Art. No. 66006, 66066, 66077 10/2016

Radarcape - High Performance ADS-B-Receiver with Embedded Linux

User Manual

jetvision.de

Radarcape:Contents

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Contents [hide]

Information for Radarcape Users

What is the Radarcape?

- About the Radarcape
- User Services and Interfaces
- Requirements for Operation
- Radarcape Specification
- Front and Back Panel
- Radarcape Online Demo

Getting Started with the Radarcape

- Radarcape Installation Guide (Getting Started)
- Direct Connection without DHCP / Static IP Address
- Restrictions of Use (Disclaimer)

Radarcape Software Maintenance and Configuration

- Radarcape Software Features (accessing data on internal webpages, TCP and serial ports)
- Radarcape Configuration Page
- Software / Firmware Versions (Release Notes)
- Hardware Watchdog
- Beaglebone White: Creation of a New SD Card
- Beaglebone Black: Restore eMMC After Failure
- Connectivity Failures
- Debian Support
- Help and Support

Supported Services and Software

- Build-in ADS-B Data Feeding to
  - Flightradar24
  - PlanePlotter Network
  - Planefinder

Information for Software Developers

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:About the Radarcape

Contents [hide]

About the Radarcape

After the big success of the Mode-S Beast we saw that there is a big demand for a more sophisticated unit with a special focus on

- remote operation
- standalone data processing
- network feeding
- low power consumption
- easy maintenance and SW update

and not at least

- enhanced multilateration

Application Features

The idea realized with the Radarcape is a combination of the Mode-S Beast and an embedded Linux board. Later are now available for a reasonable price and community support.

Example features

- Raw data streams in various qualities and with different filtering
- Web interface data access and maintenance
- The networking can support all kind of protocols you know. Currently there is only TCP support, but based on the Linux systems others are no problem. For example, we have already set up a network based on SSH tunneling which solves some firewall issues.
- There no longer are DIP switches, instead the configuration can be changed just using an web browser.

and more...

Not at least the Radarcape incorporates some improvements we've leared from the Mode-S Beast, further enhancing range and frame rate.

Multilateration

For enhanced multilateration, the Radarcape is equipped with a GPS synchronized clock with an accuracy in the nanosecond level. This is completly processed in the FPGA without any influence through the Linux system.

Currently the Radarcape does not provide Multilateration by itself but requires second

level software. However, an internal solution is under development and might be presented soon.

Network Feeding

There are sharing networks for aircraft data. Flightradar24 is using a branded version of the Radarcape as their device that feeds data into their servers. The Radarcape as delivered from us also contains a way how to share data with Flightradar24, either anonymously or with a sharing key provided by them.

Easy Usage

With its small size and the low power consumption the Radarcape is ideal for running all time without a PC behind and collecting data. The application SW on the Radarcape will output collected data in several ways, like

- raw data formats for second level software on PCs
- HTML tables for personal viewing or computer postprocessing
- KML files for mapping software, such as Google Earth

Properitary Software running on the Radarcape

The Radarcape will be open for your private enhancements and SW installations on the Linux part. We will provide all information required for such tasks and even ways how to write software under Linux, like interface descriptions, and even some installation guides for compilers and tools.

Radarcape:User Services and Interfaces

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Contents [hide]

Introduction to Radarcape Services and Interfaces

A Radarcape offers several services to a user, either for direct use or use with second level software.

All services can be used in parallel, only limited by the processing power of the Radarcape.

High Level Interfaces

High level interfaces are such which are for direct use by the operator without using special ADS-B software.

- **Aircraft List**

This is a list of the aircraft that currently are in your neighbourhood. For displaying you need nothing more than a web browser. The table can be sorted by any kind of column, like time, ICAO ID, altitude and more. It refreshes itself after a configurable time. Omit time and delete time are configurable as well.

For details see Software Features - Aircraft Table.

- **2D Map Output**

Nothing else than a simple web browser and an internet connection is needed in order to display 2D maps originated directly from the Radarcape's aircraft list. The same display may be known from other application SW. This display is ideal for quick remote observation.

For details see Software Features - 2D Display.

- **KML output (3D display)**

This output is designed for the use of KML capable readers like for example Google Earth. For the same aircraft that are displayed in the list, a 3D display of their tracks is generated. The display refreshes itself, which will result in a live display.

For details see Software Features - 3D Data KML Output.

- **Web configuration**

Configuration of various items is handled by a web interface. There are no longer DIP switches available. Still the FPGA configuration can be overwritten by high level software. The web interface is password protected, default password is "radarcape"

For details see Radarcape Configuration.

Medium Level Interfaces

Medium level interfaces are to be used by software.

- **Port 30003 data**

Port 30003 is a decoded interface in comma separated format which contains pre-decoded values. As the encoding of Mode-S and ADS-B messages is not so simple, this is the easiest way how second level software can access the data that is received by the Radarcape. The only bottleneck of this interface is its inefficieny with respect to processing power and data handling, so if you can avoid, please do.

For details see Software Features - Port 30003 Service.

- **deltadb.txt web page**

Similar to Port 30003, deltadb.txt can deliver pre-decoded data. It is serviced by the internal web server, so requestors will receive a list of changes in the aircraft list which have either happend since the last request or a time that can be given as parameter.

For details see Software Features - DeltaDB Service.

Low Level Interfaces

There is plenty of SW availabe that can decode Mode-AC, Mode-S and ADS-B by itself. For these, the Radarcape provides raw data in different qualities and quantitites:

- **TCP port 10002**

raw data as it is received by the FPGA board, includes all errors, broken frames and more. DF-11, DF-17 and DF-18 are CRC prechecked if selected in the configuration.

- **TCP port 10003**

Mode-AC, Mode-S and ADS-B raw data but all frame types in Mode-S/ADS-B have passed CRC checking. This is the recommended port if you want to route full data through a slow speed network connection.

- **TCP port 10004**

Mode-S DF-11, DF-17 and DF-18 only, all frame types prechecked. This is recommended in case that you only are interested in locatable aircraft and don't need every possible update.

- **TCP port 10005**

Mode-S frames of non-ADSB aircraft only. This is a special port for streaming multilateration clients.

For details see Software Features - Data Streaming to Network.

Build-in Feeders

The Radarcape comes with a pre-installed feeders for the platforms Flightradar24, PlanePlotter, Planefinder, and FlightAware. You may enter your private sharing key that you got from them in the configuration dialogue. The service can be disabled in the configuration menue.

For details see:

- Flightradar24 Feeder
- Planeplotter Feeder
- Planefinder Feeder
- FlightAware Feeder

Radarcape:Requirements

Radarcape Requirements

For normal operation a Radarcape requires

- a dry place with moderate temperatures
- AC power supply 110V or 230V, alternatively 5V/1A DC supply
- An ethernet connection with a DHCP server

Software Requirements

Aircraft Data Access

Since the Radarcape does not have it's own display, external software is required for displaying the aircraft data that it receives.

It depends on the users preferences how he wants to get access

- The internal aircraft list (database) and the Live 2D Map can be viewed with any kind of web browsers. The list can be sorted by various parameters (filtering in preparation) and will update automatically in a configurable time.
- KML files such as they are used by Google Earth will be provided and contain current positions and track information. Update rate (and later also contents) of these is also configurable.
- Support for 2nd level software is provided by access methodes to the raw received data or port 30003 protocol data. Such is beeing used by Planeplotter, Globe-S and more.

Note that all services are accessible in parallel without major interaction.

Maintenance and Debugging Software

As long as the user web interface does not support all configuration and maintenance tasks you may need to access to the internal Linux system. If necessary, we will describe all steps to be done here in detail, so you should not be afraid about that. We strongly recommend putty for this task, as it supports the back side serial port as well as networking SSH connections.

Antenna Requirements

The antenna should be placed as high as possible and as free as possible. Nearby metallic parts should be avoided or have a minimum distance of about 50cm. Big metallic surfaces around, for example metallic chimney casing, may cause reflections and degrade reception in some cases.

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Due to the high sensitivity of the Radarcape a preamplifier normally is not recommended. Only if your cable attenuation exceeds 6dB, you may consider using one. The cables we supply with the Radarcape do even with 10m length hold the so called "3dB rule" and can be used without doubts.

The antenna connector of the Radarcape is a standard SMA female connector on the back side.

Running without DHCP

For direct connection to a PC we will provide methodes how to install the Radarcape with a fixed IP address and without need of a DHCP server soon.

Radarcape:Specification

Radarcape

- The Radarcape consists of:
  - Beaglebone board
  - Radarcape Mode-S/ADS-B receiver
  - GPS module Trimble Resolution SMTx
  - Device case
- Power Consumption
  - 5 V external supply
  - Standard 5.5 mm/ 2.1 mm DC connector (plus inside, minus outside)
  - Electrical current with GPS (including antenna) typical 720 mA
- Dimensions and Weight
  - Length approximately 92 mm (110 mm with antenna connector)
  - Width approximately 80 mm
  - Height approximately 45 mm
  - Weight approximately 0.233 kg

Linux Distribution

The Radarcape operates an Embedded Linux. We do not guarantee stability with other Linux distributions than the delivered one.

Links

- BeagleBoard.org
- Trimble Resolution SMT web page

Radarcape:Front and Back Panel

Contents [hide]

Radarcape Front Side

- USB Extension Slot
- Power LED
- Mode-S LED
- GPS LED

USB Extension Slot

This is a USB type A connector. It is available for memory sticks or user extensions. It is fully supported by the internal Linux system.

Power LED

The power LED is illuminated green when power is applied to the back side connector.

Mode-S LED

The Mode-S LED is flashing green on each frame that becomes received from aircraft.

If no frames are received (e.g., no antenna connected), it flashes once per second in order to indicate working state.

GPS LED

The GPS LED is flashing green once each second exactly when the second changes. In case of GPS degradation, it occasionally flashes yellow.

Radarcape Back Side

- Mode-S antenna (SMA connector)
- GPS antenna (SMB connector)
- 100Base-TX Ethernet connector (LAN)
- USB Serial (USB mini-B connector)

Radarcape:Installation Guide

Required computer skills to execute these tasks: Beginner

Contents [hide]

Radarcape Hardware Installation

Plug the following connectors into the Radarcape:

- Mode-S antenna
- GPS antenna
- Ethernet/LAN cable
- Power cable

Connect the power supply with the electrical outlet.

Mode-S-Antenna Placement

The Mode-S antenna should be placed as free and as high as possible.

Make sure the Mode-S antenna has a conductive connection with protective ground. Otherwise, static electricity may cause damage to your Radarcape.

GPS Antenna Placement

The GPS antenna should be placed to a point with at least half of the sky in free sight, for example a window sill. Some users reported running the GPS antenna indoors. This is not guaranteed to work.

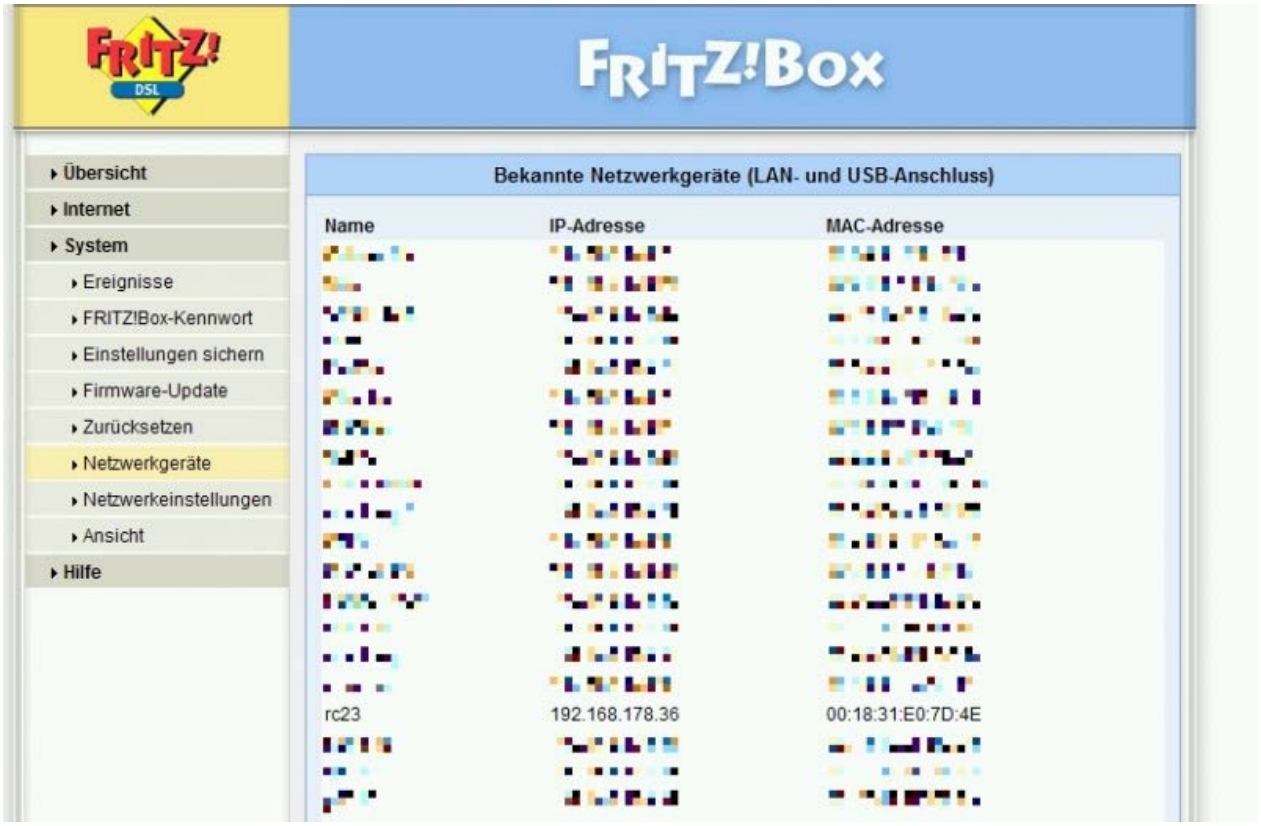
Hostname and Network Address

Network devices can be accessed in two ways:

- via IP address (e.g., *192.168.1.157*)
- via hostname (e.g., *radarcape*)

The *IP address* is provided to your Radarcape during startup via the http://en.wikipedia.org Dynamic Host Configuration Protocol] (DHCP) service. Usually, your Internet router provides the DHCP service in your network and assigns an IP address to your device. A fixed address can be configured via the Linux command line.

DHCP listing of a DD-WRT router showing hostnames RC70 and RC71 as 192.168.1.180 and 192.168.1.133



DHCP listing of a Fritzbox showing hostname RC23 as 192.168.178.36

The configuration of a fixed IP address is described this page: Radarcape:Fixed IP Address.

The hostname is a given name that is stored internally in your device (see */etc/hostname*). If there is no label on the back side of your Radarcape, the hostname is *radarcape*.

Note: Some router models require appending the *.local* to the hostname. In this case *radarcape.local* must be used instead of simply *radarcape*.

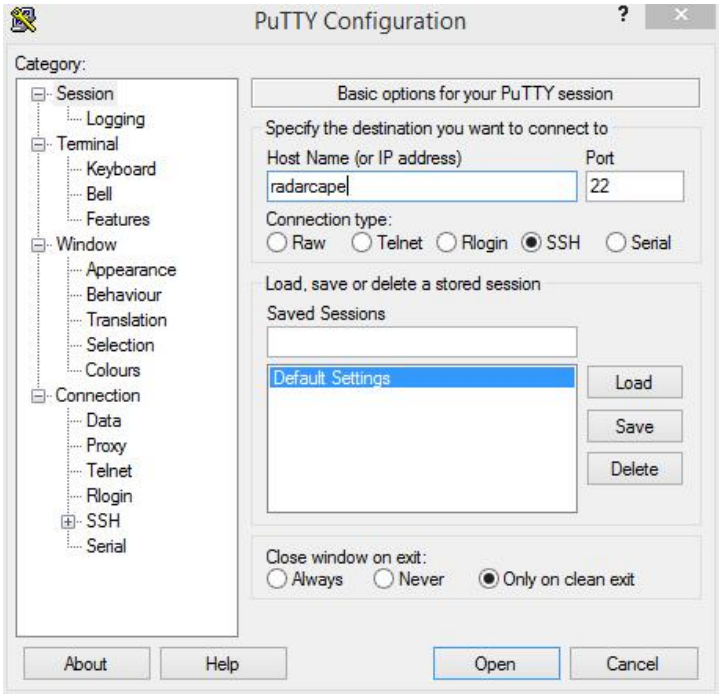
Linux Command Line Interfaces

For some special cases it is necessary to access the linux system console. There are two ways how to connect to the Linux system:

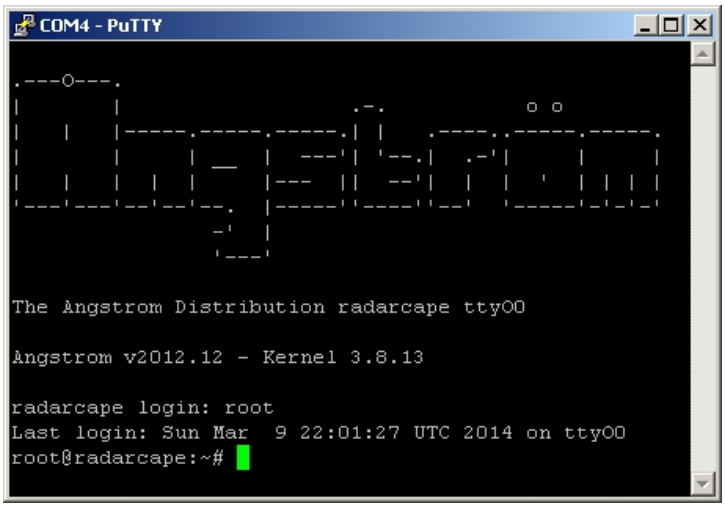
- SSH through the network (recommended)
- Back side Mini USB connector (only hardware versions with SN 1302-XXXX)

SHH Access to the Radarcape Using Putty

We recommend Putty as SSH client on Windows. Download the Putty executable from http://www.putty.org. Start Putty, set the Connection Type to *SSH*, and the Host Name to *radarcape*.



Click on the Open button and the login screen appears. Enter the username *root* and press enter when being asked for your password (or the password if you have set one).



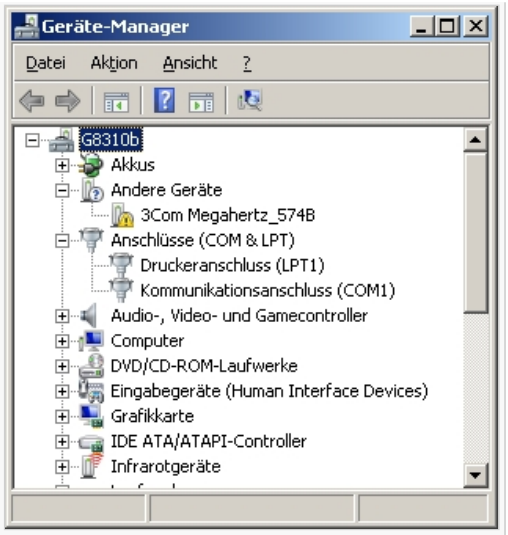
Accessing the Linux Console via USB

The Radarcape has 2 USB connectors:

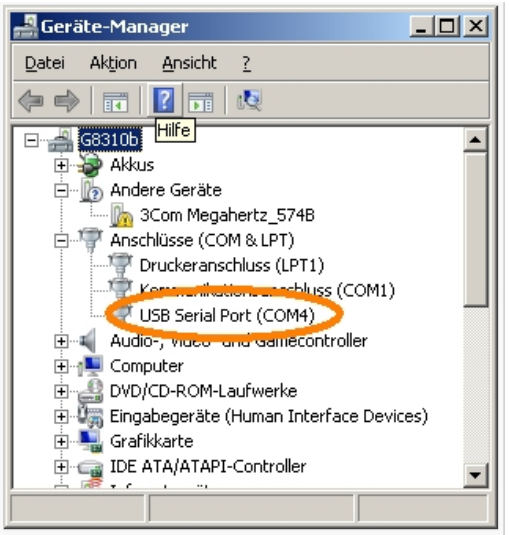
- A standard USB type A on the front panel, e.g., for memory sticks, DVB-T sticks or other extensions
- A micro USB connector on the back side. This allows you to connect to the Linux console via a virtual serial device

Note: The Linux console via the micro USB interface is only available on hardware versions with serial number SN 1302-XXXX

Take a USB cable and connect the back side USB. Check in the device manager that a COM port becomes created.



Device Manager view without a Radarcape console COM port



Device Manager view without a Radarcape console COM port

- If it does, skip the FTDI driver installation.
- If not, perform the FTDI driver installation and check if the COM port appears.

Parameters of this COM port are 115200 Bit/sec 8N1.

Install FTDI Drivers

Download the driver from the FTDI driver Web page (http://www.ftdichip.com/FTDrivers.htm) that fits to your system and install it.

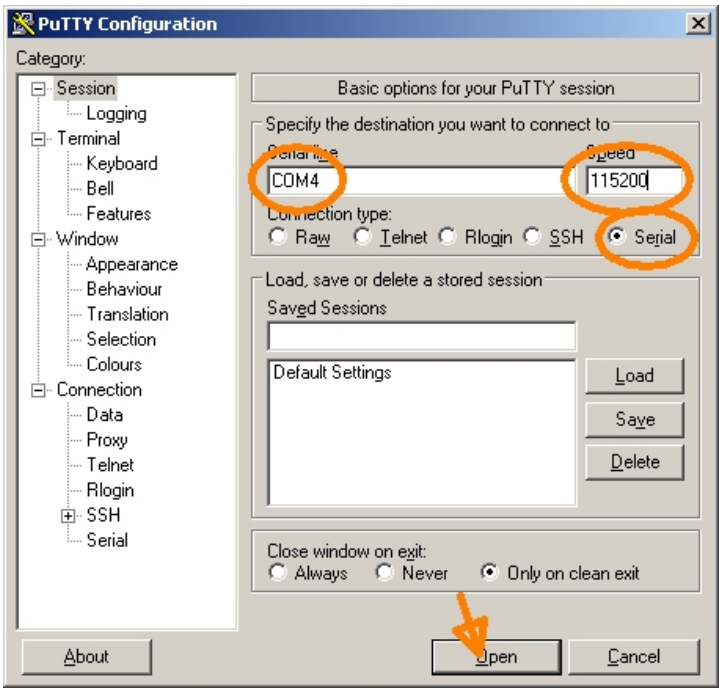
The COM port should appear in the device manager when the driver has successfully been installed.

Using Putty as Client for the Serial Console

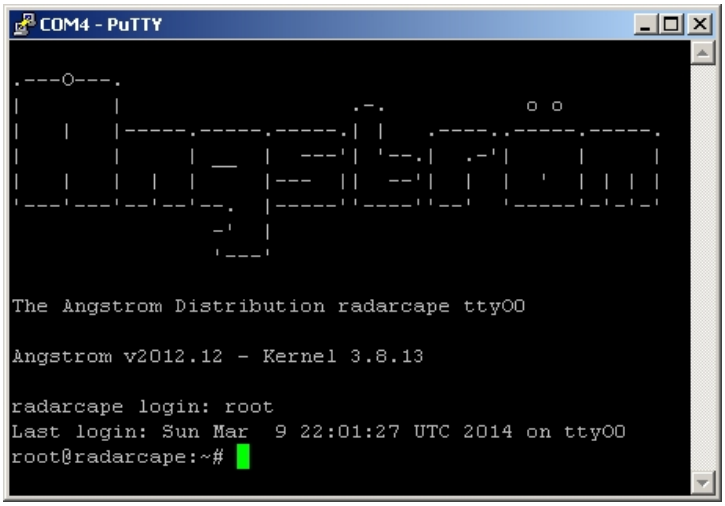
The software Putty can also be used to access the Radarcape via the virtual serial device.

Download the Putty executable from http://www.putty.org.

Start Putty and enter the above detected COM port, 115200 and serial into the startup menu:



Click on the Open button and the login screen appears. Enter the username 'root' and press enter when being asked for your password (or the password if you have set one).



Set/Change Root Password

In the default configuration, no password is set for the root user (administrator) on the Radarcape. We strongly advise you to set a password for security reasons.

Please remember your password as there is currently no other password recovery method than creating a new SD card image for your Radarcape.

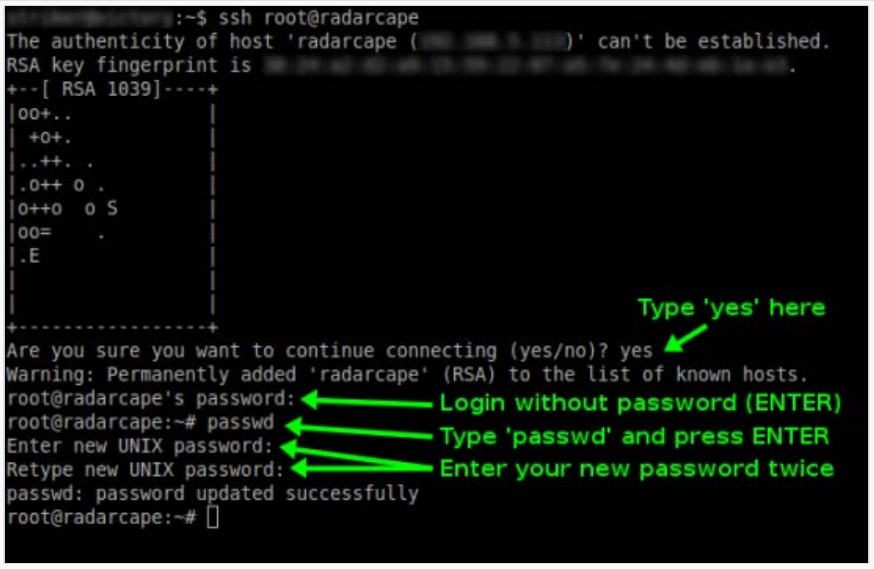
First you must login to your Radarcape via SSH.

An initial root password has not been set. Therefore, you can login with user *root* and no password.

After you have logged in on your Radarcape, you can set a new *root* password with the following command:

```
passwd
```

The following screenshot shows the complete password change procedure:



Radarcape Password Change

The message *passwd: password updated successfully* will indicate that the new password has been set.

Login to the Radarcape Web Interface

The default credentials for the Radarcape web interface are:

- User: Administrator
- Password: radarcape

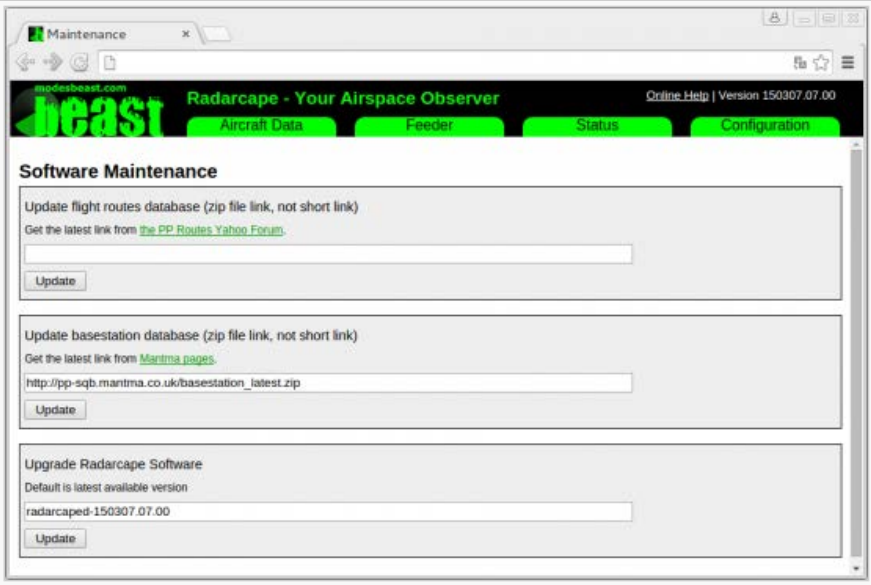
We strongly recommend you to **change your password** after the first login! Therefore, select *Configuration* > *Change Password* in the navigation menu.

Installation of Basestation and Flightroute Databases

The Radarcape can use external databases for displaying additional information to the 2D map and the aircraft list.

- *basestation.sqb*: aircraft type and *registration*.
- *flightroutes.sqb*: flight plan information (*origin* and *destination*)

These databases are not installed when we ship the Radarcape. Users can download and install them from third-party sources. This can be done via the *Software Maintenance* menu.



Radarcape Software Maintenance

The databases can be obtained from these sources:

- Plane Base NG (basestation.sqb, http://planebase.biz)
- ChrisGlobe.co.uk (basestation.sqb, http://chrisglobe.co.uk/planeplotter-aircraft-file-basestation-sqb/)
- PP Routes Yahoo Group (flightroutes.sqb, http://groups.yahoo.com/neo/groups/PP-logs-and-routes/files, registration required)

As both databases contain information that is subject to change, we recommend you to update these databases in regular intervals.

Radarcape Configuration

All changes of the Radarcape configuration can be done via the Web interface.

Radarcape Settings Web Page

If all settings have been made, scroll to the bottom of the page, enter the password, and press “Save Changes” to store the configuration. The default password for changing the configuration is *radarcape*.

Note: Due to feature enhancements and changes, this page is often subject to change.

Radarcape:Contents

Navigation
Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links
modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:Fixed IP Address

Required computer skills to execute this task: Advanced

Contents [hide]

Debian: Static IP Address / Direct Network Connection without DHCP

NOTE! This is unusal and not the regular case. It may only be necessary in these cases:

- You do not have a DHCP in your network
- You want to use a fixed IP address by whatever means
- You want to connect your Radarcape directly to your PC. For example, your PC connects to the internet router via WLAN and your Radarcape shall connect directly to the PC via Ethernet cable. (Note: In that case it might be possible that you connect the Radarcape to the Router, and then access it via WLAN and via Router, but it might fail)

On those Radarcape based on Debian, the static IP address can be configured in the Configuration - Network Settings menue.

