Competence in Surface Analysis

Surface Analysis Technology | Vacuum Components | SPECS® | Surface Analysis System Software | Computer Technology

HSA3500
HSA3500 plus
Power Supply

User Manual

Version 1.7

SPECS®

User manual for the HSA3500 HSA3500 plus Power Supply.

Version 1.7 printed 18 May 2010.

SPECS order number for this manual: 78 000 186

SPECS GmbH - Surface Analysis
and Computer Technology
Voltastrasse 5
13355 Berlin
GERMANY
PHONE:+49 (0)30 46 78 24 -0
FAX: +49(0)30 4 64 20 83
http://www.specs.de

Chapter T Table of Contents

Chapter 1 Introduction

1.1 Preface

This manual describes the entire HSA3500 and HSA3500 plus power supply manufactured by SPECS. For all common features we will use the term ”HSA” for both. When it is necessary to point out specific features, we will use the terms HSA3500 and HSA3500 plus, respectively.

1.2 General Information

With the HSA 3500 plus SPECS presents a new versatile high voltage power supply for electrostatic field applications. The HSA3500 plus is the further development of HSA3500. The modular design of the unit allows independent setting of all voltages - no voltage dividers are used.

Each module is fully galvanically floating, highly stable and linear. The voltages are controlled by high-precision 24-bit digital-to analog converters with an overall maximum settle time of 3 ms.

The HSA consists of a main CPU and various voltage and current modules. All communication goes over the CAN -Bus to the main CPU. The main CPU distributes all CAN - Bus messages to respective modules. The device provides +5V, +15V for the modules and main CPU, and +24V for external devices via CAN-Bus (e.g. EC10). On the back panel you connect all the hardware to the analyzer (e.g. SPECS Phoibos) and detector (e.g. SPECS CCD detector). On the front panel you can switch the HSA on, off and get the device status via LEDs.

Every module is calibrated with a high precision voltmeter by SPECS. That means the DAC values are assigned to voltages or current values and stored in the EEPROM (flash) from the module.

1.3 Physical Specifications

Size:	L x W x H, 45cm x 32cm x 50cm
Weight:	approx. 15kg depending on configuration
Data Connection:	CAN-Bus (500 kBaud)
Line Input:	100V – 250V, 47 – 63Hz

The HSA3500 plus has 5 additional modules, with specifications given below:

	3,5kV Bi Lens	3,5kV Bi Analy	3,5kV Uni	400V Bi	CPU / 150mA
Resolution	20 Bit	20 Bit	16 Bit	20 Bit	13 Bit
DACs	1	1	1	3	1
Relays	2	2	1	0	0
Ranges	4	2	1	3	1
TK	2.5 ppm	1.5 ppm	-	2.5 ppm	n / a
Long Term (one year operation)	400 ppm	50 ppm	-	50 ppm	-

These individual modules also have their own voltage characteristics:

400V Bi:

Range	+/- 400V	-20V / +100V	+/- 10V
Step size	805µV	125µV	20µV
Noise (with filterbox)	< 200µV	< 250µV	

3,5kV Bi Lens:

Range	+/- 3500V	+/- 1500V	+/- 600V	+/- 150V
Step Size	7,2mV	3,1mV	1,27mV	318µV
Noise (with filterbox)	< 850µV	< 400µV	< 100µV	< 50µV

3,5kV Bi Analy :

Range	+/- 3500V	+/- 1500V
Step Size	7,1mV	1,7mV
Noise (with filterbox)	< 600µV	< 500µV

1.4 Safety Instructions

Before performing any electrical or electronic operations, please consult 'SPECS Safety Instructions' and follow them closely. Please consult as well any documentation of supplied material from third party manufacturers before any operation and strictly follow the safety instructions given therein.

1.5 Initial Inspection and Packaging List

Visual inspection of the component is required to ensure that no damage has occurred during shipping. Should there be any signs of damage or parts missing, contact SPECS immediately. Please check the delivery according to the packaging list for completeness.

Quantity	Item
1	HSA3500 or HSA3500 plus
1	Printed HSA3500 Manual
1	Power cord (200V or 115V)
2	CAN-Bus cable
2	CAN-Bus terminator
2	HV cable (Channel HV, Channel Base)
1	HV cable (Screen) optional
1	HV Interlock override connector
1	EC10 Can-Ethernet-Adapter
1	Ethernet cable
1	Specification sheet (optional)

Table 1: Packaging List of the HSA

1.6 EC10 Can-Ethernet-Adapter

1.6.1 Introduction

The EC10 is an external CAN-Bus adapter for the PC. It is connected to the PC via an ethernet network. The PC's access to CAN-Bus controlled devices like the HSA is provided by the EC10 and the communication between PC and EC10 is handled by the standard TCP/IP network protocol.

Figure 1: Rear Side of the EC10

Figure 2: Front Side of the EC10

1.6.2 Connectors, diagnostic LEDs and switches

Each connector, LED, and switch is labelled, and their purpose is described below.

Label	Type	Description
Ethernet	RJ45	Ethernet Network Connector
CAN	DB9 connector male	CAN-Bus Connector

CAN	DB9 connector female	CAN-Bus Connector
RS-232	DB9 connector female	Serial port. Not used in normal operation. Do not connect anything to this connector.
24V	DC socket	Connector for external power supply. Not used in normal operation. Power supply is provided via the CAN-Bus.
TX	LED green	Ethernet transmit activity indicator. This LED will flash while the EC10 is transmitting data on the network.
LNK	LED green	Ethernet link indicator. This LED will flash if the data link between the EC10 and the HUB/switch is estalished.
RX	LED yellow	Ethernet receive activity indicator. This LED will flash while the EC10 is receiving data from the network.
ONL	LED red	This LED will flash while the PC software is connected to the EC10, e.g. while SpecsLab is running.
CAN	LED green	CAN Bus activity indicator. This LED will flash, while the EC10 receives or transmits CAN Bus packets.
SER	LED yellow	Serial interface activity indicator. This LED will not flash in normal operation.
PWR	LED green	Power indicator. This LED will flash, as long as 24 Volt power is applied to EC10.
CFG	Push-button	This button is used to initiate the con figuration of the IP-Address by the use of EC10Config program. The usage of EC10Config is described in QuickGuide-EC10.pdf.

Table 2: LEDs and Connectors of the EC10

At power-up the EC10 will execute a self-test sequence that is indicated by quickly (300 Milliseconds) flashing the diagnostic LEDs "PWR", "CAN", "ONL" and "SER". If the self-test is successful, the "PWR" LED will remain on, the other LEDs are turned off. If the self-test fails, the "PWR" LED will blink with a frequency of 2 Hertz and the other LEDs will indicate the reason of failure. In this case the device will not be operable and should be returned to SPECS for service.

If no LED flashes at all, the presence of the 24 Volt power supply should be checked. Contact SPECS for furter advice.

1.6.3 Power Requirements

The EC10 needs a 24 Volt supply voltage. Usually this voltage is provided by the HSA via the CAN-Bus cable. A 24 Volt supply voltage can be fed to the EC10 via the DC socket labelled "24V" at the front side of the housing as well. The typical current drain of the device is 48 mA.

1.6.4 Pin Setting of the CAN Bus Connector

The Pin setting of the CAN BUS connector is given in table 6, the naming is depicted in figure 3.

CAN Bus Pin	Name
1	+24V
2	CAN Low
3	0V
4	+24V
5	0V
6	
7	CAN High
8	
9	

Table 3: Pin setting of the CAN Bus connector

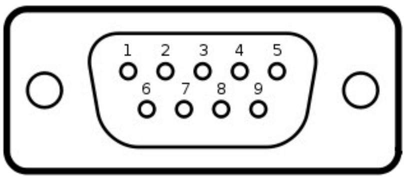


Figure 3: CAN Bus Connector (a common DBUS-9)

1.6.5 Galvanic Decoupling

The CAN-Bus connection is opto-electrically decoupled. The ethernet connection is magnetically decoupled. The electrical strength of both decouplings is greater than 500 Volts DC.

1.6.6 Propagation Delay

The propagation delay is the time the EC10 needs to convert an ethernet packet into a CAN-Bus packet and vice versa. This time is approximately 190 microseconds in each direction. The run-time of the ethernet packet (approximately 50 microseconds) adds to this time.

