

FusionEngine Message Specification

Version 0.20

Contents

Release History.....	6
Version 0.20 (2023-11-10)	6
Version 0.19 (2023-10-05)	6
Version 0.18 (2023-8-24)	6
Version 0.17 (2023-6-9)	6
Version 0.16 (2023-5-4)	6
Version 0.15 (2023-4-7)	6
Version 0.14 (2023-3-3)	7
Version 0.13 (2023-2-17)	7
Version 0.12 (2022-12-6)	7
Version 0.11 (2022-11-7)	7
Version 0.10 (2022-11-4)	7
Version 0.9 (2022-10-10)	7
Version 0.8 (2022-7-29)	8
Version 0.7 (2022-7-22)	8
Version 0.6 (2022-7-15)	8
Version 0.5 (2022-7-11)	8
Version 0.4 (2022-6-22)	8
Version 0.3 (2022-5-31)	8
Version 0.2 (2022-5-9).....	8
Version 0.1 (2022-2-18)	9
1. Overview	10
2. Message Structure.....	10
2.1 Data Types	11
2.2 Byte Ordering.....	11
2.3 Message Header	12
2.4 CRC Calculation Details	13
Generation of the <i>CRC Table</i>	13
Calculating the CRC on a <i>buffer</i> of known <i>length</i>	13



2.5 Timing	13
2.6 Data Structures/Enumeration	14
2.6.1 Timestamp	14
2.6.2 SystemTimeSource.....	14
2.6.3 SensorDataSource	15
2.6.4 MeasurementDetails	16
2.6.5 GNSS System Constants (SatelliteType)	17
2.6.6 GNSS Frequency Band Constants (FrequencyBand).....	18
2.6.7 Transmission Gear Type (GearType).....	19
2.7 NMEA-0183 Message Types	19
3. Messages	21
3.1 Command/Control Messages (Input)	21
3.1.1 MessageRequest (13001)	21
3.1.2 ResetRequest (13002)	22
3.1.3 ShutdownRequest (13005)	28
3.1.4 FaultControl (13006).....	29
3.1.5 SetConfig (13100)	32
3.1.6 GetConfig (13101)	46
3.1.7 SaveConfig (13102)	48
3.1.8 SetMessageRate (13220).....	49
3.1.9 GetMessageRate (13221)	54
3.1.10 GetDeviceID (13007)	55
3.2 Command Response Messages (Output)	56
3.2.1 CommandResponse (13000)	56
3.2.2 VersionInformation (13003)	57
3.2.4 EventNotification (13004).....	58
3.2.5 ConfigResponse (13103).....	59
3.2.6 MessageRateResponse (13222)	60
3.2.7 SupportedIOInterfaces (13223)	62
3.3 Navigation Solution Messages (Output)	63



3.3.1 Pose (10000)	63
3.3.2 GNSSInfo (10001)	66
3.3.3 GNSSSatellite (10002)	67
3.3.4 PoseAux (10003)	69
3.3.5 CalibrationStatus (10004)	70
3.3.6 RelativeENUPosition (10005)	72
3.3.7 ROSPose (12000)	73
3.3.8 ROSGPSFix (12010)	73
3.4 Sensor Measurement Output Messages	76
3.4.1 IMUOutput (11000)	76
3.4.2 HeadingOutput (11003)	77
3.4.3 ROSIMUMessage (12011)	79
3.5 Sensor Measurement Input Messages	81
3.5.1 WheelTickInput (11103)	82
3.5.4 VehicleTickInput(11104)	83
3.5.2 WheelSpeedInput (11105)	84
3.5.3 VehicleSpeedInput (11106)	85
3.6 Raw Sensor Measurement Output Messages	86
3.6.1 RawHeadingOutput (11001)	86
3.6.2 RawIMUOutput (11002)	87
3.6.3 RawWheelTickOutput (11123)	89
3.6.4 RawVehicleTickOutput(11124)	90
3.6.5 RawWheelSpeedOutput (11125)	91
3.6.6 RawVehicleSpeedOutput (11126)	92
3.6.7 WheelSpeedOutput (11135)	93
3.6.8 VehicleSpeedOutput (11136)	94
3.7 Device Status Messages	95
3.7.1 System Status (10500)	95
4 Proprietary NMEA Sentence Definitions	96
4.1 NMEA \$P1CALSTATUS	96



4.2 NMEA \$P1MSG 96

4.3 NMEA \$PQTMGNSS 97

4.4 NMEA \$PQTMVER..... 97

4.5 NMEA \$ PQTMVERNO..... 97



Release History

Version 0.20 (2023-11-10)

Compatibility with FusionEngine Protocol Release v1.22.3

- Updates for wheel encoder messages
- Fixed data types for wheel encoder and velocity inputs
- Added messages for filtered velocity and wheel encoder outputs

Version 0.19 (2023-10-05)

Compatibility with FusionEngine Protocol Release v1.22.0

- Added new SupportedIOInterfaces message
- Corrected VersionInformation documentation
- Updated WheelSensorType enum values
- Correction protocol version number

Version 0.18 (2023-8-24)

Compatibility with FusionEngine Protocol Release v1.21.0

- Updated Fault Control types to include factory testing options

Version 0.17 (2023-6-9)

Compatibility with FusionEngine Protocol Release v1.18.1

- Updated ROS IMU message documentation
- Adds Heading Bias configuration
- Changed HeadingMeasurement to RawHeadingOutput (no change to struct)
- Added HeadingOutput message
- Adds GetDeviceID message

Version 0.16 (2023-5-4)

Compatibility with FusionEngine Protocol Release v1.18.1

- Added configuration items for the troposphere and ionosphere.

Version 0.15 (2023-4-7)

Compatibility with FusionEngine Protocol Release v1.18.0

- Added leap second and GPS week rollover override configuration options
- Added UART1 Baud Rate configuration option
- Revised fields in the GNSSInfo Message to include UTC leap seconds, number of SVs, GNSS Correction age and GNSS baseline distance.
- Added MeasurementDetails definition
- Added Sensor Data Source definition for measurement inputs and outputs.
- Removed MeasurementTimestamp definition



- Removed deprecated Wheel speed and Vehicle Speed messages
- Added RawIMUOutput, RawWheelTickOutput, RawVehicleTickOutput, RawWheelSpeedOutput, RawVehicleSpeedOutput
- Added RawWheelTickInput, RawVehicleTickInput, RawWheelSpeedInput, RawVehicleSpeedInput
- Added SystemStatus message
- Added new Diagnostic Reset functionality and modified HOT, WARM and COLD start mask definitions.
- Added Revert To Default option for configuration settings.
- Clarified definition of SaveAction in SetConfig message.

Version 0.14 (2023-3-3)

Compatibility with FusionEngine Protocol Release v1.17.0

- Added GNSS system and frequency band enable/disable settings definitions
- Clarified use of WheelConfig (software data/CAN) vs HardwareTickConfig (voltage signal)

Version 0.13 (2023-2-17)

Compatibility with FusionEngine Protocol Release v1.15.0

- Added HeadingMeasurement message definition

Version 0.12 (2022-12-6)

Compatibility with FusionEngine Protocol Release v1.13.0

- Clarified payload descriptions for SetConfig, ConfigResponse, and FaultControl messages
- Adds ROS compatible messages for ROSPose, ROSGPSFix and ROSImuMessage

Version 0.11 (2022-11-7)

Compatibility with FusionEngine Protocol Release v1.13.0

- Corrected missing reserved bytes in FaultControl definition
- Added EventNotification message definition

Version 0.10 (2022-11-4)

Compatibility with FusionEngine Protocol Release v1.13.0

- Added ALL and CURRENT TransportType enumerations
- Corrected UNSUPPORTED_VERSION response code documentation
- Added FaultControl message definition

Version 0.9 (2022-10-10)

Compatibility with FusionEngine Protocol Release v1.12.0



- Added apply-and-save flag to SetMessageRate command
- Added wildcard capabilities to SetMessageRate and GetMessageRate commands
- Removed SetOutputInterfaceConfig and GetOutputInterfaceConfig commands

Version 0.8 (2022-7-29)

Compatibility with FusionEngine Protocol Release v1.11.2

- Added watchdog enable/disable setting definition
- Added message type enumeration for NMEA-0183

Version 0.7 (2022-7-22)

Compatibility with FusionEngine Protocol Release v1.11.0

- Added Vehicle Tick Measurement definition
- Added control messages for enabling/disabling/controlling rate of individual output message types
- Changed inbound measurement timestamp definitions (MeasurementTimestamps)

Version 0.6 (2022-7-15)

Compatibility with FusionEngine Protocol Release v1.11.0

- Added WheelSpeedMeasurement, WheelTickMeasurement, and VehicleSpeedMeasurement message definitions
- Fixed missing reserved bytes in GetOutputInterfaceConfig

Version 0.5 (2022-7-11)

Compatibility with FusionEngine Protocol Release v1.10.0

- Corrected descriptions for WheelConfig, HardwareTickConfig,
- Fixed reserved bytes in SetConfig and OutputInterfaceConfigurationEntry

Version 0.4 (2022-6-22)

Compatibility with FusionEngine Protocol Release v1.10.0

- Added HardwareTickConfig configuration payload
- Compatibility with FusionEngine Release v1.10.0

Version 0.3 (2022-5-31)

Compatibility with FusionEngine Protocol Release v1.10.0

- Added RelativeENUPosition and ShutdownRequest messages
- Added reboot support to ResetRequest message

Version 0.2 (2022-5-9)

Compatibility with FusionEngine Protocol Release v1.10.0

- Added ResetRequest, SetConfig, GetConfig, and ConfigResponse message definitions.



- Added solution and measurement message definitions (Pose, PoseAux, IMUMeasurement, etc.).

Version 0.1 (2022-2-18)

Compatibility with FusionEngine Protocol Release v1.9.0

- Initial version.



1. Overview

This document describes the messages used by Point One's FusionEngine software.

Point One FusionEngine messages are used for both input and output. The messages are a highly compact binary format that contain detailed information about the PVT solution, attitude solution, and other important receiver and navigation statuses. The messages are also used for configuration of the target device, including enabling and disabling other protocols, streams, etc.

Open-source examples of this protocol are available in the repository linked below. While this manual contains the full descriptions for each message, it is highly recommended to use the existing library whenever possible as it has been thoroughly tested across Point One's products.

The Point One FusionEngine Protocol is available as
an open-source library for C++ and Python:

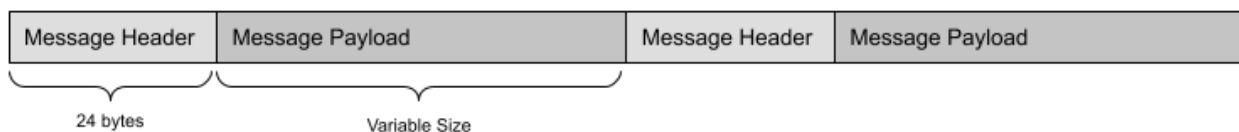
<https://github.com/PointOneNav/fusion-engine-client>

Source code documentation available at:

<http://docs.pointonenav.com/fusion-engine>

2. Message Structure

All FusionEngine messages consist of a Message Header and Message Payload. Messages are aligned to 4-byte boundaries, so the size of a message will *always* be a multiple of 4 bytes. This means that messages will contain padding bytes if necessary.



2.1 Data Types

Data Type	Size in bytes	Description
bool	1	Boolean Value (0 or 1)
u8	1	Unsigned Integer
u16	2	Unsigned Integer
u32	4	Unsigned Integer
u8[N]	N	Reserved/Padding Bytes
i8	1	Signed Integer
i16	2	Signed Integer
i32	4	Signed Integer
i64	8	Signed Integer
f32	4	IEEE Single-Precision Floating Point
f64	8	IEEE Double-Precision Floating Point
str	<i>Variable</i>	Character String (Not Null-Terminated)

2.2 Byte Ordering

All FusionEngine messages are constructed using little endian (Intel) encoding.



2.3 Message Header

The message header is 24 bytes in length and contains synchronization bytes and information required to decode or encode the Message Payload.

Field	Data Type	Description
Sync Byte 0	u8	Always 0x2E (ASCII ".")
Sync Byte 1	u8	Always 0x31 (ASCII "1")
Reserved	u8[2]	<i>Reserved for future use.</i>
CRC	u32	The 32-bit CRC of all bytes from and including the protocol version field to the last byte in the message, including the message payload. This uses the standard CRC-32 generator polynomial in reversed order (0xEDB88320).
Protocol Version	u8	The version of the FusionEngine Protocol.
Message Version	u8	The version of this message. Messages of the same type will always be backwards compatible by using reserved fields in previous versions or extending the message length.
Message Type	u16	Uniquely identifies each message type. The combination of Version and Type should be used to know how to decode the payload.
Sequence Number	u32	A sequence number that is incremented with each message.
Payload Size	u32	Size of the payload to follow in bytes.
Source Identifier	u32	Identifies the source of the message when applicable. This definition can change depending on the message type.



