NAVIPAC

TIMING PRINCIPLES

Author: Ole Kristensen
Last update: 26/06/2014
Version: 3.9

Contents

1 Fundamentals

NaviPac has a dedicated background process (DataIONT) to handle all data collection and time tagging. The process is kept as simple as possible, as it runs on the Windows OS real-time queue. All other processes are handled by the ordinary process queue – and thus given a lower priority.

For each input port in use, the process creates a thread (ie a process) to handle the communication.

Each thread reads data from the port, collects it into packets and stores data and time in a dedicated buffer area.

The process controls the navigation cycle, as it with user-determined frequency:

- Tells all threads to use new buffer area
- Stores header information in the buffer
- Tells the kernel to start processing using the buffer area
- Waits until next cycle

This gives a minimal load of the real-time process and ensures an optimal time tagging of the incoming data.

2 Simple timing

To secure the best timing, the thread should interrupt on the first character in an incoming telegram, register time and read the entire telegram.

This gives a high load of the system, and with many high-frequency sensors, the load would kill the OS system, as Windows does not give any efficient handles to that.

Instead, NaviPac has a smarter and much more efficient way to handle interrupts.

NaviPac defines an interrupt on the last character in the telegram (most often <LF>). When <LF> occurs on the port, NaviPac gets an interrupt and registers the clock (timestamp t0). Then, NaviPac reads the entire telegram.

Based on the length of the telegram, NaviPac calculates total transmission time (dT) as length multiplied by time per character:

For example, on a 9600 Baud line with N81, NaviPac estimates 0.0009375 seconds per character. A typical GPS telegram like $GPGGA,132022.00,5408.8167,N,00900.0001,E,2,5,04,112,M,0,M,0.0,1234<CR><LF> with 69 characters will be transmitted in 0.0646875 seconds.

The timestamp of the telegram is now t0 minus dT.

This time can hereafter be corrected by a fixed latency (age) value defined under a property called **Fixed age** or **Latency** in the specific instrument's setup, as can be seen below in Figure 1:

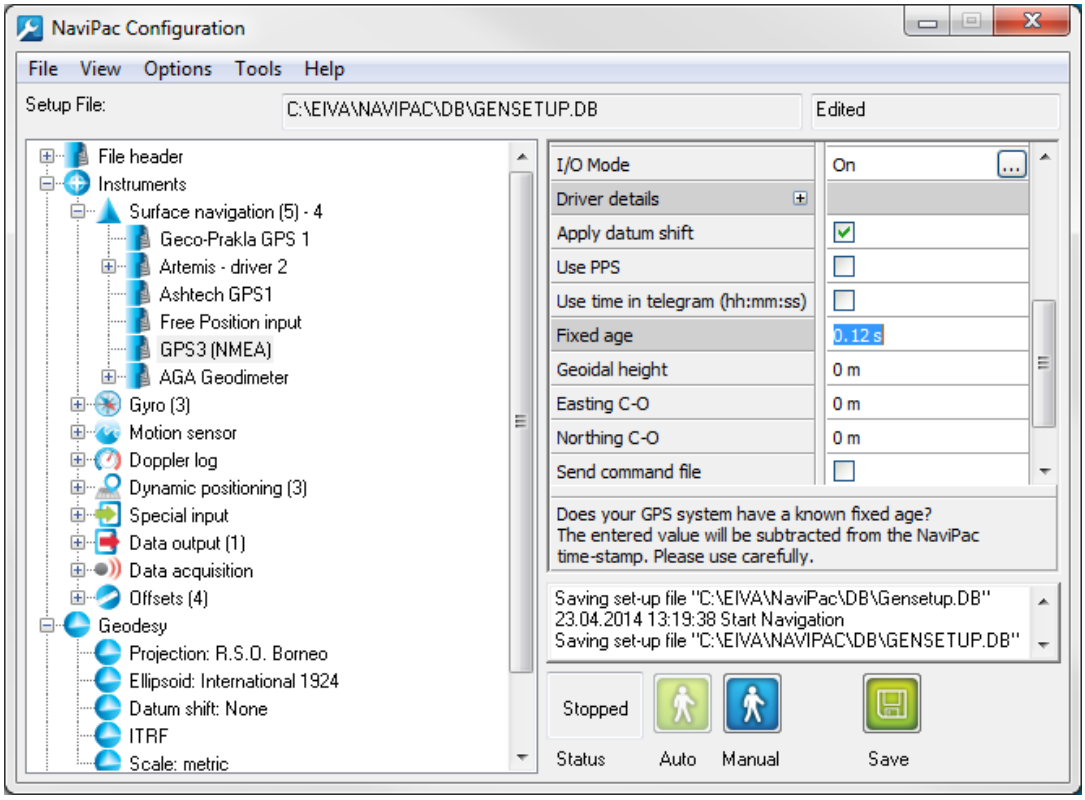

Figure 1 Fixed age property for a GPS instrument

There are some GPS instruments (eg NMEA GPS) that manage to output data in real time – which means that the position will be extrapolated (deskewed) to be valid exactly when the telegram is output. In that case, the fixed latency value defined should be zero or close to zero.

3 Timing supported by external information about latency of data

GPS instruments are based on a series of internally calculated data calculations before the information is sent on the port to NaviPac. NaviPac can calculate this data latency by using a method in which the latency is read from the position telegram.

Ashtech has extended the GGA telegram when operating in RTK, as they use the age field for two things:

- The part before the decimal point is the correction latency
- The part after the decimal point is the position latency

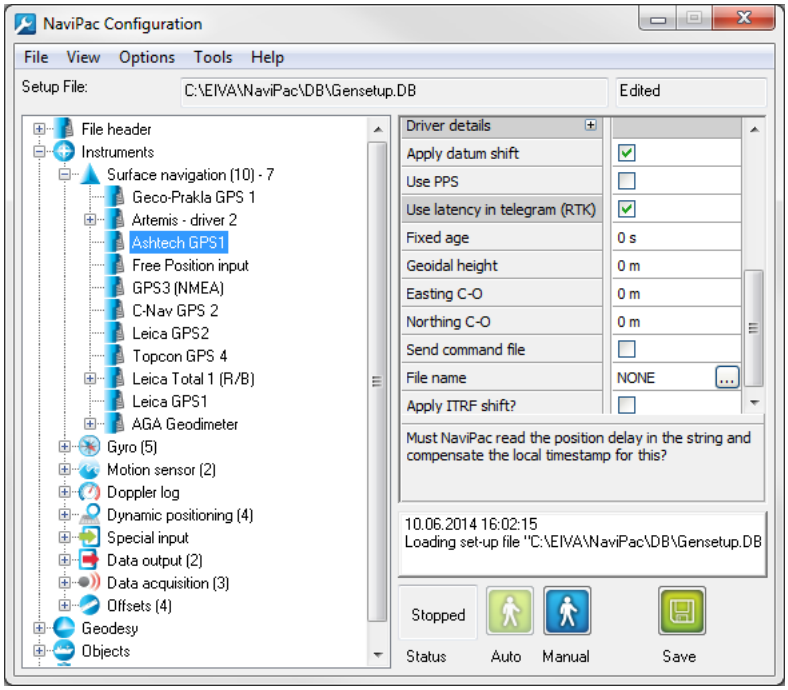

Figure 2 Special handling of Ashtech GPS latency

For example, if the GPS outputs the value ‘1.73 s’, this means that the correction from base is 1 second old and that the position is 0.73 seconds old.

Problem: This will only work for position latency of 0.99 seconds or less.

Note: This is only valid for RTK.

GPS instruments like Geco-Prakla, Sea-diff / Skyfix and Racal UKOOA GPS use a similar approach.

