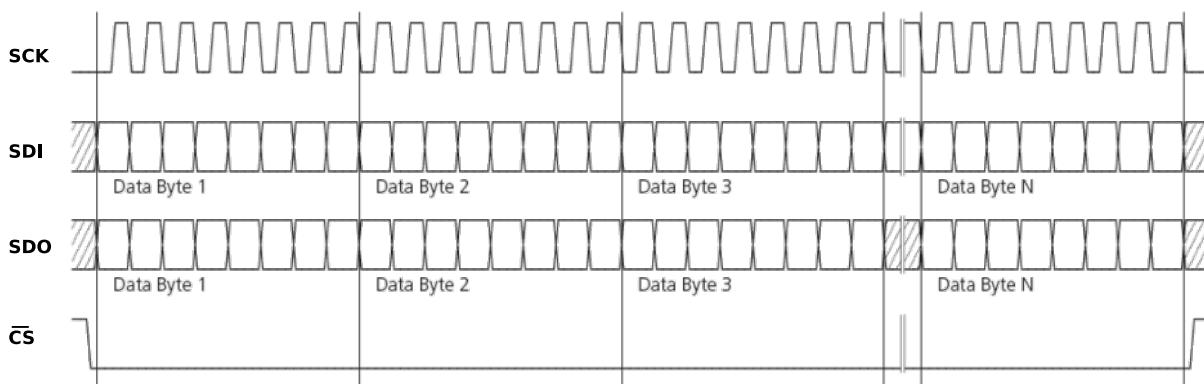
Figure 15: SPI back-to-back read/write access

3.7.4 USB interface

A single USB port is provided for host communication purposes.

The USB 2.0 FS (Full speed, 12 Mbit/s) interface can be used for host communication. Due to the hardware implementation, it may not be possible to certify the USB interface.

If the receiver executes code from internal ROM (i.e. when a valid flash firmware image is not detected), the USB behavior can differ compared to executing a firmware image from flash memory. USB host compatibility testing is thus recommended in this scenario.

The NEO-M9N receiver 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. The receiver draws only minimal amount of current (~ 1 mA) from V_USB.

☞ USB suspend mode is not supported.

☞ USB bus-powered mode is not supported.

☞ It is important to connect V_USB to ground and leave data lines open when the USB interface is not used in an application.

☞ 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 enumeration. However, messages generated between receiver startup and USB enumeration are not visible on the USB port.

V_USB needs to be powered by a separate LDO enabled by module VCC and supplied by the USB host.
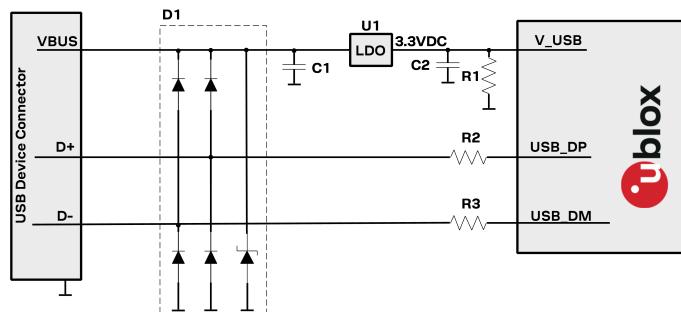
Figure 16: NEO-M9N example circuit for USB interface

Part	Recommended part	Description
U1	Seiko Instruments Inc. S-1206B33-I6T2G	LDO which provides 3.3 V to the V_USB pin.
C1, C2	Depends on USB LDO (U1) specification	U1 LDO decoupling capacitor.
D1	ST Microelectronics USBLC6-2	Low capacitance ESD protection diodes to avoid cross supply.
R1	100 kΩ	Pull-down resistor in case USB interface is not used. Not needed it if USB is used.
R2, R3	27 Ω 5% 0.1W	Serial termination resistor for USB.

Table 15: Recommended components for USB interface

3.8 Predefined PIOs

In addition to the communication ports, there are some predefined PIOs provided by NEO-M9N to interact with the receiver. These PIOs are described in this section.

☞ If hardware backup mode is used a proper isolation of the interfaces is needed.

3.8.1 D_SEL

The D_SEL pin can be used to configure the functionality of the combined UART1, I2C, and SPI pins. It is possible to configure the pins as UART1 + I2C, or as SPI. SPI is not available unless D_SEL pin is set to low. See Table 16 below.

Pin no.	D_SEL == 0	D_SEL == 1
20	SPI_SDO	UART1 TXD
21	SPI_SDI	UART1 RXD
18	SPI_CS_N	I2C SDA
19	SPI_CLK	I2C SCL

Table 16: D_SEL configuration

3.8.2 RESET_N

The NEO-M9N 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 delete all information and trigger a cold start. It should only be used as a recovery option.

3.8.3 SAFEBOOT_N

The NEO-M9N provides a SAFEBOOT_N pin that is used to command the receiver safeboot mode.

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

The safeboot mode can be used to recover from situations where the flash content has become corrupted and needs to be restored.

In safeboot 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 UART1, I2C or SPI communication is possible. For communication via UART1 in safeboot mode, the host must send a training sequence (0x55 0x55 at 9600 baud) to the receiver in order to begin communication. After this the host must wait at least 2 ms before sending any data.

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.

3.8.4 TIMEPULSE

The NEO-M9N module provides a time pulse on the TIMEPULSE pin.

More information about the time pulse feature and its configuration can be found in the Time pulse section.

3.8.5 TX_READY

This feature enables each port to define a corresponding pin, which indicates if bytes are ready to be transmitted. A listener can wait on the TX-READY signal instead of polling the I2C 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. 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 might 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 in the applicable interface description [2] for a list of the PIOs and their mapping). Each TX_READY pin is exclusively associated to one port and cannot be shared. If PIO is invalid or already in use, only the configuration for the specific 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 validate successful configuration of the TX_READY pin, the port configuration should be read back and the settings of TX-READY feature verified (will be set to disabled/all zero if the settings are invalid).

The threshold when TX_READY is asserted should not be set above 2 kB as it is possible that the internal message buffer limit is reached before this. This results in the TX_READY pin never being set as the messages are discarded before the threshold is reached.

3.8.5.1 Extended TX timeout

If the host does not communicate over SPI or I2C 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.

3.8.6 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. The external frequency aiding can be used to calibrate the clock. This enables faster fix of satellite signals (UBX-MGA-INI-FREQ or UBX-MGA-INI-TIME_XXX) and can be used during normal operation or during the production test. Another possibility to use the extint feature is to wake up the receiver after putting it into backup mode; this can be set up with UBX-RXM-PMREQ. Leave open if unused, this function is disabled by default.

☞ If the EXTINT pin is configured for on/off switching of the receiver, the internal pull-up becomes disabled. Thus make sure the EXTINT input is always driven within the defined voltage level by the host.

3.9 Multiple GNSS assistance (MGA)

The u-blox AssistNow services provide a proprietary implementation of an A-GNSS protocol compatible with u-blox GNSS receivers.

The MGA services consist of AssistNow Online and Offline variants delivered by HTTP or HTTPS protocol. AssistNow Online optionally provides immediate satellite ephemerides, health information and time aiding data suitable for GNSS receiver systems with direct internet access.

When a client device makes an AssistNow request, the service responds with the requested data using standard UBX protocol MGA messages. These messages are ready for direct transmission from the client to the receiver port without requiring any modification.

The AssistNow Offline service benefits u-blox GNSS receivers that only have occasional internet access. Users request data from the service by specifying the time period for which they want coverage (1 to 5 weeks).

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

In addition there is an MGA service called AssistNow Autonomous which does not need an internet connection and runs entirely on the receiver.

Table 17 below contains an overview of the different MGA services u-blox provides.

Requirements	AssistNow Online	AssistNow Offline	AssistNow Autonomous
Requires external flash memory	No	Optional	Optional
Requires internet connection	Permanently	Sporadically	No
Amount of internet data	Medium	High	None
Ephemeris in data	Yes	No	No
Almanac in data	Yes	Yes	Yes

Table 17: AssistNow service overview

3.9.1 Authorization

To use the AssistNow services, customers will need to obtain an authorization token from u-blox. Go to https://www.u-blox.com/en/solution/services/assistnow or contact your local technical support to get more information and to request access to the service.

3.9.2 Preserving MGA and operational data during power-off

The time-to-fix after a receiver power interruption is dependent on the amount of operational data available at startup. Satellite broadcast information plus an estimate of accurate time can be fetched form the AssistNow service. In addition, the following techniques can restore previously stored data prior to power down.

• **Battery-backed RAM:**The receiver operational state stored in this RAM can be maintained during power outages by connecting the V_BCKP pin to an independent supply, e.g a battery. This is a recommended method as it will maintain all MGA related information plus any user configuration, calibration data and an estimate of time via the Real Time Clock.
See section V_BCKP: Backup supply voltage for more information.

- **Save-on-shutdown:**The receiver can be instructed to dump its current state to flash memory as part of the shutdown procedure; this data is then automatically retrieved when the receiver is restarted. For more information, see section Save-on-shutdown feature for more info. For information on the UBX-UPD-SOS messages consult the applicable Interface description [2].
- **Database dump:** The receiver can be made to dump the state of its navigation database in the form of a sequence of UBX messages reported to the host; these messages can be stored by the host and then sent back to the receiver when it has been restarted. See the description of the UBX-MGA-DBD messages in the applicable Interface description [2] for more information.

3.9.3 AssistNow Offline

AssistNow Offline is a feature that combines special firmware in u-blox receivers and a proprietary service run by u-blox. It is targeted at receivers that only have occasional internet access and so cannot use AssistNow Online. AssistNow Offline speeds up time to first fix (TTFF), typically to considerably less than 10 s.

☞ AssistNow Offline currently supports GPS and GLONASS. u-blox intends to expand the AssistNow Offline Service to support other GNSS (such as BeiDou and Galileo) in due course.

The AssistNow Offline 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 Offline.

Users of AssistNow Offline are expected to download data from the AssistNow Offline Service, specifying the time period they want covered (1 day to 5 weeks) and the types of GNSS. This data must be uploaded to a u-blox receiver, so that it can estimate the positions of the satellites, when no better data is available. Using these estimates will not provide as accurate a position fix as if current ephemeris data is used, but it will allow much faster TTFFs in nearly all cases.

The data obtained from the AssistNow Offline Service is organized by date, normally a day at a time. Consequently the longer time for which coverage is requested, the larger the amount of data to handle. Similarly, each different GNSS requires its own data and in the extreme cases, several hundred kilobytes of data will be provided by the service. This amount can be reduced by requesting lower resolution, but this will have a small negative impact on both position accuracy and TTFF. See the section on Offline Service Parameters for details of how to specify these options.

The downloaded Offline data is encoded in a sequence of UBX-MGA-ANO messages, one for every satellite for every day of the period covered. Thus, for example, data for all GPS satellites for 4 weeks will involve in excess of 900 separate messages, taking up around 70 kb. Where a u-blox receiver has flash storage, all the data can be directly uploaded to be stored in the flash until it is needed. In this case, the receiver will automatically select the most appropriate data to use at any time. See the section on flash-based AssistNow Offline for further details.

AssistNow Offline can also be used where the receiver has no flash storage, or there is insufficient spare flash memory. In this case the customer's system must store the AssistNow Offline data until the receiver needs it and then upload only the appropriate part for immediate use. See the section on host-based AssistNow Offline for further details.

3.9.3.1 Service parameters

The information exchange with the AssistNow Offline Service is based on the HTTP protocol. Upon reception of an HTTP 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 HTTP GET request from the client to the server should contain a standard HTTP 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 not important.
- Keys and values are case-sensitive.
- Keys and values must be separated by an equals 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 (',').

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. The currently supported GNSS are: gps and glo.
period	Numeric [weeks]	Optional	The number of weeks into the future the data should be valid for. Data can be requested for up to 5 weeks in to the future. If this value is not provided, the server assumes a period of 4 weeks.
days	Numeric [days]	Optional	The number of days into the future the data should be valid for. Can be used instead of the period parameter when data for less than one week is desired.
resolution	Numeric [days]	Optional	The resolution of the data: 1=every day, 2=every other day, 3=every third day. If this value is not provided, the server assumes a resolution of 1 day.

Table 18: AssistNow Offline parameter keys

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

token=XXXXXXXXXXXXXXXXXXXXXX;gnss=gps,glo;

3.9.3.2 Time, position and almanac

While AssistNow Offline can be used on its own, it is expected that the user will provide estimates of the receiver's current position, the current time and ensure that a reasonably up-to-date almanac is available. In most cases this information is likely to be available without the user needing to do anything. For example, where the receiver is connected to a battery backup power supply and has a functioning real-time clock (RTC), the receiver will keep its own sense of time and will retain the last known position and any almanac. However, should the receiver be completely without power before startup, then it will greatly improve TTFF if time, position and almanac can be supplied in some form.

Almanac data has a validity period of several weeks, so it can be downloaded from the AssistNow Online service at roughly the same time the Offline data is obtained. It can then be stored in the host for uploading on receiver startup, or it can be transferred to the receiver straight away and preserved there (provided suitable non-volatile storage is available).

Obviously, where a receiver has a functioning RTC, it should be able to keep its own sense of time, but where no RTC is fitted (or power is completely turned off), providing a time estimate via the UBX-MGA-INI-TIME_UTC message will be beneficial to lower the time to first fix and make use of the AssistNow Offline data.

Similarly, where a receiver has effective non-volatile storage, the last known position will be recalled, but if this is not the case, then it will help TTFF to provide a position estimate via one of the UBX-MGA-INI-POS_XYZ or UBX-MGA-INI-POS_LLH messages (details can be found in NEO-M9N Interface description [2].

Where circumstance prevent the provision of all three of these pieces of data, providing some is likely to be better than none at all, as this helps to lower the TTFF.

3.9.3.3 Flash-based AssistNow Offline

Flash-based AssistNow Offline functionality means that AssistNow Offline data is stored in the flash memory connected to the chip.

The user's host system must download the data from the AssistNow Offline service when an internet connection is available, and then deliver all of that data to the u-blox receiver. As the total amount of data to be uploaded is large (typically around 100 kbytes) and writing to flash memory is slow, the upload must be done in blocks of up to 512 bytes, one at a time. The UBX-MGA-FLASH-DATA message is used to transmit each block to the receiver.

☞ AssistNow Offline data stored in flash memory is not affected by any reset of the receiver. The only simple ways to clear it are to completely erase the whole flash memory or to overwrite it with a new set of AssistNow Offline data. Uploading a dummy block of data (e.g. all zeros) will also have the effect of deleting the data, although a small amount of flash storage will be used.

3.9.3.3.1 Flash-based storage procedure

The following steps are a typical sequence for transferring AssistNow Offline data into the receiver's flash memory:

- The host downloads a copy of a latest data from the AssistNow Offline service and stores it locally.
- It sends the first 512 bytes of that data using the UBX-MGA-FLASH-DATA message.
- It awaits a UBX-MGA-FLASH-ACK message in reply.
- Based on the contents of the UBX-MGA-FLASH-ACK message it sends the next block, resends the last block or aborts the whole process.
- The above three steps are repeated until all the rest of the data has been successfully transferred (or the process has been aborted).
- The host sends a UBX-MGA-FLASH-STOP message to indicate completion of the upload.
- It awaits the final UBX-MGA-FLASH-ACK message in reply. Background processing in the receiver prepares the downloaded data for use at this stage. Particularly if the receiver is currently busy, this may take quite a few seconds, so the host has to be prepared for a delay before the UBX-MGA-FLASH-ACK is seen.

Note that the final block may be smaller than 512 bytes (where the total data size is not perfectly divisible by 512). Also, the UBX-MGA-FLASH-ACK messages are distinct from the UBX-MGA-ACK messages used for other AssistNow functions.

Any existing data will be deleted as soon as the first block of new data arrives, so no useful data will be available till the completion of the data transfer. Each block of data has a sequence number, starting at zero for the first block. In order to guard against invalid partial data downloads the receiver will not accept blocks which are out of sequence.

3.9.3.4 Host-based AssistNow Offline

Host-based AssistNow Offline involves AssistNow Offline data being stored until it is needed by the user's host system in whatever memory it has available.

The user's host system must download the data from the AssistNow Offline service when an internet connection is available, but retain it until the time the u-blox receiver needs it. At this point, the host must upload just the relevant portion of the data to the receiver, so that the receiver can start using it. This is achieved by parsing all the data and selecting for upload to the receiver only those UBX-MGA-ANO messages with a timestamp nearest the current time. As each is a complete

UBX message it can be sent directly to the receiver with no extra packaging. If required the user can select to employ flow control, but in most cases this is likely to prove unnecessary.

When parsing the data obtained from the AssistNow Offline service the following points should be noted:

- The data is made up of a sequence of UBX-MGA-ANO messages.
- Customers should not rely on the messages all being of fixed size, but should read their length from the UBX header to work out where the message ends (and where the next begins).
- Each message indicates the satellite for which it is applicable through the svId and gnssId fields.
- Each message contains a timestamp within the year, month and day fields.
- Midday (UTC) on the day indicated should be considered to be the point at which the data is most applicable.
- The messages will be ordered chronologically, earliest first.
- Messages with the same timestamp will be ordered by ascending gnssId and then ascending svId.

