Table of Contents

[**Introduction** 1](#_Toc330929759)

[**Main Windows and Functionality** 1](#_Toc330929760)

[**Setting up a connection** 2](#_Toc330929761)

[Figure 1: Connection Screen 2](#_Toc330929762)

[**Start buffering data sample** 3](#_Toc330929763)

[**Data Item Presence** 4](#_Toc330929764)

[Figure 3: Data Item presences 4](#_Toc330929765)

[**Data Item View** 5](#_Toc330929766)

[Figure 4: Data Item View 5](#_Toc330929767)

[**View by SSR Code** 6](#_Toc330929768)

[Figure 5: View data by SSR code 6](#_Toc330929769)

[**Export** 7](#_Toc330929770)

[Figure 6: Export to Earth Plot format by SSR code 7](#_Toc330929771)

[Figure 8: 3D example in Google Earth of a real test flight over Bosnia and Herzegovina 7](#_Toc330929772)

[**Plot Display** 8](#_Toc330929773)

[**Figure 9: Display of live CAT48 data (Mode A/C and Callsign)** 8](#_Toc330929774)

[**Release Notes** 9](#_Toc330929775)

[Version 1.0 9](#_Toc330929776)

[Version 1.1 10](#_Toc330929777)

[CAT 001, 002, 008 034, 048, 062, 063, 065 10](#_Toc330929778)

[CAT01 10](#_Toc330929779)

[020 Target Report Descriptor 10](#_Toc330929780)

[020 Target Report Descriptor 10](#_Toc330929781)

[Version 1.2 10](#_Toc330929782)

[Version 1.2 10](#_Toc330929783)

[Amer Kapetanovic (akapetanovic@gmail.com) 10](#_Toc330929784)

**Introduction**

ASTERIX Sniffer is a C# /.NET application developed using Microsoft Visual Studio Express 2010. Initially, it started as a small test application intended to read and decode ASTERIX messages and my interest to learn more about ASTERIX message format. Over time I decided to commit more time into it share it with others that might be interested into it.

Please keep in mind that the application is still (*and will always be as new features will be added all the time*) in the development phase and that I provide it as a Microsoft Visual Studio Express 2010 Project bundle. Once downloaded please unzip it into a directory and open it up using MVSE 2010. In order to run you can either make a build or run it in a debugger mode. Please feel free to use the code or expand it as you wish, of course I would appreciate if you send me any additional code so that I can include it into the next release.

**Main Windows and Functionality**

Once started, the ***Main Screen*** opens up and, as you might assume, there are no any data displayed. In order to read in a sample data make sure that the following requirements are met:

PC where ASTERIX Sniffer is running has to be on the same network as a PC, or some other piece of the hardware which is providing the data, is on the same network so that application gets the ASTERIX data via known multicast IP and PORT number.

*In my case, for the testing purpose, I use a host WINDOWS machine and one virtual LINUX machine where an ASTERIX recorded data (I provide CAT48 data sample (****jahorinaJan\_cat.48****) is re-played (****I use gengate provided by Skyguide****) on the same network as my host machine, using the following setup:*

1. *Host PC: 192.168.5.104, 255.255.255.0*
2. *Virtual Linux ASTERIX replay (CentOS): 192.168.5.103, replaying data on 231.27.80.1, port 4001*

*Of course, if available you can connect the HOST PC to any live ASTERIX provider.*

**Setting up a connection**

To start, from the Main Screen (Figure 2) open up ***Settings -> Connection Settings*** (Figure 1), and enter required data. It is possible to enter several connections and save them in a file. Later on, you can open up the file and just activate one of the saved connections.