1.7 Front / Back Panel Description

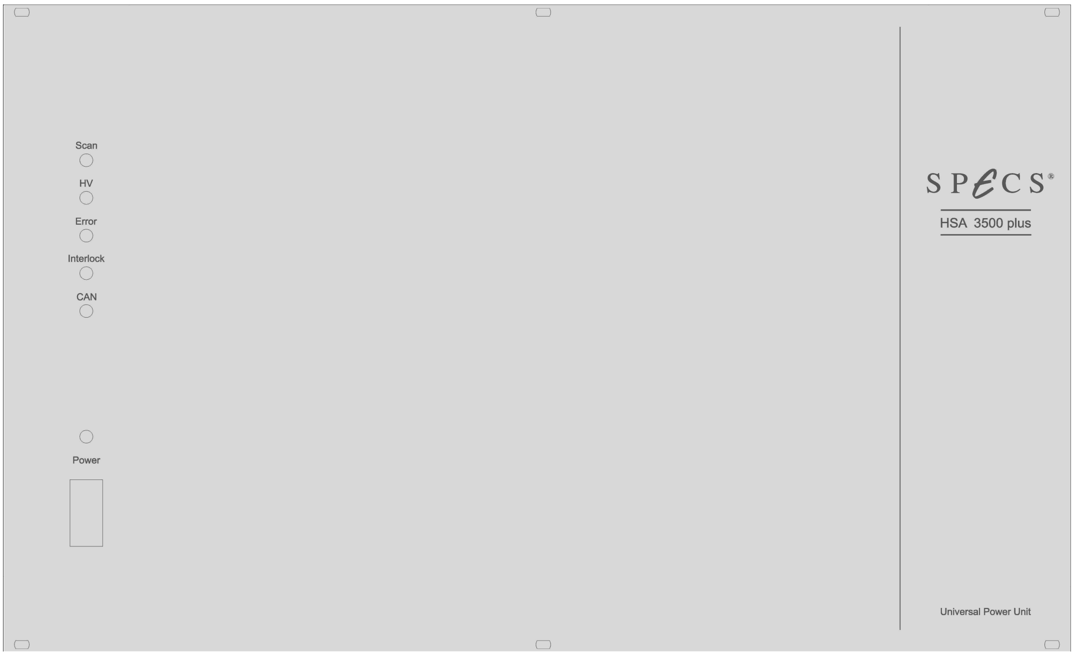


Figure 4: HSA3500 plus front panel

Label	Description
LED Scan (green)	Currently measuring (e.g. with SPECSLAB)
LED HV (yellow)	The high voltage output is activated.

LED Error (red)	Error - something not to specifications
LED Interlock (yellow)	This LED will be on if the (vacuum)interlock, is engaged.
LED CAN (green)	This LED will blink while the HSA is not connected and will be on while the HSA is connected to PC.
LED Power (green)	This LED will be on when HSA is powered up.
Power Switch	Switches the HSA on or off

Table 4: HSA3500 plus front panel description

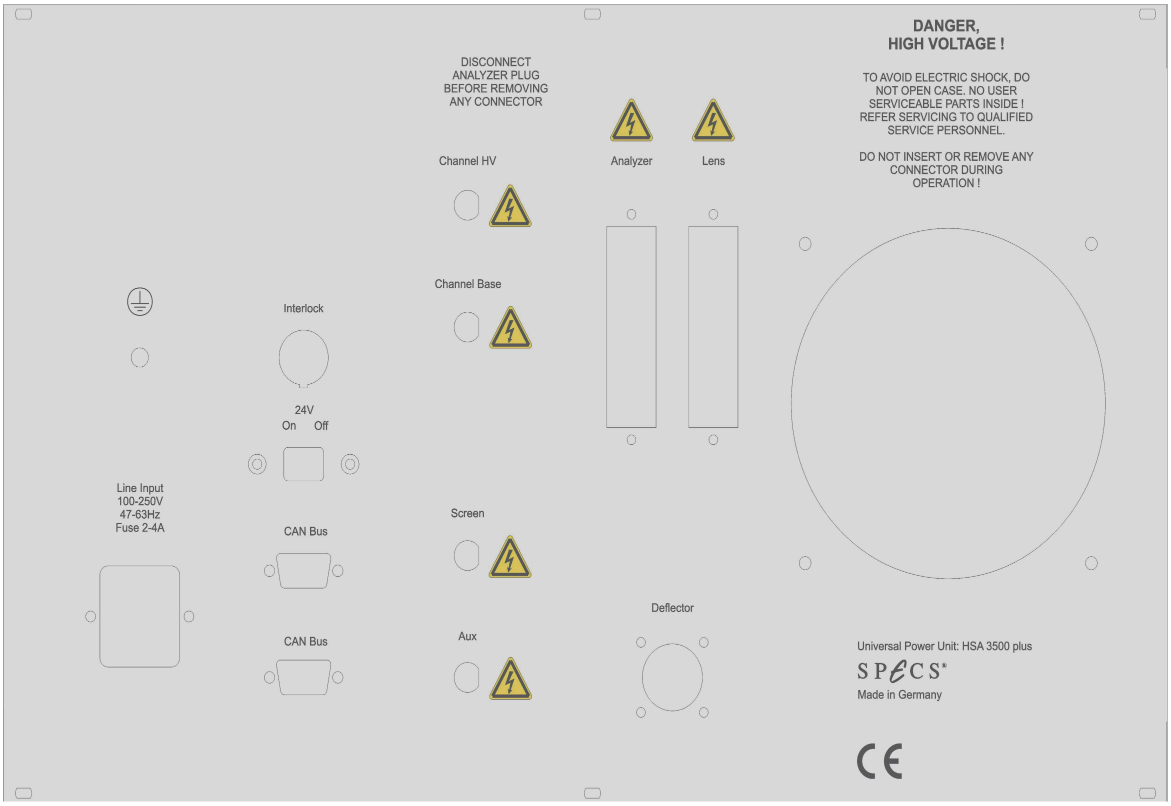


Figure 5: HSA3500 plus back panel

Label	**Type**	**Description**
Channel HV	SHV	Connection to electron multiplier output (detector andode)
Channel Base	SHV	Connection to electron multiplier input

		(detector cathode)
Screen or DLD	SHV	Connection to detector (CCD, DLD only)
Mesh or Aux	SHV	Connection defined by user
Line Input	3pin	Input for power cord
Interlock	3pin	Interlock connector
Sliding Switch		Activate the 24V on CAN-Bus connector
CAN BUS	Sub-D 9pin	Connection to CAN-Bus (Male)
CAN BUS	Sub-D 9pin	Connection to CAN-Bus (Female)
Deflector	6 pin	Connection to deflector (optional)
analyzer	30pin	Connection to analyzer (Phoibos)
Lens	30pin	Connection to lens (Phoibos)
Fan		Cooling the unit
Grounding Screw	1pin	Connection to ground

Table 5: HSA3500 plus back panel description

1.8 General Internal Layout

In Figure 6 you can see the general internal layout of the HSA. At the front side you have the even slot numbers and on the back side you have the odd slot numbers. The slot configuration depends on hardware and wiring type. For details see Table 42 in the Appendix.

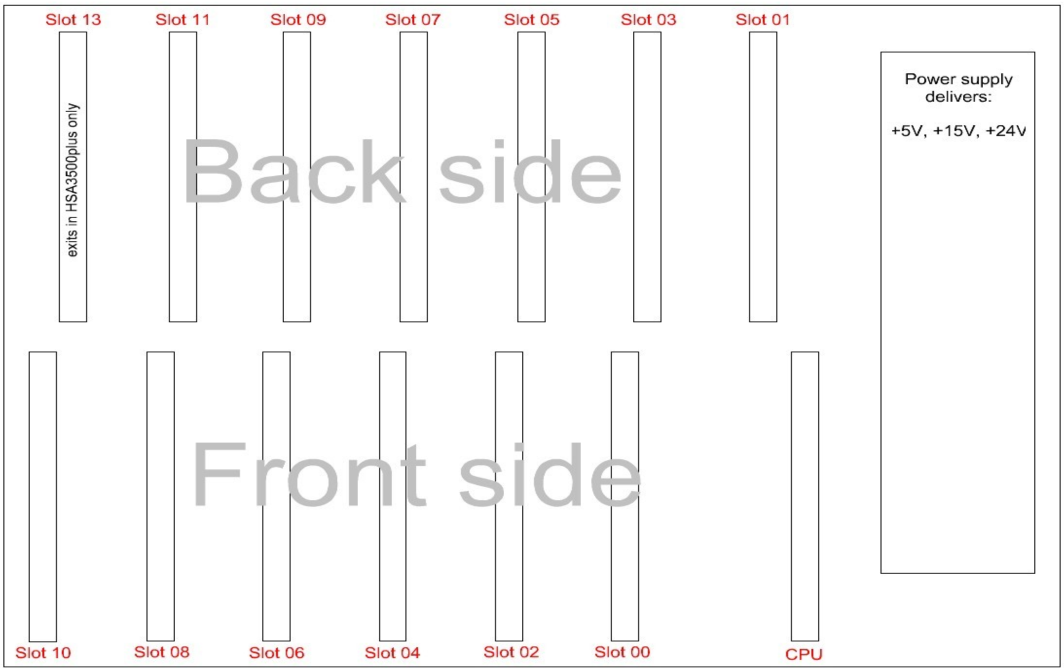


Figure 6: HSA General Internal Layout

1.9 Modules

There are different modules within the HSA. All modules have an internal type number (Table 6 and Table 7). Every module has a serial number and production date printed on the board.

