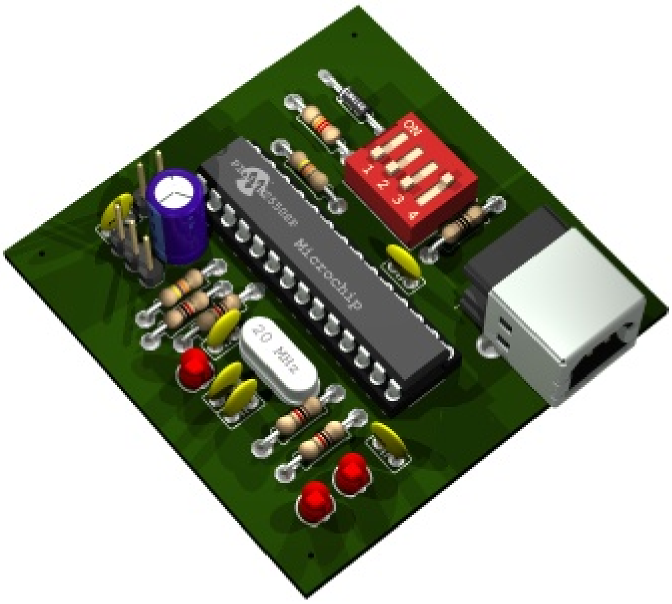
Manual for the adsbScope Software V2.6-test-6 & adsbPIC Decoder V2 Fw11



Author: sprut
Date: 12.11.2012

1 Table of contents

2 List of Figures

3 TERMS OF USE:

THIS SOFTWARE CAN BE USED WITHOUT PAYING ANY LICENCE FEE FOR PRIVATE AND COMMERCIAL USE. THIS IS BETA-SOFTWARE. IF THE SOFTWARE HAS LEFT BETA TEST, IT WILL BE PUBLISHED UNDER GPL-LICENCE.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

4 Introduction

This handbook is an early draft probably full of typing errors and other mistakes. Normally I would not publish it in this bad condition, but I like to give the user something into the hands for the first step with my decoder and PC-software. The handbook will be updated and “debugged” continuously to improve its quality.

The Handbook is based on:
- adsbScope Version 2.6 t 6
- Decoder adsbPIC hardware-version 2, firmware 11

An ADS-B-receive system can be used as virtual radar to generate a live picture of the air traffic.
It is made up from four stages:
- antenna
- receiver (frontend)
- decoder
- PC-software

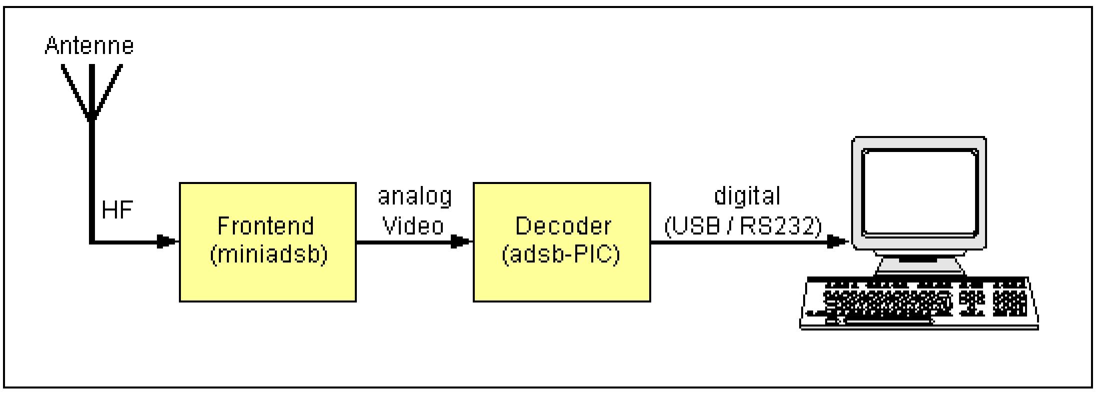


Figure 1 ADS-B receiver

1. The antenna has to receive the 1090 MHz radiation and to convert it into an electric HF-signal. (HF=high frequency = radio frequency)
2. The receiver selects, amplifies and demodulates the received signals, and generates an analog video signal.
3. The decoder converts analog video into digital video and detects ADS-B transmissions inside digital video. The ADS-B signals are then forwarded to a PC via USB (or RS232) connection.
4. The PC decodes the ADS-B information and generates the virtual radar display. It may support the exchange of ADS-B-data via the internet.

For all 4 stages multiple solutions are in use. I use the miniadsb-receiver together with a self made decoder **adsbPIC** and self made PC-software **adsbScope**.

This handbook is focused on the decoder **adsbPIC** and the PC-software **adsbScope**.

5 Hardware

To receive ADS-B-information one needs beside a standard PC

- an antenna
- a receiver and
- a decoder

5.1 Antenna

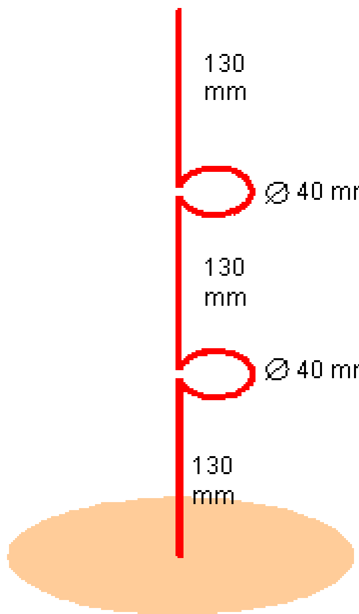


Figure 2 Antenna

The antenna has to be an omnidirectional, vertical polarized antenna with 50 Ohms impedance, tuned to 1090 MHz. Several designs are discussed in the internet.
Look at:
http://www.airnavsystems.com/forum/index.php?topic=2226.0
http://www.kineticavionics.co.uk/forums/viewtopic.php?

f=1&t=7489&sid=a97d41478e66511dd5cf715ee7c88df0&start=24

For first tests a simple 13-cm-long vertical wire may do the job.

A more capable solution is a serial connection of several halve-wavelength-dipoles with halve-wavelength delay lines in between. (see figure above)

The antenna has to be placed as high as possible at a location with a good ‘view’ into all directions.

5.2 Miniadsb-Receiver

The miniadsb is a small and simple direct detection receiver for 1090 MHz. PCB, parts and housing can be ordered as kit for 49 Euro. It is using SMD technology, thus assembling is not easy for inexperienced people.
http://miniadsb.web99.de/

Its availability and low costs made this receiver to a standard for ADS-B-hobbyists. Even the internal receivers of microADSB, GNS5890 or Beast are based on the same design.
The antenna-input is a BNC connector. The receiver has to be fed via 50 Ohms coaxial cable. The connections to the decoder are 3 wires:

- supply voltage (+4V)
- ground
- analog video

The receiver needs +4V supply voltage. I suggest putting a silicon diode in line with the power supply wire, for safety reasons. This diode increases the necessary supply voltage to +4,5V.

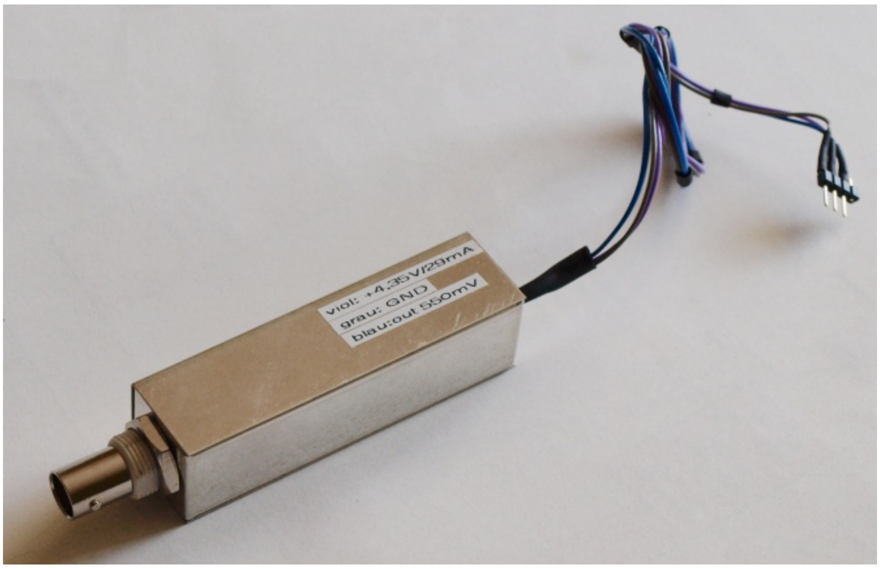


Figure 3 miniadsb receiver

The analog video output is sensitive. If by accident a voltage is fed into this pin, then the receiver will be badly damaged. Be careful.

The receiver is known to generate bad output signals for strong RF-signals. Consequently the reception of close targets may be difficult or impossible.

A perfectly assembled miniadsb-receiver can have the double frame rate of a quick-and dirty assembled receiver. I suggest the study of the tips for optimization in the miniADSB-forum.

5.3 adsbPIC-Decoder

The decoder converts the analog signal (from the receiver) into a digital signal, detects the ADS-B-frames in the signal and sends them to the PC

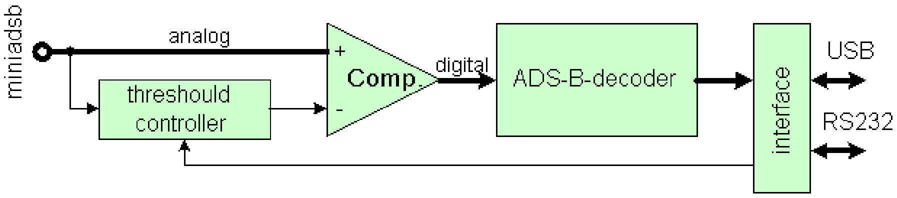


Figure 4 decoder block diagram

The heart piece of the adsbPIC-decoder is a PIC18F2550 (or PIC18F2455) microcontroller. It converts the analog video signal into a digital video signal by use of an internal comparator. Then it tries to detect ADS-B messages inside the digital video.
The received ADS-B messages are then transferred by USB to the PC.

The schematic of the decoder is attached to the end of this document. Firmware and Bootloader for the microcontroller are contained in the ZIP-File.

CAD files for printed boards are contained in the ZIP-file as well. The decoder-hardware is simple and inexpensive. It can be realized in modern SMD-technology or with manual wiring on a universal test board.

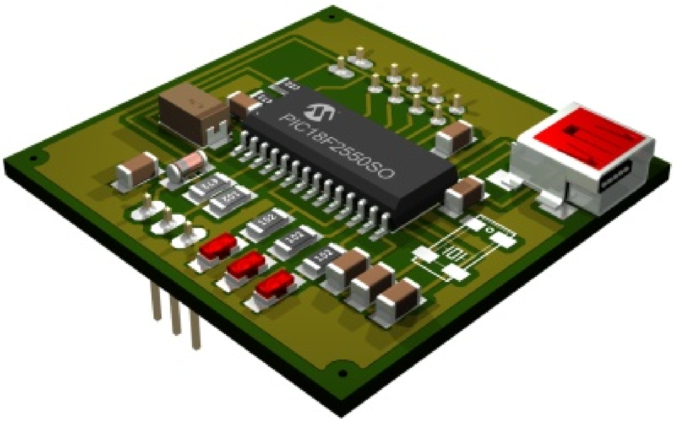


Figure 5 SMD-layout of the decoder

Assemble the decoder, burn the bootloader into the PIC18F2550 (or PIC18F2455), plug the PIC into the decoder-board and connect the decoder via USB with the PC. By the help of the USBoot software (read chapter Bootloader) the firmware is flashed into the decoder, and the decoder is ready for use.

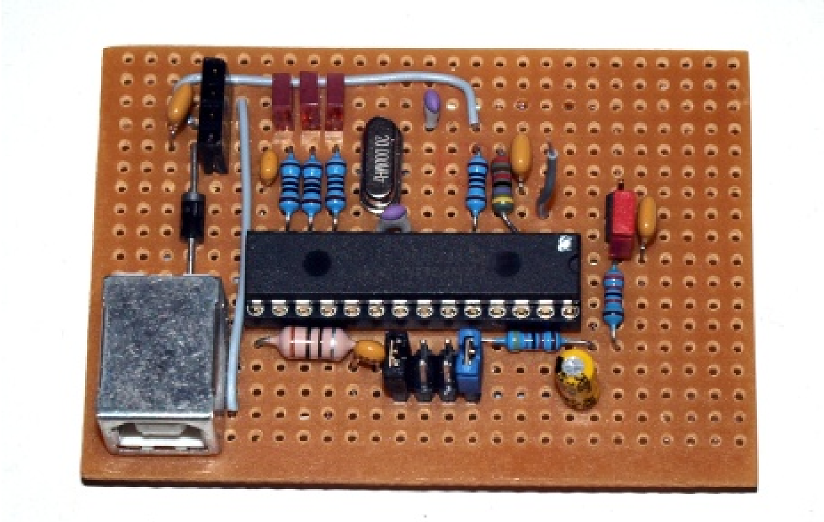


Figure 6 Decoder Prototype

5.3.1 Driver

The decoder needs Windows-drivers. The microchip-MDC-driver is required for the bootloader and the microchip-CDC-driver for the normal operation. Both drivers are included in the ZIP-file.

5.3.2 Input stage

The decoder can use analog or digital input signals.

5.3.2.1 Analog Input

The analog signal from the **miniadsb**-receiver is the normal input signal for the decoder. This signal is converted into a digital signal by the comparator of the PIC-microcontroller. The comparator-inputs are the Pins 2 and 5 of the PIC. Pin 5 is connected to the analog signal from the receiver. For correct function a reference voltage (that is some 10mV above the mean analog voltage level) has to be applied to Pin 2. This voltage is automatically generated by help of the PWM-hardware of the PIC.

The **absbScope** software can measure the analog input signal voltage level and reference voltage level. The output of the comparator is Pin 6. There is now the digital signal available. It is fed into the digital signal input of the PIC-microcontroller at Pin 11.

The LED 3 is connected to the digital signal. If the miniadsb-receiver is connected to the decoder and the reference voltage has stabilized (needs some seconds), then LED 3 just starts to smoulder a little bit. Do not decrease the value of R5!

5.3.2.2 Digital input

If a receiver with digital signal output is used, then the analog comparator of the decoder is not needed. In this case open the connection between Pin 6 and 11 of the PIC and feed the digital signal directly into Pin 11.

If the digital input is used, then the reference-voltage-regulation can be disabled via menu point **adsbPIC - Uref-offset - off** of the adsbScope -software.

5.3.2.3 Switches / Jumpers

The decoder was designed to be remotely controlled by the software adsbScope or (with limitations) Planeplotter. If alternate software is used then switches or jumpers have to be used to control the decoder instead.

The decoder-layout has by default 4 switches (or jumpers) to control the work of the decoder. The switches are connected to the pins RB0, RB1, RB2 and RB4 and can connect them to ground (Vss). For additional (less important) functions switches (or jumpers) can be connected to RB5 and RB6.

With the introduction of Fw 8 the function of Switch 2 was changed.

Switches for firmware 1 .. 7 (outdated)

	PIC-Pin	**Port-Pin**	**Function**	Default
Switch 4	21	RB0	remote	closed
Switch 3	22	RB1	RS232	**open**
Switch 2	23	RB2	CRC	**open**
Switch 1	**25**	**RB4**	**DF17**	**open**

Switches for firmware 8 and later

	PIC-Pin	**Port-Pin**	**Function**	Default
Switch 4	21	RB0	remote	closed
Switch 3	22	RB1	RS232	**open**
Switch 2	23	RB2	TimeTAG	**open**
Switch 1	25	RB4	DF17	**open**
Switch A	26	RB5	CRC	**open**
Switch B	**27**	**RB6**	**1 MBit**	**open**

5.3.2.3.1 Switch 4 - Remote control

If switch 4 is **closed**, then the decoder has to be remote controlled by the adsbScope software. The switches 1, A and B will have no effect. After power-on or reset, the decoder will not start to send data by itself; it will wait for a command from the software.

If switch 4 is open, then the decoder starts to send data immediately after power-on or reset. Which data is transferred and if CRC check is done depends on switch 1, 2 and A.

If you use **adsbScope** or the latest version of **Planeplotter,** then keep switch 4 always closed, keep all other switches always open and skip the rest of this chapter.

But if you like to use an old version of Planeplotter or other PC-software, then you should close the switch before you connect the decoder to the PC and start the software. After the software is running and waits for data, then you can open the switch to start data transfer to Planeplotter.

5.3.2.3.2 Switch 3 - RS232

Since firmware 6 the decoder supports old-fashioned RS232 interface beside the more advanced USB-interface. Which interface is used depends on the switch 3 position at power-on or reset.
If the switch 3 is open, then the USB-interface will be used. But if the switch 3 is closed, then the RS232-Interface will be used instead.
RS232 requires additional hardware and causes a reduced frame rate in comparison to USB.

5.3.2.3.3 Switch 2 - TimeTAG

If this switch is closed, then a precise time code (for MLAT) will be generated by the decoder for each received frame. The adsbPIC-decoder time tag is based on a 12 MHz clock. (Some other decoders/receivers use a 20 MHz clock instead.)
The adsbScope software can activate the time tags by software. If this was done then, then the decoder will send time tags even if the switch is in open position.
The time tag can be used to triangulate aircraft position in collaboration with other decoders. If you don’t participate in such a network, then you should deactivate the time tags to reduce the load at the USB/RS232-Interface.

5.3.2.3.4 Switch 1 - DF17

This switch has no function, if switch 4 is closed.
If switch 4 is open, then this switch controls which frames are reported to the PC. As long switch 2 is open, all received frames will be reported to the PC. But if switch 2 is closed, then only DF17, DF18 and DF19-frames will be transferred.

5.3.2.3.5 Switch A - CRC

This switch has no function, if switch 4 is closed.
If switch 4 is open, then this switch controls CRC checks at received frames.
If the switch 1 is closed, then only DF17, DF18 and DF19-frames with correct check sum will be transferred to the PC. In my opinion this is a waste of decoder-processing-power. I suggest not using this function, because the PC-software can do the same test much faster, adsbScope will do a CRC-check anyway, and the framerate would drop down with the switch closed.

5.3.2.3.6 Switch B - 1MBit

This switch has no function, if switch 4 is closed.
The decoder contains a RS232-Interface. Its default speed is 115 kBaud. If the RS232-Interface was selected (switch 3 closed or by adsbScope) and switch B is closed during power-on or reset, then 1 MBit is used as RS232 baud rate.

5.3.2.3.7 adsbScope

If the decoder is in use with the adsbScope-software, then the switch 4 (jumper 4, connects RB0 to Vss) has to be closed before the decoder is connected to the PC. Other switches are not necessary and switch 4 stays always in closed position, because adsbScope can control the decoder via USB (or RS232).

5.3.2.3.8 Planeplotter

The decoder is optimized for the use with **miniadsb**-receiver and **adsbScope**-software. However, the use in combination with other receivers and the Planeplotter -software is possible.

To use the decoder with Planeplotter the switch 4 (jumper 4, connects RB0 to Vss) has to be closed before the decoder is connected to the PC. Now connect the decoder to the PC and start Planeplotter. As Mode-S-Receiver one hast to choose "**AVR receiver**". Select the right COM-port for the **AVR receiver** and activate the processing of the software with a click on the button with the green circle.

Planeplotter will send the correct command to the decoder to start data processing and to send RAW-data (with or without time-TAG) to the PC. All other control functions (change of comparator-offset, DF17-only, CRC-test, RS232-setup) are not supported by this software. If you like to change the comparator-offset voltage or the RS232-setting, then you will have to use adsbScope to do this. However, the decoder will store these changes and use the changed values even if you later use Planeplotter again.

5.3.3 RS232

I started to work at my decoder design, because I hated the use of RS232 and RS232-USB-bridge chips on other decoders. I used the internal capabilities of the PIC-microcontrollers consequently to reduce complexity and costs of the decoder.

Thus I was really unhappy, as multiple users asked me for a decoder-version with RS232 support. Since firmware 6 the decoder supports an optional RS232-interface beside the USB-Interface. However, I suggest to all users to use the USB-interface! Please use the RS232-interface only, if you have really a need for RS232. Under which circumstances should the use of RS232 made sense?

<u>Example 1</u>
You like to use the decoder together with a special hardware e.g. a LAN-bridge. If this special hardware supports RS232 only, then you may use it.

<u>Example 2</u>
If the decoder is located far away from the PC, then RS232 may be used because it can use longer cables then USB. But in this case the user would have to use special drivers. The guarantied cable length for RS232 at 115200 baud is not more then 2 meters!

5.3.3.1 Requirement

<u>Firmware:</u>
The RS232-interface can only work, if the bootloader is programmed into the decoder! Because of this, you will have to install the USB-connector on the decoder and to use the USB-interface to load the firmware into the decoder.
That means: if you like to use the RS232-Interface, then you will have to install the USB-Hardware (connector) of the decoder too!

<u>Hardware:</u>
Early production samples of PIC18F2455 and PIC18F2550 had a long list of bugs and peculiarities. One if this is related to the RS232 hardware of this PIC. Firmware 7 (and later) can not use the RS232-interface at these obsolete chips. The problematic chips where produced until 2005. Microchip-programmers identify these chips as "**Revision A3**". My programmers call them "**Revision 0x02**". In all later produced batches this problem was fixed. If one buys today a PIC18F2455/2550, then one

should get a modern revision without this bug. But if you like to use an old PIC from your private stockpile to use RS232, then check the revision code.

