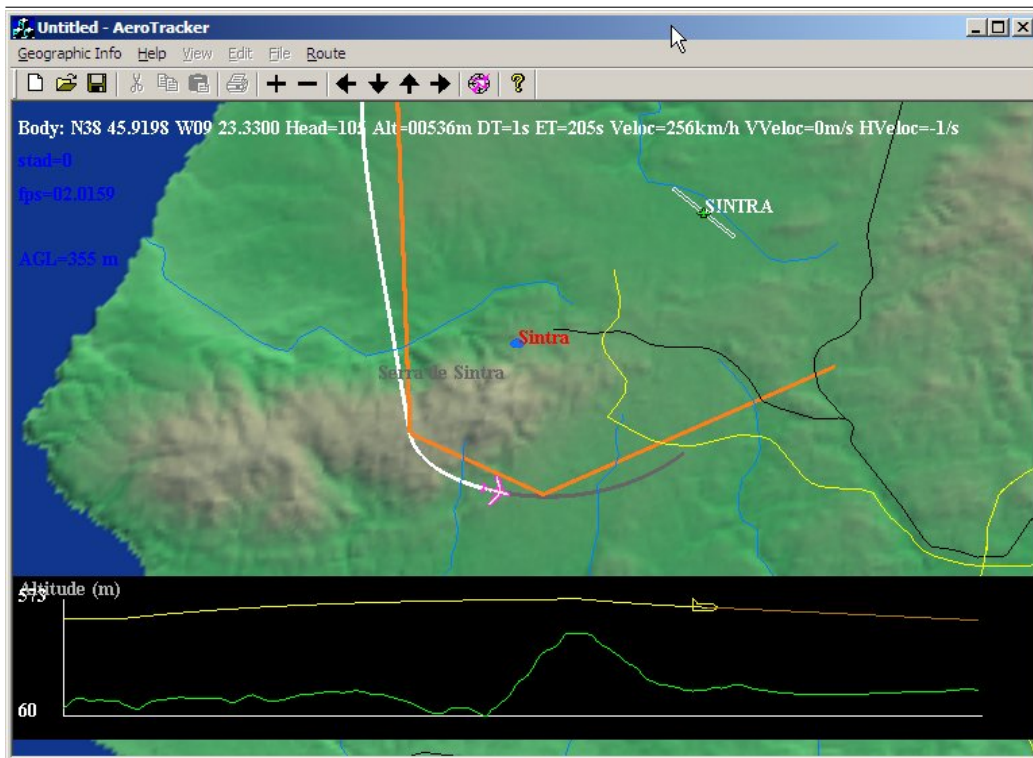




Aerotracker Flight Tracking and Route editing software

Integration Manual



Aerotracker is Copyright Protected.

©David Gil 2005
davidgil@dgadv.com
www.dgadv.com

Chapter 1

Aerotracker Integration Manual

1.1 Introduction

The Aerotracker program is an easy to use and integrate air vehicle terrain aware tracking and route edit software.

1.2 Resume

This Document Aim is to instruct the user to easy insert the Aerotracker Program in his UAV system (or other kind of aerial vehicle). This document has two parts: one for integration of the edited routes in the UAV system and the other part for communication with the UAV in order to show its position on screen.

1.3 Integrating Edited Routes

Aerotracker Saves route on a file with the following format:

lat=+nn.nnnnnn long=+mm.mmmm alt=xxxx.xx veloc=yy.yyyy
lat=+nn.nnnnnn long=+mm.mmmm alt=xxxx.xx veloc=yy.yyyy