3.9.3.4.1 Host-based procedure

The following steps are a typical sequence for host-based AssistNow Offline:

- The host downloads a copy of the latest data from the AssistNow Offline service and stores it locally.
- Optionally it may also download a current set of almanac data from the AssistNow Online service.
- The host wants to use the u-blox receiver.
- If necessary it uploads any almanac, position estimate and/or time estimate to the receiver.
- It scans through AssistNow Offline data looking for entries with a timestamp that most closely matches the current (UTC) time/date.
- The host sends each such UBX-MGA-ANO message to the receiver.

Note that when data has been downloaded from the AssistNow Offline service with the (default) resolution of one day, the means for selecting the closest matching timestamp is simply to look for ones with the current (UTC) date.

3.9.4 AssistNow Autonomous

3.9.4.1 Introduction

The assistance scenarios covered by *AssistNow Online* and *AssistNow Offline* require an online connection and a host that can use this connection to download aiding data and provide this to the receiver when required.
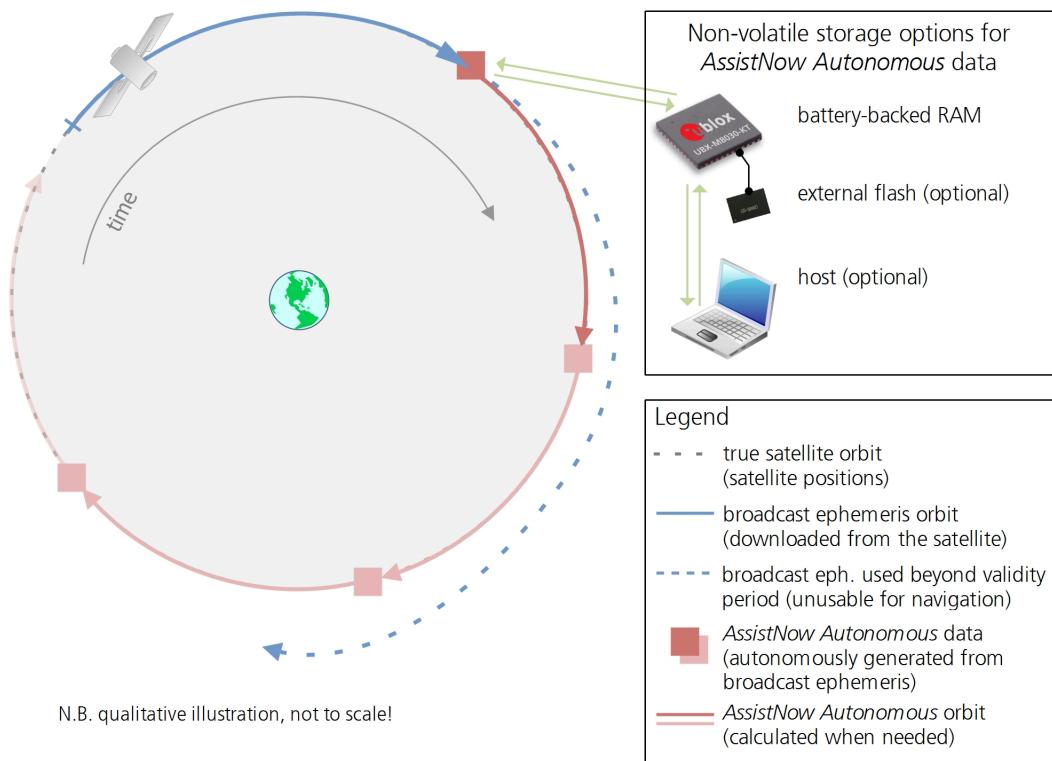
The *AssistNow Autonomous* feature provides a functionality similar to *AssistNow Offline* without the need for a host and a connection. Based on a broadcast ephemeris downloaded from the satellite (or obtained by *AssistNow Online*) the receiver can autonomously (i.e. without any host interaction or online connection) generate an accurate satellite orbit representation ("AssistNow Autonomous data") that is usable for navigation much longer than the underlying broadcast ephemeris was intended for. This makes downloading new ephemeris or aiding data for the first fix unnecessary for subsequent startups of the receiver.

☞ The *AssistNow Autonomous* feature is enabled by default. It can be disabled using the CFG-ANA-USE_ANA configuration item.

3.9.4.2 Concept

The figure below illustrates the *AssistNow Autonomous* concept in a graphical way. Note that the figure is a qualitative illustration and is not to scale.

- A broadcast ephemeris downloaded from the satellite is a precise representation of a part (for GPS nominally four hours) of the satellite's true orbit (trajectory). It is not usable for positioning beyond this validity period because it diverges dramatically from the true orbit afterwards.
- The *AssistNow Autonomous orbit* is an extension of one or more broadcast ephemerides. It provides a long-term orbit for the satellite for several revolutions. Although this orbit is not perfectly precise it is a sufficiently accurate representation of the true orbit to be used for navigation.
- The *AssistNow Autonomous data* is automatically and autonomously generated from downloaded (or assisted) ephemerides. The data is stored automatically in the on-chip battery-backed memory (BBR). Optionally, the data can be backed up in external flash memory or on the host. The number of satellites for which data can be stored depends on the receiver configuration and may change during operation.
- If no broadcast ephemeris is available for navigation *AssistNow Autonomous* automatically generates the required parts of the orbits suitable for navigation from the stored data. The data is also automatically kept current in order to minimize the calculation time once the navigation engine needs orbits.
- The operation of the *AssistNow Autonomous* feature is transparent to the user and the operation of the receiver. All calculations are done in the background and do not affect the normal operation of the receiver.
- The *AssistNow Autonomous* subsystem automatically invalidates data that has become too old and that would introduce unacceptable positioning errors. This threshold is configurable.
- The prediction quality will be automatically improved if the satellite has been observed multiple times. Improved prediction quality also positively affects the maximum usability period of the data.
- *AssistNow Autonomous* considers GPS, GLONASS, Galileo and BeiDou satellites only. It will not consider satellites on orbits with an eccentricity of >0.05 (e.g., Galileo E18). A single GLONASS broadcast ephemeris contains information only for approximately 30 minutes. This is not long enough to extend it in a usable way. Orbit information of each GLONASS satellite must be collected at least for four hours to generate data.

Non-volatile storage options for
AssistNow Autonomous data

battery-backed RAM

external flash (optional)

host (optional)

Legend

- - -   true satellite orbit
        (satellite positions)

——   broadcast ephemeris orbit
        (downloaded from the satellite)

– – –   broadcast eph. used beyond validity
        period (unusable for navigation)

■   *AssistNow Autonomous* data
        (autonomously generated from
        broadcast ephemeris)

——   *AssistNow Autonomous* orbit
        (calculated when needed)

N.B. qualitative illustration, not to scale!

3.9.4.3 Interface

Several UBX protocol messages provide interfaces to the *AssistNow Autonomous* feature. They are:

- The CFG-ANA-USE_ANA item is used to enable or disable the *AssistNow Autonomous* feature. When enabled, the receiver will automatically produce *AssistNow Autonomous* data for newly received broadcast ephemerides and, if that data is available, automatically provide the navigation subsystem with orbits when necessary and adequate. The CFG-ANA-* configuration group also allows for a configuration of the maximum acceptable orbit error. See the next section for an explanation of this feature. It is recommended to use the firmware default value that corresponds to a default orbit data validity of approximately three days (for GPS satellites observed once) and up to six days (for GPS and GLONASS satellites observed multiple times over a period of at least half a day).
- Note that disabling the *AssistNow Autonomous* feature will delete all previously collected satellite observation data from the flash memory.
- The UBX-NAV-AOPSTATUS message provides information on the current state of the *AssistNow Autonomous* subsystem. The status indicates whether the *AssistNow Autonomous* subsystem is currently idle (or not enabled) or busy generating data or orbits. Hosts should monitor this information and only power off the receiver when the subsystem is idle (that is, when the `status` field shows a steady zero).
- The UBX-NAV-SAT message indicates the use of *AssistNow Autonomous* orbits for individual satellites.
- The UBX-NAV-ORB message indicates the availability of *AssistNow Autonomous* orbits for individual satellites.
- The UBX-MGA-DBD message provides a means to retrieve the *AssistNow Autonomous* data from the receiver in order to preserve the data in power-off mode where no battery backup is available. Note that the receiver requires the absolute time (i.e. full date and time) to calculate *AssistNow Autonomous* orbits. For best performance it is, therefore, recommended to supply this information to the receiver using the UBX-MGA-INI-TIME_UTC message in this scenario.

- The Save-on-Shutdown feature preserves *AssistNow Autonomous* data (more information about the SoS feature: NEO-M9N Interface description [2].

3.9.4.4 Benefits and drawbacks

AssistNow Autonomous can provide quicker start-up times (lower the TTFF) provided that data is available for enough visible satellites. This is particularly true under weak signal conditions where it might not be possible to download broadcast ephemerides at all, and, therefore, no fix at all would be possible without *AssistNow Autonomous* (or A-GNSS). It is, however, required that the receiver roughly knows the absolute time, either from an RTC or from time-aiding (see the *Interface* section above), and that it knows which satellites are visible, either from the almanac or from tracking the respective signals.

The *AssistNow Autonomous* orbit (satellite position) accuracy depends on various factors, such as the particular type of satellite, the accuracy of the underlying broadcast ephemeris, or the orbital phase of the satellite and Earth, and the age of the data (errors add up over time).

AssistNow Autonomous will typically extend a broadcast ephemeris from three up to six days. The CFG-ANA-ORBMAXERR item allows changing this threshold by setting the «maximum acceptable modeled orbit error» (in meters). Note that this number does not reflect the true orbit error introduced by extending the ephemeris. It is a statistical value that represents a certain expected upper limit based on a number of parameters. A rough approximation that relates the maximum extension time to this setting is: *maxError [m] = maxAge [d] * f*, where the factor *f* is 30 for data derived from satellites seen once and and 16 for data derived for satellites seen multiple time during a long enough time period (see the Concept section above).

There is no direct relation between (true and statistical) orbit accuracy and positioning accuracy. The positioning accuracy depends on various factors, such as the satellite position accuracy, the number of visible satellites, and the geometry (DOP) of the visible satellites. Position fixes that include *AssistNow Autonomous* orbit information may be significantly worse than fixes using only broadcast ephemerides. It might be necessary to adjust the limits of the navigation output filters (CFG-NAVSPG-OUTFIL_XXXX).

A fundamental deficiency of any system that prevents them from precise satellite orbit predictions, are the unknown future events. Hence, the receiver will not be able to know about satellites that will have become unhealthy, have undergone a clock swap, or have had a maneuver. This means that the navigation engine might rarely mistake a wrong satellite position as the true satellite position. However, provided that there are enough other good satellites, the navigation algorithms will eventually eliminate a defective orbit from the navigation solution.

The repeatability of the satellite constellation is a potential pitfall for the use of the *AssistNow Autonomous* feature. For a given location on Earth the (GPS) constellation (geometry of visible satellites) repeats every 24 hours. Hence, when the receiver «learned» about a number of satellites at some point in time the same satellites will in most places *not* be visible 12 hours later, and the available *AssistNow Autonomous* data will not be of any help. Again 12 hours later, however, usable data would be available because it was generated 24 hours ago.

The longer a receiver observes the sky the more satellites it will have seen. At the equator, and with full sky view, approximately ten (GPS) satellites will show up in a one-hour window. After four hours of observation approx. 16 satellites (i.e. half the constellation), after 10 hours approx. 24 satellites (2/3rd of the constellation), and after approx. 16 hours the full constellation will have been observed (and *AssistNow Autonomous* data generated for). Lower sky visibility reduces these figures. Further away from the equator the numbers improve because the satellites can be seen twice a day. For

example at 47 degrees north the full constellation can be observed in approx. 12 hours with full sky view.

The calculations required for *AssistNow Autonomous* are carried out on the receiver. This requires energy and users may therefore occasionally see increased power consumption during short periods (several seconds, rarely more than 60 seconds) when such calculations are running. Ongoing calculations will automatically prevent the power save mode from entering the power-off state. The power-down will be delayed until all calculations are done.

☞ AssistNow Autonomous should be enabled if the system has sporadic access to the AssistNow Offline service. In this case the receiver will intelligently choose the more reliable orbit predictions for each satellite. This way the autonomous prediction can provide performance improvements if the offline data grows old or gets outdated.

3.10 Save-on-shutdown feature

The save-on-shutdown feature (SOS) enables the u-blox receiver to store the contents of the battery-backed RAM to an external flash memory and restore it upon startup. This allows the u-blox receiver to preserve some of the features available only with a battery backup (preserving configuration and satellite orbit knowledge) without having a battery backup supply present. It does not, however, preserve any kind of time knowledge. Save-on-shutdown must be commanded by the host. The restoring of data on startup is automatically done if the corresponding data is present in the flash. Data expiration is not checked.

The following outlines the suggested shutdown procedure when using the save-on-shutdown feature:

• With the UBX-CFG-RST message, the host commands the u-blox receiver to stop, specifying reset mode 0x08 ("Controlled GNSS stop") and a BBR mask of 0 ("Hotstart").
• The host commands the saving of the contents of BBR to the flash memory using the UBX-UPD-SOS-BACKUP message.
• For a valid request the u-blox receiver reports on the success of the backup operation with a UBX-UPD-SOS-ACK message.
• The host powers off the u-blox receiver.

The startup procedure is as follows:

• The host powers on the u-blox receiver.
• The u-blox receiver detects the previously stored data in the flash. It restores the corresponding memory and reports the success of the operation with a UBX-UPD-SOS-RESTORED message on the port on which it had received the save command message (if the output protocol filter on that port allows it). It does not report anything if no stored data has been detected.
• Additionally the u-blox receiver outputs a UBX-INF-NOTICE and/or a NMEA-TXT message with the contents `RESTORED` in the boot screen (depends on the configuration of the port and information messages) upon success.
• Optionally the host can deliver coarse time assistance using UBX-MGA-INI-TIME_UTC for better startup performance.

Once the u-blox receiver has started up it is recommended to delete the stored data using a UBX-UPD-SOS-CLEAR message. The u-blox receiver responds with a UBX-ACK-ACK / UBX-ACK-NAK message.

3.11 Power management

u-blox receivers support different power modes. These modes represent strategies of how to control the acquisition and tracking engines in order to achieve either the best possible performance or good performance with reduced power consumption.

Receiver power management can be split into two categories:

- **Externally controlled power management:** This includes various modes of power management that are directly operated by the user or host device. These modes are:
    1. External cycling of the receiver main power supply
    2. Instruct the receiver to turn on/off via the UBX-RXM-PMREQ message
    3. Instruct the receiver to turn on/off via external pins (EXTINT0 or EXTINT1)
- **Internally controlled power management:** Here the receiver makes the decision when to power down/up some/all of its internal components according to predefined parameters. It is also referred to as power save mode (PSM). In PSM one of two modes of operations can be selected:
    1. On/off operation (PSMOO)
    2. Cyclic tracking (PSMCT)

	Continuous operation	Cyclic operation	On/off operation
Main use	Continuous position tracking	Low-power continuous position tracking	Occasional position tracking
Possible applications	Navigation applications	Tracking devices with battery	Tracking devices which only need a position fix every few minutes or less

Table 19: Different power management modes

The following figure illustrates u-blox power management modes.
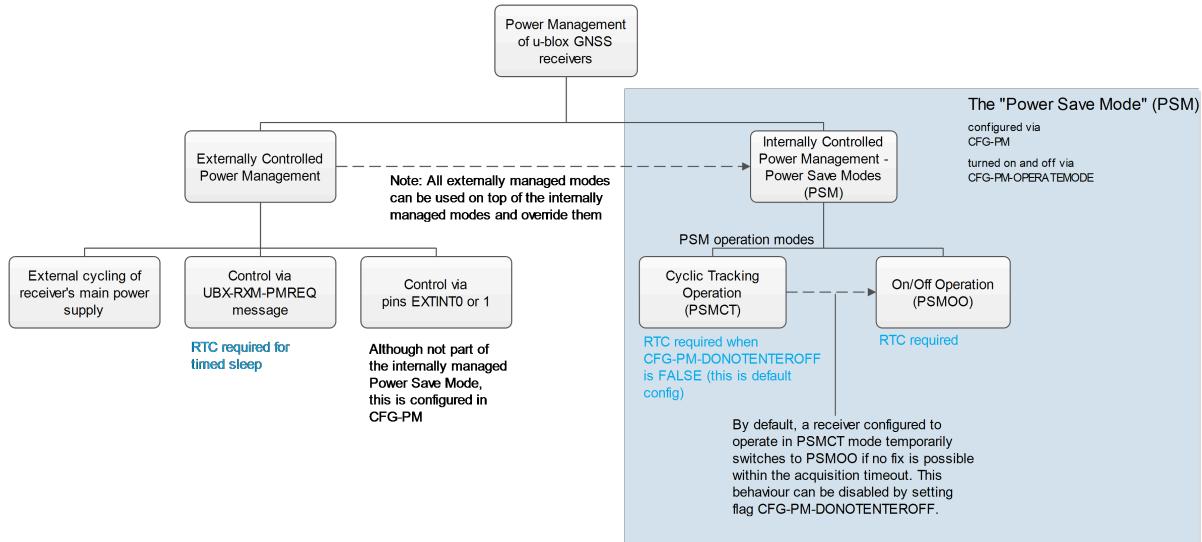
Figure 17: u-blox power management

The majority of the power management section explains the power save mode (internally controlled power management). However, some concepts relevant to the externally controlled power management are also explained, such as the EXTINT control, Wake up and Power on/off command.

