Hojo's Ham Blog

Sunday, May 15, 2016

Wisp1 Telemetry Revisited

Wisp1 Telemetry Revisited

So, in my previous blog post about telemetry, I outlined a WSPR telemetry scheme to maximize the amount of data packed into the few bits available. Alan Adamson (W7QO) and I were both thinking of using it. However, after some initial implementation, Alan realized it required a hefty bit of coding. He counter-proposed a more "positional" concept. We went back and forth for a day or two, and came up with the scheme below. It's agreeable to both of us, so we each intend to use it on our trackers. Since he did the heavy lifting with most of the design, I'm writing it up for the interwebz to enjoy.

First Packet

KD2EAT FN12 37

The fields include Callsign, Grid locator, and a power level. There are 19 discrete values permitted in in the power level field.

The first modification is to re-purpose the power level field in the WSPR packet as an "Altitude" indicator. Each of the 19 discrete values will represent 1000 meters of altitude. The altitude mapping ends up looking like this:

DBM	Encoded Altitude Meters
0	0
3	1,000
7	2,000
10	3,000
13	4,000
17	5,000
20	6,000
23	7,000
27	8,000
30	9,000
33	10,000
37	11,000
40	12,000
43	13,000
47	14,000
50	15,000
53	16,000
57	17,000
60	18,000

So, the packet above indicates that the balloon is flying at, at least, 11,000 meters. This provides altitude data in one packet with no additional modifications to the protocol, if desired. It provides for altitudes from 0-18,000 meters (59,055 feet).

Second Packet

This is where the rubber meets the road. We are encoding a lot of data in the callsign field, as well as the power field. However, the grid square field is NOT used to carry telemetry data. This means that, though the callsign will be bogus, the telemetry data will appear in the same grid square as the primary packet. It is assumed that the same data will be used to generate BOTH the primary and secondary packet, since the altitude encoding in the primary packet is refined by the telemetry packet.

We'll illustrate the encoding scheme by decoding an example.

Featured Posts

Known Flight IDs

Wisp Flight Summary

About Me

kd2eat

View my complete profile

Blog Archive

QK1SKN FN12 33

The callsign consists of 6 positions with potential values as follows:

Position	Possible Values	Use in Scheme (number of values)	Number of Used Values / Number of Possible Values
Callsign 1	Q,0	Telemetry Channel (2)	2 / 2
Callsign 2	0-9,A-Z	Battery volts (12), Altitude_fine(3)	36 / 36
Callsign 3	0-9	Telemetry Channel (10)	10 / 10
Callsign 4	A-Z	Grid Square 5th char (A-X) (24)	24 / 26
Callsign 5	A-Z	Grid Square 6th char (A-X) (24)	24 / 26
Callsign 6	A-Z, space	Temp (9), Altitude_super_fine(3)	27 / 27
Gridsquare 1	A-R	Same as Packet 1	n/a
Gridsquare 2	A-R	Same as Packet 1	n/a
Gridsquare 3	0-9	Same as Packet 1	n/a
Gridsquare 4	0-9	Same as Packet 1	n/a
DBM	0-18	Solar_volts(6), Sats(3)	18 / 19

Callsign Positions 1 & 3: Telemetry "Flight" or "Channel" number

Positions 1 & 3 allow for 2*10 = 20 possible values. This allows for up to 20 pico flights to be in operation simultaneously without telemetry confusion, provided everyone cooperates and uses a unique pair of characters in those two positions. In the example above, the two telemetry channel bytes are "Q" and "1". We can interpret a "Q" in position 1 as the 10's value, so this is "Flight Number 11" or "Channel 11".

Altitude encoding

Altitude is encoded across both the first and second WSPR packet. The first WSPR packet gives us the altitude with 1 km of granularity, from 0-18,000 meters, as described above in the first packet.

The second packet, gives us two more levels of granularity:

Altitude_fine: 0, 333, 666.
Altitude_super_fine: 0, 111, 222

Balloon altitude is thus calculated by adding the three values together. As we expand the telemetry below, we'll see in this example that:

Altitude_fine = 2, so we add 666 to the altitude.
Altitude_super_fine = 1, so we add another 111 to the altitude.

Actual Altitude = 11,000 (first packet) + 666 (second packet) + 111 (second packet) = 11,777 meters.