2.4 CRC Calculation Details

Below is an abbreviated example of the CRC algorithm. Note that some hardware may have dedicated accelerators for this algorithm but that is out of scope of this document. See `crc.cc` in the open-source library for a complete reference.

Generation of the *CRC Table*

```
// This table can be pre-computed once and reused for all messages.
// It does not depend on data content.
const uint32_t polynomial = 0xEDB88320;
for (uint32_t i = 0; i < 256; i++) {
    uint32_t c = i;
    for (size_t j = 0; j < 8; j++) {
        if (c & 1) {
            c = polynomial ^ (c >> 1);
        } else {
            c >>= 1;
        }
    }
    crc_table[i] = c;
}
```

Calculating the CRC on a *buffer of known length*

```
uint32_t initial_value = 0;
uint32_t c = initial_value ^ 0xFFFFFFFF;
for (size_t i = 0; i < length; ++i) {
    c = crc_table[(c ^ buffer[i]) & 0xFF] ^ (c >> 8);
}
return c ^ 0xFFFFFFFF;
```

2.5 Timing

All Point One devices generate a common time-based called Point One time (P1 time). P1 time is used to coordinate all measurement and solution data within the system.

P1 time may be driven by a microcontroller (MCU), or a local monotonic clock provided by the operating system. The time itself is synthetic and may be adjusted dynamically based on internal clock error models as the system operates. Users should not assume that P1 time is equal to the source clock (MCU or OS), or to GPS or POSIX times.

In addition to P1 time, some messages may optionally include additional timestamps where available, including GPS system time, POSIX time, platform operating system time. See Section 2.6.2 for more details.



2.6 Data Structures/Enumeration

2.6.1 Timestamp

This structure may be used to convey Point One time values (referenced to the start of the device), GPS times (referenced to January 6, 1980), POSIX times (referenced to January 1, 1970), or embedded device system timestamps (platform-specific). The type of time will be specified by the SysytemTimeSource (2.6.2).

Timestamp (8B):

Field	Data Type	Description
Seconds	u32	The number of integer seconds since the reference epoch. Set to 0xFFFFFFFF if invalid.
Fraction	u32	The fractional part of the timestamp in nanoseconds. Set to 0xFFFFFFFF if invalid.

2.6.2 SystemTimeSource

Represents the time base used for a message timestamp. When a measurement is supplied externally, it may either contain a timestamp generated by the sender or it can be timestamped by FusionEngine when the packet arrives.

SystemTimeSource (u8):

Type	Value	Description
Invalid	0	Timestamp not valid. Message will be timestamped on arrival.
P1 Time	1	Measurement timestamped in P1 time.
Timestamped on Reception	2	Message timestamped in device system time at the time of arrival.
Sender System Time	3	Timestamp was generated from a monotonic clock of an external system.
GPS Time	4	Message timestamped in GPS time, referenced to 1980/1/6.
Reserved	5-255	<i>Reserved for future use.</i>



2.6.3 SensorDataSource

Sensor data may come from a variety of different sources. This struct identifies the sensor data source and can be useful for verification purposes when viewing output messages.

SensorDataSource (u8):

Type	Value	Description
Unknown	0	The data source is unknown.
Internal	1	Sensor data captured internal to the device (embedded IMU, internal GNSS receiver, etc.)
Hardware IO	2	Sensor data generated via hardware voltage signal measured directly by the device (hardware wheel tick, external event, etc.).
CAN	3	Sensor data captured from a platform CAN bus.
Serial	4	Sensor Data input over a serial (UART) connection
Network	5	Sensor Data input over a network (Ethernet/Wi-Fi)
Reserved	6-255	<i>Reserved for future use.</i>



2.6.4 MeasurementDetails

All messages containing input measurements that are designed to be sent to begin with a `MeasurementDetails` structure. This structure describes the time of applicability of the sensor data.

Where possible, the user may provide known P1, GPS, or system timestamps indicating the time at which the sensor data applies. If a timestamp cannot be provided, the incoming sensor data will be timestamped by FusionEngine on arrival.

Note that measurements timestamped on arrival are subject to unknown latency due to the capture and transmission of the sensor data, including latency introduced by the data interface. Excessive latency may result in performance degradation.

Once the message is received, the specified measurement time will be converted to P1 time automatically by FusionEngine using its internal clock models. The user should not populate the P1 time field when sending measurement data to a device. Any value in the P1 time field will be ignored.

`MeasurementDetails` (20B):

Field	Data Type	Description
Measurement Time	<code>Timestamp</code> (8B)	The time of applicability of the measurement data. Interpretation dictated by Time Source.
Time Source	<code>SystemTimeSource</code> (u8)	The source of the measurement timestamp. See Section 2.6.2 .
Sensor Data Source	<code>SensorDataSource</code> (u8)	The source of the sensor. See Section 2.6.3
Reserved	<code>u8[2]</code>	<i>Reserved for future use.</i>
Output P1 Time	<code>Timestamp</code> (8B)	Leave invalid when sending measurements to the device. On output, this will be the calculated P1 time corresponding with the measurement timestamp.



2.6.5 GNSS System Constants (SatelliteType)

The following enumeration is used to specify GNSS systems (constellations).

Name	Value	Bit Mask	Description
UNKNOWN	0	0x000	GNSS constellation not known.
GPS	1	0x002	
GLONASS	2	0x004	
LEO	3	0x008	<i>Not currently used.</i>
GALILEO	4	0x010	
BEIDOU	5	0x020	
QZSS	6	0x040	
MIXED	7	0x080	<i>Not currently used.</i>
SBAS	8	0x100	
IRNSS	9	0x200	Also known as NavIC.
Reserved	10-255		<i>Reserved for future use.</i>

When sending a [3.1.5 SetConfig \(13100\)](#) message to enable/disable GNSS constellations, use the values in the Bit Mask column. For example, the mask 0x12 enables GPS and Galileo.



2.6.6 GNSS Frequency Band Constants (FrequencyBand)

The following enumeration is used to specify GNSS systems frequency bands. Each frequency band spans a range of GNSS as shown below.

Name	Value	Bit Mask	Description
UNKNOWN	0		Frequency band not known.
L1	1	0x02	1561.098 MHz (B1) -> 1602.0 (G1) Includes: <ul style="list-style-type: none">• GPS/QZSS L1 (1575.42 MHz)• Galileo E1 (1575.42 MHz)• BeiDou B1I (1561.098 MHz)• BeiDou B1C (1575.42 MHz)• GLONASS G1 (1602.0 MHz)
L2	2	0x04	1202.025 MHz (G3) -> 1248.06 (G2) Includes: <ul style="list-style-type: none">• GPS L2 (1227.6 MHz)• Galileo E5b (1207.14 MHz)• BeiDou B2I (1207.14 MHz)• GLONASS G2 (1246.0 MHz)• GLONASS G3 (1202.25 MHz)
Reserved	3-4		<i>Reserved for future use.</i>
L5	5	0x20	1176.45 MHz (L5/E5a/B2a) Includes: <ul style="list-style-type: none">• GPS/QZSS L5 (1176.45 MHz)• Galileo E5a (1176.45 MHz)• BeiDou B2a (1176.45 MHz)• IRNSS L5 (1176.45 MHz)
L6	6	0x40	1262.52 MHz (B3) -> 1278.75 (L6/E6) Includes: <ul style="list-style-type: none">• Galileo E6 (1278.75 MHz)• BeiDou B3 (1268.52 MHz)• QZSS L6 (1278.75 MHz)
Reserved	7-255		<i>Reserved for future use.</i>

When sending a [3.1.5 SetConfig \(13100\)](#) message to enable/disable GNSS frequencies, use the values in the Bit Mask column. For example, the mask 0x22 enables the L1 and L5 bands.



2.6.7 Transmission Gear Type (GearType)

The following enumeration defines the “gear” in a vehicle transmission. The field is always a u8.

Stage	Value	Description
Unknown	0	Transmission gear not known, or gear does not map to one of the values defined below.
Forward	1	Vehicle in a forward gear.
Reverse	2	Vehicle in a reverse gear.
Park	3	Vehicle is parked.
Neutral	4	Vehicle is in neutral.
Reserved	5-255	<i>Reserved for future use.</i>

2.7 NMEA-0183 Message Types

The following table lists the 16-bit (u16) message IDs assigned to each supported NMEA-0183 message, used to configure output interface settings. All other values are reserved for future use.

NMEA Message Type	Message ID
Invalid	0
<i>Standard NMEA-0183 Messages</i>	
GGA	1
GLL	2
GSA	3
GSV	4
RMC	5
VTG	6
<i>Point One Proprietary Messages</i>	
P1CALSTATUS	1000
P1MSG	1001



<i>Quectel Proprietary Messages</i>	
PQTMVERNO	1200
PQTMVER	1201
PQTMGNSS	1202



3. Messages

3.1 Command/Control Messages (Input)

These are messages sent from the host device to the FusionEngine software. They are primarily used for configuration, external sensor input, and querying saved parameters.

3.1.1 MessageRequest (13001)

Message ID: 13001

Protocol Version: 2

Message Version: 0

Description: Request transmission of a specific message type. Unless otherwise stated, the response will be generated on the same interface.

Field	Data Type	Description
Message Type	u16	The type of the message requested.
Reserved	u8[2]	<i>Reserved for future use.</i>

Supported Message Types to be requested:

Message Type Request	Message Type ID
Version Info	13003

Expected Response(s):

Responding Message Type	Responding Message ID
<Requested Message Type>	<Requested Message Type ID>

Note: If the specified message type cannot be requested, a 3.2.1 COMMANDRESPONSE (13000) will be returned with a *Value Error*.

3.1.2 ResetRequest (13002)

Message ID: 13002

Protocol Version: 2

Message Version: 0

Description: Requests the system perform a specified group of reset operations.

Field	Data Type	Description
Reset Mask	u32	Bit mask indicating the components to be reset.

Reset Mask Bits:

Bit Number	Name	Reset Triggered When Set
0	RESTART_NAVIGATION_ENGINE	Restart the navigation engine, but do not clear its position estimate.
1	RESET_GNSS_CORRECTIONS	Delete all GNSS corrections information.
2-7	Reserved	Reserved for future use.
8	RESET_POSITION_DATA	Reset the navigation engine's estimate of position, velocity, and orientation.
9	RESET_EPHEMERIS	Delete all saved satellite ephemeris.
10	RESET_FAST_IMU_CORRECTIONS	Reset bias estimates, and other IMU corrections that are typically estimated quickly.
11	Reserved	Reserved for future use.
12	RESET_NAVIGATION_ENGINE_DATA	Reset all stored navigation engine data, including position, velocity, and orientation state, as well all IMU corrections (fast and slow) and other training data.



13	RESET_CALIBRATION_DATA	<p>Reset the device calibration data.</p> <p>Note:</p> <p>This does not reset any existing navigation engine state. It is recommended that you set RESET_NAVIGATION_ENGINE_DATA as well under normal circumstances.</p>
14-19	Reserved	<i>Reserved for future use.</i>
20	RESET_CONFIG	Reset all configuration settings back to factory defaults.
21-23	Reserved	<i>Reserved for future use.</i>
24	RESET_GNSS_MEASUREMENT_ENGINE	Restart the GNSS measurement engine, performing a cold start to clear all position and satellite data.
25	REBOOT_NAVIGATION_PROCESSOR	Reboot the processor on which the navigation software is executed.
26	DIAGNOSTIC_LOG_RESET	Perform a diagnostic log reset to guarantee deterministic performance for data post-processing and diagnostic support.
26-31	Reserved	<i>Reserved for future use.</i>



Special Mask Values:

Value	Name	Description
0x00000001	HOT_START	<p>Perform a device hot start.</p> <p>A hot start is typically used to restart the navigation engine in a deterministic state, using previously stored position and time information. The device will begin navigating immediately if possible.</p> <p>To be reset:</p> <ul style="list-style-type: none">• The navigation engine <p>Not reset:</p> <ul style="list-style-type: none">• All runtime data (GNSS corrections, etc.)• Position, velocity, orientation state• Fast IMU corrections• Training parameters (slowly estimated IMU corrections, temperature, compensation, etc.)• Calibration data• User configuration settings• GNSS measurement engine

0x00000201	WARM_START	<p>Perform a device warm start.</p> <p>During a warm start, the device retains its knowledge of approximate position and time, plus almanac data if available, but resets all ephemeris data. As a result, the device will need to download ephemeris data before continuing to navigate.</p> <p>To be reset:</p> <ul style="list-style-type: none"> • The navigation engine • GNSS Ephemeris Data <p>Not reset:</p> <ul style="list-style-type: none"> • All runtime data (GNSS corrections, etc.) • Position, velocity, orientation state • Fast IMU corrections • Training parameters (slowly estimated IMU corrections, temperature, compensation, etc.) • Calibration data • User configuration settings • GNSS measurement engine
------------	------------	--



0x00000FFF	COLD_START	<p>Perform a device cold start.</p> <p>A cold start is typically used to reset the device's state estimate in the case of an error that cannot be resolved by a warm start.</p> <p>To be reset:</p> <ul style="list-style-type: none"> • The navigation engine • All runtime data (GNSS corrections, etc.) • Position, velocity, orientation state • Fast IMU corrections • GNSS ephemeris data <p>Not reset:</p> <ul style="list-style-type: none"> • Training parameters (slowly estimated IMU corrections, temperature, compensation, etc.) • Calibration data • User configuration settings • GNSS measurement engine
0x000001FF	PVT_RESET	Resets the kinematic portion of the navigation engine state. Useful for error recovery.
0xFFFFFFFF	FACTORY_RESET	Restart mask to set all persistent data, including position/orientation, calibration state, and user configuration, back to factory defaults.