Externally controlled power management operations override the internally controlled power management operations.

3.11.1 Continuous mode

u-blox receivers make use of dedicated signal processing engines optimized for signal acquisition and tracking. The acquisition engine delivers rapid signal searches during cold starts or when insufficient signals are available for navigation. The tracking engine delivers signal measurements for navigation and acquires new satellites as they become available during navigation. The resources of both engines are deployed adaptively to minimize overall power consumption.

Note that even if the acquisition engine is powered off, satellites continue to be acquired.

3.11.2 Power save mode

Power save mode (PSM) allows a reduction in system power consumption by selectively switching parts of the receiver on and off. It is selected with CFG-PM-OPERATEMODE and configured with items in the CFG-PM group.

PSM is designed to only support the operation of GPS, GLONASS, BeiDou, Galileo and QZSS. Enabling SBAS is possible only if at least one of the other systems is enabled. The PSM state machine behavior will not be altered by enabling SBAS and it will not take them into account in operation. Therefore, it is recommended to disable SBAS when operating in power save mode. They can be disabled in the CFG-SIGNAL group.

☞ The logic within power save mode is designed so that time pulse operation is not compromised. This means that entering all power saving states is delayed until the conditions necessary to produce a time pulse have been met. Therefore, in order to obtain good power save mode operation, it is essential that any time pulse is correctly configured with an appropriate time base, or that time pulses are turned off if not needed (by disabling the *TPX_ENA* item in the CFG-TP group).

3.11.2.1 Operation

Power save mode has two modes of operation:

- *Power save mode cyclic tracking (PSMCT) operation* is used when position fixes are required in short periods of 0.5 to 10 s.
- *Power save mode on/off (PSMOO) operation* is used for periods longer than 10 s, and can be in the order of minutes, hours or days.

The mode of operation can be configured, and depending on the setting, the receiver demonstrates different behavior: In on/off operation the receiver switches between phases of startup/navigation and phases with low or almost no system activity (backup/sleep). In cyclic tracking the receiver does not shut down completely between fixes, but uses low-power tracking instead.

Currently PSMCT is restricted to an update period between 0.5 and 10 seconds and PSMOO is restricted to an update period over 10 seconds. However, this may change in future firmware releases.

PSM is based on a state machine with five different states: *(Inactive) awaiting next fix* and *(Inactive) awaiting next search* states, *Acquisition* state, *Tracking* state and *power optimized tracking (POT)* state.

- *Inactive* states: Most parts of the receiver are switched off.
- *Acquisition* state: The receiver actively searches for and acquires signals. Maximum power consumption.

- *Tracking* state: The receiver continuously tracks and downloads data. Less power consumption than in *Acquisition* state.
- *POT* state: The receiver repeatedly loops through a sequence of tracking (Track), calculating the position fix (Fix), and entering an idle period (Idle). No new signals are acquired and no data is downloaded. Much less power consumption than in *Tracking* state.

The following figure illustrates the PSM state machine:
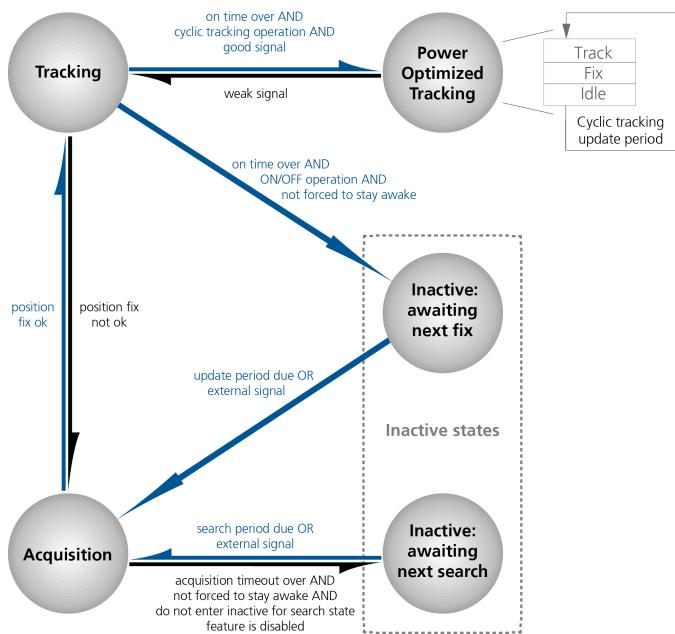
Figure 18: State machine

3.11.2.1.1 Acquisition timeout logic

The receiver has internal, external and user-configurable mechanisms that determine the time to be spent in acquisition state. This logic is put in place to ensure good performance and low power consumption in different environments and scenarios. This collective logic is referred to as Acquisition timeout.

Internal mechanisms:

- If the receiver is able to acquire weak signals but not of the quality needed to get a fix, it will transition to *(Inactive) awaiting next search* state after the timeout configured in *MAXACQTIME* or earlier if too few signals are acquired.
- If the receiver is unable to acquire any signals or it acquires a small number of extremely bad signals (e.g., no sky view), it will transition to *(Inactive) awaiting next search* state after 15 seconds or the timeout configured in *MAXACQTIME* if shorter.

User-configurable mechanisms:
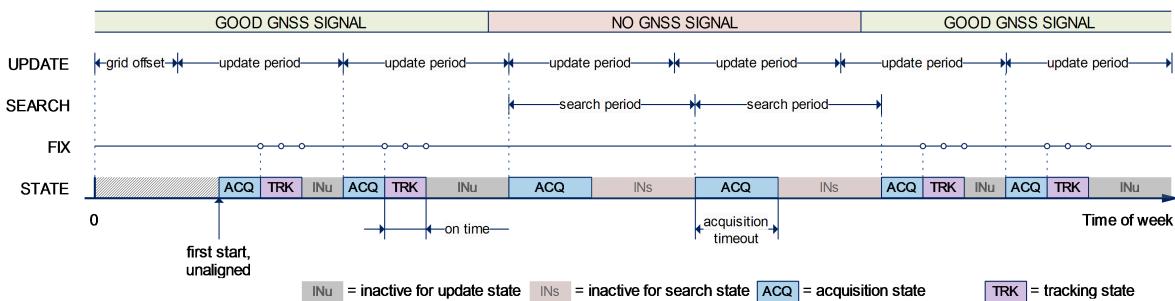
- *MINACQTIME* is the minimum time that the receiver will spend in *Acquisition* state (see MINACQTIME for details.)
- *MAXACQTIME* is the maximum time that the receiver will spend in *Acquisition* state (see MAXACQTIME for details).
- *DONOTENTEROFF* forces the receiver to stay awake and in *Acquisition* state even when a fix is not possible (see DONOTENTEROFF for details).

External mechanisms:

- The receiver will be forced to stay awake if *EXTINTWAKE* is enabled and the configured EXTINT pin is set to "high" and it will be forced to stay in *(Inactive) awaiting next search/Fix* states if *EXTINTBACKUP* is enabled and the configured EXTINT pin is set to "low" (see EXTINT pin control for details).
- The receiver will be forced to stay awake if *EXTINTINACTIVE* is enabled and the configured EXTINT is toggling and it will be forced to stay in *(Inactive) awaiting next search/Fix* states if configured EXTINT state is not changed for longer time than *EXTINTINACTIVITY* (see EXTINT pin control for details).

3.11.2.1.2 On/off operation - long update period
The diagram below illustrates how on/off operation works:
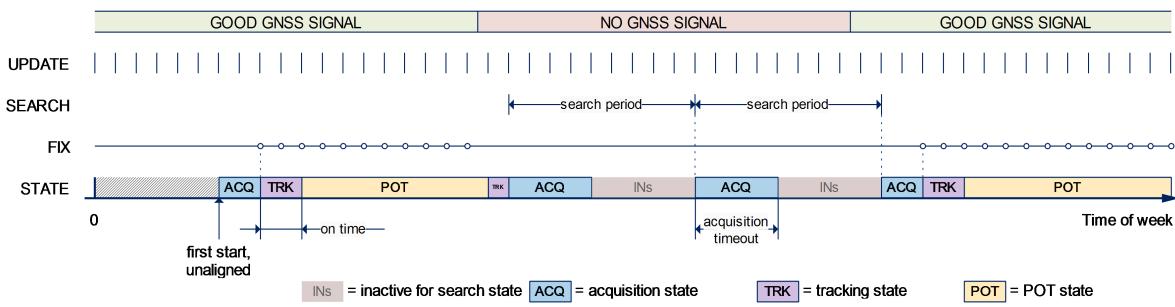
Figure 19: Diagram of on/off operation

- When the receiver is switched on, it first enters *Acquisition* state.
- If it is able to obtain a valid position fix within the time given by the Acquisition Timeout, it switches to *Tracking* state. Otherwise it enters *(Inactive) awaiting next search* state and re-starts after the configured search period (minus a startup margin).
- As soon as the receiver gets a valid position fix (one passing the navigation output filters), it enters *Tracking* state. Upon entering *Tracking* state, the ONTIME starts.
- Once the *ONTIME* is over, *(Inactive) awaiting next fix* state is entered and the receiver re-starts according to the configured update grid (see section Grid offset for an explanation).
- If the signal is lost while in *Tracking* state, *Acquisition* state is entered. If the signal is not found within the acquisition timeout, the receiver enters *(Inactive) awaiting next search* state. Otherwise the receiver will re-enter *Tracking* state and stay there until the newly started ONTIME is over.

3.11.2.1.3 Cyclic tracking operation - short update period
The diagram below illustrates how cyclic tracking operation works:


Figure 20: Diagram of cyclic tracking operation

- When the receiver is switched on, it first enters *Acquisition* state.
- If it is able to obtain a position fix within the time given by the acquisition timeout, it switches to *Tracking* state. Otherwise, it will enter *(Inactive) awaiting next search* state and re-start within the configured search grid.

- After a valid position fix, *Tracking* state is entered and the *ONTIME* starts. In other words the *ONTIME* starts with the first valid position fix.
- Once the *ONTIME* is over, *POT* state is entered.
- In *POT* state the receiver continues to output position fixes according to the CFG-RATE-*. To have maximum power savings, set the *ONTIME* to zero. This causes the receiver to enter *POT* state as soon as possible.
- If the signal becomes weak or is lost during *POT* state, *Tracking* state is entered.
- Once the signal is good again and the newly started *ONTIME* is over, the receiver will re-enter *POT* state.
- If the receiver cannot get a position fix in the *Tracking* state, it enters *Acquisition* state. Should the acquisition fail as well, *(Inactive) awaiting next search* state is entered. If *DONOTENTEROFF* is enabled and no fix is possible, the receiver will remain in *Acquisition* state until a fix is possible and it will never enter *(Inactive) awaiting next search* state.

3.11.2.1.4 User-controlled operation - update and search period of zero

Setting the CFG-PM-POSUPDATEPERIOD to zero causes the receiver to wait in the *(Inactive) awaiting next fix* state until woken up by the user. Setting the search period to zero causes the receiver to wait in the *(Inactive) awaiting next search* state indefinitely after an unsuccessful startup. Any wake-up event will re-start the receiver. See section Wake up for more information on wake-up events.

☞ External wake-up is required when setting update or search period to zero.

3.11.2.1.5 Satellite data download

The receiver is not able to download satellite data (e.g. the ephemeris) while it is working in on/off or cyclic tracking operation. Therefore it has to temporarily switch to continuous operation for the time the satellites transmit the desired data. To save power the receiver schedules the downloads according to an internal timetable and only switches to continuous operation while data of interest is being transmitted by the satellites.

Each satellite transmits its own ephemeris data. Ephemeris data download is feasible when the corresponding satellite has been tracked with a sufficient C/N0 over a certain period of time. The download is scheduled in a 30-minute grid or immediately when fewer than a certain number of visible satellites have valid ephemeris data.

Almanac, ionosphere, UTC correction and satellite health data are transmitted by all satellites simultaneously. Therefore these parameters can be downloaded when a single satellite is tracked with a high enough C/N0.

Allowing more ephemerides to be downloaded before going into *POT* or *(Inactive) awaiting next fix* state can help improve the quality of the fixes and reduce the number of wake ups needed to download ephemerides at the cost of extra time in *Acquisition* state (only when an inadequate number of ephemerides are downloaded from tracked satellites).

3.11.2.2 Configuration

Power save mode is enabled and disabled with CFG-PM-OPERATEMODE and configured with items in the CFG-PM group.

☞ When enabling power save mode, the receiver will be unable to download or process any SBAS data. Therefore, there is no benefit in enabling SBAS and it is recommended to disable it. SBAS support can be disabled in the CFG-SIGNAL group.

A number of parameters can be used to customize PSM to your specific needs. These parameters, which can be found in the CFG-PM group, are listed in the following table:

Config key	Description
OPERATEMODE	Receiver mode of operation
POSUPDATEPERIOD	Time between two position fix attempts in on/off power save mode
ACQPERIOD	Time between two acquisition attempts if the receiver is unable to get a position fix
MINACQTIME	Minimum time the receiver spends in *Acquisition* state
ONTIME	Time the receiver remains in *Tracking* state and produces position fixes
WAITTIMEFIX	Wait for time fix before entering *Tracking* state
DONOTENTEROFF	Receiver does not enter *(Inactive) awaiting next search* state if it cannot get a position fix but keeps indefinitely attempting a position fix instead
UPDATEEPH	Enables periodic ephemeris update
EXTINTSEL	Selects EXTINT pin used with pin control feature
EXTINTWAKE	Enables force-ON pin control feature
EXTINTBACKUP	Enables force-OFF pin control feature
EXTINTINACTIVE	Enables sleep on inactivity feature
EXTINTINACTIVITY	Specifies allowed inactivity period
GRIDOFFSET	Time offset of update grid with respect to start of week
MAXACQTIME	Maximum time in *Acquisition* state

Table 20: Power save mode configuration options in the CFG-PM group

3.11.2.2.1 Mode of operation (OPERATEMODE)
The mode of operation to use mainly depends on the update period: For short update periods (in the range of a few seconds), cyclic tracking should be configured. For long update periods (in the range of minutes or longer), only use on/off operation.

See section On/off operation - long update period and Cyclic tracking operation - short update period for more information on the two modes of operation.

3.11.2.2.2 Reference time standard
Where multiple GNSS can operate simultaneously, UTC time standard is used for the operation of PSM.

3.11.2.2.3 Update period (POSUPDATEPERIOD) and search period (ACQPERIOD)
The update period specifies the time between successive position fixes. If no position fix can be obtained within the acquisition timeout, the receiver will retry after the time specified by the search period. Update and search periods are fixed with respect to an absolute time grid based on reference time standard (i.e., GPS Time or UTC, see Reference Time Standard). They do not refer to the time of the last valid position fix or last position fix attempt.

☞ The update period (POSUPDATEPERIOD) setting is ignored if the receiver is set into cyclic tracking mode. It only has an impact if the receiver is set to on/off mode.

☞ New settings are ignored if the update period or the search period exceeds the maximum number of milliseconds in a week. In that case the previously stored values remain effective.

3.11.2.2.4 Minimum acquisition time (MINACQTIME)
The receiver tries to obtain a position fix for at least the time given in *MINACQTIME*. If the receiver determines that it needs more time for the given starting conditions then it will automatically prolong this time. If MINACQTIME is set to zero then the minimum acquisition time is exclusively determined by the receiver. Once the MINACQTIME has expired, the receiver will terminate the acquisition state if either a fix is achieved or if the receiver estimates that any signals received are insufficient (too weak or too few) for a fix to be possible.

MINACQTIME is applicable only when no or very poor GNSS signal is available.

3.11.2.2.5 On time (ONTIME)

The *ONTIME* parameter specifies how long the receiver stays in *Tracking* state before switching to the *POT* state (in PSMCT) or *(Inactive) awaiting next fix* state (in PSMOO).

3.11.2.2.6 Wait for time fix (WAITTIMEFIX)

A time fix is a fix type in which the receiver will ensure that the time is accurate and confirmed to within the limits set in CFG-NAVSPG. Enabling the *WAITTIMEFIX* option will force the receiver to stay in *Acquisition* state until the time is known to be within the configured limits, then it will transition to *Tracking* state. Take into account that enabling *waitTimeFix* will delay the transition from *Acquisition* state to *Tracking* state by at least two extra seconds (see Acquisition Timeout). It is necessary to enable *WAITTIMEFIX* in timing products.

The quality of the position fixes can also be configured by setting the limits in the CFG-NAVSPG group. Setting harder limits in CFG-NAVSPG will typically prolong the time in *Acquisition* state. Thus, ensuring sufficient time is given to the receiver at startup (when externally controlled) is necessary (see Acquisition Timeout Logic). When internally controlled, the receiver can make good judgment on the time needed in *Acquisition* state and no further adjustments will be needed.

3.11.2.2.7 Maximum startup state duration (MAXACQTIME)

The *MAXACQTIME* is the maximum time that the receiver will spend in *Startup* state (i.e., *Acquisition* state). If the receiver is unable to acquire a valid position fix within this maximum time, it will transition to *(Inactive) awaiting next search* state (if *DONOTENTEROFF* is disabled). Subsequently, the receiver will attempt to acquire another position fix according to the search period (see Update period (POSUPDATEPERIOD) and search period (ACQPERIOD)). If *MAXACQTIME* is set to zero, the receiver will autonomously determine the maximum time to spend in *Acquisition* state. Note that shorter settings (below about 45 s) will degrade an unaided receiver's ability to collect new Ephemeris data at low signal levels (see section Satellite data download).