4 Synchronising with external clock

Another way of handling timing is to calculate the latency by setting the PC clock and then reading the time in the positioning telegram from the GPS instrument. This calculation must then be enabled manually for each individual applicable GPS instrument under the **Use time in telegram (hh:mm:ss)** property, as shown below in Figure 3:

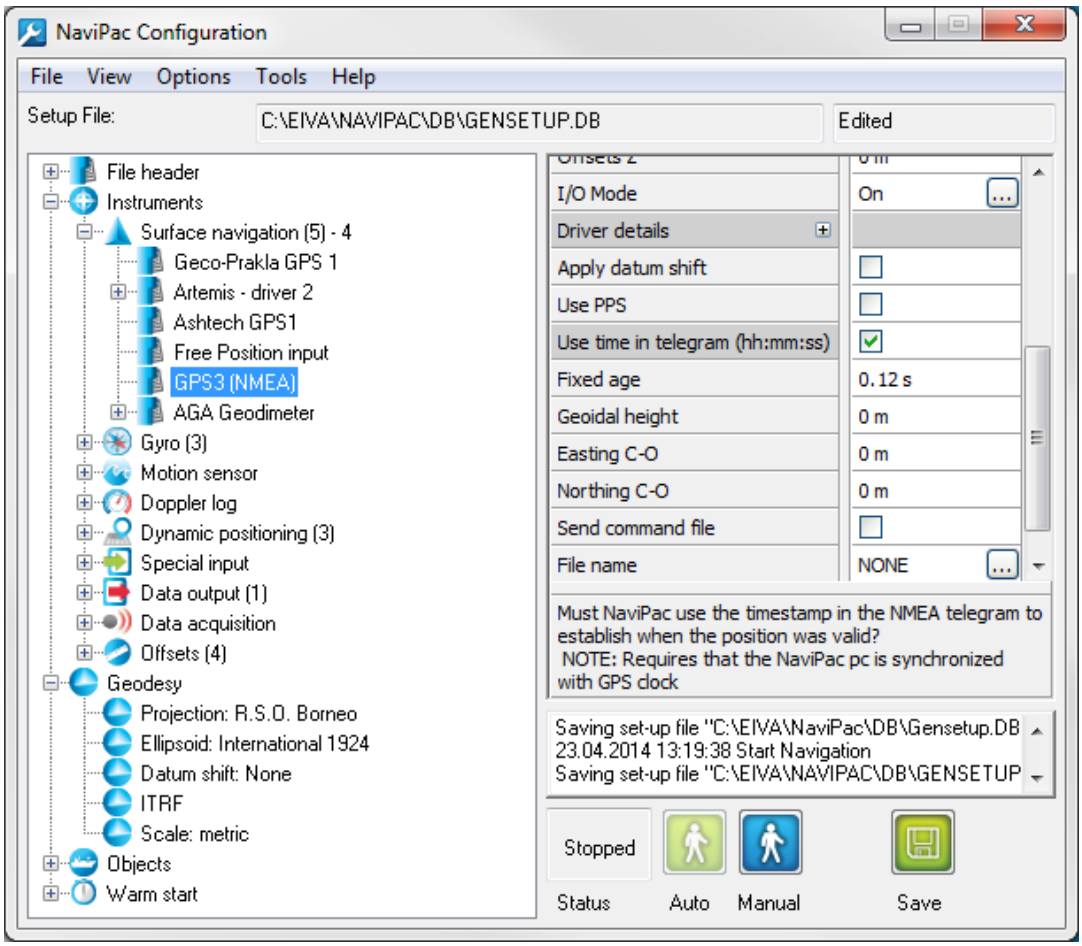

Figure 3 Time selection for GPS supporting use of time synchronisation

To do so, the PC running NaviPac must be synchronised in time with the GPS clock. This can be done either via interfacing into NaviPac, as discussed in the following sections, or via standard network time control (NTP, etc).

NaviPac supports Trimble UTC or NMEA ZDA input telegrams.

Please note that the PPS pulse must be interfaced on a special <RING> or <DSR> indicator – see Section 4.3 'PPS Indicator' for further details.

4.1 Using Trimble UTC telegram

Trimble uses a special UTC telegram 'UTC yy.mm.dd hh:mm:ss <fix> <satellites><CR><LF>'. The telegram is outputted approximately 500 milliseconds before the time is valid.

As the telegram is outputted approximately 500 milliseconds before the time is valid, NaviPac requires that it be combined with PPS to get more accurate timing.

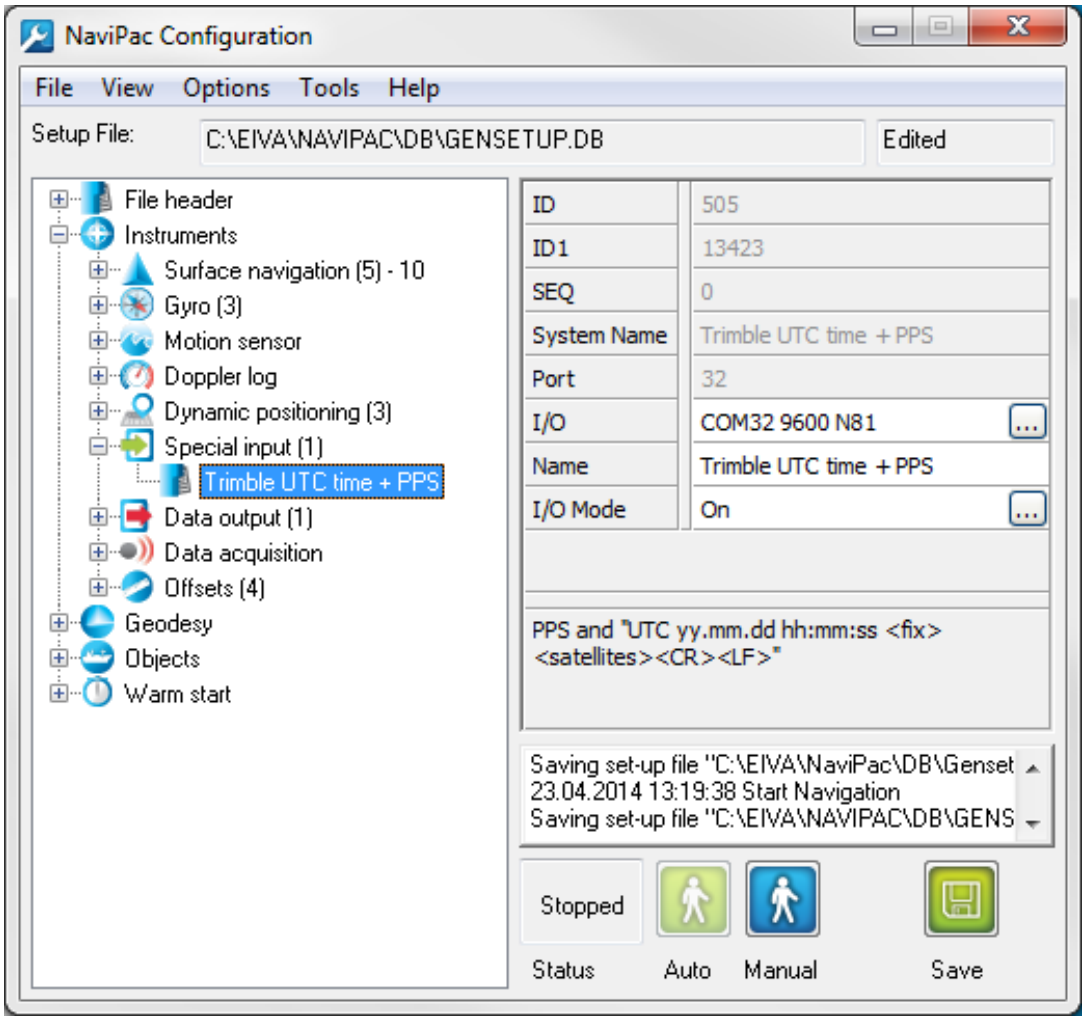

Figure 4 Setting up Trimble UTC time synchronisation

In NaviPac, you select Trimble UTC time + PPS as a special input on a separate dedicated COM port. The PC time is then automatically synchronised with the GPS clock.

You must hereafter manually specify for each individual applicable GPS instrument whether it must use time in telegram.

Trimble UTC telegram implementation:

The driver in NaviPac that uses UTC and PPS telegrams for synchronisation is implemented roughly as follows:

The port drive is set up to interrupt on PPS (ring indicator) and <CR><LF>.

On <CR><LF>:

- Read the telegram from the port.
- If the telegram is the correct UTC telegram, read time in telegram as hh, mm, ss (three two-digit numbers, not fractions).

On PPS:

- Register time of PPS into timestamp.
- If a proper hh, mm, ss exists, as defined above in regards to <CR><LF> (a UTC telegram read prior to the PPS), then hh, mm, ss equals timestamp. Set the PC time to hh, mm, ss (and ms = 00).