If the USB-interface is not used, then the decoder needs a separate power supply. A DC-voltage of 5 V (±5%) has to be fed into the Vdd/Vcc terminal. The RS232-interface schematic contains a connector for the power supply. **A supply-voltage of to high voltage level (>7V) or wrong polarity will destroy the decoder and the miniadsb-receiver!**

<u>Operation:</u>
To use the RS232-interface the user has first to close the switch 3 of the decoder and then to connect the decoder to the 5V-power-supply.

The default setting of the RS232-interface is:

- Baud rate 115200
- Data bits 8
- Stop bits 1
- Parity none
- Flow control none

For adsbScope this is the default setting.
The baud rate can be changed to 1 Mbit or 19.2 Kbit.

5.3.3.2 RS232-Speed

By default a speed of 115 Kbit is used by the decoder. The speed can be increased to **1 Mbit**. (The correct value is 921600 baud, but people are lazy.)

The speed can be decreased to **19.2 Kbit**. This may be helpful, if the decoder is separated from the PC and a long RS232-cable has to be used. It sometimes makes sense to place receiver and decoder close to the antenna, and this may be fare away from the PC. A reliable communication via long RS232-cable requires a low datarate. A speed of 115 kbit is only good for 2 meters of cable, while 19.2 kbit works at 15 meter long standard RS232-cable.
However, the low datarate will have bad influence on the ADS-B-framerate.

To change the RS232-speed use the menu point **adsbPIC - RS232-speed**. The change becomes efficient at the next reset or power-up of the decoder. Don’t forget to change the datarate in adsbScope too. This is done by the menu point **decoder - set custom baud rate**.

The alternate way to speed up the RS232-interface is switch B at the decoder. This switch is not contained in my schematics or layout, but can be added easily. Just add a switch or jumper to connect Pin 27 (RB6) of the PIC-microcontroller with Vss (ground). If the switch is closed during power-up or reset, then the RS232-interface will use 1Mbit.

5.3.3.3 RS232-Polarity

The decoder needs an external RS232-driver to generate correct signal- voltage levels. The RS232-specification requires +12V for a logic 0 and -12V for a logic 1.

External driver-chips **invert** and amplify the decoder-signals to this voltage level. Thus the decoder generates 0V for logic 0 and 5V for logic 1 and feeds this into an external driver chip.
Many modern RS232-receive-interfaces do not really need the high signal voltage levels. They accept +5V instead of +12V for logic 0 and 0V instead -12V for logic 1. But these are inverse signals compared to the typical decoder-output voltages.

The decoder can generate this inverse signals instead of the default levels. To change the RS232-polarity use the menu point **adsbPIC - RS232-polarity**. The change becomes efficient at the next reset or power-up of the decoder.

The default setting is **with driver**. If this is selected, then an external RS232-driver has to be used.
The alternate setting is **without driver**. If this is selected, then the Pins 1 and 2 of the RS232-connector of the decoder can be connected to the RS232-interface of the Computer. But for safety reasons a 22 kOhm resistor has to be used between Pin 2 of the decoder-RS232-connector and the TX-Pin at the computer. Only low data rate (115 Kbit or 19.2 Kbit) and a short RS232-cable should be used in this mode. The function of the RS232-interface without driver can not be guarantied, but there is a good chance.
I tested the driverless interface successfully at an Asus P8P67 mainboard (Sandybridge) with a 1.5 meter long cable at 115 kBaud.

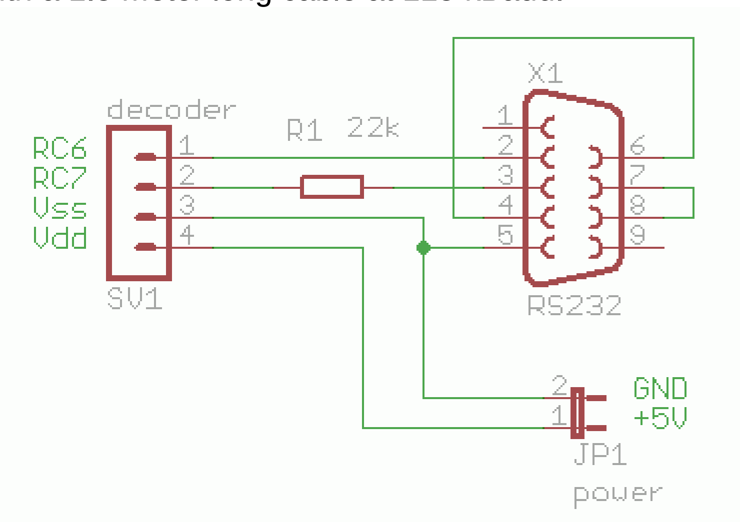


Figure 7 RS232 without driver

5.3.3.4 RS232-Driver-Hardware

In the most cases for RS232 you will need additional hardware. A MAX232-chip (or a comparable RS232 driver chip) has to be used to generate the necessary voltage levels. This interface has to be connected to RC6, RC7 Vss and Vdd of the decoder.

If the USB-interface is not in use, then the decoder needs a separate power supply. A DC-voltage of 5 V (±5%) has to be fed into the Vdd/Vcc terminal. The RS232-interface schematic contains a connector for the power supply. The ZIP-file contains special CAD-files for RS232-users.

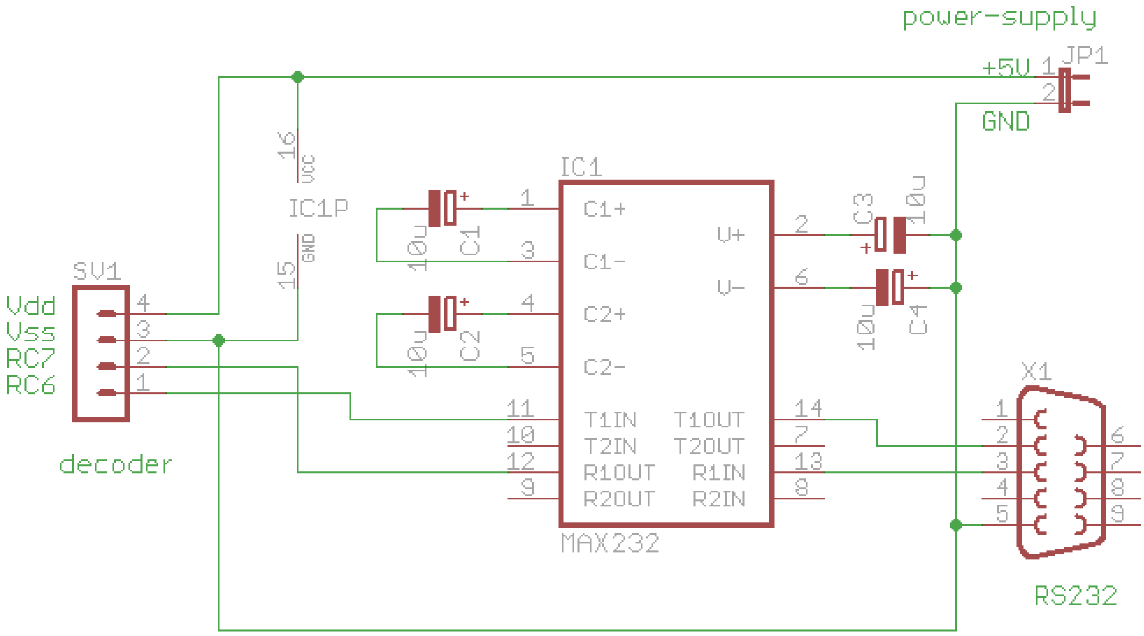


Figure 8 RS232 interface driver

Some shops sell RS232-Interfaces as module or kit for little money. In Germany a kit is offered by Pollin (www.pollin.de) for less then 4 Euro under the designation „RS232-TTL-Wandler-Bausatz“ (order number 810 036).

5.3.4 I2C-Interface

Some people use TV-satellite-tuners as front-end. Such tuners have to be initiated by a sequence of command via an I2C-interface. The adsbPIC-decoder can send these commands to the tuner.
The I2C-interface of adsbPIC-decoder is using 100 kHz clock. The I2C-interface is a 2-wire bus. Consequently the 2 pins of the microcontroller are necessary to control the bus.

- RB5 - DATA
- RB6 - CLOCK

These pins are normally reserved for the manual switches switch-A and switch-B. If the I2C-interface is needed, then these both switches can not be used.

The support of the I2C-bus was introduced in firmware Fw11 and adsbScope V2.6t6.

5.4 Receiver-Decoder GNS5890

Not all aeronautic enthusiasts are electronic hobbyists too.

There are some industrial products on the market. I tested the **GNS5890**-USB-stick. The GNS5890 is smaller then a matchbox and contains the ADS-B-receiver and the decoder. An antenna with 1 meter long antenna cable is included. This set is ideal for mobile use.

Figure 9 GNS5890 with antenna

The decoder is fully compatible to my adsbPIC-decoder, and was delivered with the (relabeled) firmware 8. The CDC-driver has to be installed to use this decoder. You can use the driver-CD of the GNS5890 or my ZIP-file (since 16.12.2011). After driver installation the GNS5890 is supported by Planeplotter too.

The GNS5890 is a very strait design without any frills. It has no DIP-switches, RS232-interface or input for external comparator. This made it possible to shrink the size and to optimize the hardware. The result is a capable and reliable device.

Figure 10 GNS5890

During the tests if outperformed my receive chain (a stocked triple dipole antenna, miniadsb-receiver and adsbPIC-decoder). The frame rate was 20% up to 100% higher, and more aircraft have been tracked.
After little modifications of my miniadsb-receiver both systems are comparable.

Later models of the GNS5890 are delivered with a firmware 9. However, this is not exactly identical to my firmware 9. I have no information about the details of this firmware.

My test-device was delivered without bootloader. Thus an easy update to a new firmware version is impossible.

5.5 Receiver-Decoder BEAST

The Mode-S-Beast is another receiver-decoder supported by adsbScope. But while the adsbPIC-decoder is based on a microcontroller, the Beast is using an FPGA-chip. The Beast is not designed to be small, cheap and easy to build. The goal of the Beast-designer was pure performance. The result is impressive, but the price tag too. The picture shows the vanilla version of the Beast (one receiver and USB-interface), but versions with up to 4 receivers and Ethernet and Bluetooth are available.

Figure 11 Beast - basic version

You can not build your own Beast from scratch; you have to order a kit. This contains all necessary parts including the printed board with all critical SMD-parts soldered on it.

The standard interface of the beast is a USB-port (employing an FTDA-chip). To use the Beast with adsbScope you will have to install the FTDI-driver. If you connect the Beast to a PC, a virtual COM-port is created (like for adsbPIC). But while for adsbPIC the baud rate of this port has no meaning, the user of the Beast has to select this baud rate carefully.
The Beast contains 2 internal switches to select one of 4 possible baud rates. By default a baud rate of 3.000.00 baud is the factory preset. Consequently in adsbScope a baud rate of 3.000.000 has to be selected (**decoder – select COM-port**).

6 PC-Software adsbScope

The software **adsbScope** is optimized for cooperation with my **adsbPIC**-decoder. It can remote control the decoder and supports all adsbPIC-functions. It decodes the received ADSB-data and creates a virtual radar display. AdsbScope can exchange data via LAN or internet.
AdsbScope is a Win32-application. It is developed and tested under Windows-XP. To use it at Windows-Vista or Windows7 it may be necessary to use the compatibility mode of these operating systems.

6.1 Installation

The software is part of a ZIP-file. Create a directory and unzip the software into this directory.

6.1.1 Subdirectories

If the ZIP-file is unzipped, several subdirectories of the program folder should be created and some of them should contain files. If they are missing, then adsbScope would create them during program start, but then they would not contain the necessary files.

The rest this chapter is not necessary for normal users. You can skip it and go to 6.2 Start the Software .

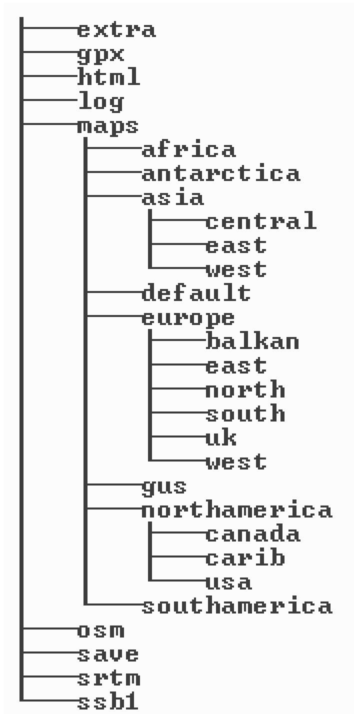


Figure 12 Directory structure relative to program directory

Thus please check, if during uncompressing the subdirectories were created too!

6.1.1.1 /save

This is the directory for saved program states. Each state is made up from up to five files. A ***.pos**-file contains numerical data, ***.jpg**-files contains the OSM- and SRTM-background pictures. ***.rri** and ***.rrj**-files contain data about the receivers maximum and minimum range.
If default -files (**default.pos** , **default.jpg …**) exist, then these files are loaded during program start automatically.
After program installation this directory is usually empty or I placed an example in this directory.

6.1.1.2 /extra

This folder contains additional helpful files. **Do not touch them!**

world1.txt
This file contains the outline of the continents. It is used instead of the more detailed individual state maps **(/maps**) if the user zooms out in the graphic display.

citys.txt
This is a list of >3500 cities. They can be shown for orientation as simple circles on the graphic display. It is based on data the company MaxMind.com released under GPL-license.

GlobalAirportDatabase_dos.txt
This is a long list of airports (based on http://www.partow.net/miscellaneous/airportdatabase/). All airports are shown as circles by default. If the user zooms into the picture, then the name and the altitude of the airport can be shown.

icao24plus.txt
This is the list of airframes that adsbScope is using. **Don't modify it!**
The Aircraft-Manager of adsbScope can be used to download the latest version of this file. (**menu - other - manage other aircraft**)

icao24plus1.txt
This file doesn't exist by default, but it can be made by the user. It is an additional airframe list, with the same format like icao24plus.txt. The idea is that the user makes his own file of additional airframes, which are not contained in the main-file.
If the software detects such a file, then it will load it in addition to the main file. If an airframe is contained in icao24plus-file too, then the software will use the entry from the icao24plus1-file instead.
The Aircraft-Manager of adsbScope can be used to create and maintain this file. (**menu - other - manage other aircraft**)

airframesunknown.txt
This file is generated by adsbScope during normal work. If an unknown airframe is detected, then its ICAO number (and if possible its nationality) is added to this file. This file is used by the Aircraft-Manager to create the icao24plus1.txt-file.

state_icao24.txt
This file contains additional information to identify the nationality of aircraft.

6.1.1.3 /log

This directory is used for log files.

6.1.1.4 /maps

This folder contains several subdirectories for simple maps (outlines) of the states of the world. Some are contained in the ZIP-file and installed by default. The missing files can be downloaded by use of the menu point **load maps - download maps from the internet**. After the download each subfolder contains 2 files. They contain the coastlines, boarders and state names of a whole continent or a part of the continent in a special file format.

In the **load maps** menu the names of available (but not loaded) maps are shown in black color. Missing maps are shown in gray. After a missing map was downloaded from the internet, its color is changed to black.

The user can load these files into the software manually with the menu option **load Maps**. The software then loads the files from the selected directory. The software will automatically load all files from all selected directories at the next program start. If many maps are selected, then this will delay program launch by several seconds.

The color of a loaded map in the menu point **load maps** turns into gray, and the name of this map is checked.

It is not possible to unload a specific map from the software. But the user can use the menu point **load maps - unload all maps** to unload all maps from the software. After this was done the user can load again the necessary maps.

The files for Russia are based on a different data source. Their borders don't mach exactly with the borders of the neighbored states.

6.1.1.5 /osm /aer /mpq

To generate background pictures, the software downloads "tiles" from the internet. All this tiles are buffered in these directories, thus they have not to be downloaded again. All tiles are normal PNG- or JPG-pictures. The user can erase these files, but this would slow down the background generation.

The software tries to detect and erase broken picture-files during program start. If many files are in this subdirectory, then the start of the software may be delayed by some seconds.

After program installation these directories should be empty.

6.1.1.6 /srtm

To generate SRTM (shuttle radar topographic mission) background pictures, the software downloads SRTM-tiles from the internet. All this tiles are buffered in this directory, thus they have not to be downloaded again. All tiles are normal JPG-pictures. The user can erase these files, but this would slow down the SRTM-background creation.

After program installation this directory should be empty.

6.1.1.7 /ssb1

The software is using some files from **http://jetvision.de/sbs.shtml**. These files are originally generated for the SSB1 receiver. Thus I place them into this subdirectory.

Download the ZIP-files for your region from the **jetvision**-homepage. Unzip the files and copy the ***.out**-files into the SSB1 subdirectory. The following files are used by adsbScope:

***_apt.out**
Such files contain information about location and length of runways. Without these files all airports are displayed as circles only. With such files runways are shown as lines inside (more or less) these circles.

***_ats.out**
Such files contain information about air routes. Today the air traffic controllers often do not stick to air routes any more, thus this files are less important. Air routes are not displayed by default.

***_ils.out**
Such files contain information about location and direction of ILS landing aid systems. They are displayed as dotted lines at the ends of runways. ILS are not displayed by default.

After program installation only some files for central Europe (especially Germany) are in the subdirectory.

radarsites.txt
This file contains the location of ground radars. It is based on http://jetvision.de/gm/list.php

6.1.1.8 /gpx

The software can show simple topographic data from gpx-files in the graphic display. Only gpx-files from the /gpx-subdirectory will be used.

After program installation this directory should be empty.

6.1.1.9 /html

This directory is used to add data from the webpage airframes.org into the database of the software.

After program installation this directory should be empty.

6.2 Start the Software

To start the software double-click the executable file. AdsbScope will open its program window centered at the monitor, check out the subdirectories and load some data files. If a default program state was saved, then this will be loaded, and adsbScope will use the saved coordinates, zoom-value, window size and background picture.

If the user has selected many maps (menu point **load maps**), then this will delay program launch by some seconds.

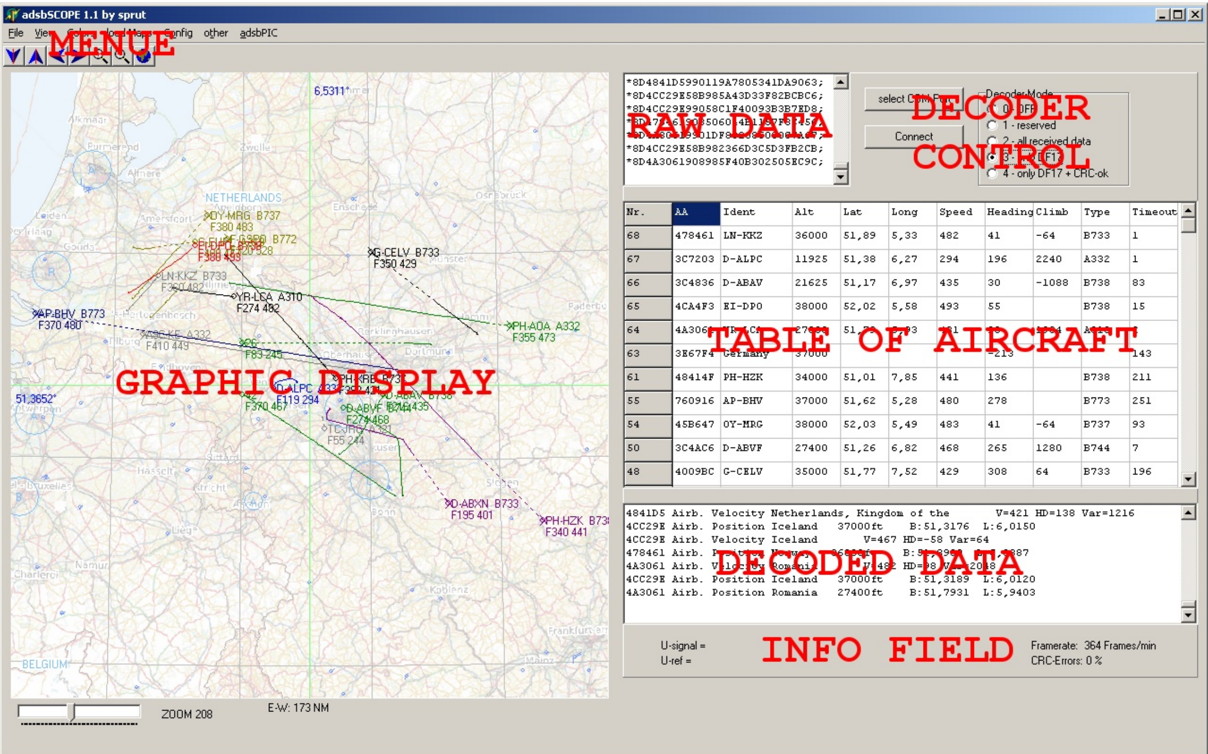


Figure 13 program window

The program window contains:

- a menu to control the program
- a graphic display
- a text box for raw data
- a text box for decoded data (disabled by default)
- a table for detected aircraft
- a decoder control area
- an information field

If no default program state is available, then it will use the starting position 6deg east 51deg north (and no OSM or SRTM picture) as start point.

The upper text box (raw data text box) will show some information about loaded data - ignore it.

