



Using GNSS+INS Systems in HYSWEEP® with Standard Drivers

By Dave Maddock

Most high-end inertial navigation systems (INS) have their own dedicated drivers in HYPACK®. These drivers have been designed around the proprietary communications protocol of the manufacturer. This allows HYPACK® to deliver a quality interface tailored to the needs of each sensor. However, there are times when it may be desirable to configure the sensor for alternative outputs or use a new GNSS+INS product that doesn't yet have a custom driver developed.

This article will demonstrate how standard HYPACK® drivers were configured to support the Applanix APX-15 in this way. The APX-15 is an INS solution for UAVs that does not provide the same proprietary data stream that the POS/MV does. Hence, the existing Applanix POS/MV drivers cannot be used with the APX.

NMEA + PASHR

The Applanix APX-15 provides a NMEA data stream like any other GNSS. However, the NMEA standard does not define sentences for inertial attitude. The APX-15, like many other manufacturers, support the pseudo-NMEA PASHR string for such data:

\$PASHR,hhmmss.ss,HHH.HH,T,RRR.RR,PPP.PP,,rr.rrr,pp.ppp,hh.hhh,1*XX

TABLE 1.

Field	Name	Symbol	Example
1	Log header	\$PASHR	\$PASHR
2	UTC Time	hhmmss.ss	133815.33
3	Heading in decimal degrees	HHH.HH	+015.00
4	True Heading	T	T
5	Roll in decimal degrees	RRR.RR	+0.5
6	Pitch in decimal degrees	PPP.PP	-0.2
7	Reserved		
8	Roll accuracy	rr.rrr	0.123
9	Pitch accuracy	pp.ppp	0.123
10	Heading accuracy	hh.hhh	0.123
11	GPS update quality flag	1	1
12	Checksum	*XX	*2A

Of course, the GPS.dll driver in HYPACK® has everything you need to parse position, heading, quality information, time syncing, and calculate real-time kinematic tides (RTK) from

a NMEA data stream, but it doesn't handle attitude data or support PASHR and other non-standard strings.

NOTE: PASHR strings can vary somewhat between manufacturers. The configuration of the parsing driver outlined below may need to be modified for your equipment. Review the manual for your equipment to verify the PASHR variant used.

PASHR IN HYSWEEP®

One option is to parse the PASHR string separately with another driver such as gendevparse.dll. However, if you're using a high-accuracy INS, odds are that you're collecting multibeam data. Hence, you want to read and log the attitude data in the HYSWEEP®.

NOTE: There is no HYSWEEP® driver dedicated to parsing PASHR strings, so the way to read this string is to configure an instance of the Generic Attitude driver.

FIGURE 1. Generic Attitude Driver Setup

Until now, this HYSWEEP® driver only worked with serial communication so to combine these two drivers in the way I've described above required several extra COM ports, serial cables, and splitters. It was also critical that HYPACK® time sync was configured appropriately or your serial data in HYSWEEP® might have latency. It will work, but what a mess!

Starting with HYSWEEP® v. 17.2.14, the Generic Attitude driver supports UDP input. Configure your INS to output the NMEA stream to a UDP port and both gps.dll and HYSWEEP® can read the same feed. To insure the most accurate timing, the Generic Attitude driver has also been updated to support parsing NMEA-style timing fields.

Figure 1 shows how to configure the driver to parse PASHR, using the time field from the string for the message timetag.

Generic Driver Setup

Synchronization

☒ Start of Line ☒ End of Line <cr><lf>

Start Character Sequence: \$PASHR

Message Length: 1

Device Setup Script:

Parsing

☒ Comma Separated Fields ☐ Space Separated Fields

☐ Fixed Field Positions

Separated Fields

	Field Number (1=1st)	Multiplier
☒ Time	2	
☒ Pitch	6	1.000
☒ Roll	5	1.000
☒ Heading	3	1.000
☐ Heave	0	0.000
☐ Sound Velocity	0	0.000
☐ Depth	0	0.000
☐ Pressure	0	0.000
☐ Temperature	0	0.000
☐ Salinity	0	0.000

☐ Output strings to UDP port 4020

OK Cancel

APPLANIX APX-15 EXAMPLE

Figure 2 demonstrates the configuration of the APX-15. The web interface will look similar for other Trimble products.

FIGURE 2. APX-15 Firmware Configuration

Turn on “UDP Mode” for the desired port and set the PASHR update rate. It may ask you to specify a destination IP address or “broadcast” mode. HYPACK® will work with either, but we have had better success avoiding broadcast mode with the APX-15.

If the system supports it—as the APX-15 does—it is a good idea to use two UDP output ports, one for standard NMEA and one for the PASHR string. This will avoid potential problems with the attitude data clogging up the gps.dll buffers.