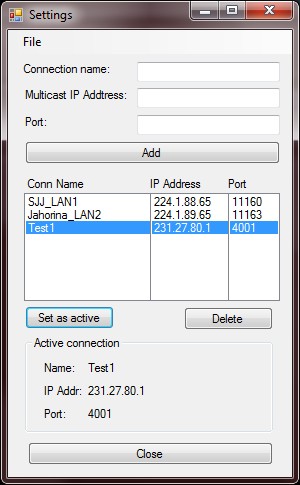
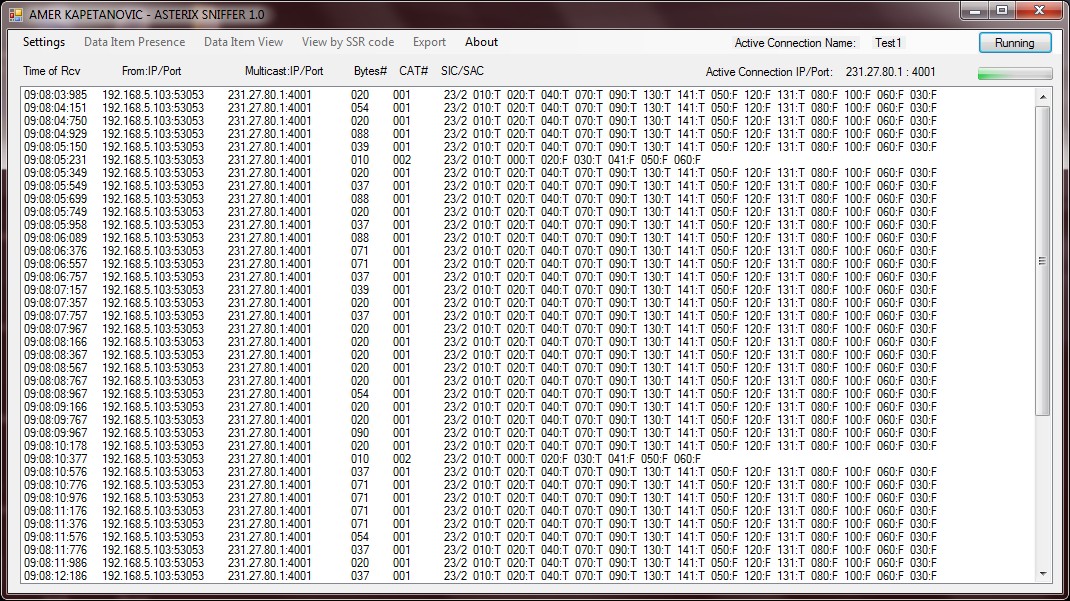


Figure 1: Connection Screen

**Start buffering data sample**

Once a connection is activated to start buffering the data it is necessary to enable it from the Main Screen using the upper right Button( ***Stopped/Running***). Once the buffering is activated the screen will start to populate in real time with the buffered data as shown in Figure 2.

Figure 2: Main Window

Once some data is buffered stop the buffering and then you can analyze the data either by looking at the Main Screen or by one of the below listed options:

**Data Item Presence**

This view will tell you what data items were detected for a given message category for the latest buffered data sample.

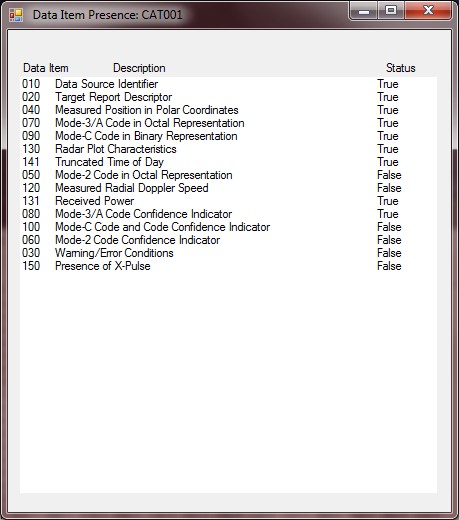
****

Figure 3: Data Item presences

So far the data item presence for the following ASTERIX categories is implemented:

1. CAT 001
2. CAT 002
3. CAT 008
4. CAT 034
5. CAT 048
6. CAT 062
7. CAT 063
8. CAT 065

**Data Item View**

This view will list you all the given data items in the order they were received for the given data sample. It does not filter the data so for CAT001 and CAT048 it is much better to use “View by SSR Ciode”.

