ARDUPILOT GPS ELLIPSOIDAL ALTITUDE ISSUE
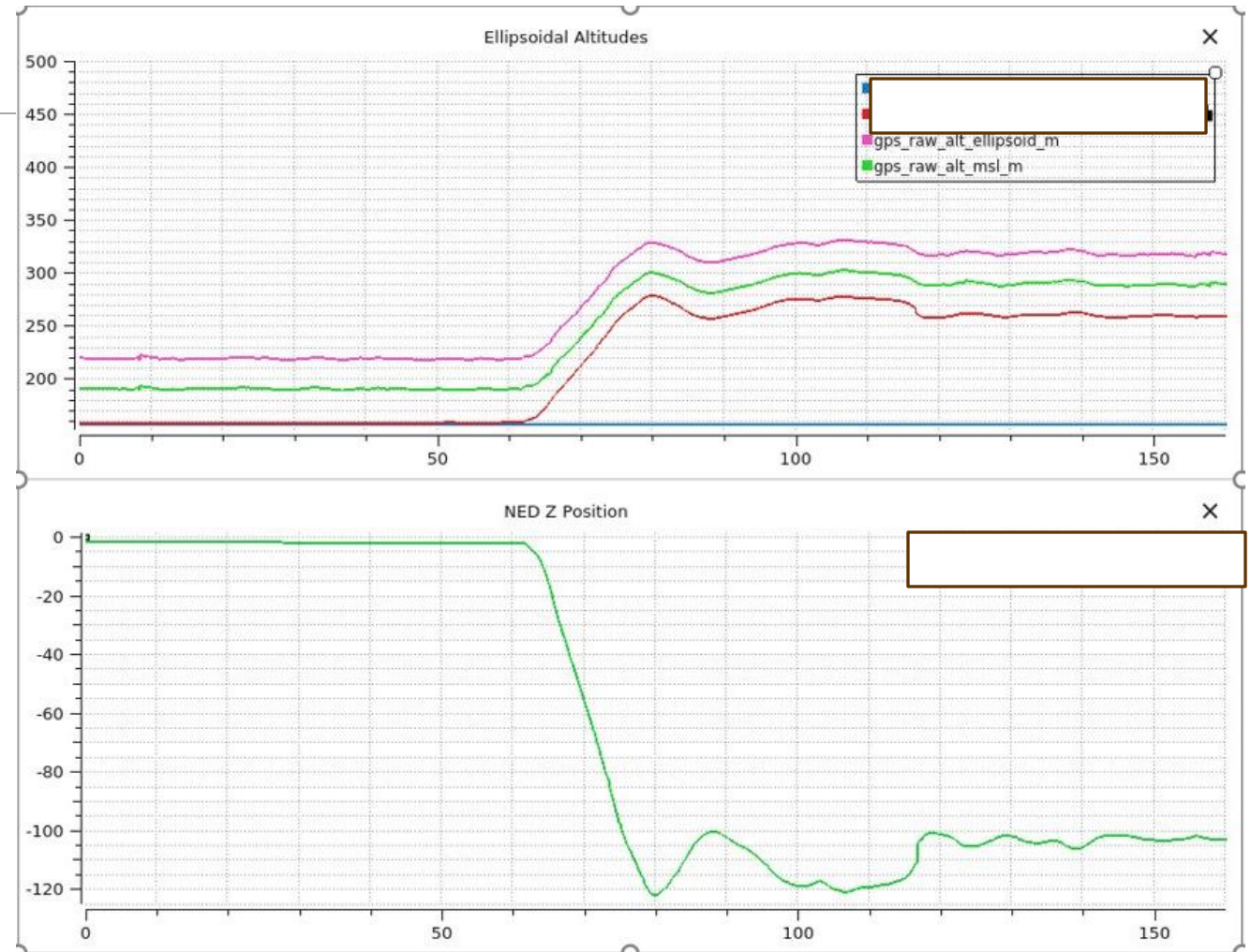
1/28/2025

Error in GPS Ellipsoidal Altitude (1/23/2025)

Plotted ellipsoidal altitudes showed a 57 meter discrepancy between /mavros/gpsstatus/gps1/raw message's alt_ellipsoid field (gps_raw_alt_ellipsoid_m) and ellipsoidal values calculated from MSL using GeographicLib (blue, red).