Expected Response(s):

3.2.1 COMMANDRESPONSE (13000)

Example Usage

Example 1: Perform a device cold start

This will reset the navigation engine including saved position, velocity, and orientation state, as well as the GNSS measurement engine, but do not reset training data, calibration data, or user configuration parameters, by setting the reset mask to 0x01000FFF.

```
2e31 0000 0acf ee8f 0200 ca32 0000 0000
```



```
0400 0000 0000 0000 ff0f 0001
```

Example 2: Perform a factory reset

Reset all settings, state, and calibration data to factory defaults by setting the reset mask to 0xFFFFFFFF.

```
2e31 0000 b39e 0bf3 0200 ca32 0000 0000  
0400 0000 0000 0000 ffff ffff
```



3.1.3 ShutdownRequest (13005)

Message ID: 13005

Protocol Version: 2

Message Version: 0

Description: Shutdown the device.

Field	Data Type	Description
Flags	u64	A bitmask of flags used to control the shutdown process. Set to 0 always.
Reserved	u8[8]	<i>Reserved for future use.</i>

Save action values:

Save Action	Value	Description
Save	0	Save all active parameters to persistent storage.
Revert to Saved	1	Revert the active configuration to previously saved values.
Revert to Default	2	Reset the active and saved configuration to default values.

Expected Response(s):

3.2.1 COMMANDRESPONSE (13000)

Example Usage

Example 1: Shutdown the device

2e31 0000 716f 5414 0200 cd32 0000 0000
1000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000



3.1.4 FaultControl (13006)

Message ID: 13006

Protocol Version: 2

Message Version: 0

Description: Testing message to trigger or clear a specified system fault.

This message is intended for factory testing and development purposes only and should not be used in any production environment.

Field	Data Type	Description
Fault Type	FaultType (u8)	The type of fault/control to be performed.
Reserved	u8[15]	<i>Reserved for future use.</i>
Value Length	u32	The length of the value to follow (in bytes).
Value	<i>Variable</i>	The parameter value for the specified fault type. The format of the value is listed in the FaultType table below.

FaultType (u8):

Name	Type	Value Format	Description
Clear All	0	None	Clear existing faults.
Crash	1	None	Force the device to crash. In a crash state, the device no longer produces any output on any interfaces, and will stop responding to commands. If the watchdog is enabled, the device will restart automatically after the watchdog timer elapses.
Fatal Error	2	None	Force the device to exhibit a fatal error. After a fatal error, the device will stop navigating and will no longer produce solution messages on any interfaces. Instead, it will output an EventNotificationMessage indicating the fault status. If the watchdog is enabled, the device will restart automatically after the watchdog timer elapses.



COCOM	3	COCOM Type (3.1.4.1) (u8)	Trigger a COCOM limit. When a COCOM limit is exceeded, the device will stop navigating and will produce Invalid solution messages. COCOM limits may be cleared via ResetRequest , or by sending a COCOM FaultControl message with the COCOMType parameter set to NONE.
Enable GNSS	4	bool	Enable/disable use of GNSS measurements (intended for dead reckoning performance testing).
Region Blackout	5	bool	Enable/disable a simulated region blackout (intended for factory tests)
Line Test Mode	6	bool	Enable/disable factory test behavior for use with GNSS simulation
Reserved	7-255		Reserved for future use.

Expected Response(s):

3.2.1 COMMANDRESPONSE (13000)

Note that the device will crash immediately after receiving a Crash Fault Type request. It will not send a CommandResponse back to the user.

Examples

Example 1: Cause the device to immediately crash.

```
2e31 0000 bde4 777b 0200 ce32 0000 0000
1400 0000 0000 0000 0100 0000 0000 0000
0000 0000 0000 0000 0000 0000
```

Example 2: Trigger an artificial COCOM altitude limit.

```
2e31 0000 0bf7 0344 0200 ce32 0000 0000
1500 0000 0000 0000 0300 0000 0000 0000
0000 0000 0000 0000 0100 0000 03
```

Note that in practice, the message sequence number should be modified, consistent with any commands sent previously, and the CRC should be updated to reflect the change.



3.1.4.1 COCOM Type (u8)

Name	Value	Description
NONE	0	Clear the current COCOM limit.
ACCELERATION	1	Trigger a maximum acceleration limit.
SPEED	2	Trigger a maximum speed limit.
ALTITUDE	3	Trigger a maximum altitude limit.
Reserved	4-254	<i>Reserved for future use.</i>

3.1.5 SetConfig (13100)

Message ID: 13100

Protocol Version: 2

Message Version: 0

Description: Set the value of a user configuration parameter.

Field	Data Type	Description
Parameter Type	ConfigType (u16)	The parameter to be configured. The types are listed in the ConfigType table below.
Save Action	SaveAction (u8)	Options on how to save the configuration.
Reserved	u8	<i>Reserved for future use.</i>
Value Length	u32	The length of the value to follow (in bytes).
Value	Variable	The new value to use for the specified parameter. The format of the value is listed in the ConfigType table below.

SaveAction (u8):

Name	Value	Description
SAVE	0	Setting this bit saves the updated configuration to persistent storage immediately on success.
REVERT_TO_SAVED	1	Revert the active configuration to previously saved values.
REVERT_TO_DEFAULT	2	Reset the active and saved configuration to default values.

ConfigType (u16):

Name	Parameter Type	Value Format	Description
Invalid	0	None	Invalid parameter.
Reserved	1-15		<i>Reserved for future use.</i>



Device Lever Arm	16	Point3f (3.1.5.1)	The location of the device IMU with respect to the vehicle body frame in meters.
Device Orientation	17	CoarseOrientation (3.1.5.2)	The orientation of the device IMU with respect to the vehicle body axes.
GNSS Lever Arm	18	Point3f (3.1.5.1)	The location of the GNSS antenna with respect to the vehicle body frame in meters.
Output Lever Arm	19	Point3f (3.1.5.1)	The offset of the desired output location with respect to the vehicle body frame in meters.
Vehicle Details	20	VehicleDetails (3.1.5.3)	Information about the vehicle including model and dimensions.
Software Wheel Config	21	WheelConfig (3.1.5.4) (Software)	Configuration for software wheel speed/rotation (tick) measurements (FusionEngine messages/CAN data).
Hardware Tick Config	22	HardwareTickConfig (3.1.5.5)	Configuration capturing hardware wheel tick measurements from an incoming voltage signal.
Heading Bias	23	HeadingBias (3.1.5.9)	Used to set horizontal (yaw) & vertical (pitch) biases (in degrees) on a dual-antenna heading platform configuration.
Enabled GNSS Systems	50	u32	A bitmask indicating which GNSS systems are enabled (see 2.6.5 GNSS Systems (SatelliteType)).
Enabled GNSS Frequency Bands	51	u32	A bitmask indicating which GNSS frequency bands are enabled (see 2.6.6 GNSS Frequency Bands (FrequencyBand)).
Leap Second Override	52	i32	Specify a UTC leap second count override value to use for all UTC time conversions. Note: Setting this value will disable all internal leap second sources and should only be used for testing . Set to -1 to disable.



GPS Week Rollover Override	53	i32	Specify a GPS legacy week rollover count override to use when converting all legacy 10-bit GPS week numbers. Setting this value will disable all internal week rollover sources and should only be used for testing . Set to -1 to disable.
Ionosphere Config	54	Ionosphere Delay Model (3.1.5.7)	Configures the active ionosphere model.
Troposphere Config	55	Troposphere Delay Model (3.1.5.8)	Configures the active troposphere model.
Interface Config	200	InterfaceConfig (3.1.5.6)	Change a configuration setting for a specified output interface.
UART 1 Baud Rate	256	u32	The UART 1 serial baud rate in bits/second. Deprecated: Use InterfaceConfig.
UART 2 Baud Rate	257	u32	The UART 2 serial baud rate in bits/second. Deprecated: Use InterfaceConfig.
UART 1 Diag Enable	258	bool	Enable/disable diagnostic output on UART 1. Deprecated: Use InterfaceConfig.
UART 2 Diag Enable	259	bool	Enable/disable diagnostic output on UART 2. Deprecated: Use InterfaceConfig.
Reserved	260-299		<i>Reserved for future use.</i>
Watchdog Enable	300	bool	Enable/disable watchdog timer to restart automatically after fatal errors.
Reserved	301-65535		<i>Reserved for future use.</i>

Expected Response(s):

3.2.1 COMMANDRESPONSE (13000)



Examples

Example 1: Set the output lever arm to [0.6, 0.0, 1.2] meters

```
2e31 0000 8cd8 859a 0200 2c33 0000 0000
1400 0000 0000 0000 1300 0000 0c00 0000
9a99 193f 0000 0000 9a99 993f
```

Example 2: Set the device orientation to point towards the left side of the vehicle and save to persistent storage immediately.

```
2e31 0000 0ac0 a42c 0200 2c33 0000 0000
0c00 0000 0000 0000 1100 0100 0400 0000
0204 0000
```

Example 3: Set the device UART 1 serial baud rate to 115200 bits/second.

```
2e31 0000 61ad a75c 0200 2c33 0000 0000
0c00 0000 0000 0000 0001 0100 0400 0000
00c2 0100
```

The response will be sent at the new baud rate if the command was sent on the effected UART.

Note that in practice, the message sequence number should be modified, consistent with any commands sent previously, and the CRC should be updated to reflect the change.

3.1.5.1 Three-Dimensional Vector (Point3f)

Field	Data Type	Description
X	f32	The X axis offset in meters.
Y	f32	The Y axis offset in meters.
Z	f32	The Z axis offset in meters.

3.1.5.2 Device Coarse Orientation (CoarseOrientation)

A device's orientation is defined by specifying how the +X and +Z axes of its IMU are aligned with the vehicle body axes. The vehicle body is defined with +X facing forward, +Y facing left, and +Z facing up (forward-left-up).

Field	Data Type	Description
X Direction	u8	The direction of the +X axis relative to the vehicle body axes.
Z Direction	u8	The direction of the +Z axis relative to the vehicle body axes.
Reserved	u8[2]	Reserved for future use.

Direction:



Name	Value	Description
Forward	0	Aligned with the vehicle +X axis.
Backward	1	Aligned with the vehicle -X axis.
Left	2	Aligned with the vehicle +Y axis.
Right	3	Aligned with the vehicle -Y axis.
Up	4	Aligned with the vehicle +Z axis.
Down	5	Aligned with the vehicle -Z axis.
Reserved	6-254	<i>Reserved for future use.</i>
Invalid	255	Invalid direction value.

3.1.5.3 Vehicle Details and Physical Dimensions (VehicleDetails)

This payload is used to configure the vehicle model from a list of supported vehicles, along with physical dimensions, used when applying differential wheel speed measurements.

Field	Data Type	Description
Model	u16	The vehicle model if one of the supported vehicles in the list below. Used to configure CAN data support. Set to 0 (unknown) if not one of the currently supported types.
Reserved	u8[10]	<i>Reserved for future use.</i>
Wheelbase	f32	The distance between the front and rear axles in meters.
Front Track Width	f32	The distance between the two front wheels in meters.
Rear Track Width	f32	The distance between the two rear wheels in meters.



Vehicle models:

Value	Description
0	Unknown vehicle type
1	Dataspeed CD4 drive-by-wire system
2	J1939 CAN message standard
20	Lexus CT200H
40	Kia Sorento
41	Kia Sportage
60	Audi Q7
61	Audi A8L
80	Tesla Model X
81	Tesla Model 3
100	Hyundai Elantra
120	Peugeot 206
140	MAN TGX
160	Faction
180	Lincoln MKZ
200	BMW 7-Series

CAN message definitions vary by manufacturer, model, and year. The vehicles listed above may not work with every available model year.

All values not listed in the table above are reserved for future use.



3.1.5.4 Software Wheel Measurement Configuration (WheelConfig)

The `WheelConfig` payload is intended for use on vehicles where wheel speed or angle (tick) data is received via software, either using FusionEngine measurement messages, or from another software mechanism such as vehicle CAN bus. Wheel data may be differential (measurements from each individual wheel), or scalar (a single speed measurement for the vehicle body).

When using software wheel data, you must also specify [3.1.5.3 Vehicle Details and Physical Dimensions \(VehicleDetails\)](#).

See section [3.5 Vehicle Sensor Measurements](#) for descriptions of the messages used to send wheel measurements to the device.