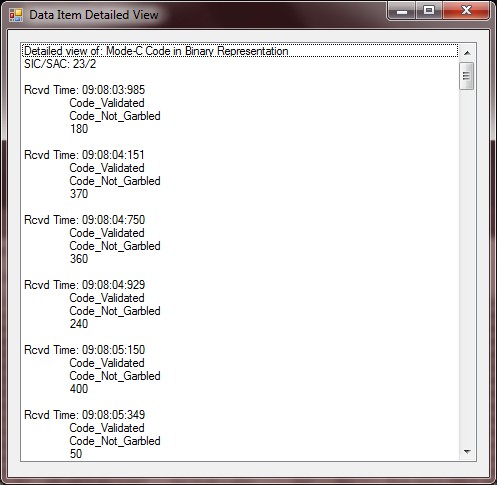
****

Figure 4: Data Item View

**View by SSR Code**

This view provides to the following data (CAT 001 or CAT048), filtered by SSR code in the order received:

1. ***Distance from the surveillance source***
2. ***Azimuth from the surveillance source***
3. ***Lat/Long from the surveillance source***
4. ***Mode C code Validated (TRUE/FALSE)***
5. ***Mode C code Garbled (TRUE/FALSE)***
6. ***Mode C code value***

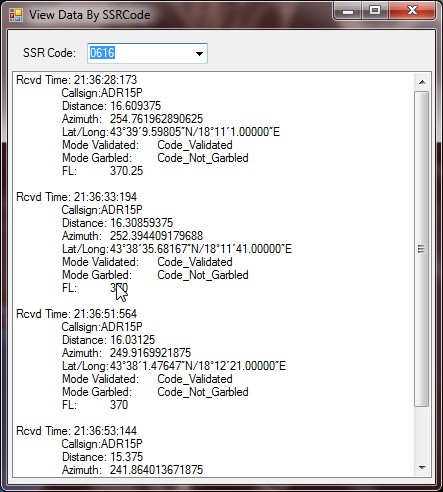
****

Figure 5: View data by SSR code

**Export**

This option enables you to decode and export a data sample, filtered by a SSR code, to either Earth Plot or GE Path supported file formats that then can be used to export data to a KML file, used by Google Earth. The final result is possibility to display any track in the data sample as a 3D track in Google Earth, as shown in Figure X.