Use the mouse to move to your home destination or load a program state with the coordinates of your home. (is described in the following chapter 6.3)

Select the menu point **Decoder** and select the type of the decoder connected to the PC. The software supports adsbPIC, GNS5890, rxControl and the Beast.

Now click on **select com port** and select the com port, which is used by the decoder. If you like to use a Beast, then set the baud rate of the COM-port equal to the baud rate of your Beast. The factory preset of the Beast is 3.000.000.

Of course the decoder has to be operational and connected to the PC. The microchip-CDC-driver has to be installed, to get the virtual COM-port for the Decoder.

If the LEDs at the decoder are flickering, then check the switch 4 at the decoder (has to be closed), disconnect and reconnect the USB-cable from the decoder. The LED1 and LED2 don't have to light up or flicker at this point of time!

Click then on **connect**.
The software will now connect to the decoder and try to confirm the decoder-selection.

If adsbPIC or GNS5890 was detected, then the virtual radar will now start in the decoder-mode 2. You can select the mode manually from the available decoder-modes:

- Mode 2 : all received adsb-data is transferred from decoder to PC
- Mode 3 : all DF17/18/19 information is transferred from decoder to PC
- Mode 4 : all DF17/18/19 information with confirmed CRC is transferred

For the beginning I suggest to use Mode 2.

Now the software starts to work, and the LEDs 2 and 3 at the decoder start to flicker.

Of course only, if the decoder reference voltage is set correctly, the miniadsb is operational and connected to the decoder and the antenna is connected to the miniadsb and receives signals.

All incoming adsb-information is listed in the upper text box.

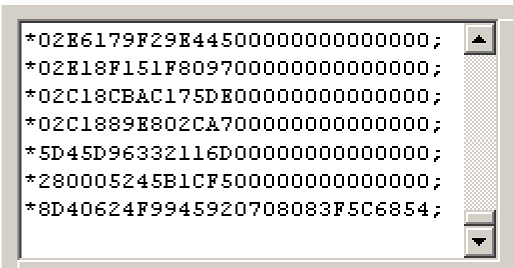
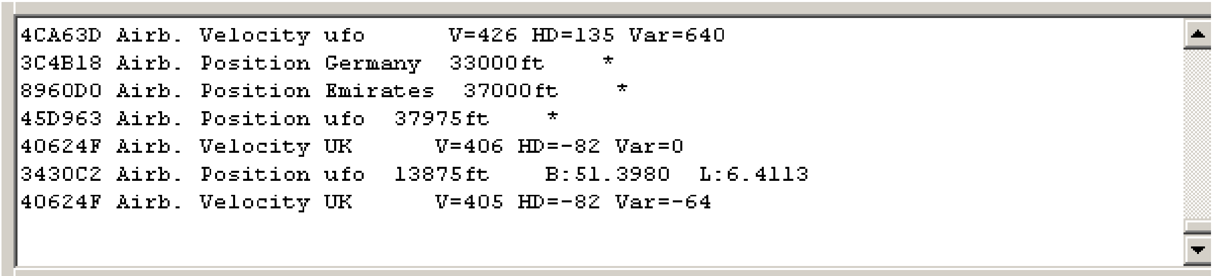


Figure 14 Typical RAW data

At high frame rates the RAW-data zips through this window to fast for the human eye. A simple mouse click on the window interrupts the list of RAW-data. A second click activates the list of data again.

If the box has collected more then 20000 lines, then it will be erased to save memory space.

The decoded content of the information can be listed in a second text box at the lower end of the program window.
The user can use the “hide decoded data”-button to hide or view this box. If the box is hidden, then more space is available for the table of detected aircraft. This text box is hidden by default.
If the box has collected more then 20000 lines, then it will be erased to save memory space. A simple mouse click on this text box interrupts the list of decoded-data. A second click activates the list of data again.



```
4CA63D Airb. Velocity ufo      V=426 HD=135 Var=640
3C4B18 Airb. Position Germany  33000ft    *
8960D0 Airb. Position Emirates  37000ft    *
45D963 Airb. Position ufo  37975ft    *
40624F Airb. Velocity UK      V=406 HD=-82 Var=0
3430C2 Airb. Position ufo  13875ft    B:51.3980  L:6.4113
40624F Airb. Velocity UK      V=405 HD=-82 Var=-64
```

Figure 15 Typical decoded data (disabled by default)

Data of detected aircraft is shown in the table between both text boxes.

The graphic picture on the left side of the program window shows position, track history and additional information of aircraft with known position.

The software counts the number of ADS-B frames (data packets) per minute and displays it in the lower right corner of the program window and in the program status bar. Behind this value is shown (in parenthesis) the average number of frames received from one aircraft per minute.

In addition the software checks the CRC-checksum of each frame. The percentage of broken frames is shown below the frame rate.
In decoder-mode 2 (all received data) it is normal, that during the first minutes a small number of CRC-errors is reported. These are DF11-replies. The software has to monitor these replies for some minutes to learn which interrogators are active. After that it can separate between broken and correct DF11-data. During the learning process all DF11-replies to unknown interrogators are counted as broken frames.

6.3 Orientation/Navigation

The graphic display is an orthographic projection from a fraction of the globe. The size and the location of this fraction can be changed easily with the mouse.

6.3.1 Zoom

Click with the right mouse button into the picture and move the mouse up and down (while the button is still pressed) to change the size of the displayed part of the earth. You can zoom out, until the whole globe fills the display or zoom in until the picture represents less then 2 NM (nautical miles).
The tool bar above the picture contains zoom-in and zoom-out buttons too.

The graphic displays level of detail depends on the zoom level. If the user zooms out, then only the coastlines of the world are shown instead of detailed maps. The resolution for latitude- and longitude-grid is 10 degree.

If one zooms in, then instead of the world coastlines the detailed maps are used (if any are loaded).

If one zooms in more, then latitude-longitude-grid is shown wit 1 degree resolution.

If one zooms in more, then labels for airports, ILS, radar-sites … are displayed.

6.3.2 Coordinates

To move the center of the picture to different coordinates, click with the left mouse button into the picture and move the mouse, while the button is still pressed. Doing this, you will shift the content of the picture, and the picture center represents different coordinates. (But this will not work, if you click on the symbol of an aircraft or very close to it.)
The coordinates of the picture center are shown in the status bar below the picture and in the picture itself. The latitude can be seen at the middle of the pictures left side. The longitude is shown at the middle of the upper side of the picture.

The coordinates of the mouse cursor position are shown in the status bar below the graphic picture.

By default the coordinate values are shown as fractional degree values (e.g. 51,1234). With the menu-point **Config - coordinates** the coordinate display can be set to degrees, minutes and seconds (51: 12'34'') or degrees and fractional minutes.

To move over a long distance in short time, one first should reduce the zoom value.

If an OSM- or SRTM-picture was displayed, then it will be erased after the center coordinates where changed. A new picture is not generated automatically.

The tool bar above the picture contains arrow-buttons to change the coordinates in small increments (5 pixels). If this buttons are used, then the OSM/SRTM-picture is not erased immediately. But software will do this later, if one clicks into the picture.

6.3.3 Display/Receiver coordinates and tracking

The coordinates of the ADS-B-receiver have an influence of the capability of the software to detect and track aircraft. By default the receiver position is identically with the center of the graphic display at program start and marked by a red circle with a red cross. If a new location is loaded (**File - load**) then the new center of the graphic is used as receiver location.
The position of the receiver can be changed. To do this move the center of the graphic display to the new location and click on the **set receiver location** button or use the menu point **other - set Receiver location**.
If a location is saved (**File - save as…** or **File - save default**) then the center of the display is stored as receiver position (and not the last known receiver location).

6.3.3.1 Go to Towns or Airports

The software knows the coordinates of >3500 cities and >4000 airports.
The menu-point **other - goto Town or airport …** opens a navigation window with two selection lists. The left list contains all selectable towns and the right lists all selectable airports. Above both lists are input lines. The user can select a town or an airport from the list with a mouse click or he can type in the name of a town or airport in the input lines. If a location was selected from a list or if the name of a known town or airport was typed in then the center of the graphic display will be moved to the new location immediately.
If multiple towns or airports have exactly the same name, then the software selects always the first town or airport from the list.

The **go back**-button moves the center of the display back to the initial location.

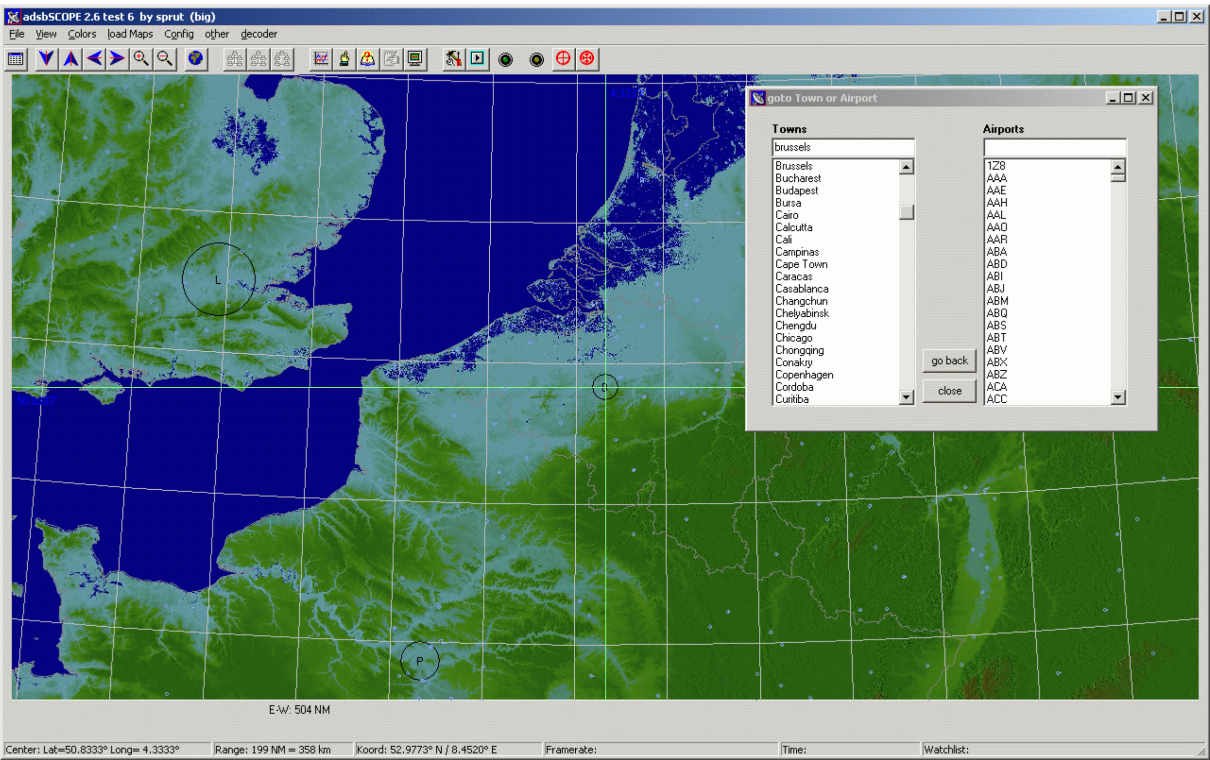


Figure 16 Got to Towns and Airports

The town in the list are selcted by the menu point **Config-towns**. The airport labels (normal, icao, iata) ar selected by the menu point **Config - Airport names**.

6.3.3.2 GPS-support

A GPS-receiver witch NMEA-data protocol (NMEA-0183) can be connected to the PC and used by the software. The software compares the GPS-coordinates with the coordinates of the centre of the graphic display. If the difference exceeds some hundred meters, then the display centre is moved to the GPS-coordinates (and of course the background picture discarded). This feature is useful for mobile use.

To use the GPS-receiver, it has to be connected to the PC. The menu-point **GPS - select COM-port** has to be used to select the correct COM-port and the baud rate (both values are stored for later use).

After the menu-point **GPS - use GPS** is checked, the GPS-data is processed and used to correct the position of the graphics display centre.
The menu-point **GPS - show GPS data** opens a window with all available GPS data: used satellites, signal quality, coordinates...

Not all GPS-receivers support the NMEA-0183-protokoll. Especially cheap devices miss this capability. Older NMEA-capable receivers use a real RS232-Interface. Modern devices have USB-interface and open a virtual COM-port if they are connected to the PC. The standard baud-rate is 4800 bps, but some devices use other baud rates. The NAVILOCK-464US e.g. is using 38400 bps by default while NAVILOCK-NL-302U is using 4800 bps.

6.3.3.3 Aircraft on the ground

As long an aircraft is on the ground (taxiing, rolling on the runway) its position can not be calculated unambiguous. The software tries to find an airport, which matches with one of the possible aircraft positions. In the very most cases this will result in an unambiguous aircraft position. But sometimes this test will not be successful, e.g. if the aircraft is taxiing on an airport which is unknown to the software.
Consequently the software then includes the receiver coordinates into the calculation to determine unambiguous coordinates for the aircraft. To get the correct coordinates of the aircraft the receiver location should not be more then 45 degree off (in latitude and longitude) from the true aircraft position (airport).

If the receiver coordinates are less then 45 NM away from the airport, then the menu option **Config - Aircraft track - all surface targets are within 45 NM** can be activated to improve aircraft detection. This option is experimental, but I suggest using it by default.

6.3.3.4 Aircraft in the air

If the receiver coordinates are less then 180 NM away from all detectable airborne aircraft, then the menu option **Config - Aircraft track - all airborne targets are within 180 NM** can be activated to improve aircraft detection. But aircraft beyond 180 NM range may be temporary indicated at wrong coordinates. This option is experimental, and I don’t suggest using it by default.

6.4 Graphic display

The user can decide which information should be displayed, and which colors are used for different kinds of objects.
The display is updated at least every second (normally up to 4 times per second).

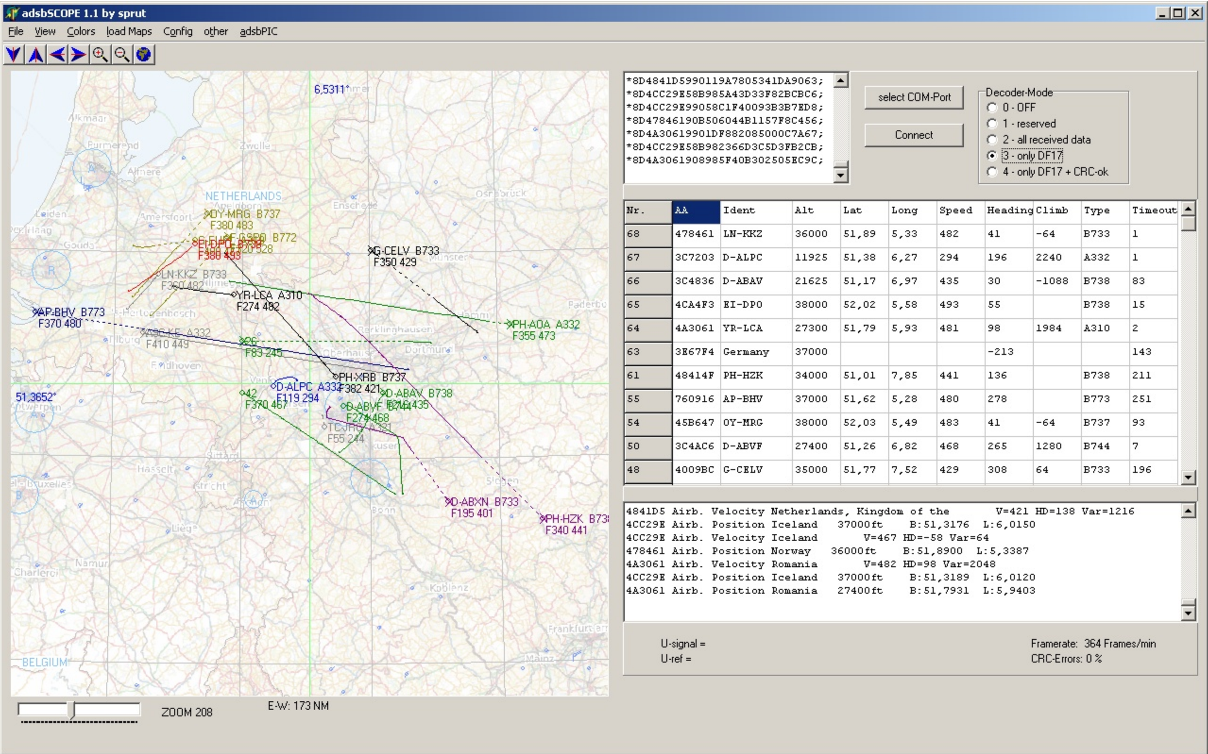


Figure 17 Main program window of adsbScope

Inside the program-window the size of the table and the list boxes is fixed. The graphic display occupies all remaining space on the left side of the program window. If the user increases or decreases the size of the program window, then the size of the graphic display (and the height of the table at the right side) is changed too. On small monitors (e.g. laptops) this may result in an uncomfortable small width of the graphics display.
A double click in the graphic display will maximize its size. Then the graphic will occupy the whole program window. The aircraft-table, the list-boxes and the decoder-control buttons are not visible and not accessible, because they are obscured by the graphic. Another double-click will reduce the size of the graphic again.

Instead of the double click, the user can use the **maximize/normalize graphic display** -button above the graphic display.

6.4.1 Aircraft

6.4.1.1 Flight Path

Detected aircraft with known position are shown in the graphic display. The software draws a line (the flight **path**) from the position where the aircraft was detected first to the momentary position. The momentary position is shown as a little symbol (circle, rectangle or aircraft-symbol).

If no aircraft data was received during the last 20 seconds, then a little diagonal cross marks this symbol. (missed aircraft). If the menu point **Config - drop Aircraft - hide it after 20 sec** is checked, then the aircraft is not shown at all anymore. However, if later data is received again, then the whole aircraft path is redrawn.

But if no aircraft data was received during the period defined in **Config - drop Aircraft** then the aircraft data is discarded.

6.4.1.2 Symbol

The symbol is by default a little airplane-figure. But the user can select a little circle or rectangle instead. If aircraft-figure-symbol was selected, then new detected aircraft may be shown as little circles for some seconds before the aircraft symbol is used. This can happen, if the software needs time to estimate the correct aircraft heading. The user can select the symbol with the menu point **Config - Aircraft track - Symbol.**

6.4.1.3 Label

Beside the symbol alphanumeric information is shown in two, three or four lines. In 2-line-mode the upper line shows the track number, followed by the aircraft identifier (e.g. **D-AHFA**) and the airframe type (e.g. **B738** stands for Boing B737-800). The lower line shows the flight altitude in 100 foot increments (**F310** stand for 31000 feet) and the speed in knots.
If the airframe is unknown (not in **icao24plus.txt**), then the upper line shows only the track number. This number refers to the first column of the table on the right side of the program window. (The track number corresponds to the first column of the table.)

In 3-line-mode the first line shows track number followed by the aircraft call sign. The second line shows the registration or the ICAO24 number of the aircraft. The 3d line shows altitude, speed and heading. Above the **Transition Altitude** (by default 6000 ft) the altitude is indicated in 100ft increments and has the prefix ‘F’. Below the Transition altitude the altitude is indicated in 1 ft increments. The Transition altitude is by default 6000 ft, but can be changed via the menu point **Config - Aircraft track - Transition Altitude/FL**.
A ‘+’ or ‘-‘ symbol behind the altitude value indicates an climbing or descending aircraft.

The user can select 2-line-, 3-line or 4-line-mode at the menu point **Config - Aircraft track - Label**.

6.4.1.4 Predicted Position

If momentary no data is received from an airframe, then its position is calculated based on last known position, speed and heading. The flight path from the last known position to the predicted position is drawn wit a doted line. The user can prohibit predicting positions with the menu point **Config - Aircraft Track - show predicted position**.

6.4.1.5 Colors

The software can choose for every aircraft a random color. These colors are relatively saturated to improve visibility. The user can select a mode, where the color of the flight path represents the altitude of the aircraft. (**Config - Aircraft Track - color by altitude**) In this mode 0 feet is represented by red while 20000 feet is green and 40000 feet is blue. Altitudes in between these values are represented by mixed colors (this is default after program installation).

The user can choose a fixed color or the randomly selected color for the label-text. (**Config - Aircraft Track - Label - random color**). The fixed label-text-color is by default black, but can be permanently changed by the user (**Colors - aircraft label**).

6.4.1.6 Example

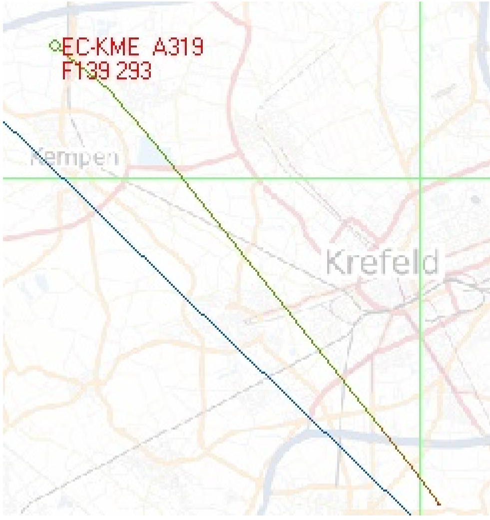


Figure 18 Aircraft track

This picture is a fraction from the graphic display. It shows the Spanish aircraft **EC-KME** flying into north-west direction. This Airbus **A319** is climbing (track color changed from red to green) and has just reached an altitude of **139**00 feet (4,2 km). Its speed is **293** knots (527 km/h). The blue track is from another aircraft flying at higher altitude.