Potential errors: If the UTC telegram is outputted much less than 500 milliseconds before its corresponding PPS telegram, the time synchronisation might lose track. By much less, it is meant so much that the two telegrams are almost merged.

4.2 Using NMEA ZDA telegram

A similar driver has been made to use ordinary NMEA ZDA telegrams to synchronise the PC time with the GPS clock. Again, this MUST be combined with PPS to get maximum accuracy. Here, it is assumed that the ZDA telegram is received after the corresponding PPS – but with a small enough delay that it can be received clearly before the next PPS. **If the GPS cannot guarantee that the PPS and the ZDA match, you risk faulty synchronisations.**

In NaviPac, you select ZDA time + PPS as a special input on a separate dedicated COM port. The PC time is then automatically synchronised with the GPS clock.

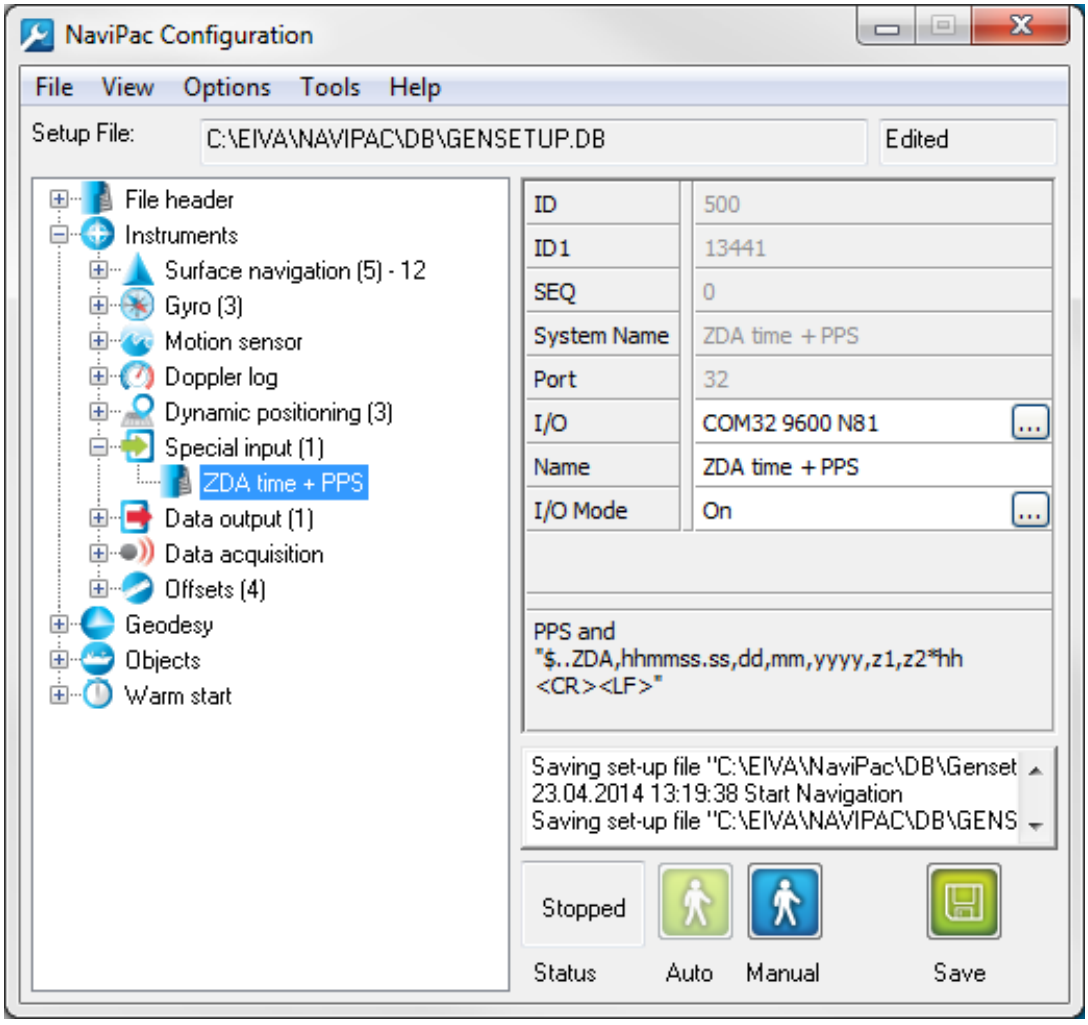

Figure 5 Setting up NMEA ZDA time synchronisation

You must hereafter manually specify for each individual applicable GPS instrument whether it must use time in telegram.

NMEA ZDA telegram implementation:

The driver in NaviPac that uses ZDA and PPS telegrams for synchronisation is implemented roughly as follows:

The port drive is set up to interrupt on PPS (ring indicator) and <CR><LF>.

On PPS:

- Register time of PPS into timestamp.
- If a proper hh, mm, ss exists, as defined below in regards to <CR><LF> (A ZDA telegram read after the PPS), then hh, mm, ss+1 equals timestamp. Set the PC time to hh, mm, ss+1 (and ms = 00).

On <CR><LF>:

- Read the telegram from the port.
- If the telegram is the correct ZDA telegram, read time in telegram as hh, mm, ss (three two-digit numbers, not fractions).

Potential errors: If the ZDA telegram is outputted too long after its corresponding PPS telegram, the time synchronisation probably will lose track and fail, as there are no tools to match PPS and ZDA.

4.3 PPS Indicator

The PPS is, as mentioned above, implemented by assigning the pulse to the ring pin on the physical serial port:

GPS	COM port (DB25)
TxD	3 (RxD)
Gnd	7 (Gnd)
PPS Coax Shield	7 (Gnd)
PPS Coax Centre	22 (RI)

Table 1 DB25 PPS configuration

To test the cable, you can connect the GPS to the COM port with the cable, then start the **NaviPac Configuration** setup program. From here, select **Test Interface Ports** from the bottom of the **Tools** menu and select the port in use. The **NaviPac Rawdata** window should now display the following:

<Ring>

$GPZDA ....

<Ring>

$GPZDA ....

<Ring>

$GPZDA ....

<Ring>

$GPZDA ....

<Ring>

<Ring > can occur more than one time between ZDA telegrams, as a PPS has a certain width (for an Ashtech Z12 instrument, approximately 2 milliseconds), which might result in more interrupts.

In some interfaces, we have observed that the ring indicator is not connected to the PC – and in that case, you may use the DSR signal instead. This can be changed under **Options > Global Parameters > Advanced > PPS Pin**.

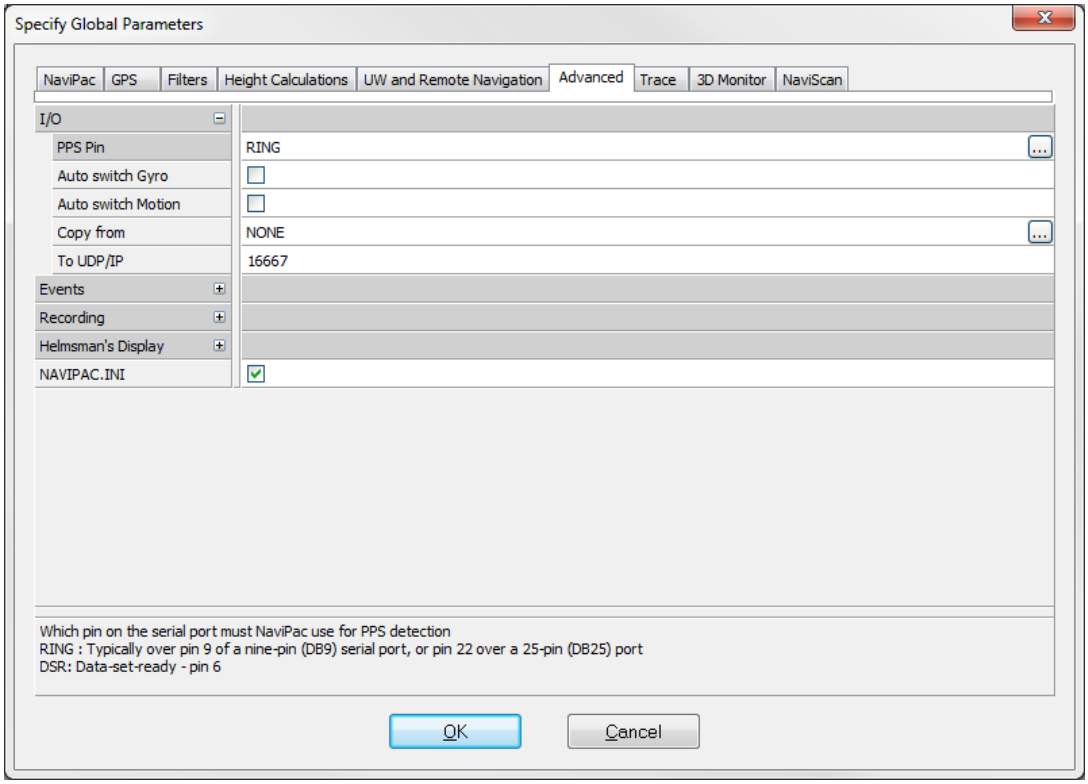

Figure 6 Selection of PPS indicator

5 Assigning PPS to specific position input

If you do not want to use the external time synchronisation, you may instead select to assign the PPS to each incoming GPS data telegram.