3.11.2.2.8 Do not enter '(Inactive) awaiting next search' state when no fix (DONOTENTEROFF)

If this option is enabled, the receiver acts differently in case it cannot get a fix: instead of entering *(Inactive) awaiting next search* state, it keeps attempting to acquire a position fix. In other words, the receiver will never be in *(Inactive) awaiting next search* state and therefore searchPeriod and MINACQTIME will be ignored.

3.11.2.2.9 Update ephemeris (UPDATEEPH)

To maintain the ability of a fast startup, the receiver needs to update its ephemeris data on a regular basis. This can be ensured by activating the update Ephemeris option (*UPDATEEPH*). The ephemeris data is updated approximately every 30 minutes. See section Satellite data download for more information.

3.11.2.2.10 EXTINT pin control

The operation of PSM can be externally controlled using either EXTINT0 or EXTINT1 pin. This external control allows the user to decide when to wake up the receiver to obtain a fix and when to force the receiver into sleep/backup mode to save power. Operating the receiver externally through the EXTINT pins will override internal functions that coincide with that specific operation.

The choice of which pin to use can be configured through the EXTINTSEL feature in CFG-PM. Only one pin can be selected at a time but it is sufficient to perform all the required tasks.

If the Force-ON (*EXTINTWAKE*) feature in CFG-PM is enabled, the receiver will not enter Inactive states for as long as the configured EXTINT pin (EXTINT0 or EXTINT1) is at 'high' level. The receiver will therefore always be in *Acquisition/Tracking* state in PSMOO or in *Acquisition/Tracking/POT* state in PSMCT. When the pin level changes to 'low' the receiver will continue with its configured behavior.

If the Force-OFF (*EXTINTBACKUP*) feature in CFG-PM is enabled, the receiver will enter Inactive states for as long as the configured EXTINT pin is set to 'low' until the next wake-up event. Any wake-up event can wake up the receiver even while the EXTINT pin is set to 'low' (see Wake up). However, if the pin stays at 'low' state, the receiver will only wake up for the time needed to read the configuration pin settings, then it will enter the Inactive state again.

If both Force-ON and Force-OFF features are enabled at the same time, the receiver PSM operation will be completely in user's control. Setting 'high' on the configured EXTINT pin will wake up the receiver to get a position fix and setting 'low' will put the receiver into sleep/backup mode.

3.11.2.2.11 Grid offset (GRIDOFFSET)

Once the receiver has a valid time, the update grid is aligned to the start of the week of the reference time standard (midnight between Saturday and Sunday). Before having a valid time, the update grid is unaligned. A grid offset shifts the update grid with respect to the start of the week of the reference time standard. An example of usage can be found in section Use grid offset.

☞ The grid offset is not used in cyclic tracking operation.

3.11.2.3 Features

3.11.2.3.1 Communication

When PSM is enabled, communication with the receiver (e.g. UBX message to disable PSM) requires particular attention. This is because the receiver may be in *Inactive* state and therefore unable to receive any message through its interfaces. To ensure that the configuration messages are processed by the receiver, even while in *Inactive* state, the following steps need to be taken:

- Send a dummy sequence of 0xFF (one byte is sufficient) to the receiver's UART interface. This will wake up the receiver if it is in *Inactive* state. If the receiver is not in *Inactive* state, the sequence will be ignored.
- Send the configuration message about half a second after the dummy sequence. If the interval between the dummy sequence and the configuration message is too short, the receiver may not yet be ready. If the interval is too long, the receiver may return to *Inactive* state before the configuration message was received. It is therefore important to check for a UBX-ACK-ACK reply from the receiver to confirm that the configuration message was received. The configuration should be saved in a non-volatile layer (such as BBR or flash memory layer).

Similarly, when configuring the receiver for PSMOO (and PSMCT when *DONOTENTEROFF* is disabled), ensure that the configurations are set in non-volatile layer. If they are not in a non-volatile layer, the receiver will enter backup mode and when it wakes up again, it has lost the configuration and has even forgotten it was in power save mode. This can be avoided if the key values of the configuration items are saved into the BBR. If they are only saved into RAM, they will be lost after the first backup entry (see Receiver Configuration for details). When operating PSM from u-center and setting the receiver to power save mode with CFG-PM-OPERATEMODE, check which layer the setting is saved to.

3.11.2.3.2 Wake up

The receiver can be woken up by generating an edge on one of the following pins:

- rising or falling edge on one of the EXTINT pins
- rising or falling edge on the RXD pin
- rising edge on NRESET pin

All wake-up signals are interpreted as a position request, where the receiver wakes up and tries to obtain a position fix. Wake-up signals have no effect if the receiver is already in *Acquisition*, *Tracking* or *POT* state.

3.11.2.3.3 Behavior while USB host connected

As long as the receiver is connected to a USB host, it will not enter the lowest possible power state. This is because it must retain a small level of CPU activity to avoid breaching requirements of the USB specification. The drawback, however, is that power consumption is higher.

☞ Wake up by pin/UART is possible even if the receiver is connected to a USB host. In this case the state of the pin must be changed for a duration longer than one millisecond.

3.11.2.3.4 Cooperation with the AssistNow Autonomous feature

If both PSM and AssistNow Autonomous features are enabled, the receiver will not enter *(Inactive) awaiting next fix* state as long as *AssistNow Autonomous* carries out calculations. This prevents losing data from unfinished calculations and, in the end, reduces the total extra power needed for *AssistNow Autonomous*. The delay before entering *(Inactive) awaiting next fix* state, if any, will be in the range of several seconds, rarely more than 20 seconds.

Only entering *(Inactive) awaiting next fix* state is affected by *AssistNow Autonomous*. This means that: in cyclic tracking operation, *AssistNow Autonomous* will not interfere with the PSM (apart from the increased power consumption).

☞ Enabling the *AssistNow Autonomous* feature will lead to increased power consumption while prediction is calculated. The main goal of PSM is to reduce the overall power consumption. Therefore for each application special care must be taken to judge whether *AssistNow Autonomous* is beneficial to the overall power consumption or not.

3.11.2.4 Examples

3.11.2.4.1 Use grid offset

Scenario: Get a position fix once a day at a fixed time. If the position fix cannot be obtained try again every two hours.

Solution: First set the update period (*POSUPDATEPERIOD*) to 24*3600s and the search period (*ACQPERIOD*) to 2*3600s. Now a position fix is obtained every 24 hours and if the position fix fails retrials are scheduled in two-hour intervals. As the update grid is aligned to midnight Saturday/ Sunday reference time standard, the position fixes happen at midnight reference time standard. By setting the grid offset (*GRIDOFFSET*) to 12*3600s the position fixes are shifted to once a day at noon reference time standard. If the position fix at noon fails, retrials take place every two hours, the first at 14:00 reference time standard. Upon successfully acquiring a position fix the next fix attempt is scheduled for noon the following day.

3.11.2.4.2 User-controlled position fix

Scenario: Get a position fix on request.

Solution: Set *POSUPDATEPERIOD* and *ACQPERIOD* to zero. Set *EXTINTSEL* to the desired EXTINT pin to be used. Enable the *EXTINTWAKE* and *EXTINTBACKUP* features.

3.11.2.4.3 Use update periods of 30 minutes

Scenario: Get a position fix once every 30 minutes and acquire a fix needed for timing products.

Solution: Set mode of operation to PSMOO. Set *POSUPDATEPERIOD* to 1800 seconds. Set the search period to 120 seconds. Enable *WAITTIMEFIX* feature.

3.11.3 Peak current settings

The peak current during acquisition can be reduced by activating the corresponding option in CFG-PM-LIMITPEAKCURR. A peak current reduction will result in longer startup times of the receiver.

☞ This setting is independent of the activated mode (continuous or power save mode).

3.11.4 Power on/off command

With message UBX-RXM-PMREQ the receiver can be forced to enter *Inactive* state (in continuous and power save mode). It will stay in *Inactive* state for the time specified in the message or until it is woken up by an EXTINT or activity on the RXD, SPI CS, or NRESET pin.

☞ Sending the message UBX-RXM-PMREQ while the receiver is in power save mode will overrule PSM and force the receiver to enter *Inactive* state. It will stay in *Inactive* state until woken up. After wake-up the receiver continues working in power save mode as configured.

3.11.5 EXTINT pin control when power save mode is not active

The receiver can be forced OFF also when the power save mode is not active. This works the same way as EXTINT pin control in power save mode. Just as in power save mode, this feature has to be enabled and configured in CFG-PM.

3.11.6 Measurement and navigation rate with power save mode

In Continuous Mode, measurement and navigation rate is configured using CFG-RATE. In power save mode however, measurement and navigation rate can differ from the configured rates as follows:

• **Cyclic operation:** When in state *Power optimized tracking*, the measurement and navigation rate is given by the value given by the item *MEAS* in group CFG-RATE. PSM can only be enabled when *MEAS* is set to 500 ms or x*1000 ms with x =1...10. The receiver can however switch to *Tracking* state (e.g. to download data). Note: When the receiver is no longer able to produce position fixes, it can switch from cyclic operation to on/off operation (if this is not disabled with the *DONOTENTEROFF* switch in CFG-PM). In that case the remarks below are relevant.

• **On/off operation:** When in state *Acquisition*, the measurement and navigation rate is **fixed to 2 Hz**. All NMEA (and UBX) messages that are output upon a navigation fix are also output with a rate of 2 Hz. This must be considered when choosing the baud rate of a receiver that uses power save mode! Note that a receiver might stay in *Acquisition* state for quite some time (can be tens of seconds under weak signal conditions). When the receiver eventually switches to *Tracking* state, the measurement and navigation rate will be as configured with CFG-RATE. However, the measurement and navigation rate will be as configured with CFG-RATE in all active states.

3.11.7 Backup modes

When the receiver is set into backup mode, it consumes only a fraction of the normal power usage and when it is powered up again will be able to make a position fix within a very short time. This is the "hot start". The hot start fix time is given in the NEO-M9N data sheet [1].

The NEO-M9N supports two kinds of backup modes. These are:
• Hardware backup mode
• Software backup mode

The "hardware backup" mode is for designs where a battery supply is always present. For designs where the battery supply is not available, software backup and the save-on-shutdown feature can be used. See more information about the save-on-shutdown feature (SOS-feature) in the NEO-M9N Interface description [2].

The software backup mode is very similar to the "power save mode on/off operation" and can be enabled with the command described in Power on/off command.

3.12 Clocks and time

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

3.12.1 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 chosen GNSS system. 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 an aiding message, it will jump to an estimate of GNSS time.

3.12.2 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 **clock bias** (and **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. This means that 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 as GLONASS time is closely linked to UTC, conversion to UTC is easier.

When insufficient information is available for the receiver to perform any of these time base conversions precisely, predefined 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 use UBX messages reporting GNSS system time or receiver local time. It is recommended to use messages that report UTC time and other messages are retained only for backwards compatibility reasons.

3.12.3 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.

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).

3.12.4 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
QZSS time	UBX-NAV-TIMEQZSS
UTC time	UBX-NAV-TIMEUTC

Table 21: GNSS times

3.12.5 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 valid for use. These flags are shown in table GNSS times in section GNSS times 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 can 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.

3.12.6 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). 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.

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 must 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.

The UBX-NAV-TIMEUTC message gives information about UTC time reference clock.

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

3.12.7 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 or 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.

3.12.8 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.

3.12.9 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 seconds since the start of the week (midnight Saturday/Sunday). In order to map this to a full date, it is necessary to know the 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 cannot 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 simulated before these dates.

3.12.9.1 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 of 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.

3.13 Timing functionality
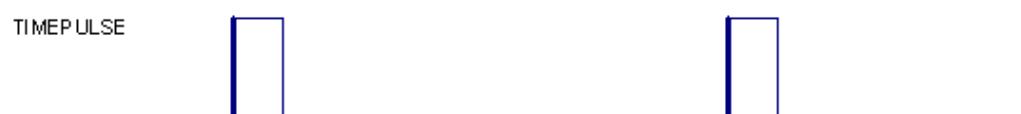
In addition to positioning and navigation applications, GNSS signals are widely used as low-cost precision time or frequency references used by remote or distributed wireless communication, industrial, financial, and power distribution equipment. By capitalizing on atomic clocks which are on-board positioning satellites, GNSS signals which contain embedded timing information can be used to synchronize equipment, as well as to provide UTC time. For wireless communication standards that utilize Time Division Multiplex (TDM) and applications such as femtocell base stations, a precision time reference is mandatory.

3.13.1 Time pulse

3.13.1.1 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 and the time source.
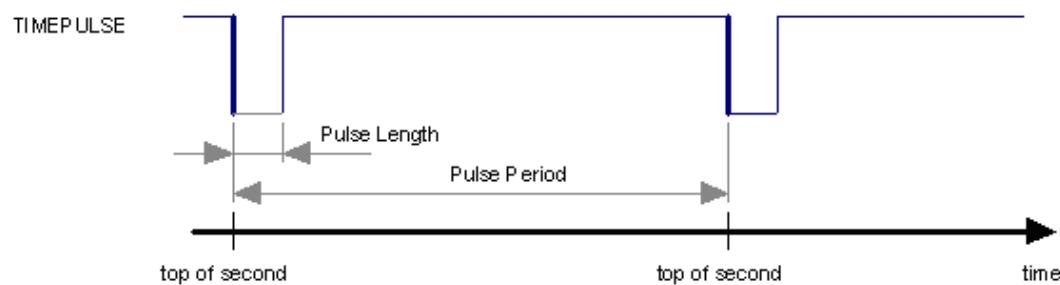

Figure 21: Time pulse

3.13.1.2 Recommendations

- The time pulse can be aligned to a wide variety of GNSS times or to variants of UTC derived from them (see the section 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 the setting of CFG-SIGNAL-* configuration group with the choice of time pulse time base.

- When using time pulse for timing applications requiring absolute time accuracy, e.g. with requirements specifying offset to UTC, it is recommended to calibrate the user's full setup for TP output against a reference timing source. To achieve best absolute and consistent accuracy (e.g. for mass deployment), it is recommended that the user should calibrate each single setup and calibrate under different GNSS modes and different temperatures which are applicable to the user's application and operating requirements. The user should take the calibrated values and configure the compensation accordingly (see the section on Time pulse configuration)

- 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 1 Hz.

☞ 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 time pulses T1 and T2 belong to T2. 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.
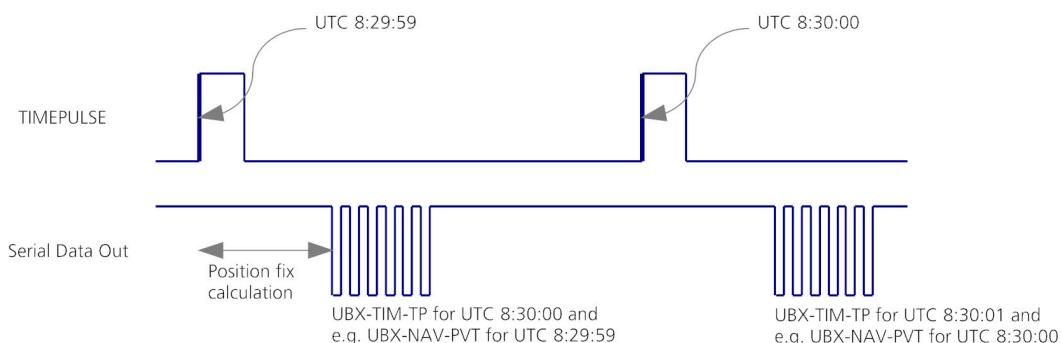
Figure 22: Time pulse and TIM-TP

3.13.1.3 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 EXTINT pins) and output (via the TIMEPULSE pin) and the parameters reported in corresponding messages.

The CFG-TP-TIMEGRID_TP* configuration item allows the user to choose between any of the supported GNSS (GPS, GLONASS, BeiDou, etc.) time bases and UTC. Also, the CFG-NAVSPG-UTCSTANDARD configuration item 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, European if Galileo is enabled, finally, NPLI if NAVIC is enabled.

The receiver will assume that an input time pulse uses the same GNSS time base as specified for the time pulse output. 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 independently choose 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.

3.13.1.4 Time pulse configuration

u-blox NEO-M9N receivers provide a time pulse (TIMEPULSE) signal with a configurable pulse period, length and polarity (rising or falling edge).

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 signal can be configured using the configuration group CFG-TP-*.

The NEO-M9N module provides only one time pulse pin. Configuration for TP1 applies to this pin.