Angstrom: Static IP Address / Direct Network Connection without DHCP

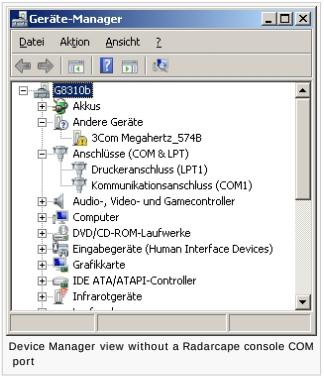
Static IP address on Angstrom is not recommended at all. Please upgrade to Debian. See Radarcape:DebianSupport#How_to_install

Accessing the Service Interface

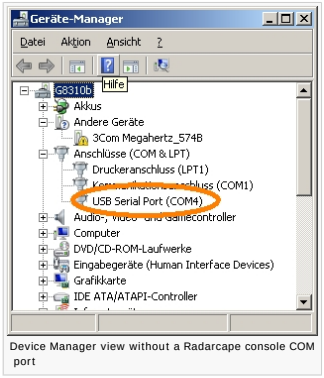
The Radarcape has 2 USB connectors:

- A standard USB type A on the front panel, e.g. for memory sticks or other extensions
- A micro USB connector on the back side, next to the antenna connectors. This allows you to connect to the service interface or *linux console*. Lets use the word **console** from now on.

Take a standard USB cable and connect the back side USB. Check in the device manager (german: Geräte-Manager) that a COM port becomes created.



Device Manager view without a Radarcape console COM port



Device Manager view without a Radarcape console COM port

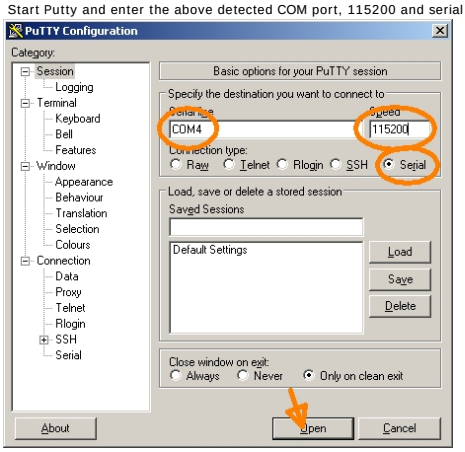
- If it does, skip the FTDI driver installation. Parameters of this COM port are 115200Bit/sec 8N1.
- If not, perform the FTDI driver installation and check if the COM port appears.

Install FTDI drivers

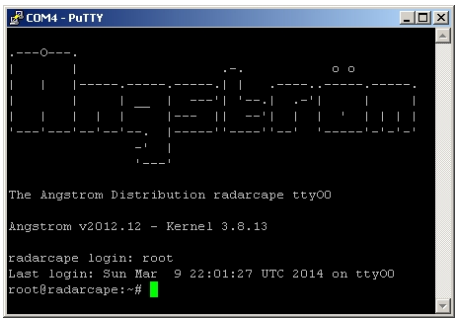
From the link FTDI driver web page, download the driver that fits to your system and install it. You should then see the COM port that becomes created from the Radarcape.
Note: On the rigtht side in the table of FTDIs page, the 'Comments' column, there is a downloadable install package, which might be easier to use.

Download and execute Putty terminal

We recommend Putty as console terminal, because it also supports SSH network connectivity.
Download Putty from here. There is no need to install, it is directly executable.
Start Putty and enter the above detected COM port, 115200 and serial into the startup menu:

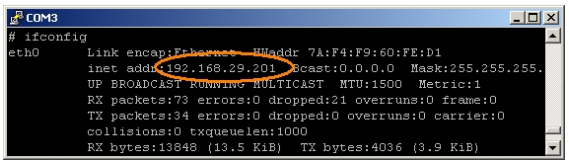


Once Putty is up, press Enter and you shall see something like below. Enter 'root' into the username prompt and simply press enter when beeing asked for your password (or mind your previously set password)



Display your Current Network Connection

Once you have established the console connection, you can see your assigned IP address with the command 'ifconfig'.



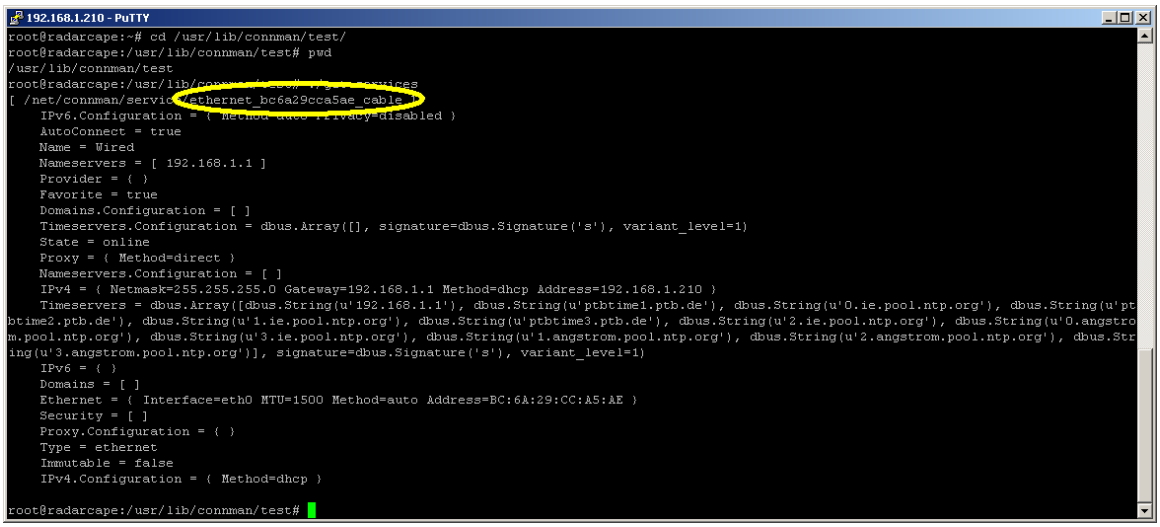
Modify Configuration for Static IP

To be continued....

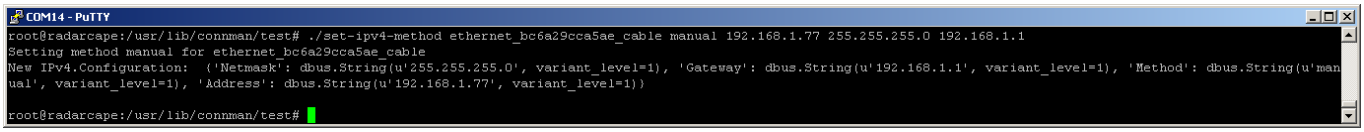
How to is described here: http://www.gigamegablog.com/2012/02/06/beaglebone-linux-101-assigning-a-static-ip-address-with-connman/

IMPORTANT NOTE: These commands must be issued through a serial console connection through the back side USB port, not via an SSH login. When the IP address is changed, it terminates the network connection, and with that it also terminates the setting process.

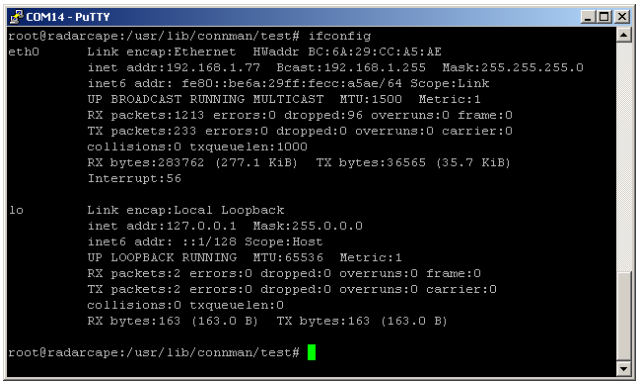
Step 1: Identify the name of your host adapter



Step 2: Swap the IP address



Step 3: Verify correct setting



Radarcape:Disclaimer

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Disclaimer

1. The Radarcape is a device made for hobbyists and airplane spotters. We ensure quality and stability and continue testing this in a high number of devices that we are operating by ourselves. Our product contains third party tools out of our control, like Linux and Linux tools, where updates are not in our responsibility but by the community or producer of these tools.
2. As a device for above mentioned circle of users, if you intend to use the device for any commercial task, you are fully responsible for any consequences. The Radarcape does not have obtained any of the aviation certification neither can we guarantee that the received data is always correct.
3. As you are free to modify the Linux part, such modifications are always in your own full responsibility. Also we do not guarantee that all Linux distributions are running stable in the Radarcape.
4. We do not ensure that the Radarcape is fully secure against unforeseen access if running in a network environment, neither public Internet nor even your local network. You should of course change the root password and remember it, and only then connect it to public Internet if you really know about the consequences.
5. As this is a small embedded device, when running heavy load with plenty of sharers or tools installed, the device may fail with lack of resources.

Forbidden Usage

Remember, that this device does not comply with any of the air traffic regulations or specifications and that it is not certified for aviation use. You MUST NOT use it for any usage case that in any part of the world would require a certification to comply with any rules. This device MUST NOT be used in services like mentioned below

- Operating an aircraft
- Operating an airport
- Operating airport services
- Air traffic or ground traffic navigation
- Air traffic management
- Airport management
- Aircraft management
- Testing of other Mode-S or Mode-AC related equipment

or any other services that are similar to those listed above.

Radarcape:Software Features

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Contents [hide]

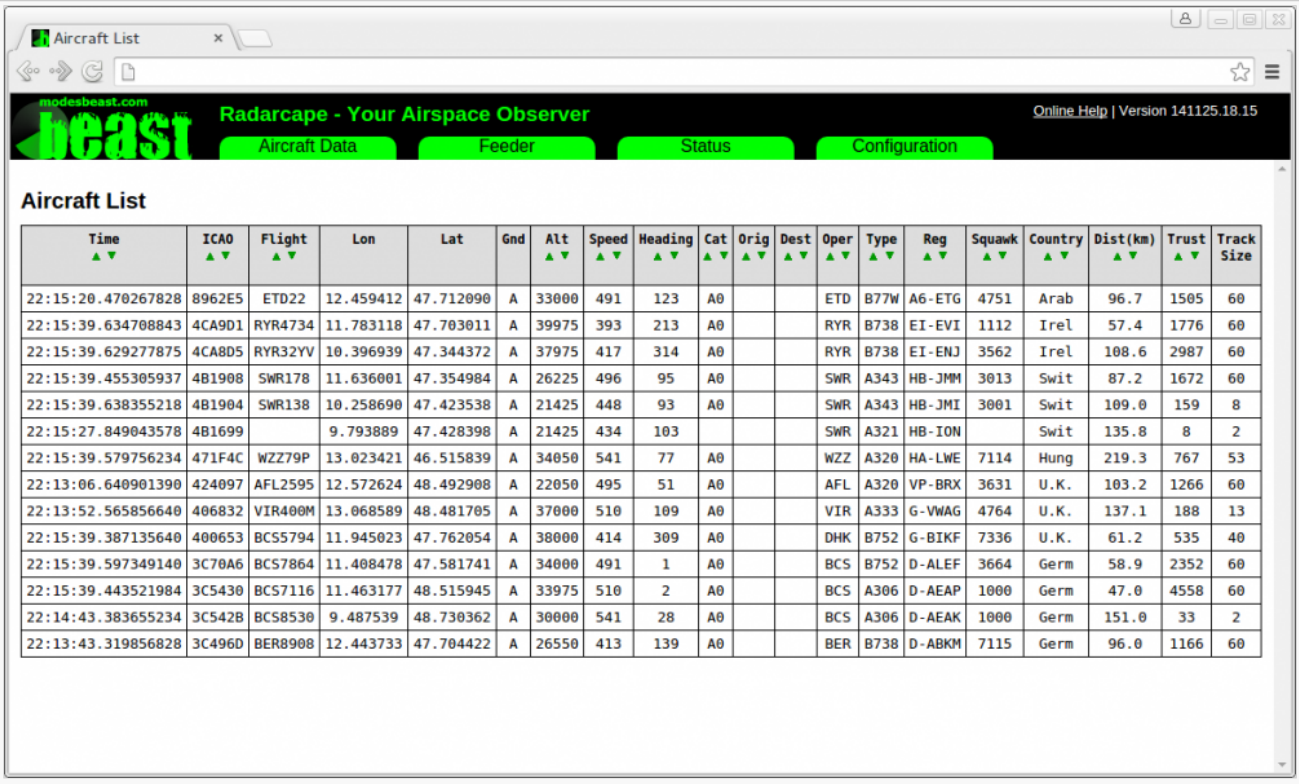
Radarcape Software Features

The Radarcape provides output of data in several levels:

- true raw data
- decoded data on a per frame basis (e.g., port 30003)
- decoded data on summary basis (e.g., aircraftlist.json, deltadb.txt)
- web browser support (aircraft list, 2D map)
- 3D output (live KML data)

Web Based Aircraft Table

A list of received aircraft can be fetched via a build-in Web server. This list can be sorted ascending and descending in each column by simply clicking on the arrows. The distances are automatically calculated from aircraft positions and your GPS coordinates.



Time	ICAO	Flight	Lon	Lat	Gnd	Alt	Speed	Heading	Cat	Orig	Dest	Oper	Type	Reg	Squawk	Country	Dist(km)	Trust	Track Size
22:15:20.470267828	8962E5	ETD22	12.459412	47.712090	A	33000	491	123	A0			ETD	B77W	A6-ETG	4751	Arab	96.7	1505	60
22:15:39.634708843	4CA9D1	RYR4734	11.783118	47.703011	A	39975	393	213	A0			RYR	B738	EI-EVI	1112	Irel	57.4	1776	60
22:15:39.629277875	4CA8D5	RYR32YV	10.396939	47.344372	A	37975	417	314	A0			RYR	B738	EI-ENJ	3562	Irel	108.6	2987	60
22:15:39.455305937	4B1908	SWR178	11.636001	47.354984	A	26225	496	95	A0			SWR	A343	HB-JMM	3013	Swit	87.2	1672	60
22:15:39.638355218	4B1904	SWR138	10.258690	47.423538	A	21425	448	93	A0			SWR	A343	HB-JMI	3001	Swit	109.0	159	8
22:15:27.849043578	4B1699		9.793889	47.428398	A	21425	434	103				SWR	A321	HB-ION		Swit	135.8	8	2
22:15:39.579756234	471F4C	WZZ79P	13.023421	46.515839	A	34050	541	77	A0			WZZ	A320	HA-LWE	7114	Hung	219.3	767	53
22:13:06.640901390	424097	AFL2595	12.572624	48.492908	A	22050	495	51	A0			AFL	A320	VP-BRX	3631	U.K.	103.2	1266	60
22:13:52.565856640	406832	VIR400M	13.068589	48.481705	A	37000	510	109	A0			VIR	A333	G-VWAG	4764	U.K.	137.1	188	13
22:15:39.387135640	400653	BCS5794	11.945023	47.762054	A	38000	414	309	A0			DHK	B752	G-BIKF	7336	U.K.	61.2	535	40
22:15:39.597349140	3C70A6	BCS7864	11.408478	47.581741	A	34000	491	1	A0			BCS	B752	D-ALEF	3664	Germ	58.9	2352	60
22:15:39.443521984	3C5430	BCS7116	11.463177	48.515945	A	33975	510	2	A0			BCS	A306	D-AEAP	1000	Germ	47.0	4558	60
22:14:43.383655234	3C542B	BCS8530	9.487539	48.730362	A	30000	541	28	A0			BCS	A306	D-AEAK	1000	Germ	151.0	33	2
22:13:43.319856828	3C496D	BER8908	12.443733	47.704422	A	26550	413	139	A0			BER	B738	D-ABKM	7115	Germ	96.0	1166	60

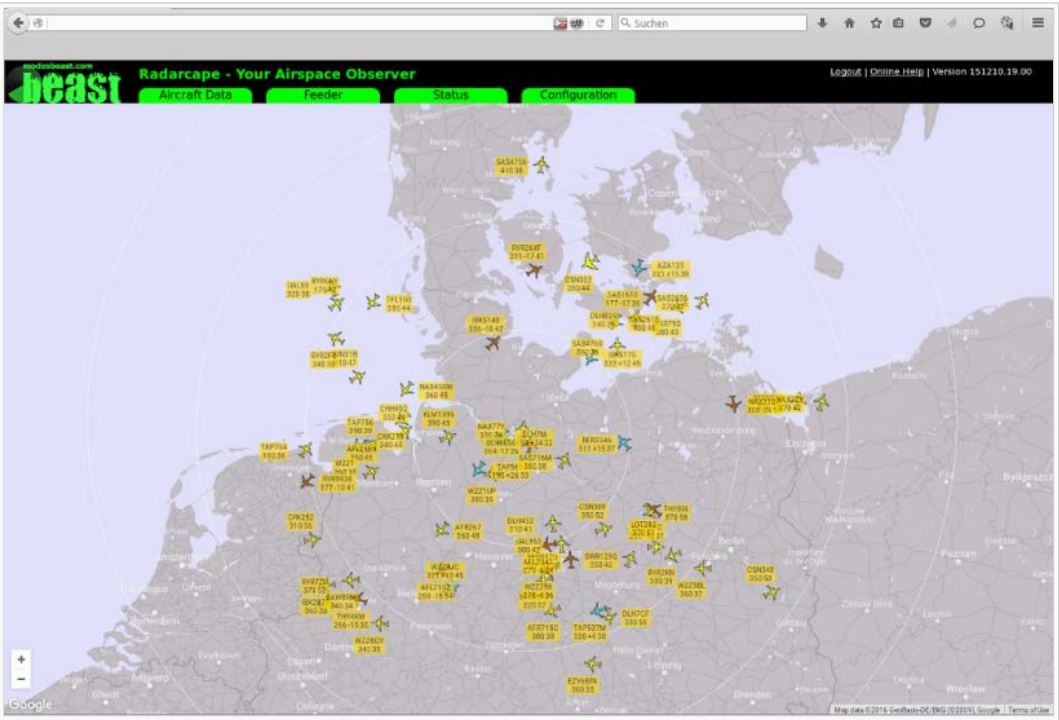
Radarcape Live Aircraft Table

Name	Description	Notes
Time (hh:mm:ss:ns)	Time of last message received from the aircraft	
ICAO	24 bit ICAO hex ID unique identification of aircraft	
Flight	the Call Sign as it is transmitted from the aircraft itself	
Lon	Longitude	
Lat	Latitude	
Src	Source of Lat/Lon: A=ADSB M=MLAT	
GndAir	Aircraft is on ground (identified with GND bit or DF-18 messages) or airborne	
Alt	Altitude (feet) at 1013 mb	
VRate	Vertical rate in feet/min	

Speed	Ground Speed in knots	
Track	Direction that the aircraft is travelling in degrees true	
Cat	Cat A0..C5 are transmitted by aircraft in Mode-S messages	
Orig	Origin of flight	taken from database, perform Maintenance -> Update flight routes database
Destin	Destination of flight	taken from database, perform Maintenance -> Update flight routes database
Oper	Flight operator	taken from database, perform Maintenance -> Update flight routes database
Type	Aircraft Type	taken from database, perform Maintenance -> Update flight routes database
Reg	Registration of aircraft	taken from database, perform Maintenance -> Update base station database
Squawk	Squawk code as it is transmitted by aircraft in Mode-S messages	
Country	Country that the aircraft is registered for, indicated through the upper bits in the ICAO hex id	
Distance	Distance to the observer if its Lat, Lon is either valid by manual entry in configuration or determined by GPS	
Trust	Number of highly trustable DF-11 or DF-17/18 messages per aircraft. Used to desinguish ghosts, as true aircraft quickly raise this number while ghosts stay at 1	
Track Size	Length of the track in 2D 3D display in 5sec sequence track points	

Live 2D Output

All received aircrafts with a known position are displayed on a 2D map in your Web browser.

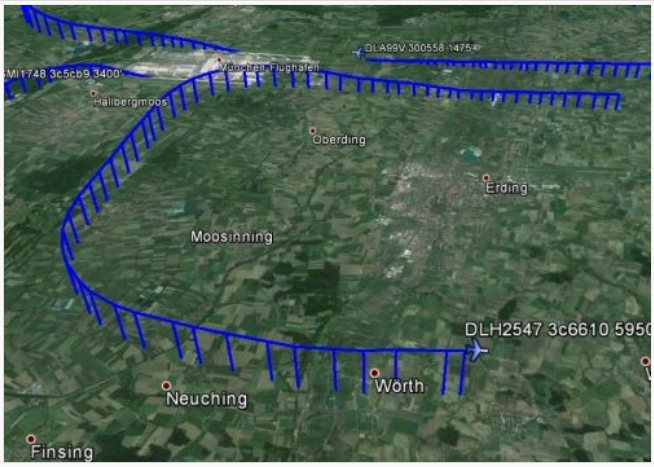


Radarcape Live 2D Display

Live 3D Output (KML/KMZ Output)

Google Earth can be attached to the Radarcape via KML/KMZ files.

Google Earth with live ADS-B data
An aircraft over Augsburg Airport



A go around in Munich EDDM

DeltaDB Service

The DeltaDB service can be accessed via **http://radarcape/deltadb.txt**. It outputs a comma separated list of all changes in the internal aircraft list since the last call or a specified time. This is an efficient replacement of port 30003 functionalities.

Aircraft List JSON Service

All data contained in the aircraft list can also be downloaded in JavaScript Object Notation (JSON). The file format can be used by other applications to access aircraft list data using the Hyper Text Transfer Protocol (HTTP) protocol.

The Aircraftlist JSON Service can be accessed via **http://radarcape/aircraftlist.json**.

The JSON output uses abbreviated identifiers for the data fields like below.

Note: Not all of the might be implemented at the time of this writing, some of them may only be available when having a special option key installed on your Radarcape.

Abbreviation	Description	Additional Information
uti	Linux timestamp of last message (contains date)	"uti":1434656441
tim	Time of last message (contains nanoseconds)	"tim":"19:27:35.143925171"
hex	ICAO Hex ID	
fli	Flight Identification	
lat	Latitude	
lon	Longitude	
gda	onGround (Ground Air)	
alt	Altitude	
spd	Ground Speed	
trk	true track	
cat	Category (A0-C7)	
org	Origin	Requires flightroutes database beeing loaded
des	Destination	Requires flightroutes database beeing loaded
opr	Operator	Requires basestation database beeing loaded
typ	Type	Requires basestation database beeing loaded
reg	Registration	Requires basestation database beeing loaded
squ	Squawk	
cou	Country	
dis	Distance	
tru	Trustlevel	
vrt	Vertical Rate	
mch	MACH	
ias	IAS	
tas	TAS	
rol	Roll angle	
tra	turn rate	
sfl	Sel FL	

qnh	QNH	
shd	Sel Heading	
hgt	Height difference	
mop	MOPS	
flg	Alert, SPI, Emerg.,IC	
tcs	TCAS alert	
nic	NIC + NACV	
apm	autopilot mode	
rec	record number	
lla	LatLon_Age	
lpa	LastPacket_Age	
tsa	Track Size ADS-B	
tsm	Track Size MLAT	

Data Streaming to Network (TCP and UDP)

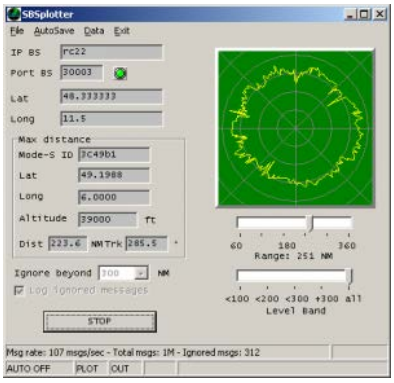
- *TCP or UDP port 10002:* This is a CRC-checked mirror of the data as it comes from the FPGA, DF-11, DF-17 and DF-18. Includes Mode-A/C data with respect to the configuration setting.
- *TCP or UDP port 10003:* Binary formatted raw data with all Modes-S data formats CRC-prechecked (eliminates transmission of the erroneous frames, reduces load on the network). All data from the FPGA is disassembled into messages and verified if correct.
- *TCP or UDP port 10004:* Binary formatted raw data, pre-checked DF-11, DF-17 and DF-18 only: minimum load for the transmission path but contains most information. No Mode-A/C data.
- *TCP or UDP port 10005:* Binary formatted raw data, only raw data frames of those aircraft where the location (latitude and longitude) is unknown. Used for special MLAT purposes. No Mode-A/C data.

The binary and AVR raw data formats are identical to those of the Mode-S Beast and documented in *Mode-S_Beast:Data_Output_Formats*. For the Radarcape, there is one additional message that contains timestamp and FPGA configuration information, which is triggered by each 1PPS from the GPS module.

Port 30003 Service (TCP, UDP, and USB-serial)

Port 30003 style output (e.g., for use with SBS Plotter) can be provided without the need of an additional application on your PC.

The Radarcape provides this data stream on TCP port 30003, UDP port 30003, and the serial USB interface.



The format of the data output can be found in this document

The *date* in Port 30003 messages is always the *Linux system date*.

The *timestamp* instead is a *GPS timestamp* when the configuration is set to GPS timestamps and system time when the Radarcape operates in legacy 12 MHz time stamp mode.

Due to the low efficiency and high processor load caused by this protocol, please do not use Port 30003 unless really necessary.

A better way of getting the same data is the deltaDB service.

On Linux, a very simple method how to access the TCP stream of Port 30003 is socat:

```
socat - TCP:radarcape:80
```

USB Serial Port Data Access

The Radarcape supports one selectable data stream out of following sources on a virtual serial port via the back side USB port:

- Raw FGPA data - including Mode-A/C data
- CRC pre-checked Mode-S with Mode-A/C data
- Mode-S Frame types DF-11, DF-17 and DF-18 only
- Mode-S Frames of all aircraft without a known location
- Port 30003 format

The output can be selected in the configuration menu. Due to processor load, it is recommended to keep this feature disabled when not required.

Output Settings

Data Stream on USB Serial Interface (Mode-S Beast compatibility)

- ○ disabled
- ◉ Raw FPGA data with Mode-A/C
- ○ Mode-S CRC pre-checked with Mode-A/C
- ○ DF11,17,18
- ○ Non ADS-B Aircraft
- ○ Port 30003

The setting can be changed on the fly and will apply without the need to reboot the Radarcape.

PC driver

The interface uses the Linux kernel's USB gadget serial driver. It will create a virtual serial COM port which you can identify in your device manager. An INF file is necessary in order to install it. This can be downloaded here: http://www.modesbeast.com/resources/g_serial.inf

The virtual serial port does not require any baud rate and handshake settings, it will work with any configuration.

Take care that when connecting you will get eventually two serial ports: One that provides access to the Linux console, and the one mentioned within this feature. If you have doubts which one to select, first try the higher number, or use a Putty terminal just to see which one outputs weird binary data (or port 30003 format if selected). The one that outputs a console screen on 115200 baud is the wrong one.

For Windows 7 the source of the driver is dropbox

Restrictions on the PC Serial Driver

- Currently the serial interface works with Putty and and test applications on Windows. It did not work with PlanePlotter under XP.
- Note that even when a Radarcape receives power through the back side USB, you MUST connect the external +5 V for the receiver and decoder to operate.
- When powering on, the external +5 V power supply MUST be connected prior to connecting the USB cable.

Further information:
USB Gadget Kernel Documentation
Some usage and driver hints - this is where the INF file comes from.

Radarcape Internal Web Pages

Navigation

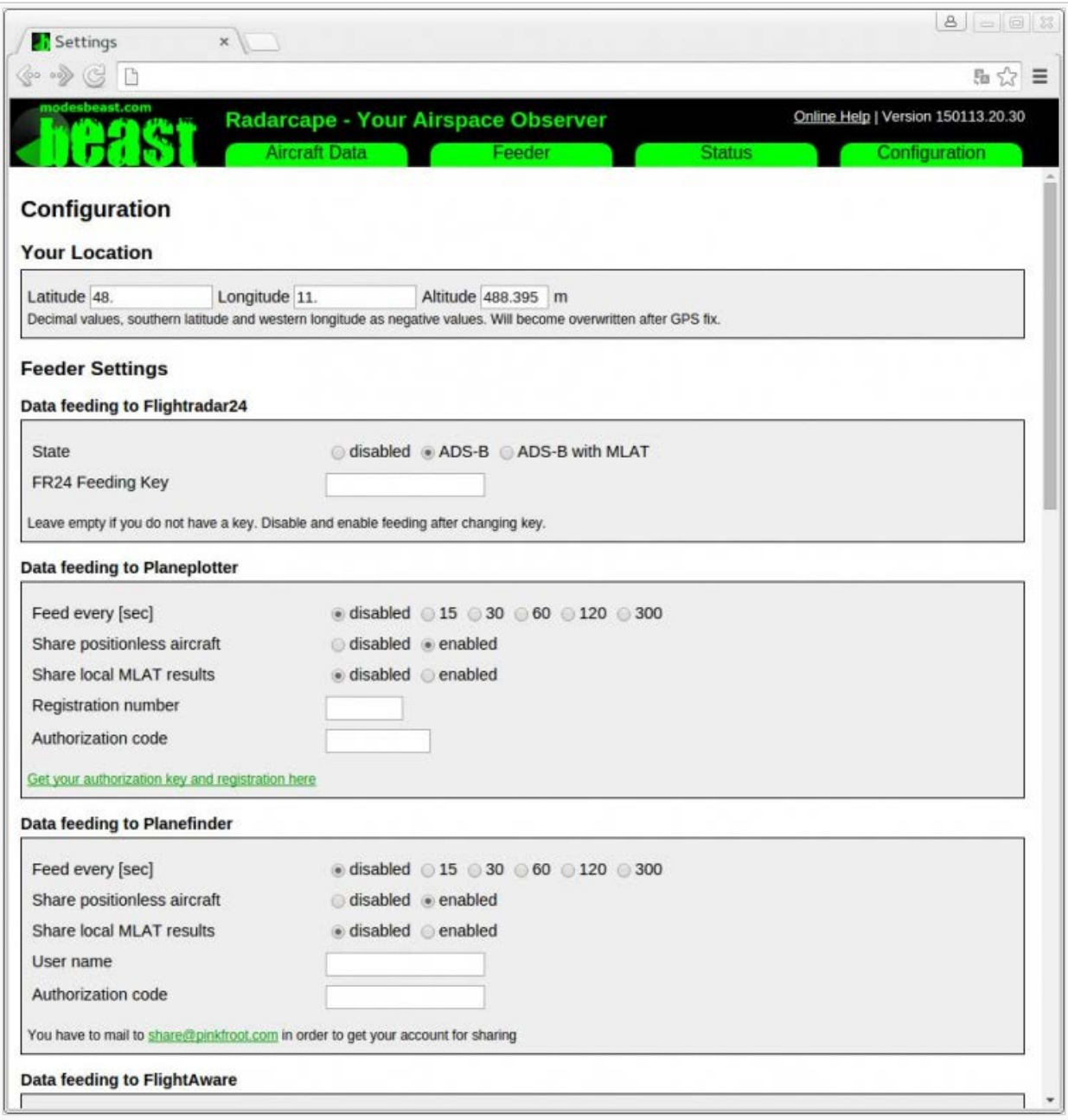
Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Radarcape Configuration

The configuration of the Radarcape can be done with a Web browser.