NaviPac has a property for each GPS instrument (**Use PPS**) where the user can select whether time tagging must be performed based on incoming data (ordinary) or based on data valid flag (PPS).

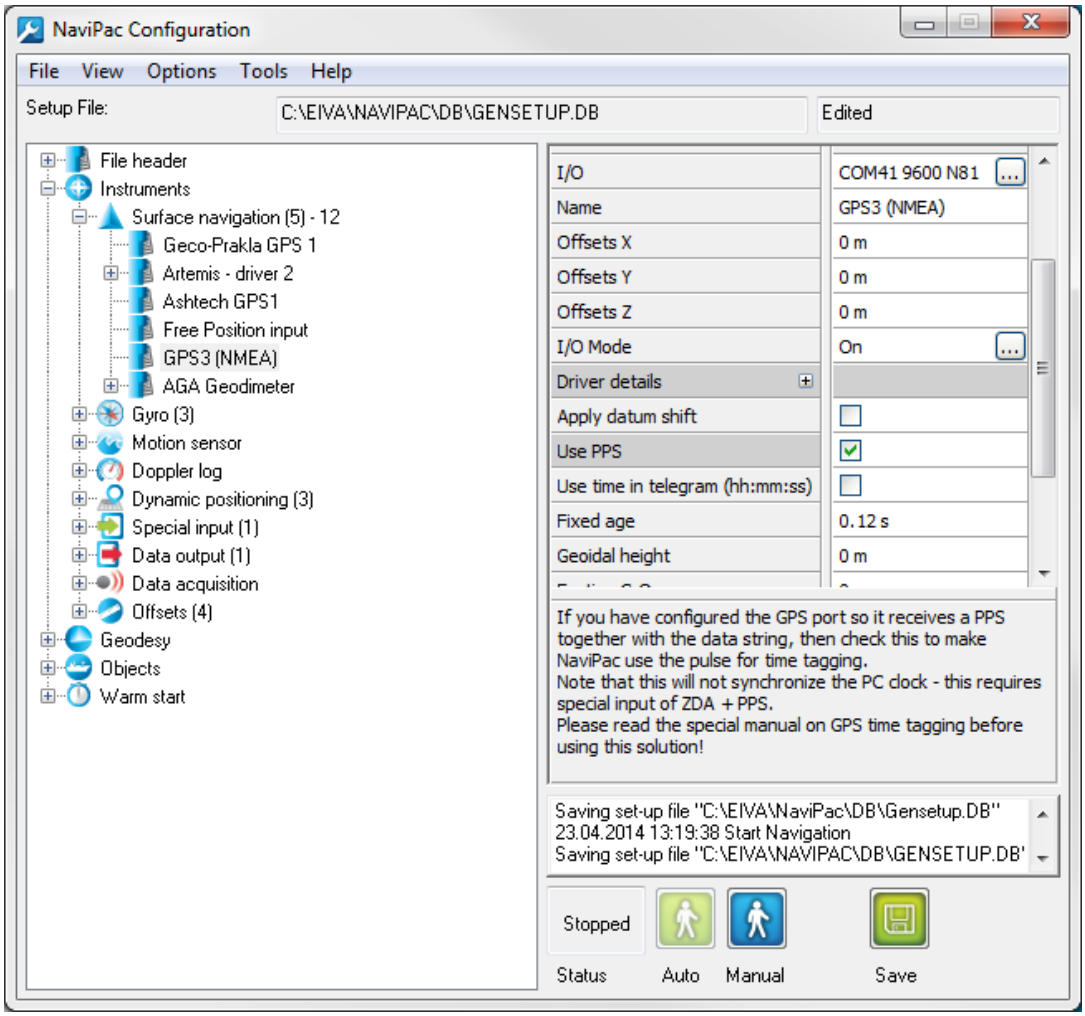

Figure 7 Using GPS PPS directly on each instrument

The cable from the GPS to NaviPac must be designed to handle data and pulse simultaneously (similar to the time telegram + pulse discussed above in Section 4.3 ‘PPS Indicator’).

NaviPac will hereafter timestamp the signal based on the PPS and not based on reception of the first byte, as is normally the case.

This is implemented based on reception of one PPS followed by zero or more GPS telegrams per second. All GPS telegrams received after a PPS will be time tagged with the PPS time.

Problem: If the latency is longer than one second (eg old Leica with more than two seconds), NaviPac cannot use PPS as standalone (see Section 4.2 ‘Using NMEA ZDA telegram’).

NaviPac has a partial solution for handling latencies slightly longer than one second: Let tMin determine an absolute minimum GPS calculation time – that is, the fastest possible output. If NaviPac receives a data set from the GPS and calculates the difference between reception time (T_now - transmit time) and the latest PPS, then the result must be larger than tMin. If not, use the last PPS. In the current version, this is hardcoded to 120 milliseconds.

Note: This will only be functional for a 1 Hz output, as all telegrams following the PPS get the same time tag. The use of PPS must furthermore be enabled manually for each individual GPS instrument under the **Use PPS** property.

6 Recommended method

We are often asked what the best method is. NaviPac offers a variety of methods, and the user must often analyse the performance of each instrument and select a method depending on the type of job and list of equipment. You must also consider whether you want to use absolute time (time-synchronised with the external GPS clock) or not.

The various methods described in the above sections are listed below in order of recommendation based on general level of accuracy, along with some notes as to which situations they are best suited for:

1. If you want to use absolute time, NaviPac should be synchronised with the GPS clock and use the timestamp in GPS telegram (and USBL). See Section 4 ‘Synchronising with external clock’ for further details. This is also the preferred solution when using ATTU for sensor time tagging
2. If you do not require absolute time with millisecond accuracy but still want to use PPS as reference – PPS can be assigned to specific position input. See Section 5 ‘Assigning PPS to specific position input’ for further details.
3. If the GPS telegram includes information about data latency (eg Geco-Prakla GPS), NaviPac can use that information to compensate the timestamp. See Section 3 ‘Timing supported by external information about latency of data’ for further details.
4. If you are not that focused on time accuracy, just use the input telegram as reference – for example, with fixed latency. See Section 2 ‘Simple timing’ for further details.

Methods 2 and 3 can be considered equally accurate.

Note: Do not combine methods on the same instrument – this may lead to arbitrary behaviour.

7 GPS instruments

This section lists how NaviPac can handle each set of GPS instrument types. Currently, NaviPac supports the following types of GPS instruments:

- Ashtech
- C-Nav
- Geco-Prakla
- NMEA
- Leica
- POSMV Group
- Racal UKOOA
- Sea-diff / Skyfix
- Sercel NS230
- Topcon
- Trimble 4000

The following sections describe how handling of the various GPS instrument types can be done in NaviPac – but we still recommend that you check with the manufacturer whether they suggest alternative handling.