Type	Name
-	Main CPU
0xf1	3.5kV bipolar
0xe1	400V bipolar
0xd1	3.5kV unipolar
0xb1	150mA unipolar

Table 6: HSA3500 modules

Type	Name
-	Main CPU with current module 150mA
0x04	3.5kV Lens bipolar
0x14	3.5kV Analyzer bipolar
0x24	400V bipolar
0x34	3.5kV unipolar

Table 7: HSA3500 plus modules

1.10 HSA Ranges

The SPECS power supply HSA3500 has one main CPU and four modules with different voltage ranges. Smaller voltage ranges has always a better resolution. See Table 8 for details. The SPECS power supply HSA3500 plus has one main CPU with current module on board and four modules with different voltage ranges. See Table 9 for details.

Type	Module	Min. Voltage	Max. Voltage
	Main-CPU		
0xf1	3.5kV	-3500V	+3500V
0xf1	3.5kV	-1500V	+100V
0xe1	400V	-400V	+400V
0xe1	400V	-40V	+40V
0xd1	3.5kV uni	-0.1V	+3500V
0xb1	150mA	0mA	150mA

Table 8: *HSA3500* Ranges

Type	Module	Min. Voltage	Max. Voltage
	Main-CPU	0mA	150mA
0x04	3.5kV Lens	-3500V	+3500V
0x04	3.5kV Lens	-1500V	+1500V
0x04	3.5kV Lens	-600V	+600V
0x04	3.5kV Lens	-150V	+150V
0x14	3.5kV analyzer	-3500V	+3500V
0x14	3.5kV analyzer	-1500V	+100V
0x24	400V	-400V	+400V
0x24	400V	-100V	+20V
0x24	400V	-10V	+10V
0x34	3.5kV unipolar	+3500	-100

Table 9: *HSA3500 plus Ranges*

Don't mix HSA3500 modules with HSA3500 plus modules in the same device.

Chapter 2 Using the HSA

2.1 Installation

For installation see PHOIBOS Manual (SPECS order number: 78 000 101).

2.1.1 Vacuum interlock

For operation of the HSA the vacuum interlock connector has either to be connected to a pressure gauge or must be bridged temporarily with the supplied connector.

Bridging the vacuum interlock can cause great damage in case of sudden pressure rise.
Use at your own risk!

The pin setting of the connector is shown in figure 7. So to bridge the interlock the pins 1 and 2 have to be shortend.

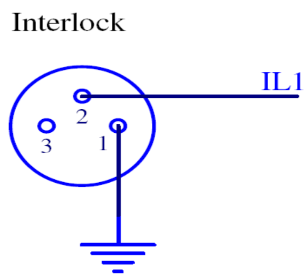


Figure 7: Vacuum interlock connector

2.2 Software for Interfacing with the HSA

Your options for controlling the HSA are:

- SpecsLab2, including the HSA Juggler.
- Custom made programs using the SpecsLab DLL “SL2_AD.dll”, see SL2_AD.h and the Software 2D Manual (SPECS order number: 78 000184)
- National Instruments' Labview® is also supported using the special DLL SL2_AD_VI.dll. SPECS provides also a sample VI in the folder programming in side the SPECSLab2 software package. For implemenation tips seeSL2_AD_DLL_Test.vi and SL2_AD.h from this software package.
- Other possibilities are remote control of applications like SpecsLab2 using Windows Messages
- Using directly the public CORBA-interfaces of the SPECS device drivers is also possible but intermediate programming skills is for this solution a good thing.
- A low level tool for advanced users called “hsa3500tool” is also available. See chapter 4.4 on page 65.

2.3 Troubleshooting

This chapter provides a guide for troubleshooting problems with the HSA. Possible problems fall into three categories.

1. More than one beep after switching the HSA on
   - Incorrect hardware type
   - Module is in wrong slot or module is missing

2. Error LED (red) on front panel is on
   - Recalibrating from one or more modules faulty (HSA3500 only)
   - No firmware
   - CAN-Bus error

3. Inobservable errors

   - No output voltage

   - Incorrect or old firmware

   - Internal ADC damaged

2.3.1 Diag-Tool

To make sure the HSA works correctly it might be a good idea to call the the diagnostic tool "Diag-Tool" from time to time to ensure proper working of the device. For troubleshooting purposes this is also the first thing to do. This tool can check the HSA module by module and write the result to a ASCII log file.

For imaging detectors (CCD and DLD) do not perform this test with detector cables (CHV, CB, Screen or DLD) plugged in, since the high voltages applied during the tests could destroy them. Also the voltage limiter box (diode box) should be disconnected, so that no wrong error messages are produced.

The Diagnostic Tool "Diag-Tool" gets the hardware type (e.g. 101) from the HSA Main CPU and reads the test file (e.g. HsaRaw101.dat). All information (mode, relays, modules) about the test is stored in this file. The next test is to check every module in every mode in the HSA with the following procedure:

- Read basic information from module (serial number, firmware, module type and calibration date)
- Set DAC voltage and read ADC in every mode and module
- Show errors, warnings and successful tests on display
- Detailed information saved in logfile (SpecsLab2\tools.bin\logfiles\diag-xxxx.log).

1. To check if the EC10 is reachable via the TCP/IP connection call the script "ec10-ping.bat" in the tools.bin folder. This tool pings the EC10 with the IP address set in the registry (HKEY_LOCAL_MACHINE\SOFTWARE\SPECS\DeviceNetServer\DLLParams*).
   Note: The ec10-ping tool can not be used for internal PCI or ISA CANBus cards.

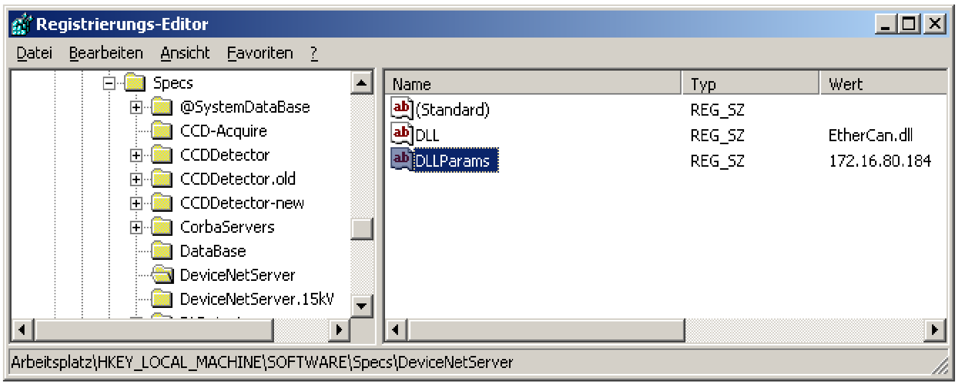


Figure 8: Windows Registry editor

2. Go to the SPECS tools.bin directory by clicking the Link “Tools” in the folder Start->SPECS. Start the diag tool by selecting the appropriate diag-*.bat file, this is diag-eth.bat for an EC10, diag-pci.bat for a PCI-card, or diag-isa.bat for an ISA-Card.
3. The log file containing the results will be saved in the folder logfiles. Calling the diag tool more than once will create multiple logfiles with suffxies -1, -2 and so on. Logfiles will **never** be overwritten.
4. DiagTool started the procedure and shows results (error, warning or successful) for every mode and module on the command box (see Figure 10). You get a summary for all errors at the end of this test.
5. If you need detailed information (see Figure 11) about the measured results open SpecsLab2\tools.bin\logfiles\diag-xxx.log”. The highest number from log file (e.g. diag-009.log) is the latest one.