Radarcape Settings Web Page
Note: due to feature enhancements and changes, this page is often due to change

NOTE: The FPGA settings can still be overwritten by external software using escape commands as described for the Mode-S_Beast:Data_Input_Formats.

The default password for changing the configuration is *radarcape*.

Radarcape:Firmware Versions

Contents [hide]

Software Packages

Starting with November 2013, the release strategy will be Linux installer packages, which include the Linux software as well as the FPGA firmware.

Release_160928.00.00.30

New Features:

- N/A (pure bugfix release)

Bugfixes:

- Fixed: Incorrect timestamp output in 12 MHz clock mode

- Fixed: Incorrect decoding of DF-17 surface positions
- Fixed: Configuration restore function
- Fixed: Missing link to PlanebaseNG
- Fixed: Map API key can now be configured via the configuration page
- Fixed: FPGA does no longer output dublicate frames in certain operation modes
- Fixed: Incorrect ground possition decoding

Known Bugs

- Platform independent:
  - Logout screen is sometimes not properly displayed.
  - After power on, the GPS may require up to 5 minutes until synchronisation. During this time, your GPS LED will not become green, also there won't be a feeder working.
  - Disable FR24 feeder prior to the software upgrade and re-enable it when the new software release was installed successfully.
- Debian:
  - none
- Angstrom:
  - Angstrom devices may be inaccessible by SSH after a package installation and require a reboot.

Direct links to the packages

Debian radarcaped-160928.00.00.30.deb
Angstrom radarcaped-160928.00.00.30.opk

Release_160619.20.00.30

New Features:

- Configuration of static IP address via web interface
- Filters for altitude, speed, distance, ICAO hex codes on the 2D Map
- Save and restore settings to/from file
- Reboot button in the web interface
- Support of GLONASS (Only new Radarcape hardware releases with SMT360 GPS receiver)
- Factory reset function for the GPS module

Bugfixes:

- Fixed: Download location of basestation.sqb
- Fixed: Do not decode frames with ICAO equal to 0x000000 or 0xFFFFFF
- Fixed: Local clients cannot connect to UDP ports via loopback interface
- Fixed: Access control for JSON, KML, and TXT files via web interface
- Fixed: Occassional timestamp jumps in P30003 messages

Known Bugs

- Platform independent:
  - Logout screen is sometimes not properly displayed.
  - After power on, the GPS may require up to 5 minutes until synchronisation. During this time, your GPS LED will not become green, also there won't be a feeder working.
  - Surface positions can be wrong by -90°, 90°, or 180°.
  - Restore function does not work.
- Debian:
  - none
- Angstrom:

- Angstrom devices may be inaccessible by SSH after a package installation and require a reboot.

Direct links to the packages
Debian radarcaped-160619.20.00.30.deb
Angstrom radarcaped-160619.20.00.30.opk

Release_151210.19.00.30

New Features:

- The Flightradar24 feeders in the previous Debian releases provided wrong locations to the server.
  (Please note that the FR24 feeder is an external delivery and not under the same maintenance process like the rest of the firmware).
- Flightradar24 and FPGA version were added to the System Information web page

Bugfixes:

- Since 01. December a login as Administrator results in an empty web page return. This bug is fixed with this release. The erroneous release 1121.19.20.30 was removed from the server

Those who already updated to the intermediate release 151204.08.00.30 should find this new version under Configuration - Software Maintenance - Upgrade Radarcape Softare. Pressing the button [Update] will then install this new release.

If you however cannot reach your maintenance menue (by whatever means), you will need to do a manual installation through the command prompt.

Using Putty terminal, enter Hostname **radarcape**, port number **22** and Connection Type **SSH**.
Then press the [Open] button. Once on the command prompt, enter the command **uname -a**

If within the output you identify a Linux version 3.8, you're Radarcape is still on Angstrom. In this case, continue with the two commands

```
wget http://www.modesbeast.com/resources/radarcaped-
151210.19.00.30.opk
opkg install --force-overwrite radarcaped-151210.19.00.30.opk
```

If you see a Linux version 4.1 or something indicating Debian, you're Radarcape is Debian based. Then continue with these three commands:

```
wget http://www.modesbeast.com/resources/radarcaped-
151210.19.00.30.deb
dpkg -i radarcaped-151210.19.00.30.deb
apt-get install -f -y
```

Known Bugs

- Platform independent:
  - Logout screen is not properly displayed.
  - **NOTE:** After power on, the GPS may require up to 5 minutes until synchronisation. During this time, your GPS LED will not become green, also there won't be a feeder working.
- Debian:
  - none
- Angstrom:
  - Angstrom devices may be inaccessible by SSH after a package installation and

require a reboot.

Direct links to the packages

Debian radarcaped-151210.19.00.30.deb
Angstrom radarcaped-151210.19.00.30.opk

Release_151204.08.00.30

Release 151204.08.00.30 was an intermediate release only and is superseeded by Release 151210.19.00.30.

Release_151121.19.20.30

This is the first official common release for Debian and Angstrom. From this point on, new Radarcapes will be delivered with Debian operating system. Functionality of Debian and Angstrom based Radarcapes is absolutely identical. There is no urgent need for updating to Debian.

For those interested in reading about the Debian upgrade, see Radarcape:DebianSupport

Changes:

- UDP output of port 10003 - 10006 and port 30003: Information available on the TCP accessible ports can now be distributed to multiple clients via UDP protocol in parallel (with reduced resolution of timestamp).
  Simple UDP client example on Linux:

```
socat -u udp-recv:30003 -
```

- Two selectable skins for Planevision and Beast style.
- Map style changed from Terrain to Montony.
- Improved web session handling.
- Three new levels for access control of the webpages:
  - Very Strict: Each internal website requires being logged-in.
  - Strict: Configuration and the status pages require being logged-in.
  - Public: Just the configuration pages require being logged-in.
- 2D Map:
  - Label for each aircraft icon with Callsign, Flightlevel/Vertical Rate, and Ground Speed.
  - Climbing aircraft are displayed in sky blue, descending aircraft in earth brown, level flight in yellow.
  - The detailed aircraft information table disappears if clicking to an empty area and on time-out.
  - Trails disappear on time-out.
- New page: Status - System Information
- Introduction of feature keys for enabling of future SW options (options none available, yet).
- GPS Improvements:
  - Improved monitoring of GPS module. This may lead to delayed functionality where a proper GPS fix is necessary, e.g. GPS timestamps. Time and PPS must be based on UTC for operation. When the almanach is not yet available, it may require up to 5 minutes until UTC time is available.
  - Support of operation without GPS antenna. Requires manual configuration of the Radarcape location as precise as possible via the web interface.

    - The 'Antenna short' message in the GPS status was fixed.
- Debian: Support of software update using the maintenance menu.
- FR24 feeder updated to new version.
- FPGA: minor bug correction which spoiled GPS time-stamp at 16:16 UTC.

Known Bugs

- Platform independent:
    - Logout screen is not properly displayed.
- Debian:
    - After an update, it eventually may require a reboot or power cycle in order to bring the new version into life.
    - Those who already did a system update from Angstrom to Debian will have to install this version manually by downloading http://jetvision.de/resources/radarcaped-151121.19.20.30.deb Debian radarcaped-151121.19.20.30.deb and executing *dpkg -i radarcaped-151121.19.20.30.deb*, as in previous versions the integrated software update was not correctly implemented. With this version, it will work.
        - **IMPORTANT:** On the first try, your installation will most likely fail with missing boost libraries. If so, simply enter *apt-get install -y -f* which will download missing libraries and continue the aborted installation automatically.
    - In some cases the power LED will not show up in green but in red.
- Angstrom:
    - Version 150219.18.30 seems to have a bug in the integrated software update, you need to download the http://jetvision.de/resources/radarcaped-151121.19.20.30.opk Angstrom radarcaped-151121.19.20.30.opk manually and install it using opkg install --force-overwrite radarcaped-151121.19.20.30.opk.
    - Angstrom devices may be inaccessible by SSH after a package installation and require a reboot.

Direct links to the packages

Download is no longer available.

Credits

Thanks to Nuno, Sascha and Andy for their contributions and all others having contributed to this version.

Release_150620.14.30

Changes:

- aircraftlist.json: removed "hed" field, it now is called "trk". Previous field "trk" is now split into "tsa" and "tsm". Also inserted new field "uti" for unix time, allows recovering the date, but does not include nanosecond information.

see also Radarcape:Software_Features#Aircraft_List_JSON_Service

Direct links to the packages

Download is no longer available.

Release_150607.17.30

Bugfixes:

- Restart GPS survey fixed
-

Added time information to aircraftlist.json and harmonized JSON field identifiers to those described in Radarcape:Software_Features#Aircraft_List_JSON_Service

Direct links to the packages

Angstrom radarcaped-150607.17.30.opk

Release_150307.07.00

On Saturday, 7th March 2015, Solar Impulse will start in Abu Dhabi to achieve the First Round-The-World Solar Flight. The flight shall demonstrate that new technologies and alternative energy sources can achive what some consider impossible. The project was initiated by the two Swiss explorers Bertrand Piccard and André Borschberg.

For this event, we added a special Solar Impulse 2 aircraft symbol to the Radarcape 2D map. So, Radarcape users are enabled to see this exceptional aircraft on their maps. The new Solar Impulse symbol is available since SW release (20150307.07.00).

If you see Solar Impulse 2 on your Radarcape, we would be pleased when you could send us a screenshot with Solar Impulse on the 2D map. The hunt for the first screenshot starts now!

Solar Impulse 2 route: Abu Dhabi (UAE), Muscat (Oman), Ahmedabad (India), Varanasi (India), Mandalay (Myanmar), Chongqing (China), Nanjing (China), Hawaii (USA), Phoenix (USA), Central USA, New York (USA), Southern Europe or Northern Africa, Abu Dhabi (UAE). (Exact route depends on weather conditions)

We wish good luck to the SI2 team and keep our fingers crossed that your journey will be successful.

Further information on SI2: www.solarimpulse.com

New Features:

- Several improvements to 2D map web page
- New symbol for Solar Impulse 2 on 2D map

Bugfixes:

- Selected range unit (km/NM) is now used on 2D map Web page
- Planeplotter feeder disable issue fixed (it now can be disabled)

Installation:

Users with the previous release installed should see this version listed in their maintenance menu.

Manual installation command:

```
opkg install -V http://www.modesbeast.com/resources/radarcaped-150307.07.00.opk
```

Direct links to the packages

Angstrom radarcaped-150307.07.00.opk

Note: For updates, your Radarcape must have an internet connection. Manual update requires copying the update package to the Radarcape by any means and issuing the opkg install command for the location of the package.

Release_150219.18.30

New Features:

- There are many new aircraft symbols in the 2D map

Courtesy of Nic Storey, we now are able to display many different symbols for aircraft as soon as the type is known.
www.virtualsky.co.uk
www.nicstorey.co.uk/planeplotter
planeplotterforum.co.uk
planebase.biz
Thank you very much for your great support, Nic!

- Aircraft tracks in 2D map: Simply click on an aircraft in order to see the track.
- New FPGA showing GPS tracking status:

- continuous red: GPS antenna short or open
- continuous yellow: Not tracking any satellite
- flashing green but yellow background: Operating, but tracking less than 3 satellites (is ok, but not recommended)
- flashing green with black background: Normal operating mode
- flashing yellow with black background: If rarely happens: normal operating mode with minor timestamp deviations

- New FR24 feeder

Changes:

- The Mode-S LED for correct operation is now flashing green instead of red. This will prepare for later red and yellow indication for other kind of notifications.

Known Issues:

- 2D map tracks do not disappear sometimes if the aircraft is no longer visible. Solution: simply select another aircraft

Direct link to the package

Angstrom radarcaped-150219.18.30.opk

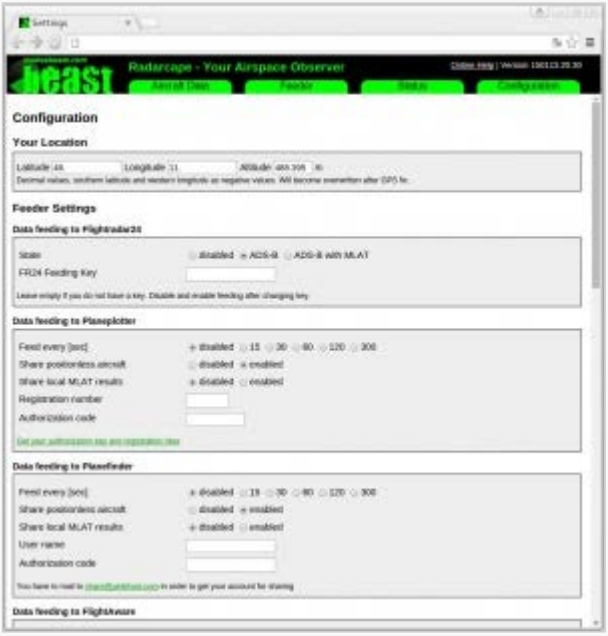
Release_141125.18.15

New Features:

- Web user interface got a modern look and feel
- The aircraft icons on the 2D map are no longer simple arrows but look now like real aircrafts
- New aircraftlist.json which contains the same data as the Web aircraft list

The new Web interface was tested with the following browsers:
- Firefox 33 or later (Linux, Windows)
- Iceweasel 31 or later (Linux)
- Google Chrome 39 or later (Linux, Windows)
- Internet Explorer 11 or later (Windows)

Other browsers may work but have not been tested.



Radarcape Web Configuration Page

Fixes:

- no data sent via delta DB file

Release_140922.19.03

New Features:

- In addition to Flightradar24 and Planeplotter, the Radarcape can now easily feed the current aircraft list to Planefinder and FlightAware servers. See the configuration menu how to register at their servers as a feeder. You can select the feeding interval separately for each in steps by 15, 30, 60, 120 and 300 seconds. There is a random value added in order to spread the feeding events a little over time.

Changes:

- The main menu and the configuration menu became redesigned for a better readability.
- Feeder status shows some more information, e.g. the true FPGA settings and the server's respose.

Fixes:

- The feeding of data is now coupled to the trust level of an entry. It must be at least 2 in order to upload data to a server. This should prevent ghost aircraft from beeing transmitted. Regular aircraft easily exceed this number within a very short time, so there is no real delay in announcing them to others. This feature should allow FEC beeing switched on in some more situations.
- Omit time should now be used in all maps and tables correctly.
- In the 2D map some aircraft appeared at 0°/0° coordinates after some time. That is fixed now.

Release_140412.08.49

New Features:

- none

I am currently working for the Peer-to-Peer MLAT, which is basically working with very

precise results, however the communication between the Radarcapes has proven to be unstable and so needs a redesign.

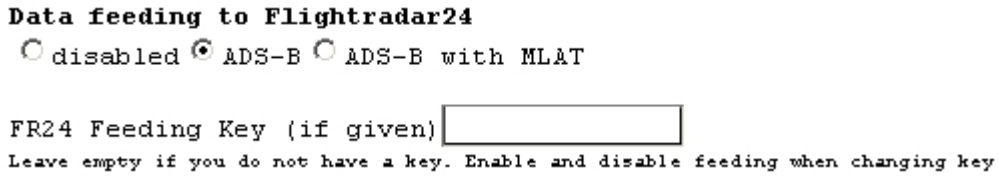
Changes:

- New FR24 feeder
- Some performance increase with Port 30003 data output
- The Flightroutes link parser was changed to handle the new structure of the links

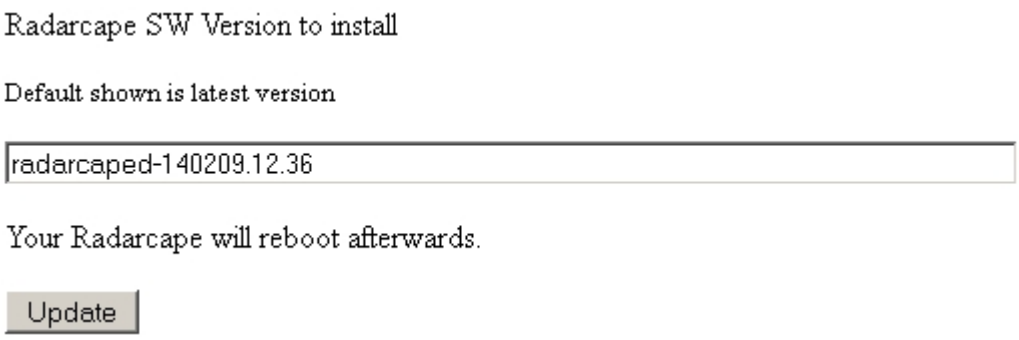
Release_140209.18.32

New Features:

- Decodes flight identification from DF20/DF21 frames (thanks, Bev)
- Decodes speed/heading from DF-17 subtype 3+4
- Automatic control of FR24 feeder setting change. No longer needs to manually sequence a setting change.



- Prepeared for next release to be installed with web GUI command. With this, it does not need to go into Linux when you want to install a software update



- The 3D map opens correctly with Windows 7

Changes:

- New FR24 MLAT capable feeder release 20140127-1444
- Some cosmetic changes
- Planeplotter sharing feeder in some cases contained false flight names

Install command:

```
opkg update
opkg install -V http://www.modesbeast.com/resources/radarcaped-
140209.18.32.opk
```

Release_140124.08.32c

New Features:

- Supports type information and routes information from common databases. This is displayed in aircraft list and 3D output currrently, it will become displayed in 2D and JSON output with next release. Origin and Destination are split, so you can sort for both in the aircraft list.
- There is a new Maintenance web page in the Radarcape where the links to the zipped archives can be pasted into, after pressing **Update** button the database is

fetched from the server, unzipped and becomes installed. The whole update procedure is displayed in the web browser.

Note: Flightroute emails mostly contain a short link to the archive. Following this link is not included here, you always must copy the final ZIP archive link. This link also invalidates after a while, so in case of problems, referesh the page displaying it and use the new link.
It does not matter if you prefer the 3 letter or 4 letter database, Radarcape will handle both.

Changes:

- New FR24 MLAT feeder: This feeder connects to raw data port, no longer to Port 30003. Its options for aircraft sharing and MLAT data sharing can be controlled through Radarcape's configuration dialogue. Note that the CPU load is somewhat more than the old feeder.
- The former field 'Ident' became renamed to 'Flight'

Install command:

```
opkg update
opkg install -V http://www.modesbeast.com/resources/radarcaped-
140124.08.32c.opk
```

Release_140106.19.14

New Features:

- Planeplotter sharing upload for positionless aircraft can be disabled now

Changes:

- uploading to Planeplotter uses upper case 6 digit ICAO id

Install command:

```
opkg install -V http://www.modesbeast.com/resources/radarcaped-
140106.19.14.opk
```

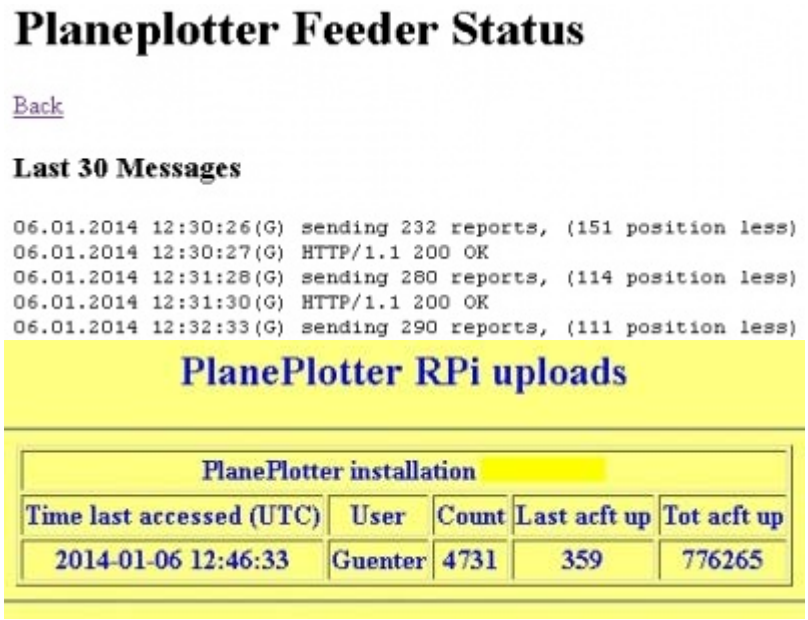
Release_140106.13.29 - Planeplotter Feeder, Ground Decoding

New Features:

- Planeplotter aircraft sharing added

This is the first release which publishes the current known aircraft to the Planeplotter network. Uploading can be enabled and disabled. You need to get a sharing authorisation, whichis explained in the configuration dialogue. There is also a status page and a link provided to the Planeplotter server which shows your past uploads. Uploading is currently done once every minute plus a random 0...14 sec. (The uploading methode was prepared for R-Pi, Bev needs to rename the service now.)

MLAT support will be added as soon as possible.



Planeplotter Feeder Status

Back

Last 30 Messages

```
06.01.2014 12:30:26(G) sending 232 reports, (151 position less)
06.01.2014 12:30:27(G) HTTP/1.1 200 OK
06.01.2014 12:31:28(G) sending 280 reports, (114 position less)
06.01.2014 12:31:30(G) HTTP/1.1 200 OK
06.01.2014 12:32:33(G) sending 290 reports, (111 position less)
```

PlanePlotter RPi uploads

PlanePlotter installation				
Time last accessed (UTC)	User	Count	Last acft up	Tot acft up
2014-01-06 12:46:33	Guenter	4731	359	776265

- Ground decoding

Ground data is decoded. It is output in all services.

Changes:

- The Flightradar24 Feeder can now be started based on the internal deltaDB methode or Port 30003 data, as some users felt that if based on Port 30003 the acceptance rate at FR24 is higher than based on deltaDB. Please note that Port 30003 is a very inefficient protocol and consumes more CPU resources than

deltaDB.

- The feeders now start only if the GPS time is available. Timestamps on web status pages contain an (X) if unknown time source, (C) if the CPU time is used and (G) if GPS time is present.
- Port 30003 output data contains GPS timestamp if available.

Install command:

```
# original version overwritten by 19.14
opkg install -V http://www.modesbeast.com/resources/radarcaped-
140106.19.14.opk
```

Release 131225.18.56

New Features:

- Remote control of FPGA configuration ("DIP switches") can be enabled and disabled in configuration

Solves:

- Timestamp problem: All timestamps are based on the GPS time. If not available CPU time is used

Install command:

```
opkg install -V http://www.modesbeast.com/resources/radarcaped-
131225.18.56.opk
```

Release 131206.22.40

New Features:

- 2D map automatically centers on your GPS position and zooms according to your current largest distance.

Solves:

- Internet Explorer may not be display the web pages at all, Mozilla Firefox seems to be tolerant. Radarcape currently does not send the correct HTML header.

Install command:

```
 opkg install -V http://www.modesbeast.com/resources/radarcaped-
131206.22.40.opk
```

Release 131202.20.55

New Features:

Solves:

- KML starter did not access KML output -> Google Earth 3D view did not work

Install command:

```
 opkg install -V http://www.modesbeast.com/resources/radarcaped-
131202.20.55.opk
```

Release 131128.21.18: Introduction of 2D Maps

New Features:

- 2D map display on web browser

Solves:

Install command:

```
 opkg install -V http://www.modesbeast.com/resources/radarcaped-
131128.21.18.opk
```

Release 131127.05.01: Solves P30003 message type problem

New Features:

- none

Solves:

- The port 30003 messages MSG,2 and MSG,3 seem to be swapped. ==> actually all Port 30003 messsages were one count too low.

Install command:

```
 opkg install -V http://www.modesbeast.com/resources/radarcaped-
131127.05.01.opk
```

Release 131126.19.46: Configuration Page Issues Solved

New Features:

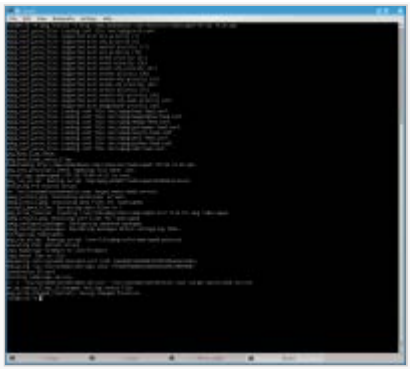
- none

Solves:

- The configuration page when beeing saved does not confirm change but show an empty page or a failure message. However the changes appear to be taken. It only happens on a few installations. Investigation is in progress.
- There was a delay when connecting to a TCP port, which is now avoided.
- HTML 'Back' link moved to the top of all pages
- Timestamp behaviour in P30003: Is now truely message generation and message capture time.
- uses GCC 4.3.3 and Boost 1.53.0

Install command:

```
 opkg install -V http://www.modesbeast.com/resources/radarcaped-
131126.19.46.opk
```



Installation output

FPGA Firmware

Firmware update on the Radarcape is absolutely simple: Just mount the Radarcape as a drive into your Windows, then replace the existing meaADSB.rbf file with the one downloaded from this web page. You may save the existing one, but the history also exists here for download (including the version name in the file name).

The GPS timestamp

The GPS timestamp is completely handled in the FPGA (hardware) and does not require any interactions on the Linux side. This is essential to meet the required accuracy. The local clock in the FPGA (64MHz or 96MHz) is stretched or compressed to meet 1e9 counts in between two pulses by a linear algorithm, in order to avoid bigger jumps in the timestamp. Rollover from 999999999 to 0 occurs synchronously to the 1PPS leading edge. In parallel, the Second-of-Day information is read from the TSIP serial data stream and also aligned to the 1PPS pulse. Both parts are then mapped into the 48bits that are available for the timestamp and transmitted with each Mode-S or Mode-A/C message.

- SecondsOfDay are using the upper 18 bits of the timestamp
- Nanoseconds are using the lower 30 bits. The value there directly converts into a 1ns based value and does not need to be converted by sample rate

```
nanosec = (msg[2] & 0x3f) << 24
        | msg[3]          << 16
        | msg[4]          << 8
        | msg[5];

daysec =  msg[0] << 10
       | msg[1] << 2
       | msg[2] >> 6;

if (daysec_tm1!=daysec)
{
    daysec_tm1 =  daysec;
    hh         = (daysec/3600) % 24;
    mm         = (daysec/60)   % 60;
    ss         =  daysec       % 60;
}
```

Example:

Timestamp of 1019UTC: 0x244bbb9ac9f0

lower 30 bits are 0x3b9ac9f0 => Nanosecond = 999999984 * 1e-9
upper 18 bits are 0x912e => daysec = 37166
hour = 37166/3600 = 10
minute = 37166 / 60 modulo 60 = 19

second = 37166 modulo 60 = 26

So time is 10:19:26.999999984 after UTC midnight

The legacy 48bit and 12MHz based timestamp however is not synchronized to 12MHz at all, so it still works as it has been since ever.

Linux and FPGA firmware package ppsjump-021 (23. Aug. 2013)

Corrections

- For enhanced stability, this version is based on Linux 3.8.
- The GPS tool is now included into the radarcape deamon. It also provides a GPS status through the web server, accessing gps.html on the integrated webserver
- The TCP ports for data streaming connect on each try, not each second.

Installation

For this version, it is essential that you update your SD card completly from scratch. Download the naked Linux 3.8 image (73MB) and make a SD card as described in Radarcape Linux Install/Configure. Please mind the two screenshots there in order to see how about the update procedure works like. (As some users had problems unpacking the XZ, there is a Linux 3.8 image ZIP Version (130MB) on the server, too)

When inserted and rebooted, either login through the serial port (Mini-USB on the back side) or via SSH/network (then your destination temporarily is “beaglebone”). Enter these commands, and don't forget to enter your known Radarcape ID when beeing asked from the skript:

```
cd
rm install38.sh
wget http://www.modesbeast.com/resources/install38.sh
sh ./install38.sh
reboot -f
```

FPGA Firmware meaADSB_ep3_143_ppsjump-020

Corrections

The GPS timestamp locked on multiplies of 32768 in situations when the 1PPS signal was disturbed by external matters.

superseeded by ppsjump-021

FPGA Firmware meaADSB_ep3_141_gpsmlat-3

Corrections

SecondOfDay (the upper 18 bits of the timestamp in GPS mode) and Nanoseconds (the lower 30 bits) are now synchronized.

Note that in order to overcome above problem with negative timestamps, the GPS read: absolute timestamp of Mode-S and Mode-A/C frames is taken at the end of the frame, at least until further notice. This does not make any difference for multilateration, as long as this feature is unique provided by the Radarcape.