Incorrect Conversion from MSL to Ellipsoidal Alt

AP_GPS is a class in ArduPilot that facilitates sending messages over mavlink.

The get_undulation function retrieves the undulation and the result is converted to an ellipsoidal altitude.

The correct form of the equation is (see geoid figure):

ellipsoidal_m = msl_m + undulation_m

```
1393    void AP_GPS::send_mavlink_gps_raw(mavlink_channel_t chan)
1394    {
1395        const Location &loc = location(0);
1396        float hacc = 0.0f;
1397        float vacc = 0.0f;
1398        float sacc = 0.0f;
1399        float undulation = 0.0;
1400        int32_t height_elipsoid_mm = 0;
1401        if (get_undulation(0, undulation)) {
1402            height_elipsoid_mm = loc.alt*10 - undulation*1000;
1403        }
```
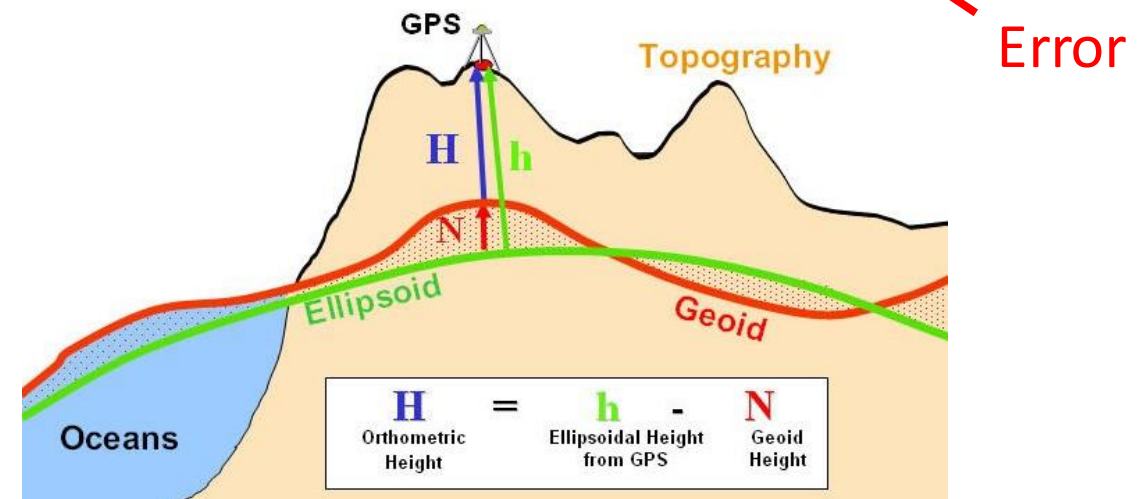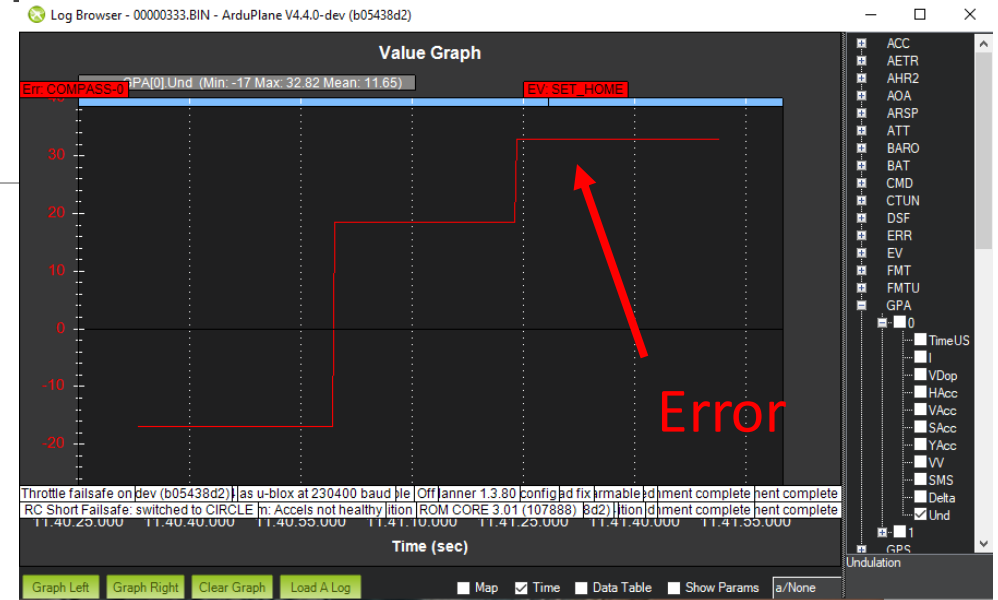
Error