7.1 Ashtech

The Ashtech GPS drivers in NaviPac support the following data formats:

- $..GGA, time, Latitude, N/S, Longitude, E/W, Quality, # Satellites, HDOP, Antenna Alt., Unit, Geoidal separation, Unit, Age, Diff ID <CR><LF>
- $..,GGK, UTC, Date, Latitude, Longitude, Quality, NoSat, DOP, Height<CR><LF>

Timing of positioning data can be performed using the following methods:

- Locally, by registering time of reception plus fixed latency (see Section 2 ‘Simple timing’)
- Locally, by the use of PPS (see Section 5 ‘Assigning PPS to specific position input’)
- Locally, by registering time of reception plus latency (see Section 3 ‘Timing supported by external information about latency of data’) – **for RTK only**

If you want to use external clock synchronisation – that is, the PC time synchronised with the GPS clock (see Section 4 ‘Synchronising with external clock’) – you should use an NMEA driver instead of an Ashtech.

You should avoid combining any of these methods.

7.2 C-Nav

The C-Nav driver in NaviPac supports two data formats:

- C-Nav TRIN format: [010201CNavV12.8 <week> <time> <latency> <latitude> <longitude> <height> 0<HDOP> <VDOP> etc etc]
- NMEA GGA: $..GGA, time, Latitude, N/S, Longitude, E/W, Quality, # Satellites, HDOP, Antenna Alt., Unit, Geoidal separation, Unit, Age, Diff ID <CR><LF>

Timing of positioning data can be performed using the following scenarios:

- Locally, by registering time of reception plus fixed latency (see Section 2 'Simple timing')
- Locally, by the use of PPS (see Section 5 'Assigning PPS to specific position input'))
- TRIN format: Locally, by registering time of reception plus latency (see Section 3 'Timing supported by external information about latency of data')
- GGA format: By external clock synchronisation – that is, by synchronising PC time with the GPS clock (see Section 4 'Synchronising with external clock')

Please note that even though the driver allows selection of ZDA-supported timing, this will not work. Most known installations use the time in telegram mode.

7.3 Geco-Prakla

The Geco-Prakla drivers in NaviPac support the following data formats:

- TRINAV: [0101xxvvvvvvvvvvwwww<time><age><latitude><longitude><height> <UKOOA info>]<CR><LF>
- GECO3: $WGPOS,vers,no,name,gps week,timestamp,age,lat,long,height,hdop, vdop,uv, varLat,varLon,varH,ER,fix, ct,ct,#sat,#ref,sats used,ref id,]

Timing of positioning data can be performed using the following methods:

- Locally, by registering time of reception plus fixed latency (see Section 2 'Simple timing')
- Locally, by the use of PPS (see Section 5 'Assigning PPS to specific position input')
- Locally, by registering time of reception plus latency (see Section 3 'Timing supported by external information about latency of data')

Most known installations use the latency in telegram mode.

7.4 NMEA

The NMEA GPS drivers in NaviPac support the following data formats:

- $..GGA, time, Latitude, N/S, Longitude, E/W, Quality, # Satellites, HDOP, Antenna Alt., Unit, Geoidal separation, Unit, Age, Diff ID <CR><LF>
- $..,GGK, UTC, Date, Latitude, Longitude, Quality, NoSat, DOP, Height<CR><LF>
- $..GLL, Latitude, N/S, Longitude, E/W, time, status, mode<CR><LF>

Timing of positioning data can be performed using the following methods:

- Locally, by registering time of reception plus fixed latency (see Section 2 ‘Simple timing’)
- Locally, by the use of PPS (see Section 5 ‘Assigning PPS to specific position input’)
- By external clock synchronisation – that is, by synchronising PC time with the GPS clock (see Section 4 ‘Synchronising with external clock’)

You should avoid combining any of these methods.

7.5 Leica

The Leica drivers in NaviPac support the following data format:

- Binary format: <Sync> <Length> <Id> <GPS Cl> <source> <pos des> <Pos fail> <Datum> <Lat> <Long> <Height> <Check>

If operated in ASCII NMEA mode, please see the information on NMEA GPS drivers in Section 7.4 ‘NMEA’.

Timing of positioning data can be performed using the following methods:

- Locally, by registering time of reception plus fixed latency (see Section 2 ‘Simple timing’).

The age is fixed and well known – please contact Leica for the actual value.

All other methods are invalid and should be avoided.

7.6 POSMV Group

The POSMV Group driver is purely based on time in the data format.

Timing of positioning data can be performed using the following method:

- By external clock synchronisation – that is, by synchronising PC time with the GPS clock (see Section 4 'Synchronising with external clock')

The interfacing will normally occur via the **POSMVIF** module – see the dedicated manual on **POSMVIF** (http://download.eiva.dk/online-training/NaviPac%20Manuals/Various/POSMVIF.pdf) for detailed information.

All other methods are invalid and should be avoided.

7.7 Racal UKOOA

The Racal UKOOA driver in NaviPac supports the UKOOA-utilised ASCII data format:

- [LVT**<utc><data><age><datum><latitude><longitude><height><ukooa parameters><cr><lf>

Timing of positioning data can be performed using the following methods:

- Locally, by registering time of reception plus fixed latency (see Section 2 'Simple timing')
- Locally, by the use of PPS (see Section 5 'Assigning PPS to specific position input')
- Locally, by registering time of reception plus latency (see Section 3 'Timing supported by external information about latency of data')

Most known installations use the latency in telegram mode.

7.8 Sea-diff / Skyfix

The Sea-diff / Skyfix driver in NaviPac supports the following data format:

- [1123mmddyyHHMMSS.S01.21N540853.17E0090000.00+0012.3111.1222.2111.1222.233344455.566.67788999112233344556667788999123456789111 1] <CR><LF>

Timing of positioning data can be performed using the following method:

- Locally, by registering time of reception plus latency (see Section 3 ‘Timing supported by external information about latency of data’)

All other methods are invalid and should be avoided.

7.9 Sercel NS230

The Sercel NS230 driver in NaviPac is just a special variant of the NMEA driver, and it supports the following data formats:

- $..GGA, time, Latitude, N/S, Longitude, E/W, Quality, # Satellites, HDOP, Antenna Alt., Unit, Geoidal separation, Unit, Age, Diff ID <CR><LF>
- $..,GGK, UTC, Date, Latitude, Longitude, Quality, NoSat, DOP, Height<CR><LF>

Timing of positioning data can be performed using the following methods:

- Locally, by registering time of reception plus fixed latency (see Section 2 ‘Simple timing’)
- Locally, by the use of PPS (see Section 5 ‘Assigning PPS to specific position input’)

You should avoid combining any of these methods.

7.10 Topcon

The TopCon drivers in NaviPac supports two data formats:

- TOPCON NP format: NP0B3,NAVPOS,V,time,0,status,{#sat,#sat2},datum,lat,long, height, height,geoidal separator indicator, separation, hdop, vdop, hRMS, vRMS,hVel,vVel,…@85
- NMEA GGA: $..GGA, time, Latitude, N/S, Longitude, E/W, Quality, # Satellites, HDOP, Antenna Alt., Unit, Geoidal separation, Unit, Age, Diff ID <CR><LF>

Timing of positioning data can be performed using the following methods:

- Locally, by registering time of reception plus fixed latency (see Section 2 ‘Simple timing’)
- Locally, by the use of PPS (see Section 5 ‘Assigning PPS to specific position input’))
- Topcon NP format: Locally, by registering time of reception plus latency (see Section 3 ‘Timing supported by external information about latency of data’)
- GGA format: By external clock synchronisation – that is, by synchronising PC time with the GPS clock (see Section 4 ‘Synchronising with external clock’)

Please note that even though the driver allows selection of ZDA-supported timing, this will not work. Most known installations use the time in telegram mode.

7.11 Trimble 4000

The Trimble 4000 drivers in NaviPac support the following data formats:

- $..GGA, time, Latitude, N/S, Longitude, E/W, Quality, # Satellites, HDOP, Antenna Alt., Unit, Geoidal separation, Unit, Age, Diff ID <CR><LF>
- $..,GGK, UTC, Date, Latitude, Longitude, Quality, NoSat, DOP, Height<CR><LF>
- ddddddd tt:tt:ttMdd:mm.mmmmmmN ddd:mm.mmmmmmE hhhhhh.hhh pp.p ....<CR><LF>
- [00 Thu 218 06-Aug-98 15:22:51d56:13.7620N 006:20.8340E +0066 02.2 0 0.06 0.31 38.3 +0.0000E+00 19,04,13,14,18,16,24] (RTK telegram)

Timing of positioning data can be performed using the following methods:

- Locally, by registering time of reception plus fixed latency (see Section 2 ‘Simple timing’)
- Locally, by the use of PPS (see Section 5 ‘Assigning PPS to specific position input’)
- By external clock synchronisation – that is, by synchronising PC time with the GPS clock (see Section 4 ‘Synchronising with external clock’)

You should avoid combining any of these methods.