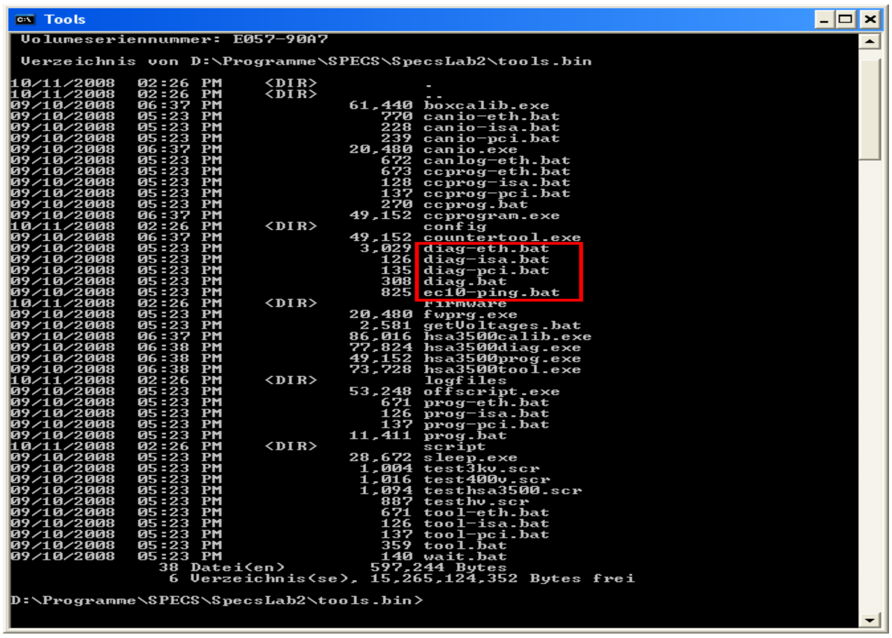


Figure 9: Start diag-.bat*

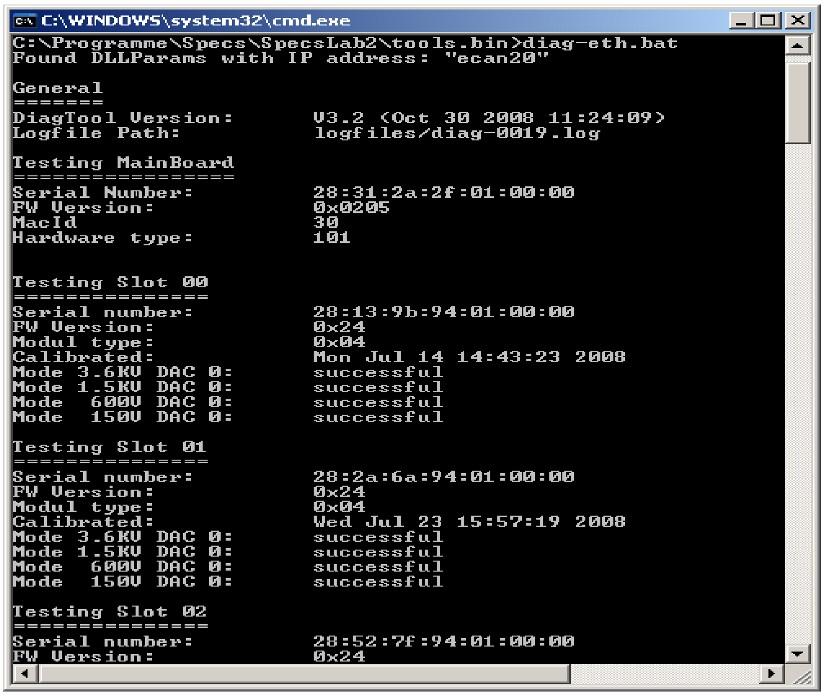
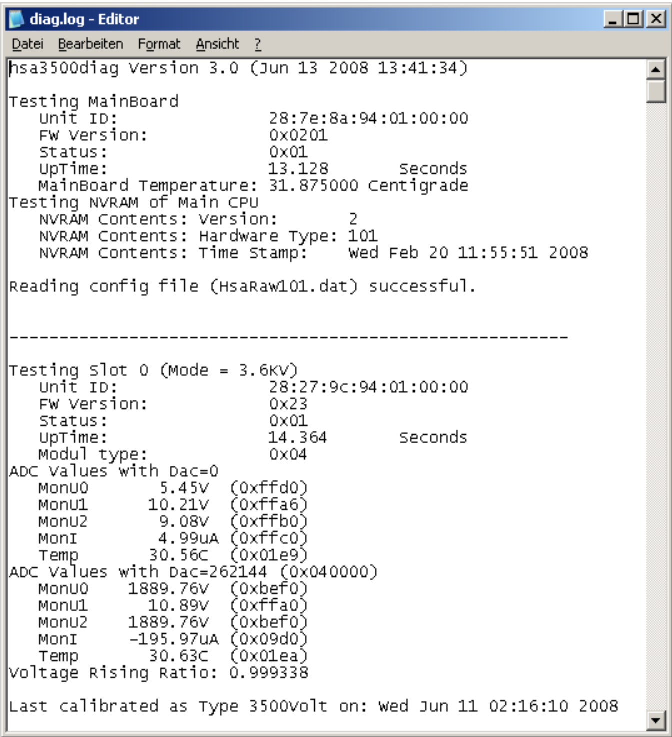


Figure 10: DiagTool command box



diag.log - Editor

Datei Bearbeiten Format Ansicht ?

```
hsa3500diag version 3.0 (Jun 13 2008 13:41:34)

Testing MainBoard
   Unit ID:                  28:7e:8a:94:01:00:00
   FW Version:               0x0201
   Status:                   0x01
   UpTime:                   13.128       Seconds
   MainBoard Temperature: 31.875000 Centigrade
Testing NVRAM of Main CPU
   NVRAM Contents: Version:        2
   NVRAM Contents: Hardware Type: 101
   NVRAM Contents: Time Stamp:     Wed Feb 20 11:55:51 2008

Reading config file (HsaRaw101.dat) successful.


--------------------------------------------------------

Testing Slot 0 (Mode = 3.6KV)
   Unit ID:                  28:27:9c:94:01:00:00
   FW Version:               0x23
   Status:                   0x01
   UpTime:                   14.364       Seconds
   Modul type:               0x04
ADC Values with Dac=0
   MonU0        5.45V  (0xffd0)
   MonU1       10.21V  (0xffa6)
   MonU2        9.08V  (0xffb0)
   MonI         4.99uA (0xffc0)
   Temp        30.56C  (0x01e9)
ADC Values with Dac=262144 (0x040000)
   MonU0     1889.76V  (0xbef0)
   MonU1       10.89V  (0xffa0)
   MonU2     1889.76V  (0xbef0)
   MonI     -195.97uA (0x09d0)
   Temp        30.63C  (0x01ea)
Voltage Rising Ratio: 0.999338

Last calibrated as Type 3500Volt on: Wed Jun 11 02:16:10 2008
```

Figure 11: Diag log file

6. Please email to SPECS, support@spesc.de, these log files to help diagnosis of the exact problem.
   Note: Particularly with analyzer connected, rise time errors are possible because of a resistance network in the analyzer box and capacitance effects. Especially the module for Ekin has a higher capacitance to supply and may cause rise time errors even if the module is okay. Values of 0.8 are within the limit.

2.3.2 Connection check for the analyzer electrodes

While performing the checks below, it may be useful to refer to the Appendix, section 3.3 for the pin assignments of the PHOIBOS feedthroughs.

2.3.2.1 Check all analyzer (HSA) voltages

Check whether the electrode voltages are present at the female contacts of the spectrometer plugs. Check the accuracy of the voltage ranges (40V, 400V) with a high precision digital voltmeter; for the high voltage ranges measurements (1500V, 3500V) a high voltage probe (1000:1) should be used to avoid an excessively high current load (-> voltage drop).

Mind the safety instructions given on page 8

Note the influence of the workfunction (the spectrometer voltage U_0 is nominal equal to *Ekin - Ep + workfunction*).

2.3.2.2 Fast Check of Voltage Ranges (energy scale).

Start a measurement in fixed energy mode or use the HSA3500 Juggler program (see SpecsLab2 manual) to check the voltages at the ground plate (pin 9 of the analyzer connector, figure 24, Schematics of the 12-pin analyzer feedthrough, page 65).

Range (V)	Ek	Ep	PIN 9	Correct with analyzer settings
40	0	0	0	offset 40V range
40	40	1	-39	gain 40V range
400	0	0	0	offset 400V range
400	400	10	-390	gain 400V range

Table 10: *Voltage (energy scale) check for 40V and 400V range*

Range(V)	Ek	Ep	PIN 9	Correct with analyzer settings
1500	0	0	0	offset 1500V range
1500	1500	50	-1450	gain 1500V range
3500	0	0	0	offset 3500V range
3500	3500	100	-3400	gain 3500V range

Table 11: Voltage (energy scale) check for 1500V and 3500V range (with high voltage probe 1000:1)

2.3.2.3 Check/set offset and gain with voltmeter

Adjustment and calibration of the power suppliy has been performed at the factory. Normally no additional work is necessary after installation. The procedures described in this chapter are only necessary for service and/or fine tuning.

