RECS®|Box User Manual

christmann informationstechnik + medien GmbH & Co. KG

June 15, 2020

Contents

Chapter 1

Introduction

Thank you for purchasing a Christmann RECS®|Box System.

1.1 Chassis

The RECS®|Box system provides various types of devices, described below in more details:

- RECS®|Box Compute Unit (RCU)
  - long version with up to 18 baseboards, external power supply (RPU) needed. Codename: Arneb
  - short version with up to 6 baseboards, 2 integrated PSUs. Codename: Antares
- RECS®|Box Power Unit (RPU)
  - 10 integrated power supply units (PSU). Codename: Alasco

1.1.1 Compute Unit

The Compute Units contain the computing elements of RECS®|Box systems. There are actually two different rack chassis available:

Arneb

The Arneb is the scale-out RECS®|Box Compute Unit, hosting up to 18 baseboards (= 18 COM Express modules or 72 Apalis modules) on 1 RU. It has a side-cooling approach, where cool air flows from the left side to the right side over all installed modules and gets blown out of the chassis at the right side. To make the Arneb best scalable, it is powered by the external Power Supply Unit (PSU) Alasco.

Antares

Figure 1.1: RECS|Box Compute Unit - Antares

The Antares is a standard 1 RU rack chassis with standard air flow (front to rear) and integrated PSUs. It can host up to 6 baseboards (= 6 COM Express modules or 24 Apalis

1.1.2 Power Supply Alasco

The PSU Alasco can be used to power several Arneb Compute Units. It converts 230V AC to 12V DC and can provide up to 3 kW total power. Its integrated management controller allows a configurable redundancy of the 10 internal power supply units, per default it's running in a N+1 redundancy mode. The amount of running power supply units are dynamically managed depending on the number of running compute modules in the connected Arneb's.

1.2 Compute Modules

1.2.1 COM Express Modules (x86)

The following **Intel** CPU based compute modules are available:

CPU Type	RAM	Onboard Network
Core i7 4th generation 4 x 2.4 / 3.4 GHz, 47 Watt TDP	max. 16 GB DDR3L 1600MT/s	1 GbE[1]
Core i5 4th generation 2 x 2.9 Ghz, 37 Watt TDP	max. 16 GB DDR3L 1600MT/s	1 GbE[1]
Core i3 4th generation 2 x 2.4 Ghz, 37 Watt TDP	max. 16 GB DDR3L 1600MT/s	1 GbE[1]
Celeron 2 x 2.2 Ghz, 37 Watt TDP	max. 16 GB DDR3L 1600MT/s	1 GbE[1]
Atom 4 x 1.91 Ghz, 10 Watt TDP	max. 8 GB DDR3L 1333 MT/s	1 GbE[1]

The following **AMD** CPU based compute modules are available:

CPU Type	RAM	Onboard Network
Embedded R max. 2.3 Ghz, 35 Watt TDP	max. 16 GB DDR3 1600 MT/s	1 GbE[1]
Embedded G max. 4 x 2.4 Ghz, 25 Watt TDP	max. 8 GB DDR3L 1600 MT/s onboard	1 GbE[1]

[1] Additionally a 1 GbE or 10 GbE compute network is available on the baseboard as described innetboard installation

1.2.2 Apalis Modules (ARM)

The following Modules built according to the **Apalis-standard** are available:

Manufacturer	CPU Type	CPU Name	GPU	RAM	Onboard Network
Toradex	ARM Cortex-A9	Nvidia Tegra 3 4 x 1.3 Ghz, 6 Watt TDP	520 MHz	max. 2 GB DDR3 1600 MT/s	1 GbE
Christmann	ARM Cortex-A15	Samsung Exynos 5250 2 x 1.7 Ghz, 20 Watt TDP	ARM Mali-T604MP4; 533 MHz	4 GB DDR3	2 x 1 GbE

1.3 Safety Instructions

CAUTION: For your safety

Please carefully read through the safety notes in the section below before commissioning your server!

1.3.1 Installation

- The power supplies in your server system may produce high voltages and energy hazards, which can cause bodily harm. Please do not open the server covers and access the inside of the server unless you have been told to do so. Do not open the cover of the power supplies at any time!
- Place the device on a hard, level surface or mount it within a server rack.
- You cannot place the server in an area with a weak floor.Under the weight of the server,the floor might give way.
- The device is designed to operate indoor only.
- A temperature difference of 15 degrees between the room temperature and server is enough for moisture to form. Please acclimatize the unit to avoid a short-circuit at least two hours.
- Protect the device of moisture, dust, oily or any other kind of liquids and steam.
- Keep the device away from radiators and other heat sources like direct sun ray.
- Keep the device away from devices that may cause electromagnetic interferences.Doing so may damage the server as well as cause fire or an electric shock.
- Leave at least 10 cm of clearance on all vented sides of the case to permit proper ventilation.
- Control it the placement of each component in the rack,before you start to install the rails.
- First you should to install the heaviest server components on the bottom of the rack,then you can work up.
- It is recommended to use a regulating uninterruptible power supply (UPS), this way you will protect the server from voltage spikes, power surges and to keep your system operating in case of a power failure.

1.3.2 Operation

- Position any cables carefully, route cables so that they cannot be stepped on or tripped over. Be sure that nothing rests on any cables.
- Never operate the device in a wet environment, keep the device and any cables dry.
- Do not immerse the device or any cable into water or other liquids.
- Do not cover any vent holes, do not stick anything into the vent holes.
- To help prevent electric shock, plug the equipment and peripheral power cables into properly grounded electrical outlets.
- Do not touch any hot power supply modules or other hot components of the server.
- Always keep all panels, components of the servers and the rack's front door closed when not ensuring proper cooling.

1.3.3 Rack Precautions

- Deploy the anti-tilt bar or legs on the equipment rack before beginning an installation.
- Ensure that the leveling jacks on the bottom of the rack are fully extended to the floor with the full weight of the rack resting on them.
- The stabilizers should be attached to the rack before you start with the installation. If you install multiple racks, first of all you should couple together the racks.
- Before you work on any components of a rack, always make sure that the rack is stable.
- You should not extend more then one components at the same time, because the rack may become unstable.
- Mechanical loading: Mounting of the equipment in the rack should be such that a hazardous condition is not created due to uneven mechanical loading.
- Reduced airflow: Installation of the equipment in a rack should be such that the amount of airflow required for safe operation of the equipment is not compromised.

Chapter 2

Server installation

The following chapter will give you an overview on how to install an maintain your RECS®|Box.
When working on the RECS®|Box always wear **ESD protection** equipment.

2.1 Rack mounting

The Antares version of the RECS®|Box comes with premountet rack-rails. These can be used to install the Antares into any 19„ rack.

The following chapter is copied from the original user manual of the rails:

1. Slide chassis member (smallest) is removable. Separate slide members by lifting left hand side lever up, and right hand side lever down. Keep slide chassis members with original outer members.
2. Slide has front and rear adjustable EIA brackets. Attach front brackets to outer slide member using #8-32 machine screws by aligning access slots on brackets to outer member holes. Front brackets can be mounted flush or 1/2" [12.7 mm] back from front of slide based on user needs.
3. Establish distance from front cabinet rail to rear cabinet rail and attach rear brackets to slide. **Do not** fully tighten rear bracket mounting screws until final adjustment is made.
4. Insert removable locator pin into center hole of brackets.
5. Slide will mount to various cabinet rails at various locations on side of chassis. See illustrations at right.
6. Place slide into desired position and fully extend against EIA rails. Locator pins will support slide and bar nut while attaching screws.
7. Attach slide to EIA rail with #10-32 screws and fasten with bar nuts. Once screws are secure, remove locator pins. Do not fully tighten screws until final adjustment is made.
8. Mount slide chassis members (smallest) to the chassis.
9. With cabinet members in the open or closed position, bring ball retainer to full forward position. Install chassis by engaging the slide members and close completely. **Locks will engage on initial insertion and must be released to fully close.** Check slide alignment by opening and closing the chassis. Any sign of binding indicates lateral stress or misalignment.
10. Adjust slide position until movement is smooth. Tighten all screws and complete installation.

NOTE: To remove chassis, lift the lever on the left hand slide up and the lever on the right hand slide down.

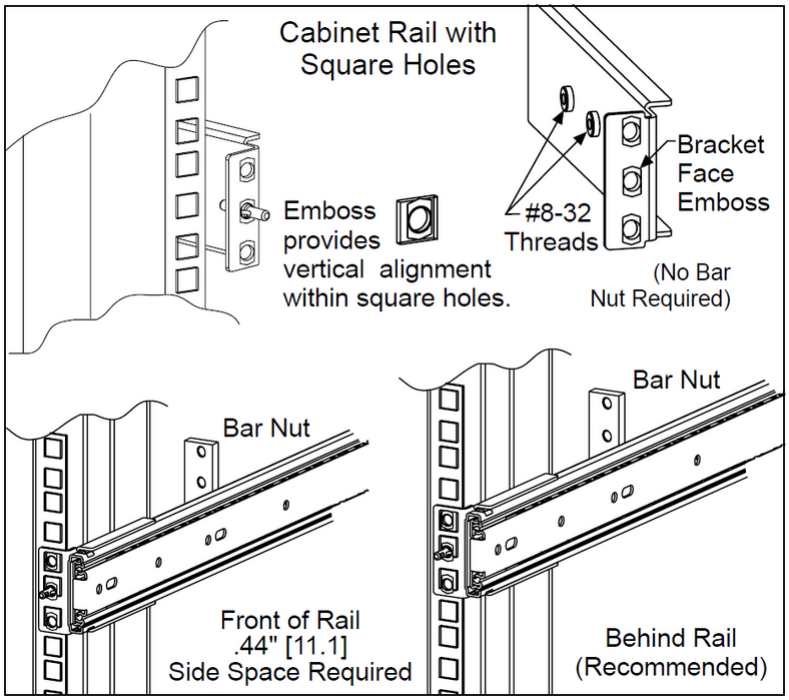

Figure 2.1: Mounting overview

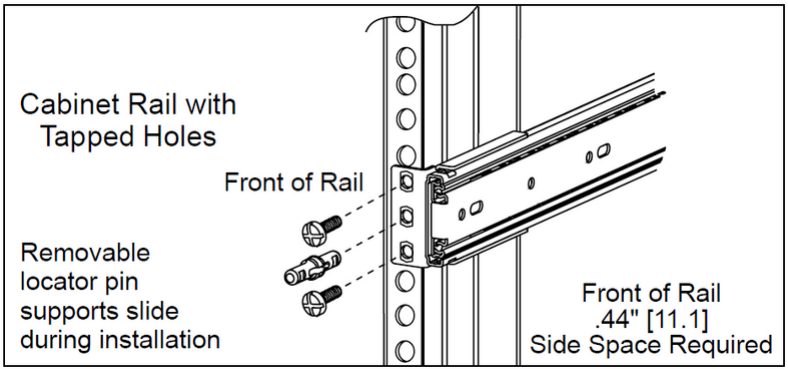

Figure 2.2: Detailed view of infront of the rail mount

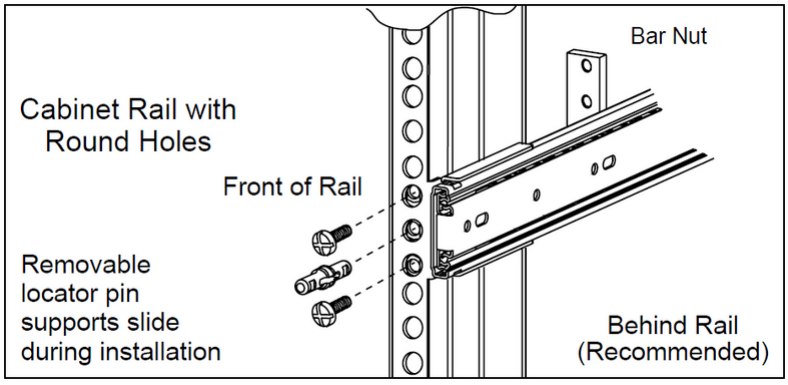

Figure 2.3: Detailed view of behind the rail mount

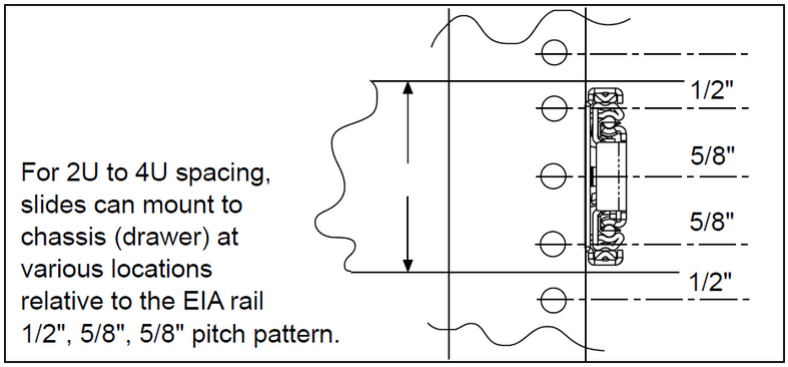

Figure 2.4: Slide frontview

2.2 Opening/closing the case

To open the case first make sure, that **no heavy load** is on the modules. Because this can lead to overheating when the top cover is removed.
Always make sure that no tools or any other stuff can fall into the case.