Undulation – Sign Error from u-blox

Geoid calculator specifies an undulation of -34.7m at the test location. The u-blox GPS produced the wrong sign in the figure at right.

We assert that this hasn't been seen before because the two sign errors cancel out (MSL->Ellipsoidal conversion and u-blox undulation).

◦ undulation = ellipsoidal - msl


```
1326        state.location.lat    = _buffer.posllh.latitude;
1327        if (option_set(AP_GPS::HeightEllipsoid)) {
1328            state.location.alt    = _buffer.posllh.altitude_ellipsoid / 10;
1329        } else {
1330            state.location.alt    = _buffer.posllh.altitude_msl / 10;
1331        }
1332        state.have_undulation = true;
1333        state.undulation = (_buffer.posllh.altitude_msl - _buffer.posllh.altitude_ellipsoid) * 0.001;
1334
```

Errors

Ellps Height (m)	Geoid Height (m)	Ortho Height (m)
0	-34.73	34.73

External Tools

You can check ArduPilot terrain data for MSL heights at many locations

For example, Geographiclib reports

◦ EGM96 Altitude of 188

◦ Undulation of -28

◦ WGS-84 Altitude of 159

This is confirmed by the online tool

◦ Ellipsoid = MSL + Undulation

◦ 159.47 = 188 + (-28.53)

And by GeoidEval, a Linux CLI tool

Ellps Height (m)	Geoid Height (m)	Ortho Height (m
159.47	-28.53	18

Parking Lot GPS Comparison (1/28/2025)

- Additional testing in the parking lot on 1/28/2025 with a vehicle shows that the alt_ellipsoid field is incorrect for one GPS and correct for the u-blox.

- At this latitude and longitude, the undulation is -29.28 m, so the alt_ellipsoid field should be **lower** than the alt field.

- As shown previously, one driver is reporting the correctly-signed undulation internally, but the AP_GPS class has a sign error that results in an incorrect alt_ellipsoid. The u-blox driver reports the undulation with an incorrect sign and the AP_GPS sign error inverts it, resulting in a correct ordering between the u-blox alt and alt_ellipsoid.


```
header:
  seq: 6714
  stamp:
    secs: 1707344245
    nsecs: 801526912
  frame id: "/wgs84"

alt: 202210
eph: 66
epv: 111
vel: 1
cog: 11251
satellites visible: 22
alt_ellipsoid: 231410
h_acc: 1840
v_acc: 3329
vel_acc: 0
hdg_acc: 0
yaw: 0
dgps_numch: 255
dgps_age: 4294967295
---
```

```
header:
  seq: 3519
  stamp:
    secs: 1707344373
    nsecs: 221649024
  frame_id: "/wgs84"

alt: 218800
eph: 83
epv: 141
vel: 5
cog: 35769
satellites visible: 14
alt_ellipsoid: 187941
h_acc: 1246
v_acc: 2319
vel_acc: 438
hdg_acc: 0
yaw: 0
dgps_numch: 255
dgps_age: 4294967295
---
```