To calibrate the voltage modules with a high voltage probe (1000:1) and a voltmeter, you have to set the kinetic energy and the workfunction via software (HSA3500 Juggler or FixedEnergies mode in SpecsLab2). For the offset calibration set all to zero (do not forget the work function). For the gain calibration please set as close to the end of the module range as possible.

1. Warm up the HSA for at least 30min. Switch off and on again to reboot and force a startup calibration.
2. Use the HSA3500 Juggler program to set all voltages or define a region in SpecsLab2 with the analyzer mode FixedEnergies and a sufficient number of values (the product of values multiplied with Dwell Time defines the time for the measurement in this FixedEnergies mode!)
3. Set the workfunction in the region to zero.
4. Set the kinetic energy and pass energy.
5. Set the multiplier voltage to 0 V (for your safety!).
6. Run a spectrum in FixedEnergies mode or apply the setting with the Juggler.
7. Check the spectrometer output voltage U_0 of the HSA (analyzer box: pin9 figure 24, page 65) with $U_0 = E_{kin} - E_p + WF$. For the high voltage ranges 1500V and 3500V, use the HV probe.
8. Change the offset value for the selected energy range in the "Analyzer/Settings" menu.
9. Depending on the software version the HSA can reboot after confirmation of the settings (versions before September 2003, include startup calibration). This needs a few seconds and ends with a beep of the unit.
10. Repeat the procedure for each desired voltage range.
11. Repeat the procedure (except item 1) with an energy value close to the end of the module range to correct the gain in each desired voltage range (see examples in table 10 and table 11 on page 19).
12. The gain is defined as $gain = \frac{required\ value}{measured\ value}$.

2.3.2.4 Fast check of all applied voltages

First run the "Diagnostic tool" as described on page 15. This tool needs no special equipment and performs a complete check of all generated voltages in the modules via software.

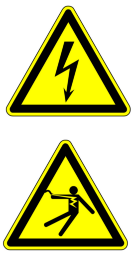
To check the voltage modules with a high voltage probe (1000:1) and a digital voltmeter, set the kinetic energy and the workfunction via software (HSA3500 Juggler or FixedEnergies mode in SPECSLAB 2).

The table 12 on page 24 shows an example for EKin=800eV and Ep=20eV in FAT mode for XPS energy range, the table 13 shows the same for UPS energies.

Tables for former releases can be found in the document "Check-PHOIBOSRx.pdf" which are supplied by SPECS upon request.

If the measured values are correct, the HSA unit as well as the analyzer box (plug) and cables are o.k.

Now the voltages can again measured either without the filterbox cable or while being applied to the analyzer.



The next two procedures should be only carried out by well trained personnel. Only touch electrical contacts with the HV-probe.

1.) To exclude the analyzer connection box and the cable, repeat the test without these connection parts.

   Measure the voltages at the rear of the HSA ("Voltages to ground for some modi (WF = 0), PHOIBOS 100/150 R6 or higher." on page 26). Disconnect the analyzer box at the rear panel of the control unit. Set a short circuit bridge between pin C1 and pin C2 at the analyzer output socket (see Appendix section 3.2, Rear panel wiring of the HSA, and be sure to use the diagram that corresponds to your model's wiring type). This bridges the safety interlock if the analyzer is not connected and enables HV output.

2.) To be able to measure the voltages while applying them to the analyzer the following procedure is required.

1. Turn the HSA off and wait for 10 minutes so that all capacitors are properly discharged.
2. Open the four screws of the HSA backplane and place the backplane so that it will not accidentally drop down.
3. Now you should see something similar as in Figure 15on page 59.
4. From now on you should not touch the power supply from the inside anymore. Contact should be only made via the HV-probe.
5. Start the HSA.
6. Start the HSA3500 Juggler and set the lens mode to Large Area and Ek to 800eV and Ep to 20eV, remember to leave the Workfunction(WF) at zero.
7. Click “On” and then “Start”.
8. Measure the voltages with the HV-probe and compare them to the values in Table 12 on page 24.
9. After measuring all voltages, set the analyzer to the state “Off” in the HSA3500 Juggler, close the Juggler and turn the HSA off.
10. Wait for a few minutes.
11. Remount the backplane.

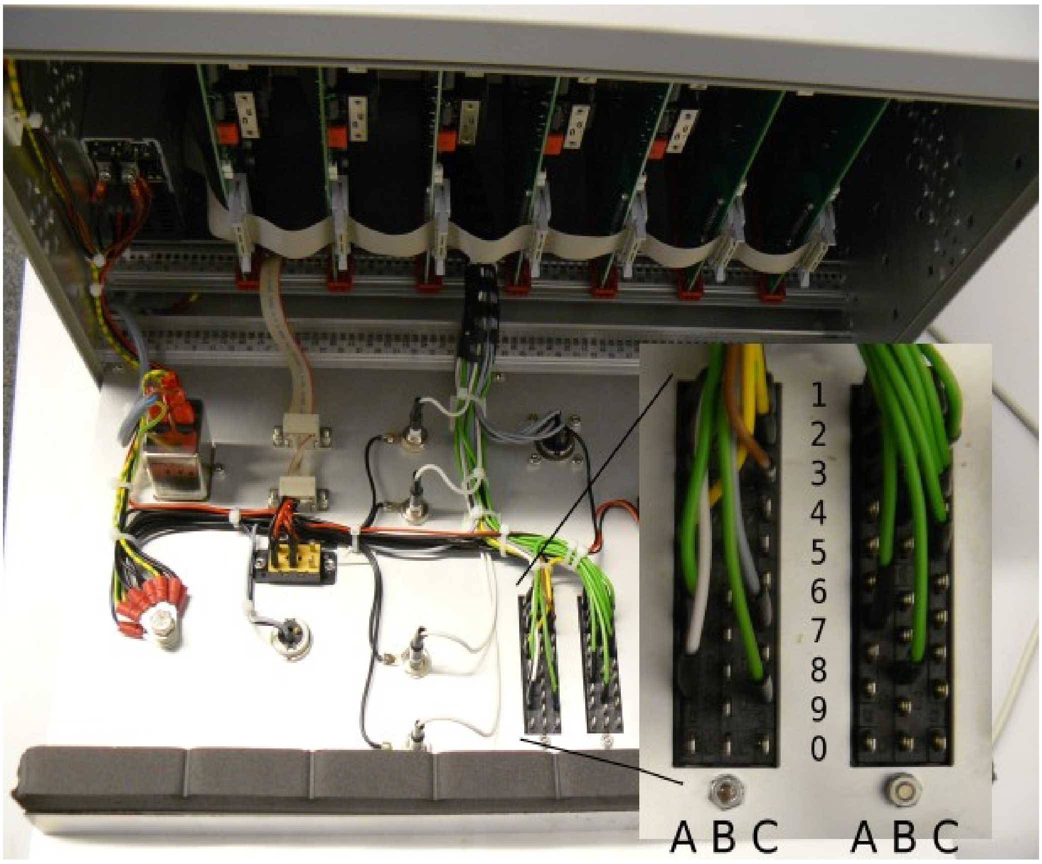


Figure 12: HSA3500 plus with opened backplane, the pin numbers are indicated in the zoomed figure.

Physical Meaning	PIN (12-pin)	PIN (30-pin)	High Magnification	Medium Magnification	Low Magnification	Large Area
IHP	1	Analyzer-C3	-766.7	-766.7	-766.7	-766.7
T2	2	Lens-B4	-660.2	-751.7	-556.8	-708
T3	3	Lens-A3	-747.4	-456.9	58	-516
OHP	4	Analyzer-C7	-788	-788	-788	-788.0
T4 - T7	5	Lens-C5	-719.9	-757.9	-720.2	-759.4
Not used	6	not used	-	-	-	-
T8	7	Lens-B8	-606.1	-679.6	-626.1	-559.4
T9	8	Lens-A7	-780	-780	-780	-536
GP	9	Analyzer-A9	-780	-780	-780	-780.0
IH	10	Analyzer-A5	-766.7	-766.7	-766.7	-766.7
OH	11	Analyzer-C9	-788	-788	-788	-780.0
Coil	12	Lens-C3	-	-	-	-

Table 12: Voltages to Ground (Ekin = 800eV, Ep = 20 eV, WF=0), PHOIBOS 100/150 R6 or higher, for typical XPS applications

Physical Meaning	PIN (12-pin)	PIN (30-pin)	High Magnification 2	Low Angular Dispersion	Wide Angle Mode
IHP	1	Analyzer-C3	-3,33	-3,33	-3,33
T2	2	Lens-B4	102,6	85	57,5
T3	3	Lens-A3	33,8	10	235
OHP	4	Analyzer-C7	-14	-14	-14
T4 - T7	5	Lens-C5	270	85	130
Not used	6	not used	-	-	-
T8	7	Lens-B8	-9,5	9	50
T9	8	Lens-A7	-10	-11	-5
GP	9	Analyzer-A9	-10	-10	-10
IH	10	Analyzer-A5	-3,33	-3.33	-3,33
OH	11	Analyzer-C9	-14	-14	-14
Coil	12	Lens-C3	-	-	-

Table 13: Voltages to Ground (Ekin = 20eV, Ep= 10 eV, WF=0), PHOIBOS 100/150 R6 or higher, for typical UPS applications

T1 - T10	Tube numbering
IH / OH	Inner / Outer hemisphere
IHP / OHP	Inner and Outer Jost-Plate
Ep / Ek	Pass energy / kinetic energy (the workfunction should be set to zero to set the ground plate potential to the difference between the kinetic energy and the pass energy)
Coil Current	PIN 12 can be used to feed the current to the trim coil of the outer hemisphere (for details see the chapter Magnetic Shielding in the Phoibos-Manual).
RR	Retardation Ratio: Ek/Ep

Table 14: Abbreviations / Terminology

The following hint is only needed if the above table with fixed kinetic energy and pass energy does not fit your needs.