Firmware meaADSB_ep3_141_gpsmlat-3

```
md5sum meaadsb_ep3_141_gpsmlat-3.rbf
86d6cdb069868e4f57d47dfc3441593c  meaadsb_ep3_141_gpsmlat-3.rbf
md5sum meaadsb_ep3_141_gpsmlat-3.zip
5bcae05ea429b4ef943b303314e22b82  meaadsb_ep3_141_gpsmlat-3.zip
```

FPGA Firmware meaADSB_ep3_141_gpsmlat-2

Firmware meaADSB_ep3_141_gpsmlat-2 has a working GPS timestamp function. Therefore, DIP#5 switch selects either the standard 12MHz timestamp (when off) or the GPS timestamp (when on).

```
md5sum meaADSB_ep3_141_gpsmlat-2.rbf
dc7a6278a668b1bdb81fd67e7a1891a6  meaADSB_ep3_141_gpsmlat-2.rbf
md5sum meaADSB_ep3_141_gpsmlat-2.zip
ba54740894406cb38e8dd95d0fc3e3e8  meaADSB_ep3_141_gpsmlat-2.zip
```

Radarcape DIP Switch Settings

Radarcape DIP Switch	**DIP#1**	**DIP#2**	**DIP#3**	**DIP#4**	**DIP#5**	**DIP#6**	**DIP#7**	**DIP#8**
Equivalent Beast DIP with resp. to PP setting	DIP#3	DIP#4	DIP#5	DIP#6	DIP#7	DIP#8	DIP#9	DIP#10
When ON	Binary format	only DF-11 and DF-17	enable MLAT in AVR format	CRC check disabled	**GPS based timestamp**	RTS hardware handshake	1 Bit FEC disabled	Mode-A/C decoding enabled
When OFF	AVR format	all usable DF	no MLAT in AVR format	CRC check enabled	**standard 12MHz timestamp**	hardware hand-shake disabled	1 Bit FEC enabled	Mode-A/C decoding disabled
Command Character	c/C	d/D	e/E	f/F	g/G	h/H	i/I	j/J
			not used in binary format					

An upper case character is equal to a DIP that is in ON position, a lower case character equal to DIP in open position

The green LED next to the SMA connector is used as GPS indicator:

1. Short on, long off: Just 1PPS is present, but no time of day information
2. On and off time equal: 1PPS present, Time of day present, but there is a synchronisation offset
3. Long on, short off: 1PPS present, Time of day present, Internal time is synchronized to GPS

It is not a problem if the clock sometimes falls back from (3) to (2), because the sensitivity of synchronisation check is +/-1 tick.

The center red LED flashes as an indication of operating about twice per second. It should flash very fast in case that hardware handshake is active.

The GPS based timestamp still uses the standard 48 bits as known from the 12MHz timestamp, but in different way:

- the lower 30 bits are the time since the last 1PPS pulse, in 1ns steps, currently 15ns resolution
- the upper 18 bits are the Seconds-Of-Day, starting with zero at midnight UTC

Known Issues (meaADSB_ep3_141_gpsmlat-2)

1. Within the GPS timestamp, the Second-Of-Day part advances in the mid of the 0-999999999ns phase next version
2. Sometimes the Second-Of-Day part does not increment at the rollover of the nanosecond part next version
3. The absolute value of the GPS timestamp of 14 bytes long Mode-S frames is offset, however since all units do have that error, it is not a big problem for multilateration wait for release
4. Negative delta time offsets between consequent frames not an issue (see below)

Negative delta time between consequent frames

If you look at the block diagram of the Mode-S Beast, recognize that there are several frame decoders working in parallel, plus the Mode-A/C decoder, which is not yet mentioned in the picture. They all work independently, their output - a ready frame - is written into a FIFO in order to buffer it for RS232 transmission.

It now may happen that during the reception of a Mode-S frame, an overlapping Mode-AC frame becomes decoded in parallel and is written into the FIFO, prior to the end of the Mode-S frame. Since the timestamps are taken at the start of frame, in that situation, the Mode-AC frame with a later timestamp is written to the FIFO before the Mode-S frame finishes. Consequentally on the output, the later timestamp of the Mode-S frame appears ahead of the Mode-S frame's.

It is easy to understand with the Mode-AC as a cause, but the same happens if one of the noise decoders or the overlapping Mode-S frame decoder outputs a frame while the other is still working.

Sorting that in the FPGA would cost too many ressources, so users of the timestamps anyway need a matching algorithm among different units, so that algorithm should be aware about this situation.

If you think about swapping them around, note that it may not happen with two frames but several, e.g. in the situation that two Mode-A/C frames do overlap a 14 byte Mode-S frame.

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:Watchdog

Contents [hide]

The tasks described in this manual require advanced knowledge of Unix/Linux.

Hardware Watchdog Overview

Starting with the 2nd production Batch, the Radarcape is equipped with a MAX6371 hardware watchdog. The watchdogs shall reboot the Radarcape in the case that the software hangs. This is especially important if the Radarcape is operated at remote locations or without a human operator available for a manual reboot. The AM3335 internal watchdog is not sufficient for such cases as it can only become activated after Linux has booted, so early faults would not be covered then.

The watchdog listens on GPIO_60 of the BeagleBone board for a toggling signal. If GIPO_60 has not been toggled within 1 minute, the BeagleBone board is being reset via the SYS_RESET pin. GPIO_60 must be toggled by software at least once every minute otherwise the Radarcape reboots. Toggling of GPIO_60 is usually being done by the Radarcape softawre (rcd). The very first timeout after power-on is 2 minutes, which leaves plenty of time for even reflashing the whole eMMC card.

Manual Hardware Watchdog Toggling

In certain situations it becomes necessary to temporarily stop or even de-install the Radarcape daemon. It then is necessary to toggle GPIO_60 by other ways to hinder the hardware watchdog from rebooting the Radarcape. This can be achived using the following script:

```
#!/bin/bash

echo *** Configure Watchdog retrigger pin & toggle first time

# Setup GPIO multiplexer (old firmware releases only)
if [ -e "/sys/kernel/debug/omap_mux/gpmc_ben1" ]
then
  echo 07 > /sys/kernel/debug/omap_mux/gpmc_ben1
fi

# Export GPIO_60 to /sys filesystem
echo 60 > /sys/class/gpio/export

# Set GPIO_60 direction to output
echo out > /sys/class/gpio/gpio60/direction

# Toggle the pin until the user presses CTRL+C
echo 'Hit CTRL+C to exit'
```

```
while 1;
do
  echo 1 > /sys/class/gpio/gpio60/value
  sleep 15
  echo 0 > /sys/class/gpio/gpio60/value
  sleep 15
done
```

The toggling loop can also be typed as one-liner:

```
while :; do echo 1 > /sys/class/gpio/gpio60/value; sleep 15; echo 0 >
/sys/class/gpio/gpio60/value; echo 'Hit CTRL+C to exit'; sleep 15;
done
```

FAQ

F: The file /sys/kernel/debug/omap_mux/gpmc_ben1 does not exist on my Radarcape?
A: This file exists only on old Radarcape firmware. If it does not exist, ignore it and continue with the next command.

F: The message *write error: Device or resource busy* appears when I execute *echo 60 > /sys/class/gpio/export*. What shall I do?
A: In this case GPIO_60 has already be exported to the /sys file system. You can unexport it by typing *echo 60 > /sys/class/gpio/unexport*?

Radarcape:SD Card Howto

Required Unix/Linux skills to execute this task: Advanced

Contents [hide]

This page describes how you can install a Linux system on the SD card. This is a bring up from zero and is necessary if you want to start from the beginning or if major changes have been applied to the Linux itself. A new SD card can be created either directly on the Radarcape or using an external Linux PC.

Restore the SD card using the Radarcape and a SD card reader

DUE TO EXTERNAL CHANGES CURRENTLY NOT WORKING EXCACTLY LIKE SHOWN

Prerequisites

- A capability to write 4GB size SDHC Micro-SD cards on the Radarcape and an USB SD card reader (I am using a Kingston FCR-HS219)
- Putty SSH client for Windows installed on your PC
- Putty can be used for both, the back side serial port via USB-Serial as well as for a SSH console via network. The network name of the Radarcape is given on the sticker on the device, e.g. rc22. The account name is root and normally there is no password, so only press enter when beeing asked for the password.
  (Note: However, when mounting network drives using samba, there is a password for root, that is different to the one for login)
- FTDI VCP Driver installed on your PC
- Eventually a micro USB cable for the rear USB port (the one besides the SMA connector)
- An Internet connection for the Radarcape

Packages

- for mkfs.msdos install the package dosfstools

- for mkfs.ext4 install the package e2fsprogs-mke2fs

Create the SD Card

For this step, you need a SDHC capable SD card reader connected to the front side USB expansion port of your Beaglebone. I am using a Kingston FCR-HS219 SD card for this purpose. Keep the SD card removed in order to meet below description

After connecting, you need to reset your Radarcape. Either power cycle it or enter “reboot” if you already have a console opened on the Linux. After rebooting, establish a Putty terminal connection to your Radarcape. Preferably to the back side USB console, but SSH will do as well.

Now enter the command

```
dmesg | grep " sd "
```

The output will show if the SD card reader was detected properly, it should tell you somehow

```
[    3.021715] sd 0:0:0:0: [sda] Attached SCSI removable disk
[    3.034153] sd 0:0:0:1: [sdb] Attached SCSI removable disk
[    3.045395] sd 0:0:0:2: [sdc] Attached SCSI removable disk
[    3.058879] sd 0:0:0:3: [sdd] Attached SCSI removable disk
```

Now insert the SD card into the matching slot. If your connection is the back side USB, you will see below message on the screen anyway, otherwise enter above command again. Look for lines like this:

```
[   81.515519] sd 0:0:0:3: [sdd] 7744512 512-byte logical blocks:
(3.96 GB/3.69 GiB)
[   81.530594] sd 0:0:0:3: [sdd] No Caching mode page present
[   81.536033] sd 0:0:0:3: [sdd] Assuming drive cache: write through
[   81.550295] sd 0:0:0:3: [sdd] No Caching mode page present
[   81.555742] sd 0:0:0:3: [sdd] Assuming drive cache: write through
```

This means that a 3.96GB card was inserted into drive sdd. Remember this drive name.

Next step is to download and run the install skript from the server. Enter the commands below

```
rm -rf bb_make-Angstrom-new.sh
wget http://www.modesbeast.com/resources/bb_make-Angstrom-new.sh
sh ./bb_make-Angstrom-new.sh sdd
```

Remember that in the 3rd line you eventually need to replace sdd in case that your SD card is not named sdd. Answer the questions and else let the script go.

After a short while the SD card is ready. Then swap the SD cards and store the one used until now on a safe place until your new card is operating correctly. '

Restore the SD Card Image on Native Linux

```
DUE TO EXTERNAL CHANGES, CURRENTLY NOT WORKING
```

Prerequisites

- A PC running with any Linux OS (e.g., Ubuntu Linux).
- The capability to write 4GB size SDHC Micro-SD cards on the Linux PC and an USB SD card reader (e.g., Kingston FCR-HS219 card reader).

- T7 (0.078 in / 1.99 mm) Torx wrench or screwdriver
- SSH Client (e.g., Putty on Windows, OpenSSH on Linux)
- An Internet connection for the Linux PC and the Radarcape

Create SD Card

- Remove power from your Radarcape
- Remove the front panel of your Radarcape
- Remove the SD card and put it into the SD card reader of your Linux PC
- Download the script Angstrom installation script

```
wget http://www.modesbeast.com/resources/make-Angstrom-new.sh
```

- Execute it with the drive name of your SD card

```
sh ./make-Angstrom-new.sh [mmcblk0|sdd]
```

- Place the updated SD card in the SD card reader of the Radarcape
- Apply the front panel of your Radarcape
- Apply power to your Radarcape

Install Radarcape Software

Install the Radarcape daemon and the FPGA Firmware as decribed here.

For experts: With this procedure you should be able to make a Radarcape from any native Beaglebone Angstrom release. However we do not guarantee that the Radarcape operates with each release, as we have seen broken ones.

Restoring an SD card with Windows

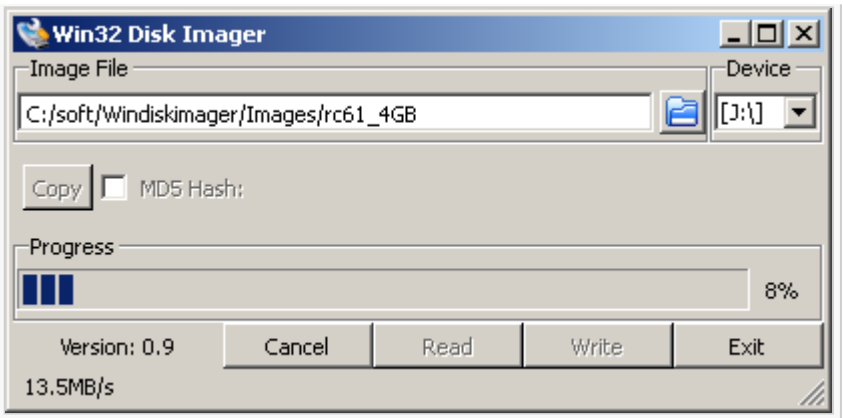
Prerequisites: You need a 4GB SDHC capable SD card reader connected or built in at your local computer.

There are two images, please select the one according to your SD card. If you do not have either a Sandisk or Kingston SD card, simply try. This is a fully working version of the given release, including the Radarcape installation. Hostname is radarcape.

Angstrom Distribution: Sandisk 4GB image with Release_131207.12.33

Angstrom Distribution: Kingston 4GB image with Release_131207.12.33

1. Get the Windiskimager from http://sourceforge.net/projects/win32diskimager/
2. Unzip the downloaded file, Windiskimager does not need installation, the extracted files work directly
3. Download the image from above link
4. Using Windiskimager, write the image to the SD card. **Carefully select the device that it is written to. Better unplug all other USB storages.**
5. After your Radarcape is back in operation, you may want to update to the latest release



Windiskimager in operation

True Miscellaneous

Packages required to create an SD card on a Radarcape:

- opkg install e2fsprogs-mke2fs rsync
- opkg install dosfstools

Radarcape:EMMC Howto

Required computer skills to execute this task: Intermediate

Contents [hide]

Scope

Beaglebone Black based devices of the Radarcape and FR24 box are working on an internal storage called eMMC while the elder Beaglebone White based units directly work with an SD card. In case of problems the contents of the eMMC can be restored with a recovery SD card. This can be done as often as you like, since the contents of the recovery SD card are not destroyed during the process.

Source of Recovery Data

Recovery Kit

If you have received such one, the recovery kit contains a Torx TX-10 screwdriver bit and a SD card.

Not necessarily the SD card is a 8GB and it will be labeled with the product to restore as well as probably with a version.

On request we can put recovery SD card images to our server so that you can make your own recovery SD card with a SD card writer.

Download SD Card Image

The recovery image can be downloaded from our server and be written to a standard SD card with WinDiskImager.

FR24 Receiver Recovery Image

The FR24 recovery image currently is not stored on the server.

Radarcape Recovery Image

- download RC-BBB-Angstrom-Writer.xz
- download the md5sum file RC-BBB-Angstrom-Writer.img.md5sum
- Perform a md5sum check before uncompressing and verify the integrity of the file.
- Uncompress the image with a xz uncompressing tool

The image currently is made for a 4GB Sandisk SD card and most probably will not fit to a 4GB Kingston card. However, flashing to a larger SD card seems to work fine.

- write the image to a SD card and then perform as below

Recovery Process

Unplug the unit from power

The other cables may be left connected or removed for convenience.

Open the front panel

Using a Torx TX10 screwdriver or the torx bit we've delivered open the two front panel

screws. Due to machine tightening, it may be hard at the starting point. Don't use a drill and please don't use doubtful screwdriver.

Set Boot Switch to SD Card Boot

There is a little slide switch on the top board. Shift this to the left hand position.

Insert the SD card

Insert the SD card. The label and text towards the upper side. Push in as far as it locks in the slot.

Apply power to the unit

Replug the power supply. You then should see the blue LEDs in the back flashing.

The recovery process takes about 1min for the Flightradar24 box and little longer for the Radarcape. If successful, the Radarcape/Flightradar24 receiver automatically powers off. All LEDs will be off then.

Remove SD Card

Remove the SD card.

Reset Boot Switch to eMMC Boot

Reset the boot switch to eMMC boot and unplug the power.

Finalisation

Finalize FR24 Recovery

First, plug your unit to the antenna and check that it is working, after you have verified this fix the front panel again and keep the recovery SD card in a safe place so that you can restore the system anytime again.

Finalize Radarcape Recovery

Note: This step can be ommited if you have done a Debian recovery, because Debian images already contain a running version.

For a Radarcape, you addtitionally need to install the latest firmware package as it is shown in Radarcape:Firmware_Versions.

Mind that you have just two minutes for the whole process after power on because afterwards the hardware watchdog will trigger a reset. However, nothing can go wrong if this happens, and you anyway know how to to restore the whole Radarcape meanwhile.

- Prepare a ssh terminal, e.g. Putty (see other examples here in the Wiki if you need more details how to achieve this)
- Connect your Radarcape to the internet and after that to power.
- Wait around 15seconds until the unit is up or simply try to establish a connection
- Use the ssh terminal to open a console terminal to the destination radarcape (eventually you need radarcape.local). Your account name is root and there is no default password.
- copy the opkg install command from the latest release to your clipboard and paste it to the console. This starts the installation sequence.

You then will see how the SW installs and afterwards see the front panel LEDS coming back into operation.

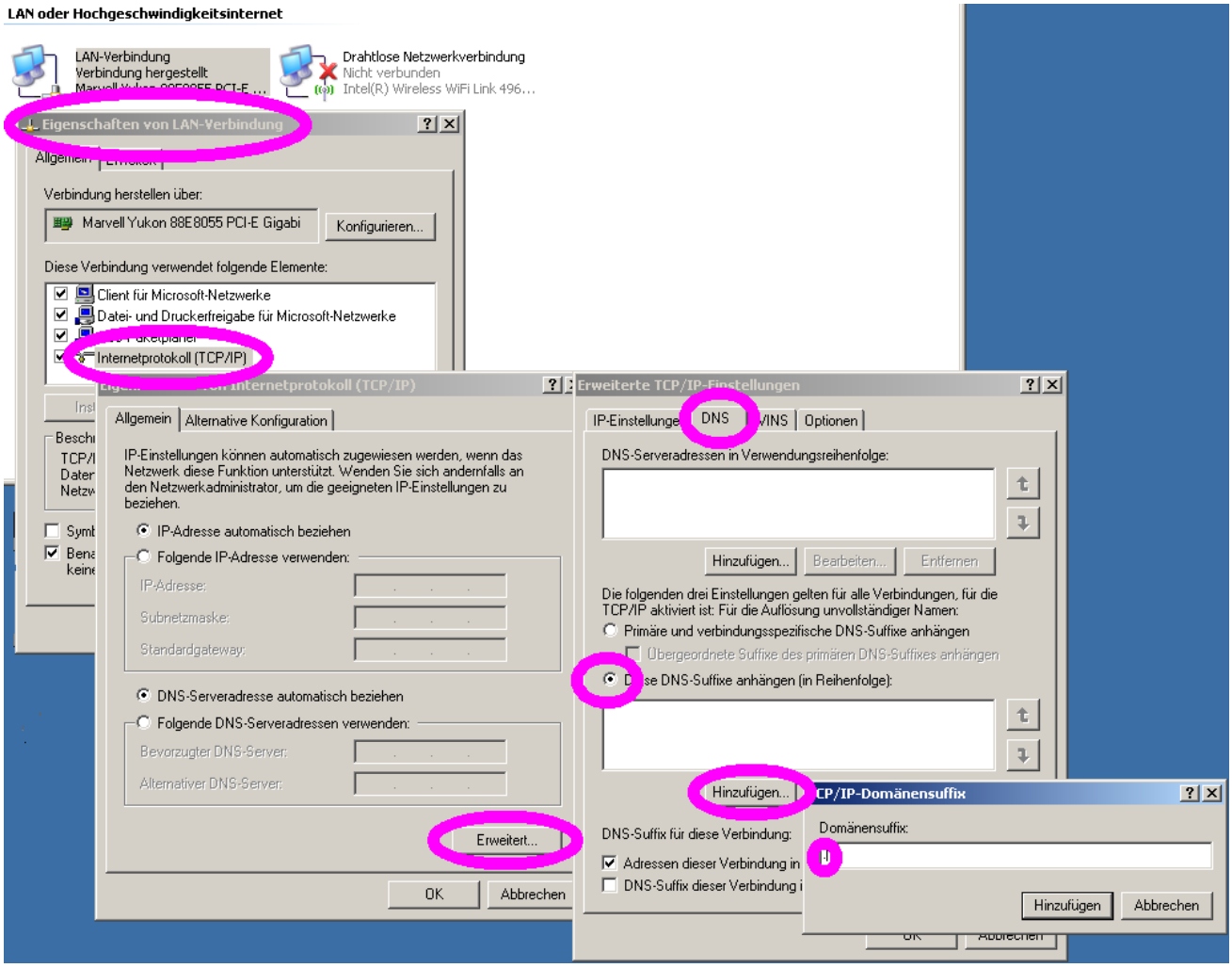
By the way, this is exactly the process that is performed in our factory.

Radarcape:ConnectivityFailures

Network connectivity problem

In some cases it may be that you cannot reach your Radarcape by entering just your hostname but only with the IP address. This is a routing problem of your Windows computer. I also had such a problem, the solution is described this forum entry.

Solution is that you either add a ".local" behind your Radarcape hostname, e.g. "radarcape.local", or you force the DNS lookups of Windows for single label domains by putting a single entry reading "." in the "*Append these DNS suffixes (in order)*" in the "*Advanced TCP/IP settings*" dialog of the network settings for TCP.



Navigation
Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links
modesbeast.com
Jetvision Shop
Beast Forum

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:DebianSupport

Contents [hide]

Scope

At the time of development, the operating system of the underlaying Beaglebone was a Linux Kernel 3.8 with Angstrom distribution. We found this was very stable in most use cases, just some networking issues came up, which we were able to fix with SW workarounds.

Another fault of this distribution was that due to a hardware misconfiguration the Linux system time did not run correctly. This was corrected by exclusivly using the GPS time within the Radarcape's application. However, some external tools and addons suffered from this.

You need to know that a distribution not only consists of Debian or Angstrom, there is a 2nd part with same importance, which is the Linux kernel itself. You may recognize this if we're now telling that the kernel version is 4.1.4-ti-r9 and the Debian distribution version 8.1. The kernel provides all functions which permit access to the hardware by the applications that are provided by the distribution. Somehow like a bios on a PC, just far more powerful.

Unluckily the support from Angstrom was discontinued. This meant that no longer general tools and applications were neither maintained nor updated. No more a new compiler, no longer any updates in the networking tools, and not at least, no support for the ARMHF architecture. *What is ARMHF?* Simply said it is the software interface to the floating point unit of the ARM processor. The ARM CPU on the Beaglebone has a floating point unit, but access to it was such a pain that direct software processing was about the same effort. With ARMHF this was corrected. (Note: CPU load of the Radarcape under ARMHF is nearly half as before)

The Beaglebone community decided to swap to Debian. We've started early trying to get a Beaglebone working based on Debian. At this time, also some changes were introduced into the kernel. In kernel 3.8 and 3.14 we found that the serial port performance was degraded significantly, confirmed about 2 months later: someone had applied a patch which did not make use of the serial port FIFOs. Waiting again - the cycle time is around 2-3 months per change - in about June 2015 we found that the latest release 4.1 fulfils all requirements, but is unstable. In a tremendeous community project the error became identified and corrected and now we have a stable kernel 4.1.4-ti-r9 which supports Debian 8.1 distribution.

Further Support Of Angstrom

Users of the Radarcape will now ask: *"Do I need to change to Debian?"*
Answer: Not at the moment. If you are happy with your Radarcape, and if you don't have problems, we will provide updates for those operating with Angstrom as long as possible. However, analysation of faults might be restricted to Debian in case that we expect it is not a Radarcape application error but one in the operating system.

Also, as long as the built-in update does not work, we will deliver all Radarcape based on Angstrom, despite some special projects.

Debian Installation

If in the text we're using "Debian", remember that this also includes the 4.1 Linux kernel. 4.1 is defined as a long time support (LTS) version.

Properties and Advantages of Debian

- Kernel 4.1 / Debian 8.1 is the latest (Mid 2015) state of the art Linux distribution
- Networking/LAN is much more stable and error corrections applied
- Due to the ARMHF architecture, CPU load is nearly half of the previous installations.
  - This should give headroom for others, like Flarm decoding.
  - We're now able to use latest compilers, which are far more efficient and easier to handle
- 2nd level tools no longer need to support the 3 years old operating system of the Radarcape
- Linux system time is correct and even using NTP as a standard.
  - As time of writing this, NTP Stratum 1 using the 1PPS pulse is already working but we've found a bug in one of the tools that are required.

How to install

Important Points:

- The installation is different wether you have a Radarcape based on Beaglebone-White or Beaglebone-Black.
- Before starting the installation, please write down, copy or print your configuration settings, as they will get lost during this process.
- If beeing asked for a login name and password from the web pages of the Radarcape, use *Administrator* and *radarcape*.
- Basically nothing can go wrong, as this recovery procedure can be repeated as often as you want. We're using the same procedure in production and testing, and it has been done successfully several thousand times.
- In difference to the Angstrom recovery, the Debian recovery already contains a working Radarcape application. There is no need to install the latest version manually.
- The Radarcape applications contained in the images are eventually different and later than those released for public, but changes are only internal and not significant for public. **The internal software update however is broken in this version**, so any later Radarcape software release must be done manually.

About end of September 2015 we will offer the two types of recovery SD cards in our shop.

Beaglebone White Installation

Beaglebone White Radarcapes can be identified by

- the serial number starts with 1302
- on the back side, you inside see green LEDs and not blue

Beaglebone-White based Radarcapes require reprogramming or swapping of the SD

card that is accessible behind the Radarcape front panel.

1. Download RC-BBW_Debian-4.1.0-rc8-bone9_151210.zip Packed size is around 212MB.
2. unzip the file, so you get an IMG file.
3. using Windiskimager, write the IMG to a SD card, minimum size is 4GB. See also Radarcape:SD_Card_Howto#Restoring_an_SD_card_with_Windows. Remember, not all 4GB cards are the same size, a Kingston 4GB may be too small, the reference is the 4GB Sandisk which came with the Radarcape. No problems are known if you're using a larger SD card.
4. Remove power from your Radarcape
5. Remove the front panel from the Radarcape. The screws are Torx TX10. Please do not use crazy tools and please do not use drills.
6. Swap the SD card
7. Apply power, wait for about 15sec, and after your front panel LEDs are showing ADS-B and GPS like before, use a web browser and access http://radarcape for verification
8. The current image for BB-White contains Release 151210.19.00.30. There may be new releases meanwhile, please read Radarcape:Firmware_Versions or simply browse to Configuration - Software Maintenance - Upgrade Radarcape Software and press the button [Update]

Beaglebone Black Installation

Beaglebone White Radarcapes can be identified by

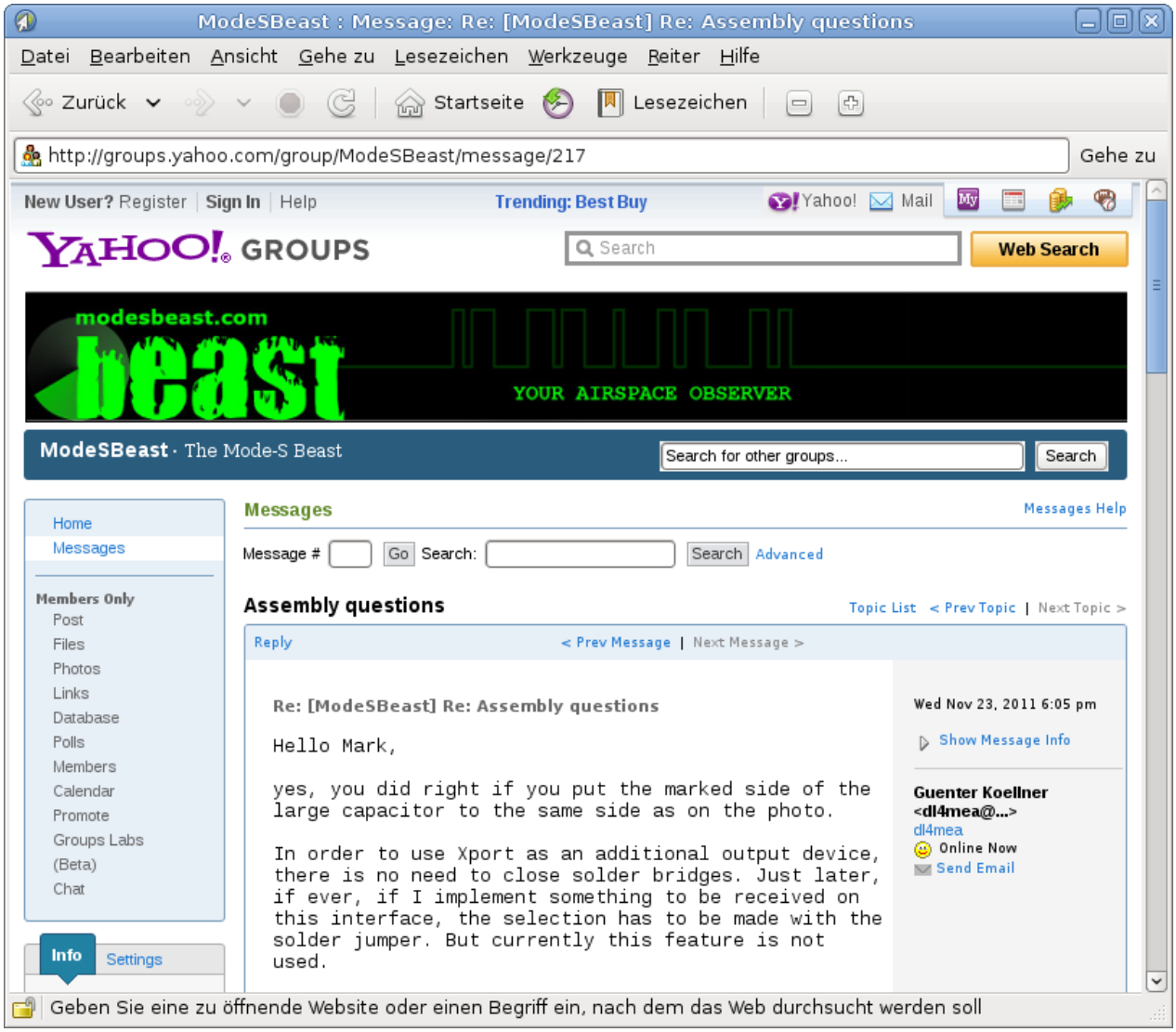
- the serial number starts with 1304
- on the back side, you inside see blue LEDs flashing

Beaglebone Black contain an internal eMMC card, which must be flashed from a separate SD card. This separate SD card can be removed afterwards and kept for any later help or reused for any other purposes.