u-blox

Ellps Height (m)	Geoid Height (m)	Ortho Height (m)
0	-29.28	29.28

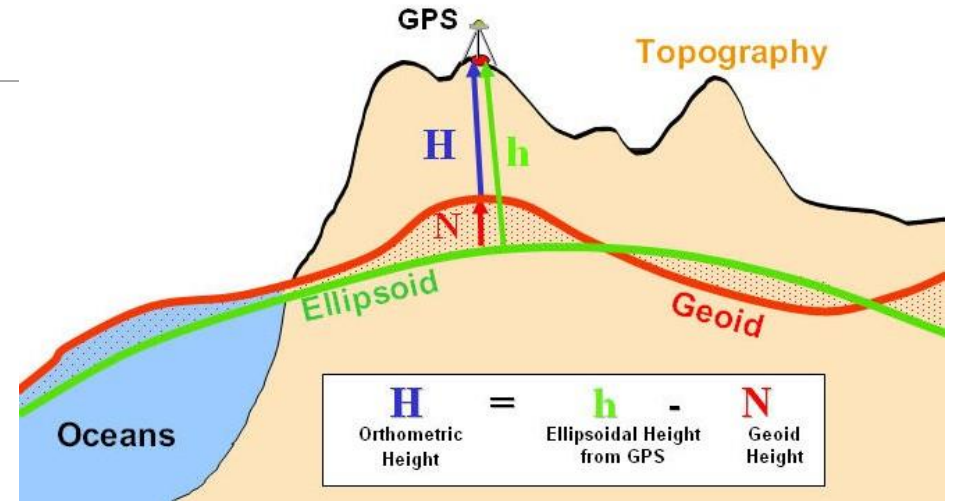# ArduPilot Undulation – Which Convention?

```
1393    void AP_GPS::send_mavlink_gps_raw(mavlink_channel_t chan)
1394    {
1395        const Location &loc = location(0);
1396        float hacc = 0.0f;
1397        float vacc = 0.0f;
1398        float sacc = 0.0f;
1399        float undulation = 0.0;
1400        int32_t height_elipsoid_mm = 0;
1401        if (get_undulation(0, undulation)) {
1402            height_elipsoid_mm = loc.alt*10 - undulation*1000;
1403        }
```

The code in words:
ellipsoid = msl - undulation

Example:
* Orthometric (AMSL) height = 100
* Ellipsoidal height = 80
* Undulation = 20 (positive)

msl = ellipsoidal - undulation
undulation = ellipsoidal - msl

Plugging in:
100 ≠ 80 - 20

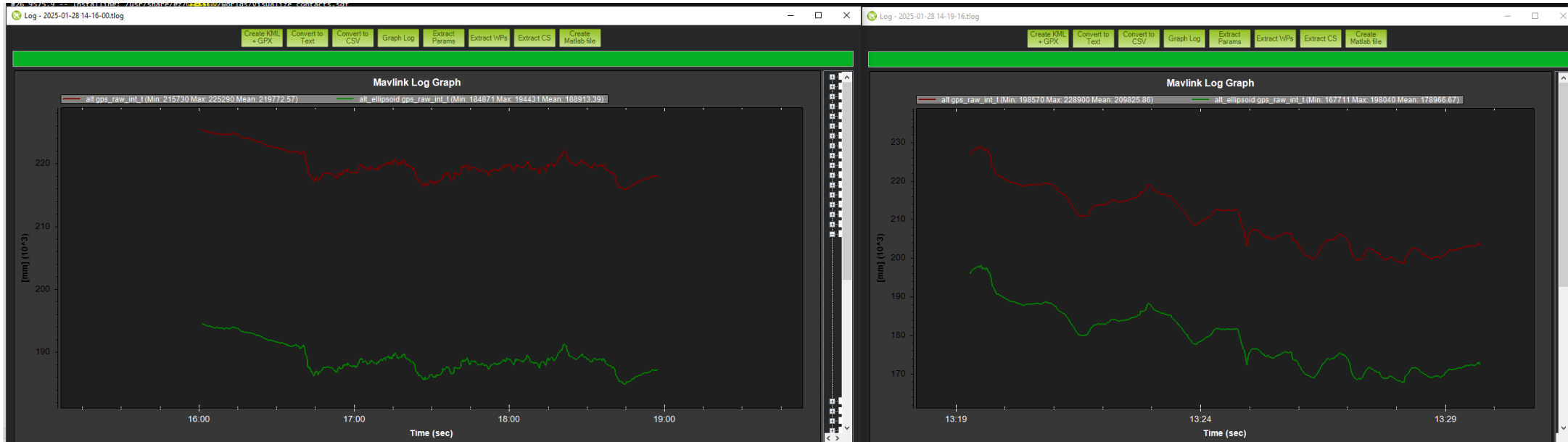ArduPilot's internal idea of undulation is backwards common convention.
This is hidden largely from MAVLink users.
GPS MSL may be broken on for drivers using undulation, but no one noticed because it wasn't probably tested
Ublox users are not impacted unless you run post-analysis on dataflash logs.

