

## APRS BASE91 COMMENT TELEMETRY

2012-01-11 Heikki Hannikainen, OH7LZB

### --- 1. Abstract ---

This document specifies a new telemetry transmission format which can be used to transmit machine-readable telemetry in the comment field of an APRS position packet.

A new APRS extension was needed for the following reasons:

- Transmitting telemetry in separate, long packets from mobile and balloon platforms wastes bandwidth and energy.
- The old Mic-E telemetry format was broken by the introduction of new Mic-E Type Codes (<http://www.aprs.org/aprs12/mic-e-types.txt>), as the identifier characters were reused.
- Plain-text human-readable comment telemetry ("12.3V 14C") is very hard for an application to automatically parse and graph.

Design emphasis was put on good compression, enhanced resolution and backward compatibility.

Please read APRS101.PDF chapter 13 ("Telemetry Data") first. This specification only adds a new format used to transmit telemetry values. Otherwise the old specification is still valid.

The protocol has been designed in collaboration by Byon Garrabrant, Bob Bruninga, Lynn W. Deffenbaugh and Heikki Hannikainen during a lengthy email discussion in March, 2010.

### --- 2. Status ---

The extension has been implemented in aprs.fi (within the open-source Ham::APRS::FAP packet parser), APRSIS32 and the TinyTrak3 platforms. It should still be considered experimental in nature. It has not been deployed on many platforms, and all of the platforms are in active development, so the specification can be enhanced further if necessary.

### --- 3. Specification ---

3.1 The new Base91 Comment Telemetry extension MAY appear in the comment field of any of the three position packet formats ("Normal" uncompressed, Mic-E, and compressed).

3.2 The Base91 Telemetry extension, if present, MUST appear after the free-form comment text entered by the end-user, but before any DAO or Mic-E type codes. The DAO extension MAY appear after the Base91 Telemetry extension. When the Mic-E Type Code is used, it must appear in the end of the packet.

3.3 Base91 telemetry is delimited at both ends by the '|' (pipe / vertical bar) character.

3.4 The telemetry sequence counter and telemetry channels are encoded using the Base91 encoding, as it is presently used in Compressed APRS position packets, the altitude extension, and the DAO extension. Two bytes are transmitted for the sequence counter and each of the channels, giving over 13 bits of resolution (values 0 to 8280). Please note that APRS uses a different definition of Base91 than the internet standard Base91 – see APRS101.PDF for details.

3.5 While the Base91 encoding provides more resolution and a larger sequence counter range, the transmitting station may use whatever resolution is available from the sensors. Values of 0 to 255 are fine for 8-bit A/D converters - upscaling to 0...8280 is not necessary.

3.6 The telemetry sequence counter MAY wrap from 255 to 0 if memory constraints require using a 1-byte variable for storing the counter. Please make sure that it and all of the telemetry values never get values higher than 8280. For example, the sequence number can be safely incremented with:

```
sequence = (sequence + 1) & 0x1FFF;
```

This will make it wrap to 0 after 8191, which will provide plenty of range.

3.7 The extension MUST contain at least a sequence counter and one channel of telemetry.

3.8 The extension MAY contain up to 5 channels of "analog values" and one 8-bit channel of "binary values", as in the traditional telemetry format.

3.9 If binary values are transmitted, they MUST appear last in the extension, after all 5 "analog" channels. They are put into a single Base91 encoded integer, where the LSB (least significant bit) corresponds to B1 of the traditional Telemetry specification, the 8th bit corresponds to B8. Bits 9 to 13 are reserved to future use and will not currently be treated as additional binary values.

3.10 Examples of valid Base91 telemetry formats:

```
|ss11|  
|ss112233|  
|ss1122334455!"|
```

Where ss is the sequence counter, 11 is the first channel, and so on. The '!' in the end would be the binary values. These examples, while useful for demonstration, would also parse correctly.

```
Sequence: Base91 'ss' decodes to 7544  
Channel 1: Base91 '11' decodes to 1472  
Channel 2: Base91 '22' decodes to 1564  
Channel 3: Base91 '33' decodes to 1656  
Channel 4: Base91 '44' decodes to 1748  
Channel 5: Base91 '55' decodes to 1840  
Binary values: '!' decodes to decimal 1, binary values 10000000, B1 is 1, B2 to B8 are 0.
```

The following minimal telemetry extension has a sequence number of 0, and Channel 1 value of 0: |!!!!|

3.11 Examples of complete packets containing Base91 telemetry

3.11.1 Mic-E position, 2 channels of Base91 telemetry, !DAO! and Mic-E Type Code:

```
N0CALL>APRS,qAR,IGATE: pZ3l-B]/' "6{ } !9' X$u !wr8! |3  
-----position----- |t1m---| !DAO! |type
```

3.11.2 Compressed position, comment text, and 3 channels of Base91 telemetry:

```
N0CALL>APRS,qAR,IGATE: !/0%3RTh<6>dS_http://aprs.fi/|"p%T'.ag|  
-----position-----comment----- |t1m-----|
```

3.11.3 Uncompressed position, PHG, comment text, and 4 channels of telemetry:

```
N0CALL>APRS,qAR,IGATE: !6304.03NN02739.63E#PHG26303/Siilinjarvi|"p%T'.agff|  
-----position-----PHG-----/comment----- |t1m-----|
```

3.12 The existing method for transmitting coefficients and channel definitions are not changed by this specification. They can still be used in conjunction with Base91 Comment Telemetry. While the format only supports positive integers, negative and decimal values can be displayed with the help of periodically transmitted coefficients, as before.

#### --- 4. Backwards compatibility considerations ---

##### 4.1 Stream switch character |

The APRS specification mentions | as being a forbidden character in APRS packets, due to it's historical use as the stream switch character in TNCs. However in practice, it appears to traverse unharmed on the APRS and APRS-IS networks. It is also used in the Mic-E Type Code for the Byonics TinyTrack family.

The stream switch character was used to address multiple concurrent connections over a serial line between a computer and a TNC. It was only used in CONVers mode (plaintext terminal mode), as opposed to the machine-readable and binary clean KISS mode. Digipeaters run in either KISS mode, or otherwise directly operate on the binary AX.25 frames, and do not process the stream switch character.

Igates using a TNC in TNC2 mode appear to work fine. There is some concern that it could cause issues if an igate in TNC2 mode would retransmit packets in 3rd-party format. However, at the time of writing over 400 trackers transmitting Base91 telemetry have been used in the field. No adverse effects on the network have been reported. '|' usage in the Mic-E Type Code should have brought up any issues early on.

#### 4.2 DAO ambiguity

A valid-looking !DAO! extension may appear in the middle of a Base91 encoded telemetry extension, since some telemetry sequence could be accidentally encoded to look the same.

According to the APRS Client Capabilities Chart maintained by Curt, WE7U, !DAO! decoding is not very widely implemented. It is implemented in APRSIS32 and aprs.fi, both of which can now decode Base91 comment telemetry. The strict order specification in 3.2 allows unambiguous decoding of both extensions.