1. Download RC-BBB-Debian-4.1.18-r57-rc160419-Writer.zip Size around 280MB.
2. unzip the file, so you get an IMG file.
3. using Windiskimager, write the IMG to a SD card, minimum size is 4GB. See also Radarcape:SD_Card_Howto#Restoring_an_SD_card_with_Windows
4. proceed like shown in Radarcape:EMMC_Howto#Recovery_Process

Radarcape:SupportAndHelp

Answers for most questions can be found in the Radarcape Wiki Pages, Radarcape FAQ and the Mode-S Beast Yahoo Group (especially, the forum provides quick reaction times due to contributions of the large Beast community).



A free Yahoo account is required to access the Mode-S Beast group!

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:Flightradar24 Feeding

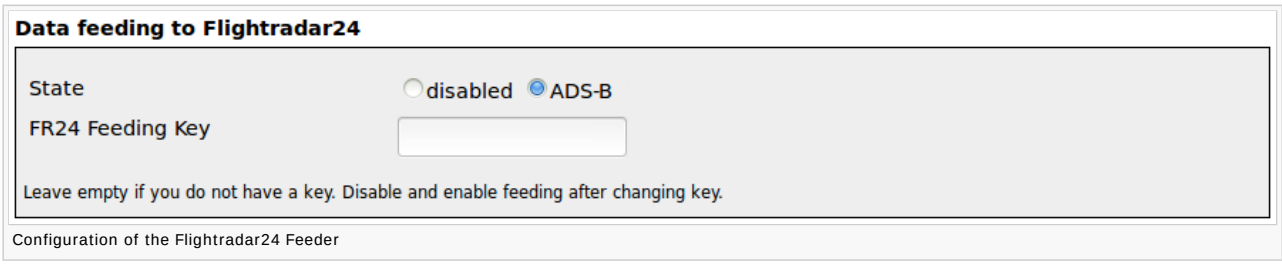
Required computer skills to execute this task: Beginner

The Radarcape comes with a pre-installed Flightradar24 (FR24) feeder which uploads your received data to flightradar24.com.

The Flightradar24 feeder options can be found on the *Settings* page of your Radarcape.

You may enter your private sharing key, that you got from Flightradar24, in the configuration dialogue. The sharing key can either be obtained via the Flightradar24 feeder software for Windoes (the Radarcape feeder uses the same key as the Windows feeder) or you can send an e-mail to *support<at>flightradar24.com* to request a key.

Flightradar24 feeding can be enabled and disabled via the *Settings* menu. Disabling takes approximately 5 seconds.



Configuration of the Flightradar24 Feeder

The Radarcape provides a status dialogue for all feeders. It shows the feeder output during the startup sequence and the latest 15 messages generated by the feeder.

If the key field is left empty, the software automaticaly generates a key from your latitude and longitude. As of the time of this writing such keys are not yet accepted by Flightradar24 and you will see error messages like shown below:

```
fr24key=@RC66,48.54391,11.24659
[i]FR24Feed v234 - built on Oct 11 2013/14:49:28
[i]Downloading configuration...OK
[i]Parsing configuration...ERROR
[failure]: Not found, check your key!
```

In this case, you should apply for your own sharing key at Flightradar24.

Radarcape:Planeplotter Feeding

Required computer skills to execute this task: Beginner

Attention: This page is about feeding data directly to the PlanePlotter network. Please look at the Using PlanePlotter with the Radarcape page if you want to connect your local PlanePlotter software with the Radarcape.

The Radarcape comes with a pre-installed feeder for the PlanePlotter network. The PlanePlotter feeder options can be found on the *Settings* page of your Radarcape.

Data feeding to Planeplotter

Feed every [sec]	◉ disabled ○ 15 ○ 30 ○ 60 ○ 120 ○ 300
Share positionless aircraft	○ disabled ◉ enabled
Registration number	
Authorization code	

Get your authorization key and registration here

Configuration of the PlanePlotter Feeder

You can request a registration number and authorization code for the PlanePlotter network via the Software Registration Request webpage.

The Radarcape provides a status webpage for every feeder. It shows the feeder output during the startup sequence and the latest 15 messages generated by the feeder.

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:Planefinder Feeding

Required computer skills to execute this task: Beginner

The Radarcape comes with a pre-installed Planefinder feeder which uploads your received data to the Planefinder webpage.

The Planefinder feeder options can be found on the *Settings* page of your Radarcape. You have to mail to share@pinkfroot.com in order to get your own Planefinder account for data sharing.

Data feeding to Planefinder

Feed every [sec]	◉ disabled ○ 15 ○ 30 ○ 60 ○ 120 ○ 300
Share positionless aircraft	○ disabled ◉ enabled
User name	
Authorization code	

You have to mail to share@pinkfroot.com in order to get your account for sharing

Configuration of the Planefinder Feeder

The Radarcape provides a status dialogue for all feeders. It shows the feeder output during the startup sequence and the latest 15 messages generated by the feeder.

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Main Page

Welcome to the Wiki of Radarcape and Mode-S Beast

The **Radarcape** and the **Mode-S Beast** are the most powerful Mode-S and Mode-A/C receivers and decoders on the spotters market.

While the Mode-S Beast is designed to be used as USB device and requires external decoding software, the Radarcape with its embedded Linux plattform adds own decoding and network connectivity.

This page provides technical support and information about the Radarcape and the Mode-S Beast.

Product Pages

Radarcape


Mode-S Beast

The **Radarcape** is a standalone device and includes the features of the Mode-S Beast plus many other features:

Easy usage

There are several ways of displaying the air traffic with nothing more than a web browser, even from multiple locations:

- Aircraft list
- 2D Map
- 3D Map (using Google Earth or similar)

Raw data access

- RAW data similar to the Mode-S Beast is available, additionally also pre-checked and pre-filtered
- Port 30003 data
- Raw data is available multiple and mixed, from several consumers at the same time, and with all services in parallel.
- Raw data is timestamped with a GPS synchronized absolute timestamp for high resolution multilateration

Low Level Access

- Several reports of the current air situation are available
- Own applications and programs can be installed in parallel to the operating software

Data Feeding

- The Radarcape comes with a Feeder for the Flightradar24 network and Planeplotter aircraft sharing (both switchable)

Remember, that all features above are available at the same time and in parallel.

The Radarcape connects to your PC via network, as such it is ideal also for remote locations.

The **Mode-S Beast** is a simple USB device, like a USB mouse,
and includes a receiver and frame decoder. It is completly USB powered.

It points with

- open protocol
- best of class reception range and sensitivity
- compatiblity to most of the decoding programs, like for example Planeplotter or ADSB-Scope
- easy usage
- small size

General Information

-

Information about antennas and filters can be found in the accessories section.

- For questions consult the Beast Forum.
- Conventions used in this Wiki

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Mode-S Beast:Contents

Contents [hide]

What is the Mode-S Beast?

- About the Mode-S Beast
- Advantages of a FPGA over a Microcontroller Solution

The Mode-S Beast Explained

- Features
- System Design
- Interfaces
- Performance

Mode-S Beast Maintenance and Software

- Important Notes
- Kit Assembly
  - Solder Jumper Settings
  - External miniADSB Connection
  - Hardware Handshake Set-Up
  - Lantronix Xport Assembly
  - Lantronix Xport Software Setup
  - BTM-222 Bluetooth Extension
- System Assembly
- DIP Switches
- LED Indicators
- Data Output Formats
- Data Input Formats
- Factory Testing

Mode-S Beast Support

- Software that can be used together with the Mode-S Beast
- Interfacing Planeplotter to the Mode-S Beast via USB
- Firmware Updates
  - How to perform firmware updates via USB
  - How to perform firmware updates with an USB blaster
- Hardware Versions
- Troubleshooting and Failure Diagnosis
- Failure Diagnosis using Putty

- Frequently Asked Questions

Miscellaneous

- The AVR Mode CRC Driver (Obsolete)

Accessories:Contents

Antenna Line Filter

- Purpose Of External Filters
- 1090MHz 3 Pole and 5 Pole Antenna Filter
- 1090MHz 3 Pole Filter Type 2

Antennas

- Medium Gain Mode-S Antenna (5dBi)
- jetvision LR-1090-N Stacked Omni Antenna (sold out)
- G7RGQ Omni Directiona Antenna (construction data)
- Quad Antenna in front of reflector wall (construction data)
- 10 Element Yagi for Mode-S (construction data)
- Sealing Tape (for waterproofing)

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Weblog

30.11.2014: New Radarcape software Release 141125.18.15 available

21.09.2014: Shops of jetvision.de and modesbeast.com were merged: http://shop.jetvision.de

03.12.2013: The new Mode-S Beast web-shop is online! Now the Radarcape is available for everyone.

11.03.2012: Note about Ghost frames added to release note of FW Version 1.40

07.03.2012: FW Version 1.40 released
Introduction of Mode-S Beast 2nd generation Radarcape

05.03.2012: Removed how to tweak the USB connection, it fortunately is not necessary and even more, sometimes made people believe that this is the right place where to set the serial interface parameters.

10.01.2012: FW Version 1.32a as a correction release to FW V1.32 released

08.01.2012: Putty Diagnosis Information added

03.01.2012: FW Version 1.32 released, more information about hardware version and solder jumpers of HW V1.1, troubleshooting added

21.12.2011: External Filters added, miniADSB cable colors corrected

15.12.2011: Antennas (G7RGQ, 10 element yagi, LR-1090-N super stacked omnidirectional antenna) added to the add-on section

13.12.2011: Updates on the webpage: LED bending, DIP-Switch updates

27.11.2011: New Mode-S Beast webpage online.

10.05.2011: After the first 5 devices from the commercial soldering have shown perfect functionality (just one resistor must be changed manually), the boards are avialable commercially.

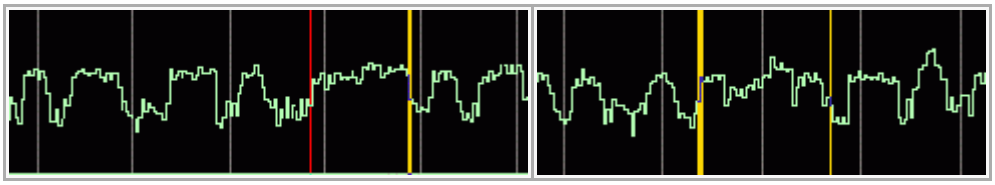
09.05.2011: Mode-A/C decoding is working. This can be enabled with DIP switch #10

01.05.2011: Forward error correction of 1 Bit errors is working. It can be disabled with DIP switch #9

18.04.2011: I today got the information that the production will happen on Wednesday after Easter. I also decided to let only 1CH and 2CH boards beeing produced. For those who urgently and only need a 4CH unit I will offer a predelivery of 2CH and later upgrade to 4CH with the next lot, which is already on the horizon. This is due to the fact that I need some more gates and without doing so a group of 4CH boards would exist without the chance of doing overlap decoding and Mode-A/C.

17.04.2011: Sorry, you did not hear much from me during the last time. Some familiy circumstances, preparation of my mother company's open house, preparation of the flight to the UK, a short excursion to those people hunting weather sondes (2 founds!), a EMC training, and another item kept me busy to above the ears. Unfortunately there is no news about the boards in production, I hope I get some more news tomorrow. I will also create a page about the Mode-S Beast operation in Andover (SP11 6EA), Martlesham-Heath (indoor use!) and Felixtowe (N 51° 58.577, E 001° 21.660°). It was very interesting, in deed.

24.03.2011: I am currently looking after the bit errors that the bit error driver tells me. It is pretty interesting, here are two samples of the signal when the driver indicated a bit error at this position. Even with an AD converter, it is hard to decide if this is a zero or a one:



I added a page about the CRC and Bit error correction driver

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Beast:Privacy policy

Datenschutzerklärung

Der Schutz der Privatsphäre der Webseiten-Besucher (im Folgenden: Besucher) ist der Betreiberin bei der Verarbeitung der persönlichen Daten ein wichtiges Anliegen. Die Verarbeitung der von den Besuchern erhobenen Daten erfolgt gemäß den Bestimmungen des Bundesdatenschutzgesetzes (BDSG) und anderen gesetzlichen Bestimmungen, die für den elektronischen Geschäftsverkehr gelten. Die Betreiberin bittet die Besucher deswegen, die folgenden Erklärungen sorgfältig zu lesen und gegebenenfalls auszudrucken oder abzuspeichern.

1. Automatisch erfasste nicht-personenbezogene Daten:

Beim Besuch von wiki.modesbeast.com und seiner Unterseiten (im Folgenden: modesbeast) werden durch den Webserver automatisch der Name des Internet Service Providers (IP-Addresse) des Besuchers, die Webseite, von der aus der Besucher kommt, die Webseiten, die der Besucher bei modesbeast besucht, sowie das Datum und die Dauer des Besuchs gespeichert. Die Betreiberin verwendet diese Informationen, um die Attraktivität von modesbeast zu ermitteln und deren Leistungen und Inhalte zu verbessern. Es werden hierdurch keine personenbezogenen Daten erfasst.

2. Erhebung, Verarbeitung und Nutzung personenbezogener Daten:

Personenbezogene Daten werden durch die Betreiberin nur erhoben, wenn diese durch den Besucher aufgrund eigener Veranlassung zur Durchführung eines Vertrages oder über das Kontaktformular mitgeteilt werden. Die vom Besucher erhobenen persönlichen Daten werden nur zu dem Zweck verwendet, für den der Besucher diese Daten der Betreiberin überlassen hat bzw. für deren Nutzung und Weitergabe er sein Einverständnis erteilt hat. Nach vollständiger Abwicklung des Vertragsverhältnisses bzw. des Nutzungszwecks werden die Daten des Besuchers mit Rücksicht auf die steuer- und handelsrechtlichen Aufbewahrungsfristen gespeichert. Spätestens mit Ablauf dieser Fristen werden die Daten gelöscht, soweit der Besucher in die weitere Verwendung dieser nicht ausdrücklich eingewilligt hat. Erhebungen von bzw. Übermittlungen persönlicher Daten an staatliche Einrichtungen und Behörden erfolgen nur im Rahmen zwingender nationaler Rechtsvorschriften. Die Mitarbeiter der Betreiberin sind vertraglich zur Verschwiegenheit und Beachtung des Datengeheimnisses verpflichtet worden.

3. Weitergabe personenbezogener Daten an Dritte:

Die Betreiberin beauftragt Dritte (Hosting-Provider) mit dem Betrieb der Webseiten und der Erfassung der Zugriffsdaten.

4. Cookies – Informationen, die automatisch auf dem Besucherrechner abgelegt werden:

Bei Besuch der Webseite unter modesbeast und deren Unterseiten werden eventuell Informationen in Form eines „Cookie“ temporär oder dauerhaft auf dem Computer des Besuchers abgelegt, die den Besucher bei seinem nächsten Besuch automatisch wiedererkennen. Cookies sind kleine Textdateien, die es z.B. erlauben, eine Webseite

den Interessen des Besuchers anzupassen. Wenn eine Ablage von Cookies nicht gewollt ist, wird darum gebeten, den verwendeten Internet-Browser so einzustellen, dass dieser Cookies von der Computerfestplatte löscht, alle Cookies blockiert oder eine Warnung abgibt, bevor Cookies abgespeichert werden.

5. Links zu anderen Webseiten:

Auf modesbeast und seinen Unterseiten können Links zu anderen Webseiten enthalten sein. Die Betreiberin ist für deren Datenschutzerklärungen oder sonstige Inhalte nicht verantwortlich.

6. Informationen zu den Besucherrechten:

Auf Aufforderung teilt die Betreiberin möglichst umgehend und schriftlich dem Besucher mit, ob und welche persönlichen Daten über den anfragenden Besucher bei der Betreiberin gespeichert sind. Sollten trotz der Bemühungen um Datenrichtigkeit und Aktualität falsche Besucherdaten gespeichert sein, werden diese auf eine entsprechende Aufforderung hin durch die Betreiberin berichtigt. Neben diesem Recht auf Berichtigung hat der Besucher gegebenenfalls ein Recht auf Sperrung und Löschung der von der Betreiberin erhobenen personenbezogenen Daten.

Für den Fall, dass weitere Fragen zur Erhebung, Verarbeitung und Nutzung der personenbezogenen Daten bzw. zur Berichtigung, Löschung oder Sperrung dieser bestehen sollten, wird um eine Kontaktaufnahme mit der Betreiberin gebeten. Die Kontaktadresse ist dem Impressum zu entnehmen.

Beast:About Us

Impressum

Günter Köllner Embedded Development GmbH
Am Rain 24
85256 Vierkirchen
Geschäftsführer Günter Köllner
dl4mea@modesbeast.com

VAT Tax ID: DE815462531
HRB München, HRB 208100

Webdesign:
Bianca Jensen
http://www.bianca-jensen.de

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Beast:General Disclaimer

Mit dem Urteil vom 12. Mai 1998 hat das Landgericht Hamburg entschieden, dass man durch die Anbringung eines Links die Inhalte der gelinkten Seiten ggf. mit zu verantworten hat. Dies kann nur dadurch verhindert werden, dass man sich ausdrücklich von diesem Inhalt distanziert. Für alle Links auf dieser Homepage gilt: Ich distanziere mich hiermit ausdrücklich von allen Inhalten aller verlinkten Seitenadressen auf meiner Homepage und mache mir diese Inhalte nicht zu eigen.

Auch wenn dieser Passus mittlerweile in der Rechtssprechung angezweifelt wird, zeigt er was ich damit eigentlich meine: Die verlinkten Seiten sind/waren zum Zeitpunkt des Verlinken thematisch mit meinem Thema verbunden, allerdings kann ich den Inhalt nicht auch automatisch als meine Meinung verantworten.

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:FlightAware Feeding

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Required computer skills to execute this task: Beginner

The Radarcape comes with a pre-installed FlightAware feeder which uploads your received data to the FlightAware webpage.

The FlightAware feeder options can be found on the *Settings* page of your Radarcape.

You have to create an FlightAware Account in order to feed data. Your account name must match with the MAC address of your Radarcape. The MAC address is displayed in the FlightAware settings of your Radarcape.

Data feeding to FlightAware

Feed every [sec]	(•) disabled ( ) 15 ( ) 30 ( ) 60 ( ) 120 ( ) 300
Share positionless aircraft	( ) disabled (•) enabled
User name is the MAC address	bc:6a:29:8c:cf:bf

Register here in order to get permission to feed

Configuration of the FlightAware Feeder

The Radarcape provides a status dialogue for all feeders. It shows the feeder output during the startup sequence and the latest 15 messages generated by the feeder.

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:Planeplotter

Required computer skills to execute this task: Beginner

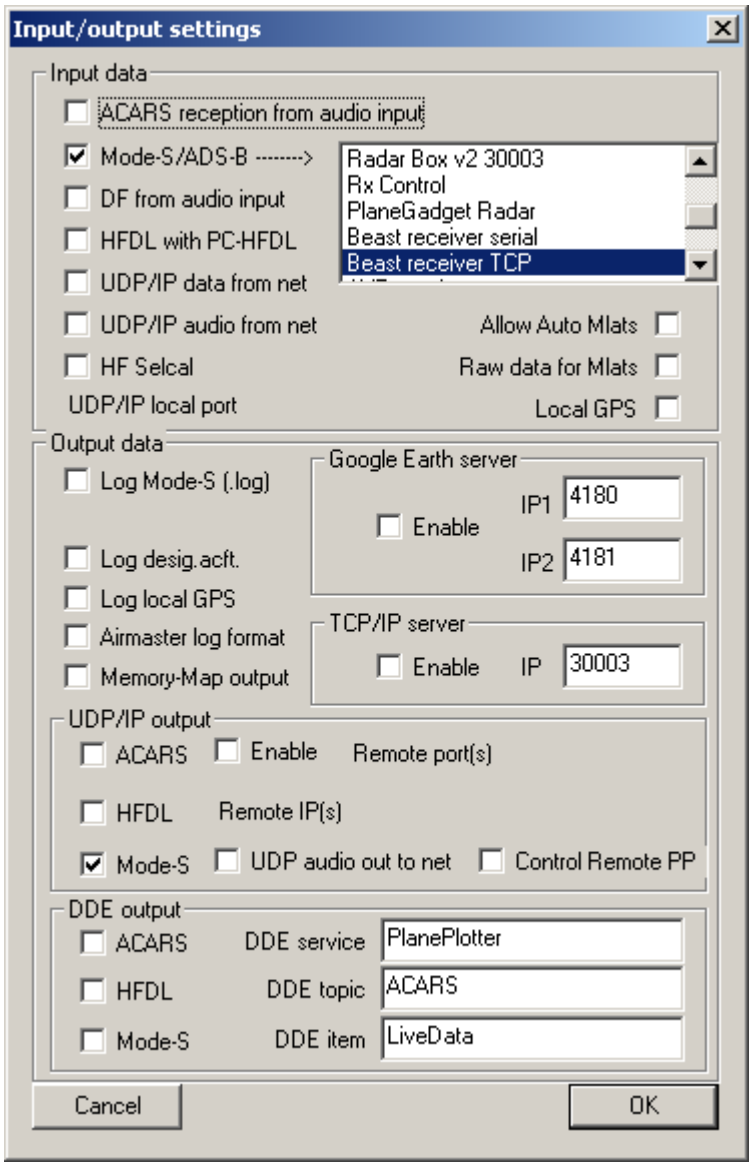
Using Planeplotter with the Radarcape

This manual describes how to connect Planeplotter with the Radarcape.

Step 1: Download, install, and start Planeplotter

- http://www.coaa.co.uk/planeplotter.htm

Step 2: Options -> IO-Settings



Step 3: Options -> Mode-S Receiver -> Beast Receiver -> Setup TCP/IP Client

Replace *beaglebone-3* with the DNS name or IP address of your Radarcape (e.g., *radarcape.local:10002*).

Step 4: Press green start button

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

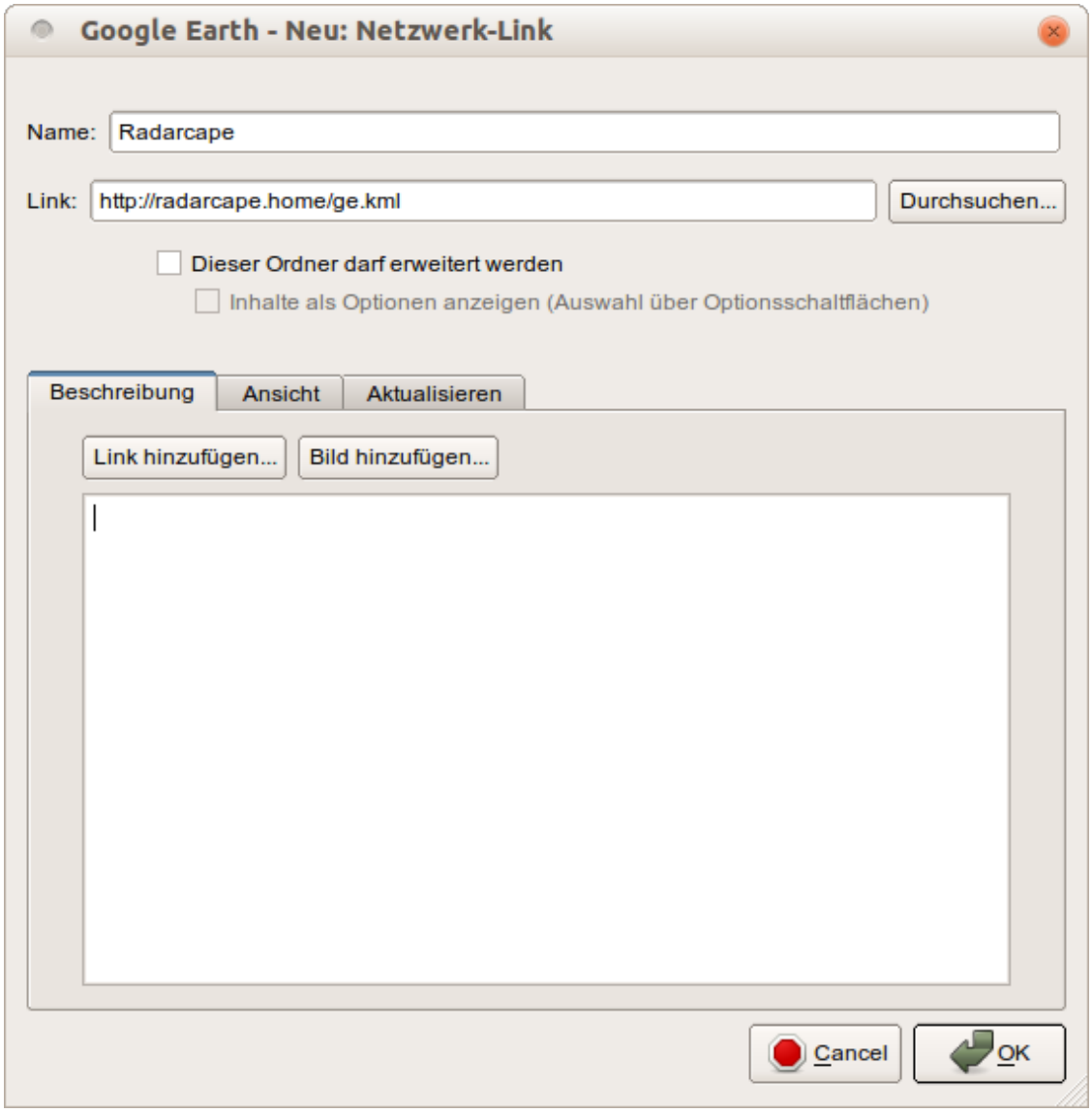
modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:Google Earth

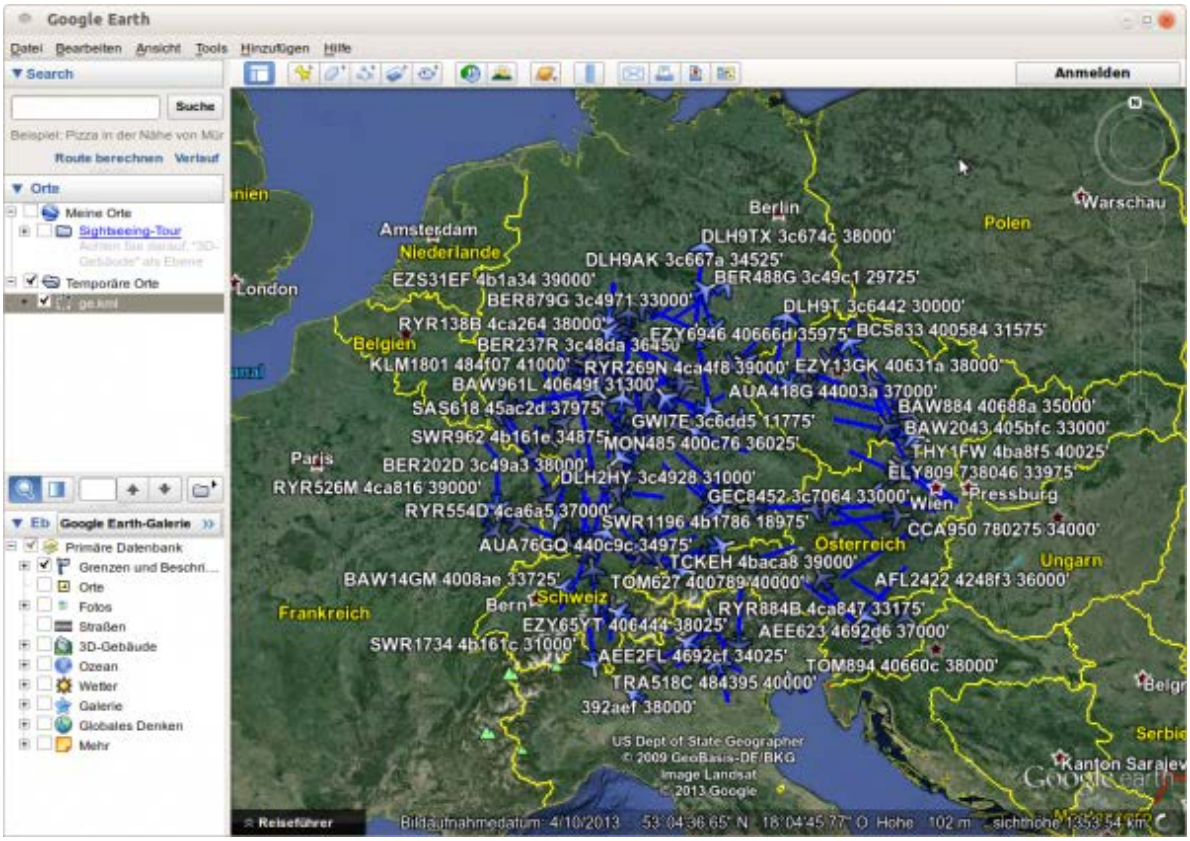
Required computer skills to execute this task: Beginner

Google Earth

- Open Google Earth
- Right click *Temporary Places => Add => Network Link*



- Set *Name* to *Radarcape*
- Set *Link* to *http://radarcape/ge.kml* (replace *radarcape* with the name or IP address of your device)
- Click *Ok* to store the settings



Google Earth feeded with ADS-B data from a Radarcape.