2.3.3 Checking the Analyzer Voltages with Custom Setttings

The SpecsLab2 software suite also includes a small batch file called “getVoltages.bat” located in the tools.bin directrory, this is in the same place located as the diag tool. This program allows to output the voltages to ground for all possible combinatins of Ep,Ek and detector voltages for all lens modes.

To use this script a installed version of the GNU AWK Interpreter is necessary. This can be installed from the SPECSLab2 CD or can be downloaded from http://gnuwin32.sourceforge.net/packages/gawk.html

Open getVoltages.bat in your favourite editor and adjust at least kinetic energy, pass energy and lens mode. Then call the script “getVoltages.bat” by double clicking it from the explorer. The script outputs the voltages at each PIN of the 12-Pin feed-through and also the detector voltages.

Note: The Screen voltage is only available from the power supply for CCD and DLD detectors.

Chapter 3 HSA config files (*.hsa)

3.1 What are HSA3500 Configuration Files?

For the analyzer power supply the (*.hsa) configuration file specifies in addition to the power supply wiring, the individual DAC voltages in terms of relationships between logical voltages within the analyzer (Kinetic Energy, Pass Energy, ...) and assignments of logical voltages to DAC voltages. Since the HSA configuration files define these mappings, the functionality of the analyzer can be changed without recompiling the control software.

The backplane of the unit incorporates plug connectors which permit various interconnection topologies among the different modules. Each HSA power supply is manufactured with a fixed combination of modules and backplane interconnectivity. Each combination constitutes a hardware type.

Each hardware type can be configured to function in one of a number of ways depending on how switchable outputs, voltage polarity and levels are set. These settings are communicated to each card's control processor according to the specifications which the main CPU extracts from the configuration file.

3.2 Nomenclature

In this chapter we use the following terms to describe the HSA3500 (plus) Power Supply.

3.2.1 Mode

1. An HSA3500 mode corresponds to an interconnection configuration and particular voltage level and polarity settings for each card.
2. A voltage or energy which can be set by the control software is called a logical voltage. Each logical voltage must be defined before it can be used in a HSA3500 mode definition. The range and the step size of the logical voltages can be different for different mode definitions. Voltages with fixed values within the configuration file can not be set by the control software.

3.2.2 DACVoltage

The various voltages are supplied by a HSA3500 HV module (via a digital to analog converter). The output voltages results from the DAC voltages modified by the back plane interconnections. All DAC voltages must be defined in a HSA3500 mode.

3.2.3 Natural Cubic Spline

A spline is a series of cubic polynomials. Each cubic polynomial goes through a set of consecutive values. The cubic polynomials are adjoined continuously and smooth. If the second derivatives at both endpoints are zero the spline is called a natural cubic spline.

Natural cubic splines are used in the HSA3500 configuration file to describe nonlinear relations between logical voltages and DAC voltages.

3.2.4 Syntax and Encoding

A HSA3500 configuration file is an ASCII text file. Each file contains a fixed header and a sequence of sections. Comments may appear throughout the file.

3.2.5 Comments

The start of a comment is marked by the character '#'. Any text between the '#' and the end of line is ignored by the configuration software. The character '#' may not be used elsewhere in the configuration file, **not even within strings**.

3.2.6 Header

The very first line of the header "#!hsa" identifies the file as a HSA3500 configuration file. The following lines contain information about the name of the configuration file and its version. The name doesn't need to match the file name and is used only to identify the configuration file to the user. The version number is tested by the control software to support changes in the HSA3500 configuration format.

The header may also contain comments in any line after the first line. A HSA3500 configuration file header looks like:

```
#!hsa
# Header Comment
Name = <configuration name>
# Version Comment
Version = <version number>
```

3.2.7 Sections

Sections are the basic information containers in the HSA configuration file. Each section has a type describing which kind of information the section contains and must have a unique name. Sections contain entries with information and may contain subsections.

A section may have an empty name. In this case, the parser will associate the section with an unique name and the section can not be referenced within the configuration file.

The order of the sections in the configuration file is not relevant, but a section reference can be only made to named sections which appear earlier in the file.

Sections looks like:

```
# First  Comment Position
[<section type>] { <section name> { = <section reference> }}  # Second Comment Pos
# Third Comment Position
{  # Fourth Comment Position
   <Entries>
   [<subsection type>] { <subsection name> { = <subsection reference> }}
```

```
}
```

If a section has an empty body (because it is inherited or all entries have default values), the brackets “{}” can be omitted.

3.2.8 Entries

An entry is a single line of information contained in a section. Each entry has a section wide unique name, a type and a value. The type of an entry determines the possible values.

Type	Example
string	exp(x), $Ekin
boolean	WAHR
unsigned long	4
unsigned long sequence	1,2,3,4,5,6
double	0,63
double sequence	–0.0098,-0.0043,0.0001,0.0032,0.00076
point	(12.00,34.08)
point sequence	((1.0,1.1),(2.0,2.21),(3.0,3.32),(4.0,4.43),(5.0,5.54))
bit masks	0x00,0x34,0x45,0x66,0x07
section reference	/AnalyzerMacros/Groundplate

Table 15: Supported types

The section type also determines whether an entry is mandatory or optional, and whether it has a default value.

A section may not contain more than one entry with a given name. Examples of entries are:

```
ShortName = “Ekin”   # string value
Unit      = V        # string value, where quotes are omitted, the value may
                     # not contain preceding or following blanks
DacId = 2.1          # also a string value (split into 2 long values after pars-
ing)
OperationMode = 5    # unsigned long value
```

3.2.9 Inheritance

In order to minimize redundancy, sections can be inherited from previously defined sections of the same type. In this case all entries and subsections of the child section are initialized with the values from the referenced section. This initialization can be overridden in the new section body.

Inheritance is declared by giving a section reference after the section name in the section declaration:

```
[ModeDef] LargeArea:1.5kV = LargeArea:3.5kV
{
   ...   # the section body may be used to override some inherited values or sec-
tions
}
```

If a section inherits from another section, any default values are taken from the referenced section rather then from global defaults.

3.2.10 Sections and Layout

The following section names have special properties and can only exist once in the HSA3500 configuration file. Typical inheritance functionality therefore does not apply to them.

1. [Info]
2. [SplineList]
3. [LogicalVoltageList]
4. [ModeList]
5. [SplineDef]
6. [LogicalVoltageDef]
7. [MacroDef]
8. [ModeDef]

3.2.10.1 Info section

This section contains general information.

An example for an [Info] section is:

```
[Info] HSA3500                            # the section name is empty
{
   HardwareType = 1
   HardwareName = HSA3500 CCD with deflector # only for information
   FileFormat   = 5
   ModeParts    = "Lens Mode:Range"    # standard mode name schema
}
```

Entry	Type	Description
HardwareType	unsigned long, mandatory	Used to validate that the configuration file matches the actual HSA hardware
HardwareName	string, optional	A hardware describing text, for information purposes only
FileFormat	unsigned long, mandatory	File format number, current value is 5
ModeParts	string, optional	Format: <part-1>:<part2->:...:<part-n> describing of what parts a mode name is composed of. Default value is "Lens Mode:Range"

Table 16: Info section description

3.2.10.2 SplineList Section

This section contains a sequence of all spline definitions in the configuration file. Each element of the sequence is a [Spline] subsection without a body.

