GPS.DLL Timing and Synchronization

By John Lindberg

The most common device driver in HYPACK® would most likely be the GPS.dll (GPS NMEA-0183). It will work with practically any position device out there. Pretty much every GPS receiver, inertial system, USBL system and total station has a way to output a NMEA string, so, when in doubt about an output format, check the device manual and see if it outputs a NMEA string that the GPS.dll can decode. Surprisingly, few users really understand the internal workings of this driver, so my intention is to clear up a few things in this article and future articles. In this article, I will discuss timing and synchronization.

TIMETAGS

Every device connected to HYPACK® gets logged with a timetag in a raw data file, along with pertinent information from that device. You can open a raw data file in a text editor and see for yourself. Below the header of the data file, you will see a series of data records, each containing a timetag. The data will look something like this:

```
POS 1 33970.670 1078307.204 248069.840 0.548
QUA 1 33970.670 7 8.304 1.696 8.000 0.000 0.000 0.000 0.000
RAW 1 33970.670 4 301090.84332 -883213.88728 -27.39979 142610.67050
GYR 1 33970.670 309.690
TID 1 33970.670 -0.548
HCP 1 33970.670 -0.022 -0.851 2.952
KTC 1 33970.670 7 -89.894 -89.894 -90.442 0.000 0.000 0.000 0.561
```

The value highlighted above would be the timetag (in seconds past midnight) for the position device. Timetags can be derived in a number of ways in the GPS.dll depending on the output messages from the GPS receiver, and also on options selected in the device driver setup. This is to accommodate different time-tagging options, as there could be many reasons for using a specific timetag.

By default, HYPACK® will time-tag data based on the computer time when the message is first received. It is not exactly the computer time, but the HYPACK® "Veritime" clock model, as this is much more stable than a typical computer clock. This method can be sufficient for many applications, but for time-critical applications (multibeam, INS, etc.), you may need to base your timetag on something else. That is where the GPS.dll options can get tricky, and understanding how to set up the driver in these situations can prevent a lot of grief later on.

NMEA STRINGS

As far as I know, any modern-day GPS receiver is capable of outputting various NMEA strings. When in doubt, consult the user manual for the receiver, and configure your GPS to output a GGA string for position. *Forget about all the other position strings* (RMC, GLL, GGK, etc.) though they will all work fine in the GPS.dll)!!

A **GGA string** has everything you need for position: latitude, longitude, elevation, and GPS quality. Whether you set the output rate to 1 Hz or 20Hz, HYPACK® can handle it! Here is a GGA string:

```
$GPGGA,172000.55,4135.62984976,N,07243.42170363,W,2,08,1.0,41.105,M,
-34.221,M,4.4,0138*40
```

The **ZDA output string** is required if you want to synchronize the computer clock to the GPS. This can be beneficial for a number of reasons:

- Theoretically, the ZDA timetag is the exact satellite time eliminating any latency issues that may occur with the GGA timetag—the time of the received position.
- Most Ethernet devices (multibeam, INS, etc.) are already time-tagged by a GPS time source and are outputting a timetag to HYPACK®. In this case, we want our GPS positions using the same time basis. Since most serial devices (sounders, magnetometers, etc.) do not output a timetag, and the PC is synchronized to the same GPS time, your serial device inputs are all being tagged on the same time basis.

This is an example of a ZDA string. HYPACK® expects a 1 Hz ZDA output from the receiver, so be sure you receiver is set up correctly for this!

```
$GPZDA,172002.009,07,09,2016,00,00*52
```

HYPACK® GPS.DLL SETUP —NO SYNCHRONIZATION

Let's see how the setup options for the GPS.dll affect the timetags in the RAW file. We will start with the default setup, where there is no synch device selected in HARDWARE.

