# Making Sense of Hexadecimal Bit Coding in HSX Files

By Mike Kalmbach

One good thing about the HYSWEEP® HSX file format is that information is saved as text and can be viewed in a text editor. The value is diminished however, if you can't make sense of what you see. That could easily be the case for certain bit coded values, so here comes an explanation.

## *What is Bit Coding?*

Bits (Binary digits) are what computers are about-- 1 or 0, On/Off, True/False, Yes/No. Everything is a 1 or a 0 to a computer. Since most people aren't computers, bits are usually viewed in groups, making them easier to work with; 8 bits is a byte, 16 bits is a word, 32 bits is a double word, and so on. So when you see the character "A" on the screen, the computer sees the 8-bit number 0100 0001 and computer user and computer are both happy. (Base 2 numbers such as 0100 0001 are somewhat unwieldy and alternatives are discussed below).

Back to bit coding. Suppose there are a number of multibeam data attributes, and that each attribute is independent of the others. For example, data is roll corrected; true or false, data is pitch corrected; true or false, dual head system; true or false, and so on. Each attribute is a bit and combining the bits into a byte (8 bits) or word (16 bits) is bit coding. Simple, right? Right!

## *Example of Bit Coding in the HSX Format*

Using the Sonar Flags field of MBI and RMB records, we provide data attributes using bit coding:

```
0000 0001 – roll corrected by sonar.
0000 0010 - pitch corrected by sonar.
0000 0100 - dual head system.
0000 1000 - heading corrected by sonar.
0001 0000 - medium depth: slant ranges recorded to 1 dm resolution.
0010 0000 - deep water: slant ranges divided by 1 m resolution.
0100 0000 - SVP corrected by sonar.
1000 0000 - topographic device; upgoing beams accepted.
```

So, if data is roll, pitch and heading corrected in medium depth water, the bit code is 0001 1011 base 2.

Base 2 numbers are not very convenient. They tend to have too many digits. Base 10 is nice for counting and math, but not so useful with bit coding. Base 16, called hexadecimal, is a good compromise.

## HEXADECIMAL

In base 2, the digits are 0 and 1. In the familiar base 10, digits are 0 – 9. In base 16 hexadecimal, where each digit represents one of 16 values, the digits 0 – 9 are used plus six extras. The extras are A – F. The conversion table shows how base 2, 10 and 16 are related.

*Conversion Table*

| Base 2 | Base 10 | Base 16 |
|---|---|---|
| 0000 | 0 | 0 |
| 0001 | 1 | 1 |
| 0010 | 2 | 2 |
| 0011 | 3 | 3 |
| 0100 | 4 | 4 |
| 0101 | 5 | 5 |
| 0110 | 6 | 6 |
| 0111 | 7 | 7 |
| 1000 | 8 | 8 |
| 1001 | 9 | 9 |
| 1010 | 10 | A |
| 1011 | 11 | B |
| 1100 | 12 | C |
| 1101 | 13 | D |
| 1110 | 14 | E |
| 1111 | 15 | F |

Going back to the example, Sonar Flags 0001 1011 base 2 can easily be seen to equal 1B in hexadecimal. Well, maybe after a little practice. It's not easy to see how the flags equal 27 base 10, and that's why base 10 isn't used for bit coding.

## SUMMARY

That wraps up hexadecimal bit coding. It is used in DV2 records to code device capabilities and in MBI and RMB records to code data attributes and to indicate the data values included with RMB records.