3.13.1.5 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** - If this item is set, the time pulse is active.
- **frequency/period type** - Determines whether the time pulse is interpreted as frequency or period.
- **length/ratio type** - Determines whether the time pulse length is interpreted as length [us] or pulse ratio [%].
- **antenna cable delay** - Signal delay owning to RF components (e.g. antenna, cable, and splitter etc.) before the receiver input. This delay parameter affects the receiver calculation of GNSS time and it is used to compensate for the signal transit time prior to the receiver and any uncompensated delay from the receiver; a positive value compensates this delay, i.e. advances the time pulse
- **pulse frequency/period** - Frequency or pulse time period when locked mode is not configured or active.
- **pulse frequency/period lock** - Frequency or pulse time period, as soon as the receiver has calculated a valid time from a received signal. Only used if the corresponding item 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 the receiver has calculated a valid time from a received signal. Only used if the corresponding item is set to use another setting in locked mode.
- **user delay** - A time offset of the TP output for adjustment in a user application. It adjusts the time pulse position only with respect to GNSS time. Configuring a positive value will add a delay, i.e. retard the pulse with respect to GNSS time. Conversely, a negative value will advance the pulse. This configuration is available for all supported TP outputs.
- **lock to GNSS freq** - If this item is set, uses the frequency gained from the GNSS signal information rather than the local oscillator's frequency.
- **locked other setting** - If this item is set, the alternative setting will be used as soon as the receiver can calculate a valid time. This mode can be used, for example, to disable time pulse if the time is not locked, or to indicate a lock with different duty cycles.
- **align to TOW** - If this item 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 polarity: rising).
- **grid UTC/GNSS** - Selection between UTC (0), GPS (1), GLONASS (2), BeiDou (3) and (4) Galileo timegrid. Also affects the time output by UBX-TIM-TP message.

☞ The maximum pulse length cannot 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.

3.13.1.5.1 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** = 1 000 000 µs
- **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** = 1 000 000 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.
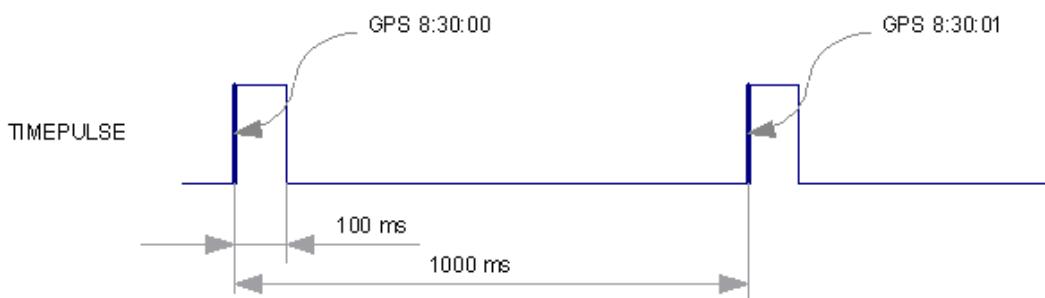
Figure 23: Time pulse signal with the example parameters

3.13.2 Time mark

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, and
- 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 time mark, new rising/falling edge indicator, time source, validity, number of marks and an accuracy estimate.

☞ 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).
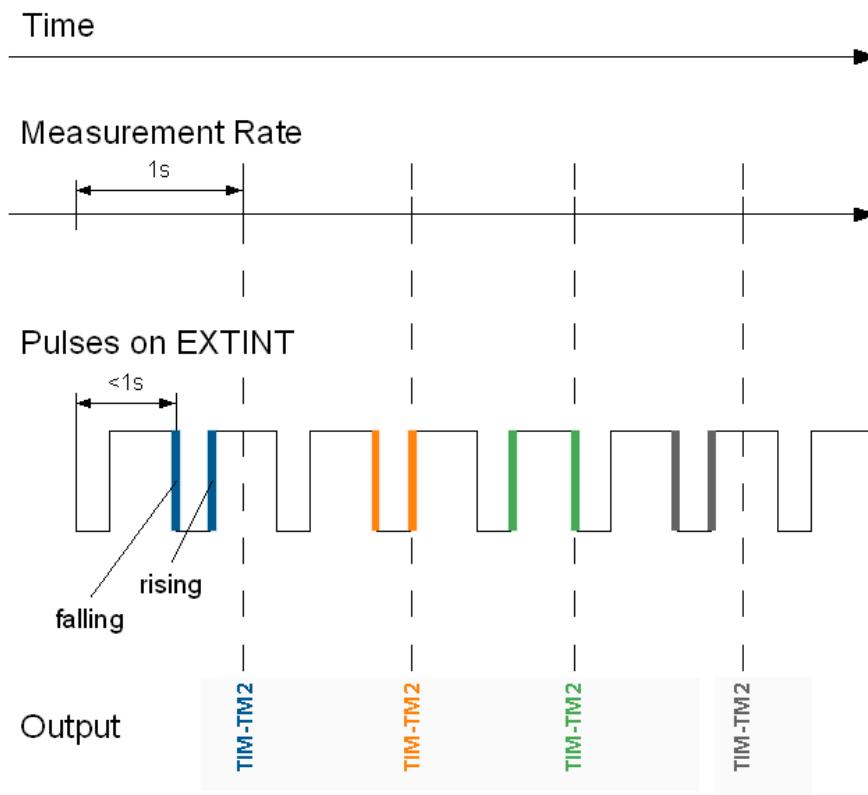
Figure 24: Time mark

3.14 Security

The security concept of NEO-M9N covers:

- the security of the receiver
- communication between the receiver and the GNSS satellites

Some receiver security functions monitor and detect threats and report them to the host system. Other security functions mitigate threats and allow the receiver to operate normally.

The table below gives an overview about possible threats and which functionality is available to detect and/or mitigate it.

Threat	u-blox solution
Over air signal security	Spoofing detection and monitoring
	Jamming interference detection and monitoring
GNSS receiver security	Secure boot
	Secure firmware update

Threat	u-blox solution
	Receiver configuration lock

Table 22: u-blox security options

3.14.1 Spoofing detection and monitoring

Spoofing is the process where a counterfeit GNSS signal is transmitted locally to deceive the receiver/user and produce an erroneous position fix and/or time solution.

The detection algorithm monitors GNSS signals for implausible changes or inconsistencies.These are evaluated with regards to spoofing.

A detection is successful when a signal is observed to transition from an initially genuine one to a spoofed version. Hence detection is not possible if the receiver is started under spoofing conditions. The detection algorithms also rely on availability of signals from multiple GNSS constellations to improve the spoofing detection capabilities.

3.14.2 Jamming interference detection and 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 threshold for the unjammed case. If the value rises significantly above this threshold, this indicates that a continuous wave jammer is present.

This monitoring function is always enabled.

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

3.14.2.1 Jamming and 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 23: Jamming and interference monitor reported states

The monitor is disabled by default. The monitor is enabled by setting the CFG-ITFM-ENABLE configuration item. 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 a passive antenna.

☞ The monitoring 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 reports any currently detected interference over all currently configured signal bands.

3.14.3 GNSS receiver security

3.14.3.1 Secure boot

The NEO-M9N boots only with firmware images that are signed by u-blox. This prevents the execution of non-genuine firmware images on the receiver.

3.14.3.2 Secure firmware update

The firmware image is signed by u-blox. The NEO-M9N verifies the signature during the firmware update.

3.14.3.3 Receiver configuration lock

The receiver configuration lock feature ensures that no configuration changes are possible once the feature is enabled. The configuration lock is enabled by setting the configuration item CFG-SEC-CFG_LOCK to "true".

The configuration lock can be applied to different configuration layers including the RAM, BBR, and flash memory. At startup, the receiver constructs the configuration database from different configuration layers and maintains it in the run-time RAM memory. When the configuration lock is set in the run-time RAM, the receiver configuration cannot be changed on any configuration layer.

☞ For more information on the configuration layers including the order of priority they are applied in, see the applicable interface description [2].

The configuration lock set on a configuration layer in volatile memory (RAM, BBR) is removed when the memory is cleared. However, the configuration lock set in non-volatile memory (flash memory) is permanent apart from one exception: during firmware upload to flash memory, the flash is erased during the process causing the configuration lock to be cleared. Refer to Firmware upload for more information on firmware update.

To test the lock functionality, set it on the RAM configuration layer. After a power cycle, the information on RAM layer is cleared and the lock is no longer set.

☞ It is recommended to apply the configuration lock on the same layer the configuration is stored.

An example of use case is that the host application locks the receiver configuration. A user communicating with the NEO-M9N through any of the available interfaces can poll, enable or send messages, but cannot change the configuration by sending UBX configuration messages.

3.15 u-blox protocol feature descriptions

3.15.1 Broadcast navigation data

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

UBX-RXM-SFRBX reports the broadcast navigation data message the receiver has collected from each tracked signal. When enabled, a separate message is generated each 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. This document uses the term "subframe", but other GNSS data structures might use different terms, for example, GLONASS uses "strings" and Galileo uses "pages".

3.15.1.1 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-RXM-SFRBX 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. Because of this, the following sections are organized by GNSS. 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).

3.15.1.2 GPS

NEO-M9N is designed to receive and track the L1C/A signals provided at 1575.42 MHz by the Global Positioning System (GPS).

3.15.1.2.1 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:
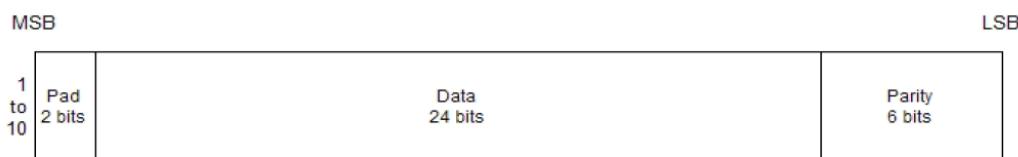
Figure 25: GPS L1C/A subframe word

3.15.1.3 GLONASS

The receivers are designed to receive and track the L1OF signals GLONASS provides at 1602 MHz +k*562.5 kHz, where k is the satellite's frequency channel number (k = -7,...,5, 6). The ability to receive and track GLONASS L1OF satellite signals allows design of GLONASS receivers where required by regulations.

The GLONASS words are arranged as follows:
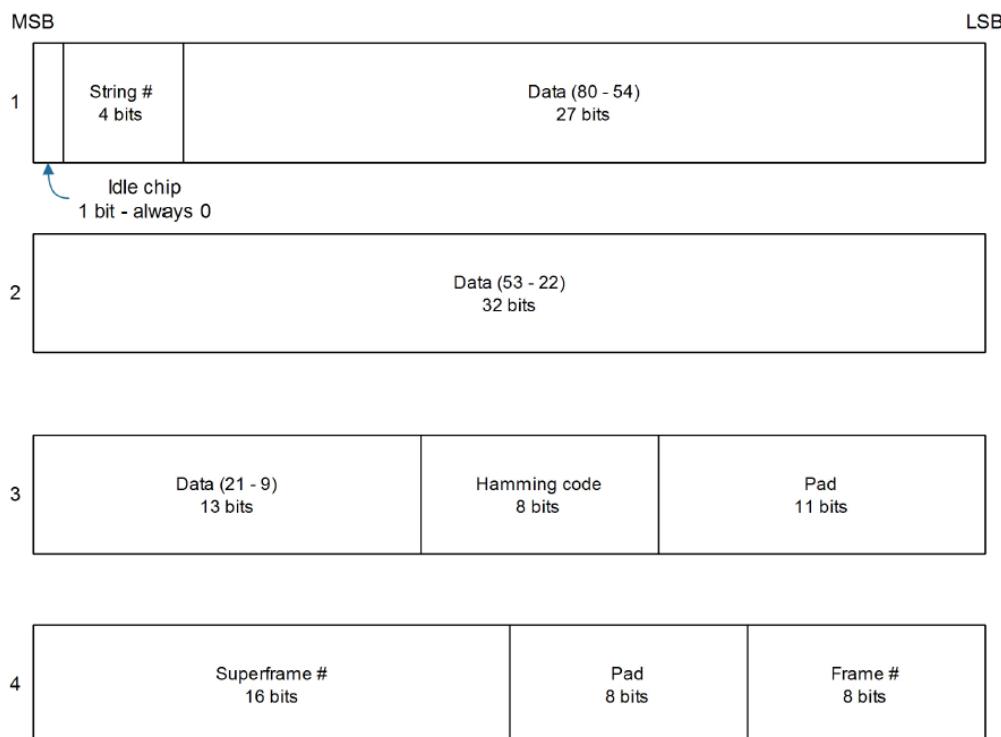
Figure 26: GLONASS navigation message data

3.15.1.4 BeiDou

u-blox M9 receivers can receive and process the B1I signals broadcast at 1561.098 MHz from the BeiDou Navigation Satellite System. The ability to receive and track BeiDou signals in conjunction with another constellation results in higher coverage, improved reliability and better accuracy.

Each word is arranged as follows:
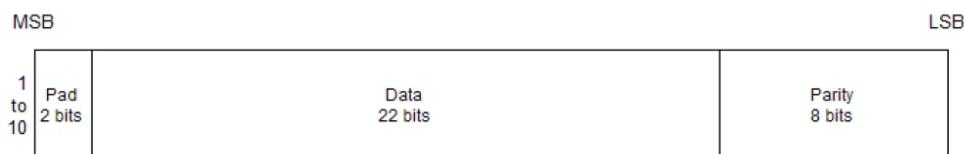
Figure 27: BeiDou subframe word
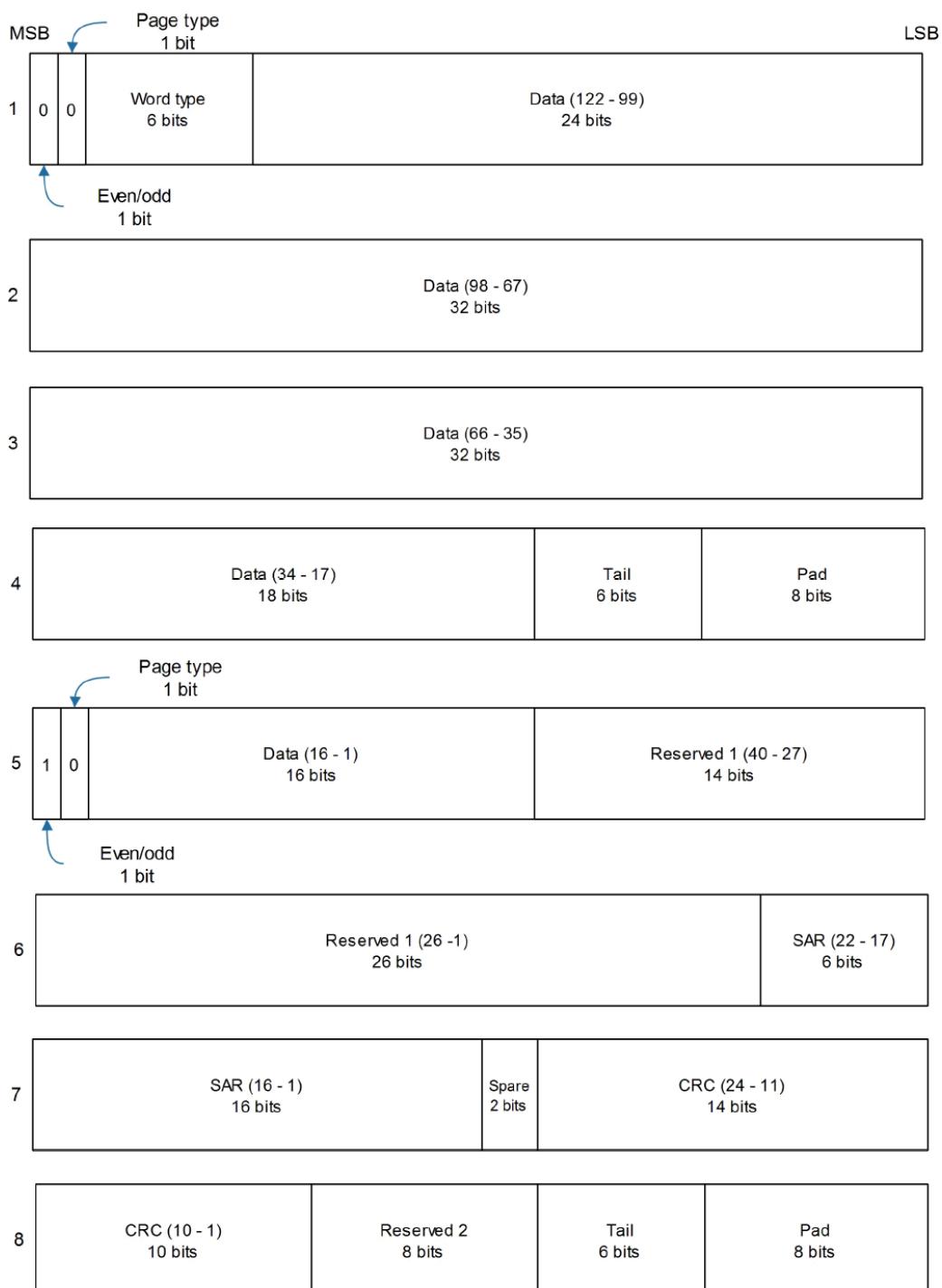
Note that as the BeiDou data words only comprise 30 bits, the 2 most significant bits in each word reported by UBX-RXM-SFRBX are padding and should be ignored.

3.15.1.5 Galileo

u-blox M9 receivers running can receive and track the E1-B/C signals centered on the GPS L1 frequency band. GPS and Galileo signals can be processed concurrently together with BeiDou and GLONASS signals, enhancing coverage, reliability and accuracy. The SAR return link message (RLM) parameters for both short and long versions are decoded by the receiver and made available to users via UBX proprietary messages and via standard NMEA RLM sentences.

3.15.1.5.1 Galileo E1-B

For the Galileo E1-B signal, 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 Galileo "Nominal" pages the eight words are arranged as follows:

Figure 28: Galileo E1-B 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).

### 3.15.1.6 SBAS

For SBAS (L1C/A) signals each reported subframe contains eight 32-bit data words to deliver the 250 bits transmitted in each SBAS data block.