2.2.1 Antares

When opening the case watch out **not to touch the 230V** cables or connector block in the back that lead to the power supplies. This can result in a danagerous electric shock.

Opening the case

First unscrew the M3 knurled screw (A) at the top back of the Antares. It may be necessary to pull the Antares out of the rack further than the rails allow. To to this pull the Antares out until the rails lock in place, and then use the levers to the left and right (as described in Rack mounting) to slide the Antares over the locking point. **But don't pull to far**, as the Antares could slip out of the rails!
Then slide the cover back (B) and carefully lift the cover so that the hooks don't get stuck and bend.

Closing the case

Place the cover onto the Antares so that the hooks fit into the holes in the chassis.
Then slide the cover to the front (B) while pushing the hooks downwards into the holes if they are stuck.
At last tighten the M3 knurled screw (A) at the top back of the Antares.

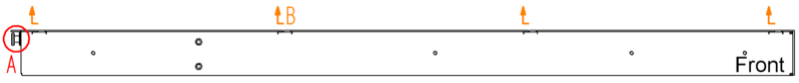

Figure 2.5:

2.2.2 Arneb

Opening the case

Unscrew the 14 (six on top, eight on the sides; marked red below) M3 screws using a T8 Torx screwdriver and lift the cover.

Closing the case

Place the cover on top of the Arneb and check if the screwholes of the cover align with the ones in the chassis. If this is not the case the cover is placed in the wrong direction. Tighten all 14 screws (marked red below) using a T8 Torx screwdriver.

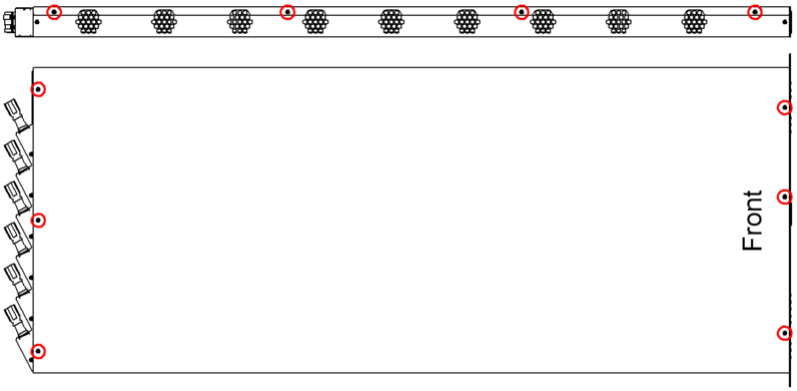

Figure 2.6: Cover screws (red circle); the four screws on the right side are not shown

2.2.3 Alasco

Do not open the case of the Alasco.
If you encounter a problem with the Alasco please contact your vendor.

2.3 Baseboard replacement

When removing a Baseboard always **work very carefully** and **do not bend** the baseboards **nor use greater force** to get them into place. Pay attention to the newtork cables that are located right under the baseboards. They or the baseboards can be damaged if not handeled with care. Check if they are still properly fixed to the case and don't stack over each other more than 2 mm.
It is recommended to turn the systempower off when changing a baseboard.

2.3.1 COM Express Baseboard

Remove the two M2,5 screws (B) with a T8 Torx screwdriver from the mSATA drive, and the four M3 screws (A) with a 5,5mm hexagonal nut.
Then lift the baseboard a bit at the end where the fan connectores are. And while moving it a bit to the left and right gently pull the baseboard out of the connector on the backplane.
Before plugging in a new baseboard make sure that there is a robust and isolating tape (like Kapton tape) taped onto the PCIe pins, that come out of the bottom of the board.
To plug the baseboard in procede the opposite way as removing it.

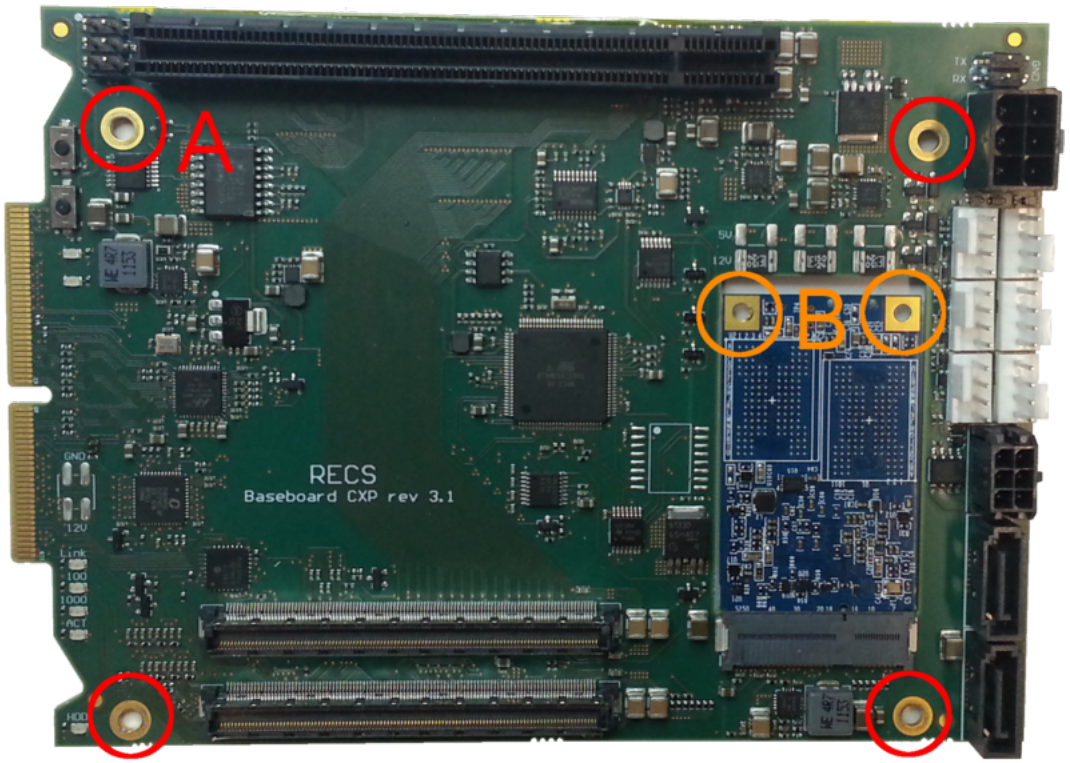

Figure 2.7:

2.3.2 ARM Baseboard

The baseboard can be replaced when the Apalis modules are plugged in. And it is not recommended to remove the plasic module-holders. If an Apalis-holder is broken try to fix it with superglue **without removing** it from the baseboard.
Remove the four M3 screws (marked red) with a T10 Torx screwdriver. It can be handy to use a T9 Torx screwdriver for the screw in the top right on the picture below, when the Apalis modules are still in place.
Then lift the baseboard a bit at the end where the fan connectores are. And while moving it a bit to the left and right gently pull the baseboard out of the connector on the backplane.
To plug the baseboard in proceed the opposite way as removing it.

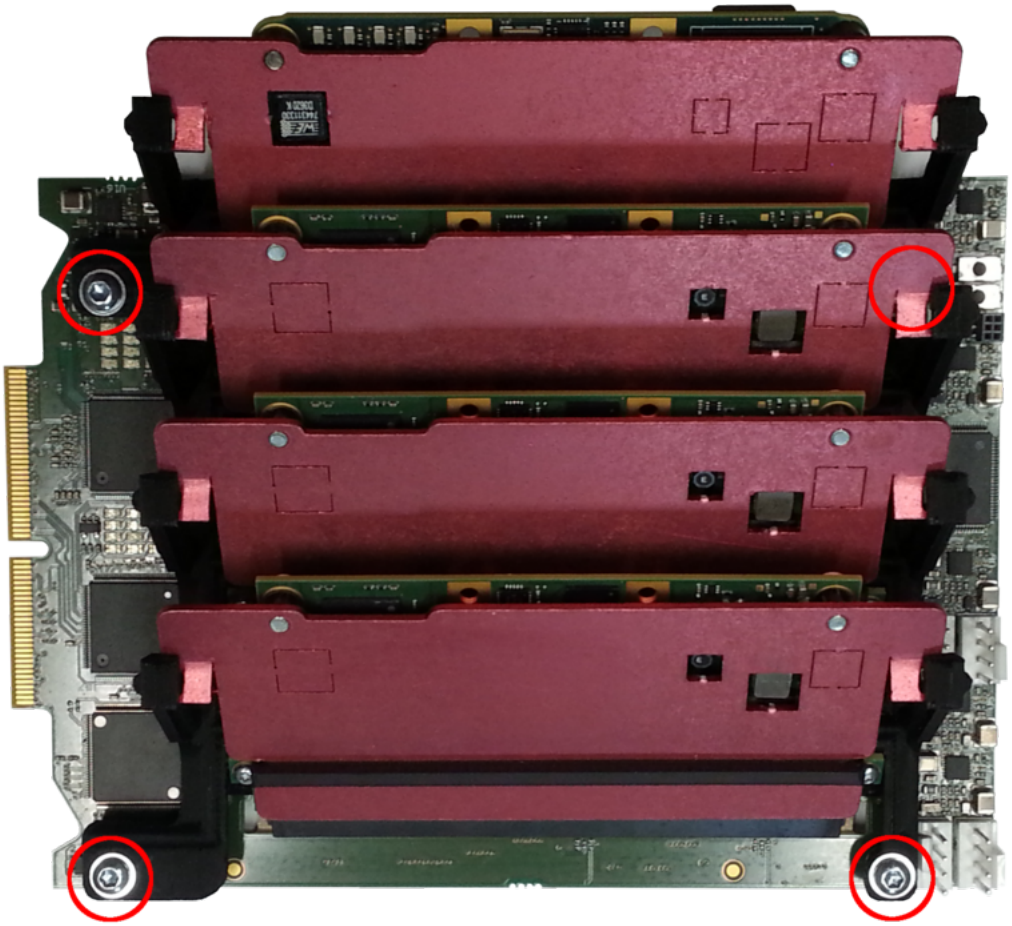

Figure 2.8:

2.4 Compute module installation

When changing a module shut this module down and turn the power for the module off in the managemant GUI. The rest of the system can stay powered.

2.4.1 COM Express module

Before installing a Com Express module check if the RAM is installed, and that the heatspreader is correctly mounted, and depending on the module if all thermal pads are in place. Also the orientation of the heatsink fins has to follow the airflow trough the case. If it is an Antares the cooling fins have to point from the front to the back. In an Arneb case they have to point from side to side.
The screwholes where the module lays on the spacer bolts (8 mm in height) have to be free. Or smaller spacer bolts (5 mm high) have to be installed.