8 Dynamic positioning instruments with timestamp

NaviPac time tags all incoming dynamic positioning telegrams (USBL, LBL, remote GPS, etc) on reception of the telegram (as discussed in Section 2 ‘Simple timing’). For some drivers, this may be overruled by using timestamp information in the telegram.

8.1 Kongsberg HiPAP/APOS and Kongsberg Hain

For the Kongsberg HiPAP/APOS and Kongsberg Hain dynamic positioning instruments, timing accuracy may be improved by synchronising NaviPac and the dynamic positioning instrument with a GPS.

The SIMSSB telegram from the HiPAP/APOS instrument ($PSIMSSB,hhmmss.ss,tp,Status,error,coord,orientation,sw,x,y, depth, ... <cr><lf>)[2] includes hereafter the time of position validity. NaviPac can be set up to use this timing information. This is done in the lower part of the instrument setup by selecting the property **Use time in telegram (hh:mm:ss)**.

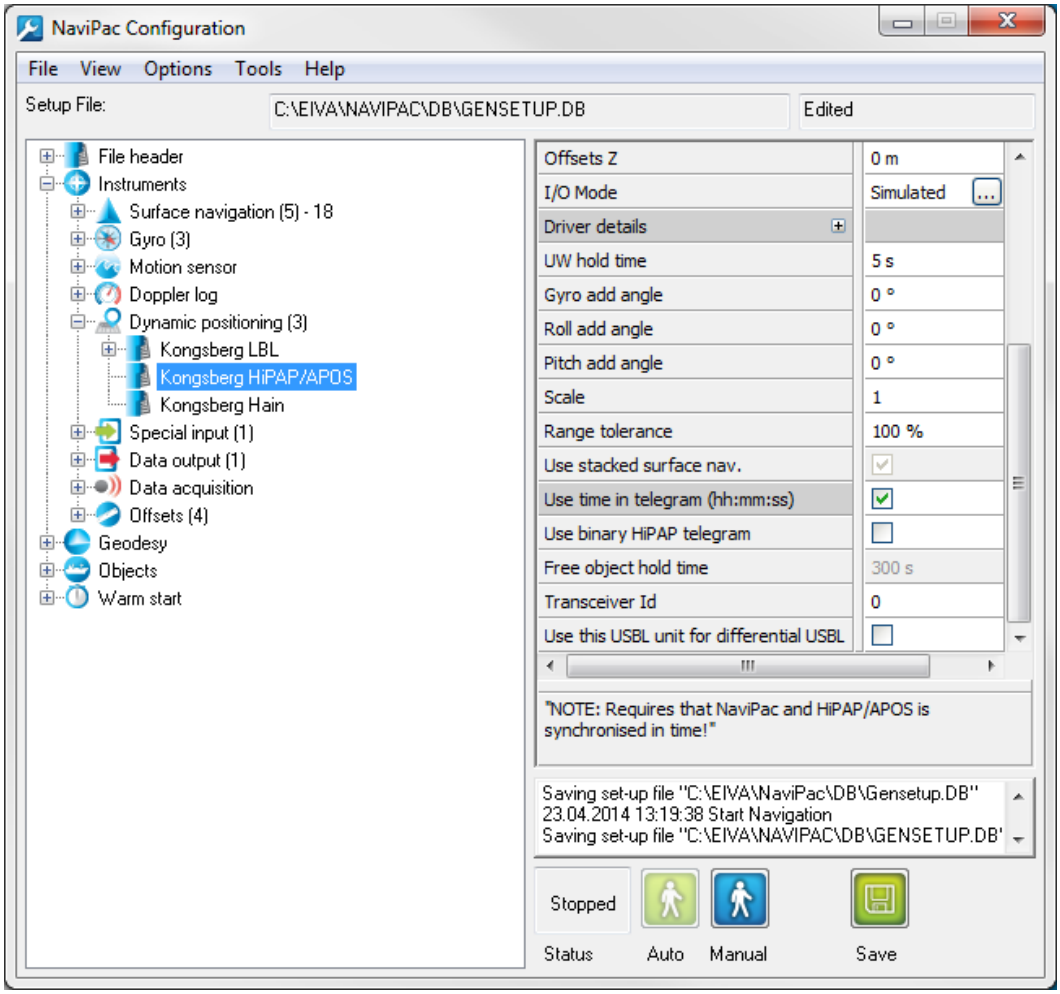

Figure 8 Time selection for Kongsberg HiPAP/APOS instrument

[2] Similar for the dual HiPAP/APOS SIMSSD telegram.

8.2 Remote GPS

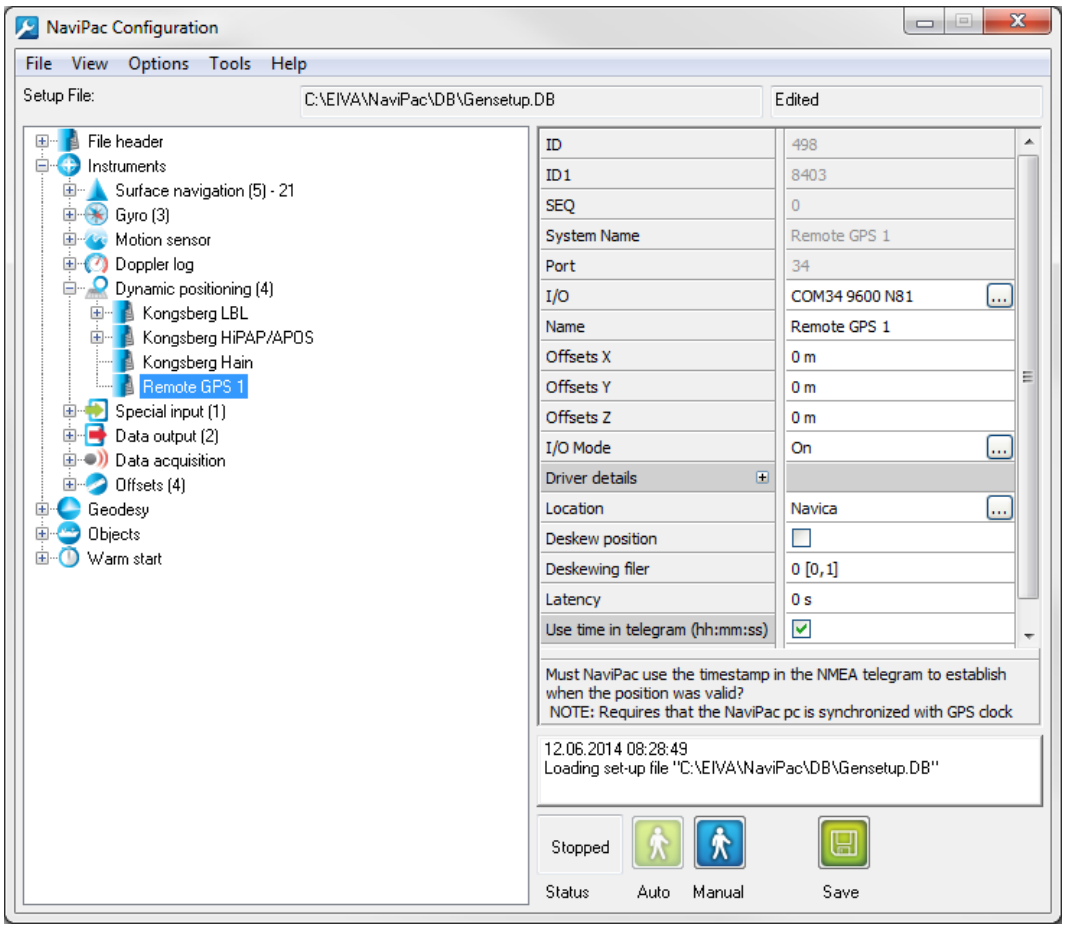

Figure 9 Time selection for Remote GPS instrument

Check the checkbox for the property **Use time in telegram (hh:mm:ss)**.

