Trackbox Protocol Specification

Version 1.0

2008-02-05

For firmware version 12

Website: http://www.paralleltrack.co.uk/

Email: mail@paralleltrack.co.uk

Table of Contents

Introduction

The standard operating mode for the Trackbox is to transmit position reports over GPRS to a server application (by default this is the Parallel Track website). However it is recognised that systems integrators / developers may wish to expand on the functionality offered on site or instead write their own custom applications, without the need to start from scratch on the hardware side (please note firmware licensing options are also available – email for details).

To this aim the Trackbox supports setting of the server where position reports are sent, along with 4 different protocols, allowing anyone with a passing familiarity in web application technology, to interface with the device. A sample PHP application is available from the Parallel Track website.

Note: When developing a custom server application it is recommended to uncheck the "Only when moving" option on the unit configuration application so that position reports are constantly sent to the server at the specified interval rather than only sent when movement threshold is exceeded.

Protocols

The choice of protocol will depend on a decision between position report reliability, cost (GPRS traffic), and ease of server side implementation.

All protocols make use of a sliding window position report history held in device memory that may contain up to 24 entries. When the limit is reached due to either a server error condition (HTTP / TBP / GPSGate TCP only) or on GPRS connection loss entries will be deleted in the order they were saved.

HTTP

The HTTP protocol allows the Trackbox to send position reports to any standard web server that can accept "post" data (i.e. the same technology used by standard contact/feedback/comments forms).

It is the simplest to implement assuming a basic level of web application knowledge in e.g. PHP, it is reliable (server must confirm receipt of positions or they will be resent), it is most likely to be supported by the SIM card (will appear as standard web traffic to provider) however depending on GPRS traffic costs it may be more expensive to run as there is the additional overhead of HTTP headers on the traffic across the connection.

With HTTP up to 5 position reports (e.g. from the last 5 minutes when set to a save time of 1 minute and a send time of 5 minutes) may be bundled into a single "post". Since both reports and authentication data are bundled into a single "post", if the server fails to indicate successful reception of the data, all 5 positions will attempt to be resent. If the position history contains more than 5 position reports (e.g. when set to a save time of 1 minute and a send time of 10 minutes), 2 "posts" will be made to the server, one after the other containing 5 position reports each.

A maximum amount of 5 positions are sent at a time in order to minimise GPRS traffic (i.e. cost) if an error condition were to occur on the server causing position report reception not to be confirmed (and otherwise a large build-up of history position reports to be resent each time). During server downtime / maintenance the best policy is to make the HTTP server unreachable to the units (i.e. use iptables to block IP space used by SIM provider), otherwise e.g. HTML error messages may be sent back to the unit resulting in a large increase in GPRS traffic.

Example

>>> Sent from unit to server via HTTP "post"

```
a=connect&v=11&i=111111111111111&d[]=183457.999,5126.0247N,
00002.8686E,5.2,70.4,3,57.63,32.11,17.32,150507,05&d[]=183558.99
9,5126.3979N,00003.0745E,5.2,70.4,3,57.63,32.11,17.32,150507,05
```

See appendix 1 for further information explanation / format of received variables.

<<< Sent from server to unit in HTTP body (of response to HTTP "post")

```
=RECVOK=
```

TBP (Trackbox Protocol)

TBP is the recommended option for implementing a custom application. It is a custom text based protocol running over TCP.   Its main advantage over the HTTP protocol is a reduction in data traffic overheads (no HTTP header sent / received combined with more a more efficient protocol), and therefore also a reduction in costs.

TBP is slightly more complicated to use than HTTP as it requires a custom socket server application listening on a specified port. Administrative / root level server access may be required to install such as application. As with HTTP, TBP is reliable as the server must confirm reception of each position report.

Sending of authentication, and position reports are handled separately in sequence meaning that an authentication failure would not result in any significant increase in GPRS traffic. A server application failure / error condition would also be significantly more likely to result in socket server unavailability (rather than transmission of long HTML error messages to the client as could happen in HTTP).

The server must confirm receipt of each position report separately, and only when it has done so will it be deleted from the device.

Example

>>> Sent from unit to server

```
a=connect&v=11&i=111111111111111\r\n
```

See appendix 1 for further information explanation / format of received variables.

<<< Sent from server to unit

```
=OK=\r\n
```

>>> Sent from unit to server

```
183457.999,5126.0247N,00002.8686E,5.2,70.4,3,57.63,32.11,17.32,1
50507,05\r\n
```

<<< Sent from server to unit

```
=OK=\r\n
```

>>> Sent from unit to server

```
183558.999,5126.3979N,00003.0745E,5.2,70.4,3,57.63,32.11,17.32,1
50507,05\r\n
```

<<< Sent from server to unit

```
=OK=\r\n
```

GPSGate (TCP/UDP)

GPSGate support is included specifically to enable the device to be used with the free online service or downloadable software tracking applications (BuddyTracker and VehicleTracker) from Franson.

However if so desired there is no reason a custom GPSGate protocol server could not be implemented rather than using TBP or HTTP.

The GPSGate UDP protocol has less overhead (than GPSGate TCP) as it does not support position receive confirmation from the server. This means it will be more cost effective (due to less GPRS network traffic) however it will also be less reliable (positions will not be resent if they fail, as there is no way to tell if the position has been successfully received).

Note: GPSGate UDP is supported in beta only.

The GPSGate TCP protocol does not support position receive confirmation on individual position report basis, therefore it may be less reliable than HTTP / TBP in areas with poor GPRS connections.

Please email for further details.

Appendices

Appendix 1 – Variable format

Variable	Description
a	Authentication – the unit password as set via the configuration utility.
v	Version – current installed firmware version (as supplied on purchase).
i	IMEI – Unique 15 digit unit hardware identifier (as supplied on purchase).
d[]	Data – Present in HTTP protocol only, one or more position reports (see appendix 2).

Appendix 2 – Position Report Format

```
183457.999,5126.0247N,00002.8686E,5.2,70.4,3,57.63,32.11,17.32,1
50507,05\r\n
```

Note: \r\n signifies a carriage return, followed by a line feed

Data Example	Data Format	Description
183457.999	`(\d{6}\.\d{3}|)`	UTC Time (hhmmss.zzz)
5126.0247N	`(\d{4}\.\d{4}[NS]|)`	Latitude (ddmm.mmmm)
00002.8686E	`(\d{5}\.\d{4}[EW]|)`	Longitude (dddmm.mmmm)
5.2	`(\d{1,2}\.\d{1}|)`	Horizontal Dilution of Precision
70.4	`((?:\-|)\d+(?:\.\d+|)|)`	Altitude above mean sea level (meters)
3	`([023])`	Fix Type (0: Invalid Fix, 2: 2D Fix, 3: 3D Fix)
57.63	`(\d{1,3}\.\d{1,2}|)`	Course over ground (direction degrees)
32.11	`(\d{1,4}\.\d{1,2}|)`	Speed over ground (Km per hr)
17.32	`(\d{1,4}\.\d{1,2}|)`	Speed over ground (knots)
150507	`(\d{6}|)`	Date of fix (ddmmyy)
05	`(\d{2}|)`	Total satellites in use