1. When the module is ready to be plugged in place the two holes next to the COM Express connector over the corresponding spacer bolt. Then slide the outer CXP connector into the outer connector on the baseboard. Do this while tilting the module a bit, so that only the outer connector slides in
2. Then push the module down where the connector is by pressing on the heatsink. While holding the pressure push the module down with a luffing movement on the end where no CXP connector is. The module should now slide into the second CXP connector and be in place
3. check if the module is mounted correctly and is not loose

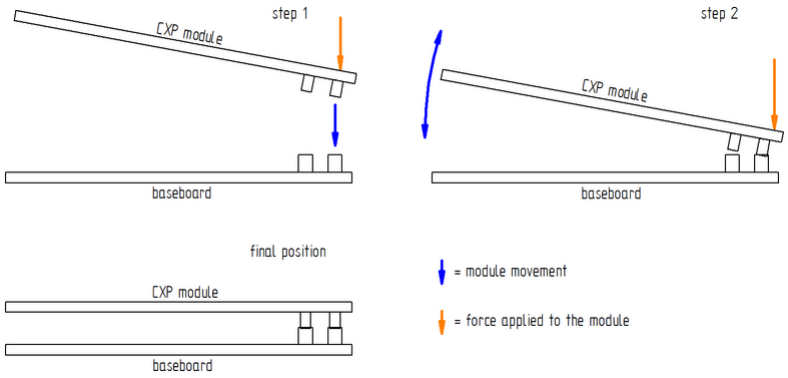

Figure 2.9:

When starting the module for the first time after installation, it is necessary to set the time again, as the modules don't have an own battery and are provided with electricity for the clock by the battery on the frontpanel.

2.4.2 Apalis module

Before installing an Apalis module check if the thermal pads are installed, and that the heatsink doesn't touch any electronic components. Especially where the breakouts in the heatsink are. Also check if the screws are tightend and the clamp is locked in place.
When bending the plasic holder be carefull and do not bend them too far as they could break off. If this happens try to repair them using superglue without detaching the plasic holder from the baseboard, as this can lead to a broken baseboard.

1. insert the module into the Apalis connector on the baseboard
2. push the module down while bending the plastic holder a bit to the outside so that the heatspreader can slide past the holder
3. release the pastic holder back to the original position and let the module snap up into the plastic holder
4. check if the module is locked in place

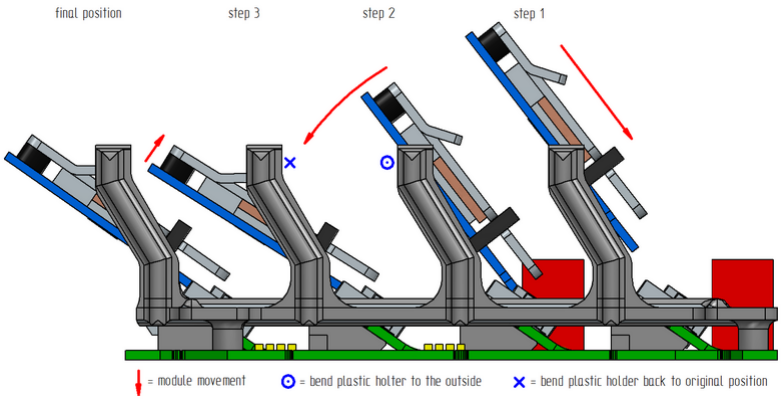

Figure 2.10:

To remove a module proceed with the opposite procedure.

2.5 Netboard installation

All netboards (1GbE or 10GbE) have the same connector (as seen below) underneath and are all mounted on the three connectors on the backplane. Each netboard provides the compute network to the two baseboards

that are plugged in right next to them. It is also possible to mix different netboards, as they are independent of each other.

1. To install a netboard simply place it over the connector with the correct orientation.
2. Align the connectors with the help of the smaller one.
3. Carefully and mostly parallel push the netboard down into the connector while wobbleing the board a bit.

Figure 2.11:

Figure 2.12:

2.6 Extension card installation

It is possible to install an extension card like a graphics card on the PCIe x16 slot of an **v3.1 CXP** baseboard. A special riser card is needed to connect the card. The I/O shield of the card has to be removed. The 6-pin power connector on the baseboard can deliver up to 300W, but a special cable may be required to connect the 6-pin plug to the card, as the cards have different connectors (6pin, 8pin or a mix).
Please contact your vendor about this cable.

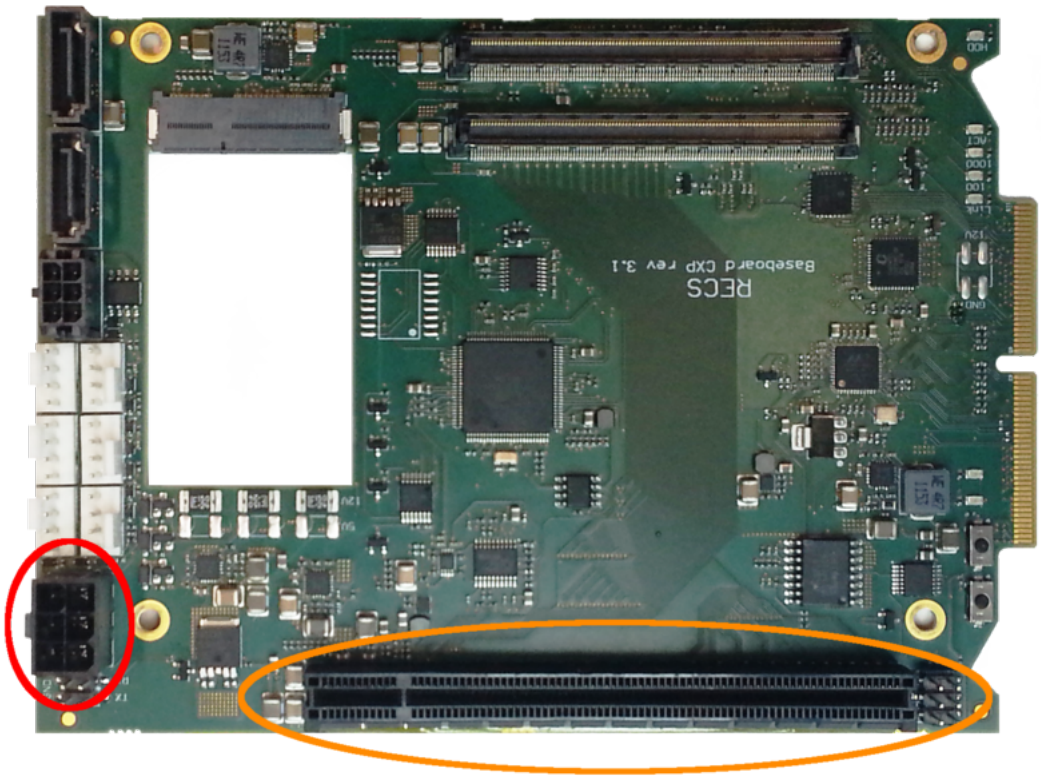

Figure 2.13:

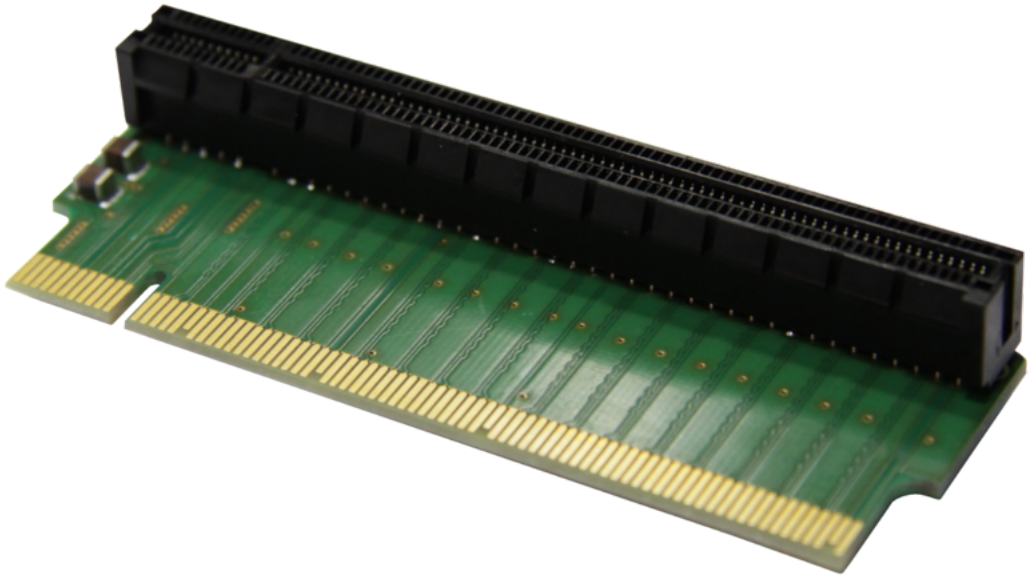

Figure 2.14:

When installing an extension card make sure that it will be cooled propperly.

2.6.1 Antares

One full sized dual slot card can be built into the back of the Antares. The card will then be connected to node 6. In the Antares, there are no mountingpoints in the chassis, so these have to be glued on to fix the card in position. Due to the fact that the airflow in the Antares is beeing changed, wenn installing a extension card, the fans have to be rearranged and additional air baffles have to be installed. The exact location of the fans and the air baggles depends on the size of the card and combination of CPU modules built into the Antares.

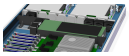

Figure 2.15:

2.6.2 Arneb

The Arneb can hold single slot cards. If the card is too long no node or netboard can be placed where the card is. There are grid plates available, that can be used to build a holding structure for the card onto. It is possible to install a dual slot card, if the heatsing is small enough not to touch the backplane.
An example of an extension card that requires the space of the netboard installed in an Arneb can be seen below.

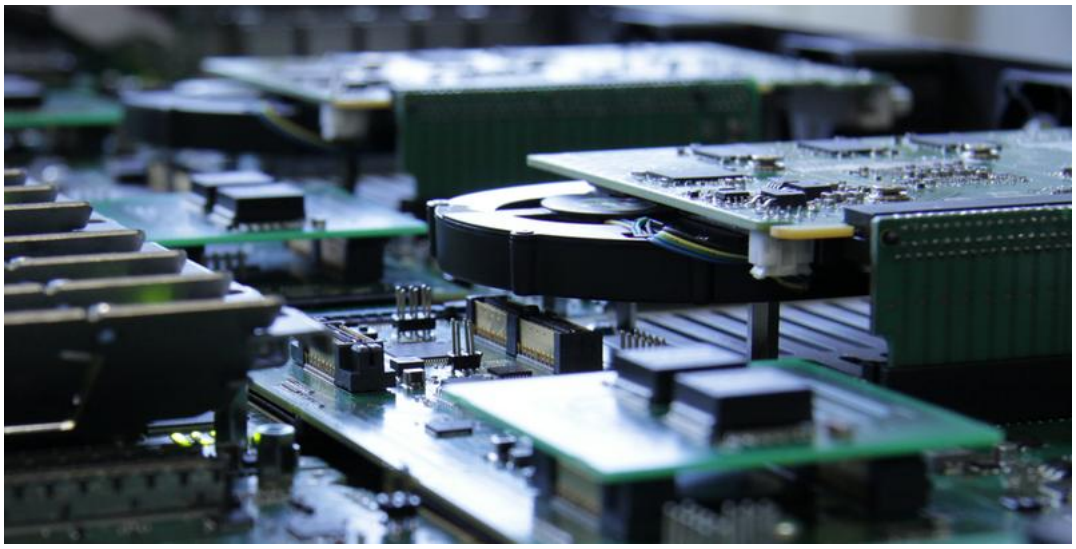

Figure 2.16:

2.7 Fan replacement

A fan that has damaged blades schould be **replaced immediately**, because the bearing will break very fast due to the imbalance and resulting vibrations.
When changing a fan on a working system be very carefull not to touch the blades or get any cables into the running fans. Also watch out that the system is not overheating during the changing process.
Watch out that the **plugs are correctly placed on the baseboard**. As a short circuit or misalignment of the pins will damage the baseboard.