Table 1.1: Waypoint Format on the route file

lat=38.890030 long=-9.454003 alt=500.000000 veloc=27.700001
lat=38.866203 long=-9.405392 alt=500.000000 veloc=27.700001
lat=38.852856 long=-9.273853 alt=500.000000 veloc=27.700001
lat=38.893845 long=-9.249071 alt=500.000000 veloc=27.700001
lat=38.924343 long=-9.209991 alt=500.000000 veloc=27.700001
lat=38.885265 long=-9.172817 alt=500.000000 veloc=27.700001
lat=38.857624 long=-9.188068 alt=500.000000 veloc=27.700001
lat=38.828075 long=-9.219522 alt=500.000000 veloc=27.700001
lat=38.812824 long=-9.272901 alt=500.000000 veloc=27.700001
lat=38.821404 long=-9.337716 alt=500.000000 veloc=27.700001
lat=38.834747 long=-9.403485 alt=500.000000 veloc=27.700001
lat=38.864296 long=-9.459723 alt=500.000000 veloc=27.700001
lat=38.889076 long=-9.451144 alt=500.000000 veloc=27.700001

Table 1.2: A Route file Sample

1.3.1 Format Description

Every line represents a waypoint of the route, the first field is latitude, the second longitude (both in degrees). For latitude positive values regard the North hemisphere and for longitude, negative values regard the West hemisphere. The third field is the waypoint altitude (in meter) and the fourth field is the speed (in meter per second).

Please note that the user must use some kind of program to send the route from the file to his airplane.

1.4 Tracking

The Tracking part on the program works by receiving Windows Messages. Despite the fact that in this section everything that is needed to build a Visual C++ application program able of communicating with Aerotracker, it is supplied along with Aerotracker the source code of a program that simulates an UAV that sends his location to Aerotracker. Also more info on this kind of messaging can be found on the [MSDN Library Visual Studio 6.0, Platform SDK](#), chapter [Messages and Message Queues](#) i.e. The Visual Studio Help files.

The messages received by Aerotracker contain the aircraft Latitude, Longitude, Altitude, Course and Time.

In a user made application, the first thing to do is to declare and the register a message identifier.

```
UINT MessieID;
MessieID=RegisterWindowMessage("0xAAAAAh");
```

Table 1.3: VC++ source code to declare and register a Message

Made this, the program will be able to communicate with Aerotracker using the Windows Messaging Functions.

```
PostMessage(HWND_BROADCAST,MessieID,param1,param2 );
```

Table 1.4: A good function to send messages in Windows Messaging

In the function shown in Table 1.4, param1 is an integer witch value changes with the type of information that you wish to send to Aerotracker, this matching can be found on Table 1.5.

param1	Message type
0	Latitude
1	Longitude
2	Not Used
3	Altitude
4	Course
5	Time (in second)

Table 1.5: param1 matching with message type

param2 is an Unsigned Long (ULONG) and it is the actual value to send to Aerotracker after codified. Codification algorithm can be found on table 1.6.

Codification is needed because the value to send must be an unsigned integer, this means that the message sending function does not support negative numbers (so as can be seen on table 1.6, most of the values need an offset value) neither decimal numbers (so most of the values need a gain to increase accuracy).

Message#	Type	param2 value to send
0	Latitude	$(latitude_{UAV} + 90.0) * 1000000$
1	Longitude	$(longitude_{UAV} + 180.0) * 1000000$
2	Not Used	-----
3	Altitude	$(altitude_{UAV} + 100.0) * 100$
4	Rumo	$(course_{UAV} + 400.0) * 100$
5	Tempo	Direct

Table 1.6: Message Codification

To the variable time, no codification is applied because this value is in seconds and AeroTrack does not support non-integer time values. This value is need in critical calculation such as speed and cinematic path predict.

The sent time value, can be preferably the GPS time but also can be the mission time in seconds.

Chapter 2

UAV Simulator

This program aim is to simulate an UAV that follows the open route and generate messages for Aerotracker. His source code is also supplied.