The eight words are arranged as follows:
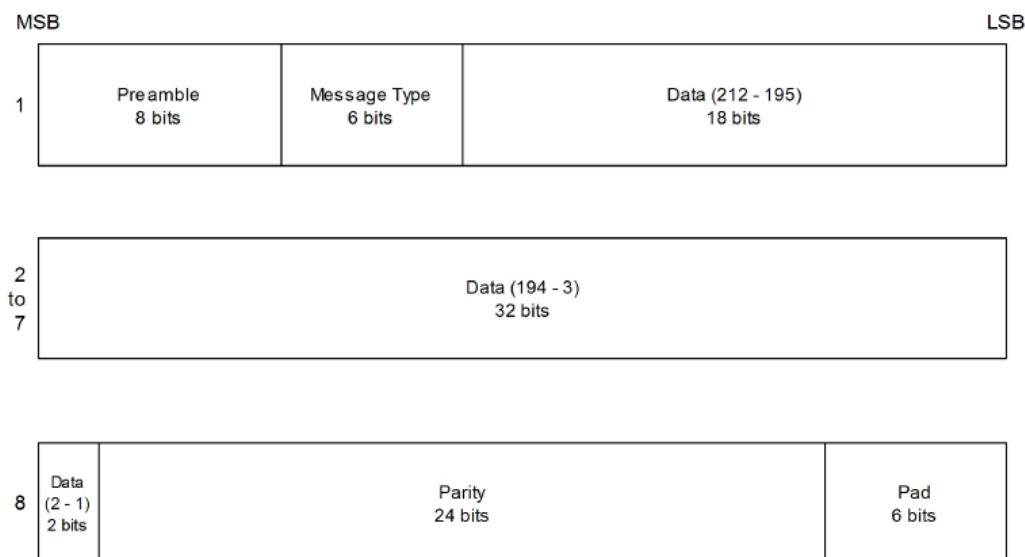
Figure 29: SBAS subframe words

3.15.1.7 QZSS

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

3.15.1.8 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
Galileo	E1 B	2	1	8	2s
BeiDou	B1I D1	3	0	10	6s
BeiDou	B1I D2	3	1	10	0.6s
QZSS	L1C/A	5	0	10	6s
GLONASS	L1OF	6	0	3	2s

Table 24: Data message formats reported by UBX-RXM-SFRBX

3.16 Forcing a receiver reset

Typically, in GNSS receivers, a distinction is made between cold, warm, and hot start, 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 Multiple 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.

Using the UBX-CFG-RST message, you 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.

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, to electrically reset the chip. This is an immediate, asynchronous reset. No Stop event is 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.
- **Hardware reset (after shutdown)** uses the on-chip watchdog. This is a reset after shutdown.
- **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.

3.17 Firmware upload

NEO-M9N is supplied with firmware. 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.

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.

Contact u-blox for more information on firmware update.

4 Design

This section provides information to help carry out a successful schematic and PCB design integrating the NEO-M9N.

4.1 Pin assignment

The pin assignment of the NEO-M9N module is shown in Figure 30. The defined configuration of the PIOs is listed in Table 25.

```
13  GND              GND  12
14  LNA_EN         RF_IN  11
15  Reserved         GND  10
16  Reserved      VCC_RF  9
17  Reserved     RESET_N  8
          NEO-M9N
          Top View
18  SDA / SPI CS_N  V_USB  7
19  SCL / SPI SLK  USB_DP  6
20  TXD / SPI MISO USB_DM  5
21  RXD / SPI MOSI EXTINT  4
22  V_BCKP      TIMEPULSE  3
23  VCC            D_SEL  2
24  GND       SAFEBOOT_N  1
```

Figure 30: NEO-M9N pin assignment

Pin no.	Name	I/O	Description
1	SAFEBOOT_N	I	SAFEBOOT_N (used for FW updates and reconfiguration, leave open)
2	D_SEL	I	Interface select (open or VCC = UART + I2C; GND = SPI)
3	TIMEPULSE	O	TIMEPULSE (1 PPS)
4	EXTINT	I	EXTINT (PIO 7)
5	USB_DM	I/O	USB data (DM)
6	USB_DP	I/O	USB data (DP)
7	V_USB	I	USB supply
8	RESET_N	I	RESET (active low)
9	VCC_RF	O	Voltage for external LNA
10	GND	I	Ground
11	RF_IN	I	GNSS signal input
12	GND	I	Ground
13	GND	I	Ground
14	LNA_EN	O	Antenna/LNA control
15	Reserved	-	Reserved

Pin no.	Name	I/O	Description
16	Reserved	-	Reserved
17	Reserved	-	Reserved
18	SDA / SPI CS_N	I/O	I2C data if D_SEL = VCC (or open); SPI chip select if D_SEL = GND
19	SCL / SPI SLK	I/O	I2C clock if D_SEL = VCC (or open); SPI clock if D_SEL = GND
20	TXD / SPI SDO	O	UART output if D_SEL = VCC (or open); SPI SDO if D_SEL = GND
21	RXD / SPI SDI	I	UART input if D_SEL = VCC (or open); SPI SDI if D_SEL = GND
22	V_BCKP	I	Backup voltage supply
23	VCC	I	Supply voltage
24	GND	I	Ground

Table 25: NEO-M9N pin assignment

4.2 Power supply

The u-blox NEO-M9N module has three power supply pins: VCC, V_BCKP and V_USB.

(The VCC_RF pin is an output pin that can supply an active antenna or external LNA.)

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 applicable 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 startup, u-blox NEO-M9N 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 a built-in 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 NEO-M9N 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.

Maintaining time and GNSS orbit data improves the receiver performance at startup. Especially if the RTC and BBR contents are still current, i.e. after a short VCC outage.

If V_BCKP is not provided, the module performs a cold start at power up.

If a host is connected to NEO-M9N, V_BCKP can be partially emulated by using UBX-UPD-SOS functionality. BBR data can saved to the host and restored at startup. See the applicable 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 or connect to a high impedance in HW backup mode (V_BCKP supplied when VCC is removed). See the Interfaces section. It is essential that the interfaces and PIOs are isolated with open-drain buffers when VCC is removed to prevent the chip from being powered through these circuits.

In software backup mode the receiver remains supplied and backup is handled with software commands. See the NEO-M9N Interface description [2] for more information.

4.2.3 V_USB: USB interface power supply

V_USB supplies the USB interface. If the USB interface is not used, the V_USB pin must be connected to GND. For more information about correctly handling the V_USB pin, see Interfaces section.

4.3 NEO-M9N minimal design

The minimal electrical circuit for NEO-M9N operation using the UART1 interface is shown in Figure 31 below.
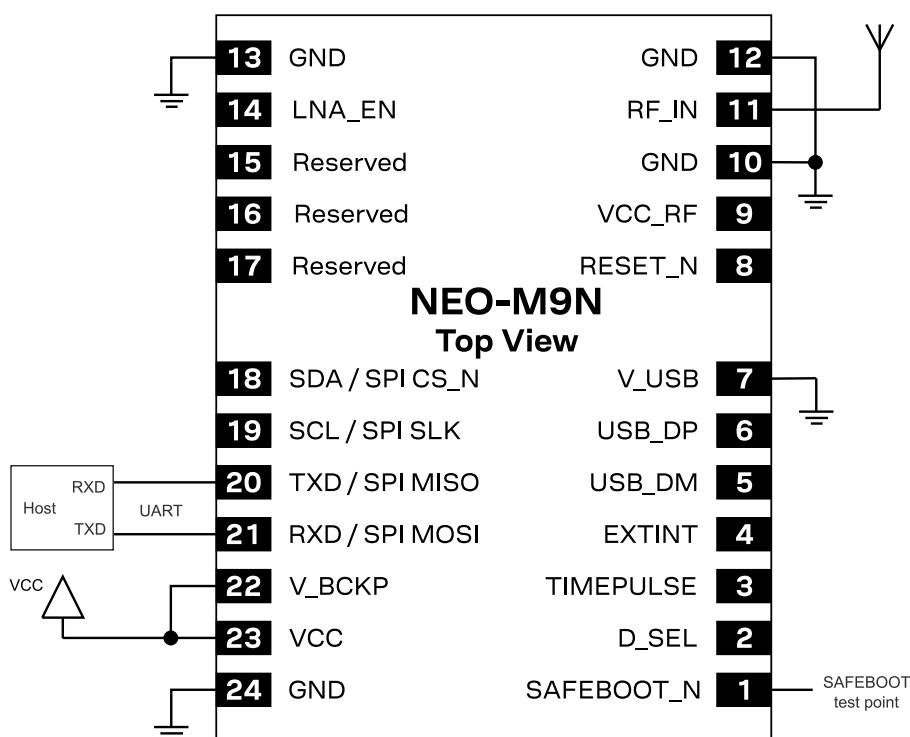
**Figure 31: Minimal NEO-M9N design**

For a minimal design with the NEO-M9N 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 NEO-M9N GNSS module.

- Choose the required serial communication interfaces (UART, USB, SPI or I2C) and connect the appropriate pins to your application. If SPI is used D_SEL must be connected to ground.
- The NEO-M9N module includes an LNA enabling the receiver to work well with a simple passive antenna.
- SAFEBOOT_N pin can be used to enable future firmware update.

4.4 NEO-M9N internal components

The NEO-M9N includes all necessary internal components for a complete GNSS solution.

4.4.1 Clock generation

The NEO-M9N includes a stable and clean TCXO oscillator for frequency generation to provide good GNSS performance.

4.4.2 Real-time clock (RTC)

The NEO-M9N module includes a real-time clock (RTC). The RTC section is located in the backup domain and used to maintain time in the event of power failure at the main supply (VCC). In the case of a power failure at VCC, the backup domain is supplied by V_BCKP. The RTC enables hot start, warm start, AssistNow Autonomous, AssistNow Offline and on/off operation of the power save mode.

4.4.2.1 Time aiding feature

Current time can also be sent by UBX message on startup of the receiver. If time will be aided after restarting the receiver, e.g. wake-up from backup mode, then the time kept by the internal RTC is not used. To enable hot starts, the time information must be provided accurately and thus the time mark feature must be used. The time aiding and time mark features are described in the NEO-M9N Interface description [2].

☞ Time aiding cannot be applied when using the power save mode operations, because the host does not have any information when the GNSS receiver turns from OFF status to ON status during on/off operation of the power save mode.

4.4.3 SAW and LNA

The NEO-M9N module includes a SAW filter, an LTE band 13 notch filter and a low-noise amplifier (LNA) to reduce external RF intereference and to improve GNSS signal reception sensitivity.

4.4.4 VCC_RF antenna bias voltage

The NEO-M9N module can provide filtered power to external antenna or external LNA via the VCC_RF output pin.

4.4.5 Flash memory

NEO-M9N includes an internal flash memory that can be used to store the firmware image, save logging data or save the AssistNow Offline data. How to enable these services and functions is described in the corresponding chapters of the Receiver functionality.

4.5 Antenna

GNSS signals operate with very low signal levels, ranging from -130 dBm to about -167 dBm. This is already a quite challenging parameter for GNSS receiver performance. In addition to that out-of-band interferers such as GSM, CDMA, WCDMA, LTE, Wi-Fi or Bluetooth wireless systems have a much higher signal level. This makes the receiver layout even more challenging.

Performance of the GNSS antenna and the related front-end RF signal path components have a decisive impact on the performance of the overall GNSS application. The NEO-M9N module includes an LTE band 13 notch filter, a SAW filter and an LNA to reduce external RF intereference and to improve GNSS signal reception sensitivity. The SAW filter is before the LNA in the RF path. This decreases sensitivity to out-of-band interference and also prevents the LNA from getting into saturation by very strong interferers.

The included LNA and SAW filter enable the NEO-M9N module to work well with an inexpensive passive antenna. For applications where passive antenna performance is not sufficient, the module supports controlling and supplying an external LNA or active antenna.

The NEO-M9N has an internal DC block and 50 Ω impedance matching for GNSS signal input, so there is no need to add these components to the RF path.

The antenna must have sufficient bandwidth to receive all needed GNSS constellations[2], and 50 Ω impedance at GNSS frequencies.

4.5.1 Antenna design with passive antenna

A design using a passive antenna requires attention to the layout of the RF section. Typically, a passive antenna is located near electronic components; therefore, care should be taken to reduce electrical noise that may interfere with the antenna performance. The antenna must have sufficient bandwidth to receive all needed GNSS constellations. Also, the antenna should have a suitable ground plane to achieve good performance. Passive antennas do not require a DC bias voltage and can be directly connected to the RF input pin RF_IN.

The example circuit in Figure 32 shows a setup for a design with a good passive GNSS antenna.
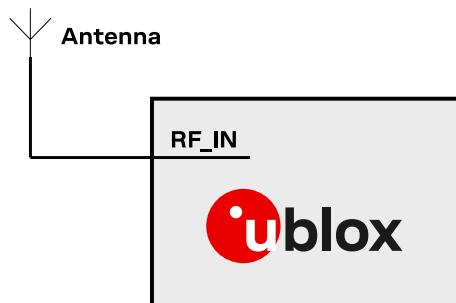
Figure 32: Passive antenna example circuit
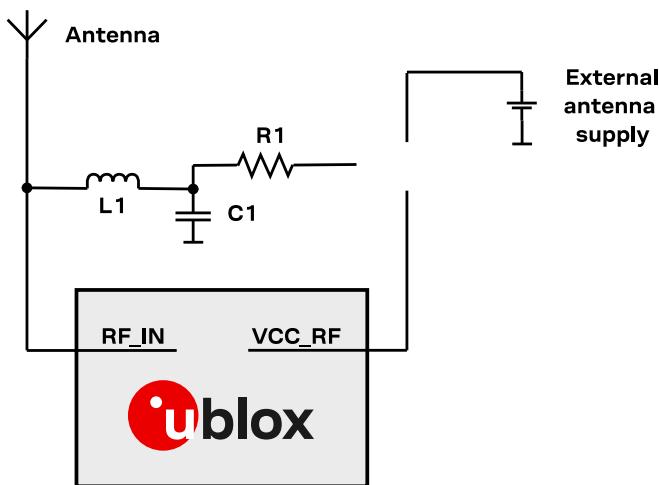
4.5.2 Antenna design with external LNA or active antenna

Passive antenna may be inadequate if the antenna gain is low or if the antenna must be placed far away from the module. For such designs the NEO-M9N module supports controlling an additional external LNA or an active antenna. The antenna control signal pin LNA_EN is active high to enable the external LNA or the active antenna when the receiver needs GNSS signal. If needed, the VCC_RF output pin can be used to supply filtered DC bias voltage to these external components. In this case, the supply voltage of the NEO-M9N module must match the external components' working voltage (for example, 3.3 V). If the VCC_RF voltage does not match with the supply voltage of the active antenna, use a filtered external supply.

The external LNA or the active antenna with an integrated LNA requires a power supply that contributes an additional current of typically 5 to 20 mA to the system's power consumption budget.

---

2   GNSS L-band frequencies: 1559…1563 MHz; 1573…1578 MHz; 1598…1606 MHz;

Active antennas for GNSS application are usually powered through a DC bias on the RF cable. A simple bias-T as in Figure 33 can be used to add this DC current to the RF signal line. The bias voltage can be supplied either by the VCC_RF pin of the module or by an external antenna supply.


Figure 33: NEO-M9N active antenna example circuit (bias-T)

Part	Recommended value	Description
C1	X7R 100 nF 10% 16 V	Bias-T capacitor removes high frequency noise coming from the DC supply.
L1	27 nH, impedance > 500 Ω at GNSS frequency, rated current > 300 mA	Bias-T inductor isolates RF from the DC path.
R1	10 Ω 10% 0.25 W	Current limiter series resistor in the event of a short circuit.

Table 26: Components in active antenna example circuit (bias-T)

Design recommendations:

- If the application has a passive antenna with sufficient gain but the antenna is located far away from the module, an additional external LNA must be placed close to the passive antenna.
- If the application needs an active antenna, it is recommended to use an OEM active antenna module meeting u-blox's specification.
- A suitable ground plane is required for the antenna to achieve good performance.
- Active antenna may have strong gain. For maximum external gain, see the Data sheet [1] in Related documents.
- A series current limiting resistor is required to prevent short circuits destroying the bias-t inductor.
- If the VCC_RF voltage of the NEO-M9N module does not match the supply voltage of the active antenna, use a filtered external supply instead of the VCC_RF.
- The antenna circuit should include filtering to ensure adequate protection from the nearby transmitters' interface. Take care in designing the circuit of antennas placed close to cellular or Wi-Fi transmitting antennas.
- An ESD protection diode should also be connected to the input.

It is recommended to use active current limiting. If active current limiting is not used, take into account the following important points:

☞ The bias-t inductor and current limiting resistor must be selected to be reliable with a short circuit on the antenna feed. In cases 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 short circuit conditions. For example, a value of 19 Ω or more is required at a module supply of 3.3 V 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.

☞ The power dissipation in the resistor and inductor needs to be taken into account based on the supply voltage and the short circuit current.

4.5.3 Out-of-band blocking immunity

Out-of-band RF interference may degrade the quality and availability of the navigation solution. Out-of-band immunity limit describes the maximum power allowed at the receiver RF input with no degradation in performance. Minor violation of the immunity limit may reduce C/N0 of the received signals but does not necessarily affect the overall receiver performance. However, a significant violation may reduce receiver sensitivity or cause a complete loss of signal reception. The severity of the interference depends on the repetition rate, frequency, signal level, modulation, and bandwidth of the signal.