****

Figure 6: Export to Earth Plot format by SSR code

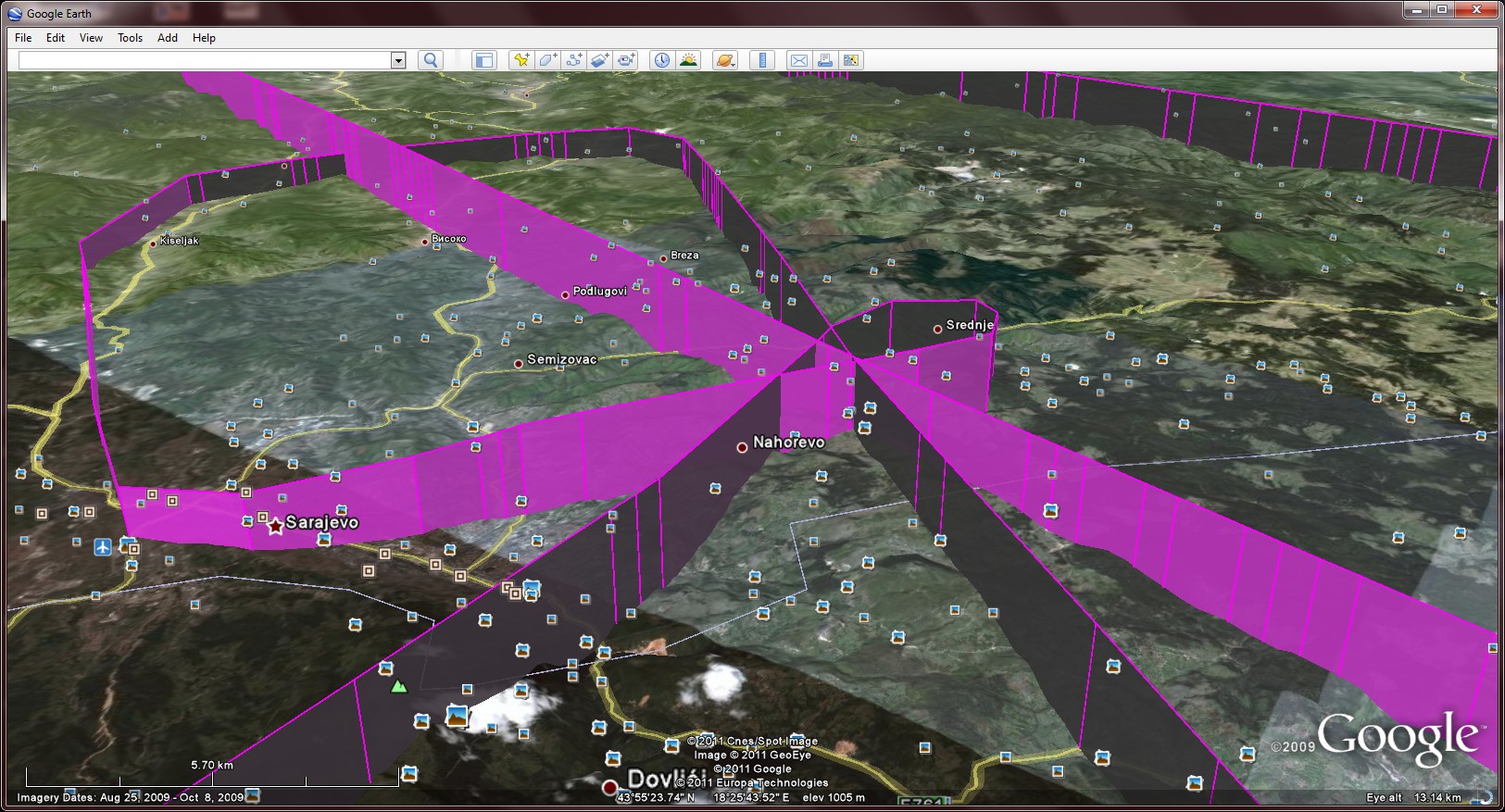
****

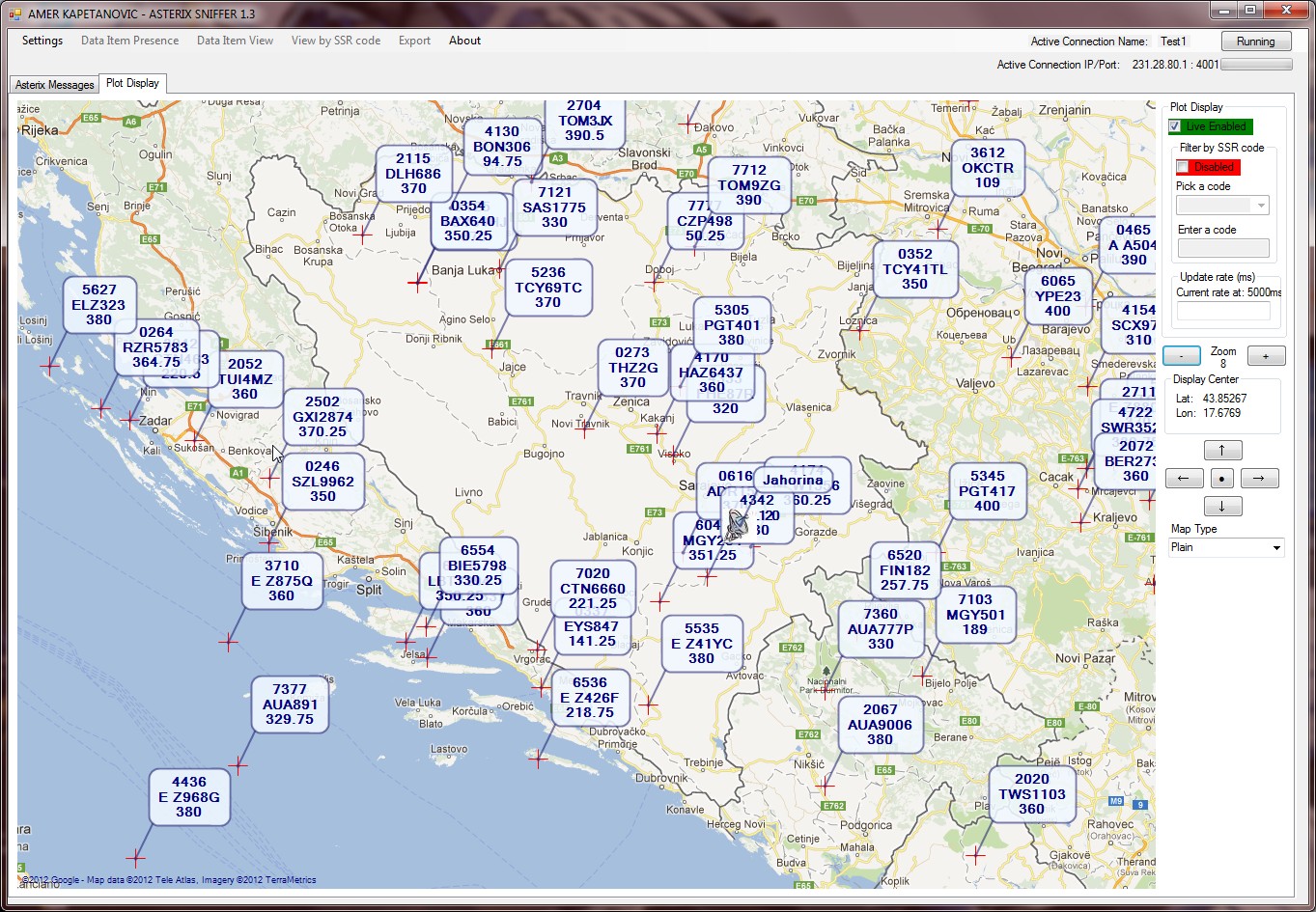
Figure 8: 3D example in Google Earth of a real test flight over Bosnia and Herzegovina