Callsign Position 2: Battery volts, and "fine" altitude

We encode volts and altitude fine as:

Volts	Encoded Volts
3.0 or lower	0
3.2	1
3.4	2
3.6	3
3.8	4
4.0	5
4.2	6
4.4	7
4.6	8
4.8	9
5.0	10
5.2 or higher	11

Altitude Fine	Encoded Altitude Fine
0	0
333	1
666	2

Having the two encoded values, we calculate the value for this "character" as follows:

*value = (EncodedVolts * 3) + EncodedAltitudeFine.*

Conversely, to get the Encoded values, you do the following:

EncodedVolts = value / 3
EncodedAltitudeFine = value % 3

The value of the second column is a letter 'A' - 'Z', or number '0' - '9'.

Encoded Value	Letter
0	0
1	1
...	...
9	9
A	10
B	11
...	...
K	20
...	...
Z	36

In our example, we have the letter "K" in position 2, which represents "20".

EncodedVolts = 20 / 3 = 6. So, our **Volts = 4.2v**.

EncodedAltitudeFine = 20 % 3 = 2. So, our **AltitudeFine is 666**. Added to our 11,000 meters from the first packet, our altitude is at least 11,666 feet.

Callsign Position 4&5: Grid Square 5&6.

This is straightforward. We simply take these two characters, and append them to the grid square.

QK1SKN FN12 33

"SK" is added to our grid square "FN12". By convention, the last two characters of the grid square are in lower case, so our **6-character grid square is "FN12sk"**.

Callsign Position 6: Temperature and Altitude "super fine"

This position is permitted the characters 'A' - 'Z', or a space, making 27 total values. We reserve 9 values for Temperature, and 3 values for "super fine" altitude, as follows.

Temperature (c)	Encoded Temperature
-35 or less	0
-30	1
-25	2
-20	3
-15	4
-10	5
-5	6
0	7
5 or more	8

Altitude Super Fine	Encoded Altitude Super Fine
0	0
111	1
222	2

Having the two encoded values, we calculate the value for this "character" as follows:

*value = (EncodedTemperature * 3) + EncodedAltitudeSuperFine.*

Conversely, to get the Encoded values, you do the following:

EncodedTemperature = value / 3
EncodedAltitudeSuperFine = value % 3

The value of the second column is a letter 'A' - 'Z', or a space.

Encoded Value	Letter
A	0
B	1
...	...
N	13
...	...
Z	25
(space)	26

In our example, we have the letter "N" in position 6, which represents "13".

EncodedTemperature = 13 / 3 = 4. So, our **Temperature = -15c**.

EncodedAltitudeSuperFine = 13 % 3 = 1. So, our **AltitudeSuperFine is 111.** Added to our 11,666 meters from above, our altitude is at least **11,777 feet.**

DBM value: Solar Volts, Number of Satellites

The DBM field has 19 potential values, which we consider 0..18 (as in the Altitude table in the first packet above). We use 18 of these 19 values.

We encode 6 values for Solar Voltage, and 3 values for Number of Satellites as follows:

Solar Volts	Encoded Solar Volts
0.2 or lower	0
0.4	1
0.6	2
0.8	3
1.0	4
1.2 or greater	5

Number of Satellites	Encoded Number of Satellites
0 or no fix	0
4 - 7	1
8 or more	2

Having the two encoded values, we calculate the value for this field as follows:

*value = (EncodedSolarVolts * 3) + EncodedSatellites.*

Conversely, to get the Encoded values, you do the following:

EncodedSolarVolts= value / 3
EncodedSatellites = value % 3

We follow the same DBM to Encoded Value scheme as the Altitude does in packet 1.

In our example, we have a DBM of 33, which represents a "value" of 10.

EncodedSolarVolts = 10 / 3 = 3. **Solar Volts = 0.8v.**

Encoded Satellites = 10 % 3 = 1. Satellites = 1. **4-7 satellites.**

Posted by kd2eat at 1:07 PM

1 comment:

Mike August 18, 2021 at 11:00 AM

Hi! Two quick questions:
1. If we reserve one of these "channels", will it disrupt anything to use a different encoding technique for the "telemetry" packet on :02?
2. For the first packet, am I correct in thinking that we could also use the callsign field for encoding here (as long as it's a 0/Q sign)?

Reply

Enter Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)