An example for a [SplineList] section is:

```
[SplineList] GlobalSplines  # list of all global defined splines
{
  [Spline] HighMagnification_T2
  [Spline] HighMagnification_T3
  [Spline] HighMagnification_T4_T7
  [Spline] HighMagnification_T8
  [Spline] HighMagnification_T9
   ...
}
```

Section	Type	Description
Spline	string, optional, no body	Name of a spline definition

Table 17: SplineList section description

3.2.10.3 LogicalVoltageList Section

This section contains a sequence of all logical voltages defined in the configuration file. Each element of the sequence is a [LogicalVoltage] subsection without a body. This list determines the order of the logical voltages dialogue elements within the user interface.

An example for a [LogicalVoltage] section is:

```
[LogicalVoltageList] LogicalVoltages   # the section name is empty
{
   [LogicalVoltage] "Kinetic Energy"
   [LogicalVoltage] "Pass Energy"
   [LogicalVoltage] "Detector Voltage"
   [LogicalVoltage] "Conversion Voltage"

}
```

Section	Type	Description
LogicalVoltage	Multiple, optional, no body	Name of a LogicalVoltage definition

Table 18: LogicalVoltageList section description

3.2.10.4 ModeList Section

This section defines the modes which will be available to the user control software. It contains a sequence of [Mode] subsections. Any mode defined in the configuration file but not contained in this list can be referenced within the file only. The order of apperance in this sequence determines the order in the mode selection combo boxes of the control software.

This list makes it easy to disable a HSA3500 mode temporarily without removing its definition from the configuration file.

An example for a [ModeList] section is:

```
[ModeList] ExternalModes  # list of all external modes
{
  [Mode] HighMagnification:3.5kV
  [Mode] MediumMagnification:3.5kV
  [Mode] LowMagnification:3.5kV
  [Mode] HighAngularDispersion:3.5kV
  [Mode] MediumAngularDispersion:3.5kV
  [Mode] LowAngularDispersion:3.5kV
  [Mode] WideAngleMode:3.5kV
  [Mode] LargeArea:3.5kV
  [Mode] MediumArea:3.5kV
  [Mode] SmallArea:3.5kV
  [Mode] HighMagnification2:3.5kV
  [Mode] SmallArea2:3.5kV
  [Mode] ImagingMode:3.5kV
```

```
  [Mode] HighMagnification:1.5kV
...
}
```

Section	Type	Description
Mode	string, optional, no body	Name of a HSA3500 mode definition

Table 19: ModeList section description

3.2.10.5 SplineDef section

Sections of this type define global splines, witch can be referenced within DAC voltage definitions later in the configuration file. This section has a single entry with name “points”, its value is a sequence of the spline grid points. The sequence can have any length, but must be written as a single line of text.

An example for a [SplineDef] section is:

```
[SplineDef] MyLensMode_T2
{
   Points = (1,0.6),(5,3.0),(7,3.78),(10,4.5),(20,6.0),(100,7.3),(150,6.0)
}
```

Entry	Type	Description
Points	point sequence, mandatory	A list of RR-NL pairs, where RR is the retarding ratio (Ekin – WF)/Ep and NL is a nonlinear factor used in the lens voltages definitions. The spline will be called as MyLensMode_T2(-(-$Ekin-$WF)/$Ep) and would return an interpolated NL-value.

Table 20: SplineDef section description

3.2.10.6 LogicalVoltageDef Sections

A section of this type defines a logical voltage, which can be set up by the control software. A defined logical voltage is valid for any HSA3500 mode, but some of the [LogicalVoltageDef] section entries may be overridden with new values in a HSA3500 mode definition.

An example for a [LogicalVoltageDef] section is:

```
[LogicalVoltageDef] "Kinetic Energy"
{
  ShortName = Ekin
```

```
    Unit = eV
    AdjustOnly = false
    AnalyzerOnly = true
    MappedToDacId = 9.0          # this means HSA3500 slot no 9 DAC no 0
    PolarityFlip = true
    StandbyValue = 0.0
    StandbyGroup = 1
} # Step – must not defined, the software can calculate this by “MappedToDacId”
```

Entry	Type	Description
ShortName	string, mandatory	Gives a short name for use in the control software user interface .
Unit	string, optional	Unit of the logical voltage (Types are: V, eV, mA or nu meaning no unit).
AdjustOnly	boolean, mandatory	Signals that this logical voltage is used as an offset to some DAC voltages for adjustment and optimization only. If set to true, this logical voltage is set to 0.0 and ignored by the control software main program. Default: false
AnalyzerOnly	boolean, mandatory	Indicates that this logical voltage is correlated to a basic Phoibos analyzer parameter and cannot be used by other devices (e.g. the detector). Default: false
PolarityFlip	boolean, mandatory	Set to true if the sign of the logical voltage must be switched internally for ion operation. Default: false
MappedToDacId	string, optional	Associates the logical voltage directly with a HSA DAC output. If this entry is present, the value of the ‘Step’ entry can be calculated from the DAC parameter by the software. The string must have the format “<hsa no.><slot no>.<DAC.no>” and can be overridden in a [ModeDef] section. See also the description to DacId on page 39
MinValue	double, mandatory	Minimal value for this logical voltage used to check input data in control software user interface. Limitation of the DAC voltage calculation may restrict the possible value much more strongly than this value. Default: 0.0, can be overridden in a [ModeDef] section
MaxValue	double, mandatory	Maximal value for this logical voltage, see MinValue above. default: 0.0, can be overridden in a [ModeDef]

Entry	Type	Description
		section
MinValuePP	double, optional	Same as MaxValue, but used in the ion operation mode (positive polarity). Can be overridden in a [ModeDef] section
MaxValuePP	double, optional	Same as MaxValue, but used in the ion operation mode (positive polarity). Can be overridden in a [ModeDef] section
Step	double, mandatory	Minimal step width the logical voltage can be changed. The value depends on the DAC resolution, the DAC voltage calculation and the hardware wiring. default: 0.0, can be overridden in a [ModeDef] section

Table 21: LogicalVoltageDef section description

3.2.10.7 Macro Sections

This section has no entries and contains only prototypes to be used in [ModeDef] sections by reference. For following [ModeDef] subsections a [Macro] section can contain protopypes:

Entry	Type	Description
LogicalVoltage	optional, multiple	Can be referenced from [LogicalVoltage] subsection definition in a [ModeDef] section
DacVoltage	optional, multiple	Can be referenced from [DacVoltage] subsection definition in a [ModeDef] section
SlotHVOff	optional, multiple	Can be referenced from [SlotHVOff] subsection definition in a [ModeDef] section
Spline	optional, multiple	Can be referenced from [Spline] subsection definition in a [ModeDef] section

An example for a [Macro] section is:

```
[[Macro] AnalyzerVoltages
{
      [DacVoltage] MacroGroundPlate
          {
                 DacId = 11.1
                 MaxStep = 45.0
                 OperationMode = 2
                 Formula = 0
          }
...
```

```
}
```

3.2.10.8 ModeDef Sections

The configuration file must contain a [ModeDef] section for each HSA3500 mode (this is typically a lens mode in a special voltage range). This section contains all information required by the software to activate and operate a HSA3500 mode. The most essential data are the calculation formulas for all DAC voltages used in a mode, as described in the [DacVoltage] subsections. All these sections and subsections can be declared by inheritance.

Entry	Type	Description
ShortName	string,optional	Short name for the HSA3500 mode for use in the control software user interace.
MinRR	double, optional	Minimum value for the retardation ratio (Ekin - WF)/Ep in this mode. Used to determine which logical voltage settings are vaild. Default: -1.0, ignored for negative values
MaxRR	double, optional	Maximum value for the retardation ratio (Ekin/Ep) in this mode. Used to determine which logical voltage settings are valid. Default: -1.0, ignored for negative values
Hidden	optional	If set, this mode is disabled by the control software software until it is explicitly enabled in the program options. Default: false
LogicalVoltage	optional, multiple	Describe the values of a global defined logical voltage are overridden in this HSA3500 mode (for details see below).
SlotHVOff	optional, multiple	In some modes an existing HSA3500 power supply is not used to generate any output voltage, but its relays must be configured. This is done by this sections. This is also the only way to delete a DAC voltage definition inherited by a parent [ModeDef] section (for details see below).
DacVoltage	optional, multiple	Describe a DAC voltage generated by the HSA3500 in this mode (for details see below).