**Plot Display**

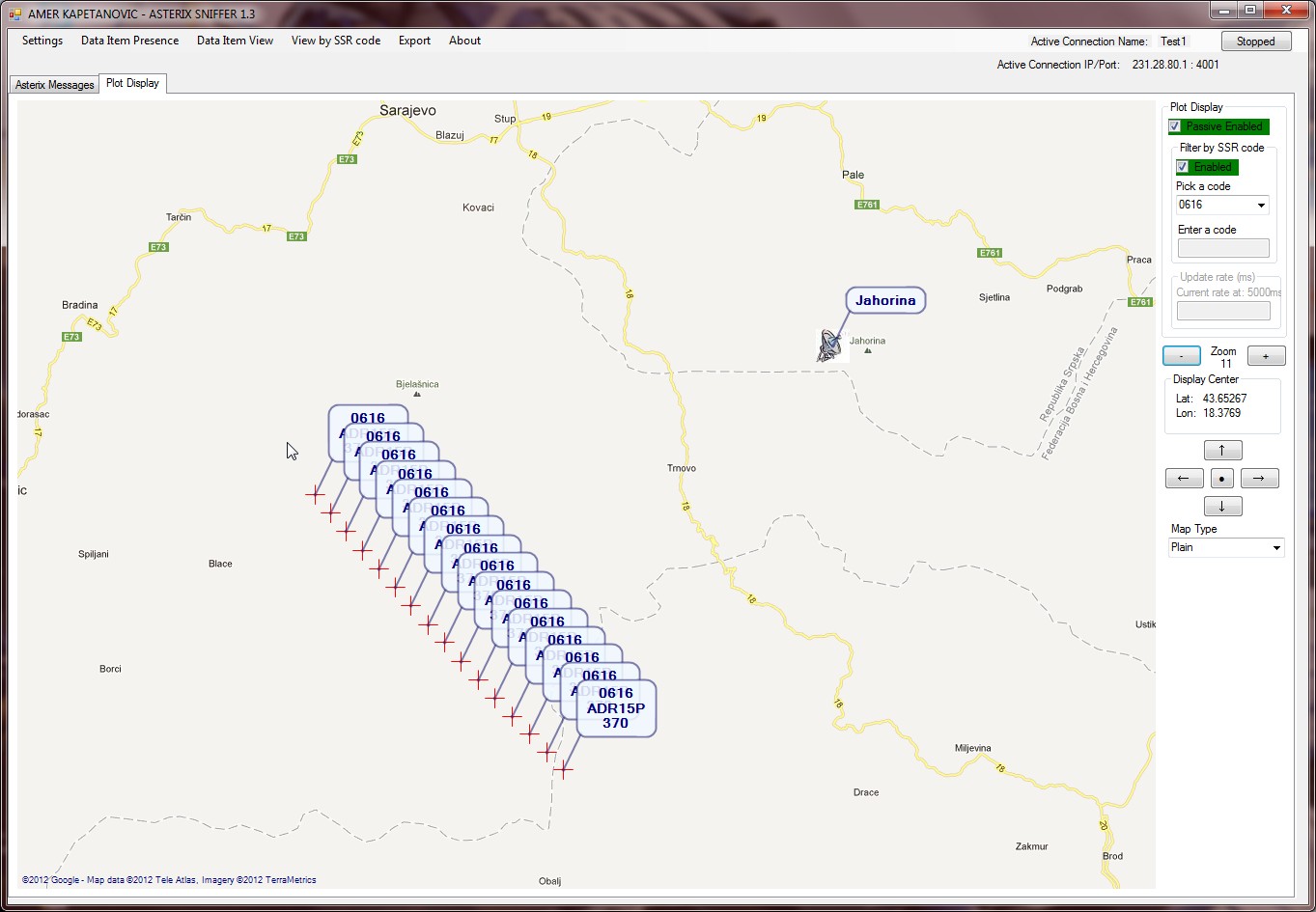
Plot display functionality enables you to display received plots either in real time or buffered data. It provides filter capability (by call sign) as well as to dynamically adjust update rate in order to match antenna time period of a specific radar.

