RTCA Special Committee 186, Working Group 5

ADS-B UAT MOPS

Meeting #4

Coding Selected Altitude

Presented by James Maynard, UPS Aviation Technologies

SUMMARY
There was a discussion during the last hour or so of our Melbourne meeting about the encoding of the "selected altitude" field. Although this discussion was not recorded in the minutes, I recall the group agreeing with Bob Saffell to encode selected altitude as a binary numeral in which the LSB has a weight of 64 feet. I believe that would be a mistake, and want to raise the issue again. This paper argues my point.

An ADS-B system includes both ADS-B transmitting subsystems aboard aircraft (and ground vehicles) and ADS-B receiving subsystems aboard other aircraft (and ground vehicles). This is true regardless of the data link.

On the UAT data link, a transmitting participant generates an ADS-B message just once a second. It is a very light burden, indeed, for a transmitting ADS-B participant to convert, its current "selected altitude" value from a binary numeral in which the LSB has a weight of 1 foot, to a binary numeral in which the LSB has a weight of 100 feet. One simply divides the first binary numeral (from the ARINC 429 data word) by 100 to get the second binary numeral. This need be done at most once per second, and perhaps even less frequently, when the selected altitude value changes.

Note that the only possible values of selected altitude are multiples of 100 feet, because those are the only values that can be entered for this variable into an aircraft's mode control panel (MCP).

If the selected altitude from the MCP were simply truncated, retaining the bit with weight 64 feet as the LSB, then reconstructing the intended selected altitude at the receiving end becomes a more complicated task. The report assembly algorithm must necessarily depend on the knowledge that the selected altitude is intended to be a multiple of 100 feet, even though it was coded as a multiple of 64 feet. This makes a more complicated decoding algorithm. This more complicated algorithm must be performed not just once per second, but as many times per second as UAT messages are received from other aircraft. The slight saving of effort in generating the transmitted message is purchased at the price of several times that effort on the receiving end.

It is unreasonable for us to expect the client application to assume the burden of decoding a message in which the LSB is a multiple of 64 feet to generate a selected altitude that is a multiple of 100 feet. This function, if it is to be performed, should be performed by the report assembly function within the receiving ADS-B subsystem. Indeed, if this decoding were not performed by the report assembly function, but left to the client application, we would have an ADS-B implementation that fails to meet the very first two requirements listed in the DO-242 MASPS:

R2.1: "When the full resolution of available aircraft data cannot be accommodated within an ADS-B message, a common quantization algorithm shall (R2.1) be used to ensure consistent performance across different implementations. To minimize uncertainty, a standard algorithm for rounding/truncation is required for all parameters. For example, if one system rounds altitude to the nearest 100 ft and another truncates, the same measured altitude could be reported as different values." – DO-242, §2.1.1.1

R2.2 "The output of ADS-B shall (R2.2) be standardized so that it can be translated without compromising accuracy." – DO-242, §2.1.1.2

An ADS-B transmitting system that represents selected altitudes that are multiples of 100 feet as binary numerals that are multiples of 64 feet does not ensure consistent performance, as the table below shows. (The pattern of coding errors repeats every 1600 feet.)

Selected Altitude from MCP	**Encoded selected altitude**	**Coding Error**
0 feet	$0 \times$ (64 feet) = 0 feet	0 feet
100 feet	$1 \times$ (64 feet) = 64 feet	-36 feet
200 feet	$3 \times$ (64 feet) = 192 feet	-8 feet
300 feet	$4 \times$ (64 feet) = 256 feet	-44 feet
400 feet	$6 \times$ (64 feet) = 384 feet	-16 feet
500 feet	$7 \times$ (64 feet) = 448 feet	-52 feet
600 feet	$9 \times$ (64 feet) = 576 feet	-24 feet
700 feet	$10 \times$ (64 feet) = 640 feet	-60 feet
800 feet	$12 \times$ (64 feet) = 768 feet	-32 feet
900 feet	$14 \times$ (64 feet) = 896 feet	-4 feet
1000 feet	$15 \times$ (64 feet) = 960 feet	-40 feet
1100 feet	$17 \times$ (64 feet) = 1088 feet	-12 feet
1200 feet	$18 \times$ (64 feet) = 1152 feet	-48 feet
1300 feet	$20 \times$ (64 feet) = 1280 feet	-40 feet
1400 feet	$21 \times$ (64 feet) = 1344 feet	-56 feet
1500 feet	$23 \times$ (64 feet) = 1472 feet	-28 feet
1600 feet	$25 \times$ (64 feet) = 1600 feet	0 feet

To decode the encoded value on the receiving end, the decoding algorithm must test whether the encoded value (the binary numeral that is actually transmitted in the message) is a multiple of 25. This probably involves examining the remainder after dividing the encoded value by 25. If the remainder is zero, just shift the encoded value left by 2 bits. Otherwise … well, the algorithm is a bit complex, and it involves performing a division anyway. It would be more efficient to perform that division <u>once</u> per second when assembling the message on the transmitting side, rather than <u>N times</u> per second on the receiving side, when assembling the reports about N transmitting participants.

It's so much simpler if the selected altitude (which is known, after all, to be a multiple of 100 feet) is just encoded by dividing it by 100. Then, on the receiving end, the encoded value is multiplied by 100. That's a very simple algorithm. The corresponding table for that algorithm would look like this:

Selected Altitude from MCP	**Encoded selected altitude**	**Coding Error**
0 feet	$0 \times$ (100 feet) = 0 feet	0 feet
100 feet	$1 \times$ (100 feet) = 100 feet	0 feet
200 feet	$2 \times$ (100 feet) = 200 feet	0 feet
300 feet	$3 \times$ (100 feet) = 300 feet	0 feet
400 feet	$4 \times$ (100 feet) = 400 feet	0 feet
500 feet	$5 \times$ (100 feet) = 500 feet	0 feet
600 feet	$6 \times$ (100 feet) = 600 feet	0 feet
700 feet	$7 \times$ (100 feet) = 700 feet	0 feet
800 feet	$8 \times$ (100 feet) = 800 feet	0 feet
900 feet	$9 \times$ (100 feet) = 900 feet	0 feet
1000 feet	$10 \times$ (100 feet) = 1000 feet	0 feet
1100 feet	$11 \times$ (100 feet) = 1100 feet	0 feet
1200 feet	$12 \times$ (100 feet) = 1200 feet	0 feet
1300 feet	$13 \times$ (100 feet) = 1300 feet	0 feet
1400 feet	$14 \times$ (100 feet) = 1400 feet	0 feet
1500 feet	$15 \times$ (100 feet) = 1500 feet	0 feet
1600 feet	$16 \times$ (100 feet) = 1600 feet	0 feet

Summary

I contend that the "selected altitude" parameter in UAT messages should be encoded in the way the Chris Moody originally proposed: as a binary numeral in which the LSB has a weight of 100 feet.