6.4.2 Latitude-Longitude-Grid

A latitude-longitude-grid is shown with 10 degree resolution (global view) or 1 degree resolution (zoomed in).
The grid is enabled by default, but can be disabled by the user.
The color can be changed.

6.4.3 Crosshair

A crosshair marks the center of the picture.
The crosshair is enabled by default, but can be disabled by the user.
The color (default is light green) can be changed.

6.4.4 Receiver Position

A (by default red) circle with a (by default red) cross marks the position of the ADS-B-receiver. After program start this is by default the center of the graphic display. Correct receiver coordinates have (a small) influence on the detection- and tracking-capability of the software. I suggest setting the indicated receiver position to the real receiver position. To do this simply shift the display center to the right coordinates (OSM-background picture may be helpful) and click on the **set receiver location** button or use the menu point **set Receiver location**.
If a location is stored (**File - save**) then the center of the display is stored as new receiver location.
The receiver symbol is enabled by default, but can be disabled by the user. The color of the symbol (default is red) can be changed.

6.4.5 States

Outlines of states can be displayed, if their maps are loaded.
The display of borders is enabled by default, but can be disabled by the user.

The color (default is gray) can be changed.

6.4.6 Towns

Nearly all towns (>3500) with more then 100.000 inhabitants can be shown as simple circles. The diameter represents the population of the town (without suburbs) and the first letter of the towns name is displayed inside the circle. At large zoom-values the full names of the cities are shown. The user can decide to display only towns with more then 300 thousand, 1 million or 3 million inhabitants (**Config - Towns**).
The display of towns is enabled by default, but can be disabled by the user.
The color (default is blue) can be changed.

6.4.7 Airports

Airports are shown as small circles. Beside the circle the airport name and the airports altitude (in feet above the ocean level) is displayed, if the user zooms in enough.
The displayed name can be the normal name, the ICAO-name or the IATA-name. (**Config - Airport names**)
The airports and there IATA-names are enabled by default, but can be disabled by the user. The altitude value is disabled by default but can be enabled by the user.
The color (default is blue) can be changed.

Runways can be displayed only, if the **/ssb1** subfolder contains the right ***_apt.out**-file for the selected region.

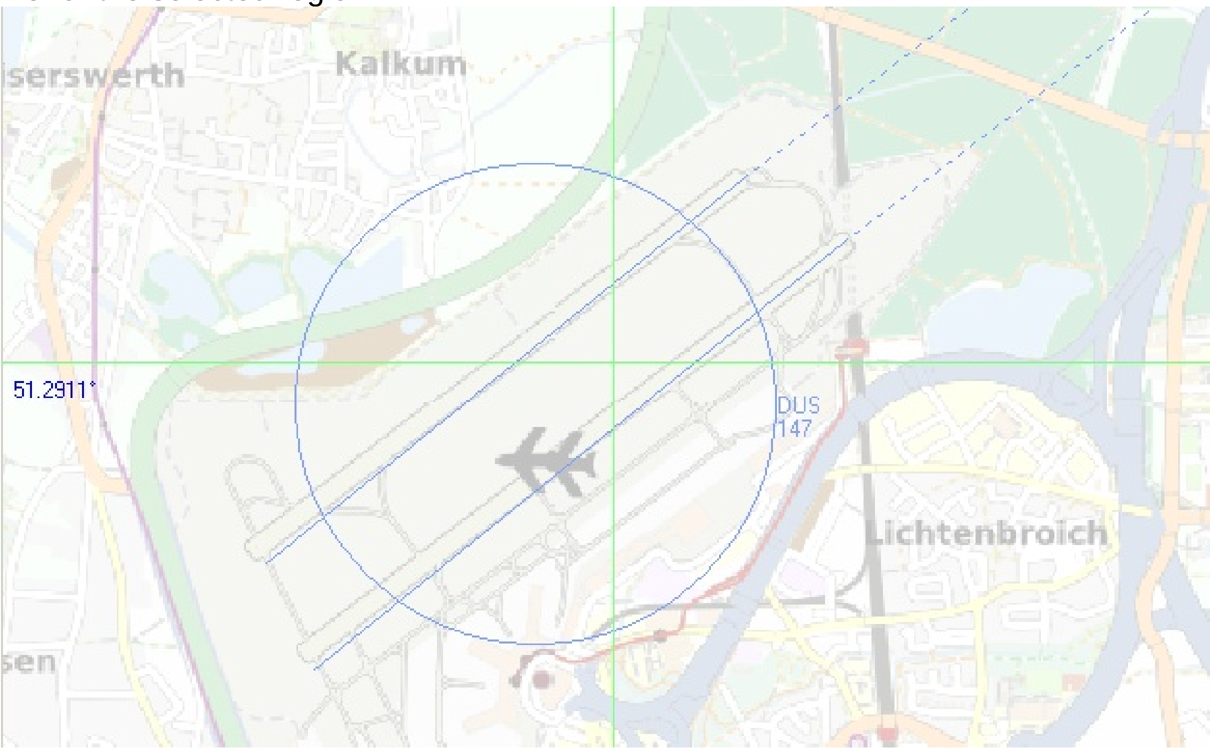


Figure 19 Airport

This picture shows Dusseldorf airport (DUS) in Germany. One can see the circle, the name (**DUS**) and the altitude (**147** ft). All this is basic information (from GlobalAirportDatabase_dos.txt) and is available for all airports by default.

In this picture one can see the runways (blue solid lines) and the ILS path (blue doted lines). This is additional information from the files **eddk_apt.out** and **eddk_ils.out**. To get these files I downloaded the file **EDDK.ZIP** from the page http://jetvision.de/sbs.shtml. This zip-file contains both **eddk_*_out**-files. I copied them into the **/ssb1** subdirectory.

The background picture is the OSM-map.

6.4.8 ILS

ILS paths can be displayed only, if the **/ssb1** subfolder contains the right ***_ils.out**-file for the selected region.
The ILS paths are disabled by default, but can be enabled by the user.
The color is identical to the color of airports. They are drawn with doted lines.

6.4.9 Range rings

Around the display center can be drawn circles of defined radius. They make it easier to guess the range to objects on the display.
The rings are disabled by default, but can be enabled by the user.
The color (default is light green) can be changed.

With the menu option **Config - Range Rings** the user can switch between:

50NM
Up to 5 circles are displayed. Their radiuses are multiples of 50 NM.

50km (default)
Up to 8 circles are displayed. Their radiuses are multiples of 50 km.

5000ft altitude
The maximum possible detection range of an airframe is limited by the round earth shape and depends on the aircraft altitude.
Up to 8 circles are displayed. Their radiuses represent the maximum possible detection range to aircraft at specific minimum altitudes. These altitudes are multiples of 5000 feet.

By the way:
There is a **Range:** -information displayed in the status bar of the program window. This is the distance between the display center and the mouse cursor position in nautical miles and in kilometer.

6.4.10 Maximum Detection Range

If the menu point **Maximum Range** is activated, then software monitors all aircraft detections to determine the maximum detection range in all directions from the receiver. A wind-rose-like diagram around the receiver position shows the maximum range. The angular resolution is 5 degree.

At the menu point **Config - Maximum Range** the user can choose between a simple display of the maximum range and a display with maximum ranges for different altitudes. The simple maximum range display is using black color by default, but this can be changed by the user. The “Range by altitude”-display is using different colors for different altitude levels.

- below 10000 ft: red
- 10000 ft .. 19999 ft dark green / brown
- 20000 ft .. 29999 ft green
- above 29999 ft blue

If the position is saved (**File - save as…/save default**) then the maximum detection range values are saved too, and if a position is loaded, then this values are loaded too and can be displayed again. The values are erased if a new receiver position is selected.
The Maximum Detection Range is disabled by default, but can be enabled by the user.

6.4.11 Ground RADAR sites

The position of ground radar sites can be displayed as small circle. Beside the circle the radar name is shown (if you zoom in enough).
The ground radars are disabled by default, but can be enabled by the user.
The color (default is red) can be changed.
This option has not much value and may be removed in later software versions.

6.4.12 ATS Routes

Air routes can be displayed only, if the **/ssb1** subfolder contains the right ***_ats.out**-file for the selected region.
The air routes are disabled by default, but can be enabled by the user.
The color (default is yellow) can be changed.

6.4.13 GPX-Data

Simple GPX-files with topographic routes/tracks can be shown. All files from the **/gpx** subfolder are loaded and can be displayed.
GPX-support is very limited at the moment. Only route <rte> and track <trk> data is used by the software.
The GPX-data display is disabled by default, but can be enabled by the user.
The color (default is red) can be changed.
The example shows a bicycle-tour (the red line) (taken from http://www.bbbike.de) through Berlin. The background is an OSM-picture.

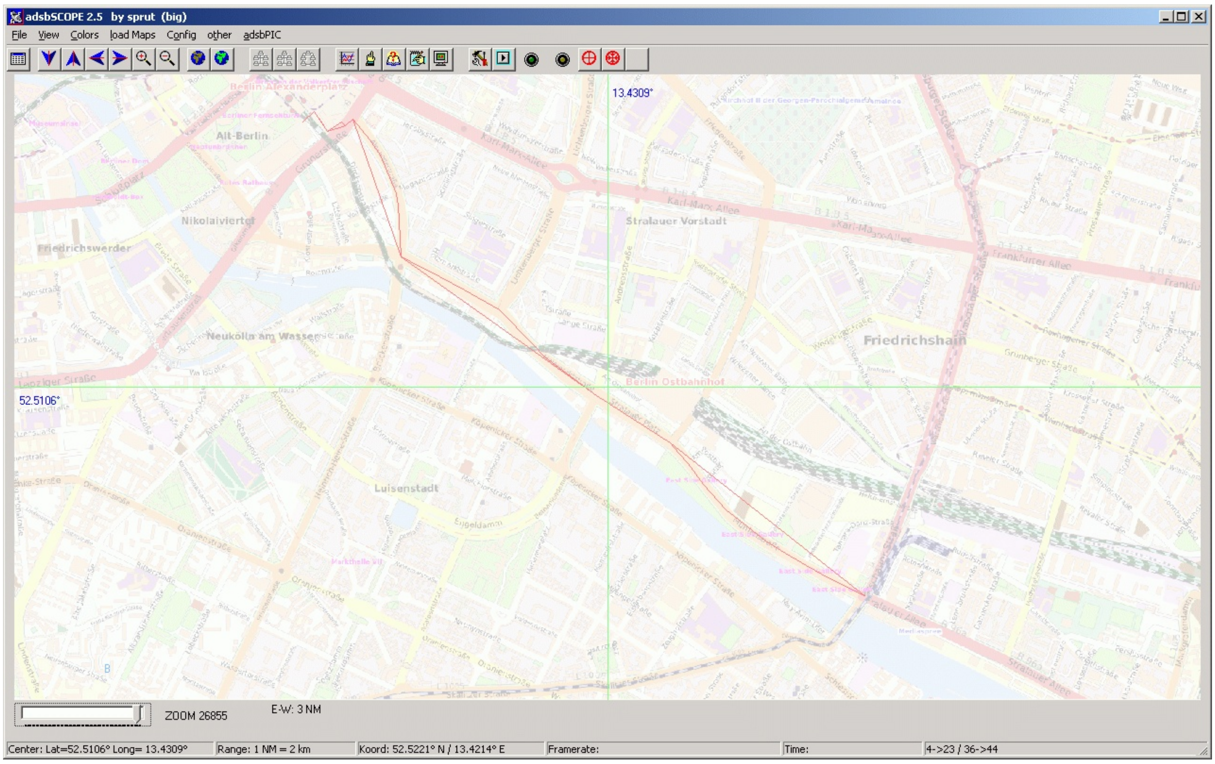


Figure 20 GPX-data example

6.4.14 Background picture

The software can generate a background picture using data from MapQuest, Open Street Map or SRTM. This requires an internet connection.
The data source (and consequently the type of picture) is selected by the menu point **Config – Background Picture**.

SRTM-pictures are synthetic color pictures with color-encoded altitude information. They are based on space shuttle radar data.

MapQuest/OSM pictures are open street map based pictures from the MapQuest-project. They have fewer details then the original OSM-pictures. But for the use in AdsbScope “this is a feature and not a bug”.

MapQuerst/Aerial -pictures are based on satellite photos. They may not be available for all areas and zoom-levels.

OpenStreetMap-pictures are the original OSM-pictures. Compared to MapQuest/OSM they have a better resolution and more details. But that is not necessary for the use in adsbScope. Please use this option only if you really-really need it to keep the load on the OSM-servers low. I delay access to OSM-servers always by 10 seconds.

AdsbScope buffers all downloaded data for later use.

6.4.14.1 MapQuest / OSM

To start the generation process one can use the menu point **Maps – Background Picture** or click on the on the button with the blue-gray earth picture in the toolbar.

The picture generation requires several seconds. The progress is visible on the display.

The program buffers all internet data in the **/osm, /mpq** and **/aer** subdirectories. Thus no internet connection is necessary; if a picture has to be generated for nearly the same region with nearly the same zoom level again.

With the menu point **File - save as…** or **File - save default** the program state and the actually used background picture is saved. If a saved state is loaded again, then the saved background-picture is immediately used again.

If the zoom-level or the program window size is changed, then the background-picture is scaled to the correct size. If you zoom out, then the picture will not fill the whole display anymore, if you zoom in, then the picture may show larger pixels. You can generate a new background-picture to fix this.

If the Coordinates of the picture-center are changed, then the background-picture will be moved on the display too for better orientation. However, the shifted background-picture will not exactly match to the terrain anymore. (Slightly different distortion caused by projection.) Because of this, the software will erase the background-picture at the end of the move process. You have to generate a new background-picture to fix this.
If a GPS is used to keep the center of the display to your coordinates, then the background picture will be discarded for the same reason.

The example shows the Chicago O'Hare International Airport as OSM-background picture.

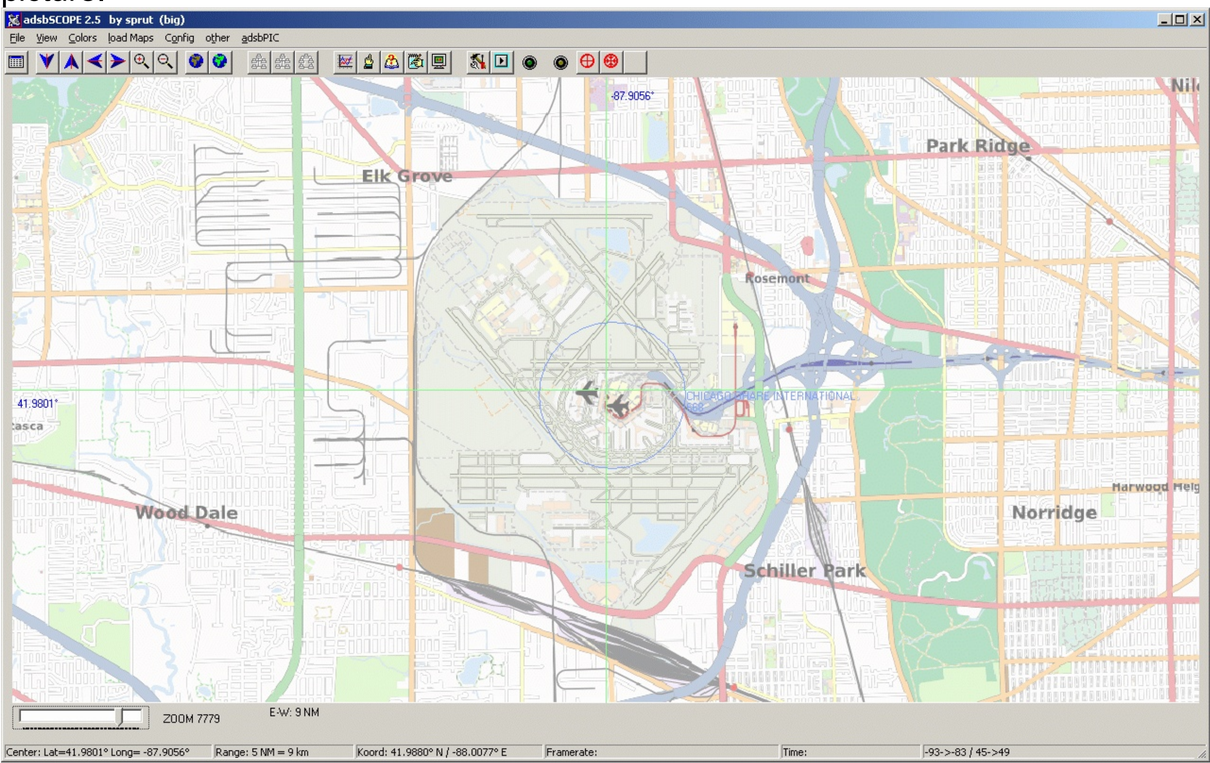


Figure 21 OSM: Chicago O'Hare International

Color:

The original colors of MapQuest-, OSM- and SRTM-pictures are to much saturated for a use as background. Other objects (like aircraft) would not be easy to see anymore. The software can reduce the color saturation or convert the picture into a gray picture. The saturation level can be selected wit the menu point **Colors – Background Picture**.
This option has no effect on maps generated prior the change.

6.4.14.2 SRTM background picture

The software can generate a background picture from data that was collected during the shuttle radar topographic mission in the year 2000. This requires an internet connection. This type of background picture has to be selected by the menu point **Config – Background Picture - SRTM**.

To start the generation process one can use the menu point **Maps - Background Picture** or click on the on the button with the blue- gray earth picture in the toolbar.
The picture generation requires several seconds. A progress bar is shown in the lower right corner of the program window.
The picture is a so called synthetic color picture with color-encoded altitude information.
SRTM-pictures can NOT be generated for latitudes above 60-deg-north or below 60-deg-south. The software rejects the generation of SRTM-pictures if the area of the graphic display covers more then 40 degree in longitude or more then 20 degree in latitude.
The program buffers all internet data in the **/srtm** subdirectory. Thus no internet connection is necessary; if a picture has to be generated for nearly the same region again.

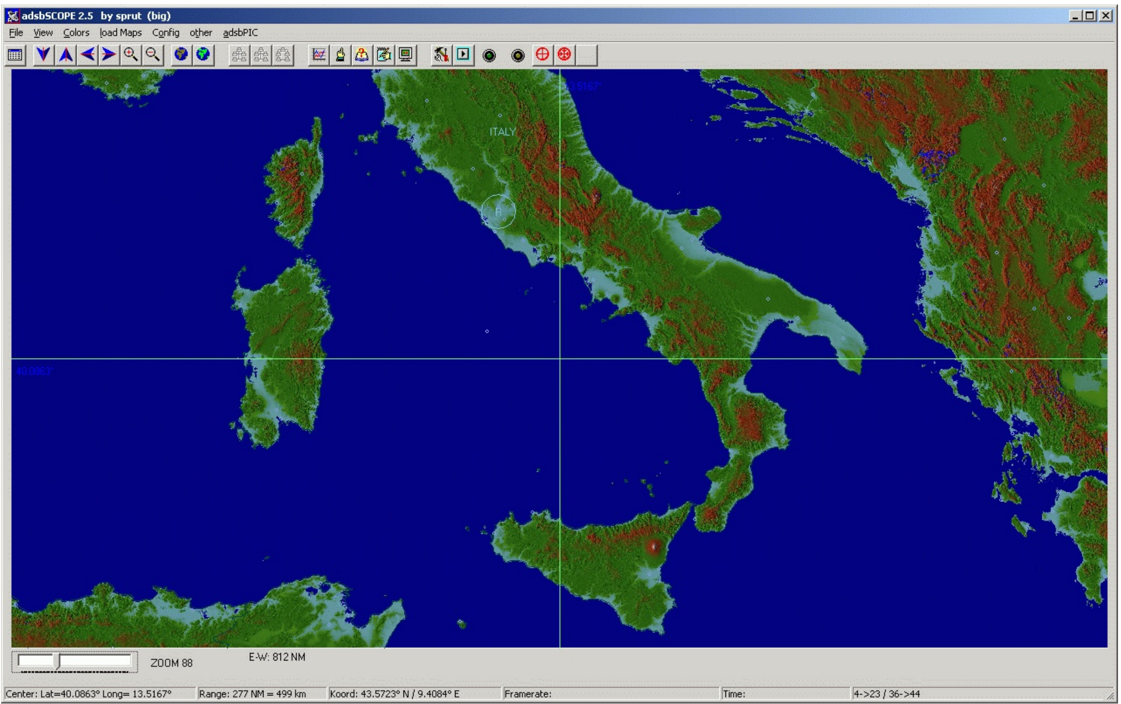


Figure 22 SRTM-background picture of southern Italy

The software handles SRTM-pictures just like MapQuest- or OSM-pictures. Please read the MapQuest/OSM-section for more information. For example one can use reduced chrominance (**Color - Background Picture - pale**) to guaranty the visibility of aircraft.