Open the case according to the Opening/closing the case chapter of this wiki. First disconnect the fan from its power source and make sure the cable is not stuck anywhere.

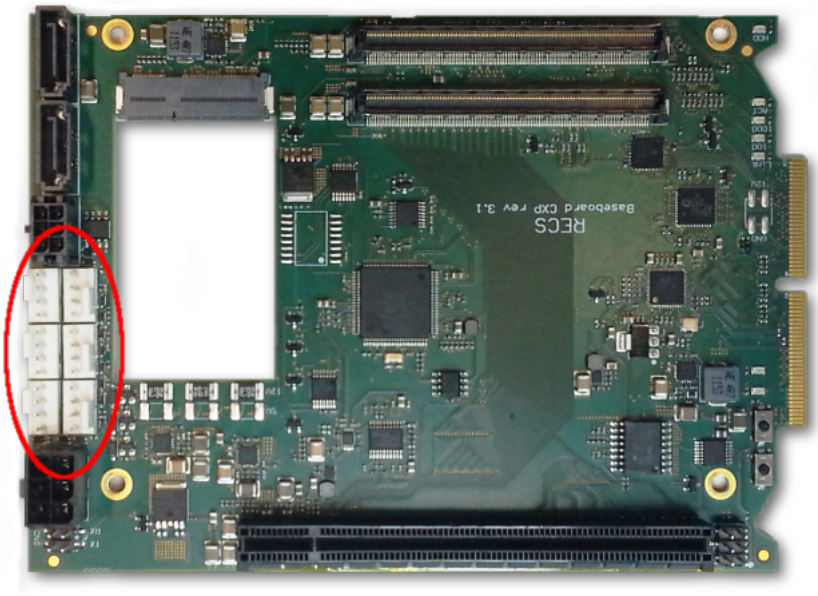

Figure 2.17:

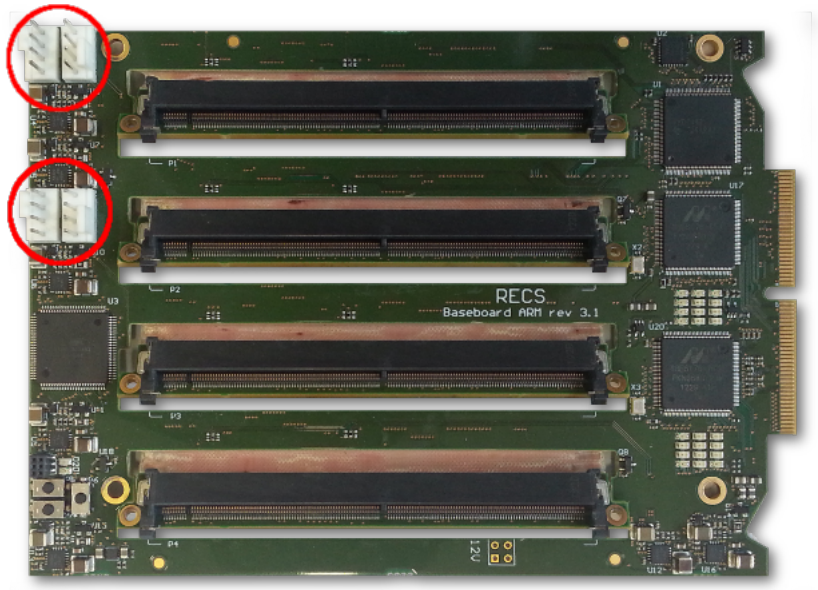

Figure 2.18:

2.7.1 Antares

1. To remove the fan simply drag it out of the mounting plate diagonally.
2. Unscrew the anti-vibration-rubber from the broken fan with a PH2 screwdriver, and screw them diagonally onto the new fan so that the cable and airflow point in the right direction when installed.
3. Push the fan back into the mounting plate.
4. Arrange the cable under the fans and connect it.

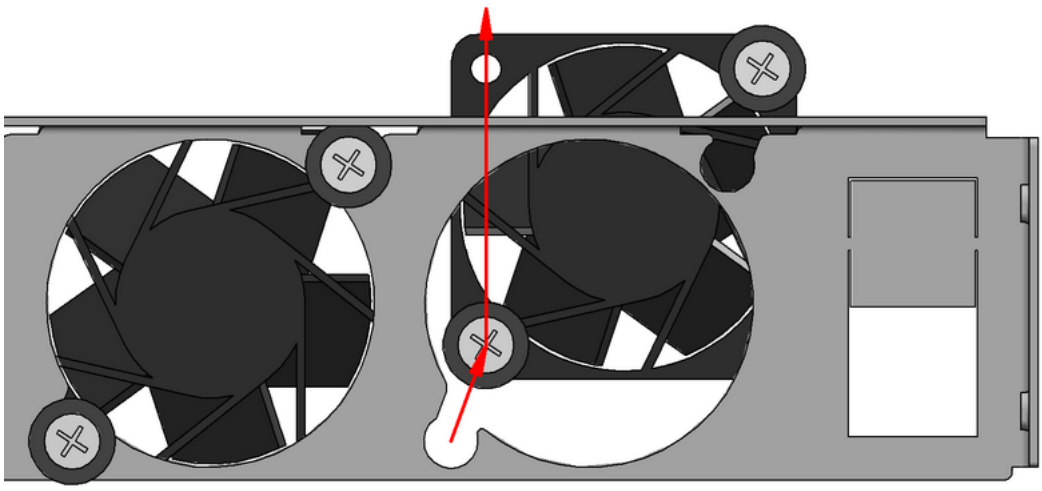

Figure 2.19:

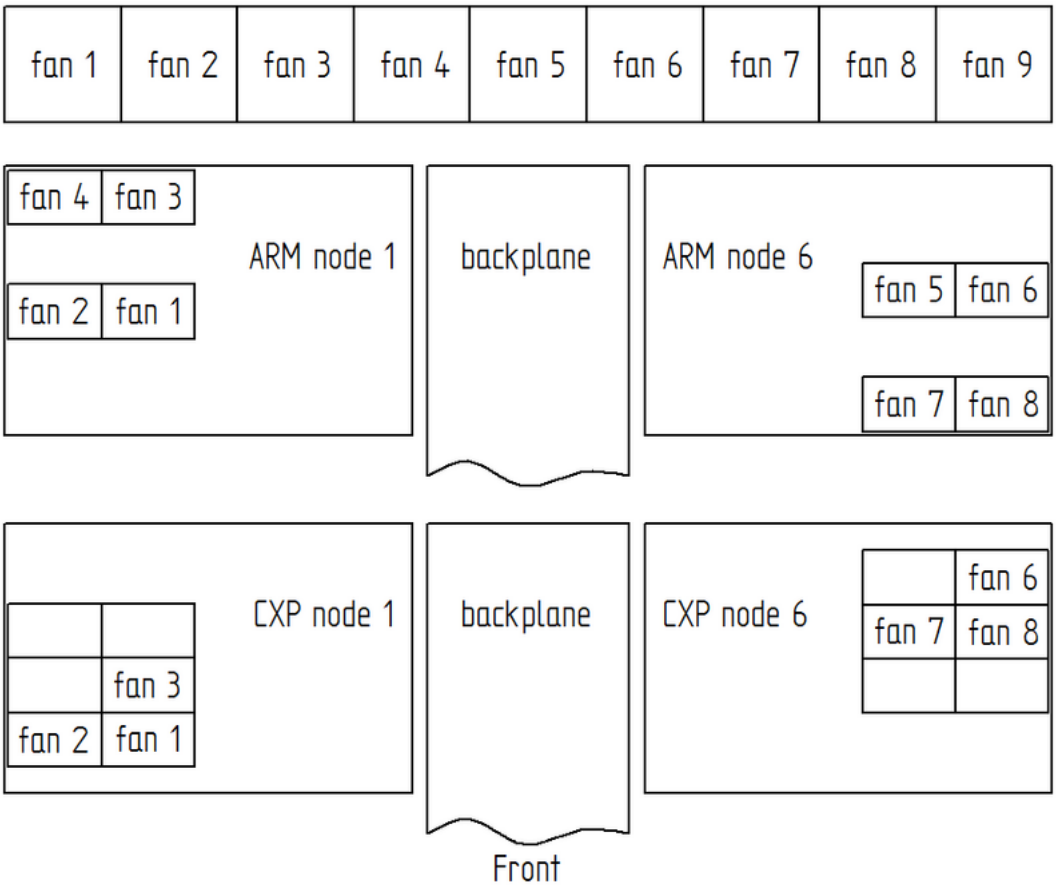

Figure 2.20:

2.7.2 Arneb

1. Unscrew the two screws that hold the fan to the chassis with a PH2 screwdriver.
2. Replace the fan with a new one considering the airflow direction.
3. Tighten the two screws.

4. Plug the cable onto the baseboard.

Chapter 3

Physical interface

3.1 RCU interface

3.1.1 Front panel connectors

The following items are located at the front of a RECS®|Box RCU:

- Two line display
- A five button D-pad
- Four internally switched LAN ports that connect to the 1GbE management network of the nodes
- One LAN port that connects to the internal management controller (marked with „Management")
- Four USB ports that can be switched to every module via KVM
- One HDMI port that can be switched to every module via KVM
- Six (Antares) or 18 (Arneb) LAN ports that connect to the compute network of the labelled node (network speed depends on the installed netboards Netboard installation

Figure 3.1:

Figure 3.2:

3.1.2 Control panel

The display on the front panel of the RCU shows the status of the unit on multiple screens that automatically change every couple of seconds and are updated live.

Antares	Arneb	Description
		Shows the IP address of the RCU.
		Displays the current health status of the RCU an
		In the first row the number of currently turned o
		Shows the node on/off status for each baseboard

The buttons next to the display will be used for a local management menu later on. Currently they are not used except for the „down" button: Press this key for a few seconds to safely shut down the management controller in the RCU before disconnecting power to the unit.

3.1.3 Power connectors

Antares

The Antares has one male C14 jack on the back that can be connected to an 230V power outlet using a standard C13 cable. Both built in power supplies are connected to this jack.
The exhaust air openings in the back must be kept free. Keep that in mind when installing cables behind the Antares.

Arneb

The Arneb has six 12V power connectors on the back. These are connected to an Alasco power supply. The upper male plug is the (+) pin and the bottom female plug is the (-) pin. All six connectors are joined together to two copper rails. One (+) and one (-). The plugs are coded and can only be plugged in in the right way. How many of these connections to the Alasco are needed depends on the hardware configuration and the total power consumption of the Arneb.

3.2 RPU interface

3.2.1 Front panel connectors

The Alasco power supply unit has the following items in its frontpanel:

- Two line display
- One LAN port that connects to the internal Colibri with the management software of the power supply

3.2.2 LCD display

Display	Description
	Shows the IP address of the RPU
	Displays the current health status of the RPU and the number of power supplies installed.
	Shows the numer of power supplies that are currenty on, and the power provided by the RPU. A percentage bar (10% steps) of the load is shown in the lower line.
	Shows the on/off status for each power supply (0=off 1= on). A dash ("-") indicates power supplies that are not populated or where not found. "L" and "R" mark the orientation of the display and are shown as looking onto the front panel.

3.2.3 Power connectors

The Alasco has one male C19 jack on the back that can be connected to an 230V power outlet using a C20 cable. Connect this cable to an 16A fuse, as the Alasco can require up to 3600W.
There are nine 12V plugs in the back, that can be connected to one or more Arnebs via the coded connectors. The upper male plug is the (-) pin and the bottom female plug is the (+) pin.