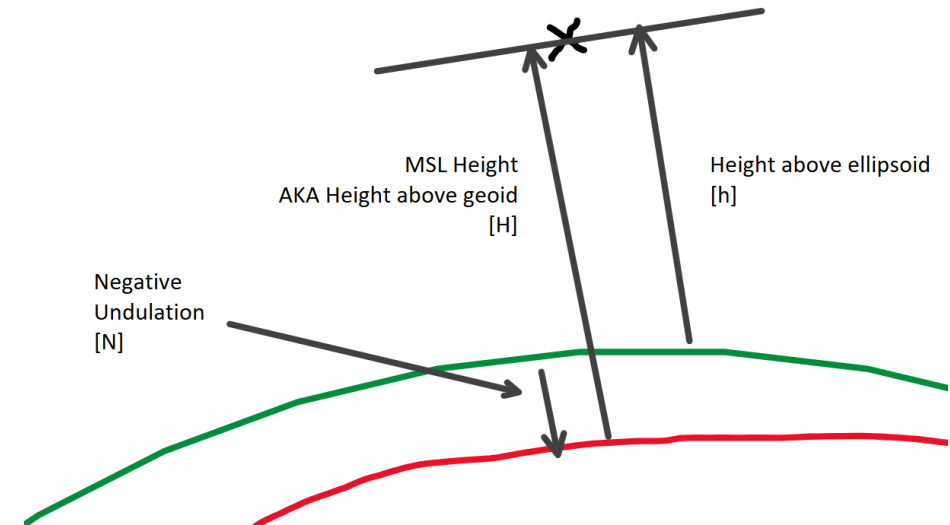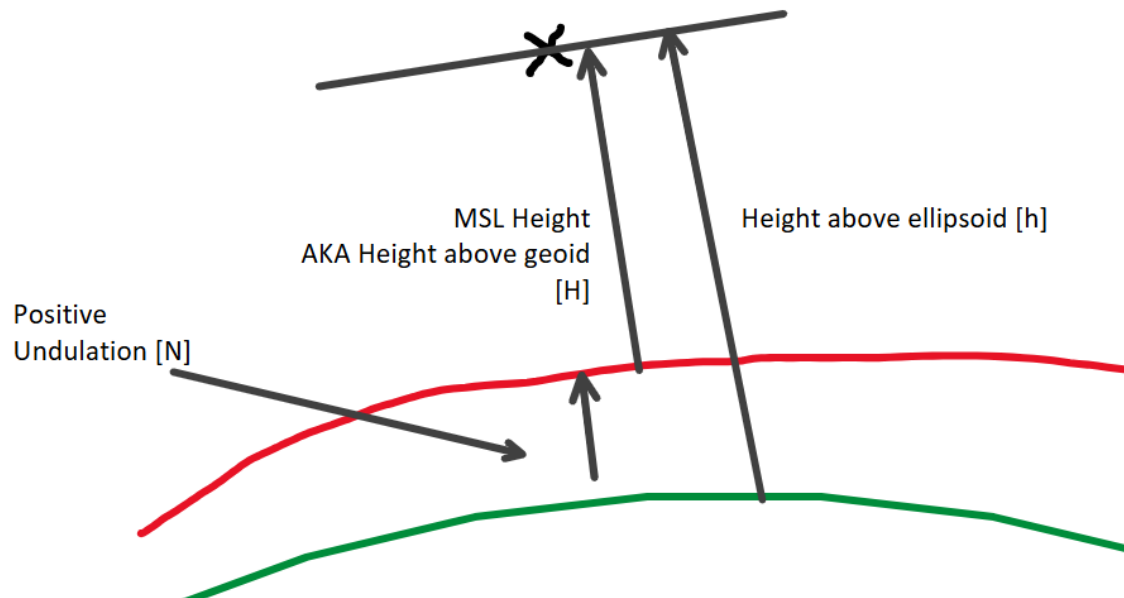# Tlog ground test – MAVLink Data

ublox

With the patch, The MAVLink TLOG shows two GPS's reporting greater MSL height than ellipsoid height, which matches expectations for the test.
The plot shows GPS_RAW_INT.alt (red) vs GPS_RAW_INT.alt_ellipsoid(green)