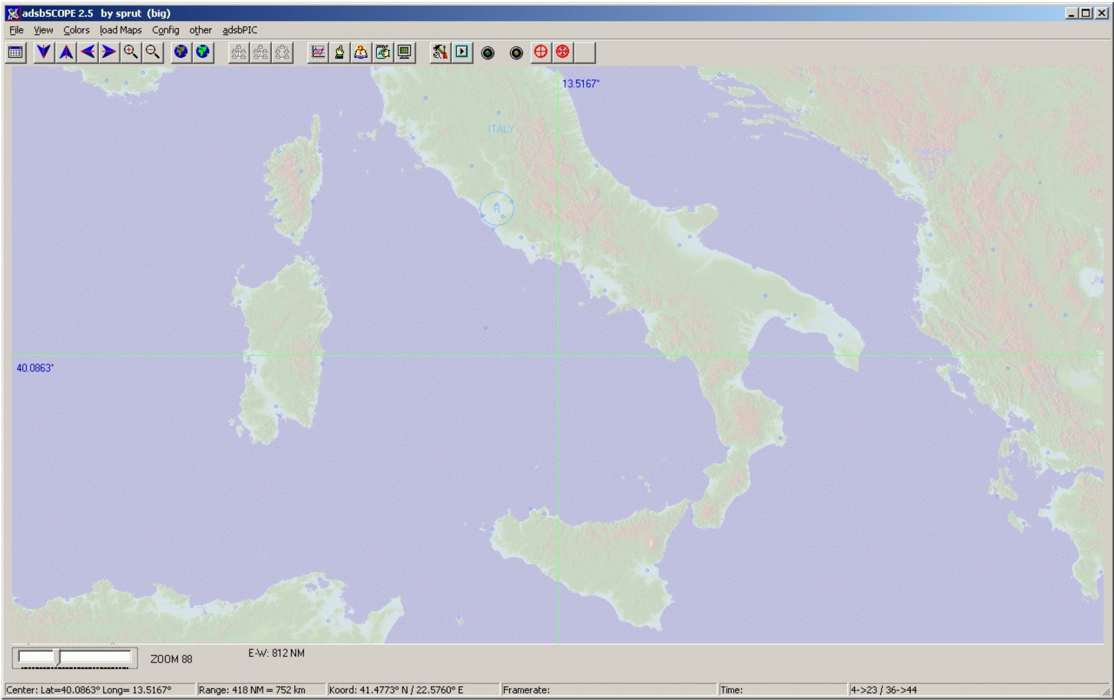


Figure 23 SRTM-background picture with reduced colors

6.4.15 Data Overlay Display

The user can select one aircraft by a click on the aircraft in the graphic display or by a click into the table of detected aircraft (small or big table). The selected aircraft is marked with a circle around its position in the graphic display. The data of this selected aircraft will be shown in the upper left corner of the graphic display. This makes it possible to have fast excess to aircraft data while the graphic display was maximized.

The text color is by default blue, but can be changed by the user.

If the upper left corner of the display is cluttered with other information, then it may become difficult to read the overlay data. In this case the user can make the overlay-data-background non-transparent (**Config – Text overlay – background not transparent**)

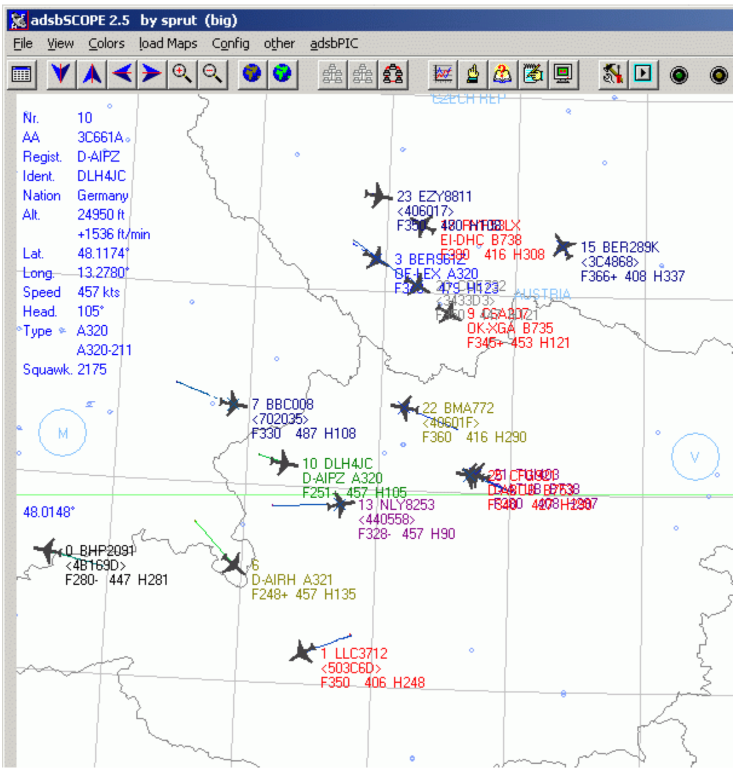


Figure 24 Data Overlay

6.5 Drop Aircraft

If from a tracked aircraft no information is received anymore, then the software starts to show its estimated position based on the last known data. After 20 seconds the aircraft is declared “missed”. A little diagonal cross marks now the position circle.
If from the aircraft was not received any information for a defined time, then the aircraft is dropped from the list of tracked aircraft. This time is by default 5 minutes, but can be changed by the user with the menu point **Config - drop aircraft**.

6.6 Table of detected Aircraft

At the right side of the program window a table shows all momentary detected aircraft. This includes aircraft of unknown position, which are not shown in the graphic display.
By default new detected aircraft are shown at the top of the table. The order can be changed by a mouse click into any field of the head line of the table. The aircraft can be sorted by Number, AA, Identifier, Altitude …. Another click into the same field will reverse the order of the aircraft.

The last column shows the time since the last information was received in seconds.

Nr.	AA	Ident	Alt	Lat	Long	Speed	Heading	Climb	Type	Timeout
24	45D963	OY-VKC	37975			485	43		A321	0
23	4CA63D	ufo	20300	51.82	5.95	426	135	640		0
22	4BCD62	TC-SKB					-213		B734	104
21	501DA9	ufo					-213			69
20	3C5EE5	D-AGWE	23575			366	269	2752	A319	1
19	3C11A1	D-AVRO					-213		RJ85	115
18	393174	F-GMLU					-213		MD83	72
17	3430C2	EC-KME	13075	51.39	6.44	293	321	3328	A319	0
16	8960D0	A6-EBI	37000			443	285		B77W	1
15	40624F	UK	37000	51.51	7.16	406	278			0
14	511038	ES-ABL				379	217	-1408	B735	115

Figure 25 Table of detected aircraft

The picture shows an example. In aviation the royal units are still in use instead of metric units (quarter-pounder with cheese). Altitude is measured in feet (~30 cm), speed is measured in knots (1.8 km/h) and the climb rate in feet per minute.
Heading is in degrees (measured from north-direction, clockwise).
The aircraft are reverse ordered by Number.

If one clicks on data of an aircraft with known position, then this aircraft will be marked with a large horizontal red cross in the graphic display for the next 30 seconds.
During this time the textbox with the decoded data (below this table) will show only new data from the selected aircraft. A click into the headline of the table can be used to deactivate the selection of the aircraft before the 30 seconds are over.
For the selected aircraft the overlay text display in the graphic display will be activated as well and the selected aircraft will be marked with a circle around its position in the graphic display.

The available space on the main window is limited. If you prefer a lager table, simply click on the **big table** button (the very left one) or use the menu point **other - Big Table - activate**. This will open a separate window for a larger version of this table.

The big table can be activated by the hotkey **F1**. Key **F1** can be used to toggle between then main program window and the big table.

6.7 adsbPIC-Decoder / GNS5890-Control

The adsbPIC-decoder and the GNS5890 are remotely controlled via USB by adsbScope.
But as a requirement the software has to be “connected” to the decoder. This is done by pressing the **Connect** Button.
(AdsbScope can use the RS232-interface of adsbPIC too, but I suggest using USB.)

After the connection was established the menu point adsbPIC or GNS5890 becomes visible.

6.7.1 Dataflow / Decoder Mode

The **Decoder-Mode** select box controls the dataflow from the decoder to the PC. By default all received data is forwarded to the PC (**2- all received data**).

If **0 - off** is selected, then the decoder don’t sends any ADS-B-data to the PC.
If **2 - all received data** is selected, then all receive ADS-B information is transferred to the PC. This includes data that can not be used by the PC-software and data with errors (wrong CRC).
If **3- only DF17** is selected, then all receive DF17, DF18 and DF19 data is transferred to the PC. This data is the most valuable for the PC-software. I suggest using this setting.
If **4- only DF17+CRC ok** is selected, then all receive DF17, DF18 and DF19 data with correct CRC checksum is transferred to the PC. This requires additional processing time for the PIC and increases the chance of missing ADS-B signals. It is a better idea to let the PC do the CRC-test. (adsbScope is doing it anyway.)

The drop-down-menu contains the menu-point **adsbPIC - enable MLAT** (formerly: enable time TAG). If this option is enabled (checked), then a precise time code will be generated by the decoder for each received frame. The adsbPIC-decoder time tag is based on a 12 MHz clock. (Some other decoders/receivers use a 20 MHz clock instead.) With the menu point **Config - MLAT clock** (formerly: TAG frequency) the correct clock can be selected.
The decoder (except GNS5890) has a Time-TAG-switch. If this switch is closed, then the decoder will send time tags even if this is not enabled by software.
The time tag can be used to triangulate aircraft position in collaboration with other decoders. If you don’t participate in such a network, then you should deactivate the MLAT-time-tags to reduce the load at the USB/RS232-Interface.

6.7.2 Auto-connect to decoder

If the menu-point **decoder-connect at start** is checked, then adsbScope will automatically connect to the decoder and start data processing at program start. You can use this feature after the virtual-COM-port of the decoder was selected once and the function of the decoder was tested. If this function was activated, then there is no need anymore to use the **Connect** button.
To deactivate this auto-connect-feature you have to uncheck the menu point **adsbPIC-connect at start**.

An alternate way to activate this feature is the command line parameter /A.

6.7.3 Comparator Reference voltage offset

The Decoder sets the comparator reference voltage by default to 100mV above the mean analog signal level. The menu point **adsbPIC - Uref offset** can be used to change this offset to a value between 40mV and 250mV.
If the user changes the offset-voltage level, then the new voltage level is stored in the decoder permanently and is used as new default.

Normally 80...120 mV is a good choice. If the offset-voltage becomes small then noise is detected by the comparator. In this case LED 1 is flashing often, but LED2 not very often. If the offset-voltage is high, then the decoder becomes insensitive.

If the offset value is changed in adsbScope, then it needs about 2 seconds, until the decoder uses the new voltage.

If the option **off** is selected, then the reference-voltage is not regulated anymore and keeps its momentary level. This can be used, if an external comparator is used, and the decoder-generated offset voltage is not needed.

If the option **automatic** is selected, then the offset voltage is adjusted by adsbScope to optimize the framerate of the decoder. The offset voltage is only changed slowly and in small steps. The user can monitor the voltage in the framerate-window.
This option is highly experimental. DO NOT use this option in parallel with **adsbPIC – near targets !**

6.7.4 Test PWM

For test reasons the comparator reference voltage can be set to unusual low (5%) and unusual high (50%) values. The automatic control circuit of the Decoder will then adjust the voltage again. This only demonstrates the function of the control circuit.
If one uses the menu point **adsbPIC-Test PWM 5%**, then LED3 will light up. After some seconds it has to reduce its brightness again until it smoulders.
If one uses the menu point **adsbPIC-Test PWM 50%**, then LED3 will go off. After some seconds it has to smolder again.

6.7.5 Reset

The menu point **adsbPIC-Reset** will reset the decoder. This will force the decoder to go off the USB bus and to reconnect to the PC.
The adsbScope-software can then not use the decoder until the **connect** button was pressed again.
If the adsbScope software is closed, then a decoder-reset will be initiated too.

6.7.6 Activate Bootloader

(Except GNS5890) The menu point **activate Bootloader** will activate the bootloader of the decoder and reset the decoder. A warning is displayed before the bootloader is activated. Then the decoder can not be used anymore, until a new firmware is flashed into the decoder.
The old firmware can not be activated anymore.

The GNS5890 has no bootloader.

6.7.7 Signal levels

If the adsbPIC-decoder is connected to the adsbScope software, then the voltage level of the analog in put signal and of the comparator reference voltage can be measured and displayed at any time. A click on the **U-signal=** label will ask the decoder for the momentary analog signal level. The result will be displayed behind this label. A click on the **U-ref=** label will do the same for the comparator reference voltage.

6.7.8 Hear Beat - Decoder Monitoring

(since firmware 9, except GNS5890)
For test purpose the decoder can send every 1.3 seconds a status report to the PC. This is called the heart beat. To activate the heart beat use the menu point **adsbPIC - enable heart beat**. To disable the monitoring function select the menu point **adsbPIC - disable heart beat**.

If the heart beat is activated, then the labels for the signal levels (U-signal and U-ref) are updated every 1.3 seconds. To see more information one has to activate the heart-beat window. This can be done by a click on the **show Decoder status (heart-beat)** button or by the menu point **other - decoder status**. The heat beat window shows the signal levels, header errors, data errors and the number of received frames during the last 5 minutes in 1.3 second increments.

I implemented this function for debugging during R&D. I suggest disabling this function during the normal use of the decoder.

6.7.9 Near Targets

(except GNS5890) The widely used miniadsb-receiver has problems to produce correct output signals, if the antenna signal is large. As a result the decoder may not be able to detect aircraft at short ranges, but can see aircraft at medium and long range. This is a bug in the receiver and should be fixed by receiver modification. However, the decoder firmware 9 contains a special mode to reduce this effect. If it is activated, then the comparator threshold is periodically increased (for short range detection) and then set to normal again.

I don’t have the test environment to check the efficiency of this mode, but as a side effect the framerate of the decoder will be decreased. I don’t suggest using this mode by default. But if e.g. somebody is located close to an airport, this mode may be worth a try.
To activate this mode use the menu point **adsbPIC - near targets**.

6.7.10 I2C-Support

(since firmware 11, except GNS5890) Some people use satellite-TV-tuner as frontend for the adsbPIC. Such tuners often have to be controlled via I2C-bus to tune it to 1090 MHz. adsbScope together with adsbPIC can send a sequence of bytes via I2C to an attached tuner. This happens automatically after a connection to the decoder was established by adsbScope. The user has the choice between a standard byte sequence (C2-30-8C-8E-00; for Alps BSJE3-159A-tuner) or a user defined byte sequence. All settings are done via the menu point **adsbPIC - I2C** (available after “connect” only.) The selection (no I2C / standard code / user defined

code) is stored permanently and used every time a connection to the decoder is established.
The menu point **adsbPIC - I2C - setup individual code** opens an editor window.

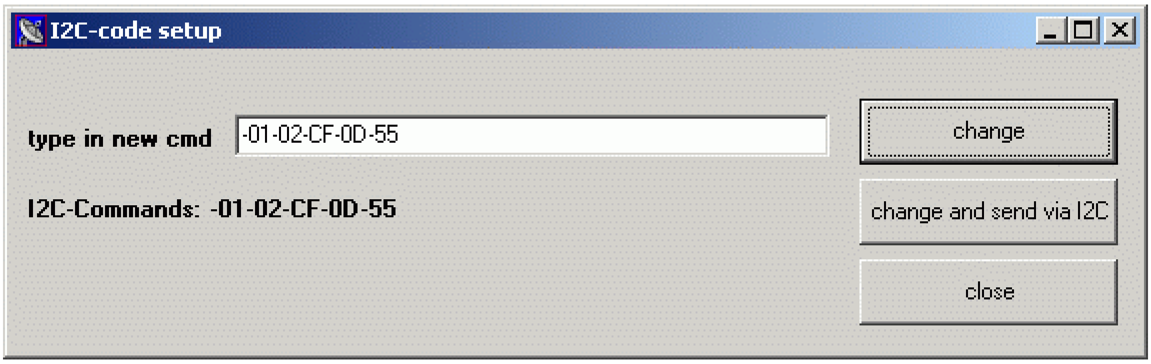


Figure 26 I2C-Editor

After the label **I2C-Commands** the momentary selected individual code is shown. Above this label one can see an editor line to type in a new code. Initially this line contains a copy of the actual sequence.

The new code is typed in as a sequence of hexadecimal values. To separate the individual bytes spaces or minuses (or whatever) can be used. After a click on the **change**-button the new code is red and shown behind the **I2C-Commands** label.

The button **change and send via I2C** will do the same and finally send the new code immediately to the tuner. This function may be useful to test out new byte sequences.

The new code sequence is permanently stored.

6.8 Networking

The software can distribute decoded data and received raw data via network/internet. The default network settings can be changed via the Network window. (Menu: **other - Network - network setting**).

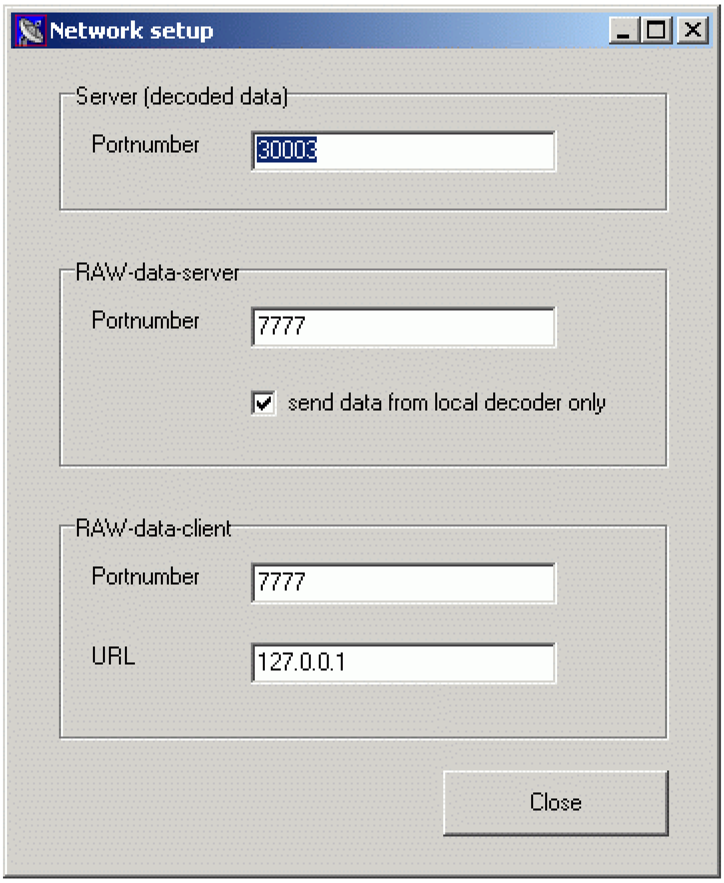


Figure 27 network setup

6.8.1 Server (decoded data)

The software contains a server at port 30003. Via this server the received and decoded information can be forwarded to multiple clients.
The output format is (hopefully) compatible to RadarBox and SBS-Basestation. However, I was not able to test this in detail. Any comments/feedbacks are welcome.
At program start the server is not active. To start the server one can click the left of the both gray network-server-icons (right of the OSM icon). The server is started, a comment is written in the upper log-window and the icon gets colors. Now clients can connect the server at port 30003. To deactivate the server just click on the icon again.
The menu-point **other - Network - Server (decoded data) active** can be used to start or stop the server too.

To test the running server you can start your internet-browser on the same computer and type in http://localhost/30003/.

To activate the server at program load, the user can use the command line parameter ‚**/S**'.

If the program is closed while the server is running, then the server will start automatically during next program start.

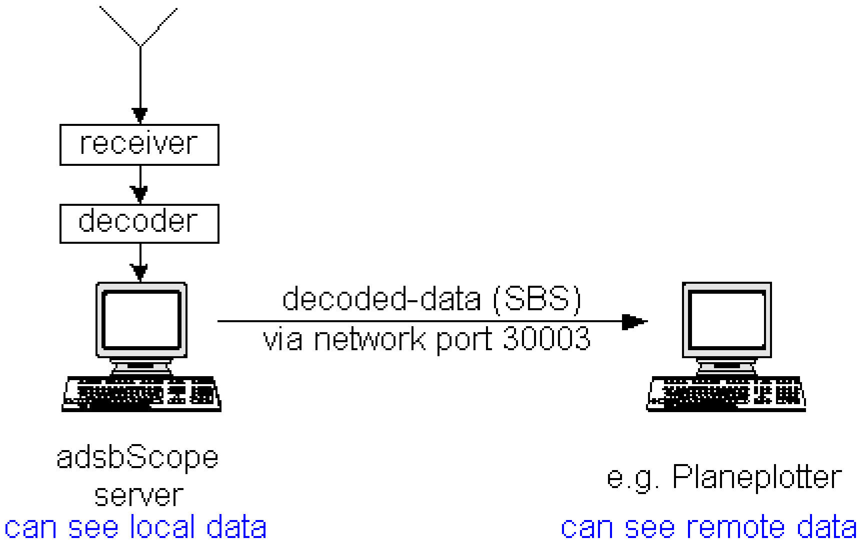


Figure 28 Server for decoded data

Aircraft heading is reported in degrees. Some clients require values from 0 to 360, other clients accept values from -180 to +180. By default AdsbScope sends data in the range from 0 ...360, but this can be permanently changed by the menu point **Config - Heading**.
The selected format will be used by the server, in the table of detected aircraft and in the aircraft labels at the graphic display as well.

6.8.2 RAW data Server

The software can send raw decoder data via network. By default port 7777 is used, but this can be changed permanently via the network-setup.

To start the server one can click the second gray network-server-icon (right of the OSM icon). The server is started, a comment is written in the upper log-window and the icon gets colors. Now clients can connect the server at port 7777. To deactivate the server just click on the icon again.
The menu-point **other - Network - RAW-data Server active** can be used to start or stop the server too.