Note: Do not use this message for vehicles using a hardware wheel tick voltage signal. Instead, use [3.1.5.5 Hardware Wheel Tick Configuration \(HardwareTickConfig\)](#).

Field	Data Type	Description
Wheel Sensor Type	<code>WheelSensorType</code> (u8)	The type of vehicle/wheel speed measurements produced by the vehicle.
Applied Speed Type	<code>AppliedSpeedType</code> (u8)	The type of vehicle/wheel speed measurements to be applied to the navigation solution.
Steering Type	<code>SteeringType</code> (u8)	Indication of which of the vehicle's wheels are steered.
Reserved	u8	<i>Reserved for future use.</i>
Wheel Update Interval	f32	The nominal rate at which wheel speed measurements will be provided in seconds.
Wheel Tick Output Update Interval	f32	The nominal rate at which wheel tick measurements will be provided (in seconds)
Steering Ratio	f32	The ratio between the angle of the steering wheel and the angle of the wheels on the ground.
Meters/Tick Scale Factor	f32	The scale factor to convert wheel encoder ticks to distance in meters/tick. Used for <code>TICKS</code> wheel sensor type only.



Wheel Tick Max Value	u32	<p>The maximum value before the wheel tick measurement will roll over.</p> <p>Rollover behavior depends on the value of the Wheel Ticks Signed setting. For example, a maximum value of 10 will work as follows:</p> <ul style="list-style-type: none"> • Wheel Ticks Signed (true): [-11, 10] • Wheel Ticks Signed (false): [0, 10] <p>Signed values are assumed to be asymmetric, consistent with a typical 2's complement rollover.</p>
Wheel Ticks Signed	u8	1 if the wheel reported wheel tick measurements should be interpreted as signed integers, or 0 if they should be interpreted as unsigned integers.
Wheel Ticks Always Increase	u8	1 if the wheel tick measurements increase by a positive amount when driving forward or backward. 0 if wheel tick measurements decrease when driving backward.
Reserved	u8[2]	<i>Reserved for future use.</i>

WheelSensorType (u8):

Name	Value	Description
NONE	0	Wheel/vehicle speed data not available.
Reserved	1	<i>Reserved for future use.</i>
TICKS	2	Individual rotation angle measurements for multiple wheels, reported as accumulated encoder ticks. Will be scaled to meters using the Tick/Meter Scale Factor.
WHEEL_SPEED	3	Individual speed measurements for multiple wheels, reported in meters/second.
VEHICLE_SPEED	4	A single value indicating vehicle speed in meters/second.
VEHICLE_TICKS	5	A single wheel rotation angle, reported as accumulated encoder ticks.



Reserved	6-255	<i>Reserved for future use.</i>
----------	-------	---------------------------------

AppliedSpeedType (u8):

Name	Value	Description
NONE	0	Speed data not applied to the navigation solution.
REAR_WHEELS	1	Apply rear wheel speed data. Recommended for most use cases.
FRONT_WHEELS	2	Apply front wheel speed data.
FRONT_AND_REAR_WHEELS	3	Apply both front and rear wheel speed data.
VEHICLE_BODY	4	Apply aggregate vehicle speed data.
Reserved	5-255	<i>Reserved for future use.</i>

SteeringType (u8):

Name	Value	Description
UNKNOWN	0	Steered wheels unknown/not used.
FRONT	1	Front wheels are steered.
FRONT_AND_REAR	2	Front and rear wheels are steered.
Reserved	3-255	<i>Reserved for future use.</i>

3.1.5.5 Hardware Wheel Tick Configuration (HardwareTickConfig)

The `HardwareTickConfig` payload is intended for use on vehicles with a physical voltage signal, generated by a wheel encoder, producing a series of voltage pulses (encoder ticks) as the vehicle's wheel rotates. These ticks will be captured by the device on an input pin and used to indicate vehicle speed.

Additionally, an optional voltage signal may be provided to indicate vehicle direction. **If this signal is not connected to a real pulse (even if pulled up/down), the `TickDirection` field MUST be set to OFF otherwise there will be substantial errors in dead reckoning.**

Note: DO NOT use this configuration for vehicles using software wheel speed/tick information, including data sent using FusionEngine messages or a vehicle CAN bus. Instead, use [3.1.5.4 Software Wheel Measurement Configuration \(WheelConfig\)](#).

Field	Data Type	Description
Tick Mode	<code>TickMode</code> (u8)	The signal edge to use when capturing the incoming wheel tick voltage signal.
Tick Direction	<code>TickDirection</code> (u8)	The way to interpret the optional direction voltage signal.
Reserved	u8	<i>Reserved for future use.</i>
Meters/Tick Scale Factor	f32	The scale factor to convert wheel encoder ticks to distance in meters/tick. Used for <code>TICKS</code> and <code>TICK_RATE</code> wheel sensor types only.

`TickMode` (u8):

Name	Value	Description
OFF	0	Hardware wheel tick capture disabled.
<code>RISING_EDGE</code>	1	Capture a tick on the rising edge of the incoming pulse.
<code>FALLING_EDGE</code>	2	Capture a tick on the falling edge of the incoming pulse.
Reserved	3-255	<i>Reserved for future use.</i>



TickDirection (u8):

Name	Value	Description
OFF	0	Vehicle direction signal disabled.
FORWARD_ACTIVE_HIGH	1	Assume vehicle is moving forward when direction signal voltage is high, and backward when voltage is low.
FORWARD_ACTIVE_LOW	2	Assume vehicle is moving forward when direction signal voltage is low, and backward when voltage is high.
Reserved	3-255	<i>Reserved for future use.</i>



3.1.5.6 Interface Config Submessage (InterfaceConfigSubmessage)

Message used to configure a particular interface on the device. This includes functions such as baud rate, diagnostics, etc.

Field	Data Type	Description
Interface ID	InterfaceID (u32)	The Interface ID to target. Note that type <i>ALL</i> is <i>not supported</i> .
Interface Config Type	InterfaceConfigType (u8)	The Interface Setting corresponding to this submessage.
Reserved	u8[3]	<i>Reserved for future use.</i>
Interface Config Payload Data	<i>variable</i>	Each Interface Config Type has a different config data content and length specified in the Interface Config Type table below.

InterfaceID (u32):

Describes a I/O interface, such as serial port or a TCP Server.

Field	Data Type	Description
Transport Type	TransportType (u8)	The type of the IO interface.
Index	u8	The ID of the IO interface. For UARTs, typically UART1 is index 1.
Reserved	u8[2]	<i>Reserved for future use.</i>

TransportType (u8):

Describes a type of I/O interface.

Name	Value	Description
Invalid	0	Not used
Serial Port	1	A serial data interface (e.g. an RS232 connection)
File	2	An interface that writes to or reads from a file
TCP Client	3	A TCP Client usually on ethernet/Wi-Fi
TCP Server	4	A TCP Server usually on ethernet/Wi-Fi
UDP Client	5	A UDP Client usually on ethernet/Wi-Fi



UDP Server	6	A UDP Server usually on ethernet/Wi-Fi
Current	254	The interface on which the command was received
All	255	Wildcard to get/set ALL interfaces. <i>Not supported for all commands.</i>

InterfaceConfigType (u8):

Name	Value	Payload Format	Description
OUTPUT_DIAGNOSTICS_MESSAGES	1	bool	Enable/disable output of diagnostic data on this interface. Note this will override the message rate/off settings for some FusionEngine messages.
BAUD_RATE	2	u32	Serial Baud Rate in bits/second. Only applies to SERIAL PORT transports.
REMOTE_ADDRESS	3	char[64]	Configure the network address for a client to connect to. Only applies to TCP and UDP Client transports. String must be NULL terminated.
PORT	4	u16	The network port for a client or server. Applies to TCP/UDP Client/Server transports.



3.1.5.7 Ionosphere Delay Model

Model used to predict ionospheric delays.

IonoDelayModel (u8):

Name	Value	Description
AUTO	0	Uses the best available ionospheric model.
OFF	1	Explicitly disables an ionospheric model. This is not recommended except for use in simulation.
KLOBUCHAR	2	Uses the Klobuchar ionospheric model.
Reserved	3-255	<i>Reserved for future use.</i>

3.1.5.8 Troposphere Delay Model

Model used to predict tropospheric delays.

TropoDelayModel (u8):

Name	Value	Description
AUTO	0	Uses the best available tropospheric model.
OFF	1	Explicitly disables a tropospheric model. This is not recommended except for use in simulation.
SAASTAMOINEN	2	Uses the Saastamoinen ionospheric model.
Reserved	3-255	<i>Reserved for future use.</i>



3.1.5.9 Heading Bias

Used to configure known offsets in the heading output from dual-antenna setups.

Field	Data Type	Description
Horizontal Bias (Degrees)	HorizontalBiasDeg (f32)	<p>Bias is defined as the angle between the vector pointing from the primary GNSS antenna to the secondary heading antenna, and the vector pointing from the primary antenna pointing in the forward direction of the vehicle. A positive angle means the secondary antenna is offset in a counter-clockwise direction from the forward vector (i.e. a positive yaw rotation).</p> <p>For example, if the primary antenna is in the back of the vehicle and the secondary antenna is in the front, a positive bias angle would indicate that the secondary antenna is offset to the left side of the vehicle.</p>
Vertical Bias (Degrees)	VerticalBiasDeg (f32)	<p>The offset between the heading measured by a secondary GNSS device and the vehicle's direction of motion in the vertical plane (defined by the vehicle's forward and up axes).</p> <p>A positive angle means the secondary antenna is offset in the downward direction. For example, if the primary antenna is in the back of the vehicle and the secondary antenna is in the front, a positive bias angle would indicate that the secondary antenna is mounted below the primary antenna.</p>

3.1.6 GetConfig (13101)

Message ID: 13101

Protocol Version: 2

Message Version: 0

Description: Query the value of a user configuration parameter.

Field	Data Type	Description
Parameter Type	ConfigType (u16)	The parameter to be queried (see 3.1.5 SetConfig (13100)).



Configuration Source	Configuration Source (u8)	The data source to be queried (active, saved, etc.).
Reserved	u8	<i>Reserved for future use.</i>

ConfigurationSource (u8):

Configuration Source	Value	Description
ACTIVE	0	Active configuration settings currently in use by the device.
SAVED	1	Settings currently saved in persistent storage.

Expected Response(s):

3.2.5 CONFIGRESPONSE (13103)

3.1.7 SaveConfig (13102)

Message ID: 13102

Protocol Version: 2

Message Version: 0

Description: Save or reload configuration settings.

Field	Data Type	Description
Save Action	SaveAction (u8)	Indicates the action to perform on the saved configuration.
Reserved	u8[3]	Reserved for future use.

SaveAction (u8):

Save Action	Value	Description
Save	0	Save all active parameters to persistent storage.
Revert to Saved	1	Revert the active configuration to previously saved values.
Revert to Default	2	Reset the active and saved configuration to default values.

Expected Response(s):

3.2.1 COMMANDRESPONSE (13000)

Example Usage

Example 1: Save configuration changes to persistent storage

2e31	0000	c48c	1ee8	0200	2e33	0000	0000
0400	0000	0000	0000	0000	0000	0000	0000

Example 2: Reset configuration settings to factory defaults

2e31	0000	4f44	1742	0200	2e33	0000	0000
0400	0000	0000	0000	0200	0000	0000	0000

This has the same effect as issuing a [3.1.2 ResetRequest \(13002\)](#) with the bit mask set to RESET_CONFIG.



3.1.8 SetMessageRate (13220)

Message ID: 13220

Protocol Version: 2

Message Version: 0

Description: Enable/disable an individual message on a specified output interface or configure its output rate.

Some messages can only be generated on change, and do not support rate control requests.

Note: If the `Diag Enable` configuration setting is enabled for the interface being configured, the specified rate may be temporarily overridden. This can be verified using the [GetMessageRate \(13221\)](#) command.

For a list of u16 message IDs corresponding with supported NMEA-0183 messages, see [2.7 NMEA-0183 Message Types](#).

Field	Data Type	Description
Transport Type	TransportType (u8)	The transport type of the target interface (e.g., serial port).
Index	u8	An identifier desired interface (e.g., 3 for serial port 3).
Reserved	u8[2]	<i>Reserved for future use.</i>
Protocol Type	ProtocolType (u8)	The message protocol to be configured. If ALL (255) is specified, set the rate for all protocols and messages on the specified interface.
Flags	u8	A bitmask of flags to modify the command (see the table below).
Message ID	u16	The ID of the message type to be configured. If ALL (65535) is specified, set the rate for all supported messages for the specified protocol.
Message Rate	MessageRate (u8)	The desired output rate or on/off status. If the rate is an interval, and Message ID or Protocol Type is set to a wildcard, the command will only affect messages that are currently enabled. To modify all messages, including disabled messages, set the FLAG_INCLUDE_DISABLED_MESSAGES flag.