8.3 ORE BATS ($POREG)

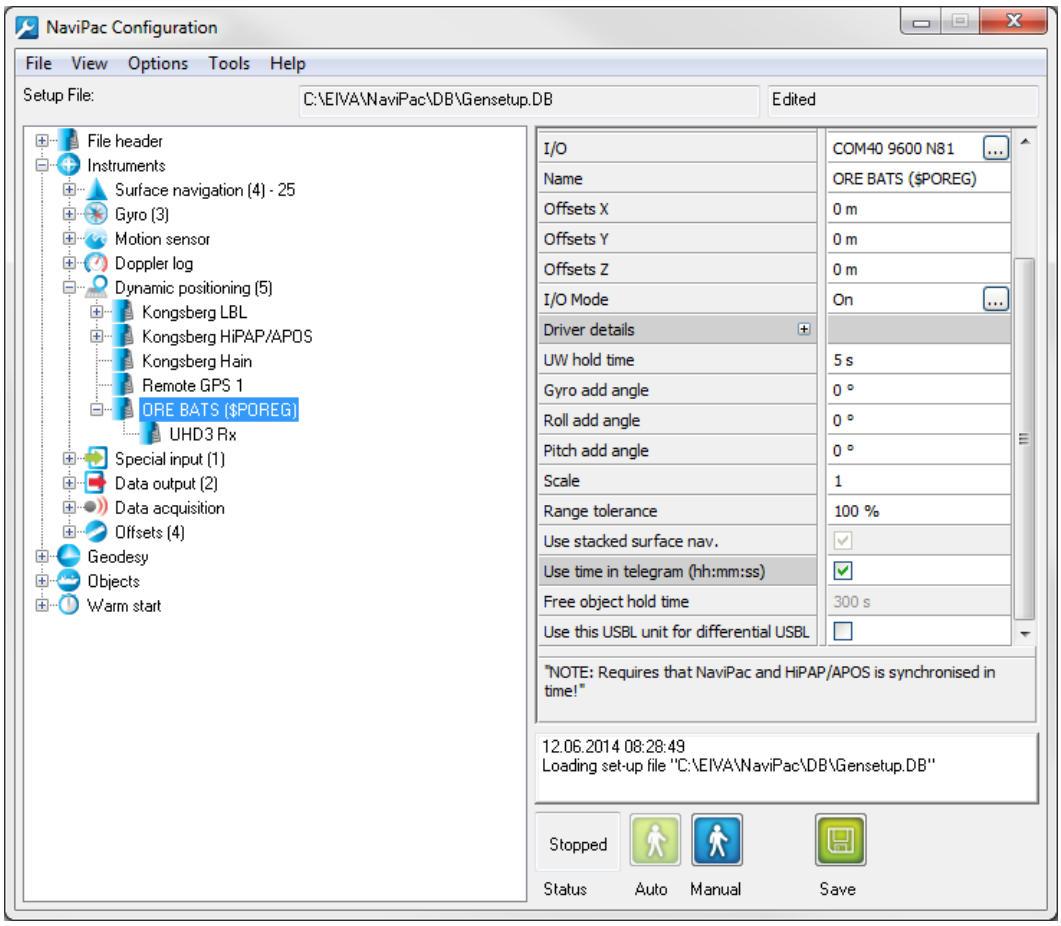

Figure 10 Time selection for ORE BATS ($POREG) instrument

If the ORE BATS ($POREG) instrument is synchronised with the GPS clock, you may select **Use time in telegram (hh:mm:ss)**.

8.4 PHINS/ROVINS Position

This instrument is handled via the special PHINS/ROVINS Position driver as discussed in the PHINS/ROVINS driver manual (http://download.eiva.dk/online-training/NaviPac%20Manuals/Appendix/PhinsOut.pdf).

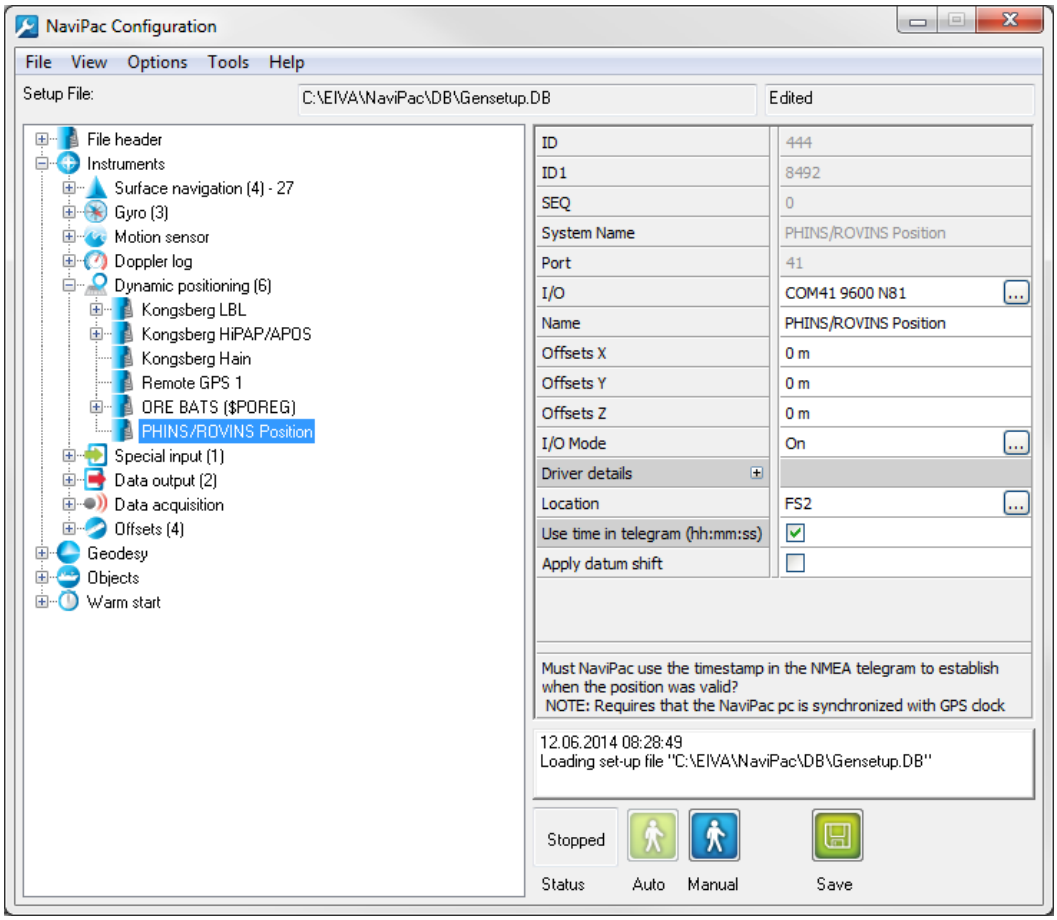

Figure 11 Time selection for PHINS/ROVINS Position instrument

The driver uses the timestamp from the PHINS/ROVINS Position instrument, and it is thus important that the systems are in sync with the GPS clock.

If the PHINS/ROVINS Position instrument is synchronised with the GPS clock, you may select **Use time in telegram (hh:mm:ss)**.

