SURVEY Clock Synchronization with ZDA Sentences

By Pat Sanders

In some survey operations with multibeam sonar systems, you *must* synchronize (synch) the HYPACK® SURVEY clock to the GPS' UTC clock. The primary reason being that the multibeam system sends packets of information that are time-stamped with UTC times. In order to correlate all of the information from HYPACK® SURVEY with the external network packets, all systems have to be on the same time basis.

In other survey operations, hydrographers are not required to synch the clocks. Even though it is not required, we get feedback that many hydrographers go ahead and synch the clock, even though it is not required.

It can probably be stated as a Hydrographic Law that the more accurately you synch the clocks, the better your survey results will be.

(I think I'll write Pat's Book of Hydrographic Laws when I retire….)

Survey Clock Synchronization Theory

In HYPACK® SURVEY, there are two methods that can be used to synch the HYPACK® SURVEY clock to the GPS UTC clock. Both of them involve the 1Hz output of the ZDA sentence from your GPS.

- **Method 1: ZDA Only**

  In this method, HYPACK® SURVEY continuously monitors the feed from the GPS unit. Every time it receives the first character of a new NMEA sentence (the '$' in the "$GPxxx…") it goes to the VERITIME clock model to get the matching HYPACK® SURVEY time. The $GPZDA message contains the UTC time.

  It is hoped that the GPS transmits the $GPZDA message at exactly the UTC time tic. If it does, then this technique should work fine. If it does not, then you are going to have a time-synch mess on your hands.

  Note: We only want the GPS to transmit one $GPZDA sentence per second (1Hz). Faster or slower update rates can adversely affect the time synchronization.

  HYPACK® SURVEY compares the UTC time stamp in the $GPZDA sentence. It then makes small adjustments to speed up or slow down the VERITIME clock model to bring it into coincidence with the GPS UTC clock. It is better to make small adjustments than to make giant adjustments back and forth.

  Note: It takes about a minute for us to synchronize.

- **Method 2: ZDA and 1PPS Pulse**

  Most GPS systems can output a 1PPS pulse at the UTC second tic. In order to read this pulse, we have to bring it into a special box that can change it to a 0/+5V sawtooth. Our

1PPS box then feeds this signal into the CTS line of a serial port, which we can monitor to determine the arrival of the 1PPS pulse.

We capture the VERITIME (HYPACK® SURVEY) time at the arrival of the 1PPS pulse. We then wait for the next $GPZDA sentence to arrive that provides us the UTC time of the 1PPS pulse.

After that, the process is the same as Method 1. We have the GPS UTC time and the HYPACK® SURVEY clock time and can compare the two and make small adjustments to the VERITIME clock model inside of HYPACK® SURVEY.

TESTING

In order to test the clock synchronization accuracy, we set up a Trimble 461SPS GPS unit. This unit is one that we recommend to users and are pretty familiar with. We can set it up to output NMEA sentences via an RS-232 connection, a TCP/IP connection or a UDP connection.

In the RAW data files of HYPACK® SURVEY, we can instruct the GPS device driver to output a synchronization message ("SYN"). This message is written as each ZDA sentence arrives and contains the clock time (in microseconds) from the ZDA arrival and the corresponding HYPACK® SURVEY time from VERITIME.

By taking the difference between the two times, we can determine the difference between the two clocks and get an idea of how closely the two times are tracking. I collected data for each connection type (RS-232, TCP/IP, UDP) in two minute segments and then analyzed the difference in times. What was most interesting to me was the Standard Deviation of the time differences, which would show me how much 'wandering' there was between the two clocks. Also of interest were the Minimum and Maximum time differences, which would show me the maximum error between the two clocks.

TESTING SYNCHRONIZATION USING ZDA ONLY

The results shown below are for Method 1, where we did not use the 1PPS pulse to aid the time synch. In these tests we did not manipulate windows (re-sizing and moving them) or start/stop other programs (*.exe).

TABLE 1. *Testing Clock Synchronization Using Only the ZDA Message from a Trimble 461SPS*

Connection Type	RS-232 (MSec)	TPC/IP (MSec)	UDP (MSec)
Average Time Difference	0.049	0.034	0.023
Standard Deviation of Time Differences	1.159	0.745	0.563
Minimum Time Difference	-1.970	-1.833	-1.107
Maximum Time Difference	1.999	2.142	1.141

The UDP connection gave the most accurate clock synchronization with a standard deviation of less than 1 millisecond and maximum time differences of just over 1 millisecond. Even in the 'worst case' RS-232 connection, the standard deviation was just over 1 millisecond and maximum time difference was just under two milliseconds. That's 2/1000ths of a second for those of you who skipped math classes. All of the different connections proved that they could synchronize the HYPACK® SURVEY clock with extreme precision.

Testing Synchronization With Other Windows® Activity

In our next test, I wanted to determine the effects that 'playing' with Windows® during the data collection would have on our ability to synchronize the clock, once again using Method 1. During the data collection, I would re-size windows, click-and-drag windows about the screen, start other Windows® applications (Paint, Word, Notepad, etc.). Tests were done with the RS-232 and TCP/IP connections to determine the effect.