Reserved	u8[3]	<i>Reserved for future use.</i>
----------	-------	---------------------------------

TransportType (u8):

Transport Type	Value
Invalid	0
Serial Port	1
Current Interface	254
All Interfaces	255

ProtocolType (u8):

Protocol	Value
Invalid	0
FusionEngine	1
NMEA	2
RTCM	3
ALL	255

Flag bit definitions:

Flag	Bit Position	Description
APPLY_AND_SAVE	0	Setting this bit saves the updated configuration to persistent storage immediately on success.
FLAG_INCLUDE_DISABLED_MESSAGES	1	Setting this bit applies bulk interval changes to all supported messages instead of only currently enabled messages.
Reserved	2-6	Reserved for future use.

MessageRate (u8):

Message Rate	Value
Off	0
On Change	1
10 ms	2
20 ms	3
40 ms	4
50 ms	5
100 ms	6
200 ms	7
500 ms	8
1 second	9
2 seconds	10
5 seconds	11
10 seconds	12
Device Default	255

NMEA-0183 protocol message IDs:

Message Type	Message ID
Invalid	0
<i>Standard NMEA-0183 Messages</i>	
GGA	1
GLL	2
GSA	3
GSV	4
RMC	5
VTG	6
<i>Point One Proprietary Messages</i>	
P1CALSTATUS	1000
P1MSG	1001
<i>Quectel Proprietary Messages</i>	
PQTMVERNO	1200
PQTMVER	1201
PQTMGNSS	1202

Expected Response(s):

3.2.1 COMMANDRESPONSE (13000)

Example Usage

Example 1: Enable FusionEngine PoseMessage output at its max rate on serial port 1 (UART1)

```
2e31 0000 6a84 7724 0200 a433 0000 0000
0c00 0000 0000 0000 0101 0000 0100 1027
0100 0000
```

Example 2: Enable FusionEngine PoseMessage output at its max rate on serial port 1 (UART1) then save the change to persistent memory.

```
2e31 0000 de8f 0082 0200 a433 0000 0000
0c00 0000 0000 0000 0101 0000 0101 1027
0100 0000
```



Example 3: Set all message rates on all interfaces to their factory default values.

```
2e31 0000 aa8d 150d 0200 a433 0000 0000
0c00 0000 0000 0000 ff00 0000 ff02 ffff
ff00 0000
```

Note that if `FLAG_INCLUDE_DISABLED_MESSAGES` is not set, only the messages already enabled will be modified.

Example 4: On UART 1, set all enabled NMEA messages that support rate control to output at 1Hz.

```
2e31 0000 a936 db31 0200 a433 0000 0000
0c00 0000 0000 0000 0101 0000 0200 ffff
0900 0000
```



3.1.9 GetMessageRate (13221)

Message ID: 13221

Protocol Version: 2

Message Version: 0

Description: Query the configuration for the requested message type on the specified interface.

Field	Data Type	Description
Transport Type	TransportType (u8)	The transport type of the target interface (e.g., serial port). Transport type ALL is not supported for GetMessageRate requests.
Index	u8	An identifier desired interface (e.g., 3 for serial port 3).
Reserved	u8[2]	<i>Reserved for future use.</i>
Protocol Type	ProtocolType (u8)	The message protocol to be queried. If the ALL (255) is specified, return the rate for all protocols and messages on the specified interface.
Configuration Source	ConfigurationSource (u8)	The data source to be queried.
Message ID	u16	The ID of the message type to be queried. If ALL (65535) is specified, return the rate for all supported messages for the specified protocol.

ConfigurationSource (u8):

Configuration Source	Value	Description
ACTIVE	0	Active configuration settings currently in use by the device.
SAVED	1	Settings currently saved in persistent storage.

Expected Response(s):

3.2.6 MessageRateResponse (13222)



3.1.10 GetDeviceID (13007)

Message ID: 13007

Protocol Version: 2

Message Version: 0

Description: Contains various device identifiers in the host device. Note: The strings are **not** null terminated.

Field	Data Type	Description
System Time	i64	System Time in nanoseconds.
Device Type	DeviceType (u8)	The type of device responding.
HW ID Length	u8	Length of the HW ID Field.
User ID Length	u8	Length of the User ID Field.
GNSS ID Length	u8	Length of the GNSS ID Field.
Reserved	u8 [4]	Reserved for future use.
HW ID	str Length of HW ID Length	The value of the HW ID.
User ID	str Length of User ID Length	The value of the User ID.
GNSS ID	str Length of GNSS ID Length	The value of the GNSS ID.

DeviceType (u8):

Response Type	Response Code
Unknown	0
Atlas	1
LG69T_AM	2
LG69T-AP	3
LG69T-AH	4



3.2 Command Response Messages (Output)

These are messages out of the module to the host device. This will include navigation information, system health, and other output useful for understanding the state of the device.

3.2.1 CommandResponse (13000)

Message ID: 13000

Protocol Version: 2

Message Version: 0

Description: Response to Input Messages

Field	Data Type	Description
Source Sequence Number	u32	Sequence number of the Request message being responded to.
Response Code	ResponseCode (u8)	Indicates the type of the response.
Reserved	u8 [3]	<i>Reserved for future use.</i>

ResponseCode (u8):

Response Type	Response Code
OK	0
Unsupported Message Version	1
Unsupported Feature	2
Value Error	3
Insufficient Space	4
Execution Failure	5
Inconsistent Payload Length	6
Data Corrupted	7



3.2.2 VersionInformation (13003)

Message ID: 13003

Protocol Version: 2

Message Version: 0

Description: Describes the version of the system and its subcomponents.

Note that the strings contained in this message are not null terminated.

Field	Data Type	Description
System Time	i64	Current system time (in nanoseconds).
Firmware Version Length	u8	The length of the firmware string (in bytes).
Engine Version Length	u8	The length of the engine string (in bytes).
Operating System (OS) Version Length	u8	The length of the operating system string (in bytes).
Receiver Version Length	u8	The length of the receiver string (in bytes).
Reserved	u8[4]	<i>Reserved for future use.</i>
Firmware String Description	str	ASCII string representing the firmware version.
Engine String Description	str	ASCII string representing the navigation software version (FusionEngine).
Operating System (OS) String Description	str	ASCII string representing the operating system, kernel, and/or bootloader version.
Receiver String Description	str	ASCII string representing the software version of the underlying measurement engine.
<Padding>	u8	This message will be padded to a length that is a multiple of 4 bytes.



3.2.4 EventNotification (13004)

Message ID: 13004

Protocol Version: 2

Message Version: 0

Description: Notification of a system event for logging purposes.

Field	Data Type	Description
Event Type	EventType (u8)	The type of event that occurred.
Reserved	u8[3]	<i>Reserved for future use.</i>
System Time	i64	The system time when the event occurred (in nanoseconds).
Flags	u64	Flags associated with the event (if applicable).
Description Length	u16	The length of the event description string (in bytes).
Description	str	ASCII string description of the event.

EventType (u8):

Event Type	Value	Description
LOG	0	Event containing a logged message string from the device.
RESET	1	Event indicating a device reset occurred. The event flags will be set to the requested reset bitmask. The payload will contain a string describing the cause.
CONFIG_CHANGE	2	Notification that the user configuration has been changed.
COMMAND	3	Notification that the user performed a command (e.g., configuration request, fault injection enable/disable).
COMMAND_RESPONSE	4	Record containing the response to a user command. Response events are not output on the interface on which the command was received; that interface will receive the response itself.



3.2.5 ConfigResponse (13103)

Message ID: 13103

Protocol Version: 2

Message Version: 0

Description: The response to a 3.1.6 GETCONFIG (13101) message.

If an invalid or unsupported parameter was requested, the parameter type in the response will be set to `INVALID`, and the Response Code will indicate the reason for the failure.

Field	Data Type	Description
Configuration Source	Configuration Source (u8)	The source of the parameter value (active, saved, etc.).
Active Differs From Saved	u8	1 if the active value for this parameter differs from the value saved in persistent storage. 0 if the active and saved values are the same.
Parameter Type	ConfigType (u16)	The type of parameter being returned (see 3.1.5 SetConfig (13100)). Set to <code>INVALID</code> if the 3.1.6 GetConfig (13101) request was rejected.
Response Code	ResponseCode (u8)	Indicates the status of the response on success or failure (see 3.2.1 CommandResponse (13000)).
Reserved	u8[3]	<i>Reserved for future use.</i>
Value Length	u32	The length of the value to follow (in bytes).
Value	Variable	The current value of the requested parameter. The format of the value is listed in the ConfigType table in 3.1.5 SetConfig (13100)

ConfigurationSource (u8):

Configuration Source	Value	Description
ACTIVE	0	Active configuration settings currently in use by the device.
SAVED	1	Settings currently saved in persistent storage.



3.2.6 MessageRateResponse (13222)

Message ID: 13222

Protocol Version: 2

Message Version: 0

Description: Response to a [3.1.9 GetMessageRate \(13221\)](#) message.

Field	Data Type	Description
Configuration Source	ConfigurationSource (u8)	The source of the parameter value (active, saved, etc.).
Response Code	ResponseCode (u8)	Indicates the type of the response.
Number of Rates	u16	The number of rates being reported in this message.
Transport Type	TransportType (u8)	The transport type of the interface being reported (e.g., serial port).
Index	u8	An identifier for interface (e.g., 3 for serial port 3) being reported.
Reserved	u8[2]	<i>Reserved for future use.</i>
Rates	MessageRateResponseEntry [Number of Rates]	A repeated field containing Number of Rates entries, one for each message in the response.

3.2.6.1 Message Rate Response Entry (*MessageRateResponseEntry*)

Field	Data Type	Description
Protocol Type	ProtocolType (u8)	The message protocol.
Flags	u8	A bitmask of flags to modify the command (see the table below).
Message ID	u16	The ID of the message type.
Configured Rate	MessageRate (u8)	The configured output rate or on/off status.
Effective Rate	MessageRate (u8)	The actual output rate for this message, factoring in effects of additional configuration settings that may override the configured rate such as enabling diagnostic output. In general, this will be equal to the configured rate.
Reserved	u8 [2]	<i>Reserved for future use.</i>

Flag bit definitions:

Flag	Bit Position	Description
FLAG_ACTIVE_DIFFERS_FROM_SAVED	0	Flag to indicate the active value for this configuration differs from the value saved to persistent memory.
Reserved	1-7	<i>Reserved for future use.</i>



3.2.7 SupportedInterfaces (13223)

Message ID: 13223

Protocol Version: 2

Message Version: 0

Description: A list of transport interfaces supported by the device. This message is sent in response to a [3.1.1 MessageRequest \(13001\)](#) message.

Field	Data Type	Description
Number of Interfaces	u8	Number of interfaces, N, reported by this message
Response Code	ResponseCode (u8)	Indicates the type of the response.
Reserved	u8 [7]	<i>Reserved for future use.</i>
Interface Descriptions	InterfaceID [N]	Descriptions of each interface supported by the device. E.g., Serial 1, Serial 2, TCP, etc.

3.3 Navigation Solution Messages (Output)

3.3.1 Pose (10000)

Message ID: 10000

Protocol Version: 2

Message Version: 1

Description: Position, velocity, and attitude (orientation) solution.

Field	Data Type	Description
P1 Time	Timestamp (8B)	The time of the solution, expressed in the P1 timebase.
GPS Time	Timestamp (8B)	The GPS time of the solution, if available.
Solution Type	SolutionType (u8)	The type of the position solution (see below).
Reserved	u8	<i>Reserved for future use.</i>
Undulation	s16	The geoid undulation at the current location in cm, defined as follows: Orthometric Height = Ellipsoid Height - Undulation <i>Set to -32768 if invalid.</i>
Latitude	f64	The geodetic latitude in degrees, expressed using the WGS-84 reference ellipsoid.
Longitude	f64	The geodetic longitude in degrees, expressed using the WGS-84 reference ellipsoid.
Height	f64	The geodetic height above the ellipsoid in meters, expressed using the WGS-84 reference ellipsoid.
Position Std Dev (East)	f32	The standard deviation of the position in the East direction in meters.
Position Std Dev (North)	f32	The standard deviation of the position in the North direction in meters.
Position Std Dev (Up)	f32	The standard deviation of the position in the Up direction in meters.



Yaw	f64	<p>The platform yaw in degrees, expressed as intrinsic Euler-321 angles (applied: yaw, then pitch, then roll) with respect to the local ENU tangent plane.</p> <p>Positive yaw indicates a left turn, negative yaw indicates a right turn. Yaw is measured from East in a counterclockwise direction (e.g., North is +90 degrees yaw).</p>
Pitch	f64	<p>The platform pitch in degrees. Positive pitch points the nose of the vehicle down, negative pitch points the nose up.</p>
Roll	f64	<p>The platform roll in degrees. Positive roll indicates a roll toward the right, negative roll is a roll to toward the left.</p>
Yaw Std Dev	f32	Yaw standard deviation in degrees.
Pitch Std Dev	f32	Pitch standard deviation in degrees.
Roll Std Dev	f32	Roll standard deviation in degrees.
Forward Velocity	f64	Body forward velocity in meters/second. Set to <code>NAN</code> if body orientation information is not available.
Left Velocity	f64	Body left velocity in meters/second. Set to <code>NAN</code> if body orientation information is not available.
Up Velocity	f64	Body up velocity in meters/second. Set to <code>NAN</code> if body orientation information is not available.
Forward Velocity Std Dev	f32	Forward velocity standard deviation in meters/second.
Left Velocity Std Dev	f32	Left velocity standard deviation in meters/second.
Up Velocity Std Dev	f32	Up velocity standard deviation in meters/second.
Aggregate Protection Level	f32	The estimated 3D protection level in meters.
Horizontal Protection Level	f32	The estimated 2D horizontal protection level in meters.
Vertical Protection Level	f32	The estimated vertical level in meters.



