

Hojo's Ham Blog

Tuesday, March 1, 2016

Wisp1 telemetry

There are several encoding schemes being used, out there, to encode telemetry data into WSPR transmissions. They all involve sending a second WSPR packet, with a bogus callsign. The callsign is legal within the WSPR protocol, but uses characters in some positions which result in illegal ham callsigns. This makes the telemetry packets easily distinguishable in the WSPR stream.

I've devised my own scheme, documented below, which is heavily based on the one used by VE3KCL, which he documented heavily in his [S-4 flight blog](#). My wisp1 telemetry makes several modifications. I'll document them below.

My pico balloon data will be encoded into two WSPR transmissions, which is fairly standard for picos. As a refresher, a typical WSPR transmission might look something like this:

KD2EAT FN12 27

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

Primary packet

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	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 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).

Secondary Packet

Featured Posts

[Known Flight IDs](#)

[Wisp Flight Summary](#)

About Me

kd2eat

[View my complete profile](#)

Blog Archive

- ▶ [2020](#) (1)
- ▶ [2019](#) (3)
- ▶ [2018](#) (3)
- ▶ [2017](#) (15)
- ▼ [2016](#) (42)
 - ▶ [November](#) (1)
 - ▶ [October](#) (5)
 - ▶ [September](#) (5)
 - ▶ [August](#) (6)
 - ▶ [July](#) (5)
 - ▶ [June](#) (3)
 - ▶ [May](#) (3)
 - ▶ [April](#) (4)
 - ▼ [March](#) (3)
 - [I2C design error](#)
 - [Wisp1 assembly and first problem](#)
 - [Wisp1 telemetry](#)
 - ▶ [February](#) (7)

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, in this wisp1 version of telemetry, the grid square 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.

Before going into the telemetry available, let's look at the amount of data we can encode.

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

Position	Possible Values	Use in Wisp1	Number of Values
1	0,Q	Telemetry Flag	2
2	A-Z,0-9	Data	36
3	0-9	Telemetry Flag	10
4	A-Z	Data	26
5	A-Z	Data	26
6	A-Z, space	Data	27

Positions 1 & 3 allow for $2 \times 10 = 20$ possible values. This allows for up to 20 pico flights to be in operation simultaneously without colliding, provided everyone cooperates and uses a unique pair of characters in those two positions.

Positions 2,4,5,6 allow for $36 \times 26 \times 26 \times 27 = 657,072$ possible values.

The "Power" field accounts for another 19 possible values, $19 \times 657,072 = \mathbf{12,484,368}$ possible telemetry values.

Within those possible values, I encode the following parameters:

Telemetry	Possible Values	Granularity	Number of Values
Grid Char 5	A-X	One Character	24
Grid Char 6	A-X	One Character	24
Altitude Granular	0-666	333 meters	3
Temperature	-45c - 5c	5c	11
Lipo Battery	3.2v - 4.8v	0.2v	9
Solar Cell	0 - 1.2v	0.2v	7
Satellites	0-9	1	10

This scheme allows for $24 \times 24 \times 3 \times 11 \times 9 \times 7 \times 10 = \mathbf{11,975,040}$ needed values, which comfortably fits into the 12,484,368 available.

Calculating the value (example)

- Actual telemetry:
 - Callsign: KD2EAT
 - Grid: FN12MX
 - Altitude: 8,500m
 - Temperature: -21c
 - Lipo: 4.35v
 - Solar: 0.8v
 - Satellites: 6
 - Pico tag letters "0" and "9"
- The two WSPR packets for this data would be:
 - KD2EAT FN12 27
 - 0S9SBU FN12 17

How we got Packet 1

- Packet 1 is standard WSPR except the DBM field. 27 DBM encodes the value "8", for "8,000 meters" (See the table above). So we know the balloon is between 8000-9000 meters. We'll get more granularity from packet 2.