3.3 Sensors

Each Baseboard of the RECS®|Box system includes five temperature sensors that are regularly polled by the management software. This data is available in raw format via all software interfaces provided by the TOR Master. Additionally, the TOR Master calculates some derived temperatures: Inlet- and outlet temperature (Arneb only) as well as the highest temperatures on baseboard, RCU and cluster level.

3.3.1 Locations

The following pictures indicate the location of the various temperature sensors on the baseboards along with their indices in the management data. The golden finger contacts on the right side of the picture always point towards the backplane.

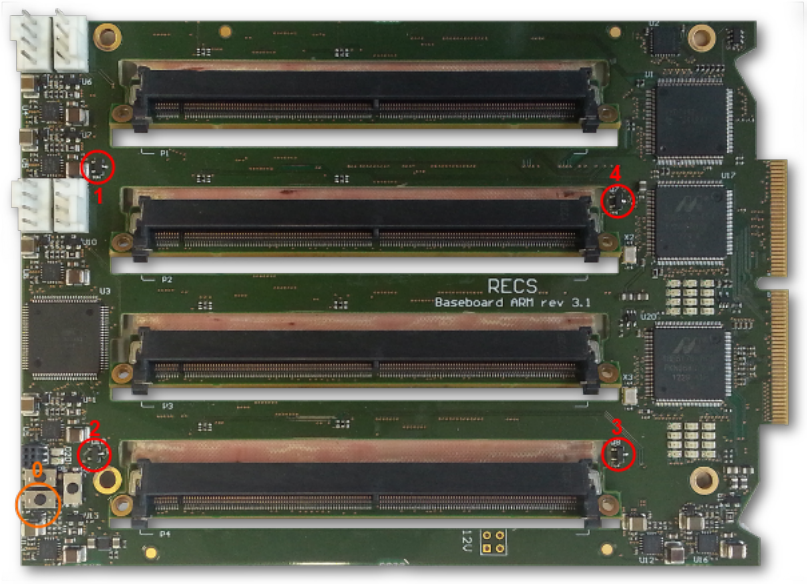

Figure 3.3: Temperature sensors on ARM baseboard. Orange: Bottom side

Please note that temperature sensor 1 on the ARM baseboard (near the four fan connectors) is somehow thermally connected to the voltage regulators (located between the four fan connectors). This results in high temperature of this sensor if the modules 1 and 2 are powered on and consuming a lot of electrical power.

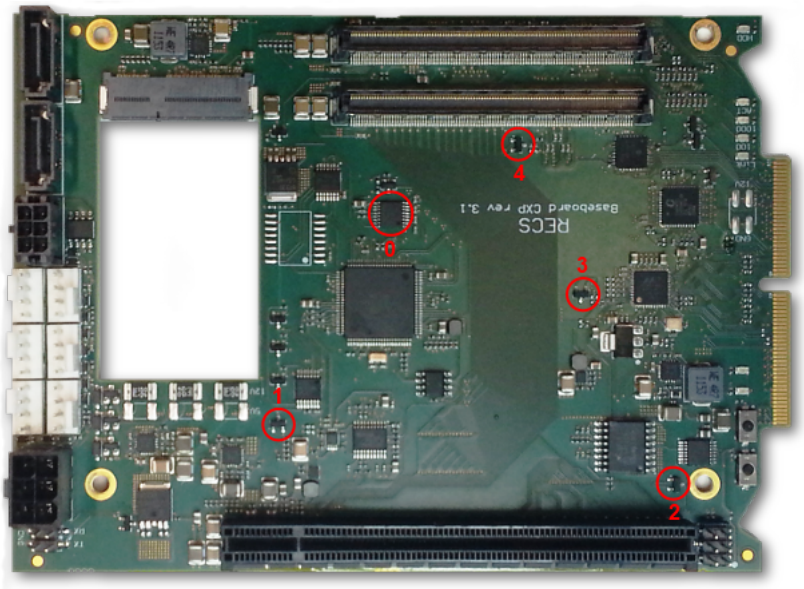

Figure 3.4: Temperature sensors on COM Express baseboard

3.3.2 Assignment

Depending on baseboard type and location, the Arneb RCU calculates inlet- and outlet temperature for each baseboards by averaging different temperature sensor readings. The following table shows which sensor(s) are used, where „left" and „right" are seen as looking onto the LCD of the unit.

Temperature	Type	Left	Right
Inlet	CXP	1	3
	APLS	1, 2	3, 4
Outlet	CXP	3	1
	APLS	3, 4	1, 2

Chapter 4

Software interface

There are several software interfaces available to monitor the status of the RECS®|Box system. These are the Management WebGUI, a REST API providing XML based monitoring and management functionality and a native NRPE based Nagios interface.

4.1 Management WebGUI

The Management WebGUI is established on every RECS®|Box unit. Accessible by any known browser on the assigned IP address and the default port 80. The following views are dependent on the device and assembly.

In general these symbols have the following meaning on every page:

	Everything is OK. Also indicated by a green line in a graph.
	Warnung. Something is wrong, but the system is still fully functional. The system has to be checked so the problem doesn't get worse. Indicated by a yellow line in a graph.
	Critical Error. The system must be checked immediately and maybe has to be shut down to prevent hardware damage. indicated by a red line in a graph.

Figure 1 shows the first call of the Management WebGUI. It is organized into three columns. The first is on the left-hand side and contains the following:

Overview: General overview of all managed RCUs, RPUs, installed nodes and health status
Management: Selection of every managed RCU and RPU in the rack with a sensor view button for the Arneb
Global settings: IP filter and firmware update
Log: Logs from the management software about system health and java messages. The logs can be downloaded as a zipfile

The second colum contains the buttons and sliders to manipulate the system. While the third colum is mostly for history information like power usage and temperature graphs.

4.1.1 Overview

All units that are installed in the rack and that are managed by the software are summarized on this page. The total power usage is summed up over all managed units.

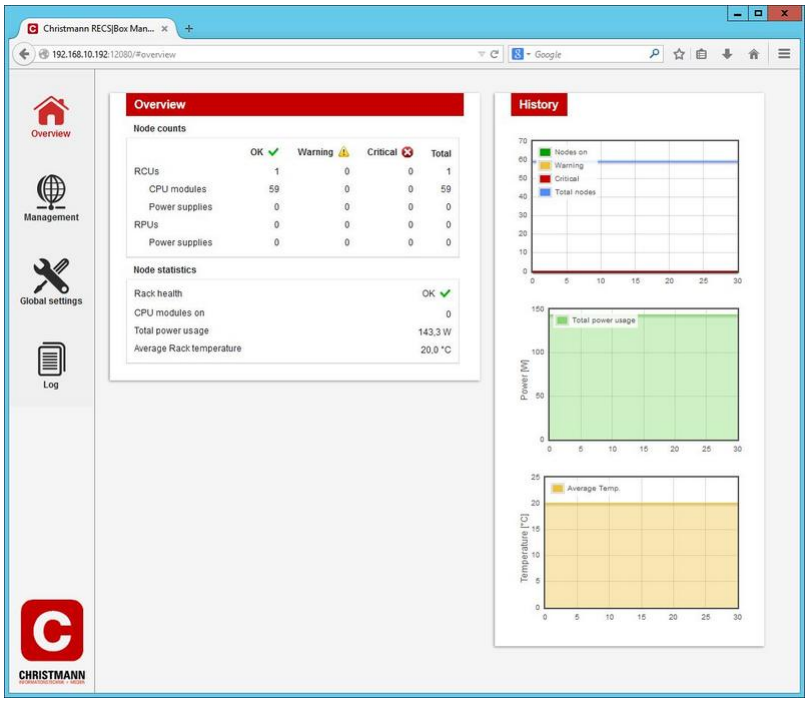

Figure 4.1:

4.1.2 Management

An overview of the selected unit can be seen in this tab. The fans can be regulated by dragging the slider to the desired percentage. And multiple nodes can be selected. By klicking on a node theNode management page of the node is shown.

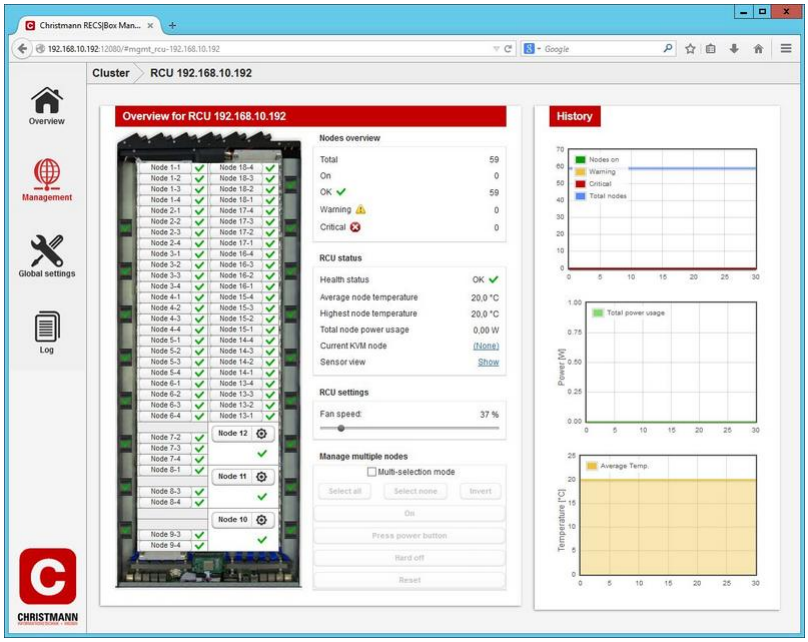

Figure 4.2:

A quick menu to control a node can be opened by klicking on the gear next to an CXP node. In this menu the node can be switched on and off and the KVM can be switched to the node.

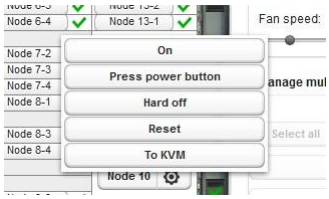

Figure 4.3:

Apalis nodes do not show a management pop-up button due to size constraints.
Click on the node button while pressing the „Shift" key to open the management pop-up instead of navigating to the node view.

When pressing the „Shift" key while clicking, the „Select all" and „Select none" buttons select only nodes currently on or nodes currently off, respectively.

Node management

On this page the selected node can be controlled and detailed status values and graphs can be seen.
By klicking on the arrow, pointing downwards in the upper bar next to the nodename, the other nodes of the unit can be chosen.