SolutionType (u8):

Type	Value	Description
Invalid	0	Invalid, no position available.
AutonomousGPS	1	Standalone GNSS fix, no corrections data used.
DGPS	2	Differential GNSS pseudorange solution using a local RTK base station, SSR corrections data, or SBAS corrections data.
Reserved	3	<i>Reserved for future use.</i>
RTK Fixed	4	GNSS RTK solution with fixed integer carrier phase ambiguities (most accurate position solution).
RTK Float	5	GNSS RTK solution with floating point carrier phase ambiguities.
Integrate	6	Dead reckoned position, computed by integrating IMU data or an internal motion model.
Reserved	7-8	<i>Reserved for future use.</i>
Visual	9	Position solution using vision measurement inputs.
PPP	10	GNSS precise point positioning (PPP) pseudorange/carrier phase solution.
Reserved	11-255	<i>Reserved for future use.</i>



3.3.2 GNSSInfo (10001)

Message ID: 10001

Protocol Version: 2

Message Version: 1

Description: Information about the GNSS data used in the corresponding 3.3.1 POSE (10000) message.

Field	Data Type	Description
P1 Time	Timestamp (8B)	The time of the solution, expressed in the P1 timebase.
GPS Time	Timestamp (8B)	The GPS time of the solution.
Leap Second	u8	The current UTC leap second (offset between UTC and GPS time), if known. Set to 0xFF if invalid.
Number of Satellites	u8	The number of satellites used in the current position solution.
Reserved	u8[2]	Reserved
Corrections Age	u16	The age of the most recently received GNSS corrections data (in 0.1 seconds). Set to 0xFFFF if invalid.
Baseline Distance	u16	The distance between the device and the GNSS corrections base station. Stored in units of 10 meters: $\text{baseline_distance_m} = \text{baseline_distance} * 10$ Set to 0xFFFF if invalid.
Reference Station ID	u32	The ID of the differential base station, if used. Set to 0xFFFFFFFF if invalid.
GDOP	f32	The geometric dilution of precision (GDOP).
PDOP	f32	The position dilution of precision (PDOP).
HDOP	f32	The horizontal dilution of precision (HDOP).
VDOP	f32	The vertical dilution of precision (VDOP).
GPS Time Std Dev	f32	The standard deviation of the GPS time solution (i.e., the GPS clock bias estimate) in seconds.



3.3.3 GNSSSatellite (10002)

Message ID: 10002

Protocol Version: 2

Message Version: 1

Description: Information about the GNSS satellites used in the 3.3.1 Pose (10000) message.

Field	Data Type	Description
P1 Time	Timestamp (8B)	The time of the solution, expressed in the P1 timebase.
GPS Time	Timestamp (8B)	The GPS time of the solution.
Number Of Satellites	u16	The total number of available satellites (N).
Reserved	u8[2]	<i>Reserved for future use.</i>
Satellite Infos	<i>SatelliteInfo</i> [N]	A repeated instance of SatelliteInfo blocks describing the satellites used in the solution.

SatelliteInfo:

Field	Data Type	Description
Satellite Type	SatelliteType (u8)	The GNSS constellation to which this satellite belongs (see Section 2.6.5)
PRN	u8	The satellite identifier (PRN).
Usage Mask	u8	A bitmask indicating how this satellite was used in the position solution (see below).
C/N0	u8	The carrier-to-noise density ratio (C/N0) for the L1 signal on the satellite in dB-Hz. Set to 0 if invalid. If the satellite is not tracking L1 (or the near-L1 equivalent signal for the constellation), but is tracking a signal on another frequency, that signal's C/N0 value will be reported.
Azimuth	f32	The azimuth of the satellite in degrees.
Elevation	f32	The elevation of the satellite in degrees.



Usage mask:

Bit	Description
0	One or more signals from this satellite used in the position solution.
1-7	<i>Reserved for future use.</i>

3.3.4 PoseAux (10003)

Message ID: 10003

Protocol Version: 2

Message Version: 0

Description: Auxiliary position solution information.

Field	Data Type	Description
P1 Time	Timestamp (8B)	The time of the solution, expressed in the P1 timebase.
Position Std Dev (Forward)	f32	The standard deviation of the position resolved in the vehicle body forward direction in meters.
Position Std Dev (Left)	f32	The standard deviation of the position resolved in the vehicle body left direction in meters.
Position Std Dev (Up)	f32	The standard deviation of the position resolved in the vehicle body up direction in meters.
Position Covariance	f64[9]	The 3x3 position covariance matrix (in meters ²), resolved in the local ENU frame and stored in row-major order.
Attitude Quaternion	f64[4]	The platform body orientation with respect to the local ENU frame, represented as a quaternion with the scale component last (x, y, z, w).
East Velocity	f64	Platform velocity in the East direction in meters/second.
North Velocity	f64	Platform velocity in the North direction in meters/second.
Up Velocity	f64	Platform velocity in the Up direction in meters/second.
East Velocity Std Dev	f32	East velocity standard deviation in meters/second.
North Velocity Std Dev	f32	North velocity standard deviation in meters/second.
Up Velocity Std Dev	f32	Up velocity standard deviation in meters/second.



3.3.5 CalibrationStatus (10004)

Message ID: 10004

Protocol Version: 2

Message Version: 1S

Description: Current device calibration status.

Field	Data Type	Description
P1 Time	Timestamp (8B)	The time of the solution, expressed in the P1 timebase.
Calibration Stage	CalibrationStage (u8)	The current calibration stage (see below).
Reserved	u8 [3]	<i>Reserved for future use.</i>
Yaw Mounting Angle	f32	The estimated IMU yaw mounting angle offset in degrees.
Pitch Mounting Angle	f32	The estimated IMU pitch mounting angle offset in degrees.
Roll Mounting Angle	f32	The estimated IMU roll mounting angle offset in degrees.
Yaw Std Dev	f32	The estimated yaw mounting angle standard deviation in degrees.
Pitch Std Dev	f32	The estimated pitch mounting angle standard deviation in degrees.
Roll Std Dev	f32	The estimated roll mounting angle standard deviation in degrees.
Travel Distance	f32	The accumulated calibration travel distance in meters.
Reserved	u8 [24]	<i>Reserved for future use.</i>
State Verified	u8	1 if the navigation engine state has been verified after initialization, 0 otherwise.
Reserved	u8 [3]	<i>Reserved for future use.</i>
Gyro Bias Percent	u8	The completion percentage for IMU gyro bias estimation, stored in units of 0.5% (range [0, 200]).
Accel Bias Percent	u8	The completion percentage for IMU accelerometer bias estimation, stored in units of 0.5% (range [0, 200]).
Mounting Angle Bias Percent	u8	The completion percentage for IMU mounting angle estimation, stored in units of 0.5% (range [0, 200]).
Reserved	u8 [5]	<i>Reserved for future use.</i>
Min Travel Distance	f32	The minimum travel distance needed to complete mounting angle calibration in meters.
Max Yaw Std Dev	f32	The maximum yaw mounting angle standard deviation (in degrees), above which calibration is incomplete.
Max Pitch Std Dev	f32	The maximum pitch mounting angle standard deviation (in degrees), above which calibration is incomplete.
Max Roll Std Dev	f32	The maximum roll mounting angle standard deviation (in degrees), above which calibration is incomplete.



CalibrationStage (u8):

Stage	Value	Description
Unknown	0	Calibration stage not known.
Mounting Angle	1	Estimating IMU mounting angles.
Reserved	2-254	<i>Reserved for future use.</i>
Done	255	Calibration complete.



3.3.6 RelativeENUPosition (10005)

Message ID: 10005

Protocol Version: 2

Message Version: 0

Description: Rover position relative to a local base station expressed in the ENU frame.

Field	Data Type	Description
P1 Time	Timestamp (8B)	The time of the solution, expressed in the P1 timebase.
GPS Time	Timestamp (8B)	The GPS time of the solution.
Solution Type	SolutionType (u8)	The type of the position solution (see 3.3.1 Pose (10000)).
Reserved	u8[3]	<i>Reserved for future use.</i>
Reference Station ID	u32	The ID of the differential base station, if used. Set to 0xFFFFFFFF if invalid.
East Position	f64	The position offset in the East direction relative to the local base station position (in meters).
North Position	f64	The position offset in the North direction relative to the local base station position (in meters).
Up Position	f64	The position offset in the Up direction relative to the local base station position (in meters).
Position Std Dev (East)	f32	The standard deviation of the position in the East direction in meters.
Position Std Dev (North)	f32	The standard deviation of the position in the North direction in meters.
Position Std Dev (Up)	f32	The standard deviation of the position in the Up direction in meters.



3.3.7 ROSPose (12000)

Message ID: 12000

Protocol Version: 2

Message Version: 0

Description: Position, velocity, and attitude (orientation) solution, in ROS Pose message format. Note that this message is designed to be easily convertible to ROS datatypes but cannot be directly cast into ROS datatypes.

Field	Data Type	Description
P1 Time	Timestamp (8B)	The time of the solution, expressed in the P1 timebase.
East Position	f64	The position offset in the East direction relative to the local base station position (in meters).
North Position	f64	The position offset in the North direction relative to the local base station position (in meters).
Up Position	f64	The position offset in the Up direction relative to the local base station position (in meters).
Attitude Quaternion	f64[4]	The platform body orientation with respect to the local ENU frame, represented as a quaternion with the scale component last (x, y, z, w).

3.3.8 ROSGPSFix (12010)

Message ID: 12000

Protocol Version: 2

Message Version: 0

Description: Information about the position, altitude, and error in the ROS GPSFix message format. Note that this message is designed to be easily convertible to ROS datatypes but cannot be directly cast into ROS datatypes.

Field	Data Type	Description
P1 Time	Timestamp (8B)	The time of the solution, expressed in the P1 timebase.
Latitude	f64	The geodetic latitude in degrees, expressed using the WGS-84 reference ellipsoid.



Longitude	f64	The geodetic longitude in degrees, expressed using the WGS-84 reference ellipsoid.
Height	f64	The geodetic height above the ellipsoid in meters, expressed using the WGS-84 reference ellipsoid.
Track Angle	f64	The vehicle direction from north in degrees.
Vehicle Speed	f64	The vehicle ground speed in meters/second.
Climb Speed	f64	The vehicle vertical speed in meters/second.
Pitch	f64	The platform pitch in degrees. Positive pitch points the nose of the vehicle down, negative pitch points the nose up.
Roll	f64	The platform roll in degrees. Positive roll indicates a roll toward the right, negative roll is a roll to toward the left.
Yaw	f64	<p>The platform yaw in degrees, expressed as intrinsic Euler-123 angles (applied: yaw, then pitch, then roll) with respect to the local ENU tangent plane.</p> <p>Positive yaw indicates a left turn, negative yaw indicates a right turn. Yaw is measured from East in a counterclockwise direction (e.g., North is +90 degrees yaw).</p>
Dip	f64	The platform dip angle in degrees.
GPS Time	Timestamp (8B)	The GPS time of the solution, if available.
GDOP	f64	The geometric dilution of precision (GDOP).
PDOP	f64	The position dilution of precision (PDOP).
HDOP	f64	The horizontal dilution of precision (HDOP).
VDOP	f64	The vertical dilution of precision (VDOP).
TDOP	f64	The time dilution of precision (TDOP).
3D Error	f64	The spherical position uncertainty in meters.
Horizontal Error	f64	The horizontal position uncertainty in meters.



Vertical Error	f64	The vertical position uncertainty in meters.
Track Error	f64	The track uncertainty in meters.
Vehicle Speed Error	f64	The vehicle ground speed uncertainty in meters/second
Climb Speed Error	f64	The vehicle vertical speed uncertainty in meters/second.
Time Error	f64	The time uncertainty in seconds.
Pitch Error	f64	The platform pitch uncertainty in degrees.
Roll Error	f64	The platform roll uncertainty in degrees.
Dip Error	f64	The platform dip uncertainty in degrees.
Position Covariance	f64[9]	The 3x3 position covariance matrix (in meters ²), resolved in the local ENU frame and stored in row-major order.
Position Covariance Type	u8	The method in which position covariance was populated (see below).
Reserved	u8[3]	<i>Reserved for future use.</i>

CovarianceType (u8):

Type	Value	Description
Unknown	0	Position covariance is unknown.
Approximated	1	Position covariance entries are approximate.
Diagonal Known	2	Position covariance diagonal entries are known.
Known	3	Position covariance is known.
Reserved	4-255	<i>Reserved.</i>



3.4 Sensor Measurement Output Messages

These are the primary sensor outputs that are compensated, calibrated, and filtered outputs of the system. In general these are the best outputs to use for upstream processing. For Raw data, see [Section 3.6](#).