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

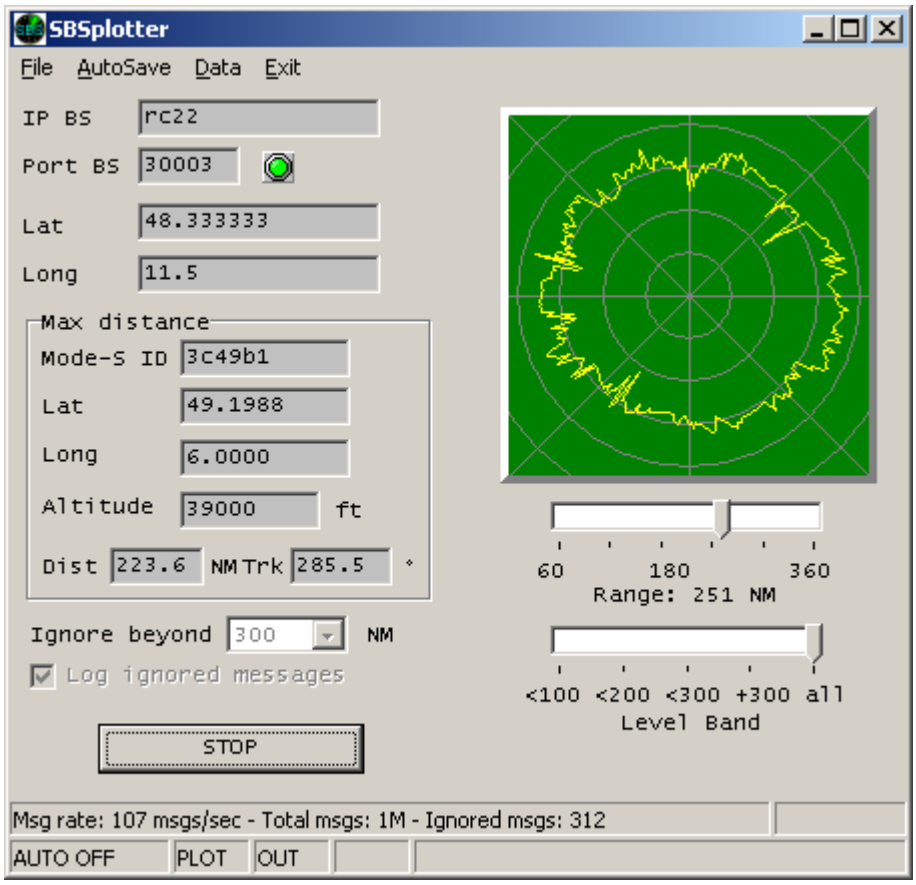
modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:SBS Plotter

Required computer skills to execute this task: Beginner

SBS Plotter

- SBS Plotter can be downloaded on the Jetvision website.
- Unzip the download archive and execute *sbsplotter1.exe*.



- Set *IP BS* to the IP address of your Radarcape.
- Set *Port BS* to *30003*.
- Set *Lat* and *Long* to the coordinates of your home location.
- Press the *Start* button.

Radarcape:ADSBScope

Required computer skills to execute this task: Beginner to Intermediate

ADSB-Scope using TCP Data Stream

Therefore, you need below virtual COM port, and then you simply use one side of the com0com link pair for TCP, the other one for ADSB-Scope.

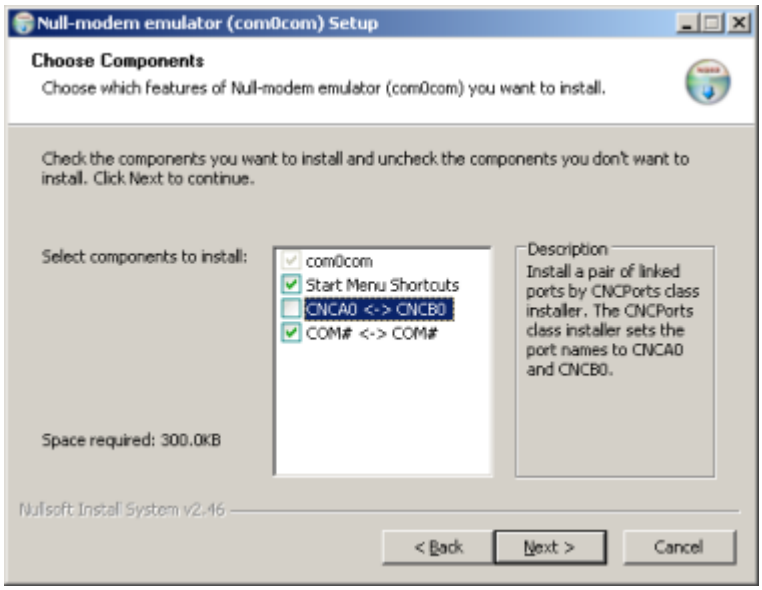
Setting up a virtual COM port in Windows

The Radarcape, as mentioned above, outputs data on TCP data streams. If applications do not offer a TCP connection to the device, this TCP data stream must be converted to a virtual COM port. We suggest the free COM0COM toolset for this.

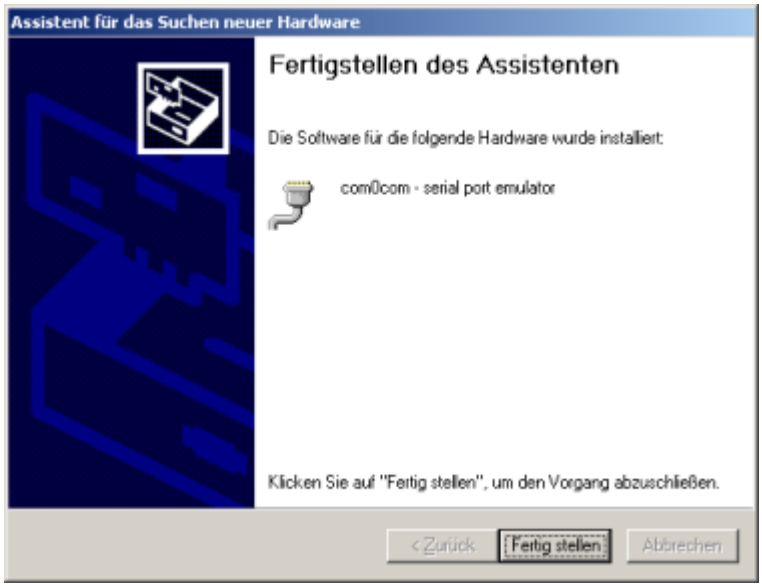
COM0COM Installation

Shown here for Windows XP (german version) - maybe someone can make Youtube videos for this process? If below description is not what you are looking for, have a parallel visit to http://com0com.sourceforge.net/doc/UsingCom0com.pdf

- Download the COM0COM package from http://sourceforge.net/projects/com0com
- Unzip the archive
- Within setup, uncheck the CNCA0↔CNCB0 pair, since this is not needed

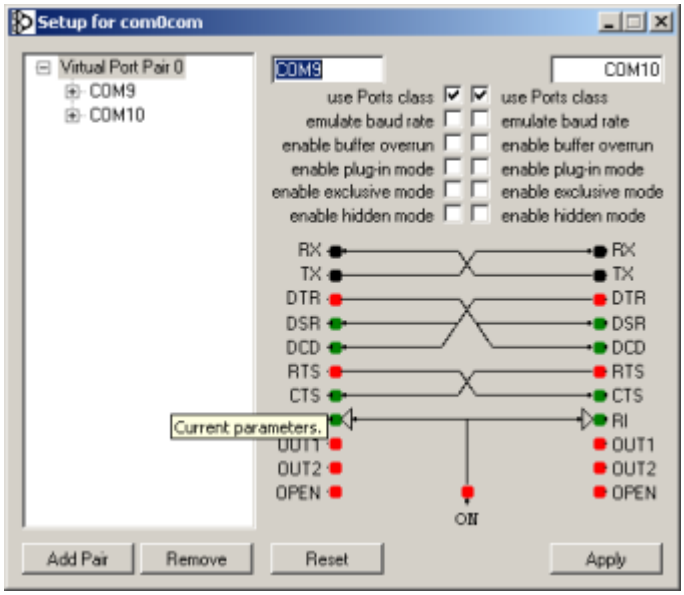


- When beeing asked that this is a non-signed driver, accept
- When beeing asked if the system should look on the internet for updates, say “no”, otherwise let it search for the correct driver on the local harddisk.
- When ready, the installation will finish with this message:



- You don't need to “launch setup”, in device manager the two ports beeing created will appear this way (COM9 and COM10):

More explanation: Now there is a virtual link. If you have two applications that each has a serial port for data input or output, they can be linked togeher, one using COM9, the other one using COM10. In the COM0COM setup it looks like this:



- One of the two COM ports is used for the application that needs the serial data, recommended here is **COM9**
- The other COM port needs a connection to the TCP data stream from the Radarcape. Therefore, download the com2tcp-1.3.0.0-386.zip package.
- Unzip the archive and copy com2tcp.exe best to the same folder as com0com, most probable C:\Program Files\com0com
- For Mode-S/AC data streams, open a windows command line and enter this line (modify port for the selected quality of service).

The command window is just good to see the error messages if such appear, later you can use any command line tool, like I do with the command line of Total Commander

```
"c:\Program Files\com0com\com2tcp" --baud 3000000 \\.\com10
beaglebone-3 10002
```

If in parallel you want to read in the TCP stream from the GPS port:

- Install a 2nd COM pair with com0com setup
- Install Trimble Studio
- One of the new COM ports then is used for Trimble Studio, the other one for a new TCP connection
- The command for com2tcp is

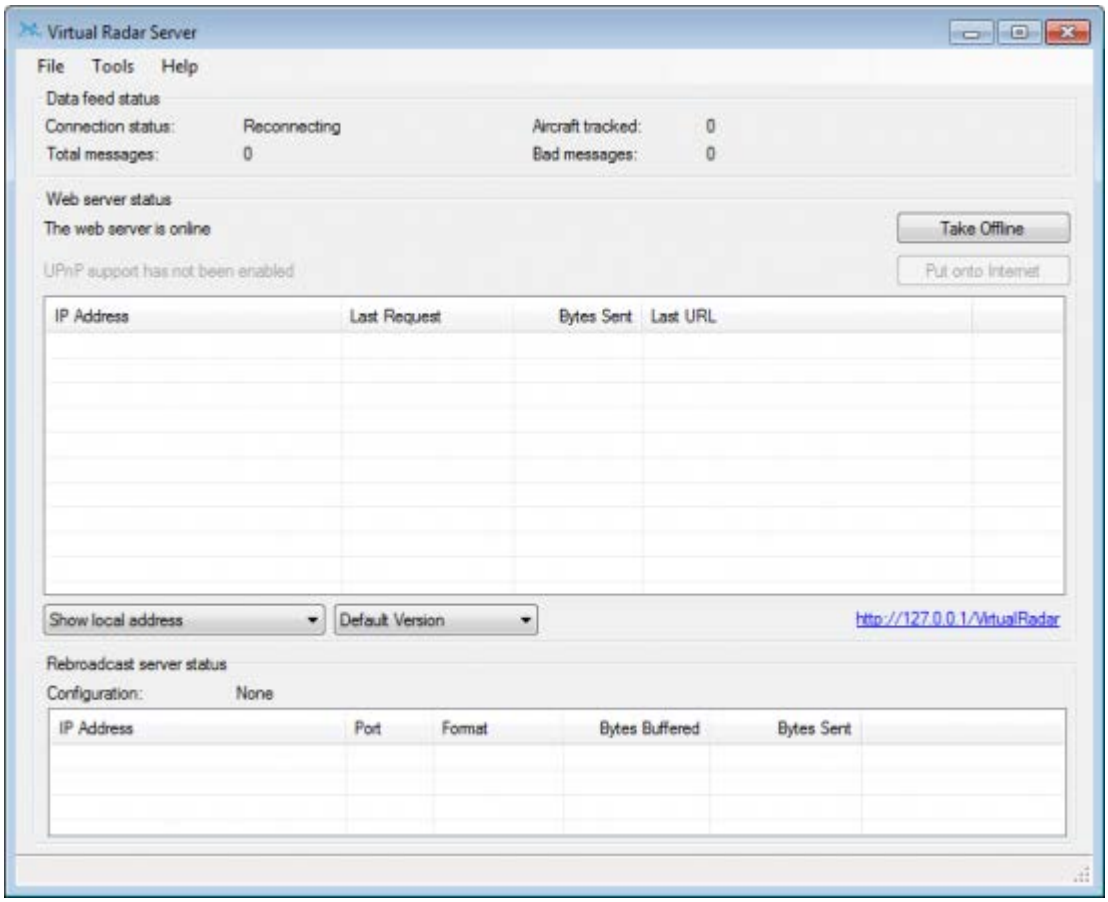
```
"c:\Program Files\com0com\com2tcp" --baud 9600 --parity o \\.\
<COM_PORT> beaglebone-3 10685
```

Radarcape:Virtual Radar Server

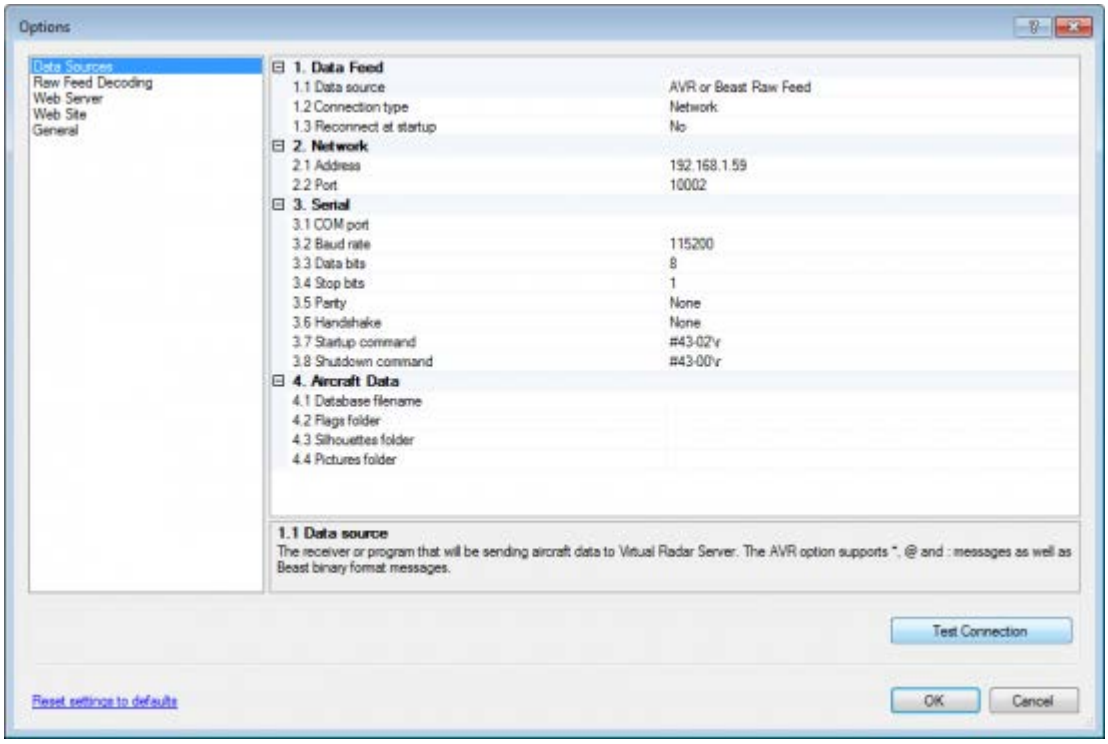
Required computer skills to execute this task: Beginner

Virtual Radar Server

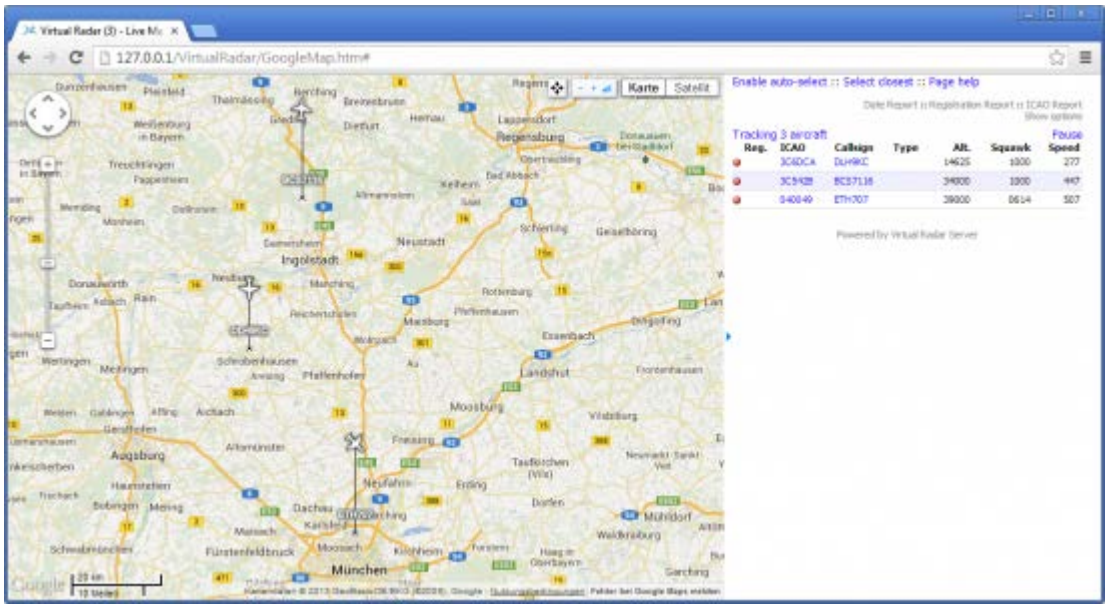
- Download and install Virtual Radar Server
- Start Virtual Radar Server



- Goto *Tools => Options*
  - Set *Data Source* to *AVR or Beast Raw Feed*
  - Set *Connection Type* to *Network*
  - Set *Address* to your Radarcape IP address
  - Set *Port* to *10002*, *10003*, or *10004* (port *10003* recommended)
  - Test the connection by clicking *Test Connection*
  - Save settings by clicking *Ok*



- Click *Take Online*
- Open Virtual Radar in your browser by clicking *http://127.0.0.1/VirtualRadar* in the GUI



Radarcape:XHSI

Required computer skills to execute this task: Beginner/Intermediate

Contents [hide]

eXternal High-fidelity Simulator Instruments (XHSI)

The eXternal High-fidelity Simulator Instruments (XHSI) are a open source simulation of cockpit instruments of a Boing 737NG and Airbus A320. They were made as an extension to the [www.x-plane.com X-Plane] flight simulator. But they can also be feed with data from the Radarcape.

Two additional tools and one database are required to connect the Radarcape with XHSI:

- modesmixer2 (Flanefinder Forum)
- rtl1090XHSI.exe (sources|binary)
- AptNav201310XP1000.zip (http://data.x-plane.com)

Setup on Ubuntu Linux

Step 1: Download Tools and AptNav-Database

Download the following files (or newer releases) and store them in /tmp:

- modesmixer2_x86_64_20150314.tgz (for 64 Bit) or modesmixer2_i386_20150314.tgz (for 32 Bit)
- rtl1090XHSI.exe
- XHSI-2-0-Beta-7.zip
- AptNav201310XP1000.zip

Step 2: Install Software

Install modesmixer2 on your system:

```
sudo mkdir /usr/local/modesmixer2
cd /usr/local/modesmixer2
sudo tar -xvzf /tmp/modesmixer2_x86_64_20150314.tgz
```

Install rtl1090XHSI:

```
sudo mkdir /usr/local/rtl1090XHSI
sudo mv /tmp/rtl1090XHSI.exe /usr/local/rtl1090XHSI/
sudo apt-get install mono-complete
```

Install XHSI:

```
sudo mkdir /usr/local/XHSI
sudo unzip /tmp/XHSI-2-0-Beta-7.zip -d /usr/local/XHSI/
sudo unzip AptNav201310XP1000.zip -d /usr/local/XHSI/
sudo chmod 755 /usr/local/XHSI/XHSI2_app/Linux/XHSI2
```

Step 3: Start and Configure Applications

Start modesmixer2 in console window 1:

```
/usr/local/modesmixer2/modesmixer2 --inConnect XXX.XXX.XXX.XXX:10003 --globes 31008:tableb:UUDD
```

Replace XXX.XXX.XXX.XXX with the IP address of your Radarcape.

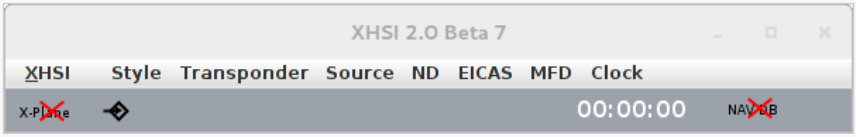
Start rtl1090XHSI in console window 2:

```
mono /usr/local/rtl1090XHSI/rtl1090XHSI.exe
```

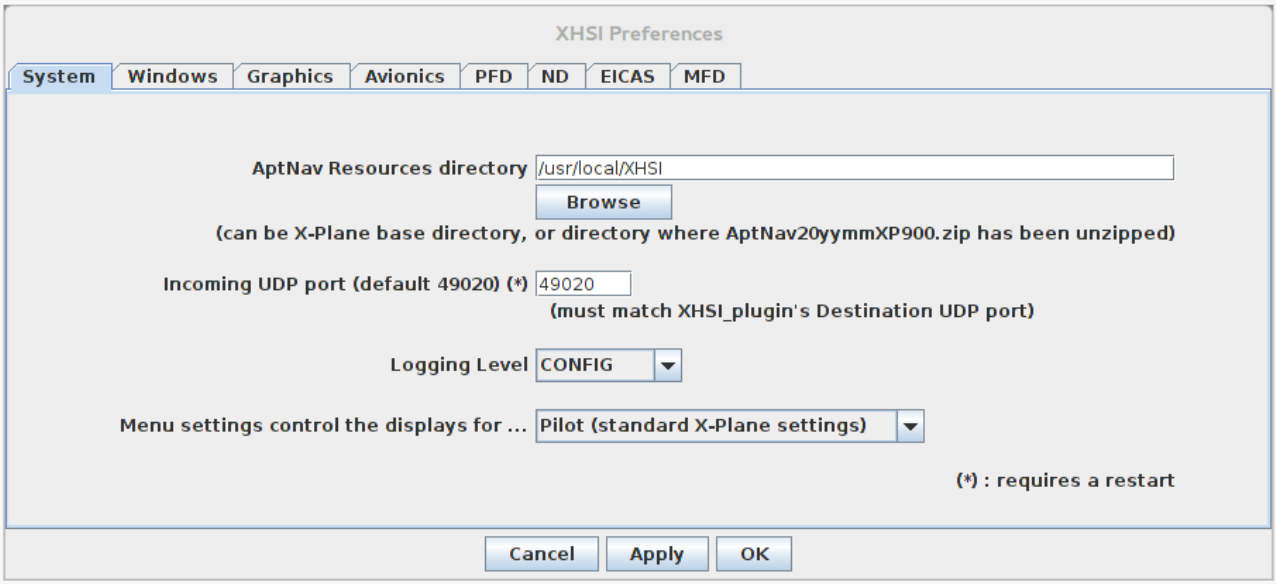
Start XHSI in console window 3:

```
/usr/local/XHSI/XHSI2_app/Linux/XHSI2
```

Setup the AptNav database within XHSI. Therefore, open the Preferences Menu from the XHSI main menu (XHSI -> Preferences ...):



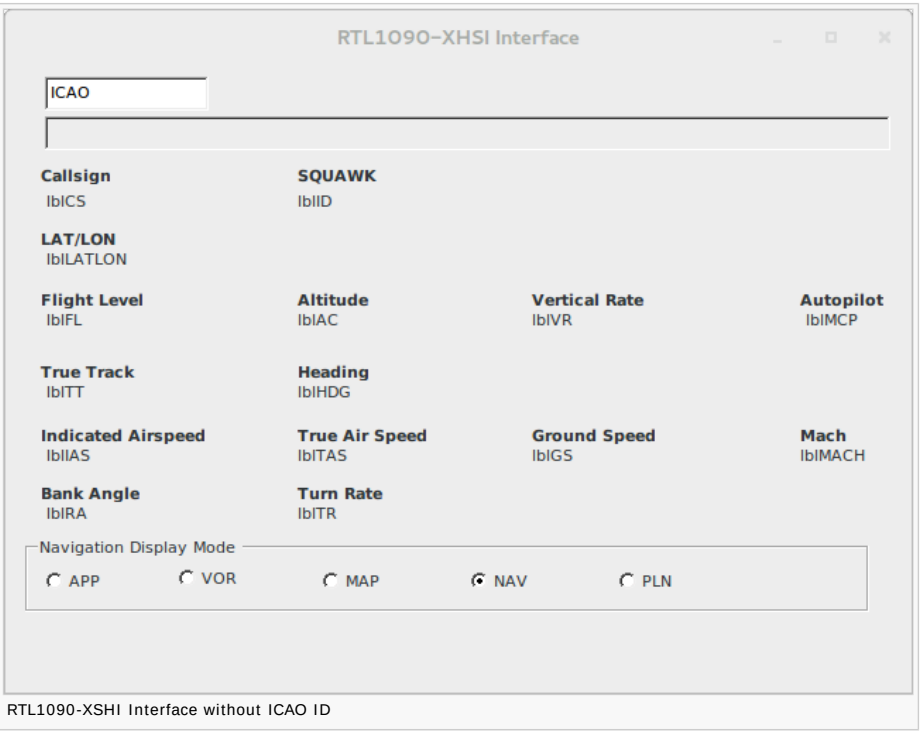
XHSI Main Menu



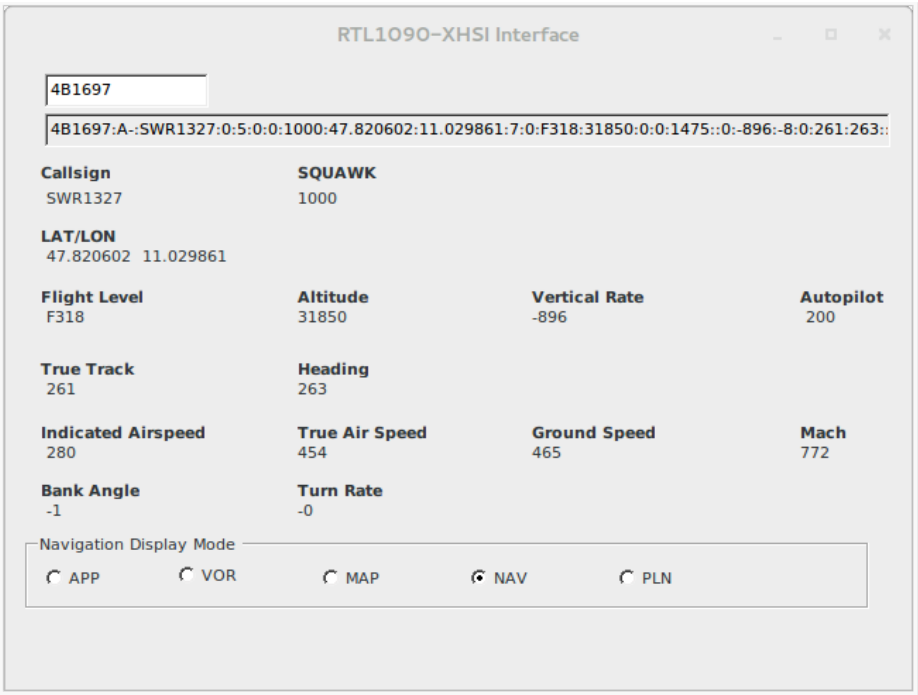
XHSI Preferences Menu

Set the *AptNav Ressources directory* to */usr/local/XHSI* and confirm the changes by clicking the "Ok" button.

Get the ICAO ID of the flight you want to monitor from your Radarcape (e.g., from the Aircraft List or 2D Map). Enter the ICAO ID (e.g., 4B1697) in the RTL1090-XHSI Interface.



RTL1090-XSHI Interface without ICAO ID



RTL1090-XSHI Interface with ICAO ID

Enjoy your eXternal High-fidelity Simulator Instruments

XHSI Primary Flight Display

XHSI Navigation Display (NAV Mode)

XHSI Navigation Display (PLN Mode)

XHSI Navigation Display (MAP Mode)

Additional Information

- ADS-B Onboard a 737 with Realtime Primary Flight and Navigation Display

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:Active Display Lite

Required computer skills to execute this task: Beginner

Active Display Lite

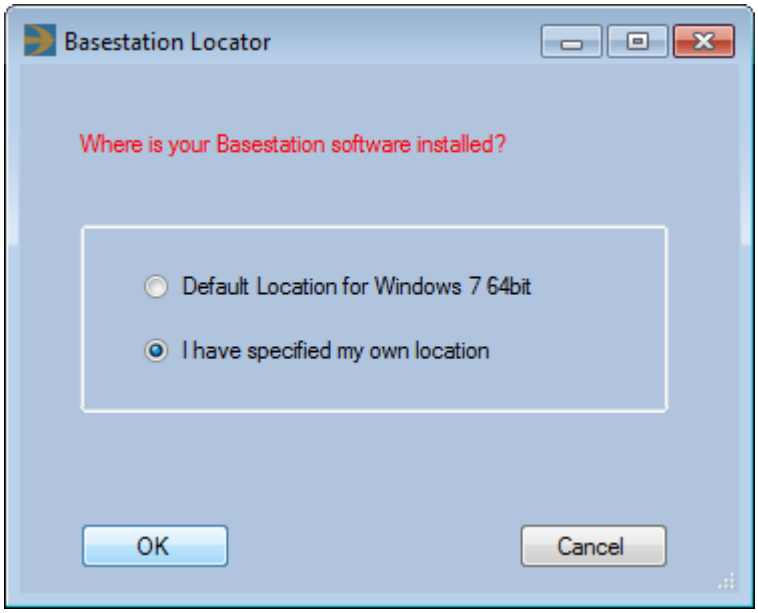
```
ADL REQUIRES BASESTATION SOFTWARE TO RUN ON PC.
FREQUENT ADL CRASHES REPORTED BY USERS.
```

Download Resources

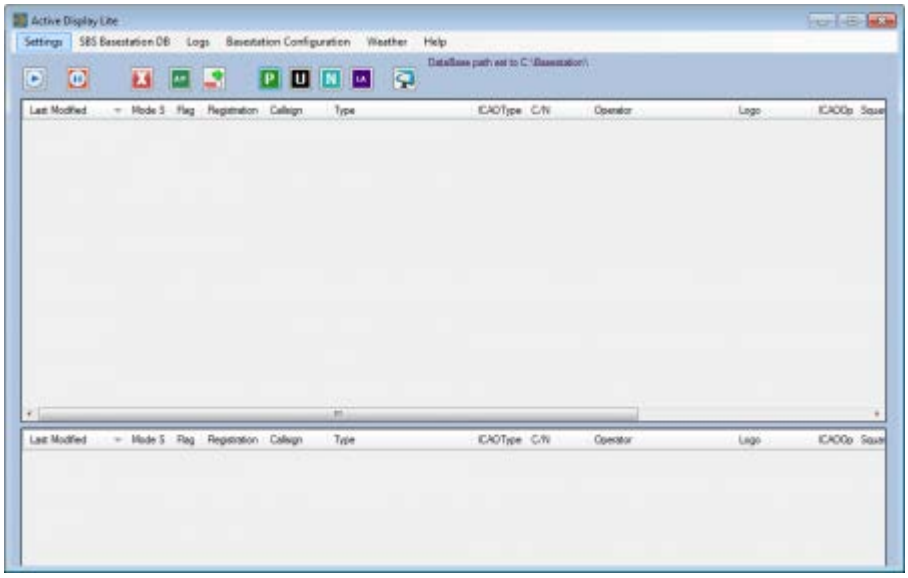
- Download *Basestation.sqb* file from http://pp-sqb.mantma.co.uk.
- Download *Active Display Lite* installer from http://www.gatwickaviationsociety.org.uk.