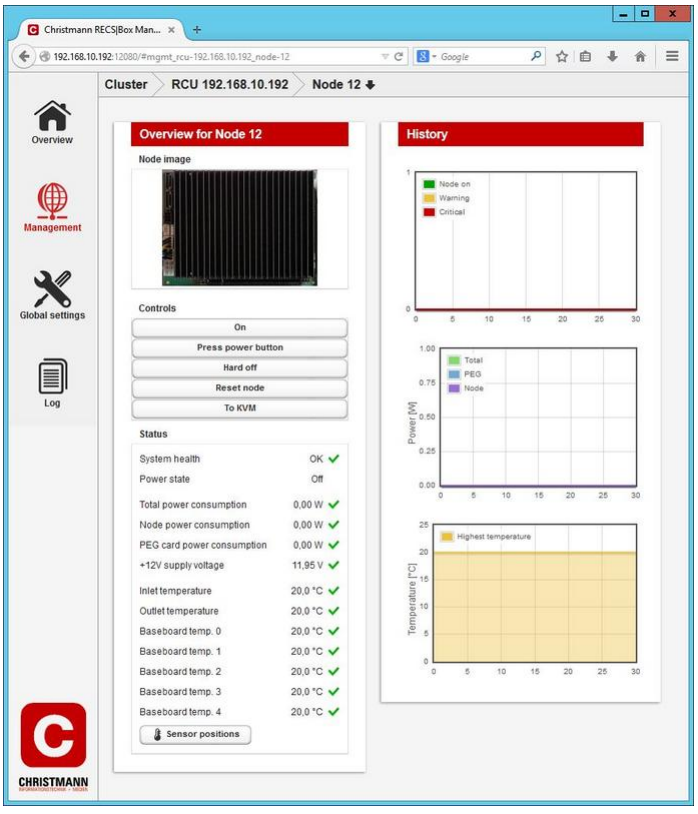

Figure 4.4:

4.1.3 Global settings

All IPs that are allowed to access the Nagios interface have to be listet here.
The firmware for the whole RECS®|Box can be uploaded here by klicking on the „Upload Firmware File" button and selecting the file. The update-process starts right after the file was uploaded.
For the update process **all modules will be powered off!**

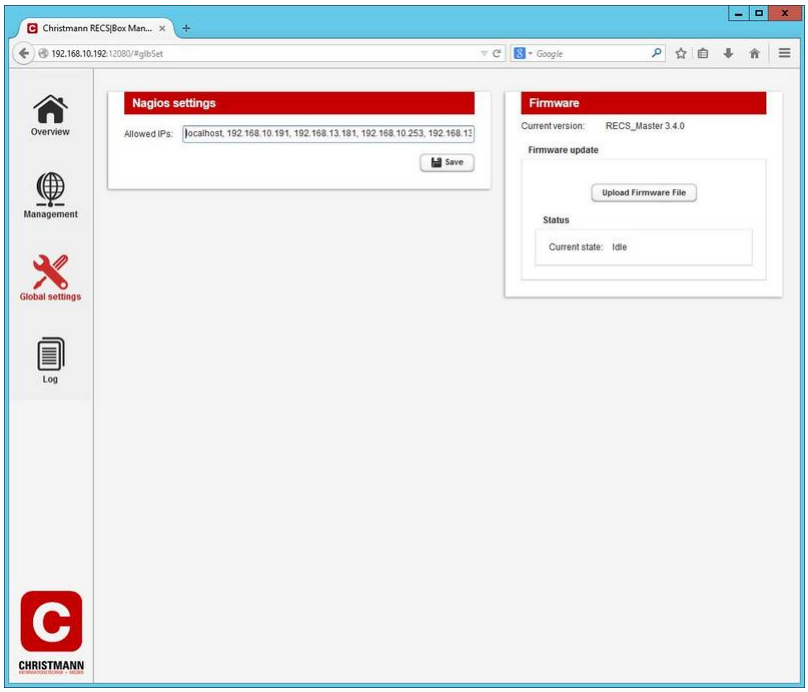

Figure 4.5:

4.1.4 Log viewer

In the system healths tab of the log page the status changes of the sensors, fan and boards can be seen.

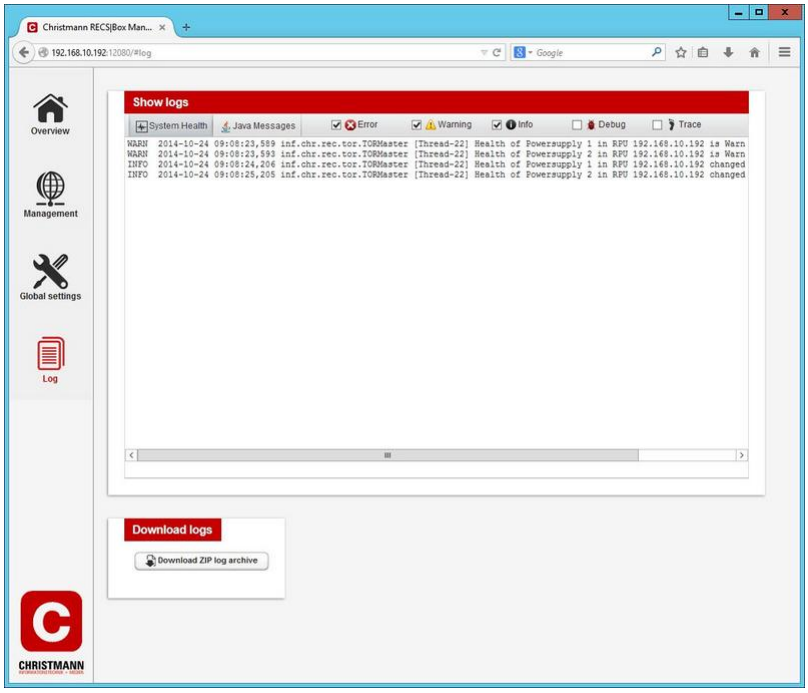

Figure 4.6:

In the java tab of the log page all messages regarding the software can be found.

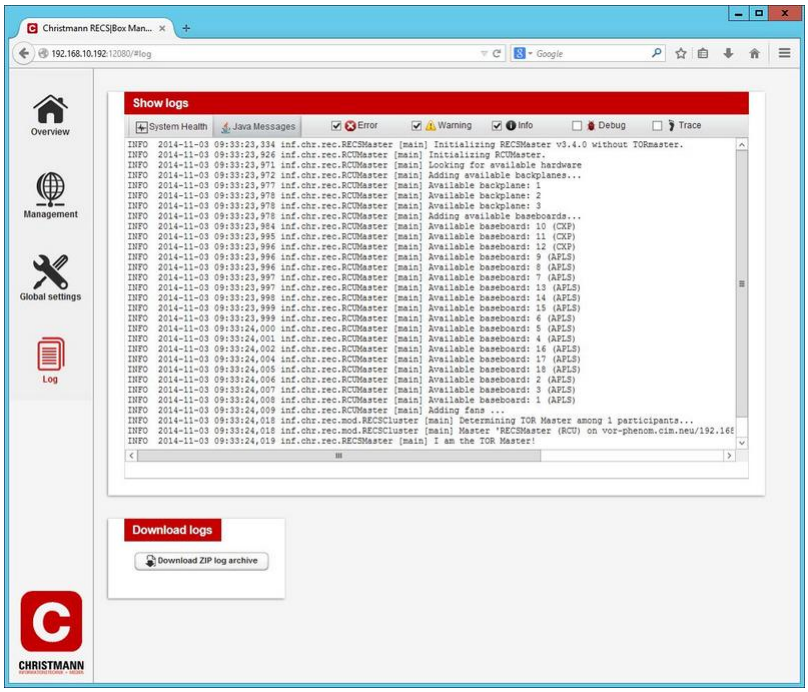

Figure 4.7:

Several filters can be set for both tabs at the top.
At the bottom the whole log can be downloaded as a ZIP file containing the individual logfiles.

4.2 REST API

4.2.1 Access

The RECS®|Box Management API is accessible via the IP-Address or the hostname of the TOR-Master of the cluster. The basic URL of the API has the format `https://TOR-Master/REST/` or `http://TOR-Master/REST/`.

Accessing the REST API requires HTTP Basic authentication. The authenticated user has to be in the „Admin" or „User" group to be able to execute the POST/PUT management calls.

4.2.2 Components

The RECS®|Box Management API makes all hardware components in the cluster available as XML trees in software. The following components are supported by the API:

Attribute	Description
`node`	A single node
`backplane`	A backplane can be equipped with zero or more baseboards
`baseboard`	A baseboard can be equipped with zero or more nodes
`rcu`	A RECS®|Box Computing Unit (RCU) can be equipped with zero or more baseboards
`rack`	A rack consists of several RCUs

Many resources also return lists of components. These are named according to the scheme <component name>List (e.g. nodeList, rcuList) and contain the elements of the list.

Example of a backplaneList:

```
<backplaneList>
<backplane position="1" id="RCU_84055620466592_BP_1" infrastructurePower="0.0">
<temperatures>24.0</temperatures>
<temperatures>25.0</temperatures>
<temperatures>26.0</temperatures>
<temperatures>27.0</temperatures>
<temperatures>28.0</temperatures>
</backplane>
</backplaneList>
```

Node

Example XML:

```
<node baseBoardPosition="0" maxPowerUsage="44" actualNodePowerUsage="32.426884399865166"
actualPEGPowerUsage="15.12053962324833" actualPowerUsage="47.54742402311349" architectur
baseBoardId="RCU_84055620466592_BB_1" health="OK" id="RCU_84055620466592_BB_1_0" inletTe
lastSensorUpdate="1465470151268" macAddressCompute="70:b3:d5:56:40:48" outletTemperature
highestTemperature="20.0" voltage="12.072700851453936"/>
```

The following table shows the possible attributes (some are optional) and their meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
actualPowerUsage	Actual power consumption of a node (Node + PEG)	W	Double
actualNodePowerUsage	Actual power consumption of a node (Node only)	W	Double
actualPEGPowerUsage	Actual power consumption of a PEG card	W	Double
maxPowerUsage	Maximum power the node can draw	W	Integer
baseBoardId	ID of the baseboard which hosts the node	-	String
baseBoardPosition	Position of the node on the baseboard	-	Integer
state	Power state of the node (0=Off, 1=On, 2=Soft-off, 3=Standby, 4=Hibernate)	-	Integer
architecture	Architecture (x86, arm, UNKNOWN)	-	String
health	Health status of the node (OK, Warning, Critical)	-	String
inletTemperature	Temperature of the inlet air	°C	Double
outletTemperature	Temperature of the outlet air	°C	Double
highestTemperature	Highest temperature measured on the node's baseboard	°C	Double
voltage	Supply voltage of the baseboard	V	Double
lastSensorUpdate	Timestamp of the last sensor update	ms	Long
macAddressCompute	MAC address of the NIC connected to the compute network (optional)	-	String
macAddressMgmt	MAC address of the NIC connected to the management network (optional)	-	String

In accordance to the component node the API offers nodeList which returns multiple instances of node.

Backplane

Example XML:

```
<backplane position="1" id="RCU_84055620466592_BP_1" infrastructurePower="0.0">
<temperatures>24.0</temperatures>
<temperatures>25.0</temperatures>
<temperatures>26.0</temperatures>
<temperatures>27.0</temperatures>
<temperatures>28.0</temperatures>
</backplane>
```

The attributes have the following meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
position	Position of the backplane in the RECS®|Box Computing Unit	-	Integer
infrastructurePower	Power usage of the infrastructure components on the backplane	W	Double
temperatures	List of temperatures measured on the backplane	°C	Double

In accordance to the component backplane the API offers backplaneList which returns multiple instances of backplane.

Baseboard

Example XML:

```
<baseBoard rcuPosition="6" baseboardType="APLS" id="RCU_84055620466592_BB_6" infrastructu
<nodeId>RCU_84055620466592_BB_6_1</nodeId>
<nodeId>RCU_84055620466592_BB_6_2</nodeId>
<nodeId>RCU_84055620466592_BB_6_3</nodeId>
<temperatures>20.0</temperatures>
<temperatures>20.0</temperatures>
<temperatures>20.0</temperatures>
<temperatures>20.0</temperatures>
<temperatures>20.0</temperatures>
</baseBoard>
```

The attributes have the following meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
rcuId	Unique ID of the RECS®|Box Computing Unit hosting the baseboard	-	String
rcuPosition	Position of the baseboard inside the RECS®|Box Computing Unit	-	Integer
infrastructurePower	Power usage of the infrastructure components on the baseboard	W	Double
baseboardType	Type of the baseboard (CXP, APLS)	-	String
nodeId	List of IDs of the nodes installed on the baseboard	-	String
temperatures	List of temperatures measured on the backplane	°C	Double

In accordance to the component baseboard the API offers baseboardList which returns multiple instances of baseboard.