TABLE 2. *Testing Clock Synchronization to the ZDA Message - Effects of Other Windows Activity*

Connection Type	TCP/IP	TCP/IP	RS-232	RS-232
Windows Activity:	No Playing	Playing	No Playing	Playing
Average Time Difference:	0.034 mSec	0.108 mSec	0.049 mSec	0.014 mSec
Standard Deviation of Time Differences:	0.745 mSec	2.247 mSec	1.159 mSec	0.748 mSec
Minimum Time Difference:	-1.833 mSec	-6.010 mSec	-1.970 mSec	-1.972 mSec
Maximum Time Difference:	2.142 mSec	9.187 mSec	1.999 mSec	1.933 mSec

In this test, the TCP/IP results showed significant deterioration when playing with Windows® during SURVEY. The maximum time difference between the GPS clock and the HYPACK® SURVEY clock grew from 2.142mSec (not playing) to 9.187mSec (playing). However, for the RS-232 tests, the 'playing' results were either the same, or better, than the non-playing results.

I talked to Mircea Neacsu, the HYPACK programmer who developed the VERITIME clock model and our time synchronization routines, about why the network results would be degraded, but the RS-232 results improved. He put the RS-232 results down as an abnormality. They should be about the same (and the Minimum and Maximum time differences were almost identical.). The low level communications is handled by a routine in HYPACK® named COMMDRV. This routine operates at a very 'high priority' thread that takes priority over all SURVEY tasks. It monitors input streams (both serial and network) and shares the information with different device drivers. Before your electrons run up the cable and into COMMDEV, they must also pass through lower level Windows® routines. It is Mircea's opinion that the network handling functions of Windows® has a lot more going on than the RS-232 handling functions and is affected more by the Windows® OS when you go to manipulate windows and run other programs.

So, in summary of this test:

- There is a penalty to pay in reduced synchronization accuracy when you play with your Windows.

- Normally, the effect is small and probably will not have any apparent effect on your survey results.
- Stop playing DOOM while surveying. (Just in case…)

TESTING SYNCHRONIZATION USING THE 1PPS BOX

We had done previous tests to demonstrate the improved accuracy of using the 1PPS output to synch the clocks. I repeated that test, using the RS-232 output, which was the worst case of our Method 1 tests. During the test, I decided to 'stress' the serial port by sending GGA, VTG, GSV, GST and HDT sentences at 10Hz and the ZDA sentence at 1Hz. That's a lot of info to pump over a serial connection, so I boosted the baud rate to 115,200. The 1PPS pulse was brought into the same serial connection, using a HYPACK 1PPS box.

The results were pretty amazing:

TABLE 3. *Testing Clock Synchronization Using the 1PPS Box*

Connection Type	RS-232 without 1PPS	RS-232 with 1PPS
Windows Activity:	No Playing	No Playing
Average Time Difference:	0.049 mSec	0.001 mSec
Standard Deviation of Time Differences:	1.159 mSec	0.005 mSec
Minimum Time Difference:	-1.970 mSec	-0.005 mSec
Maximum Time Difference:	1.999 mSec	0.025 mSec

Those numbers are in milliseconds. Read them again. Using the 1PPS box, HYPACK® SURVEY was able to synchronize the clock to a **maximum** error of less than 3/100,000ths of a second. That's incredible!

LEICA MN1250 TEST

Towards the end of my testing, we received a Leica MNS1250 RTK GPS in the office. I decided to perform a Method 1 test (no 1PPS feed) using the unit's RS-232 output. I set it up to output their $GPGGK message (positions) at 10Hz and the $GPZDA message at 1Hz. The results were very impressive, even better than the numbers we had achieved with the Trimble 461SPS.

Would you notice the difference in your survey data? Probably not. We're looking at a maximum difference of 2 mSecs (2/1000ths of a second). If you were rolling 5 degrees per second, a mis-synchronization of 2 mSecs would cause a roll correlation error of 0.01 degree. That's less than the roll accuracy of your MRU.

TABLE 4. *Testing Clock Synchronization Using the Leica MNS1250*

GPS Unit:	Trimble 461 (RS-232)	Leica MNS1250 (RS-232)
Average Time Difference	0.049mSec	0.007mSec
Standard Deviation of TDs	1.159mSec	0.073mSec

GPS Unit:	Trimble 461 (RS-232)	Leica MNS1250 (RS-232)
Minimum Time Difference	-1.970mSec	-0.124mSec
Maximum Time Difference	1.999mSec	0.155mSec

CONCLUSIONS

So, in conclusion, what have I learned from all of this……

- HYPACK SURVEY is doing a hell of a job synchronizing its internal clock to the GPS clock.
- Operating without a 1PPS box would seem to satisfy the needs of most surveyors. However, if I was performing multibeam surveys in a dynamic environment, I would probably add the 1PPS output, as every little bit helps.
- Although playing with Windows® (re-sizing, moving, starting/stopping other programs) did not greatly affect the time synchronization statistics, they did cause a slight degradation in the time synching in the TCP/IP case. Therefore, just in case, don't mess with your windows when you are 'On-Line'.