In HYPACK® SURVEY, the GPS device window will show "computer" as the timing source as shown in Figure 1:

FIGURE 1. *Synchronize Computer Clock Set to None (left), GPS Device Window (righ)*
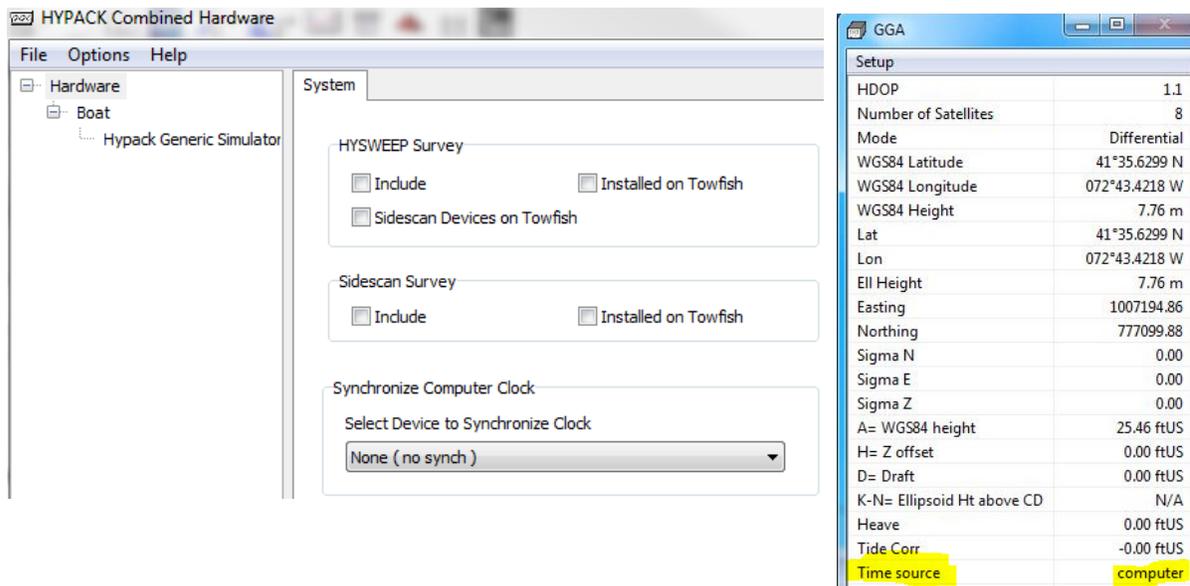
Now let's look inside the raw file:

```
POS 0 43273.768 1007195.721 777099.547
QUA 0 43273.768 7 3.000 1.000 9.000 2.000 0.000 0.000 0.000
RAW 0 43273.768 4 413562.98033 -724342.16262 6.62200 160111.65000
EC1 1 43273.781 10.000
```

All records are time-stamped with the computer time. Notice that the POS, QUA and RAW records have the exact same timetag (43273.768). These messages are always logged from the GPS.dll, and will always have the same exact same timetag. Now look at the last value in the RAW record. This is the UTC time (16:01:11.65) from the GGA string. In order to compare the UTC time, you need to add the local offset (for Eastern Time it is currently -4 hours - 12:01:11.65) and convert to seconds past midnight. You will notice the HYPACK® timetag does not match the UTC time (43271.65), so in this case you can be sure that time synchronization is not being used.

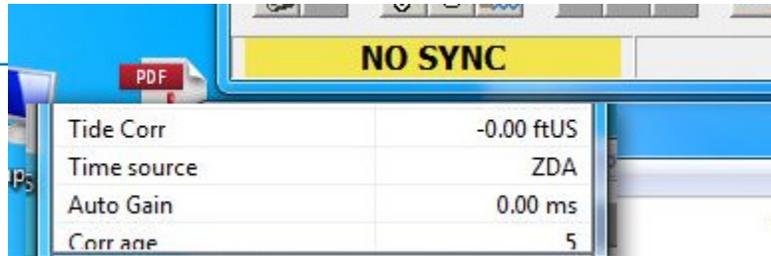# HYPACK® GPS.DLL SETUP — SYNCH TO THE GPS.DLL

Now we will select the GPS.dll as the sync device.