Installation and Configuration

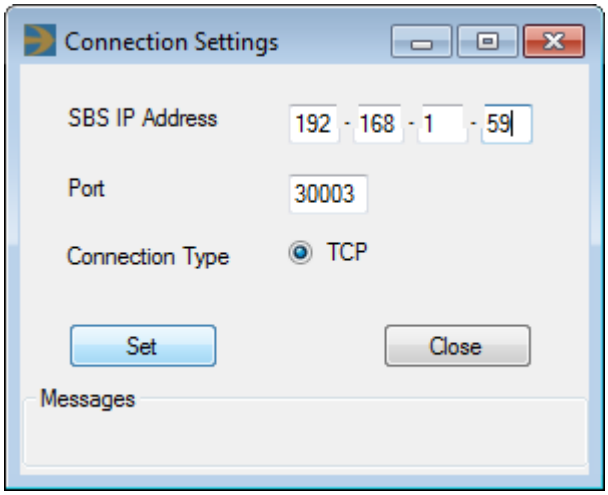
- Extract the *basestation.sqb* file from the archive (e.g., to c:\basestation\basestation.sql)
- Extract the Active Display Lite archive and execute *setup.exe*.



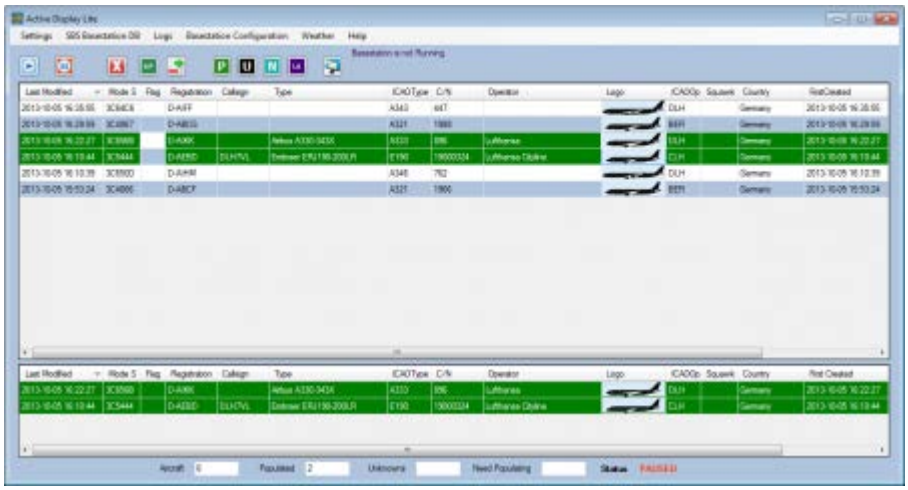
- Select *I have specified my own location" and commit with* Ok.
- Choose the directory in which you have saved *Basestation.sqb* in the file browser dialog.



- Go to *Settings => Receiver...*



- Set *SBS IP Address* to the address of your Radarcape.
- Leave the *Port* set to *30003*.
- Commit settings with *Set*.
- Close the dialog with *Close*.



- Start airspace observation by clicking the "play" symbol.

- Enjoy Active Display Lite.

Radarcape:Trimble GPS Studio

Required computer skills to execute this task: Intermediate

Accessing the GPS via PC Virtual Serial Port

The GPS module's serial data is streamed to TCP port 10685 and can be mirrored to any PC serial port with COM0COM/HUB4COM tools, so that Trimble Studio software can access and control it.

- Download and install COM0COM and HUB4COM from their Sourceforge page.
- Create one internal COM0COM link and rename the ports to COM numbers, e.g. COM100↔COM200.
- Execute in a command line

```
c:\Program Files\com0com\hub4com-2.0.0.0-386\com2tcp --baud 9600 --
parity o \\.\com100 <your_rcd_hostname_eg_rc22> 10685
```

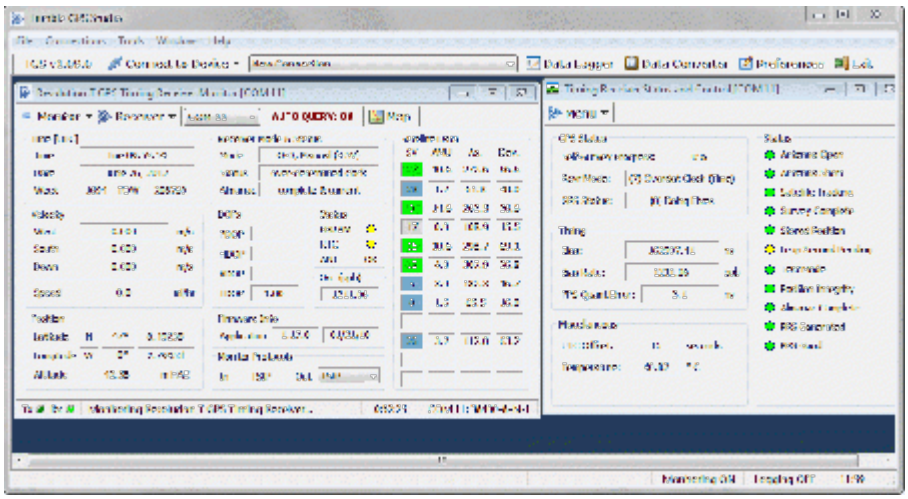
- Start Trimble Studio and connect to COM200

Note 1: In the com2tcp command line above, beaglebone-3 may be replaced with the Beaglebone's DHCP or statically assigned IP address. If running Windows 7, as the com0com drivers are not signed, the following procedure may be used to place Windows 7 in test mode.

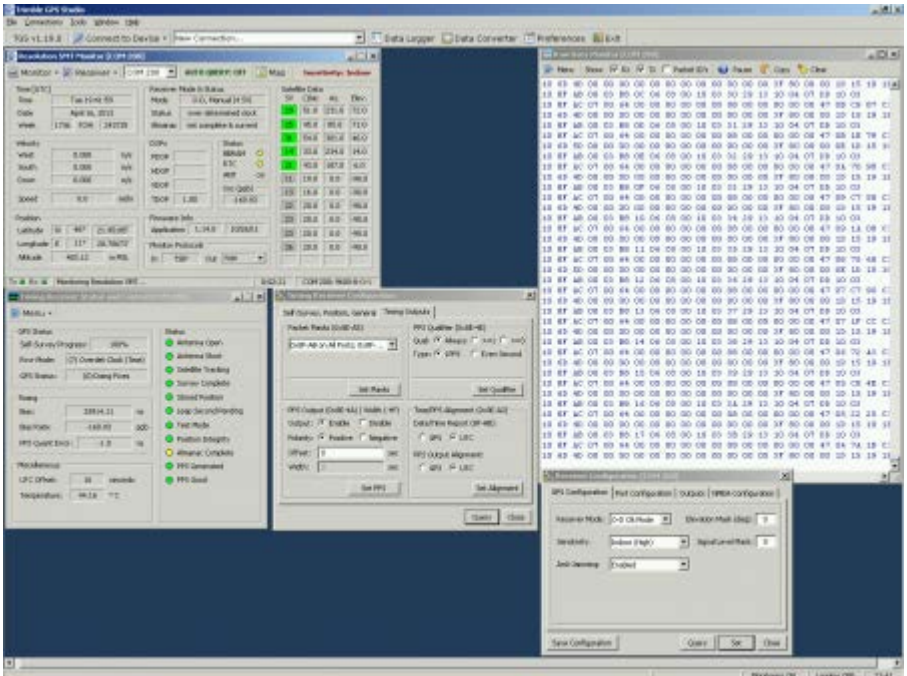
One editor has put this information in, but not the purpose when and where to use it:

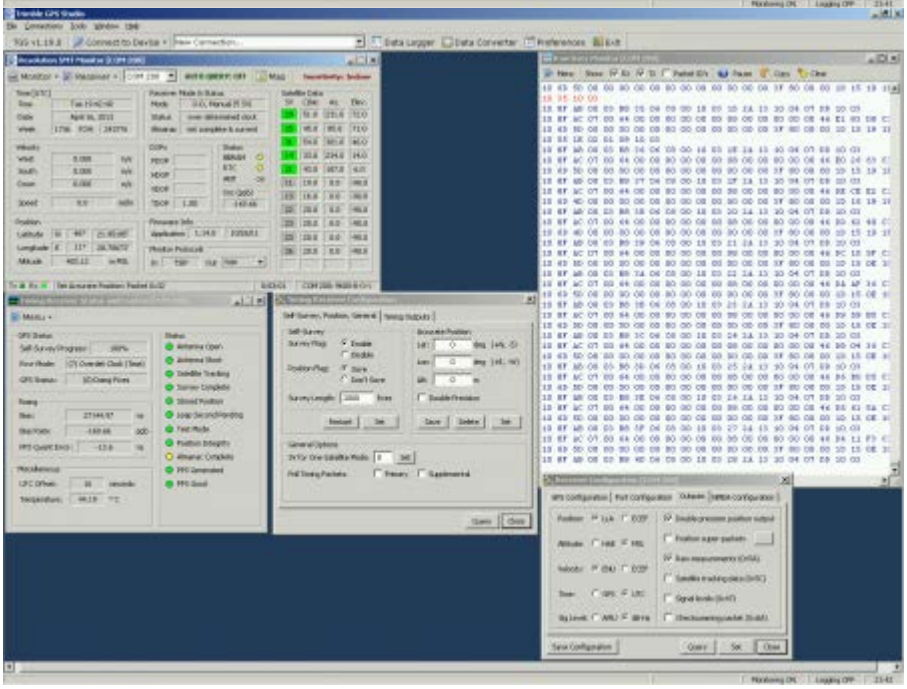
- Run command prompt as administrator
- Enter the command *bcdedit -set TESTSIGNING ON*
- Reboot (Windows will place a “test Mode” note on the bottom right of the desktop)

After the connection with the Trimble GPS is established, one should see a screen like this:



These two pictures show the correct initalisation of the Trimble SMT module:





Radarcape:NTP

Contents [hide]

Required Linux/Unix skills to execute this task: Advanced

NTP Synchronisation on Radarcapes with Debian

NTP time as a client

NTP as a client is standard for Debian. There is nothing to do in special.

NTP as a stratum 1 device

NTP Synchronisation on Radarcapes with Angstrom (deprecated)

As of state mid 2015, this is deprecated. NTP should be used with Debian only.

Radarcape and Local Time Synchronisation via NTP

In order to have correct time on Angstrom based Radarcapes, due to a hardware probelm in our cape board, a special kernel is required.

The Radarcape's hardware design will support true NTP using the GPS. Both, the Trimble serial data stream as well as the 1PPS, are available on the Beaglebone. However due to time constraints using those in order to create a Stratum 1 NTP server was not yet implemented.

Important Note: Timestamping of the Mode-AC/Mode-S raw data frames is _not_ done using the system time which is described here. All received frames are directly time stamped in the FPGA without any interaction of the local CPU. However, the __date__ in Port 30003 messages is always the Linux system date. The __timestamp__ instead is a GPS timestamp when the config is set to GPS timestamps and system time when the Radarcape operates in legacy 12MHz timstamp mode.

Meanwhile, all Radarcapes should synchronize their date and time via connmand and its NTP service. This is for example essential for the port 30003 protocol. So here is a list of checks in case that your Radarcape's date and time are not correct. Experience with this will tell us later which steps may be necessary in case that NTP does not work.

Basic System Checks

Check your Linux kernel version, connmand version and date/time.

```
root@rc12:~# uname -a
Linux rc12 3.8.13 #1 SMP Tue Jul 30 11:56:13 CEST 2013 armv7l GNU/Linux
root@rc12:~# connmand -v
1.4
root@rc12:~# date
Sat Nov  9 14:01:08 UTC 2013
```

connmand settings are in /var/lib/connman/settings. I added the PTB servers into this, as then we have a reliable 2nd server farm for NTP. You may do so, too.

```
root@rc12:~# cd /var/lib/connman
root@rc12:/var/lib/connman# cat settings
[global]
Timeservers=ptbtime1.ptb.de;ptbtime2.ptb.de;ptbtime3.ptb.de;0.angstrom.pool.ntp.org;1.angstrom.pool.ntp.org;2.angstrom.pool.ntp.org;3.angstrom.pool.ntp.org

OfflineMode=false

[Wired]
Enable=true

[WiFi]
Enable=true
```

Startup check: NTP is started once the network is up. journalctl can tell you what happend. Below you can see the first setting immediately after startup, and another synchronisation with 85sec offset later. If your Radarcape was running for longer, you will see a resynchronisation message about once each hour.

```
root@rc12:/var/lib/connman# journalctl --no-pager | grep -i ntp
Jan 10 06:54:20 rc12 connmand[122]: connmand[122]: ntp: time slew +436516610.758686 s
Jan 10 06:54:20 rc12 connmand[122]: ntp: time slew +436516610.758686 s
Nov 09 13:49:41 rc12 connmand[122]: connmand[122]: ntp: time slew +85.321813 s
Nov 09 13:48:15 rc12 connmand[122]: ntp: time slew +85.321813 s
```

Note: On a few devices I saw that after some days the time sync via NTP is suddenly lost. NTP update messages disappear in journalctl.

Links

http://stackoverflow.com/questions/11219832/what-is-the-best-way-to-run-ntpdate-at-reboot-only-after-network-is-ready

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:Samba on Beaglebone

Required Linux/Unix skills to execute this task: Advanced

If not already done, install the samba package:

```
opkg install samba
```

First, you need to edit */etc/samba/smb.conf*. Use *vi /etc/samba/smb.conf* from the command line. With the vi command */public* (with a dash, press *n* for next location, search for the section [public] and edit it to look like this:

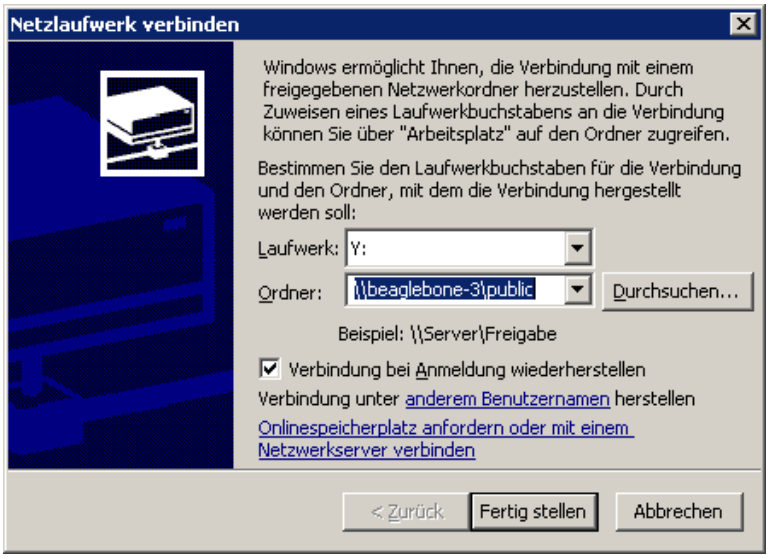
```
[public]
  comment = home root
  path = /home/root
  public = yes
  writable = yes
  printable = no
  guest ok = no
```

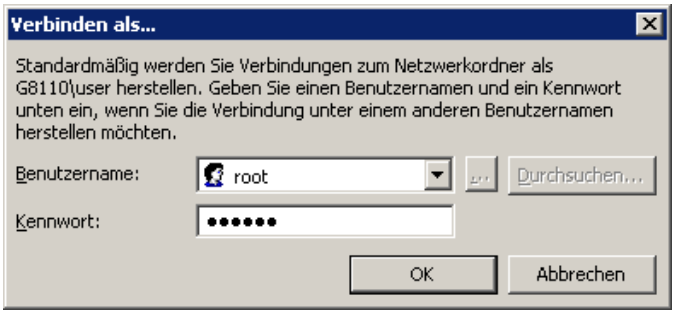
In order to get network access from a Windows PC to the Beaglebone, samba needs some configuration. Even if a user root already exists on the Beaglebone, it must be added for Samba separately:

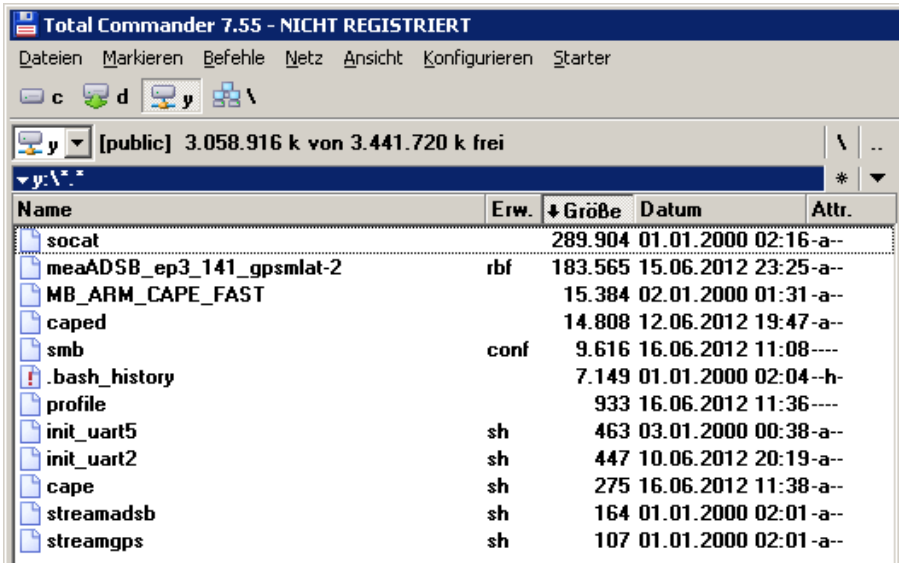
```
smbpasswd -a root
```

then type in a (simple) password, which of course you should remember properly.

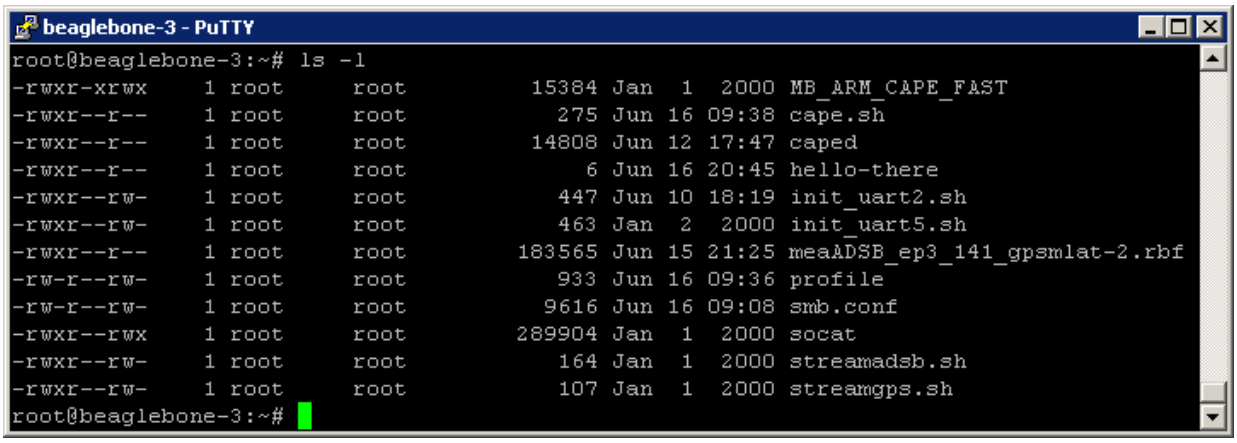
After that, you can mount a network drive from Windows to the Beaglebone. The identification on Windows is \\radarcape\public and the account that you have to use is root and your <password>. We're doing that with Total Commander, using **Network → Connect Network Drive** ending up like this:







Then, as a test, please copy a dummy file to the Beaglebone folder. I did so with the file “hello-there”. On a command window on Beaglebone, excute a ls -l and check if the group and owner are both root and not nobody:nobody.



Link: Samba Documentation

Radarcape:SSH Tunneling

Required Linux/Unix skills to execute this task: Advanced

Contents [hide]

SSH Tunneling Howto

Imagine you know a remote location with internet access, someway far off and not reachable from your home network, where you like to place a Radarcape. Unfortunately this network is not directly accessible from your home network, as no domain name (like modesbeast.com) is given to it. In that case you can let the Radarcape establish a tunnel connection to a known address, reachable by both, the user and the Radarcape. Such a SSH tunnel is secured by SSH, and this is the common way in networking.

Firewall Port Opening

One way to achieve access is to open the firewall into the remote network. This is known as port mapping. Therefore, consult your router/firewall manual and enter the ports required into the mapping table. In most cases it also is necessary to give it a DNS name, as mostly the IP adress is not stable. Unfortunately that weakens its security, and sometimes the administrator/owner of the far end local network does not permit doing so.

Tunneling of the receiver access through firewalls (SSH tunneling)

EXPERTS ONLY

Installation of a SSH tunnel

The SSH tunnel is a way to prepare a connection without opening a firewall. With this methode, the Radarcape establishes a connection to a given server and provides its ports right there.

Radarcape essentials

- Generate a SSH key pair on the local Radarcape

```
cd ~/.ssh
```

```
dropbearkey -t rsa -f id_rsa
```

- note the given public key string. If forgotten, you may later retrieve it with

```
dropbearkey -t rsa -f ~/.ssh/id_rsa -y
```

- set attributes of ~, .ssh and authorized_keys to 600
- check that on the Radarcape the /home/root is also owned by user root. Recently it occured sometimes that xroot:xroot was the owner

```
chown root:root /home/root
```

The file id_rsa is the so called private key, a file which never should leave your room. You should not transfer it over public lines. Another information, the so public key, which most probable starts with *ssh-rsa AAAAB3NzaC1y* and ends with *root@radarcape*, needs to be inserted into the file ~/.ssh/authorized_keys on the computer you want to connect (aka server). This phrase is safe to be published.

Server essentials

The server is the common connection point for the user and the Radarcape. It is not necessarily a computer for its own, it can even be the computer that hosts the PC application, or even a Radarcape by its own.

Copy the public key given from above command to the server folder ~/.ssh/authorized_keys. Maybe you need to use an editor in order to edit the given single line public key to an existing file. Mind that the attributes of ~, .ssh and authorized_keys are set to 600.

If correctly done, you must be able to login via ssh from the Radarcape to the server without entering a password. As long as this does not work, there is still some fault. dropbear ssh, which is used on the Radarcape, does not automatically use the ssh keys. Therefore, you need to specify

```
ssh -i ~/.ssh/id_rsa <your_servername_or_ip>
```

Radarcape 2nd step

On the local Radarcape, add the command below to cape.sh. Remember that cape.sh runs without user settings, so you need to specify the path to the SSH key absolute.

```
./autossh -M 6667 -f -p <server_ssh_port> -i /home/root/.ssh/id_rsa -N
 -R *:8002:localhost:80 -R *:1302:localhost:10003 -R
*:2202:localhost:22 root@<server_domain> &
```

Now the local Radarcape's ports 80, 10003 and 22 are accessible on <server_domain> under port 8002, 1302 and 2202.

Note: autossh is a tool we have locally compiled and will provide on the server later

Server Settings

If the server is also a Radarcape, and in case that you want to get access from external devices to the ports through the tunnel, you need to add switch "**-a**" to the dropbear startup file **/lib/systemd/system/dropbear@.service**.

Radarcape:Contents

Radarcape:Acarsdec

Contents [hide]

acarsdec on the Radarcape

acarsdec is a software which uses a DVB-T stick in order to decode up to 4 channel with ACARS signals within a 1MHz band segment. The front panel extension port of the Radarcape may be used to plug such a DVB-T stick. The decoding process requires some mathematical functions and obtains plenty of processing resources. Due to that it is not guaranteed to run on a heavily loaded Radarcape and/or together with Mode-AC or FR24 feeder. You may need to disable some features.

Installation requires some skills in handling a console but at the end is easy.

The original places where to find the used components are

- ACARS decoder for Raspberry Pi
- Acarsdec
- How to blacklist kernel modules
- ARM Cortex-A Processors and GCC Command Lines

Installation requires some skills in handling a console and edit files, but at the end is easy.

Installation on a Debian distribution based Radarcape

Prepare Installation (Debian)

First, you need to install build tools on the Radarcape. You may need to enter one line after the other separately.

```
apt-get update
apt-get upgrade -y
apt-get install -y build-essential cmake git libusb-1.0-0-dev
sync
```

This will take some time.

Install librtlsdr (Debian)

Install rtl-sdr library. You need to blacklist the kernel DVB-T stick driver before using this new driver.

```
echo blacklist dvb_usb_rtl28xxu >> /etc/modprobe.d/fbdev-blacklist.conf
```

Then you are ready to build and install the library.

```
git clone git://git.osmocom.org/rtl-sdr.git
cd rtl-sdr
mkdir build
cd build
cmake ../ -DINSTALL_UDEV_RULES=ON
make
make install
ldconfig -v
cd ..
cp -v rtl-sdr.rules /etc/udev/rules.d/
```

```
sync
```

After this step, reboot your Radarcape without DVB-T stick connected. Wait a few 10 seconds, plug your stick and check with command dmesg if the stick became recognized correctly:

```
[   30.254667] musb-hdrc musb-hdrc.1.auto: VBUS_ERROR in a_wait_bcon (91, <VBusValid), retry #1, port1 00000104
[   30.676143] usb 1-1: new high-speed USB device number 2 using musb-hdrc
[   30.827718] usb 1-1: New USB device found, idVendor=0bda, idProduct=2838
[   30.827740] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[   30.827751] usb 1-1: Product: RTL2838UHIDIR
[   30.827760] usb 1-1: Manufacturer: Realtek
[   30.827769] usb 1-1: SerialNumber: 00000001
[   30.841091] PM: am33xx_prepare_push_sram_idle: EMIF function copy failed
```

Next, check with the command rtl_test if it works:

```
root@rc66:~# rtl_test
Found 1 device(s):
  0:  Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4 15.7 16.6 19.7 20.7 22.9
25.4 28.0 29.7 32.8 33.8 36.4 37.2 38.6 40.2 42.1 43.4 43.9 44.5 48.0 49.6
[R82XX] PLL not locked!
Sampling at 2048000 S/s.

Info: This tool will continuously read from the device, and report if
samples get lost. If you observe no further output, everything is fine.

Reading samples in async mode...
lost at least 232 bytes
```

One 'lost' message is ok during startup, but there should none appear later.

Install acarsdec (Debian)

With a simple **cd** command, go back into your root folder.

acarsdec source package can be found on sourceforge. There is a green button with the latest version, click on it and then look for the ' *direct link* ', which you should cut & paste, edit and execute with wget like shown below.After uncompressing the archive, change to the folder that became created by tar.

Note that at the time of writing version 3.2 was latest, it might be different sometimes later.

```
cd
wget http://downloads.sourceforge.net/project/acarsdec/acarsdec/3.2/acarsdec-3.2.tar.gz
tar xvf acarsdec-3.2.tar.gz
cd acarsdec-3.2
```

Now it comes... the complicated part is that ALSA lib is not necessary for the build here, and it seems that the Angstrom distribution that is used on the Radarcape has a different library structure.
That means you need to edit the Makefile:

- comment out all lines with CFLAGS with a # sign in the first column.
- insert a new CFLAGS line

```
CFLAGS= -g -O3 -flto -mcpu=cortex-a8 -mfpu=vfpv3 -ffast-math -funroll-loops -pthread -D
WITH_RTL
```

- in the line showing LDLIBS, remove -lasound, so it looks like

```
modified: LDLIBS=  -lm -pthread  -lrtlsdr
```

Start compilation with the command *make*

You then should see some output like this:

```
root@radarcape:~/acarsdec-3.2# make
cc -g -O3 -flto -mcpu=cortex-a8 -mfpu=vfpv3 -ffast-math -funroll-loops -pthread -D WITH_RTL -D
 WITH_SNDFILE   -c -o acarsdec.o acarsdec.c
cc -g -O3 -flto -mcpu=cortex-a8 -mfpu=vfpv3 -ffast-math -funroll-loops -pthread -D WITH_RTL -D
 WITH_SNDFILE   -c -o acars.o acars.c
cc -g -O3 -flto -mcpu=cortex-a8 -mfpu=vfpv3 -ffast-math -funroll-loops -pthread -D WITH_RTL -D
 WITH_SNDFILE   -c -o msk.o msk.c
cc -g -O3 -flto -mcpu=cortex-a8 -mfpu=vfpv3 -ffast-math -funroll-loops -pthread -D WITH_RTL -D
 WITH_SNDFILE   -c -o rtl.o rtl.c
cc -g -O3 -flto -mcpu=cortex-a8 -mfpu=vfpv3 -ffast-math -funroll-loops -pthread -D WITH_RTL -D
 WITH_SNDFILE   -c -o air.o air.c
cc -g -O3 -flto -mcpu=cortex-a8 -mfpu=vfpv3 -ffast-math -funroll-loops -pthread -D WITH_RTL -D
 WITH_SNDFILE   -c -o output.o output.c
```

```
cc -g -O3 -flto -mcpu=cortex-a8 -mfpu=vfpv3 -ffast-math -funroll-loops -pthread -D WITH_RTL -D
 WITH_SNDFILE   -c -o alsa.o alsa.c
cc acarsdec.o acars.o msk.o rtl.o air.o output.o alsa.o -o acarsdec -lm -pthread  -lrtlsdr
```

Execute acarsdec. It might be good if you don't see messages from specific frequency to leave them out from decoding. The command also needs to be modified for networking according to its original description if you like to send decoded messages to other computers.

```
./acarsdec -o1 -v -p 79 -g 400 -r 0 131.725
```

And now you will see its output.

```
root@radarcape:~/acarsdec-3.2# ./acarsdec -o1 -v -p 79 -g 400 -r 0 131.525 131.725 131.825

Found 1 device(s):
  0:  Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Tuner gain : 40.200000
Set center freq. to 131750000Hz
Exact sample rate is: 2500000.107620 Hz
Decoding 1 channels
#1 (L:-32 E:0) 08/09/2016 10:34:58 .A6-EDU EK0015 2 Q0 S76A
#1 (L:-38 E:0) 08/09/2016 10:34:59 .PH-BXK KL1652 G 10 M67A ETA93A08103456KLM1652 LIPZEHA
#1 too many parity errors
#1 (L:-33 E:0) 08/09/2016 10:35:06 .A6-EDU EK0015 2 Q0 S76A
```

Tweaking (Debian)

Probably it makes sense to disable automatic power savings on the CPU.

```
apt-get update
apt-get install -y cpufrequtils
cpufreq-set -g performance
```

This is not automatically permanent.