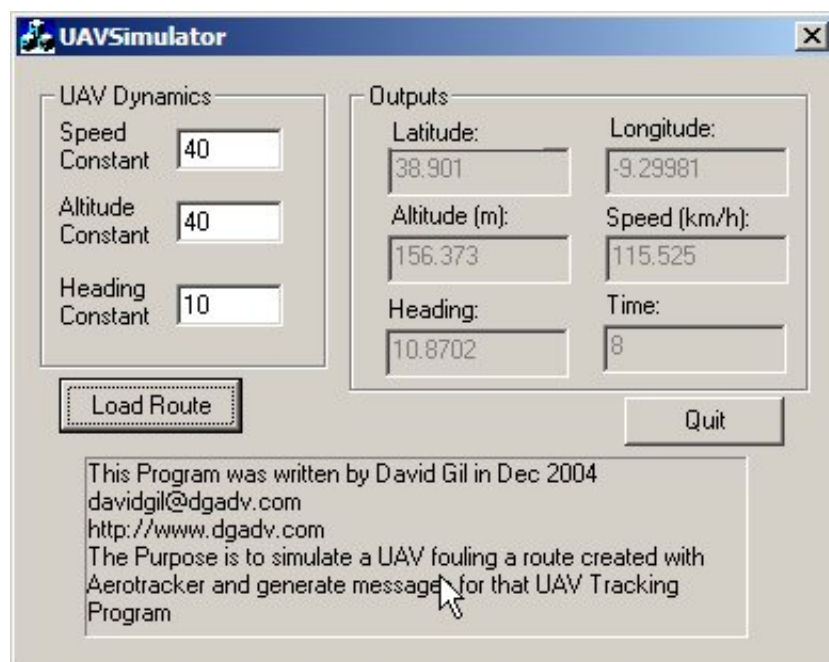


Figure 2.1: *UAV Simulator program Dialog*

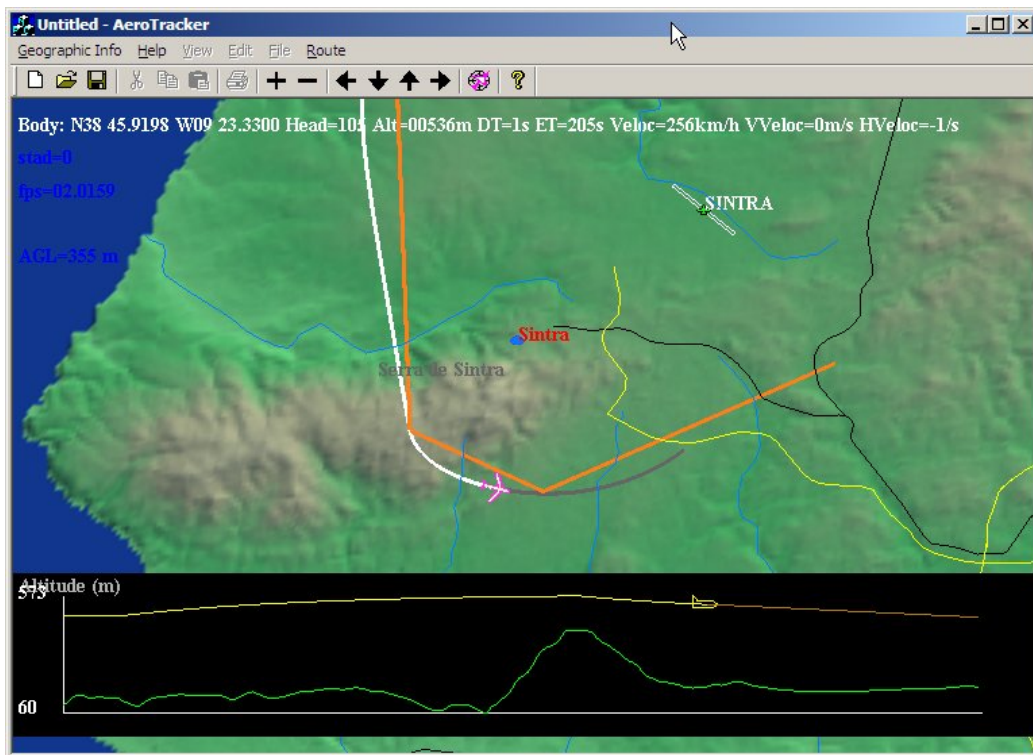


Figure 2.2: *Aerotracker running with UAV Simulator messages*

Chapter 3

Author

Aerotracker is Copyright Protected.
©David Gil 2005
davidgil@dgadv.com
www.dgadv.com