FIGURE 2. *Synchronizing to the GPS Driver*
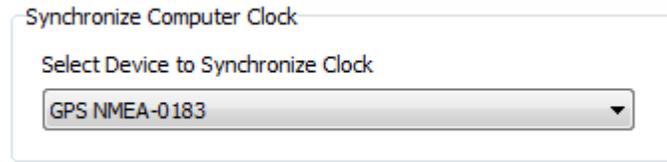
In HYPACK® SURVEY, you will notice a yellow box displaying NO SYNC (Figure 3):

FIGURE 3. *HYPACK® SURVEY shows "No Sync"*

This is normal, as SURVEY needs about 30 seconds or so to fully sync the PC clock. Once synced, the yellow box will disappear and the GPS window will display ZDA for the time source.

Now we will look inside a data file:

```
EC1 1 43504.163 10.000
POS 0 43504.100 1007194.265 777100.088
QUA 0 43504.100 7 3.000 1.100 9.000 2.000 0.000 0.000 0.000
RAW 0 43504.100 4 413562.98924 -724342.19457 8.15000 160504.10000
SYN 0 43504.007 6 43504007.999 43504007.546 261.000 0.000 0.000 -
2.124
```

The POS, QUA, and RAW records are stamped with the GGA time. Convert the UTC time plus the local offset (12:05:04.100000) and you will see the timetags match perfectly, demonstrating that the computer is synced to the GPS (The HYPACK® timetag can vary +/-1ms due to rounding errors.) Notice the SYN message in the above snippet. This is always logged when syncing the PC clock (more on the SYN message in a future article).

Since the depth record from the sounder in this example is a serial device, it is still time stamped with the computer time, as there is no timetag available in the output string. Because you are syncing everything to GPS time, you can be assured all your devices are time stamped correctly.

BE CAREFUL!

- **Do not manually adjust your computer clock while time syncing!** Your PC time may not change immediately due to Windows® and PC inconsistencies. Manually changing the time will just compound issues.
- **If trying to sync on a Windows® 10 computer**, see my May 2016 newsletter (http://www.hypack.com/new/portals/1/pdf/sb/05_16/Workaround%20for%20Time%20Sync%20Issue%20in%20Win10.pdf)

- In the GPS.dll Setup under the Advanced tab, there is a section **USE AT YOUR OWN RISK!** Please abide by the instructions!

FIGURE 4. USE AT YOUR OWN RISK!
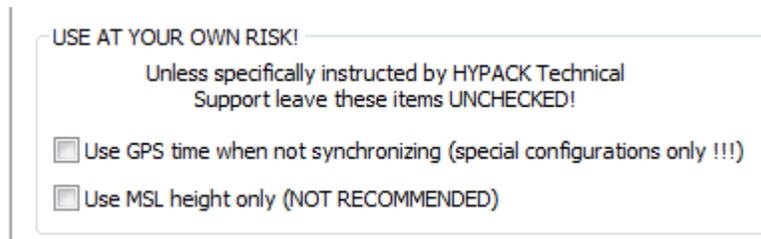
- The **Use GPS Time** option will *always* use the GGA timetag (including local offset), even when not syncing the PC clock. This can cause grief if you have serial devices that are relying on the computer timetag. You may end up with erratic timetags as shown below.

```
POS 0 43571.950 1007194.220 777099.901
QUA 0 43571.950 7 3.000 1.100 9.000 2.000 0.000 0.000 0.000
RAW 0 43571.950 4 413562.98617 -724342.19554 7.96900 160611.95000
EC1 1 43263.999 10.000
EC1 1 43264.015 10.000
EC1 1 43264.030 10.000
POS 0 43572.000 1007194.220 777099.901
QUA 0 43572.000 7 3.000 1.100 9.000 2.000 0.000 0.000 0.000
RAW 0 43572.000 4 413562.98617 -724342.19554 7.96900 160612.00000
```

The example above shows a 307 second time difference between position and depth records! While the data is most likely salvageable, there is no need to get yourself into a predicament without a good understanding of why or why not you are selecting options.

So hopefully you can see the importance of the time sync options in the GPS.dll. More GPS.dll tips and tricks to come!