8.5 Sonardyne USBL ($PSONUSBL)

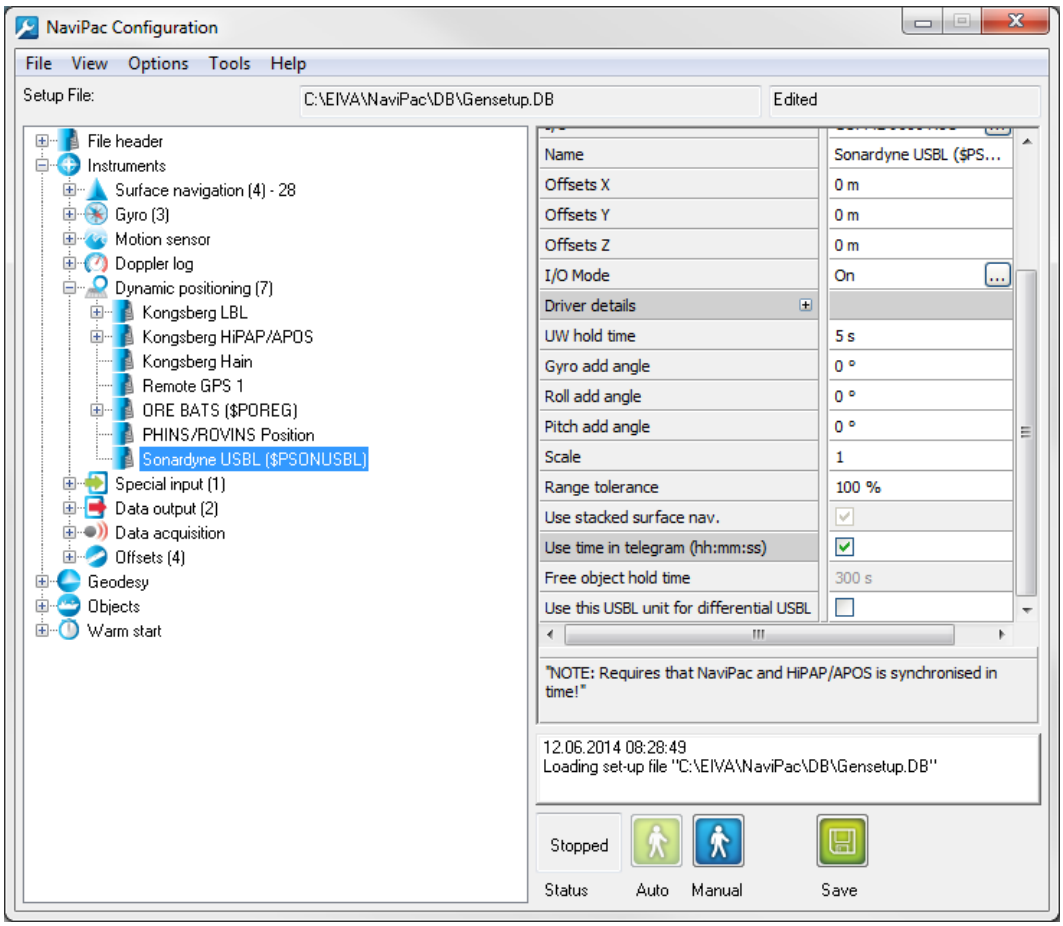

Figure 12 Time selection for Sonardyne USBL($PSONUSBL) instrument

If the Sonardyne USBL($PSONUSBL) instrument is synchronised with the GPS clock, you may select **Use time in telegram (hh:mm:ss)**.

9 ATTU

We see more and more systems where time tagging is handled via the ATTU (The EIVA time-tagging unit formerly called TimeBox). Please see the dedicated ATTU user manual for detailed information (http://download.eiva.dk/online-training/Various%20Manuals/ATTU%20Manual.pdf).

Using the ATTU, you can obtain time tagging accuracy significantly better than 1 millisecond – and thus guarantee that no errors are introduced into the data due to time tagging[3].

In most cases, you will still interface some instruments that are time tagged by the source (eg GPS data) directly into NaviPac – and it is an absolute requirement that you in these cases synchronise the PC time with the GPS clock.

10 Test of timing

How do you test whether the timing is correct? Well, that is a rather good question!

10.1 Checking calculated latency

The first step is to validate that the calculated/shown latencies are reasonable. This can be monitored in the **GPS Status** window (see the dedicated manual on **GPS Status** (http://download.eiva.dk/online-training/NaviPac%20Manuals/8_GPS%20Status.pdf) for detailed information):

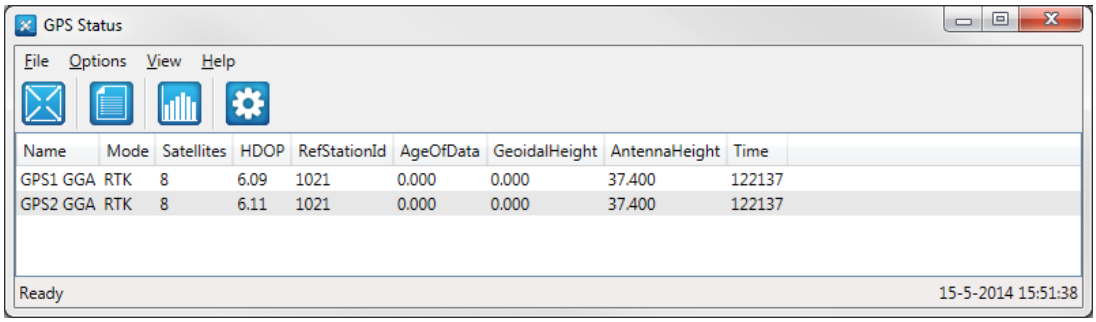

Figure 13 GPS Status window illustrating age of data

[3] According to specification and tests performed by the Norwegian time group founded by Statoil.

Here, the **AgeofData** field shows either the dynamic latency in the telegram (eg Ashtech GPS and Racal UKOOA) or the calculated latency (eg Trimble).

Please note that the calculated latency shows how old the navigation data is when NaviPac is using it – this cannot be compared with the latency when it is transmitted from the instrument. The **AgeofData** value might typically be up to one second higher.

Another test can be done by logging some data in general format in the **LogData** module, where the incoming telegrams are displayed together with the final time tagging:

- R44 5 2002:01:09:14:00:49.000 $GPGGA,130049.00,5408.75287,N,0918.29618,E,3,7,2.10,+00037.4,M,+00000.0, M,0.87,1021/R44

If time is synchronised with a GPS, there should be a good correlation between the time tagging and the time in the telegram.

The same principle can be used for time-synchronised Kongsberg HiPAP/APOS instruments:

- R496 10 2002:01:09:14:00:49.021 $PSIMSSD,130049.02, B02, A,, M, 100.00,10.00,60.00,1.21,1.1,1.2,1.3,23.4,,,*hh-B /R496
- R496 10 2002:01:09:14:00:49.010 $PSIMSSB,130049.01, B02, A,, C, H, M,100.07,10.10,61.26,1.2, N,,*hh-A /R496
- R496 10 2002:01:09:14:00:49.030 $PSIMSSD,130049.03, B02, A,, S, 101.00,10.00,61.00,1.21,1.1,1.2,1.3,23.4,,,*hh-C /R496

10.2 Testing of time synchronisation

If NaviPac is set up to synchronise the internal PC clock via interfacing of either the UTC or the ZDA telegrams, it might be rather hard to check the validity of the synchronisation.

One method is again to check a general log file, where the time in the incoming telegram should correspond to the timestamp:

- R505 12 2001:01:25:21:48:04.001 UTC 01.01.25 21:48:04 56 /R505

Please note, however, that identical timing does not guarantee correct time synchronisation.

If you want to check the synchronisation further, it is a good idea to start the NaviPac **Alarm Monitor** (**Online >View > Alarm Monitor**).

	Type	Limit	V1	V2	Val	Alarm	Count	Reference
NaviPac	Error	0.000	0.000	0.000	0.000	No data for offset number 22	12	
NaviPac	Warning	0.000	0.000	0.000	0.000		0	
NaviPac	Info	0.000	0.000	0.000	0.000	GPS 1: Non RTK	2	
DataIO	Input	0.000	0.000	0.000	0.000		0	
NaviScan	Message	0.000	0.000	0.000	0.000		0	

Figure 14 Alarm Monitor data

In addition to normal messages, alarms, errors and warnings, this module includes messages regarding the time synchronisation:

- *UTC/ZDA Port - xx PPS received without incoming telegram:* PPS received – multiple PPSs have been received since last telegram.
- *String discarded T tttttt (PPS ppppp) PPS Count xx:* String received – multiple PPSs have been received since last string.
- *UTC/ZDA Port - No PPS received for yyy ms:* Telegram received – but it is a long time since NaviPac received a PPS.
- *UTC/ZDA Port - No PPS received:* Telegram received – but NaviPac has never received a PPS.
- *PPS Time has not been performed for xx cycles - Check if UTC telegram is correct*: PPS received – but could not synchronise the clock.
- *First update time from hh:mm:ss.sss to hh:mm:ss.sss:* NaviPac has synchronised the time for the first time.
- *hh:mm:ss - PPS Time sync received on strange time hh:mm:ss.sss:* The PPS corresponding to time (first hms) was received, but the clock differs too much. This cannot be real – if it continues, try a forced time synchronisation under **Online > Edit Date and Time** or restart NaviPac.
- *hh:mm:ss Update time from hh:mm:ss.sss to hh:mm:ss.sss:* PPS signal was received successfully, and the clock was adjusted.