Figure 34 shows a typical out-of-band immunity level for the NEO-M9N RF input. The measurement is done at room temperature using a continuous-wave (CW) test signal. The immunity limit is at constant +13 dBm apart from immediate proximity of the GNSS in-band at 1440 MHz. The integrated LTE band 13 notch and the SAW filters protect the RF input at 787 MHz region. For wide-band signals such as an LTE signal, the immunity may be reduced.

☞ It is recommended to verify that the receiver performance is not affected or is at an acceptable level in the presence of interference.
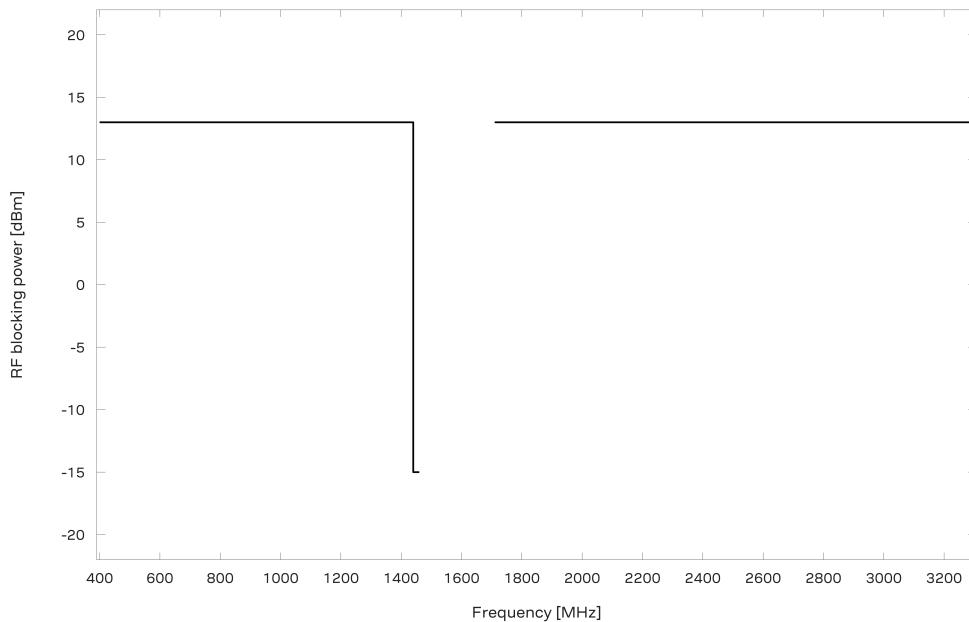

Figure 34: NEO-M9N out-of-band immunity level at 400 - 1460 MHz and 1710 - 3300 MHz. CW test signal is used.

4.6 EOS/ESD precautions

⚠ 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 applicable Data sheet [1].

When integrating GNSS receivers into wireless systems, pay special attention 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 help in designing robust yet cost-effective solutions.

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.
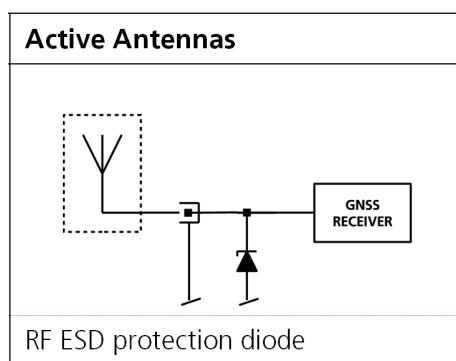
Figure 35: RF ESD precautions

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 is 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.
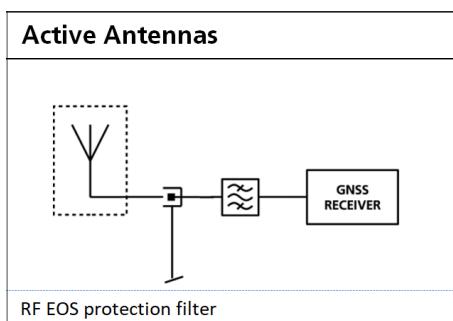
Figure 36: Active antenna EOS protection

4.6.3 Safety precautions

The NEO-M9N 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 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 is 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 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, such as 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 2 W (+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 to prevent damaging the receiver from large input signals. Here the GNSS performance under interference conditions is not important and suppression of the signal is permitted. It is sufficient to just observe the maximum RF power ratings of all of the components in the RF input path.
- GNSS performance must be guaranteed even under interference conditions. In such a case, not only the maximum power ratings of the components in the receiver RF path 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 such as 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 an 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, such as 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 NEO-M9N module.

4.8.1 Placement

GNSS signals at the surface of the Earth are 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.

4.8.2 Thermal management

During design-in do not place the receiver near sources of heating or cooling. The receiver oscillator is sensitive to sudden changes in ambient temperature which can adversely impact satellite signal tracking. Sources can include co-located power devices, cooling fans or thermal conduction via the PCB. Take into account the following questions when designing in the receiver.

- Is the receiver placed away from heat sources?
- Is the receiver placed away from air-cooling sources?
- Is the receiver shielded by a cover/case to prevent the effects of air currents and rapid environmental temperature changes?

⚠ High temperature drift and air vents can affect the GNSS performance. For best performance, avoid high temperature drift and air vents near the receiver.

4.8.3 Package footprint, copper and paste mask

Copper and solder mask dimensioning recommendations for the NEO-M9N module packages are provided in this section.

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

Refer to the NEO-M9N data sheet [1] for the mechanical specification.

4.8.3.1 Footprint
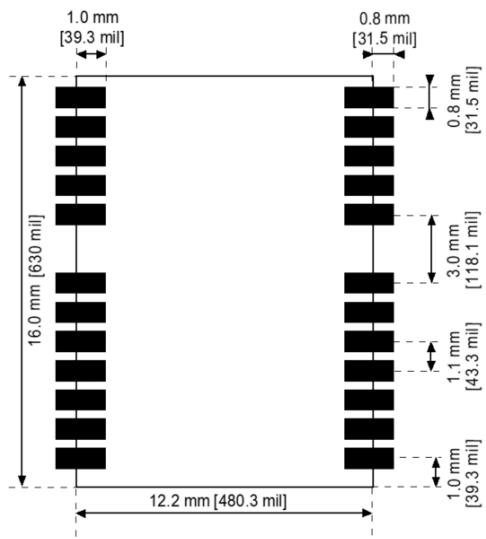
Figure 37: NEO-M9N suggested footprint (i.e. copper mask)

4.8.3.2 Paste mask
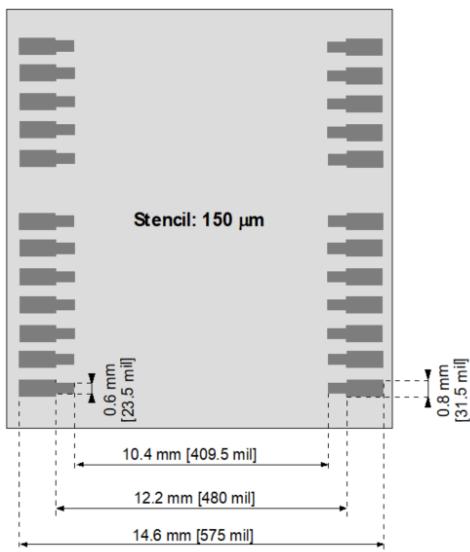
Figure 38: NEO-M9N suggested paste mask

4.8.4 Layout guidance

The presented layout guidance reduces the risk of performance issues at design level.

4.8.4.1 RF In trace

The RF In trace has to work in the middle L-band frequencies.

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

Focus on these layout guidelines:

- RF input co-planar waveguide referenced to layer 2, matched to 50 Ω.

- Place RF input matching component close to RF_IN pin.
- Use vias to 3D grounding at ground plane.
- No stubs must occur on the RF input trace.

The RF trace must be shielded by vias to ground along the entire length of the trace and the NEO-M9N RF_IN pad should be surrounded by vias as shown in the figure below.
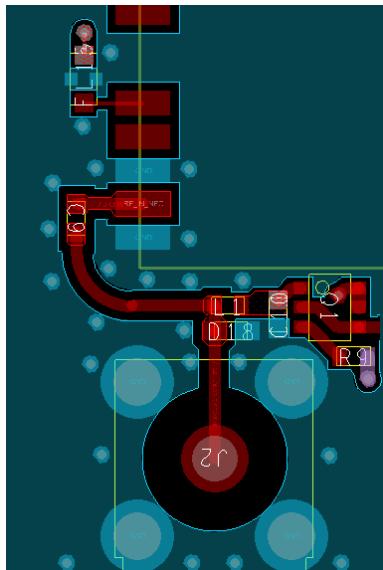
Figure 39: RF input trace

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

4.8.4.2 VCC pad

The VCC pad for the NEO-M9N module needs to have as low an impedance as possible with large vias to the lower power layer of the PCB. The VCC pad needs a large pad and the decoupling capacitor must be placed as close as possible. This is shown in the figure below.
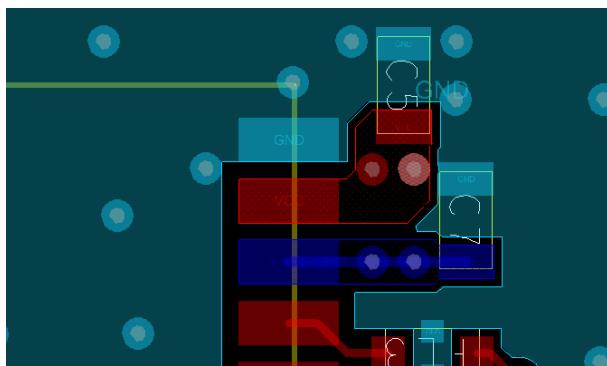
Figure 40: VCC pad

4.9 Design guidance

4.9.1 General considerations

Check power supply requirements and schematic:

- Is the power supply voltage within the specified range and noise-free?

- If USB is not used, connect the V_USB pin to ground.
- For USB devices: Is the V_USB voltage within the specified range?
- It is recommended to have a separate LDO for V_USB that is enabled by the module VCC. This is to comply with the USB self-powered specification.
- If USB is used, is there a 1 uF capacitor right near the V_USB pin? This is just for the V_USB pin.
- Is there a 1 uF cap right next to the module VCC pin?
- Compare the peak current consumption of the NEO-M9N 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. See the NEO-M9N Power supply section in the Design chapter for more information on the power supply requirements.
- Allow all I/O to Float/High impedance (High-Z) when VCC is not applied.

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 that your battery backup supply can provide the battery backup current specified in the applicable data sheet.
- Allow all I/O including UART and other interfaces to Float/High impedance in HW backup mode (battery backup connected with VCC removed).

4.9.3 RF front-end circuit options

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

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 any optional filtering components meet the group delay variation requirements?

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: Can the user access the antenna connector?
- 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 cellular transceivers, interference signals may exceed the maximum power rating of the RF_IN 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 is mandatory to avoid device destruction.

ESD discharge cannot be avoided during assembly and / or field use. Note that SAW filters are susceptible to ESD damage. To provide additional robustness an ESD protection diode may be placed at the antenna RF connector to GND.

4.9.4 Antenna/RF input

Check RF input requirements and schematic:

- Make sure the antenna is not placed close to noisy parts of the circuitry and does not face any other noisy elements (for example microcontroller, display).
- ESD protection on the RF input is mandatory.
- Possible external antenna bias-t inductor must be L-band frequency selected with high impedance in the GNSS band.
- Ensure RF trace is tuned for 50 Ω to ensure proper GNSS bandwidth.

4.9.5 Ground pads

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

If a patch type antenna is used, a sufficient antenna ground plane is required.

4.9.6 Schematic design

For a **minimal design** with the NEO-M9N GNSS modules, consider the following functions and pins:

- Connect the power supply to VCC and V_BCKP.
- V_USB: If USB is used it is recommended V_USB is to be powered as per USB self-powered mode specification.
- If USB is not used connect V_USB to ground.
- Ensure an optimal ground connection to all ground pins of the NEO-M9N GNSS module.
- Choose the required serial communication interfaces (UART, USB, SPI or I2C) 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.

4.9.7 Layout design-in guideline

- Is the receiver placed away from heat sources?
- Is the receiver placed away from air-cooling sources?
- Is the receiver shielded by a cover/case to prevent the effects of air currents and rapid environmental temperature changes?
- Is the receiver placed as recommended in the Layout and Layout guidance?
- 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 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/co-planar waveguide, around the serial communication lines, underneath the module, etc.

5 Product handling

5.1 ESD handling precautions

⚠️ CAUTION! Risk of electrostatic discharge (ESD) damage. u-blox chips and modules are electrostatic sensitive devices containing highly sensitive electronic circuitry. A discharge of static electricity may damage the device or reduce the life expectancy of the device. To avoid ESD damage, adhere to the standard guidelines for handling ESD devices.

Consider the following:

Preventing electrostatic discharge

- Keep components in their original packages during transport.
- Open the package within an ESD-protected area (EPA), as in Figure 41.
- At a workstation, store components in an EPA.
- Place ESD sensitive devices inside of shielding packaging or containers when transported outside of an EPA.
- Use protective clothing and proper personnel grounding at all necessary points when touching electrostatic sensitive device or assembly. For instance, wear ESD-safe clothing and shoes and wear an ESD wrist strap connected to a grounded workstation. Use heel straps when standing on conductive floors or dissipating floor mats.
- Hold the devices by the edges and avoid touching component contacts, pins, or circuitry

Product handling

- When handling RF transceivers and patch antennas, work in an EPA.
- When connecting test equipment or any other electronics to the module (as a standalone or PCB-mounted device), the first point of contact must always be between the local ground and the PCB ground.
- Before mounting a ceramic patch antenna, connect the device to ground.
- When handling the RF pin, do not touch any charged capacitors. Be especially careful when handling materials like patch antennas (~10 pF), coaxial cables (~50-80 pF/m), soldering irons, or any other materials that can develop charges.
- If there is any risk of touching an exposed antenna area in a 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, use an ESD-safe soldering iron (tip)
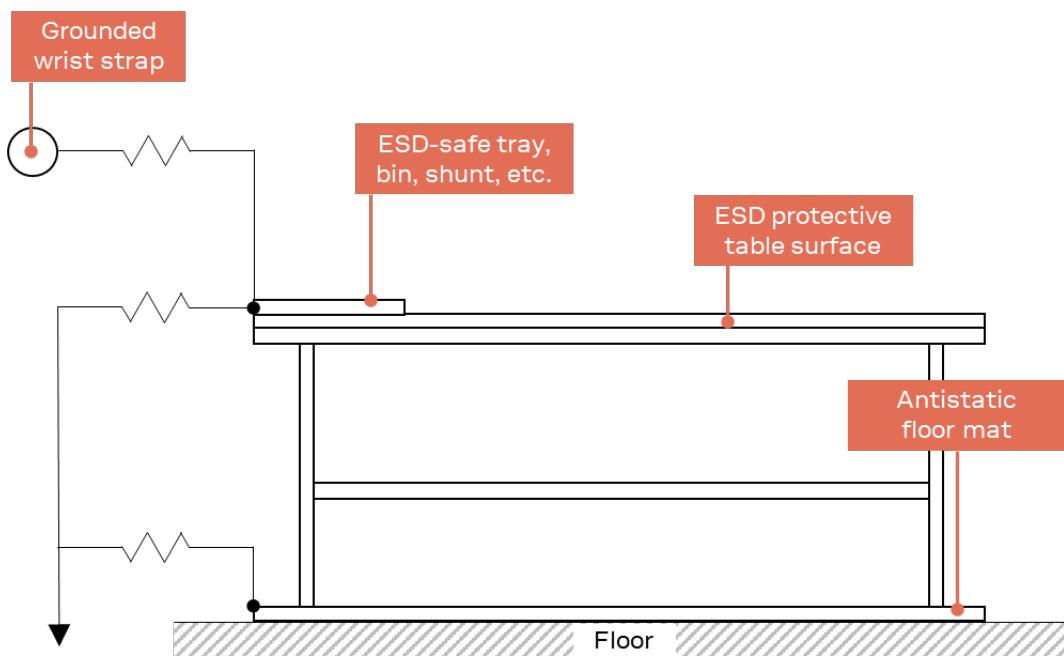
Figure 41: Standard workstation setup for safe handling of ESD-sensitive devices

5.2 Soldering

Soldering paste

Use of "no clean" soldering paste is highly recommended, as it does not require cleaning after the soldering process. The paste 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 customer's specific production processes (e.g. soldering).

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.

As a reference, see "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 the preheat phase does not replace prior baking procedures.