Installation on an Angstrom distribution based Radarcape (may work)

Angstrom is outdated and this description is only hold for completeness. Especially because Debian is able to use the floating point unit of the ARM, for Acarsdec Debian is strongly recommended.

Prepare Installation (Angstrom)

First, you need to install build tools on the Radarcape.

```
opkg update
opkg install packagegroup-core-buildessential
opkg install cmake
opkg install libusb-1.0-dev
opkg install libsndfile-dev
sync
```

This will take some time.

If problems appear somehow around something that is led_aging or so, mind this link https://www.mail-archive.com/beagleboard@googlegroups.com/msg15617.html

Install librtlsdr (Angstrom)

Download one of the packaged releases: Look for the eventually greyed .tar.gz text, cut & paste the link of it and fetch it with wget on your Radarcape (in this case it is version 0.5.3, which was current while writing this descrition):

```
wget --no-check-certificate https://github.com/steve-m/librtlsdr/archive/v0.5.3.tar.gz
```

Untar/unpack it. Note that the filename eventually may differ for your download

```
tar xvf v0.5.3.tar.gz
```

As described here, navigate to ' *Building with cmake:* ' and follow these steps. You don't need sudo, as you are already root user on your Radarcape. The first command 'cd' must be adopted to what you have got as folder when unpacking your archive in the latest step.

```
cd librtlsdr-0.5.3
mkdir build
cd build
cmake ../
make
make install
ldconfig
```

Install acarsdec (Angstrom)

acarsdec source package can be found on sourceforge. There is a nice green button with the latest version, click on it and then look for the ' *direct link* ', which you should cut & paste, edit and execute with wget like shown below.

But **before**, with a simple **cd**, go back into your root folder. After uncompressing the archive, change to the folder that became created by tar

```
cd
wget http://downloads.sourceforge.net/project/acarsdec/acarsdec/3.0/acarsdec-3.0.tar.gz
tar xvf acarsdec-3.0.tar.gz
cd acarsdec-3.0
```

Now it comes... the complicated part is that ALSA lib is not necessary for the build here. Remove -D WITH_ALSA from the CFLAGS line. Next, I found that with different compiler options the CPU load is about one third of the original ones.

```
original: CFLAGS= -g -Ofast -ftree-vectorize -funroll-loops -pthread -D WITH_RTL -D
WITH_SNDFILE -D WITH_ALSA
modified: CFLAGS= -g -O3 -flto -funroll-loops -ffast-math -pthread -D WITH_RTL -D WITH_SNDFILE
```

You then should see some output like this:

```
root@beaglebone:~/acarsdec-3.0# make
cc -O3 -flto -ffast-math -funroll-loops -pthread -D WITH_RTL -D WITH_SNDFILE     -c -o
acarsdec.o acarsdec.c
cc -O3 -flto -ffast-math -funroll-loops -pthread -D WITH_RTL -D WITH_SNDFILE     -c -o acars.o
 acars.c
cc -O3 -flto -ffast-math -funroll-loops -pthread -D WITH_RTL -D WITH_SNDFILE     -c -o msk.o
msk.c
cc -O3 -flto -ffast-math -funroll-loops -pthread -D WITH_RTL -D WITH_SNDFILE     -c -o rtl.o
rtl.c
cc -O3 -flto -ffast-math -funroll-loops -pthread -D WITH_RTL -D WITH_SNDFILE     -c -o
output.o output.c
cc -O3 -flto -ffast-math -funroll-loops -pthread -D WITH_RTL -D WITH_SNDFILE     -c -o
soundfile.o soundfile.c
cc -O3 -flto -ffast-math -funroll-loops -pthread -D WITH_RTL -D WITH_SNDFILE     -c -o alsa.o
alsa.c
cc acarsdec.o acars.o msk.o rtl.o output.o soundfile.o alsa.o -o acarsdec -lm -pthread  -
lrtlsdr -lsndfile -lasound
```

In order to execute it for the first, time you need to set LD_LIBRARY_PATH. For later restarts, this one time write /usr/local/lib into /etc/ld.so.conf, which will be active after the next reboot. (see: [1])

```
echo "/usr/local/lib" > /etc/ld.so.conf
ldconfig
export LD_LIBRARY_PATH=/usr/local/lib
```

Execute acarsdec. My stick has +79ppm offset and the gain is fixed to 40dB. With the 3 channels that are active within the decodeable 1MHz segment here, the CPU load is just 35%. That fits to a running Radarcape, if not too busy with Mode-AC. In case of doubts, open another ssh terminal and execute htop command there.

The command needs to be modified for networking according to its original description if you like to send decoded messages to other computers.

```
./acarsdec -o1 -v -p 79 -g 400 -r 0 131.525 131.725 131.825
```

And now you will see its output.

```
Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Tuner gain : 40.200000
Set center freq. to 131850000Hz
Exact sample rate is: 1000000.026491 Hz
Decoding 3 channels
#2 too many parity errors
#3 (L:-21 E:0) 02/01/2000 04:25:07 .OY-KBH SK0610 2 B9 M36A /EKCH.TI2/040EKCHAC5B3
#1 (L:-20 E:0) 02/01/2000 04:25:12 .D-AILA LH03RV R 1L M74A 02467226782VUYY93,EBB*14GK0
#2 parity error(s): 2
#2 crc error
#2 not able to fix errors
```

```
#3 (L:-21 E:0) 02/01/2000 04:25:14 .OY-KBH SK0610 2 _d S88A
#1 (L:-25 E:0) 02/01/2000 04:25:17 .G-BNWM BA0155 2 Q0 S09A
#1 (L:-20 E:0) 02/01/2000 04:25:18 .D-AILA LH03RV R _d S35A
#1 (L:-19 E:0) 02/01/2000 04:25:21 .D-AILA LH03RV R _d S36A
#2 (L:-28 E:0) 02/01/2000 04:25:22 .D-AICC DE03FC X _d S58A
#2 (L: -3 E:0) 02/01/2000 04:25:29 .PH-TFC OR0717 E Q0 S11A
#3 (L:-22 E:0) 02/01/2000 04:25:29 .OY-KBH SK0610 2 _d S89A
```

Radarcape:Faq

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Contents [hide]

Q: Will you provide solutions of all the application ideas?

A: This is far outside my capabilities and it is left to some skilled people in the community. However, I will provide some basic services like 1) or some code libraries that will make Mode-S decoding easier. I most probably will also enforce the idea of peer-to-peer multilateration.

Q: How does the Radarcape perform compared to the Mode-S Beast?

A: The Radarcape got all the improvements implemented that I've learned with the Mode-S Beast, so it's performance generally is a little bit better than the Mode-S Beast, however there are influences that can let the other one look better in some cases. The main focus on the development is versatiltiy in the use cases and not performance.

Q: Will the final user still need to solder difficulty components?

A: The Radarcape will only be available as ready made unit, so no end user completion tasks are necessary.

Q: What if I am not familiar with Linux at all?

A: It much looks as if it as easy as handling of a USB stick, not more, so copying software to it and handling it from there should be easy. Updates easily can be done with a SCP tool, as described in this wiki, as another way the Radarcape while it is connected to the internet can download SW automatically. Otherwise it might be

possible to supply ready preinstalled SD cards that only need to be plugged into the Beaglebone's slot.

Q: Why did you select the Beaglebone and not the Raspberry Board?

A: The Beaglebone is a device that is made for expandability while the R-Pi is a nice gadget but unfortunately does not provide some major features like a power connector or enougth signals on the extension port. Finally it cannot easily become integrated into a case, since the connectors extend to all 4 sides and are not aligned. All these are items which solution would easily eat up the cheaper price.

Q: Will the Radarcape work with the Beaglebone-Black?

A: The Beaglebone-Black unfortunately has pins assigned to eMMC and HDMI which are used to interface to the Radarcape. Due to this, the Radarcape in its current hardware design cannot work with BBB.

Q: Can I operate the Radarcape without GPS?

A: Yes. As long as you don't want to use the absolute GPS timestamp, the Radarcape can operate without GPS.

Q: Can the Radarcape, equal to the Mode-S Beast, decode 2 independant antenna signals?

A: Basically yes, but at the moment there is no firmware support for this provided. 4 channels however are not possible.

Q: Do I need special tools in order to operate the Radarcape?

A: No. Unlike the Mode-S Beast, which required a FPGA programming adapter in its first version, the Radarcape can be managed with software only. All you may need are some basic software tools like WinSCP or Putty.
Just for rewriting a SD card a 4GB SDHC micro SD card writer is useful, which is in many cases already part of modern computers.

Q: Will the Radarcape provide an USB serial interface like the Mode-S Beast?

A: Yes. This is planned.

Q: Will the Radarcape work without an ethernet connection?

A: Yes. This is planned once the USB serial port is implemented. After that, the Radarcape will behave mostly like a Mode-S Beast.

Q: Is it possilbe to power the Radarcape from USB only?

A: No. Power consuption of the whole device is about twice as much as USB can supply, so in any way of operation, an external 5V supply is required.

Q: My Radarcape does not show up as disk drive when connecting the back side USB as my Beaglebone / Beaglebone Black does?

A: The Radarcape comes with the reduced embedded root file system which does not have this feature. Systems with that feature need the Cloud9 images. You can run the Radarcape on top of a Cloud9 installation, but on your own risk.

Q: I do not see Linux help on the Radarcape?

A: The Radarcape comes with the reduced embedded root file system which does not have this feature. Systems with that feature need the Cloud9 images. You can run the Radarcape on top of a Cloud9 installation, but on your own risk.

Q: How can I downgrade to a old Radarcape software version?

A: A software downgrade must be done via SSH on the commandline: *wget <link to radarcaped-YYMMDD.XX.YY> && opkg install -V --force-downgrade <radarcaped-YYMMDD.XX.YY>*

Q: wget does not work when trying to maintain databases

A: Try to send these commands in the command shell:

```
chattr -i /usr/bin/wget.wget
chmod u+rwx /usr/bin/wget.wget
```

Q: What are the default credentials for the web interface?

A: The default user name is *Administrator* and the default password is *radarcape*. We strongly recommend you to change the password after the first login!

Q: What are the default credentials for the SSH login?

A: The default user name is *root* and the default password is empty (just press enter). We strongly recommend you to change the password after the first login!

Radarcape:Links

Radarcape

- Jetvision Shop - You can buy the Radarcape here
- Radarcape Demo - Online demo of the Radarcape

Databases

- Plane Base NG (basestation.sqb)
- ChrisGlobe.co.uk (basestation.sqb)
- PP Routes Yahoo Group (flightroutes.sqb, registration required)

Hardware

- Beaglebone revisions, known issues and possibly necessary modifications
- Trimble Resolution T GPS Module
- Site de F5ANN Information about Mode-S, ADS-B, MLAT and antennas
- Russian Antennas

Software Development

- Port 30003 Format

Miscellaneous

- How to debug shell skripts
- Short Introduction into VI (PDF)

Radarcape:Miscellaneous

Using SOCAT to copy TCP to a file

```
socat -u TCP:localhost:10002 OPEN:radarcape.bin,creat
```

Using SOCAT to access the GPS receiver via network

The Radarcape provides access to the raw data of the GPS device via a network socket. This can be used to provide GPS information to the Linux gpsd daemon.

```
socat pty,link=/tmp/ttyGPS tcp:192.168.1.1000:10685
gpsd /tmp/ttyGPS
```

You may test if gpsd has successfully been connected to your Radarcape using the *xgps* tool.

Cannot Execute a Binary

Dynamic lib missing

On Debian, apt-get upgrade leads to an error with LED aging

Description and replacing script

On Debian: "not a dynamic executable" or any other problems with ld-linux-armhf.so.3

cd /lib
ln -sf arm-linux-gnueabihf/ld-linux.so.3 .

Disable IPv6

https://wiki.debian.org/DebianIPv6#How_to_turn_off_IPv6

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:ApplicationDevelopmentOnRC

Contents [hide]

The tasks described in this manual require expert knowledge of Unix/Linux and embedded systems.

The Radarcape contains a full Linux system based on the Beaglebone hardware, so it can even be used for developing your own code. You can find plenty of information in the web. Here, it shall only be showed how some small "hello world" and other small applications can be compiled and run on the Radarcape.

Packages For C/C++ Development On Beaglebone (Angstrom)

These packages are needed for C or C++ development on the Beaglebone:

```
opkg install update
opkg install gcc
opkg install gcc-symlinks
opkg install g++
opkg install g++-symlinks
opkg install make
opkg install boost
opkg install libc6-dev
opkg install binutils
opkg files libgcc-s-dev
```

It is recommended that these packages be updated in the explicit sequence above rather than all-at-once.

Building Boost For Beaglebone

Mind these pages:

Cross Compilation

Note 1: take care that the user-config.jam is in the search path of b2
Note 2: you eventually need to install these libs for completeness

```
sudo apt-get install python-dev
sudo apt-get install python-bzutils
sudo apt-get install libbz2-dev
```

Invoke b2 (command-syntax)

Invocation (targets and switches)

In case of fails with respect to something with Python

My compile command:<br\>

```
  ./b2 -j 2 toolset=gcc-armhf --libdir=/home/dl4mea/boost_1_56_0/arm-
 linux-gnueabihf/lib install
```

Building Boost For Windows (CodeBlocks, mingw)

- Links: Boost Instructions
- path variable must contain a setting for the mingw gcc compiler binaries, e.g. c:\Programme\CodeBlocks\MinGW\bin
- open cmd window and change dir to c:\boost_1_56_0\tools\build
- execute - *mingw* option is important!!! - **bootstrap.bat mingw**
- b2 install --prefix=c:\boost_1_56_0
- cd c:\boost_1_56_0
- add the c:\boost_1_56_0\bin to your path
- cd C:\boost_1_56_0
  tools\build\b2 -j2 toolset=gcc --without-mpi --without-python --build-type=complete stage
  Less succes with: b2 -j2 --build-dir=c:\boost_1_56_0 toolset=gcc --build-type=complete stage

It needs around 2h for building on a dual core machine.

Radarcape:ARM Crosscompiler

Currently, these are the tools we are using for the development of the Linux applicationin

- Angstrom gcc 4.7.3 toolchain for ARM
- Boost 1.53.0
- Eclipse IDE

ldd missing

http://stackoverflow.com/questions/6150000/cross-compiler-ldd

```
#!/bin/sh
arm-none-linux-gnueabi-readelf -a $1 | grep "Shared library:"
```

Navigation

Main page
Radarcape
Mode-S Beast
Accessories
Weblog

Links

modesbeast.com
Jetvision Shop
Beast Forum

Radarcape:Devicetree

Contents [hide]

The tasks described in this manual require expert knowledge of Unix/Linux and embedded systems.

This chapter describes how the Radarcape is attached into the Linux system of the Beaglebone. It is only informational, as this is already done on all Radarcapes provided to customers. You just need it if you bring up Linux from zero or want to learn how this step is done.

Device Tree Overlays

Step 0: Get the device tree compiler:

```
1. # when using Angstrom
2. opkg install dtc
3.
4. # when using Debian
5. apt-get install device-tree-compiler
```

see also: link

Device Tree Overlay for Beaglebone (white)

Step 1 - Beaglebone (white): Create *~/BB-W-Radarcape.dts*:

```
1. /*
2. * Copyright (C) 2012 Texas Instruments Incorporated -
   http://www.ti.com/
3. *
4. * This program is free software; you can redistribute it and/or
   modify
5. * it under the terms of the GNU General Purpose License Version
   2 as
6. * published by the Free Software Foundation
7. *
8. * Original from: github.com/jadonk/validation-
   scripts/blob/master/test-capemgr/
9. *
10. * Modified by Guenter Koellner for the Radarcape
11. * using: UART5 handshake signals and some GPIOs
12. * with a great thanks to Derek Molloy -
   http://derekmolloy.ie/beaglebone
13. *
14. * As on the Beaglebone Black pins beeing used are occupied by
   eMMC and HDMI,
15. * it will require a different setting (and obviously, a
```

```
    different cape hardware)
16. */
17.
18. /dts-v1/;
19. /plugin/;
20.
21. /{
22.     compatible = "ti,beaglebone";
23.     part-number = "radarcape";
24.     version = "00A0";
25.
26.     fragment@0 {
27.         target = <&am33xx_pinmux>;
28.
29.         __overlay__ {
30.             pinctrl_radarcape: radarcape_Pins {
31.                 pinctrl-single,pins = <
32.
33.                     /* GPIO Values: */
34.                     /* GPIO 0x07 Output         */
35.                     /* GPIO 0x27 Input pullup   */
36.                     /* GPIO 0x37 Input pulldown */
37.                     /* GPIO 0x2f Input no pull  */
38.
39.                     /* Watchdog retrigger output */
40.                     0x078 0x07  /* P9_12 GPIO1_28 #60 Output
    Mode7 */
41.
42.                     /* GPS 1PPS Input */
43.                     0x040 0x37  /* P9_15 GPIO1_16 #48 Input
    Mode7 pullup    PPS1 input */
44.
45.                     /* FPGA Pins */
46.                     0x018 0x37  /* P8_3  GPIO1_6  #38 Input
    Mode7 pullup    CONF_DONE  */
47.                     0x01c 0x07  /* P8_4  GPIO1_7  #39 Output
    Mode7           DCLK       */
48.                     0x008 0x37  /* P8_5  GPIO1_2  #34 Input
    Mode7 pullup    nStatus    */
49.                     0x030 0x07  /* P8_12 GPIO1_12 #44 Output
    Mode7           DATA0      */
50.                     0x014 0x07  /* P8_22 GPIO1_5  #37 Output
    Mode7           nCONF      */
51.
52.                     /* UART5 CTS/RTS as addon to the default
    UART5 device tree settings */
53.                     0x0d8 0x36  /* P8_31 UART5_CTSN   Input
    Mode6 pullup    CTS input  */
54.                     0x0dc 0x06  /* P8_32 UART5_RTSN   Output
    Mode6 no pull   RTS output */
55.                 >;
56.             };
57.         };
58.     };
59.
60.     fragment@1 {
61.         target = <&ocp>;
62.         __overlay__ {
63.             test_helper: helper {
64.                 compatible = "bone-pinmux-helper";
65.                 pinctrl-names = "default";
66.                 pinctrl-0 = <&pinctrl_radarcape>;
67.                 status = "okay";
68.             };
69.         };
```

```
70.     };
71. };
```

Step 1 - Beaglebone-Black: Create *~/BB-B-Radarcape.dts*:

```
 1. /*
 2.  * Copyright (C) 2012 Texas Instruments Incorporated -
    http://www.ti.com/
 3.  *
 4.  * This program is free software; you can redistribute it and/or
     modify
 5.  * it under the terms of the GNU General Purpose License Version
     2 as
 6.  * published by the Free Software Foundation
 7.  *
 8.  * Original from: github.com/jadonk/validation-
    scripts/blob/master/test-capemgr/
 9.  *
10.  * Modified by Guenter Koellner for the Radarcape
11.  * with a great thanks to Derek Molloy -
    http://derekmolloy.ie/beaglebone
12.  *
13.  * Beaglebone Black Version
14.  * requires Radarcape cape board version 3.0 or later
15.  *
16.  * compile using
17.  * dtc -O dtb -o BB-B-Radarcape-00A0.dtbo -b 0 -@ BB-B-
    Radarcape.dts
18.  */
19.
20. /dts-v1/;
21. /plugin/;
22.
23. /{
24.     compatible = "ti,beaglebone-black";
25.     part-number = "radarcape-v3";
26.
27.     version = "00A0";
28.
29.     fragment@0 {
30.         target = <&am33xx_pinmux>;
31.
32.         __overlay__ {
33.             pinctrl_radarcape: radarcape_Pins {
34.                 pinctrl-single,pins = <
35.
36.                     /* GPIO Values: */
37.                     /* GPIO 0x07 Output         */
38.                     /* GPIO 0x27 Input pullup   */
39.                     /* GPIO 0x37 Input pulldown */
40.                     /* GPIO 0x2f Input no pull  */
41.
42.                     /* Watchdog retrigger output */
43.                     0x078 0x07  /* P9_12 GPIO1_28 #60 Output
    Mode7 */
44.
45.                     /* GPS 1PPS Input */
46.                     0x040 0x37  /* P9_15 GPIO1_16 #48 Input
    Mode7 pullup    PPS1 input */
47.
48.                     /* FPGA Pins */
49.                     0x034 0x37  /* P8_11 GPIO1_13 #45 Input
    Mode7 pullup    CONF_DONE  */
```

```
                    0x03c 0x07  /* P8_15 GPIO1_15 #47 Output
    Mode7           DCLK        */
                    0x038 0x37  /* P8_16 GPIO1_14 #46 Input
    Mode7 pullup    nStatus     */
                    0x030 0x07  /* P8_12 GPIO1_12 #44 Output
    Mode7           DATA0       */
                    0x07c 0x07  /* P8_26 GPIO1_29 #61 Output
    Mode7           nCONF       */

                    /* UART5 CTS/RTS as addon to the default
    UART5 device tree settings */
                    0x0d8 0x36  /* P8_31 UART5_CTSN   Input
    Mode6 pullup    CTS input   */
                    0x0dc 0x06  /* P8_32 UART5_RTSN   Output
    Mode6 no pull   RTS output */
                    >;
            };
        };
    };

    fragment@1 {
        target = <&ocp>;
        __overlay__ {
            test_helper: helper {
                compatible = "bone-pinmux-helper";
                pinctrl-names = "default";
                pinctrl-0 = <&pinctrl_radarcape>;
                status = "okay";
            };
        };
    };
};
```

Device Tree Overlay Compilation and Installation

Step 2: Compile (on Beaglebone)

```
# Beaglebone White
dtc -O dtb -o BB-W-Radarcape-00A0.dtbo -b 0 -@ BB-W-Radarcape.dts

# Beaglebone Black
dtc -O dtb -o BB-B-Radarcape-00A0.dtbo -b 0 -@ BB-B-Radarcape.dts
```

Step 3: Copy to */lib/firmware*

```
cp BB-%-Radarcape-00A0.dtbo /lib/firmware
```

Step 4b: Create a shortcut to the slots

```
export SLOTS=$(find /sys/devices -name slots)
```

Step 4b: Verify slots before appyling

```
cat $SLOTS
```

Must result in:

0: 54:PF—
1: 55:PF—
2: 56:PF—
3: 57:PF—

Step 5: Apply the pinmux patches

```
echo BB-W-Radarcape > $SLOTS
```

Step 6: The dmesg command will inform you about what has been done (in case that you do that over serial console, you already will have seen this output)

```
dmesg -s 20
```

Must result in:
[31.862078] bone-capemgr bone_capemgr.8: part_number 'BB-W-Radarcape', version 'N/A'
[31.874269] bone-capemgr bone_capemgr.8: slot #4: generic override
[31.880473] bone-capemgr bone_capemgr.8: bone: Using override eeprom data at slot 4
[31.887937] bone-capemgr bone_capemgr.8: slot #4: 'Override Board Name,00A0,Override Manuf,BB-W-Radarcape'
[31.904636] bone-capemgr bone_capemgr.8: slot #4: Requesting part number/version based 'BB-W-Radarcape-00A0.dtbo
[31.926576] bone-capemgr bone_capemgr.8: slot #4: Requesting firmware 'BB-W-Radarcape-00A0.dtbo' for board-name 'Override Board Name', version '00A0'
[31.952355] bone-capemgr bone_capemgr.8: slot #4: dtbo 'BB-W-Radarcape-00A0.dtbo' loaded; converting to live tree
[31.967771] bone-capemgr bone_capemgr.8: slot #4: #2 overlays
[31.983656] bone-capemgr bone_capemgr.8: slot #4: Applied #2 overlays.

Step 7: Verify if all is correct

```
cat $SLOTS
```

Must result in: 0: 54:PF—
1: 55:PF—
2: 56:PF—
3: 57:PF—
4: ff:P-O-L Override Board Name,00A0,Override Manuf,BB-W-Radarcape

Step 8: Apply the rest of the interfaces from standard device tree files

```
echo BB-UART5 > $SLOTS
echo BB-UART2 > $SLOTS
```

Step 9: verify if all is correct

```
cat $SLOTS
```

Must result in:
0: 54:PF—
1: 55:PF—
2: 56:PF—
3: 57:PF—
4: ff:P-O-L Override Board Name,00A0,Override Manuf,BB-W-Radarcape
5: ff:P-O-L Override Board Name,00A0,Override Manuf,BB-UART5
6: ff:P-O-L Override Board Name,00A0,Override Manuf,BB-UART2

On **Beaglebone-Black** it will show up like:
root@rc301:~# SLOTS=$(find /sys/devices -name slots)
root@rc301:~# cat $SLOTS

```
0: 54:PF---
1: 55:PF---
```

```
 2: 56:PF---
 3: 57:PF---
 4: ff:P-O-L Bone-LT-eMMC-2G,00A0,Texas Instrument,BB-BONE-EMMC-2G
 5: ff:P-O-- Bone-Black-HDMI,00A0,Texas Instrument,BB-BONELT-HDMI
 6: ff:P-O-- Bone-Black-HDMIN,00A0,Texas Instrument,BB-BONELT-HDMIN
 7: ff:P-O-L Override Board Name,00A0,Override Manuf,BB-UART5
 8: ff:P-O-L Override Board Name,00A0,Override Manuf,BB-UART2
 9: ff:P-O-L Override Board Name,00A0,Override Manuf,BB-B-Radarcape
```

Slot #5 and #6, both showing up with HDMI, are not indicated as **L**oaded.

Radarcape:BeagleboneBlack

Contents [hide]

The tasks described in this manual require advanced experience with Unix/Linux.

Scope

Eventually, Radarcape will use a Beaglebone Black Rev. C in order to provide

- more computing power (1GHz instead of 720MHz)
- more memory: 512MB instead of 256MB
- an internal 4GB eMMC card instead of an external SD card

The only disadvantage seen is that the Radarcape looses the native serial console that was accessible through the back side USB.

This page documents changes required for a Radarcape on a Beaglebone Black Rev. C. Please note that the Radarcape requires at least Revision 3.0 cape board, as some pins that are now used by the eMMC, had to be re-routed.

For fallback reasons, an external SD card still can be used. See below.

Changed Pin List

Function	Beaglebone white	Beaglebone Black
CONFD	GPIO1_6	GPIO1_13
DCLK	GPIO1_7	GPIO1_15
NSTAT	GPIO1_2	GPIO1_14
DATA0	GPIO1_12	GPIO1_12
NCONF	GPIO1_5	GPIO1_29

Boot Requirements

In order to make /dev/ttyO5 available, the internal HDMI must be disabled already in kernel command line. For this purpose, on the first (FAT formatted) partition, a file uEnv.txt needs to be present which contains

```
optargs=quiet capemgr.disable_partno=BB-BONELT-HDMI,BB-BONELT-HDMIN
```

Disabling HDMI was the compromise decided that the application SW is equal on BB-white and BB-black.

Finally, the first partition contains

```
root@bbb-setup1:~/mmc1# ls -l
-rwxr-xr-x    1 root     root         99976 Jul  5  2014 MLO
-rwxr-xr-x    1 root     root        379412 Jul  5  2014 u-boot.img
-rwxr-xr-x    1 root     root            68 Jan  1 00:00 uEnv.txt
```

Device Tree

Radarcape:Ideas for Applications

Contents [hide]

Network Streaming or USB output

The most simple and versatile: Replacing the combination of Mode-S Beast and Xport and simply forwarding of all Mode-S frames to USB or a local network, even parallel or multiple pathes. This is what we have available on the existing test units. This means that the AVR- or binary format frames are without preprocessing transmitted on either Ethernet or USB virtual COM port. Main advantage here is that there is no longer a capacity bottleneck as it is with the Xport (due to its maximum 921600bit/sec data rate).

On-The-Fly Data Filtering and Compression

Prefiltering and compression of information that is sent to the main processor. This might be of interest when the unit is operated remotely with slow speed links, even GPRS or UMTS. When beeing used at remote locations or behind slow speed internet connections, it might be useful to limit the information by filters, and/or compress the data so that a minium only is beeing transmitted to the host.

Preconverted Data Formats

The Radarcape due to its processing capacity can act as a low cost Asterix CAT 21 decoder or do about the same as a so called “Port 30003 Server”

Sharing Network Server

Standalone data gathering for sharing networks. You no longer need to run a power consuming PC in order to supply data to your favorized sharing network.

Data Recorder

Data recording or event recording using the local SD card or an external mass storage connected via USB

Standalone Use

On top of the Beaglebone/Radarcape one can install available LCD monitors, and have a small GPS navigator size standalone unit.

Parallel Demodulation of Other Signals

ACARS and VDL can be demodulated/decoded with the Beaglebone's 100kSample ADC.\\It might be an idea to connect a SDR receiver like the Funcube dongle to the USB port and use this with some SDR software like GNURadio in order to receive air traffic, ACARS, VDL and similar.

Peer-to-peer Multilateration

The Radarcape will be designed to provide timestamps as accurate as possilble and synchronized with GPS. At least three Radarcapes at decent locations will exchange their information and with simple triangulation they can localize all non-ADS-B traffic in addition to the already known aircraft. Planeplotter already does something in this kind, so maybe the Radarcape can become included into this network or even act as a standalone data processor inside it.

Distribution
Günter Köllner Embedded Development GmbH
85256 Vierkirchen
Germany

jetvision.de