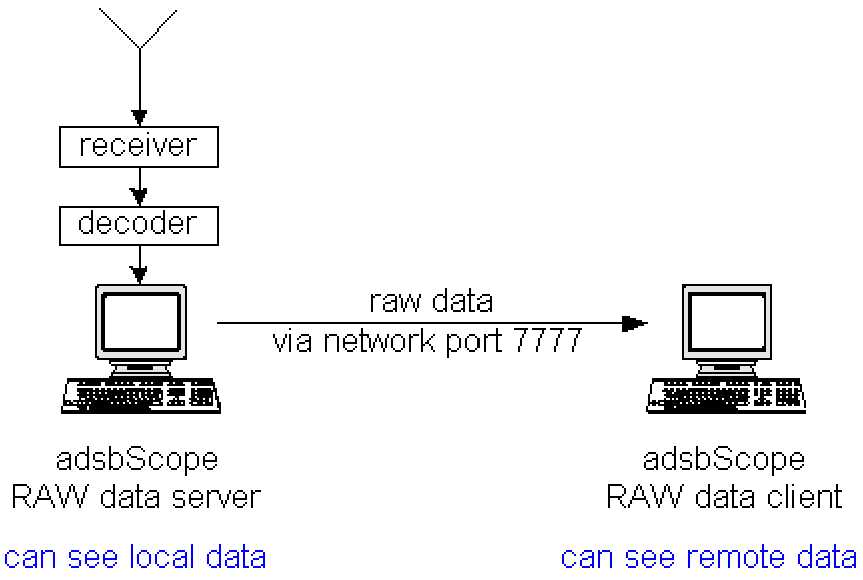


Figure 29 RAW data server and client

If the checkbox **send data from local decoder only** is checked, then only the data from the local decoder (connected via USB or RS232) is transferred by the RAW-data server. Then one can cross-connect two adsbScope-systems directly to exchange data in both directions. (Without this option the same data would ping-pong between the both installations.) If this checkbox is not checked then even data received via network (RAW-data client) is transferred. Thus adsbScope can work as relay for RAW-data or can combine RAW-data from other RAW-data-servers wit locally received RAW-data.

To test the running server you can start your internet-browser on the same computer and type in http://localhost/7777/.

To activate the RAW data server at program load, the user can use the command line parameter ‚**/R**'.
If the program is closed while the server is running, then the server will start automatically during next program start.

6.8.3 RAW data Client

The software can receive raw decoder data via network. You can connect to an active RAW-data-server via the menu point **other - Network - RAW-data client active** or by a click on the grey **RAW-data Client** button right of the server buttons.

By default URL and port number of http://hafik.zuban.name/~zuban/miniadsb/ is used. This is a FPGA-based adsb-receiver located in Prague, the capitol of the Czech Republic. After you connected to this server you can see nearly all aircraft in the Czech Republic airspace. Of course normally you would have to move the center of the graphic display to this location. For convenience the software will do this automatically.
Currently, there are only 4 available sockets in this system, so please do connect only for evaluation and do not connect for long time to keep this available also for others. The software will disconnect after 2 minutes, if you use this server.

Of course you can choose any other RAW-data-server, e.g. receivers with adsbScope-Software. You can change the IP-address and the port number of the used RAW-data-server via the network-setup. Connections to other RAW-data severs have no time limit.

It is possible to use a local decoder and the RAW-data-server in parallel. Then adsbScope will pool together the data from the local decoder and from the remote RAW-data server.

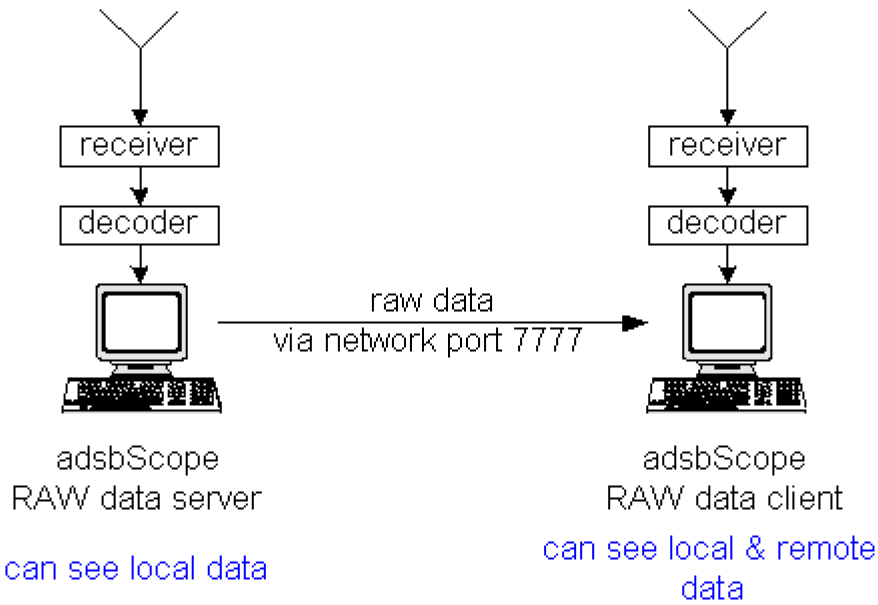


Figure 30 RAW data client with local decoder

If the connection to the server becomes interrupted, then the client tries to reconnect (every second) until the connection is re-established or the client is manually switched off.

6.9 Frame rate history display

A frame-rate-history window can be activated by the menu point **other - frame rate history** or by the **show frame-rate-diagram button** above the graphic display. The display shows a diagram of the frame-rate over the time.

The diagram can only be generated, after the software has received data for at least 2 minutes. Up to 3 values are shown in the diagram:

- in blue: number of frames per minute
- in green: average number of frames per aircraft and minute
- in red: comparator offset voltage

The frame rate is the number of frames received from the decoder per minute. Blue color is used to make the drawing of the frame rate and the scale for the frame rate. The frame rate depends on the number of aircraft that are in range of the antenna, and can vary from some hundred up to 6000 (or more). The scale is visible at the left side of the diagram and adjusted automatically dependent on the frame rate.

The number of frames received from a single aircraft is a good means to describe the quality of the decoder itself. This value is calculated and displayed in the diagram with green color. A fixed scale is used. The maximum is 500 frames per minute and aircraft. Typical values are 200 … 300 with all types of frames activated (decoder mode 2). If only DF17-frames are activated (decoder mode 3) then the theoretical limit is 120 and values above 50 are more then satisfying.

If an adsbPIC-decoder is used, then the diagram shows the selected comparator-offset voltage in red color. The scale for this voltage is adjusted automatically and shown at the right side of the diagram.

If the mouse cursor is moved over the diagram, then below the diagram the time, the frame rate, the frame rate per aircraft and the comparator offset voltage of the selected point of the diagram are shown.

6.10 Log-Function

The Log-functions of adsbScope are under development. More advanced functions will become part of future software versions.
The menu point **other - Log** opens the following window:

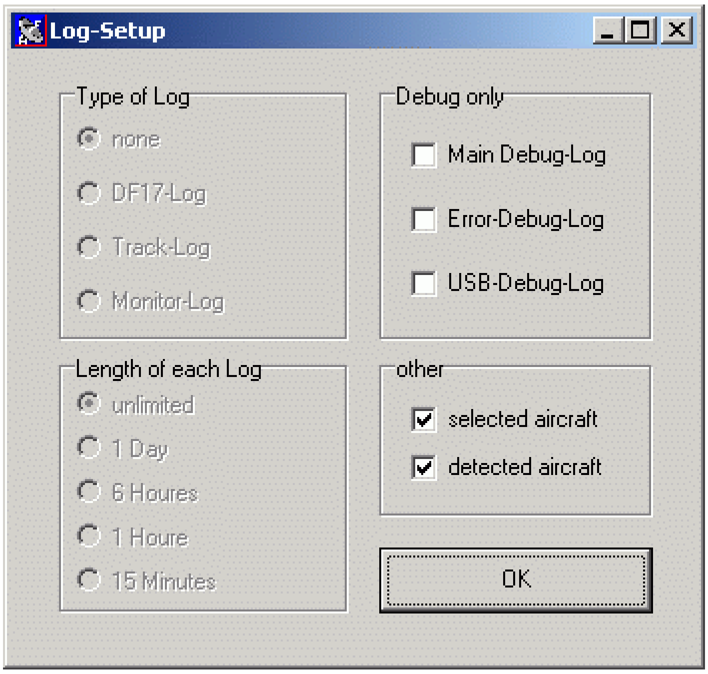


Figure 31 Log-Setup

At the moment only the both options **selected aircraft** and **detected aircraft** can be controlled in this window.

6.10.1 Selected Aircraft

If the **selected aircraft** option is activated, then every time the user clicks with the mouse on an aircraft in the graphic display basic data of this aircraft is written into a log file. The name of the log-file is log\airframesselect-yymmdd.txt while yymmdd stands for date. For every day a new file will be created.

6.10.2 Detected Aircraft

If the **detected aircraft** option is activated, then every time a new aircraft is detected or a tracked aircraft was lost basic data of this aircraft is written into a log file. The name of the log-file is log\airframesseen -yymmdd.txt while yymmdd stands for date. For every day a new file will be created.

6.10.3 Debug-Logfile

The software can write all received and decoded data into a logfile in subdirectory /log. This option is deactivated by default but may be useful for debugging or detailed analyze of received data. The logfile contains more information then the normal outputs of the software.
To activate the log-function, the user can use the command line parameter ‚**/L**'. The Log-Setup-Window can NOT be used to activate this function, but it shows if this function was activated. (**Debug only - Main Debug-Log**)

The logfile name is **logfile-YYMMDD-HHMM.txt**, where YYMMDD is the date of the day and HHMM is hour and minute of program launch.

6.10.4 Logfileplayer

The Logfileplayer is a simple tool to replay debug-log-files (and only debug-log-files!). It loads debug-log-files and acts as RAW-data-server to sends the contained data via network port 7777. adsbScope can receive this data and treats' the data like life data.

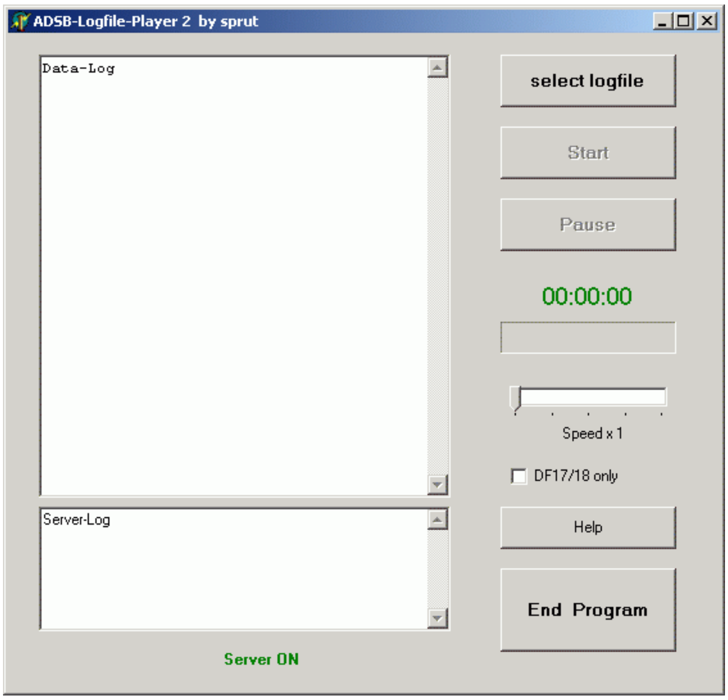


Figure 32 Logfileplayer

I developed this tool not to analyze log files but to debug my software. However, some users may find this tool helpful.
After program-start click on the Help-button to read a short user instruction.

The speed button can be used to multiple the replay speed. But this may cause big problems for adsbScope to determine the coordinates of aircraft. Probably aircraft will be shown at wrong locations or will jump around.

6.11 Watch list

The software contains a watchlist for up to 15 airframes (targets). If any of these airframes is detected, then its position is written into a special log file every 5 seconds, until the contact to this airframe is lost.
A click on the watchlist button (hand pointing up) or at the menu point **other - Watchlist** presents the watchlist-window.

It contains a table with 15 rows for up to 15 monitored airframes.

Aircraft can be identified by the callsign or by the 24-bit-ICAO-identifier.

The first column of every row shows the callsign (**Flight**) of the airframe. To change the callsign simply click on the cell in the table. Then it can be changed in the “Flight”-editor line below the table.

The second column of every row shows the 24-bit-ICAO-identifier (**ICAO24**) of the airframe. To change these numbers simply click on the number. Then it can be changed in the “ICAO24”-editor line below the table. If the software knows the Registration-code of the airframe, then it is shown in the “Reg”-column.

The **active**-column is used to switch on or off the monitor function for each target. This is done by a simple mouse click into this field of the table.

The **present** column is used to show the presence of an airframe that is monitored. If an aircraft is present, then the last column (**plane No**) contains the number of this aircraft from the table of aircraft in the main program window.

The **info** column is used to activate an acoustical information (a beep sound) if the airframe was detected. A click on the cell toggles between on and off.

There is no need to keep the watchlist-window visible the whole time, because detected targets are listed in the right side of the status line of the main program window too. If at least one monitored aircraft is detected, then there a label is visible:
Watch=Table: X=Y A=B ….
Each pair of numbers represents one target. The first number (X and A) is the target number in the watchlist table. The second number (Y and B) is the target number in the list of detected aircraft from the main program window.

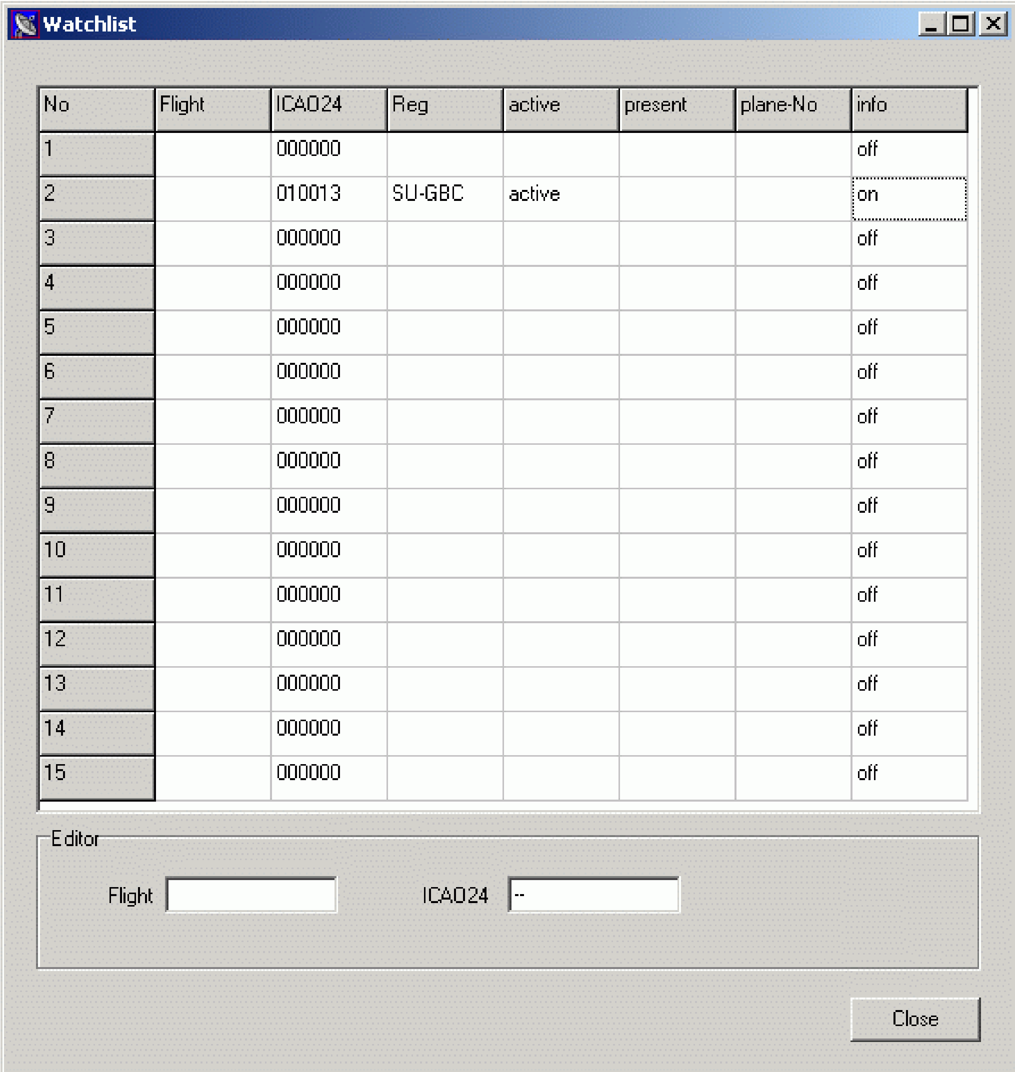


Figure 33 Watchlist Window

The positions of detected targets are logged into a special log file in the log subdirectory every 5 seconds. The filename is **watchlog-yymmdd.txt**, where yymmdd stands for the date of the day. Only one file is generated for each day. The first entry of a target starts with a >>. While the target is present, then each entry starts with ==. The last entry for a target (lost contact) starts wit <<.

6.12 Aircraft select Filter

If the user likes to monitor one specific aircraft, then he can use the **filter** window. (Menu: **other - filter**). This window shows continuously only the data from one specific aircraft. The aircraft is selected by a click into the aircraft table, or by typing in the 24-bit ICAO-Identifier into the text field at the bottom of the filter window.

A blue progress bar above the data window shows the time since the last frame was received in one second increments. The maximum is 20 seconds.

A specific log file for this aircraft can be generated, but this part of the software is still a stub. At the moment only the raw-frame-data, the receive time and the Frame mode (e.g. DF17) is written to this log file.

The name of the log file is the combination of the 24-bit-ICAO-identifier, date and time. The file name **4840D4_2010_10_07_17_00_27.txt** is a log of the airframe 4840D4. Recording started at the 7th October 2010 at 17:00:27 local time.

6.13 The Aircraftmanager

The software tries to identify all detected aircraft by help of the 24-bit-ICAO-identifier. But the database of the software (**/extra/icao24plus.txt**) knows only about 110000 aircraft.
All unknown aircraft are listed in the file **/extra/airframesunknown.txt**. Every line in this file contains the 6 symbols long 24-bit-ICAO-identifier and the origin of one unidentified aircraft. If the origin can not be predicted, then the label UFO is used as origin.

6.13.1 Manage unknown Aircraft

The aircraftmanager (AM) can be used to add new/unknown aircraft to the database of the software. To start the aircraftmanager (AM) select the menu point **other - manage unknown aircraft** or click on the AM-Button (book with question mark).

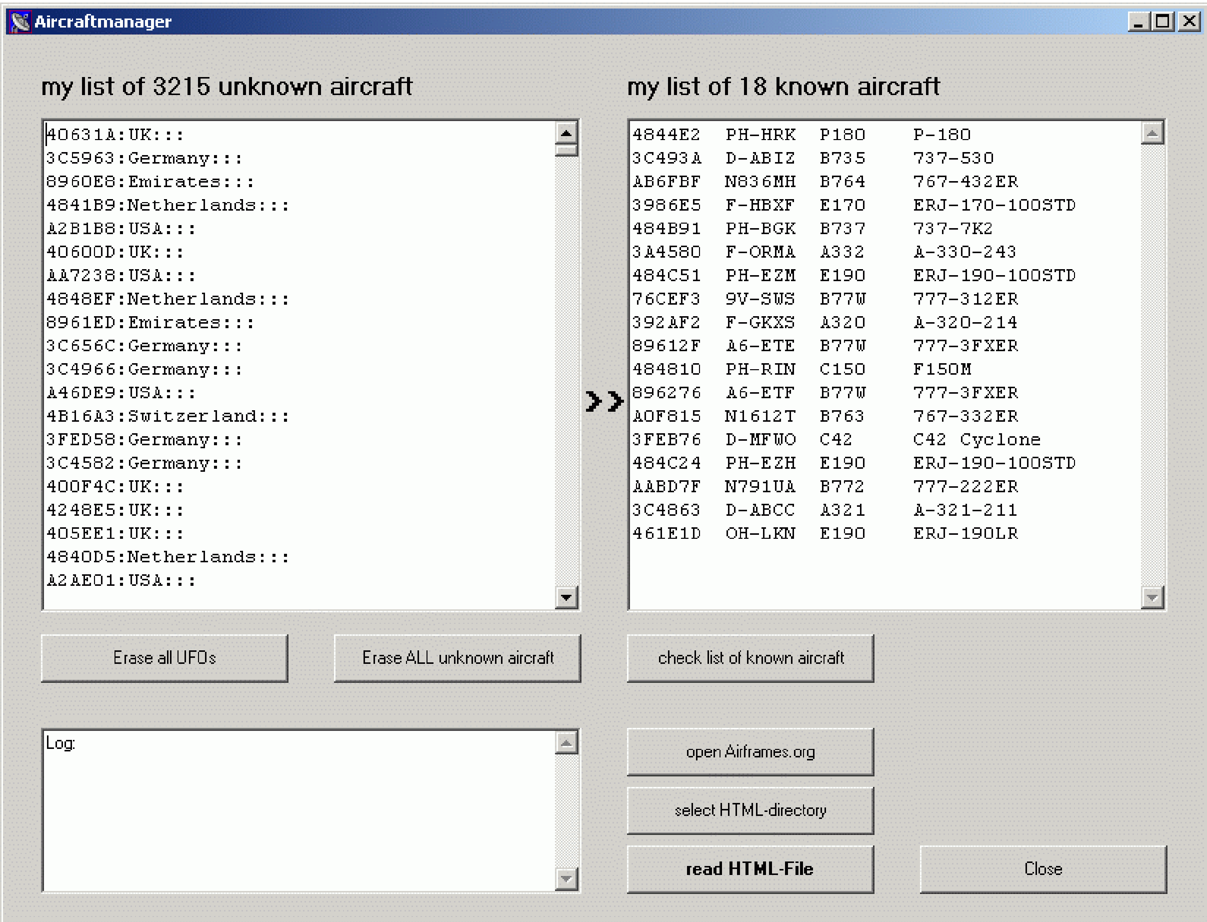


Figure 34 Aircraft Manager - Aircraft 40631A is unknown

The AM-Window contains 2 large text boxes. The left shows all detected unknown aircraft. The right one shows your list of aircraft, that you added to the software database. If you start the AM for the first time, then the right box will not contain data.