***Please note:*** Application uses WEB map providers (google, yahoo, etc..) so first time application is powered it is necessary to be on the internet so application is able to cache maps. While on the net zoom in so appropriate maps are downloaded. Later on maps will be cached and application can be used offline.

Additionally, center is hard coded to the center of Bosnia and Herzegovina. Should you want app to start at a different location pleaser modify gMapControl.Position = new PointLatLng(44.05267, 17.6769); in FrmMain.cs to a desired location and make a new build.

****

**Figure 9: Display of live CAT48 data (Mode A/C and Callsign)**



**Release Notes**

Version 1.0

1. This is the first release with a limited message categories implemented so far. Each new release will be implemented additional categories (those planned already included in SW, but disabled). It is planned that in the next release the documentation will also be updated to keep detailed track of the implemented vs. planned functionality.
2. Currently there is a significant drop of the messages as I do not check the total size of the received buffer returned from ***sock.Receive***… against the total size of an individual ASTERIX message, resulting in only one message being extracted from each return of ***sock.Receive***…. This, for sure, will get fixed in the next release.

Version 1.1

1. So far the application can detect the presence of the following messages and its associated data items:

CAT 001, 002, 008 034, 048, 062, 063, 065

1. So far the following data item decoders are implemented:

CAT01

020 Target Report Descriptor

040 Measured Position in Polar Coordinates

070 Mode-3/A Code in Octal Representation

090 Mode-C Code in Binary Representation

CAT48

020 Target Report Descriptor

040 Measured Position in Slant Polar Coordinates

070 Mode-3/A Code in Octal Representation

090 Flight Level in Binary Representation

1. This version also fixes the original issue of dropped messages in the case a data block contains more than one message.

Version 1.2

1. This release implements Plot/Track display functionality that enables user to see the see plots/tracks in either real time or to buffer the data and display all the received plots/tracks reports. In addition to that it enables you to use filter (By SSR code), so that only specific aircrafts are displayed. Lastly the display update rate can be controlled as well. Currently the application is hard coded to use Google maps, but any other internet map provider can be used. Further updates will include more user friendly versions. The map zoom is controlled using mouse middle button.

Version 1.2

Amer Kapetanovic (akapetanovic@gmail.com)