RCU

Example XML:

```
<rcu rcuType="ANTARES" fanSpeed="60" rackId="RCK_1" name="RECSMaster␣(RCU)␣on␣192.168.56
<backplaneId>RCU_84055620466592_BP_1</backplaneId>
<baseBoardId>RCU_84055620466592_BB_1</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_2</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_3</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_4</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_5</baseBoardId>
<baseBoardId>RCU_84055620466592_BB_6</baseBoardId>
</rcu>
```

The attributes have the following meaning:

Attribute	Description	Unit	Data type
id	Unique ID for referencing the component	-	String
rackId	ID of the rack which hosts the RECS®|Box Computing Unit	-	String
rackPosition	Position of the RECS®|Box Computing Unit in the rack	-	Integer
name	Name of the RECS®|Box Computing Unit	-	String
ip	IP address of the RECS®|Box Computing Unit	-	String
rcuType	Type of the RECS®|Box Computing Unit (SIRIUS, ARNEB, ANTARES)	-	String
kvmNode	ID of the node to which the KVM system is switched (optional)	-	String
fanSpeed	Current speed setting of the fans in the RECS®|Box Computing Unit	%	Integer
backplaneId	List of IDs of backplanes which are installed in the RECS®|Box Computing Unit	-	String
baseBoardId	List of IDs of baseboards which are installed in the RECS®|Box Computing Unit	-	String

In accordance to the component rcu the API offers rcuList which returns multiple instances of rcu.

Rack

Example XML:

```
<rack description="Default␣rack" id="RCK_1">
<rcuId>RCU_84055620466592</rcuId>
</rack>
```

The attributes have the following meaning:

Attribute	Description	Unit	Data type
`id`	Unique ID for referencing the component	-	String
`description`	Description of the rack	-	String
`rcuId`	List of IDs of RECS®|Box Computing Units which are installed in the rack	-	String

In accordance to the component rack the API offers rackList which returns multiple instances of rack.

4.2.3 Resources

The resources are split into monitoring resources (for pure information gathering) and management resources (for changing the system configuration or state).

Monitoring

For monitoring the following resources are available:

Attribute	Description	HTTP Method
/node	Returns a nodeList with all nodes of the cluster	GET
/node/{node_id}	Returns information about the node with the given ID	GET
/baseboard	Returns a baseboardList with all baseboards of the cluster	GET
/baseboard/{baseboard_id}	Returns information about the baseboard with the given ID	GET
/baseboard/{baseboard_id}/node	Returns a nodeList with all nodes that are installed on the baseboard with the given ID	GET
/backplane	Returns a backplaneList with all backplanes of the cluster	GET
/backplane/{backplane_id}	Returns information about the backplane with the given ID	GET
/rcu	Returns an rcuList with all RECS®|Box Computing Units of the cluster	GET
/rcu/{rcu_id}	Returns information about the RECS®|Box Computing Unit with the given ID	GET
/rcu/{rcu_id}/baseboard	Returns a baseboardList with all baseboards that are installed in the RECS®|Box Computing Unit with the given ID	GET
/rcu/{rcu_id}/backplane	Returns a backplaneList with all backplanes that are installed in the RECS®|Box Computing Unit with the given ID	GET
/rcu/{rcu_id}/node	Returns a nodeList with all nodes that are installed in the RECS®|Box Computing Unit with the given ID	GET
/rack	Returns a rackList with all racks of the cluster	GET
/rack/{rack_id}	Returns information about the rack with the given ID	GET
/rack/{rack_id}/rcu	Returns a rcuList with all RECS®|Box Computing Units that are installed in the rack with the given ID	GET

Management

The management of individual components can be found under the „manage" path of the component.

Attribute	Description	HTTP method	Parameter
/node/{node_id}/manage/poweron	Powers on the node with the given ID and returns updated node XML	POST	
/node/{node_id}/manage/poweroff	Powers off the node with the given ID and returns updated node XML	POST	
/node/{node_id}/manage/reset	Resets the node with the given ID and returns updated node XML	POST	
/node/{node_id}/manage/switchkvm	Switches the KVM port of the RECS®|Box Computing Unit containing the node to the node with the given ID and returns updated node XML	PUT	
/rcu/{rcu_id}/manage/setfans	Sets the overall fan speed of the RCU with the given ID and returns the curent status of the RCU	PUT	percent={value}

Errors

Information about the success or failure of management requests are returned via HTTP status codes. Please have a look at RFC2616 for an overview about the defined HTTP status codes.

4.3 Nagios API

The software integration work to monitor the RECS®|Box System is quite simple as the the TOR-Master provides monitoring information in the Nagios native NRPE format. So only the NRPE plugin has to be installed and configured as follows. Here, a sample output can be found:

```
$ /usr/lib/nagios/plugins/check_nrpe -H 10.11.12.244 \
            -c check_temp -a 10.11.12.244 10 2 70:104 105:

OK - Temperature: 29 C|temp=29,000000;70:104;105;70,000000;105,000000
```

The options are used as following:

Option	Description
-H	Host to ask for data, this is always the IP of the TOR-Master (example: 10.11.12.244)
-c	Plugin to run. Can be check_temp or check_power
-a	Arguments to pass to the plugin, see more details in tables below

Arguments for check_temp plugin:

Argument example	Description
10.11.12.244	Get sensor values from device with this IP (RCU/RPU)
10	Get sensor values from this baseboard (1 - 18)
2	Get values from this sensor (max, inlet, outlet, 0, 1, 2, 3, 4)
70:104	Range of warning threshold
105	Range critical threshold (ending with : means open end)

Arguments for the check_power plugin:

Argument example	Description
10.11.12.244	Get sensor values from device with this IP (RCU/RPU)
10	Get sensor values from this baseboard (1 - 18)
2	Get sensor values from this node (1, 2, 3, 4)
80:109	Range of warning threshold [Watt]
110:	Range of critical threshold [Watt] (ending with : means open end

Chapter 5

RECSDaemon

5.1 Introduction

The RECSDaemon is a small program that can be installed on compute modules in a RECS®|Box system to be able to forward OS-level monitoring data to the integrated management system of the RECS®|Box. It is written to be cross-platform, running on Microsoft Windows as well as Linux and on x86, x64 and ARM systems. To be able to adapt to different platforms, the RECSDaemon uses plugins for different purposes. To configure these plugins and other settings an .ini file is used. The RECSDaemon is also able to execute commands sent by the management system to the node (e.g. for shutting down the OS gracefully).

5.2 Installation

The RECSDaemon is open source and available as source code on github, including a documentation how to compile and install it from source. Pre-packed Linux packages (rpm and deb) are under construction and will follow shortly.

The installation script will try to auto-detect some of the configuration parameters, but as platforms supported by the daemon are very diverse, manual configuration of remaining parameters most probably will be necessary.

RECSDaemon comes with both systemd and classical init scripts to allow automatically starting the daemon during system boot. Depending on your distribution, use either

```
systemctl enable RECSDaemon
```

or

```
update-rc.d RECSDaemon defaults
```

5.3 Configuration

The RECSDaemon will by default be installed to

`/opt/RECSDaemon`

The configuration file thus will be

`/opt/RECSDaemon/conf/recsdaemon.ini`

As this is a standard INI file, it is divided into different sections (denoted by square brackets) with parameters that are set to a certain value (e.g. updateInterval=1000). You can edit this file using a text editor, e.g. nano or vi. However, you probably will need root privileges to do so.

The different aspects that need to be configured will be described in the following chapters.

5.3.1 Communication

To be able to send sensor values and to receive commands, the RECSDaemon has to communicate with the management system of the RECS®|Box. This can happen via one of two different communication channels: On the one hand it is possible to use the internal management bus (I2C) of the RECS®|Box. On the other hand, regular TCP/IP is also supported when an external connection between the node the RECSDaemon is running on and the management Ethernet port of the RECS®|Box is made. Without this external connection, only I2C can be used.

As access to the external I2C bus of compute modules can differ between module vendors, there are multiple plugins available:

Plugin	Use for
LinuxCommunicatorDev	All modules that provide the external I2C bus as a /dev/i2c-* device
CommunicatorCongatec	COM Express modules from Congatec, uses CGOS
CommunicatorKontron	COM Express modules from Kontron, uses KEAPI
CommunicatorTCP	Communication via TCP/IP
CommunicatorDummy	Testing only, no external data transfer happening

Some of these plugins need further configuration. The necessary parameters are shown in the next chapters, all of which belong in the `[Comm]` section.

LinuxCommunicatorDev

If necessary, the I2C bus to be used can be changed. This is done by setting the `i2cBus` property. The value is used to determine the device path by appending it to the base path „/dev/i2c-".

Example configuration:

```
[Comm]
PluginName=LinuxCommunicatorDev
i2cBus=0
```

CommunicatorTCP configuration

When utilizing I2C as the communication channel, the RECSDaemon automatically can determine on which baseboard in the RECS®|Box it is running. However, when using TCP/IP, this information has to be supplied in the configuration file. This is done by setting the `baseboard` parameter to the number of the baseboard this module is currently plugged into. Please remember to update this value when you move the module to another baseboard.

Also necessary is the IP address of the RECS®|Box management system, which is set by the `controller` property. Set this value to the IP of the RCU the module is contained in.

Example configuration:

```
[Comm]
PluginName=CommunicatorTCP
baseboard=2
controller=192.168.13.56
```

5.3.2 Slot detection

Because some of the RECS®|Box baseboards support multiple modules, nodes need to know in which slot of the baseboard they are located. This is done by pins on the module connector that are tied to different logic levels depending on the slot position. These values can be read by the RECSDaemon using a SlotDetector plugin.

Plugin	Detection method
LinuxSlotDetectorGPIO	Accesses GPIOs via the Linux GPIO Sysfs interface

If no SlotDetector can or should be used, the slot can be manually changed by setting the `defaultSlot` property in the `[Slot]` section. Slots are configured from 0 to 3, but shown in the WebGUI of the RECS®|Box as 1 to 4.

LinuxSlotDetectorGPIO configuration

This plugin needs the numbers of the GPIO pins used to sense the slot position as used by the running kernel. These are set with the `Bit0GPIO` and `Bit1GPIO` settings.

Example configuration:

```
[Slot]
slotPluginName=LinuxSlotDetectorGPIO
Bit0GPIO=133
Bit1GPIO=134
```

5.3.3 Sensor providers

Sensor providers, as the name implies, add one or more sensors to the RECSDaemon's sensor list. This is in contrast to normal sensors that are manually configured (see chapter „Sensors"). Sensor providers can have two different internal designs: They either can directly provide and update sensor objects or they can use JSON as the interface to the RECSDaemon. Depending on the type, the plugins are enabled by adding them to either the `SensorProviders` or `SensorProvidersJSON` properties in the `[Plugins]` section.

Currently the following sensor provider plugins are available:

Plugin	Type	Sensors provided
LinuxSensorProviderEth	Direct	Ethernet link status/speed, bytes/sec RX/TX
SensorProviderSystem	Direct	CPU utilization, RAM free, System disk free
SensorProviderZynq	Direct	Serial transceiver link, utilization, bandwidth, frame errors, soft errors
SensorProviderZynqModule	JSON	Sensors provided by the module's microcontroller: 10 voltage rails, 5 power measurements, 7 temperatures

Example configuration:

```
[Plugins]
SensorProviders=SensorProviderZynq,LinuxSensorProviderEth,SensorProviderSystem
JSONSensorProviders=SensorProviderZynqModule
```

The plugins with „Zynq" in their name are only usable on the christmann RECS®|Box Zynq COM Express module.

Some of the plugin need further configuration. Please see the following chapters for details.

SensorProviderZynq

This plugin needs the base address of the monitoring peripheral it should read in the Zynq's memory space. Configure it with the `auroraMonitorBaseAddress` property in the `Plugins` section.

SensorProviderZynqModule

This plugin needs the name of the serial port of the management microcontroller on the Zynq module. Configure it with the `zynqSerialPort` property in the `Plugins` section. On Linux, it typically would be „/dev/ttyUSB0".

5.3.4 Sensors

Besides SensorProviders that automatically add a set of sensors when enabled, the RECSDaemon also allows manual configuration of additional sensors.