- Temperature rise rate: max 3 °C/s. If the temperature rise is too rapid in the preheat phase, excessive slumping may be caused
- 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 prevents negative metallurgical effects of the solder (solder becomes more brittle) 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 such as the 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.
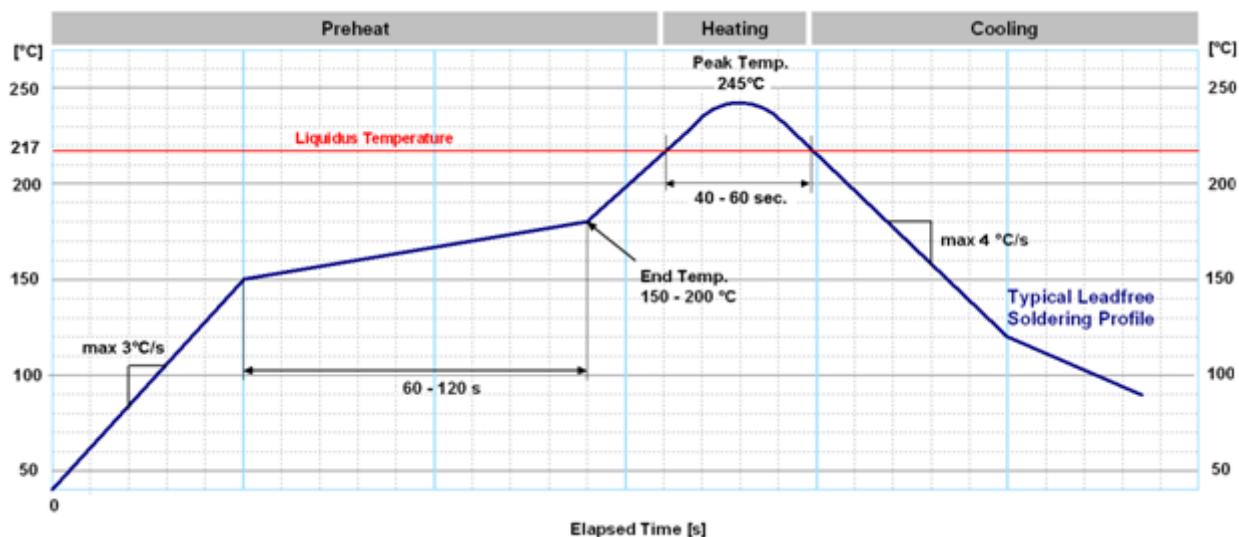
Figure 42: Soldering profile for professional grade NEO-M9N

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

Optical inspection

After soldering the module, consider optical inspection.

Cleaning

☞   Do not clean with water, solvent, or ultrasonic cleaner:

- Cleaning with water will lead to capillary effects where water is absorbed into 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 flowing underneath the module, into areas that are not accessible for post-cleaning inspections. The solvent will also damage the sticker and the 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

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

A board that is populated with components on both sides may require more than one reflow soldering cycle. In such a case, the process should ensure the module is only placed on the board submitted for a single final upright reflow cycle. A module placed on the underside of the board may detach during a reflow soldering cycle due to lack of adhesion.

The module can also tolerate an additional reflow cycle for rework purposes.

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 it is an uncontrolled process and can 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 reapplying solder paste, placing and reflow soldering a new module.

⚠ Never attempt a rework on the module itself, e.g. by replacing individual components. Such actions will immediately void 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 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. Apply the coating carefully.

☞ 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 ultrasonic processes.

5.3 Packaging

The NEO-M9N modules are delivered as hermetically sealed, reeled tapes in order to enable efficient production, production lot set-up and tear-down. For more information about packaging, see the u-blox Package Information Guide [4].

5.4 Tapes

Figure 43 shows the feed direction and illustrates the orientation of the NEO-M9Ns on the tape:
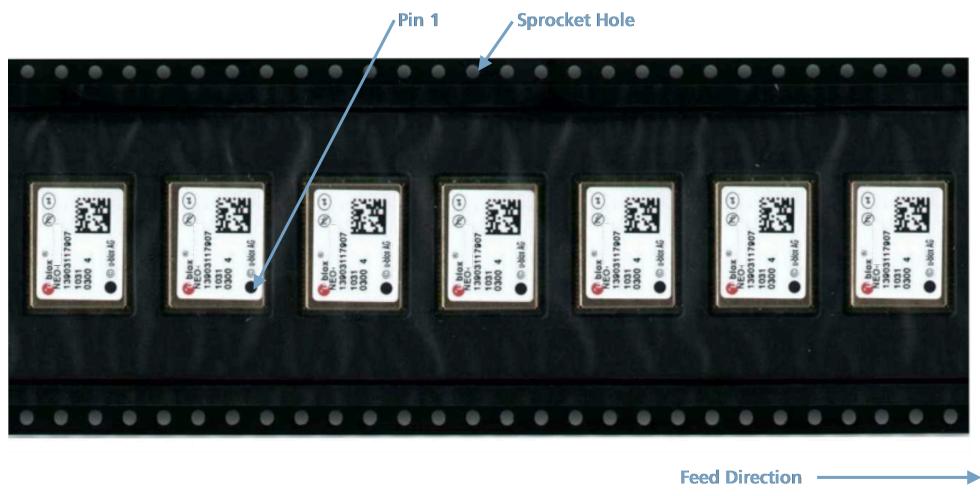
Figure 43: Orientation of NEO-M9N on the tape

The dimensions of the tapes for NEO-M9N are specified in Figure 44 (measurements in mm).
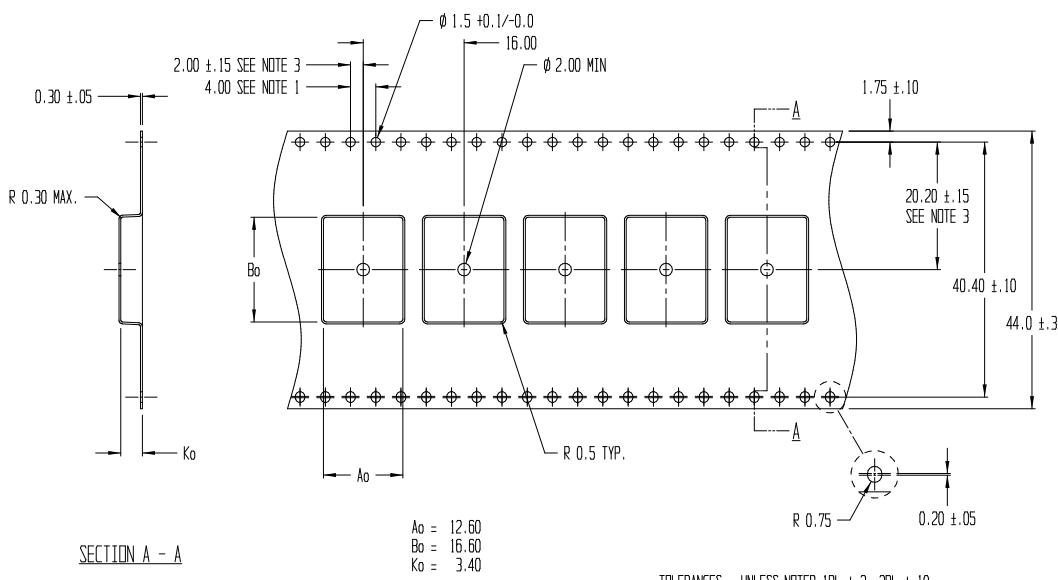
Figure 44: NEO-M9N tape dimensions (mm)

5.5 Reels

The NEO-M9N receivers are deliverable in quantities of pieces on a reel. The receivers are shipped on reel type B, as specified in the u-blox Package Information Guide [4].

5.6 Moisture sensitivity levels

The moisture sensitivity level (MSL) for NEO-M9N is specified in the table below.

Package	MSL level
LCC	4

Table 27: 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 [4].

Appendix

A Migration

u-blox is committed to ensuring that products in the same form factor are backwards compatible over several technology generations. This section describes important differences to consider when migrating from NEO-7, NEO-8Q, and NEO-M8 modules to the NEO-M9N module. A design review with a u-blox support team is recommended to verify proper use of the product features. Table 28 presents the key **hardware-related changes** between NEO-M9N and NEO-7/8Q/M8 modules.

Feature	Change	Action needed / Remarks
VCC supply range	NEO-M9N: 2.7 V - 3.6 V NEO-7M/M8M: 1.65 V - 3.6 V	A HW change is required when migrating from a NEO-7M/M8M design that uses less than 2.7 V VCC supply. Refer to the min and max VCC ramp requirements in the NEO-M9N Data sheet [1].
V_BCKP supply range	NEO-M9N: 1.65 V - 3.6 V NEO-7/8Q/M8: 1.4 V - 3.6 V	A HW change is required when migrating from a NEO-7 or NEO-8Q/M8 design that uses less than 1.65 V V_BCKP supply.
Backup current	NEO-M9N: I_BCKP = 45 µA NEO-7/8Q/M8: I_BCKP = 15 µA	The hardware backup current into V_BCKP in NEO-M9N is higher than in NEO-7 and NEO-8Q/M8 modules. Check that the backup battery capacity is still suitable for the target design/application requirements.
Software backup current	NEO-M9N: I_SWBCKP = 0.36 mA NEO-M8N/Q/J/M: I_SWBCKP = 30 µA NEO-8Q: I_SWBCKP = 22, 30 µA NEO-7N: I_SWBCKP = 35 µA NEO-7M: I_SWBCKP = 20 µA	The software backup current into VCC in NEO-M9N is higher than in NEO-7 and NEO-8Q/M8 modules.
Power consumption	NEO-M9N peak current: 100 mA NEO-7/8Q/M8 peak current: 67 mA	I_VCC is slightly higher in u-blox M9 than in u-blox 7/8/M8 which applies also to acquisition and tracking currents. Ensure that the power supply is still able to provide clean and stable voltage with the higher current demand in NEO-M9N.
Pin-out	1:1 pin-out mapping with NEO-7N/8Q/M8N/Q/J/M modules.	NEO-7M/M8M (pin 14) is reserved and NEO-M9N (pin 14) is LNA_EN. NEO-7M/M8M designs can be updated to use pin 14 on NEO-M9N for external LNA and antenna control. ANT_ON (pin 14) name in NEO-7N changed to LNA_EN in NEO-8Q/M8 and NEO-M9N modules. VDD_USB (pin 7) name changed to V_USB in NEO-M9N to agree with a common naming convention across u-blox modules.
USB interface	Due to the hardware implementation, it may not be possible to certify the USB interface on NEO-M9N.	Consider using other available interfaces (UART, I2C, SPI) if possible.
SAW-LNA RF front-end	NEO-M9N includes a Band 13 notch filter, a SAW filter, and LNA on the RF path, which allows the use of a passive antenna. NEO-M9N noise figure: 3.5 dB NEO-7N/8Q/M8 noise figure: 2 dB	The Band 13 notch filter and the SAW filter used in NEO-M9N provide improved out-of-band immunity compared to NEO-7 and NEO-8Q/M8 modules.
Antenna, External RF path	NEO-M9N supports simultaneous use of GPS, GLONASS, Galileo, BeiDou, and various SBAS constellations.	Check that the RF path (antenna and filtering) is suitable for all of the GNSS signals in use.
Absolute max RF input power	NEO-M9N: +13 dBm NEO-8Q/M8 modules: +15 dBm	Check if the slight reduction in NEO-M9N affects the target design/application requirements.

Feature	Change	Action needed / Remarks
RESET_N	NEO-M9N: clears the RTC time and BBR contents (GNSS orbit data).	After resetting NEO-M9N with the RESET_N pin, the TTFF is similar to performing a cold start.
	NEO-7/8Q/M8: clears only the RTC time.	After resetting the NEO-7/8Q/M8 modules, the TTFF is typically around 6 s in open-sky conditions.
Digital IO	External IO isolation is required.	**Caution!** Risk of receiver damage. If the IO pins are not isolated, the GNSS receiver may get permanently damaged when high current is drawn from the IO pins. Use external IO isolation to avoid receiver damage.
	The output current of the digital IOs is 2 mA compared to 4 mA in NEO-7 and NEO-8Q/M8 modules, except for the TIMEPULSE pin, which still provides 4 mA drive strength.	Do not drive IOs when VCC is not supplied. Removing the VCC supply does not isolate the module IO pins from the the host device IO pins.
		Additional external pull-up resistors may be required to increase the drive strength of the digital IO output pin or adjust the load accordingly.

Table 28: NEO-M9N hardware features compared to NEO-7/8Q/M8 modules

Refer to NEO-7, NEO-8Q, NEO-M8, and NEO-M9N Data sheets for details on performance comparison [1].

Table 29 presents a summary of the key **firmware-related changes** between NEO-M9N and NEO-7/8Q/M8 modules, as well as required actions during migration.

Feature	Change	Action needed / Remarks
Default baud rate	NEO-M9N: 38400 baud.	Code change (optional)
	NEO-7N/7M/8Q/M8N/Q/J/M: 9600 baud.	
Default GNSS config	NEO-M9N: GPS, Galileo, GLONASS, BeiDou, QZSS and SBAS.	Code change (optional)
	NEO-M8N/Q/J/M: GPS, GLONASS, QZSS and SBAS.	
	NEO-7/8Q: GPS, QZSS and SBAS.	
QZSS-IMES	Not supported in the SPG 4.04 firmware.	Code change (optional)
New signal support	QZSS L1S, BDSBAS L1C/A.	Code change (optional)
RF spectrum view	New message: UBX-MON-SPAN shows in-band RF spectrum around the GNSS band. The spectrum information in this message can be used to identify potential in-band RF interference sources in the design.	Code change (optional)
Time pulse	The quantization error output via UBX-TIM-TP2 is no longer supported. u-blox recommends using dedicated timing products for precise time information.	Code change
NMEA	Supports NMEA 0183 V4.10, V4.0, V2.3, and V2.1.	Code change (optional)
	NMEA V4.10 is enabled by default.	
RTCM	RTCM 3.3 is supported. RTCM 2.3 is not supported.	Code change
Power-save modes (PSM)	PSM configuration options have changed. PSMCT: 4 Hz navigation rate is not supported. PSMCT period is configured with CFG-RATE configuration group.	Code change
Navigation update rate	Supports up to 25 Hz for all supported GNSS constellation combinations.	Code change (optional)
Galileo return link messages	Galileo search and rescue (SAR) return link messages are available in UBX-RXM-RLM and NMEA-Standard-RLM.	Code change (optional)
AssistNow	Simultaneous operation of AssistNow Online, Offline and Autonomous. New AssistNow Offline data download options are available. The time period can be limited to a resolution of days to reduce the data package size.	Code change (optional)
Location batching	New feature that can be used to reduce power consumption by saving navigation solutions on the receiver for up to 5 minutes and polling the solutions afterwards to a host MCU.	Code change (optional)

Feature	Change	Action needed / Remarks
Configuration lock	New security feature that is enabled with CFG-SEC-CFG_LOCK message for locking the receiver configuration.	Code change (optional)
Unique chip identifier	The unique chip identifier is available in the UBX-SEC-UNIQID message and at startup from the boot screen.	Code change (optional)
Configuration messages	New configuration scheme using UBX-CFG-VALSET and UBX-CFG-VALGET messages.	Code change
Additional changes for NEO-7 migration	Support for multiple GNSS and new signals: Galileo E1, BeiDou B1I, and GLONASS L1OF.	Code change (optional)
	Multiple GNSS assistance (MGA) with AssistNow services.	
	Geofencing support.	
	UBX message integrity mechanism added.	
	Spoofing detection added.	
	Power save modes support.	
	Wrist dynamic model added.	
	Odometer to measure traveled distance with support for different user profiles.	
	Time pulse and time mark reference changed from GPS to UTC.	

Table 29: NEO-M9N firmware features compared to NEO-7/8Q/M8 modules

For more information on the supported features and messages in the u-blox M9 receiver, refer to the SPG 4.04 Release note and Interface description [3], [2].

Related documents

[1]   NEO-M9N Data sheet, UBX-19014285
[2]   u-blox M9 SPG 4.04 Interface description, UBX-21022436
              u-blox M9 SPG 4.04 Interface description, UBX-21022437 (NDA required)
[3]   u-blox M9 SPG 4.04 Release notes, UBX-20036165
[4]   Packaging information for u-blox chips, modules, and antennas, UBX-14001652

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

Revision history

Revision	Date	Status / comments
R01	28-Nov-2019	Advance information
R02	13-Mar-2020	Advance information Firmware version updated. Isolation example updated in section 3.5. SBAS summary added to section 3.13.1. Reset after shutdown added to section 3.14. Pin assignment table corrected in section 4.1. Tape dimensions diagram updated in section 5.5.
R03	31-Mar-2020	Advance information Added description of the Course over ground low-pass filter to section 3.1.6. Updated description of recording in section 3.3.4.
R04	25-May-2020	Advance information Ordering and numbering of sections updated. Added section 3.1.5 RTCM corrections. Removed old section 4.3.4 NEO-M9N power supply. Updated description of 50 Ω matching in section 4.5 Antenna.
R05	11-Sep-2020	Early production information Updated firmware version to 4.04. Changed topic title to "Weak signal compensation" in section 3.1.7.6. Clarified I2C addressing in section 3.5.2. Changed AssistNow Autonomous default setting in section 3.7.6.1.
R06	08-Jul-2021	Document status removed from Document information. Refer to NEO-M9N Data sheet [1] for Product status. Updated section Receiver configuration, added sections SBAS, Spectrum analyzer, Out-of-band blocking immunity, and Thermal management. Minor editing throughout the document.
R07	13-Dec-2021	Updated section 3.6 Spectrum analyzer, section 3.9 Multiple GNSS assistance, and NEO-8/M8 to NEO-M9N Migration section.
R08	14-Dec-2022	Removed section Mechanical dimensions. Refer to the product data sheet for the mechanical specification.
R09	09-Feb-2024	Added NEO-7 module to the Migration section.

Contact

u-blox AG

Address: Zürcherstrasse 68
                  8800 Thalwil
                  Switzerland

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