How we got Packet 2

- Each telemetry value is converted to an integer. Ex:
 - Grid Char 5: "M" = 12
 - Grid Char 6: "X" = 23
 - Altitude Granular: 8,500 (8000 was in packet 1. 500 remains). Rounds to 666. = 2
 - values: (0=0, 333=1, 666=2)
 - Temperature: -21c. Rounds to -20c. = 5
 - (-45=0, -40=1, -35=2, -30=3, -25=4, -20=5, etc)
 - Lipo: 4.35v. Rounds to 4.4v = 6
 - (3.2=0, 3.4=1, 3.6=2, 3.8=3, 4.0=4, 4.2=5, 4.4=6, 4.6=7, 4.8=8)
 - Solar: 0.89v. Rounds to 0.8v. = 4
 - (0.0=0, 0.2=1, 0.4=2, 0.6=3, 0.8=4, etc)
 - Satellites: 6 = 6
- Making the big number
 - Big number =
 - num = grid5
 - num = num * **24** + grid6
 - num = num * **3** + altitude
 - num = num * **11** + temperature
 - num = num * **9** + lipo
 - num = num * **7** + solar
 - num = num * **10** + satellites
 - For this example, the telemetry would be:
 - 12
 - $12 * 24 + 23 = 311$
 - $311 * 3 + 2 = 935$
 - $935 * 11 + 5 = 10,290$
 - $10,290 * 9 + 6 = 92,616$
 - $92,616 * 7 + 4 = 648,316$
 - $648,316 * 10 + 6 = 6,483,166$
 - 6,483,166

Making WSPR from the telemetry value

- WSPR char 1: "0" or "Q". Hard coded.
- WSPR char 3: "0" - "9". Hard Coded.
- Remaining WSPR Characters, as follows:
 - Num = telemetry number as above.
 - DBM = num % **19**; num = num / **19** // Values 0,3,7,10,13,17,20,23,27,30,33,37
 - WSPR6 = num % **27**; num = num / **27** // Values a-z,space
 - WSPR5 = num % **26**; num = num / **26** // Values a-z
 - WSPR4 = num % **26**; num = num / **26** // Values a-z

- WSPR2 = num (should be 0-35) // Values a-z,0-9

WSPR encoding our telemetry above

- Telemetry value: 6,483,166
- DBM = 6,483,166 % 19 = 5 = "17 DBM"
- WSPR6 = 341,219 % 27 = 20 = "U"
- WSPR5 = 12,637 % 26 = 1 = "B"
- WSPR4 = 486 % 26 = 18 = ""S"
- WSPR2 = 18 = "S"
 - Assuming we chose our hard=coded ID characters as "0" and "9"
 - Assuming Grid is FN12
 - WSPR string:
 - 0S9SBU FN12 17

Checking our work

- Decoding the WSPR
 - Starting value: 0S9SBU FN12 17
 - Pico ID: 09
 - Grid: FN12
 - WSPR2 = "S" = 18; num = 18
 - WSPR4 = "S" = 18; num = 18 * 26 + 18 = 486
 - WSPR5 = "B" = 1; num = 486 * 26 + 1 = 12,637
 - WSPR6 = "U" = 20; num = 12,637 * 27 + 20 = 341,219
 - DBM = "17" = 5; num = 341,219 * 19 + 5 = 6,483,166
- Converting to telemetry
 - 6,483,166 % 10 = 6; num = 6,483,166 / 10; Sats = 6 (6 sats)
 - 648,316 % 7 = 4; num = 648,316 / 7; Solar = 4 (0.8v)
 - 92,616 % 9 = 6; num = 92,616 / 9; Lipo = 6 (4.4v)
 - 10,290 % 11 = 5; num = num / 11; Temp = 5 (-20c)
 - 935 % 3 = 2; num = num / 3; Alt Granular = 2 (666m)
 - 311 % 24 = 23; num = num / 24; Grid6 = 23 = "X"
 - 12; Grid 5 = 12 = "M"

Caveats

- Any temperature warmer than 10c, will be sent as 10c.
- Any temperature colder than -45c will be sent as -45c.
- Battery and Solar voltages are assuming LiPo battery, and solar cells wired in parallel, not series, to keep the voltage low.
- Satellites of "0" will mean NO GPS lock. Satellites of 9 will represent 9 or more.

Differences from VE3KCL's scheme

- The Grid locator is not overlaid, so that the map view on wsprnet.org will show telemetry beside the balloon.
- I use the option of a space in the 6th character of the callsign to squeeze a little more data in.
- I encode altitude data in the first WSPR packet.
- I encode data differently in the second packet.

Next steps

I'd like a few fellow pico/WSPR folks to review this document and provide feedback.

Posted by [kd2eat](#) at [8:33 PM](#)

No comments:

Post a Comment



Enter Comment

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)