Currently, the following plugins can be used to instantiate sensors manually:

Plugin	Use for
SensorFileReader	Value read from arbitrary file

Sensors are instantiated in the `[Sensors]` section of the configuration file. The value `count` gives the total number of configured sensors. For each sensor (counting from 1 to count) three properties are read: `pluginNameX`, `sensorNameX` and `optionsX`, where `X` is replaced with the current count (e.g. `pluginName1`, `pluginName2`, etc.).

`pluginName` gives the name of the plugin to be used for this sensor, `sensorName` the name of the sensor (max. 29 characters) as shown in e.g. the WebGUI of the RECS®|Box. The value of the `options` property is given to the plugin and is used for further configuration. It is usually a space-separated list of „option=value" pairs.

An example configuration for one manual sensor could be:

```
[Sensors]
count=1

pluginName1=SensorFileReader
sensorName1=Test
options1=path=test.txt type=U8
```

SensorFileReader configuration

This plugin reads and parses the first line of a given file each time it is requested to update the sensor value. It recognizes the following options:

Option	Possible values	Required	Description
path	String	Yes	Path of the file to be read
type	U8, U16, double	Yes	Data type of the sensor. U8 and U16 mean unsigned integers of the given width.
multiplier	Double	No	When type is „double", the read value can be multiplied with the value in this option (e.g. to calculate a temperature from a raw sensor reading)
unit	°C	No	Unit of the sensor value
lowerCriticalThreshold	Double	No	Lower critical threshold as double
lowerWarningThreshold	Double	No	Lower warning threshold as double
upperWarningThreshold	Double	No	Upper warning threshold as double
upperCriticalThreshold	Double	No	Upper critical threshold as double

5.3.5 Other settings

In the `Update` section the rate with which the RECSDaemon updates its virtual sensors and sends them to the management system can be configured with the `updateInterval` property. The value is in milliseconds and gives the time between two updates. Beware that this only changes the update rate of the daemon. The RECS®|Box management system has its own update rate with which it collects sensor values. Thus, if you set this value smaller than the update rate of the management system, the effective update rate will still be that of the management system.

5.4 TCP/IP server

The RECSDaemon provides a simple TCP/IP server (by default on port 2023) that can be used by external programs to gain information or provide additional sensors.

The following commands are currently supported:

Command	Action
getnodeid	Returns ID of this node as shown in the WebGUI
monitor	Returns sensor data (current, voltage, temperatures) from the baseboard as a JSON string
addsensors	Used to add sensors by giving a JSON description
updatesensors	Updates sensors added previously
exit	Terminates connection

5.4.1 Getting monitoring data

Using the `monitor` command, the RECSDaemon can be utilized to obtain measurements with a faster update rate than via the regular RECS®|Box management system. Each time the daemon receives the command, it causes the baseboard to sample its local sensors and reads the results. Because this uses the module's I2C connection, it will not work when the CommunicatorTCP plugin is used.

The returned JSON string has the following format:

```
[
{"name": "nodeCurrent", "value": 0.0, "unit": "A"},
{"name": "pegCurrent", "value": 0.0, "unit": "A"},
{"name": "12vSupply", "value": 12.0, "unit": "V"},
{"name": "temperatures", "values": [0.0, 0.0, 0.0, 0.0, 0.0], "unit": "°C"}
]
```

The `pegCurrent` entry is optional and only available on COM Express baseboards.

5.4.2 Adding and updating sensors

Sensors added via the TCP/IP server are managed in groups that are identified by arbitrary strings. When adding sensors the group is defined via the `addsensors` command and referenced when using the `updatesensors` command.

See chapter „JSON sensor description" for the syntax of the expected JSON string for `addsensors`. To add a group „mySensors" with one sensor „mySensor", send the following command, terminated with a newline (\n):

```
addsensors mySensors [{"name": "mySensor", "dataType": "double"}]
```

To update a sensor group, send a `updatesensors` command with the group name and a JSON array with as many values as the group contains sensors:

```
updatesensors mySensors [1.0]
```

5.5 JSON sensor description

A JSON sensor description contains one or more sensors, thus the outer element is an Array ([]). Inside that, the sensors are defined as objects ({}) that support the following properties:

Property	Possible values	Required	Description
name	String, max. 29 characters	Yes	Name of the sensor
dataType	U8, U16, U32, U64, double, string	Yes	Data type of the sensor. U8 to U64 mean unsigned integers of the given width.
maxDataSize	1 - 255	Only for dataType „string"	Maximum length of the sensor value
unit	W, A, V, °C, RPM	No	Unit of the sensor value
lowerThresholds	JSON array with two doubles	No	Lower critical and warning threshold as doubles
upperThresholds	JSON array with two doubles	No	Upper warning and critical threshold as doubles

Example:

```
[
  {
    "name": "1.0␣V",
    "dataType": "double",
```

```
    "unit": "V",
    "lowerThresholds": [0.5, 0.8],
    "upperThresholds": [1.2, 1.5]
  },
  {
    "name": "Temp. Heatsink",
    "dataType": "double",
    "unit": "°C"
  }
]
```

Chapter 6

PXE-Server

It is possible to boot modules via a PXE server, e.g. to avoid the installation of local disk images.

6.1 Installing the PXE server

This tutorial was done with a Debian 8. It will only be possible to automatically boot an OS according to the module's MAC address. In this configuration there is no GUI to select a boot image.

All operations where done with root permissions.

First the following services have to be installed:

```
apt-get install dnsmasq nfs-kernel-server
```

`dnsmasq` serves as the DHCP Server to tell the clients what IP address to use and which kernel they have to load. Also it runs the TFTP Server that provides the kernel images.

`nfs-kernel-server` provides the NFS network share for the RootFS and a persistent storage.

6.1.1 Setting up dnsmasq

The config file can be opened with

```
nano /etc/dnsmasq.conf
```

All DHCP Options used in the following config can be found here Link The following shows an example configuration for 3 different modules and 4 different kernels.

```
#Binds the DHCP server to eth0
interface=eth0
#Sets the IP range with a lease time of 8 hours
dhcp-range=192.168.13.201,192.168.13.249,255.255.255.0,8h

#Enables the TFTP service of dnsmasq
enable-tftp
#Bind the TFTP root folder  to the set folder
tftp-root=/srv/tftpboot/
#DHCP Option 66 sets which TFTP Server will be used by the client
dhcp-option=66,192.168.13.80
#DHCP Option 42 sets a NTP time server
dhcp-option=42,130.133.1.10


#The following sets a flag for all MAC addresses starting with the given number,
#so it is possible to set different setups for different modules
dhcp-host=00:14:2d:*:*:*,set:toradexApalis
```

```
dhcp-host=00:0e:c6:*:*:*,set:toradexColibriAndroid
#"set:christmannExynosLinux" has to be changed to "set:christmannExynosAndroid" in order
#the Android kernel
dhcp-host=70:b3:d5:56:*:*,set:christmannExynosLinux


#These different tags can be used with "tag:"tagname","option""
#"dhcp-boot" defines which kernel will be loaded from the TFTP root path
#"dhcp-option" gives an option to the set MAC address group

#Kernel filename for Toradex Apalis T30 provided by the TFTP Server
dhcp-boot=tag:toradexApalis,toradexApalis_uImage
#Root path that is shared by the nfs-kernel-server
dhcp-option=tag:toradexApalis,17,/srv/nfs/apalisT30/rootfs/

#Kernel filename for Toradex Colibri T20
dhcp-boot=tag:toradexColibri,toradexColibri_uImage
#Root path
dhcp-option=tag:toradexColibri,17,/srv/nfs/colibriT20/linux/rootfs/

#Kernel filename for Christmann Apalis Exynos Linux
dhcp-boot=tag:christmannExynosLinux,uImage-ExynosLinux
#Root path
dhcp-option=tag:christmannExynosLinux,17,/srv/nfs/apalisExynos/linux/rootfs/

#Kernel filename for Christmann Apalis Exynos Android
dhcp-boot=tag:christmannExynosAndroid,uImage-ExynosAndroid
#Root path
dhcp-option=tag:christmannExynosAndroid,17,/srv/nfs/apalisExynos/android/rootfs
```

After the `dnsmasq.conf` has been changed, the dnsmasq service has to be restarted in order to apply the changes using:

```
/etc/init.d/dnsmasq restart
```

Then the `uImage` files must be copied to `/srv/tftpboot/*` with the name that was defined in `dhcp-boot`.

6.1.2 Setting up nfs-kernel-server

The RootFS and/or a persistent storage are exported to the client via NFS. This is the location where the root filesystem of the desired distribution must be copied to.

The shared/exported folders can be edited in this file:

```
nano /etc/exports
```

Here is an example configuration:

```
/srv/nfs/apalisT30/rootfs       *(rw,no_root_squash,no_subtree_check)
/srv/nfs/storage/               *(rw,async,no_root_squash)
/srv/nfs/colibriT20             *(rw,no_root_squash,no_subtree_check)
/srv/nfs/apalisExynos/          *(ro,no_root_squash,no_subtree_check)
```

First in line is the folder that will be shared via NFS.
The * in front of the bracket sets the clients that are allowed to access the share. In this case everyone can access the share, but also singe IP addresses or ranges can be set.
The Arguments in the brackets are the following:

rw	Read Write access
ro	Read Only access
async	Better performance with the danger of data loss if the server crashes or is shut down. Default is sync
no_root_squash	Enables access to files that are owned by root on the server
no_subtree_check	disables the subtree checking when accessing a file, this in on by default

A detailed description can be found at Ubuntuusers or with `man exports`
To apply the changes the `nfs-kernel-server` has to be restarted.

```
/etc/init.d/nfs-kernel-server restart
```

6.2 Client configuration

The clients have to be set up in order to boot from the PXE server and not from the internal flash storage or HHD.
For CXP modules the network boot option in the BIOS has to be activated. In most cases that is enough.

The Apalis and Colibri modules must have an u-boot that supports booting over network (version and higher). And some things ave to be set in the u-boot enviornmentals as described below.
The u-boot can be reached by connecting a serial console to the module and then pressing any key when starting the module to abort the booting process.

6.2.1 Christmann Apalis Exynos

Tested with **U-Boot 2015.07-rc1-00408-g012681b-dirty**
The following variables have to be set in u-boot:

```
setenv usbethaddr
setenv nfsboot 'usb reset; dhcp; mmc dev 1; mmc read 22000000 3000 100; bootm 23e00000 -
setenv bootcmd 'run nfsboot; usb start; mmc dev 1; mmc read 0x20008000 600 2700; mmc rea
setenv bootargs 'root=/dev/nfs rw netdevwait console=ttySAC2,115200n8 init --no-log'
```

:!: But it is necessary at the moment to have an SD card with the original Image, or with the flattend device tree blob at sector 3000 of the SD card.
:!: At the moment the kernel stops booting with `drm_kms_helper: panic occurred, switching back to text console`

6.2.2 Toradex Apalis T30

Tested with **U-Boot 2014.10**
Only the option for `nfsboot` has to be added to the `bootcmd` in u-boot. Note that in the following steps the MAC address has to be replaced with the actual address of the module.:!:
If only network boot is desired the following commands have to be executet in the u-boot enviornment:

```
setenv serverip
setenv ipaddr
setenv ethaddr 00:14:2D:00:00:00
setenv nfsboot 'run setup; setenv bootargs ${defargs} ${nfsargs} ${setupargs} ${vidargs}
setenv bootcmd 'run nfsboot; echo; echo nfsboot failed'
saveenv
```

If the original bootorder should not be replaced the commands are the following:

```
setenv serverip
setenv ipaddr
setenv ethaddr 00:14:2D:00:00:00
setenv bootcmd 'run nfsboot; echo; echo nfsboot failed; run emmcboot; echo; echo emmcboo
saveenv
```

But this will result in one error message `*** ERROR: erverip' not set` because the module then trys to load a flash image from a server whos IP is not set in `serverip`.

Chapter 7

Glossary