Aircraft with the origin UFO are mostly nonsense-data. They may be a result of DF11-dataframes with uncorrectable CRC-errors. Under normal conditions this UFOs can be erased from the list. The button **Erase all UFOs** can do this for you. The button **Erase all unknown aircraft** will erase the whole list after a warning.

You can identify an unknown aircraft and move it to the list of known aircraft. Then the software will be able to identify this aircraft in future. To identify the aircraft you have to use the webpage www.Airframes.org. This internet service knows nearly every aircraft. But the operators of this webpage don't like any automatic query of data. I will of course accept this.
Consequently you will have to look for an unknown aircraft on this webpage manually, but I try to make this as convenient as possible for you.

Step 1)
Click on the button **open Airframes.org**. Your web browser will start and open the webpage www.Airframes.org.

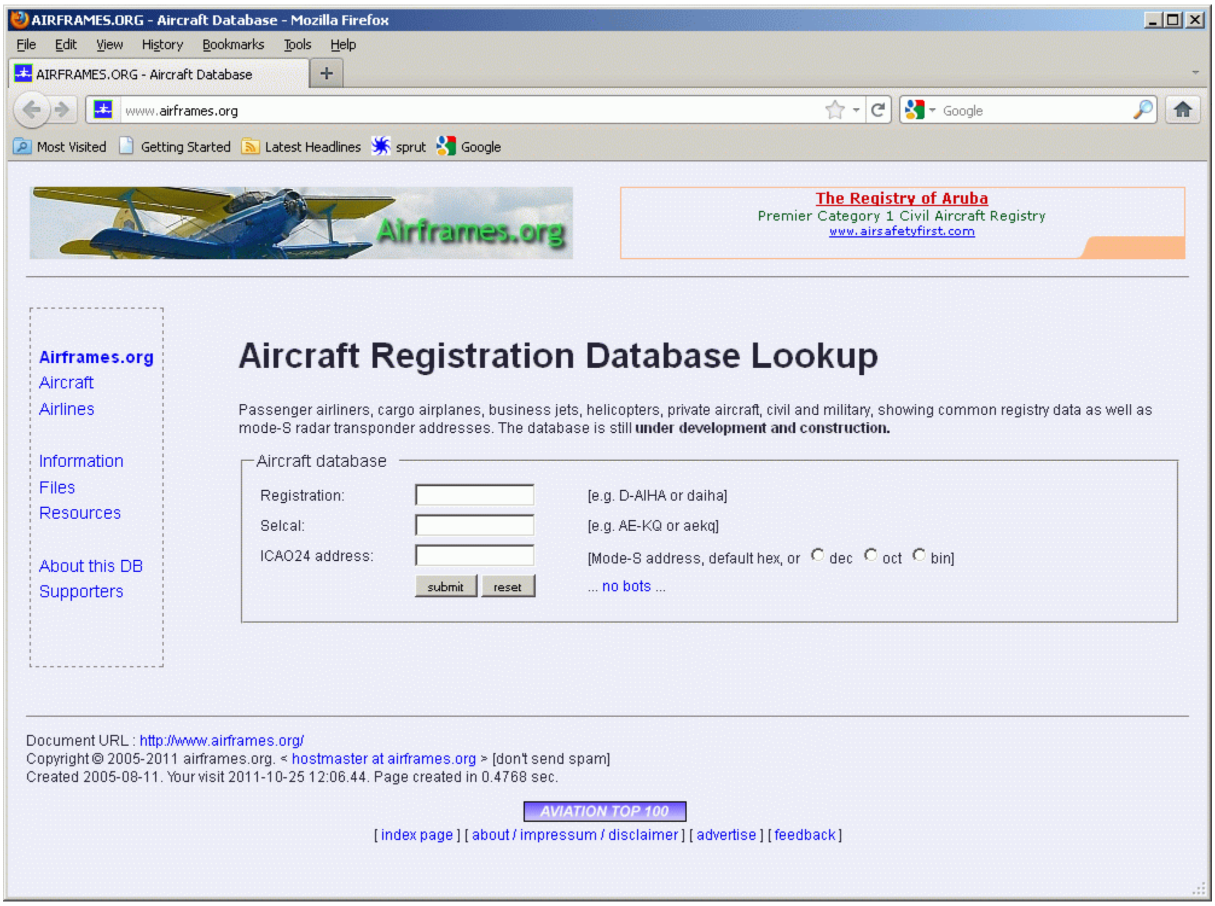


Figure 35 Airframes.org-webpage

Step 2)
Now copy the 6 symbols long 24-bit-ICAO-identifier of an unknown aircraft (everything in front of the first colon) into the field **ICAO24 address** of the webpage and click on **submit**.
The webpage reloads and will now contain a table with data about this aircraft:

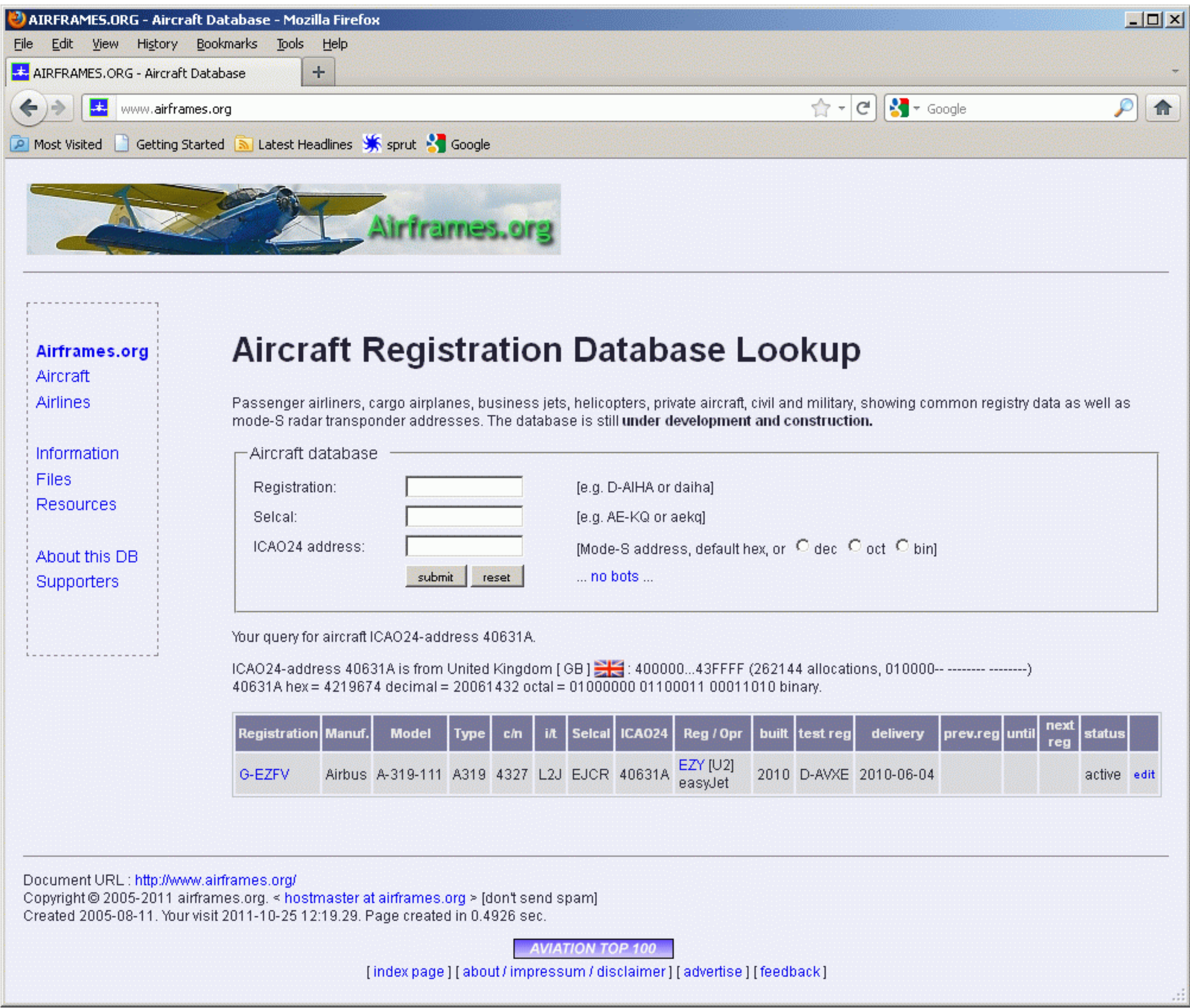


Figure 36 Webpage with data

Now save the webpage on your computer (**File - Save Page As…**). Don't change the name of the file (AIRFRAMES.ORG - Aircraft Database.htm) and look into which directory the file is saved. By default this will be **my Documents - Downloads**, but you can change this.

Step 3)
Now go back to the AM and click on **select HTML-directory**. A file-search-dialog will pop up, look for the saved file and click on it and click on **Open**.

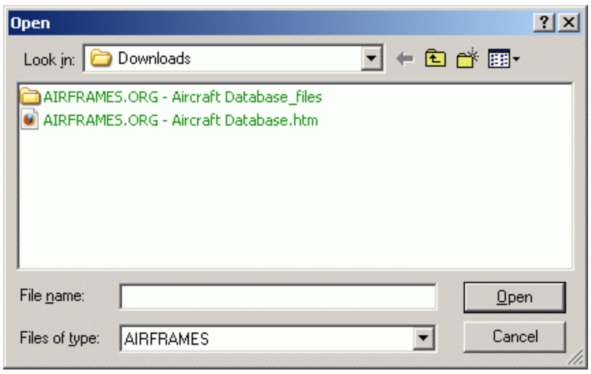


Figure 37 Select the HTML-directory

The software will memorize the directory of this file, and will in future look inside this directory for *.htm-files.

Step 4)
Finally click on the **read HTML-file**-button of the AM. The AM reads the saved *.htm-file and extracts the aircraft-information. The unknown aircraft is removed from the list of unknown aircraft, and added to the list of known aircraft. The *.htm-file will be erased.

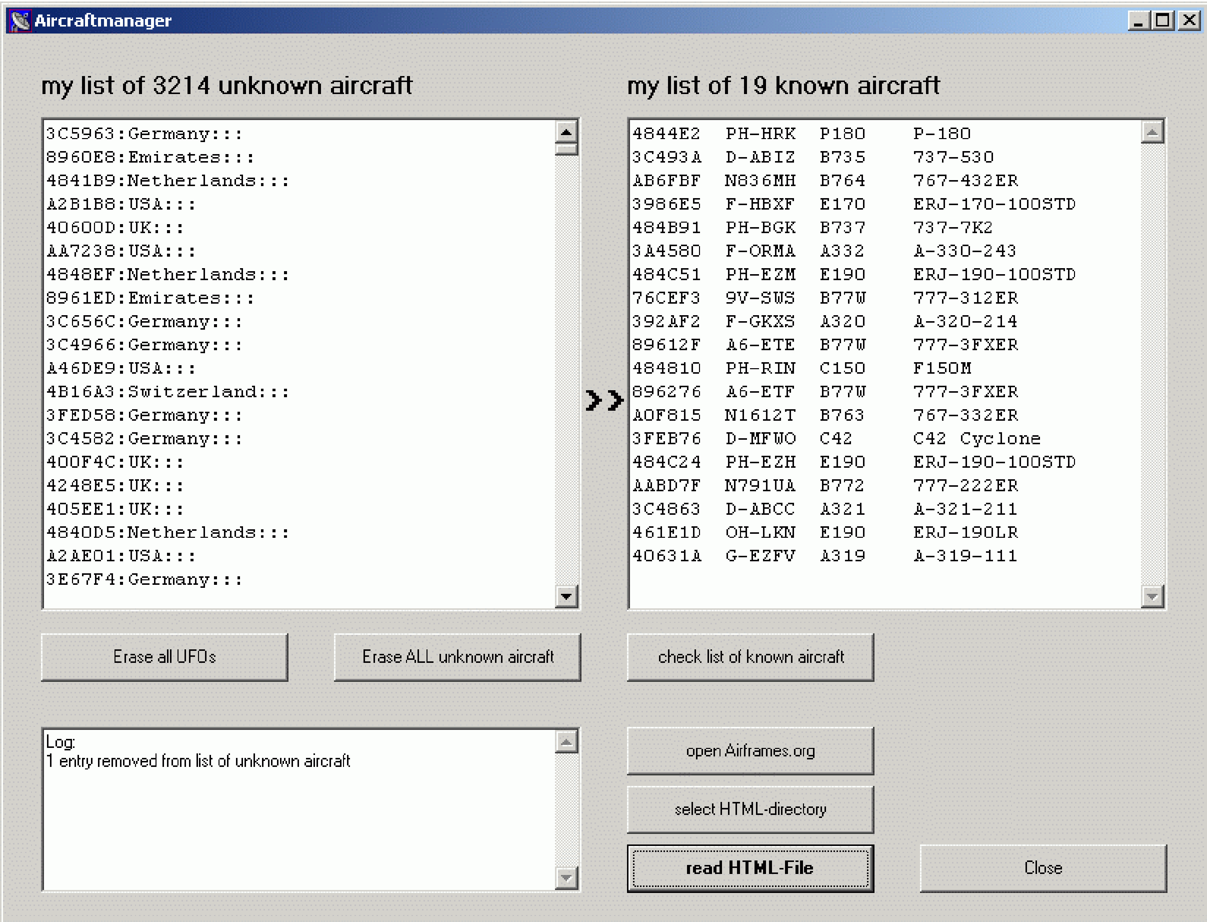


Figure 38 AM - Aircraft 40631A identified

To identify more aircraft **repeat the step 2 only**. The software will monitor the content off the download-directory and import data from all saved AIRFRAMES.ORG - Aircraft Database.htm-files until the Aircraft-Manager-window will be closed.

6.13.2 Update the Main-Aircraft-Database

The button “download new aircraft-database” can be used to update the main aircraft-database of the software. adsbScope will download the newest icao24plus.txt-file from the internet. After a successful download the date and time of the new database will be shown in the log-window.
If an aircraft is contained in the main-database and in the local list of known aircraft as well, then the software will prefer the data from the local list of known aircraft.

6.14 Heading

The heading of the aircraft is reported in degrees. The user can choose between values from -180º to +180º or from 0º to 360º degrees at the menu point **Config - Heading**. The software will use the selected format in the table of detected aircraft and for the server function.

6.15 Init-File

The software stores information (especially about user selected values and settings) is the file **inifile.txt** in the program directory. Do not try to edit this file manually!

If this file is erased while the program is <u>not</u> running, then the program will use default values at the next program start.

6.16 Command Line Parameters

Some command line parameters can be used during program launch.

- L - activate debug-Log-function
- S - activate server for decoded data
- R - activate RAW-data server
- A- auto-connect to decoder

To activate the debug-log-function, the user can use the command line parameter ‚**/L**'. The generated log-file is for debug use only!

To activate the server for decoded data (port 30003) at program load, the user can use the command line parameter ‚**/S**'.

To activate the RAW-data server at program load, the user can use the command line parameter ‚**/R**'.

The command line parameter '**/A**' (auto-connect) will force adsbScope to connect automatically to the decoder and to start data processing at program launch. This works only, if during a previous run of adsbScope this decoder was used and the COM-port-number of the virtual COM-port was not changed.
After this parameter was used a single time, adsbScope will always connect to the decoder and start data processing, even if this command line option is not set anymore! To deactivate this "auto-connect"-feature, the user has to uncheck the menu point **adsbPIC - connect at start**.

If you like to use command line parameter, then create a shortcut to the software: Pull the program icon on the Windows desktop with the right mouse button pressed. Then select **create shortcuts here**.
Click on the shortcut-icon with the right mouse button and select **Properties**. Then you can select the **shortcut** card rider (default) and add the parameter at the end of the **Target** line.

6.17 How to remove the software

To remove the software the user hast to erase its directory with all files and subdirectories. The software doesn’t writes into other directories or into the registry.

6.18 Known problems - Tips and Tricks

1)
Sometimes an **out of range** error message window message popped up during tests. The message window had to be closed before the program accepted user inputs again.
I really had a hard time to identify the root of this error. I am not sure, that the problem was permanently fixed. Please report, if you see this error message.

2)
If the software needs to long to start up, then check which maps are loaded (checked in the load maps menu). Too many maps delay program launch.

3)
If the software is not able to connect to the decoder, but you know, that all settings are correct and the decoder was operational before, then you have to disconnect and reconnect the USB-cable. This may happen, if the computer was in S4-sleep mode.

4)
If the program window is not visible on your monitor after you started the program, then close the program by help of the task manager and erase the **initfile.txt** before you start the software again.
Generally: if you “messed up” some settings and you erase the initfile.txt after the program was terminated, then the software will use save default values at the next program start. (Your individual settings are lost.)

5)
If you like to use command line parameter, then create a shortcut to the software:
Pull the program icon on the windows desktop with the right mouse button pressed. Then select **create shortcuts here**.
Click on the shortcut-icon with the right mouse button and select **Properties**. Then you can select the **shortcut** card rider (default) and add the parameter at the end of the **Target** line.

6)
The correct supply voltage level is important for the function of the miniadsb-receiver. If you read my description carefully, then you know: I feed the receiver from the USB-voltage trough a serial connection of **two** silicon diodes. (one inside the decoder and one inside the receiver) The idea is to reduce the voltage to nearly 4 Volts.
If the USB-voltage of your computer is unusually low (< 4.8V) then a single silicon diode may be the better option.

7 Bootloader

The adsbPIC-decoder (and probably microADSB) contains a bootloader. From time to time a new version of the firmware will be published to fix errors and to integrate new features. The bootloader is a simple to use tool to load the new firmware into the PIC.

The bootloader is small software, which has to be programmed into a special area of the control PIC of the adsbPIC. To program it into the control PIC a PIC programmer is needed. This can be e.g. a Brenner5 (with Windows-Software P18) or a Brenner8. The bootloader is available in a separate hex-file.

7.1 Start the Bootloader

For normal operation of adsbPIC the bootloader is not necessary, it is inactive. But if a new firmware version has to be flashed into the adsbPIC, then the bootloader has to be activated. There are two ways to do this

- Activation by the adsbScope-Software
- Activation with Jumper

If the bootloader is active, then LED1 and LED2 light up while LED3 is off.

7.1.1 Activation by adsbScope Software

If the menu point **adsbPIC - activate Bootloader** is used, then (after a warning is displayed) the bootloader is activated.

WARNING
If the bootloader is activated by adsbScope, then the firmware can not be used anymore. A new firmware has to be loaded into the adsbPIC before it can be used again as decoder.

7.1.2 Activation by the Jumper

If the activation of the bootloader by software is (by any reason) not possible, then a hardware backup solution can be used:

Unplug the adsbPIC-decoder from the PC and install Jumper JP1. If your version of the decoder don’t has a jumper JP1, then connect Pin 1 of the decoders control-PIC with Vss/Ground (Pin 8 or Pin 19). Now connect the adsbPIC with your PC.

The bootloader becomes active, and the Jumper is not necessary anymore.

If this is done for the very first time, then Windows may ask for a driver. In this case the Microchip MCD driver has to be installed.

HINT

The jumper will disable the firmware only temporarily. If no new firmware is flashed into the PIC and the jumper is removed again and the decoder disconnected and reconnected to the PC, then the old firmware becomes active again.

7.1.3 Upload new Firmware into the adsbPIC-decoder

Once the bootloader is activated you can load new firmware into the decoder by the software USBoot.
Start the software USBoot. You will see the following program window. (If a different program window with only one large button is visible, then the bootloader failed to start.)