3.4.1 IMUOutput (11000)

Message ID: 11000

Protocol Version: 2

Message Version: 0

Description: Corrected IMU measurement data.

The data contained in this message has been corrected for estimated accelerometer and gyro errors, including biases and scale factors, and has been rotated into the vehicle body frame from the original IMU orientation.

Field	Data Type	Description
P1 Time	Timestamp (8B)	The time of the solution, expressed in the P1 timebase.
X Acceleration	f64	The vehicle X acceleration in meters/second ² .
Y Acceleration	f64	The vehicle Y acceleration in meters/second ² .
Z Acceleration	f64	The vehicle Z acceleration in meters/second ² .
X Acceleration Std Dev	f64	The standard deviation of X acceleration in meters/second ² .
Y Acceleration Std Dev	f64	The standard deviation of Y acceleration in meters/second ² .
Z Acceleration Std Dev	f64	The standard deviation of Z acceleration in meters/second ² .
X Rotation Rate	f64	The vehicle X axis gyroscope rate of rotation in radians/second.
Y Rotation Rate	f64	The vehicle Y axis gyroscope rate of rotation in radians/second.
Z Rotation Rate	f64	The vehicle Z axis gyroscope rate of rotation in radians/second.
X Rotation Std Dev	f64	The standard deviation of the X rate of rotation in radians/second.



Y Rotation Std Dev	f64	The standard deviation of the Y rate of rotation in radians/second.
Z Rotation Std Dev	f64	The standard deviation of the Z rate of rotation in radians/second.

3.4.2 HeadingOutput (11003)

Message ID: 11003

Protocol Version: 2

Message Version: 0

Description: The heading angle (in degrees) with respect to true north as measured by a secondary heading measurement device, pointing from the primary antenna to the secondary antenna.

NOTE: The bias **must** be configured for this message to be published.

Field	Data Type	Description
Measurement Time	Timestamp (8B)	The time of applicability of the measurement data. Interpretation dictated by Time Source.
Time Source	SystemTimeSource (u8)	The source of the measurement timestamp. See Section 2.6.2 .
Reserved	u8[3]	<i>Reserved for future use.</i>
P1 Time	Timestamp (8B)	The calculated P1 time corresponding with the measurement timestamp.
Solution Type	SolutionType (u8)	Invalid: Heading not available RTKFixed: Heading available See section 3.3.1 Pose (10000).
Reserved	u8[3]	<i>Reserved for future use.</i>
Reserved	u32	<i>Reserved for future use.</i>
Yaw	f32	The measured Yaw vector (in degrees), resolved in the ENU frame. Intrinsic Euler-321 rotation, i.e., yaw, pitch, then roll. NaN if bias is not set.



Pitch	f32	The measured Pitch vector (in degrees), resolved in the ENU frame. Intrinsic Euler-321 rotation, i.e., yaw, pitch, then roll. NaN if bias is not set.
Roll	f32	The measured Roll vector (in degrees), resolved in the ENU frame. Intrinsic Euler-321 rotation, i.e., yaw, pitch, then roll. NaN if bias is not set.
Heading	f32	The corrected heading angle (in degrees), with respect to true North, pointing from the primary antenna to the secondary antenna. Set to NaN if unavailable.



3.4.3 ROSIMUMessage (12011)

Message ID: 12011

Protocol Version: 2

Message Version: 0

Description: Corrected IMU measurement data in the ROS IMU message format. This data is an output from the system. Note that this message is designed to be easily convertible to ROS datatypes but cannot be directly cast into ROS datatypes.

The data contained in this message has been corrected for estimated accelerometer and gyro errors, including biases and scale factors, and has been rotated into the vehicle body frame from the original IMU orientation.

See http://docs.ros.org/api/sensor_msgs/html/msg/Imu.html for the related ROS message definition.

Field	Data Type	Description
P1 Time	Timestamp (8B)	The time of the solution, expressed in the P1 timebase.
Orientation	f64[4]	The platform body orientation with respect to the local ENU frame, represented as a quaternion with the scale component last (x, y, z, w).
Orientation Covariance	f64[9]	<p>The orientation covariance in row-major format about the X, Y, Z axes.</p> <p>Per the ROS IMU message specification:</p> <ul style="list-style-type: none">- If the a value is known but its covariance is not, its covariance matrix will be set to 0.0- If a value is not known or not available, its covariance matrix will be set to -1.0- The value itself will be set to `NAN`, as this is not specified in the ROS message definition <p>Note that the ROS IMU message does not use `NAN` in the covariance matrix to represent either data or covariance not known.</p>
X Rotation Rate	f64	The vehicle X axis gyroscope rate of rotation in radians/second.



Y Rotation Rate	f64	The vehicle Y axis gyroscope rate of rotation in radians/second.
Z Rotation Rate	f64	The vehicle Z axis gyroscope rate of rotation in radians/second.
Angular Velocity Covariance	f64[9]	The vehicle X/Y/Z gyroscope rate of rotation covariance matrix.
X Acceleration	f64	The vehicle X acceleration in meters/second ² .
Y Acceleration	f64	The vehicle Y acceleration in meters/second ² .
Z Acceleration	f64	The vehicle Z acceleration in meters/second ² .
Acceleration Covariance	f64[9]	The vehicle X/Y/Z acceleration covariance matrix.



3.5 Sensor Measurement Input Messages

FusionEngine supports integration of several types of vehicle information to aid improve dead reckoning and navigation performance. The user can input these messages to enable delivery of wheel and vehicle gear information.

Wheel **speeds** are defined as rate sensors that have an indication of the velocity of each wheel. Wheel **ticks** are defined as counters that increment a fixed amount per rotation of the wheel; commonly found on anti-lock braking systems.

Messages that provide 2 channel wheel data (ticks or speeds) and gear data will have better DR performance and is recommended for production designs.

For wheeled vehicles, the system supports the following types inputs:

- [3.5.1 WheelTickInput \(11103\)](#) - Wheel encoder tick counts for each wheel, plus optional transmission gear
- [3.5.2 VehicleTickInput \(11104\)](#) - Single wheel encoder tick count representing along-track (forward/backward) speed, plus optional transmission gear
- [3.5.3 WheelSpeedInput \(11105\)](#) - Speed data for each wheel, plus optional transmission gear
- [3.5.4 VehicleSpeedInput \(11106\)](#) - Single along-track (forward/backward) speed, plus optional transmission gear

Only one type of Vehicle Sensor Measurement should be provided to the system at a time. You should not use Hardware IO for ticks or direction if you are using these messages.

For any of these measurements to be used by the system, the corresponding configuration must be set in the 3.1.5 SetConfig (13100) message.



3.5.1 WheelTickInput (11103)

Message ID: 11103

Protocol Version: 2

Message Version: 0

Description: Conveys the wheel encoder tick counts for one or more wheels. The number and type of wheels expected, and the interpretation of the tick count values varies by vehicle. You must configure vehicle dimensions and a description of the sensor configuration (number of wheels, etc.). See [3.1.5 SetConfig \(13100\)](#).

Field	Data Type	Description
Measurement Time	Timestamp (8B)	The time of applicability of the measurement data. Interpretation dictated by Time Source.
Time Source	SystemTimeSource (u8)	The source of the measurement timestamp. See Section 2.6.2 .
Sensor Data Source	SensorDataSource (u8)	The source of the sensor. See Section 2.6.3
Reserved	u8[2]	<i>Reserved for future use.</i>
Reserved	u8[8]	<i>Reserved for future use.</i>
Front Left Tick Count	u32	The front left wheel tick count.
Front Right Tick Count	u32	The front right wheel tick count.
Rear Left Tick Count	u32	The rear left wheel tick count.
Rear Right Tick Count	u32	The rear right wheel tick count.
Gear	GearType (u8)	The transmission gear currently in use (0 if unavailable).
Reserved	u8[3]	<i>Reserved for future use.</i>



3.5.4 VehicleTickInput(11104)

Message ID: 11104

Protocol Version: 2

Message Version: 0

Description: A single wheel encoder tick count that corresponds to forward vehicle movement. You must configure a description of the sensor configuration (tick rollover value, etc.). See [3.1.5 SetConfig \(13100\)](#).

Field	Data Type	Description
Measurement Time	Timestamp (8B)	The time of applicability of the measurement data. Interpretation dictated by Time Source.
Time Source	SystemTimeSource (u8)	The source of the measurement timestamp. See Section 2.6.2 .
Sensor Data Source	SensorDataSource (u8)	The source of the sensor. See Section 2.6.3
Reserved	u8[2]	<i>Reserved for future use.</i>
Reserved	u8[8]	<i>Reserved.</i>
Tick Count	u32	The current encoder tick count.
Gear	GearType (u8)	The transmission gear currently in use (0 if unavailable).
Reserved	u8[3]	<i>Reserved for future use.</i>



3.5.2 WheelSpeedInput (11105)

Message ID: 11105

Protocol Version: 2

Message Version: 0

Description: Used to convey the speed of each individual wheel on the vehicle. The number and type of wheels expected varies by vehicle. You must configure a description of the sensor configuration in [3.1.5 SetConfig \(13100\)](#).

Field	Data Type	Description
Measurement Time	Timestamp (8B)	The time of applicability of the measurement data. Interpretation dictated by Time Source.
Time Source	SystemTimeSource (u8)	The source of the measurement timestamp. See Section 2.6.2 .
Sensor Data Source	SensorDataSource (u8)	The source of the sensor. See Section 2.6.3
Reserved	u8[2]	<i>Reserved for future use.</i>
Reserved	u8[8]	<i>Reserved.</i>
Front Left Speed	f32	The front left wheel speed (in m/s). Set to NaN if unavailable.
Front Right Speed	f32	The front right wheel speed (in m/s). Set to NaN if unavailable.
Rear Left Speed	f32	The rear left wheel speed (in m/s). Set to NaN if unavailable.
Rear Right Speed	f32	The rear right wheel speed (in m/s). Set to NaN if unavailable.
Gear	GearType (u8)	The transmission gear currently in use (0 if unavailable).
Is Signed	bool	True if the wheel speed data is signed (positive forward, negative reverse), or false otherwise (positive in both directions).
Reserved	u8[2]	<i>Reserved for future use.</i>



3.5.3 VehicleSpeedInput (11106)

Message ID: 11106

Protocol Version: 2

Message Version: 0

Description: A single number (usually an average of the non-steered wheels) that corresponds to forward vehicle velocity. Note that only using a single wheel as “vehicle speed” can result in suboptimal performance while the vehicle is turning.

You must configure a description of the sensor configuration (tick rollover value, etc.). See [3.1.5 SetConfig \(13100\)](#).

Field	Data Type	Description
Measurement Time	Timestamp (8B)	The time of applicability of the measurement data. Interpretation dictated by Time Source.
Time Source	SystemTimeSource (u8)	The source of the measurement timestamp. See Section 2.6.2 .
Sensor Data Source	SensorDataSource (u8)	The source of the sensor. See Section 2.6.3
Reserved	u8[2]	<i>Reserved for future use.</i>
Reserved	u8[8]	<i>Reserved.</i>
Vehicle Speed	i32	The vehicle speed estimate (in m/s * 2 ⁻¹⁰). Set to 0x7FFFFFFF if unavailable.
Gear	GearType (u8)	The transmission gear currently in use (0 if unavailable).
Flags	u8	Bitmask of flags associated with measurement data: Bit 0: 1 if values signed (positive forward) <i>Bit 1-7: Reserved</i>
Reserved	u8[2]	<i>Reserved for future use.</i>



3.6 Raw Sensor Measurement Output Messages

Raw outputs are available for offboard processing. These messages are *not* compensated for any offset or calibration. The only additional processing is the calculation of the P1 Time which is added to the messages.

3.6.1 RawHeadingOutput (11001)

Message ID: 11001

Protocol Version: 2

Message Version: 0

Description: The raw heading angle (in degrees) with respect to true north as measured by a secondary heading measurement device, pointing from the primary antenna to the secondary antenna. This is *not* corrected for any biases.

Field	Data Type	Description
Measurement Time	Timestamp (8B)	The time of applicability of the measurement data. Interpretation dictated by Time Source.
Time Source	SystemTimeSource (u8)	The source of the measurement timestamp. See Section 2.6.2 .
Reserved	u8[3]	<i>Reserved for future use.</i>
P1 Time	Timestamp (8B)	The calculated P1 time corresponding with the measurement timestamp.
Solution Type	SolutionType (u8)	Invalid: Heading not available RTKFixed: Heading available See section 3.3.1 Pose (10000).
Reserved	u8[3]	<i>Reserved for future use.</i>
Reserved	u32	<i>Reserved for future use.</i>
Relative Position (East)	f32	The East component of vector from the primary antenna to the secondary antenna in meters.
Relative Position (North)	f32	The North component of vector from the primary antenna to the secondary antenna in meters.
Relative Position (Up)	f32	The Up component of vector from the primary antenna to the secondary antenna in meters.
Position Std Dev (East)	f32	The standard deviation of the position in the East direction in meters.