Correct Conventions (Wikipedia)

Positive Undulation
H<h, N>0

Negative Undulation
H>h, N<0

MSL Height
AKA Height above geoid
[H]

Height above ellipsoid [h]

Positive
Undulation [N]

MSL Height
AKA Height above geoid
[H]

Height above ellipsoid
[h]

Negative
Undulation
[N]

The US generally has a negative undulation

Update: Patched ArduPilot

The ellipsoidal altitude calculation was corrected with the fix to AP_GPS.

The u-blox driver was incorrect in two places in addition to the shared AP_GPS error. All three errors have been fixed.

Both GPS messages shows the correct relationship between alt and alt_ellipsoid.

```
header:
  seq: 35859
  stamp:
    secs: 1707344829
    nsecs: 745898304
  frame_id: "/wqs84"


alt: 211940
eph: 39
epv: 60
vel: 0
cog: 26190
satellites_visible: 37
alt_ellipsoid: 182740
h_acc: 1244
v_acc: 2104
vel_acc: 0
hdg_acc: 0
yaw: 0
dgps_numch: 255
dgps_age: 4294967295
---
```

```
header:
  seq: 10567
  stamp:
    secs: 1707344323
    nsecs:   29066304
  frame_id: "/wqs84"


alt: 211900
eph: 40
epv: 58
vel: 1
cog: 30509
satellites_visible: 36
alt_ellipsoid: 182700
h_acc: 1236
v_acc: 2000
vel_acc: 0
hdg_acc: 0
yaw: 0
dgps_numch: 255
dgps_age: 4294967295
```

u-blox

[u-blox Protocol](#) Documentation

o u-blox provides ellipsoid and MSL in multiple messages

UBX-NAV-PVT continued

Byte Offset	Number Format	Scaling	Name	Unit	Description
24	I4	1e-7	lon	deg	Longitude
28	I4	1e-7	lat	deg	Latitude
32	I4	-	height	mm	Height above ellipsoid
36	I4	-	hMSL	mm	Height above mean sea level

5.14.10 UBX-NAV-POSLLH (0x01 0x02)

5.14.10.1 Geodetic Position Solution

Message	UBX-NAV-POSLLH						
Description	**Geodetic Position Solution**						
Firmware	Supported on: • u-blox 9 with protocol version 27.11						
Type	Periodic/Polled						
Comment	See important comments concerning validity of position given in section Navigation output filters in Integration manual. This message outputs the Geodetic position in the currently selected ellipsoid. The default is the WGS84 Ellipsoid, but can be changed with the message CFG-NAVSPG-USE_USRDAT.						
	Header	Class	ID	Length (Bytes)		Payload	Checksum
Message Structure	0xB5 0x62	0x01	0x02	28		see below	CK_A CK_B

Payload Contents:

Byte Offset	Number Format	Scaling	Name	Unit	Description
0	U4	-	iTOW	ms	GPS time of week of the navigation epoch. See the section iTOW timestamps in Integration manual for details.
4	I4	1e-7	lon	deg	Longitude
8	I4	1e-7	lat	deg	Latitude
12	I4	-	height	mm	Height above ellipsoid

u-blox ZED-F9P Interface Description - Manual

UBX-NAV-POSLLH continued

Byte Offset	Number Format	Scaling	Name	Unit	Description
16	I4	-	hMSL	mm	Height above mean sea level
20	U4	-	hAcc	mm	Horizontal accuracy estimate
24	U4	-	vAcc	mm	Vertical accuracy estimate

References

https://www.ardusimple.com/ellipsoidal-orthometric-and-geoid-height/

https://www.ngs.noaa.gov/PUBS_LIB/gislis96.html#:~:text=It%20is%20a%20straightforward%20procedure,H%20%3D%20h%20%2D%20N%20.

https://www.unavco.org/software/geodetic-utilities/geoid-height-calculator/geoid-height-calculator.html

https://github.com/ArduPilot/ardupilot/pull/21075#pullrequestreview-1065920977

https://github.com/ArduPilot/ardupilot/pull/23942