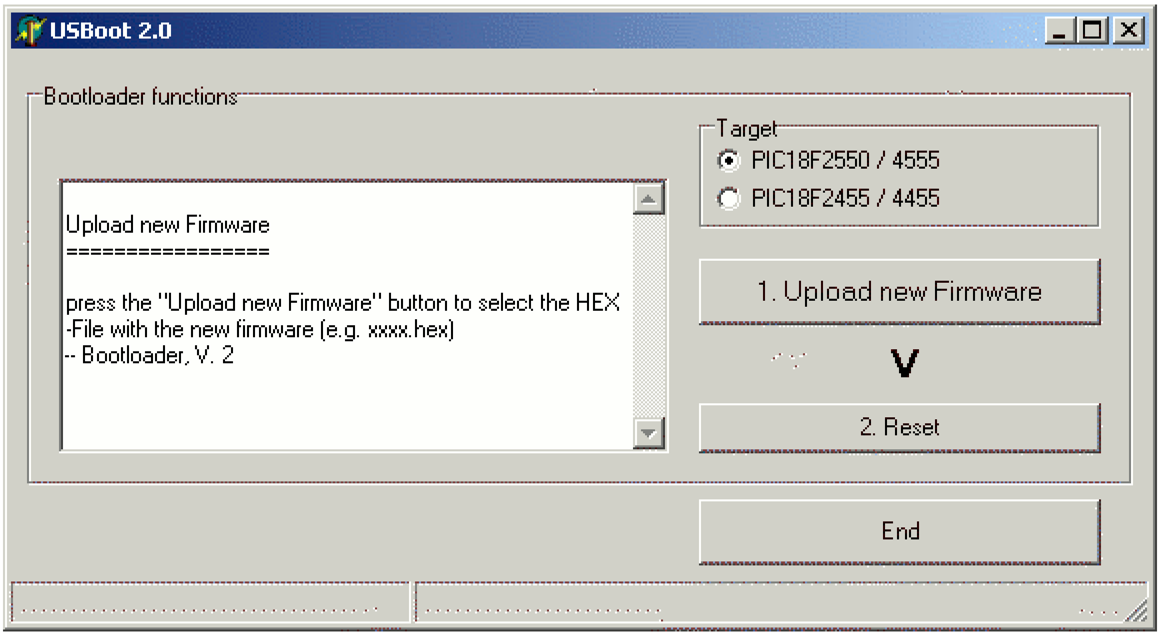


Figure 39 USBoot Software

Flashing the firmware is done in 3 simple steps.

First select the correct type of the controller PIC. The PIC18F2550 is selected by default. If a PIC18F2455 is used, then the **Target** setting has to be changed.

Then click on **Upload new Firmware**.
The software will present you a file selection window, to choose the right HEX-file with the new firmware. After a HEX-file was selected, it will be flashed into the decoder. A first blue progress bar will show this process of flashing. After that a second green progress bar will show the progress of testing.

If the jumper is still installed, then please remove it now.

A click on **Reset** will switch the decoder back into the normal adsbPIC mode. The new firmware is now running.

8 Schematics and PCB-Layouts

The decoder is not very complex and can be realized on a universal test board, but a printed board may be more reliable and looks much nicer. A printed board offers the possibility to use SMD-parts to minimize the size of the decoder.
Some possible layouts for printed boards are contained in the ZIP-file, and attached to this document.

8.1 Switches

My layouts and schematics contain 4 switches to control the work of the decoder in detail. However, the decoder supports up to 6 switches. Read chapter 7.2.3 to decide, if you need more then 4 switches.

8.2 Parts

The values of all resistors and capacitors are not critical. Any value can be changed by 25% without an influence to the function of the decoder.

The resistors R3 .. R5 control the brightness of the LEDs. There values should not be decreased to increase the LED-brightness. Large LED currents may have a bad influence on the analog parts of the decoder circuitry. If low-current-LEDs are used (2 mA types) then the values of the resistors can be increased to 2.2 kOhm.

The resistor R2 (at pin 5 of the microcontroller) has influence on the decoder sensitivity. It is a safety resistor, which should not be necessary under normal conditions. Its purpose is to safe the life of the miniadsb-receiver, if during tests somebody messed up the wiring or burned wrong firmware into the PIC. Any voltage accidentally fed into the miniadsb-receiver output would destroy the receiver.
After the correct function of the decoder together with the miniadsb-receiver was proved, then this resistor can be replaced by a wire to improve signal quality.

The crystal has to be a 20-MHz-type. Any other value would disable the USB-communication between decoder and PC.

If you develop your own board layout, then pay attention to the pin-numbers of the USB-connector. The order in the schematic is not identical to the physical order.

The diode D1 has to be a standard silicon type. Its purpose is to reduce the supply voltage level for the receiver. A shottky diode would not be able to do this.

The inductor (coil) can be a cheap one, but should have a low resistivity. If you don’t have one, then replace it with a wire.

The microcontroller can be a PIC18F2550 or PIC18F2455. Low-power-versions of these microcontrollers (PIC18LFxxxx) should not be used because of there clock frequency limits.

8.3 Non-SMD-Layout

The controller (DIL-housing) should not be soldered on the board. Instead an IC-socked should be used. Thus the controller can be programmed (with the bootloader-hex-file) in a normal development programmer (e.g. Brenner8) and then plugged into the socked.

8.4 SMD-Layout

The SMD-layout enables smallest dimensions, but the bootloader has to be programmed into the controller after it was soldered on the board.

Assemble the decoder and place all parts on the board except R8 and D1. Then connect the ICSP-cable to the board. All necessary controller pins are accessible by solder pads or connector pins. Now the bootloader can be programmed into the chip. After this was done, R8 and D1 can be soldered on the board. (The programming should be possible with R8 and D1 installed too, but without these parts the probability of immediate success is higher.)

For resistors and capacitors I prefer the 1206 size, if possible.
Small SMD-crystals are available in different sizes and shapes. Maybe you have to change the layout for your crystal.

8.5 RS232-Interface

The RS232-interface is an extra option of the decoder. You may decide to use RS232 as interface for ADSB-data, but you will have to use USB to upload the firmware! You can not burn the firmware directly into the chip by a programmer; the RS232-interface is only operational, if the bootloader is contained in the controller.

Consequently you will have to build up a normal decoder with USB-interface first and to upload the firmware via USB and bootloader. After this is done, you can add the RS232-interface hardware.

The RS232-interface-hardware and the decoder are connected via 4 wires (TX, RX, Vdd and Vss).

The decoder’s normal source for electrical power is the USB-cable. Without this cable attached to the decoder, a different power source has to be used. The supply voltage of 5 V (±5%) has to be attached between Vdd (+) and Vss (-). The decoder together with the miniadsb-receiver should not consume more then 50 mA.
The layout for the RS232-Interface-hardware contains a connector for the power-supply.

9 Glossary

ADS-B
Automatic dependent surveillance-broadcast.
This is a cooperative surveillance technique for air traffic control. An ADS-B-equipped aircraft determines its own position and periodically broadcasts this position and other relevant information to potential ground stations and other aircraft with ADS-B-in equipment. (http://en.wikipedia.org)

Antenna
Converts electromagnetic waved into electric signals. For best results it has to have the right center frequency (depends on its physical size), polarization (depends on antenna orientation) and impedance (depends on design). For ADS-B the antenna

- has to be tuned to 1090 MHz
- has to receive vertical polarized signals
- has to have an impedance of 50 Ohms (for the miniadsb-receiver)

Atmel
This is a family of microcontrollers. (also known as AVR-microcontrollers) Such microcontrollers are used in some decoders.

AVR
This is a family of microcontrollers. (also known as Atmel-microcontrollers) Such microcontrollers are used in some decoders.

Beast
The Beast is a combination of 1...4 (modified) miniadsb-receiver(s) and an FPGA-based decoder. It is available with USB, Ethernet or Bluetooth interface. adsbScope supports the USB-variant of the Beast.

Bootloader
This is a little software that stays inside a device (e.g. a decoder) to load (flash) new firmware into the decoder without a special programming hardware.

Comparator
It converts the analog video signal from the receiver into a digital video signal.

Decoder
Detects ADS-B-data (frames) inside the digital video signal from the comparator and send it to the computer.

DF
(download format) This is the type of an ADS-B-frame.

Frame
ADS-B-data is radiated in small packages. Such a package is called a frame. There are different types of frames radiated. The type is called DF (download format). The most valuable frame type is DF17, it contains aircraft coordinates.

Frames contain 56 bits (DF0 ... DF15) or 112 bits (DF16 ... DF31) of information. Every aircraft radiates some hundred frames per minute. Normally every flying aircraft radiates two DF17-frames per second.

GNS5890
The GNS5890 is an industrial produced combination of a (modified) miniadsb-receiver and the adsbPIC-decoder in a USB-stick-housing.

ILS
An **instrument landing system** (**ILS**) is a ground-based instrument approach system that provides precision guidance to an aircraft approaching and landing on a runway, using a combination of radio signals and, in many cases, high-intensity lighting arrays to enable a safe landing during instrument meteorological conditions (IMC), such as low ceilings or reduced visibility due to fog, rain, or blowing snow. (http://en.wikipedia.org)

Knots
Knots is the common unit to measure speed in aeronautic and shipping. One knot is equal to one nautical mile per hour or 1.8 kilometers per hour.

MLAT
Some aircraft don’t report their position via mode-S. But they can be located by multilateration. Multilateration requires cooperation of multiple receivers via internet. Decoders in such an MLAT-network have to add a precise time-TAG to every received mode-S-frame. Aircraft are located based on the different time of arrival of pulses at different receiver locations.
adsbScope is not MLAT-capable, but can send MLAT-data from the decoders into an MLAT-network.

NM
Nautical miles (NM) is the common unit to measure distances in aeronautic and shipping. One NM is equal to 1.8 kilometers. One nautical mile is the length of one longitude-minute at the equator. Consequently the circumference of the earth is 21600 NM (360 x 60).

NMEA
NMEA is a standard for interfacing marine electronic devices (e.g. GPS). It was developed by National Marine Electronics Association. Most GPS receivers understand the most popular NMEA format: NMEA 0183, and deliver data via this protocol.

OSM
OSM stands for open street map. It’s a free collaborative project to create detailed maps of the earth for free use.

PIC
This is a family of microcontrollers made by the company Microchip. Some decoders are using a PIC-microcontroller.

PWM

Pulse Width Modulation is used in my decoder to generate a reference voltage for the comparator

RAW-data
This is the unprocessed data that the decoder delivers to the computer. Normally this is the received frame-data as text-strings.

Receiver
It converts the 1090 MHz-electric signal from the antenna into an analog video signal.

STRM
SRTM stands for shuttle topographic radar mission. During an 11 day long flight in February 2000 the space shuttle Endeavour mapped the earth shape by help of special C/X-band radar. The data is released to the public.

TAG (timeTAG)
Here: the precise time tags, which decoders can generate. It makes it possible to measure the time of arrival of every data-frame with a precision of fractions of microseconds. This is necessary for MLAT.

10 Attachments

The latest versions of schematics and layouts are available from www.sprut.de.

10.1 adsbPIC - schematic

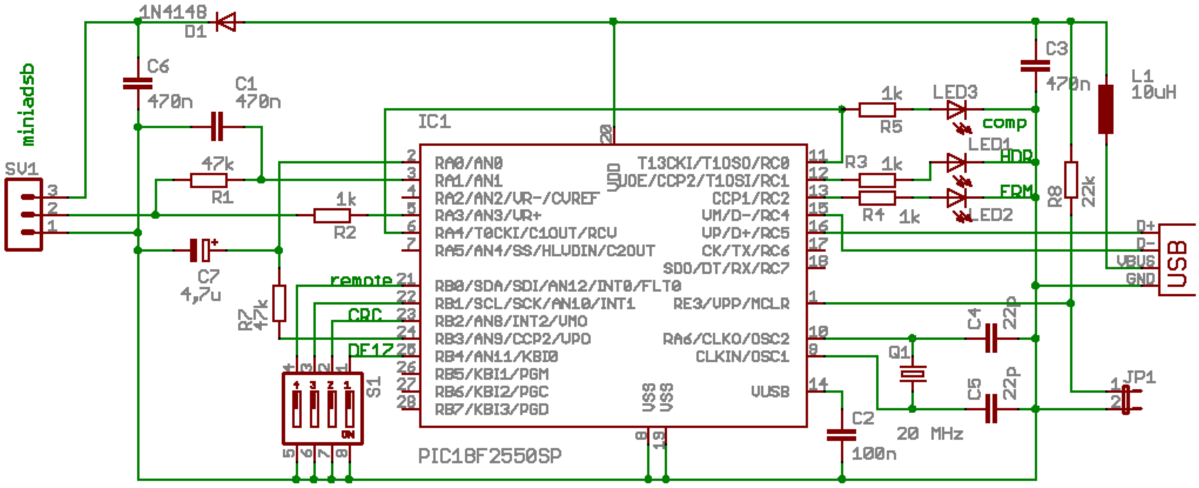


Figure 40 Schematic of adsbPIC (V. 2)

10.2 adsbPIC - component side

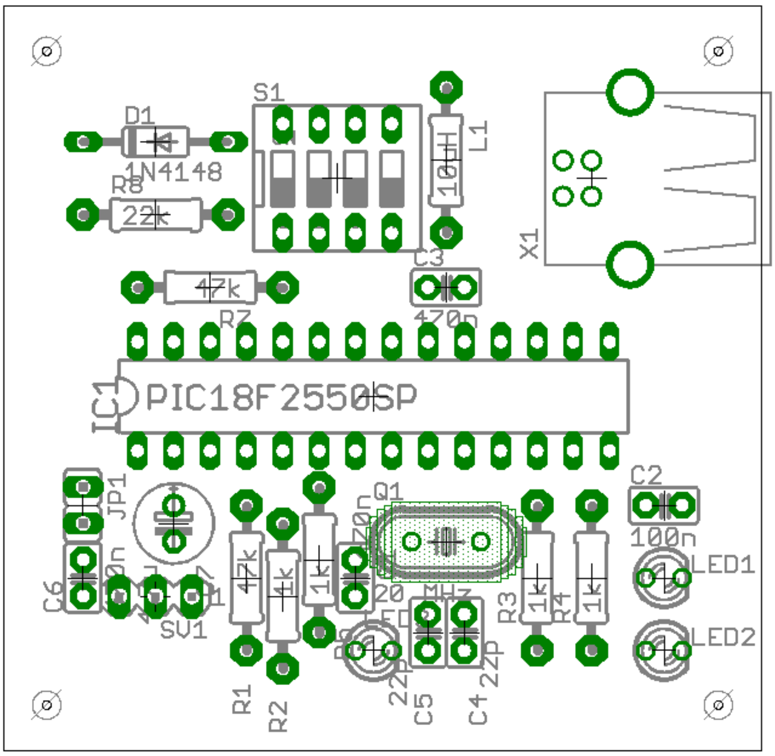


Figure 41 Component side of adsbPIC (V. 2 , not to scale)

10.3 adsbPIC - partlist

Partlist

Part	Value	Reichelt	Conrad
C1,3,6	470n	Z5U-2,5 470N	
C2	100n	Z5U-2,5 100N	
C4, 5	22p	KERKO 22P	
C7	4,7 uF	RAD 4,7/35	
D1	1N4148	1N 4148	
IC1	PIC18F2550SP	PIC 18F2550-I/SP	
L1	10uH	SMCC 10u	
LED1..3	LED3MM	LED 3MM 2MA RT	
Q1	20 MHz HC49U-S	20,0000-HC49U-S	
R1	47k	Metall 47k	
R2..5	1k	Metall 1k	
R7	47k	Metall 47k	
R8	22k	Metall 22k	
JP1	Jumper	SL 1x36G 2,54	
		Jumper 2,54 GL RT	
S1		NT 04	
SV1		BL 1X10G8 2,54	
X1	USB-B-H	USB BW	

SV1 - cut to the right size, I suggest to use a 4st pin as key
JP1 - cut to the right size

10.4 adsbPIC - schematic for RS232-option

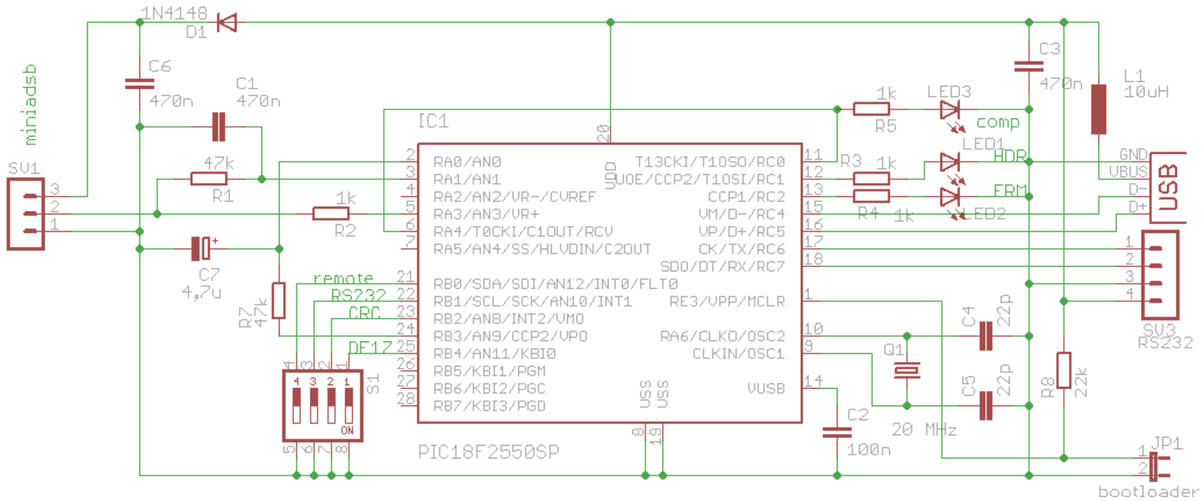


Figure 42 Schematic of adsbPIC with RS232-option

10.5 adsbPIC - component side for RS232-option

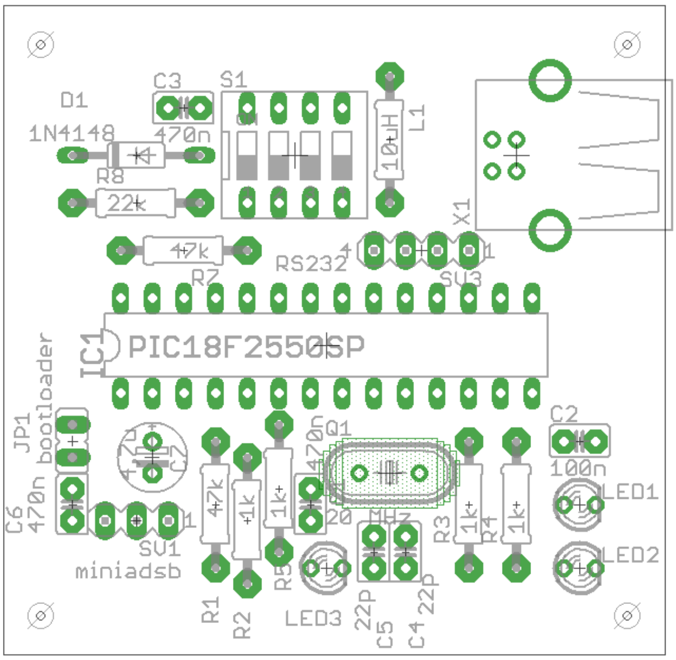


Figure 43 Component side of adsbPIC with RS232-option (not to scale)

10.6 RS232-driver schematic

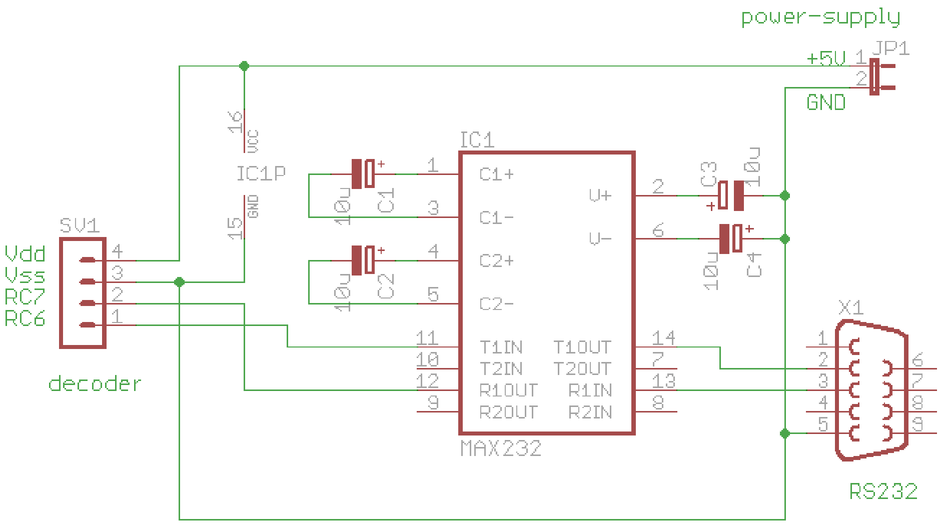


Figure 44 Schematic of RS232 driver

10.7 RS232-driver component side

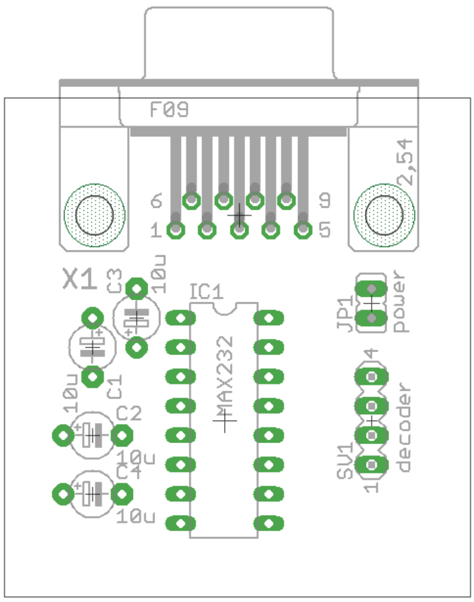


Figure 45 Component side of RS232-driver (not to scale)