Position Std Dev (North)	f32	The standard deviation of the position in the North direction in meters.
Position Std Dev (Up)	f32	The standard deviation of the position in the Up direction in meters.
Heading	f32	The measured heading angle (in degrees), with respect to true North, pointing from the primary antenna to the secondary antenna. Set to NaN if unavailable.
Baseline Distance	f32	The distance between the primary and secondary antennas (in m). Set to NaN if unavailable.

3.6.2 RawIMUOutput (11002)

Message ID: 11002

Protocol Version: 2

Message Version: 0

Description: Raw IMU measurement data.

This message is an output from the device containing raw IMU acceleration and rotation rate measurements. These measurements come directly from the sensor and do not have any corrections or calibration applied.

Field	Data Type	Description
Measurement Time	Timestamp (8B)	The time of applicability of the measurement data. Interpretation dictated by Time Source.
Time Source	SystemTimeSource (u8)	The source of the measurement timestamp. See Section 2.6.2 .
Sensor Data Source	SensorDataSource (u8)	The source of the sensor. See Section 2.6.3
Reserved	u8[2]	<i>Reserved for future use.</i>
Converted P1 Time	Timestamp (8B)	The calculated P1 Time corresponding with the measurement timestamp.
Reserved	u8[6]	<i>Reserved for future use.</i>



Temperature	i16	The IMU temperature (in Celsius * 2^{-7}). Set to 0x7FFF if invalid
X Acceleration	i32	Measured x acceleration (in meters/second ² * 2^{-16}), resolved in the sensor measurement frame. Set to 0x7FFFFFFF if invalid
Y Acceleration	i32	Measured y acceleration (in meters/second ² * 2^{-16}), resolved in the sensor measurement frame. Set to 0x7FFFFFFF if invalid
Z Acceleration	i32	Measured z acceleration (in meters/second ² * 2^{-16}), resolved in the sensor measurement frame. Set to 0x7FFFFFFF if invalid
X Rotation Rate	i32	Measured x rate of rotation (in radians/second * 2^{-20}), resolved in the sensor measurement frame. Set to 0x7FFFFFFF if invalid.
Y Rotation Rate	i32	Measured y rate of rotation (in radians/second * 2^{-20}), resolved in the sensor measurement frame. Set to 0x7FFFFFFF if invalid.
Z Rotation Rate	i32	Measured z rate of rotation (in radians/second * 2^{-20}), resolved in the sensor measurement frame. Set to 0x7FFFFFFF if invalid.



3.6.3 RawWheelTickOutput (11123)

Message ID: 11123

Protocol Version: 2

Message Version: 0

Description: This message is an output from the device that contains wheel encoder tick counts for each individual wheel on the vehicle. These measurements come directly from the sensor, and do not have any corrections or calibration applied.

Field	Data Type	Description
Measurement Time	Timestamp (8B)	The time of applicability of the measurement data. Interpretation dictated by Time Source.
Time Source	SystemTimeSource (u8)	The source of the measurement timestamp. See Section 2.6.2 .
Sensor Data Source	SensorDataSource (u8)	The source of the sensor. See Section 2.6.3
Reserved	u8[2]	<i>Reserved for future use.</i>
Converted P1 Time	Timestamp (8B)	The calculated P1 Time corresponding with the measurement timestamp.
Front Left Tick Count	u32	The front left wheel tick count.
Front Right Tick Count	u32	The front right wheel tick count.
Rear Left Tick Count	u32	The rear left wheel tick count.
Rear Right Tick Count	u32	The rear right wheel tick count.
Gear	GearType (u8)	The transmission gear currently in use (0 if unavailable).
Reserved	u8[3]	<i>Reserved for future use.</i>



3.6.4 RawVehicleTickOutput(11124)

Message ID: 11124

Protocol Version: 2

Message Version: 0

Description: This message is an output from the device that contains a wheel encoder tick count representing the along-track speed of the vehicle (forward/backward). This value comes directly from the sensor, and does not have any corrections or calibration applied.

Field	Data Type	Description
Measurement Time	Timestamp (8B)	The time of applicability of the measurement data. Interpretation dictated by Time Source.
Time Source	SystemTimeSource (u8)	The source of the measurement timestamp. See Section 2.6.2 .
Sensor Data Source	SensorDataSource (u8)	The source of the sensor. See Section 2.6.3
Reserved	u8[2]	<i>Reserved for future use.</i>
Converted P1 Time	Timestamp (8B)	The calculated P1 Time corresponding with the measurement timestamp.
Tick Count	u32	The current encoder tick count.
Gear	GearType (u8)	The transmission gear currently in use (0 if unavailable).
Reserved	u8[3]	<i>Reserved for future use.</i>



3.6.5 RawWheelSpeedOutput (11125)

Message ID: 11125

Protocol Version: 2

Message Version: 0

Description: This message is an output from the device that contains the speed of each individual wheel on the vehicle. These measurements come directly from the sensor, and do not have any corrections or calibration applied.

Field	Data Type	Description
Measurement Time	Timestamp (8B)	The time of applicability of the measurement data. Interpretation dictated by Time Source.
Time Source	SystemTimeSource (u8)	The source of the measurement timestamp. See Section 2.6.2 .
Sensor Data Source	SensorDataSource (u8)	The source of the sensor data. See Section 2.6.3
Reserved	u8[2]	<i>Reserved for future use.</i>
P1 Time	Timestamp (8B)	P1 Time corresponding with the measurement.
Front Left Speed	i32	Front left wheel speed (in $\text{m/s} * 2e-10$). Set to 0x7fffffff if unavailable.
Front Right Speed	i32	Front right wheel speed (in $\text{m/s} * 2e-10$). Set to 0x7fffffff if unavailable.
Rear Left Speed	i32	Rear left wheel speed (in $\text{m/s} * 2e-10$). Set to 0x7fffffff if unavailable.
Rear Right Speed	i32	Rear right wheel speed (in $\text{m/s} * 2e-10$). Set to 0x7fffffff if unavailable.
Gear	GearType (u8)	The transmission gear currently in use (0 if unavailable).
Flags	u8	Bitmask of flags associated with measurement data: Bit 0: 1 if values signed (positive forward) <i>Bit 1-7: Reserved</i>
Reserved	u8[2]	<i>Reserved for future use.</i>



3.6.6 RawVehicleSpeedOutput (11126)

Message ID: 11126

Protocol Version: 2

Message Version: 0

Description: This message is an output from the device that contains the along-track speed of the vehicle (forward/backward). These measurements come directly from the sensor, and do not have any corrections or calibration applied.

Field	Data Type	Description
Measurement Time	Timestamp (8B)	The time of applicability of the measurement data. Interpretation dictated by Time Source.
Time Source	SystemTimeSource (u8)	The source of the measurement timestamp. See Section 2.6.2 .
Sensor Data Source	SensorDataSource (u8)	The source of the sensor. See Section 2.6.3
Reserved	u8[2]	<i>Reserved for future use.</i>
Converted P1 Time	Timestamp (8B)	The calculated P1 Time corresponding with the measurement timestamp.
Vehicle Speed	f32	The vehicle speed estimate (in m/s). Set to NaN if unavailable.
Gear	GearType (u8)	The transmission gear currently in use (0 if unavailable).
Flags	u8	Bitmask of flags associated with measurement data: Bit 0: 1 if values signed (positive forward) <i>Bit 1-7: Reserved</i>
Reserved	u8[2]	<i>Reserved for future use.</i>



3.6.7 WheelSpeedOutput (11135)

Message ID: 11135

Protocol Version: 2

Message Version: 0

Description: This message is an output from the device that contains the speed of each individual wheel, after applying any estimated corrections (wheel scale factor, sign, etc).

Wheel odometry may be received via a software input from a host machine, a vehicle CAN bus, or a hardware voltage signal (wheel ticks). The Data Source field indicates which type of data source was provided to the system.

Note: When odometry is provided using hardware wheel ticks, the output rate of this message may differ from the wheel tick input rate. For higher accuracy applications FusionEngine may integrate tick counts over longer time intervals to improve performance.

Field	Data Type	Description
P1 Time	Timestamp (8B)	P1 Time corresponding to the sensor measurement.
Sensor Data Source	SensorDataSource (u8)	The source of incoming data, if known. See Section 2.6.3
Gear	GearType (u8)	The transmission gear currently in use (0 if unavailable).
Flags	u8	Bitmask of flags associated with measurement data: Bit 0: 1 if values signed (positive forward) <i>Bit 1-7: Reserved</i>
Reserved	u8	<i>Reserved for future use.</i>
Front Left Speed	f32	The front left wheel speed (in m/s). Set to NaN if unavailable.
Front Right Speed	f32	The front right wheel speed (in m/s). Set to NaN if unavailable.
Rear Left Speed	f32	The rear left wheel speed (in m/s). Set to NaN if unavailable.
Rear Right Speed	f32	The rear right wheel speed (in m/s). Set to NaN if unavailable.



3.6.8 VehicleSpeedOutput (11136)

Message ID: 11136

Protocol Version: 2

Message Version: 0

Description: This message is an output from the device that contains along-track speed of the vehicle (forward or backward) and applying any estimated corrections (scale factor, etc).

Odometry data may be received via software input from a host machine, a vehicle CAN bus, or a hardware voltage signal (encoder ticks). Data Source will indicate the data source providing the odometry measurements.

Note: When odometry is provided using hardware wheel ticks, the output rate of this message may differ from the wheel tick input rate. For higher accuracy applications FusionEngine may integrate tick counts over longer time intervals to improve performance.

Field	Data Type	Description
P1 Time	Timestamp (8B)	P1 Time corresponding to the sensor measurement.
Sensor Data Source	SensorDataSource (u8)	The source of incoming data, if known. See Section 2.6.3
Gear	GearType (u8)	The transmission gear currently in use (0 if unavailable).
Flags	u8	Bitmask of flags associated with measurement data: Bit 0: 1 if values signed (positive forward) <i>Bit 1-7: Reserved</i>
Reserved	u8	<i>Reserved for future use.</i>
Vehicle Speed	f32	Vehicle body velocity (in m/s). Set to NaN if unavailable.



3.7 Device Status Messages

3.7.1 System Status (10500)

Message ID: 10500

Protocol Version: 2

Message Version: 0

Description: This message describes the overall system status of the device running FusionEngine.

Field	Data Type	Description
P1 Time	Timestamp (8B)	The time of the status entries, expressed in the P1 timebase.
GNSS Temperature	i16	The temperature of the GNSS receiver. in Celsius * 2 ⁻⁷ . Set to 0x7FFF if invalid.
Reserved	u8[118]	<i>Reserved for future use.</i>



4 Proprietary NMEA Sentence Definitions

Point One supports a limited set of a proprietary NMEA messages. These messages offer a limited subset of the functionality of the binary protocol.

4.1 NMEA \$P1CALSTATUS

Description: Indicates the status of the internal calibration algorithm.

Sentence Structure

\$P1CALSTATUS,<stage>,<state_verified>,<gyro_percent_complete>,<accel_percent_complete>,<mounting_angle_percent_complete>
S

Field	Description
stage	Represents the current calibration stage <ul style="list-style-type: none">0 - Unknown/not started1 - Initial mounting angle convergence (performance may be slightly degraded)2 - Final mounting angle convergence255 - Done
state_verified	<ul style="list-style-type: none">0 – Calibration is waiting for the navigation state to be verified.1 - The navigation state has been verified after a hot start and calibration can proceed
gyro_percent_complete	Gyroscope correction calibration status percentage
accel_percent_complete	Accelerometer correction calibration status percentage
mounting_angle_percent_complete	IMU mounting angle correction completion percentage

4.2 NMEA \$P1MSG

Description: Generic message to indicate information for the user from the navigation system.

Sentence Structure

\$P1MSG,<system_time_ns>,<severity>,<message>

Field	Description
system_time_ns	Timestamp of the internal MCU. This is locked to P1 Time.
severity	A number indicating the severity of the message. Lower numbers are more severe.
message	A text string indicating the message.



4.3 NMEA \$PQTMGNSS

Description: Quectel specific message to output information regarding GNSS.

Sentence Structure
\$PQTMGNSS, <TBD>

Field	Description
TBD	TBD

4.4 NMEA \$PQTMVER

Description: Quectel specific message to indicate the version of the software.

Sentence Structure
\$PQTMVER, <TBD>

Field	Description
TBD	TBD

4.5 NMEA \$PQTMVERNO

Description: Quectel specific message to indicate the version of the software.

Sentence Structure
\$PQTMVERNO, <TBD>

Field	Description
TBD	TBD