Table 22: ModeDef section context

Entry	Type	Description
MappedToDacId	string, optional	Overrides the value of the 'MappedToDacId' entry in a global [LogicalVoltageDef] section with the same name for this HSA3500 mode
MinValue	double, optional	Overrides the value of the 'MinValue' entry in a global [LogicalVoltageDef] section with the same name for this HSA3500 mode
MaxValue	double, optional	Overrides the value of the 'MaxValue' entry in a global [LogicalVoltageDef] section with the same name for this HSA3500 mode
MinValuePP	double, optional	Overrides the value of the 'MinValuePP' entry in a global [LogicalVoltageDef] section with the same name for this HSA3500 mode
MaxValuePP	double, optional	Overrides the value of the 'MaxValuePP' entry in a global [LogicalVoltageDef] section with the same name for this HSA3500 mode
Step	double, optional	Override the value of the 'Step' entry in a global [LogicalVoltageDef] section with the same name for this HSA3500 mode

Table 23: LogicalVoltage subsection in a ModeDef section

Entry	Type	Description
Slot	unsigned long, optional	Number of the HSA3500 slot containing a HSA35000 power supply module, which should not generate any DAC voltage in this mode. If there exists a "DacVoltage" definition with a name matching the first part of the "SlotHVOff" definition, their slot number is taken over.
OperationMode	unsigned long, optional	Operation mode of the unused HSA3500 power supply, necessary if the module contain relays needed in this mode. If there exist no "DacVoltage" definition with a name matching the first part of the "SlotHVOff" definition this entry is mandatory.

Table 24: SlotHVOff subsection in a ModeDef section

Entry	Type	Description
DacId	string, optional	Identifies the DAC generating the DAC voltage. The string must have the format “<HSA no.>.<SLOT no.>.<DAC.no.>”. - <HSA no.> is the number of the HSA and can be left out in case of only one HSA. The HSA numbers are defined in [Info]Hsa3500->HardwareType. - <SLOT-no.> is the number of the HSA3500 slot containing the power supply module (0 .. 13) - <DAC-no.> is the number of the DAC on the module (0 for 3.5kV modules and current modules, 0..2 for 400V modules).
OperationMode	unsigned long, optional	Operation mode of the HSA3500 power supply module. This value has to be given in hexadecimal format, i. e. prefixed with 0x. If the [ModeDef] section contains more than one DAC voltage for the same HSA3500 module (= slot) the effective operation mode is calculated by a bitwise OR-operation of all values. For possible values check the corresponding table on page 48 and following. Default: 0x00
Formula	string, optional	String with the name of the formula used to calculate the DAC voltage. The formula can consist of the following different parts: - integers and floats - references to logical voltages by their shortname prefixed with $, e. g. $Ekin. - the following fuctions are supported , where x and y denote the parameters (the trigonometric functions expect their arguments to be in radians): pi (the constant 3.1415..); div(x,y); mod(x,y); sin(x); cos(x); tan(x); sqrt(x); ln(x); log(x); exp(x); asin(x); acos(x); atan(x); abs(x); sign(x); min(x,y); max(x,y); mean(x,y) - Offset and Gain values. References to Offset and Gain have to be used like: &Offset[0]. They have to be indexed with values as defined in [Info] Hsa3500->HardwareType. In the normal case of one HSA this index can only be zero. For multiple HSAs, e. g. HardwareType = 0:60,1:61,2:62,3:501 the index is the suffixed by the HSA macId. They are saved in the HSA hardware and can be different for each voltage range. - Splines as defined in [SplineList] and [SplineDef] sections, they have to be prefixed with spline_, e. g. spline_HighMagnification_T8
MaxStep	double, optional	If this entry exists and is greater than 0.0, it gives the maximal voltage step for a single DAC output operation. E.g. if the DAC output is 50V, MaxStep is set to 10V and the control software will change the output to 200V, then this is done by 15 steps of

		10V: 50V,60V,...,200V. If there are any other DAC voltages to change at the same time, these voltages will also be changed in 15 equal steps, even if their MaxStep limit is deactivated. Default: 0.0 (deactivated if less or equal 0.0)
DelayUp	unsigned long, optional	If this entry exists and is greater than 0, it gives the minimal delay between two DAC output operations for increasing output voltages (|U1| < |U2|). default: 0 (ignored if 0) Units: ms
DelayDown	unsigned long, optional	If this entry exists and is greater than 0, it gives the minimal delay between two DAC output operations for decreasing output voltages (|U2| < |U1|). Default: 0 (ignored if 0) Units: ms
StandByValue	string	Value of DAC Voltages while the Unit is in standby. Can be given as an absolute (e.g. 1000) or relative value (e.g. 20%).
StandByValuePP	string	Same as above for positive polarity (detection of ions).

Table 25: ModeDef section description

An example for a [ModeDef] section is:

```
[ModeDef] LargeArea:3.5kV ###############################################
{
  MinRR =  0.0
  MaxRR = 1000.0
  ShortName = LA:3.5kV
  RelaisMode = (0,0)                        # no relays used
  [Parameter] Ekin
  {
     Formula = $Ekin
     Unit    = eV
  }
  [Parameter] Epass
  {
     Formula = $Ep
     Unit    = eV
  }
  [Parameter] NonDisp
  {
     Formula = %1/10 + %2
     Unit    = mm
  }
  [LogicalVoltage] "Kinetic Energy"
```

```
{
  MinValue =    -0.0
  MaxValue = +3500.0
}
[LogicalVoltage] "Pass Energy"
{
  MinValue =   -0.0
  MaxValue = +660.0
}
[LogicalVoltage] "Focus Displacement 1"
{
  MinValue = -1
  MaxValue = +1
}
[LogicalVoltage] "Focus Displacement 2"
{
  MinValue = -1
  MaxValue = +1
}
[DacVoltage] CoilCurrent
{
  DacId = 16.0
  OperationMode = 0
  Formula = $Icoil * 0.001    # User Interface Unit = mA
}
[DacVoltage] NonLinearLens1
{
  DacId = 1.0
  OperationMode = 3
  Formula = -$Ekin+$WF+ $Ep * spline_LargeArea_T2(-(-$Ekin + $WF) / $Ep)
}
[DacVoltage] NonLinearLens2
{
  DacId = 3.0
  OperationMode = 3
  Formula = -$Ekin+$WF+ $Ep * spline_LargeArea_T3(-(-$Ekin + $WF) / $Ep)
}
[DacVoltage] Lens3
{
  DacId = 5.0
  OperationMode = 3
  Formula = -$Ekin+$WF+ $Ep * spline_LargeArea_T4_T7(-(-$Ekin + $WF) / $Ep)
}
[DacVoltage] Lens4
{
```

```
   DacId = 7.0
   OperationMode = 3
   Formula = -$Ekin+$WF+ $Ep * spline_LargeArea_T8(-(-$Ekin + $WF) / $Ep)
  }
  [DacVoltage] Lens5
  {
    DacId = 6.0
    OperationMode = 3
    Formula = -$Ekin+$WF+ $Ep * spline_LargeArea_T9(-(-$Ekin + $WF) / $Ep)
  }
  [DacVoltage] EkBase
  {
    DacId = 9.0
    MaxStep = 45.0
    OperationMode = 3
    Formula =  -(&Offset[0] + &Gain[0] * $Ekin) + $Ep + $WF          # + Ekin +
Epass by hardware
  }
  [DacVoltage] OuterHemisphere      = AnalyzerVoltages/MacroOuterHemisphere
  [DacVoltage] GroundPlate          = AnalyzerVoltages/MacroGroundPlate
  [DacVoltage] InnerHemisphere      = AnalyzerVoltages/MacroInnerHemisphere
  [DacVoltage] OuterHemispherePlate      = AnalyzerVoltages/MacroOuterHemisphere-
Plate
  [DacVoltage] InnerHemispherePlate      = AnalyzerVoltages/MacroInnerHemisphere-
Plate

  [DacVoltage] ConversionVoltage
  {
    DacId = 0.0
    MaxStep = 45.0
    OperationMode = 3
    Formula = -$Ekin + $Uconv + $Ep + $WF
  }
  [DacVoltage] DetectorVoltage
  {
    DacId = 2.0
    MaxStep = 45.0
    OperationMode = 3
    Formula = $Udet
  }
  [DacVoltage] ScreenVoltage
  {
    DacId = 4.0
    MaxStep = 0.0
    OperationMode = 3
```

```
    Formula = $Uscr     # + Ekin + Epass + Edet + Econv by hardware
  }
}
```

3.2.11 Adding Custom Voltages

Sometimes it is desirable to add custom logical voltages to the config file.
The following example adds a bias voltage to lens tubus 2.

1.) Adjust to the LogicalVoltageList

```
[LogicalVoltageList] LogicalVoltages  # list of all logical voltages
{
  :
  [LogicalVoltage] "Adjust"
  :
}
```

2.) Define the LogicalVoltage in the block LogicalVoltageDef

```
[LogicalVoltageDef] "Adjust"
{
  ShortName = A
  Unit = eV
  AdjustOnly = false
  AnalyzerOnly = false
  MinValue =  -500.0
  MaxValue =  500.0
  StandbyValue = 0.0
  StandbyGroup = 1
}
```

3.) Add the voltage at the corresponding lens mode

```
[DacVoltage] NonLinearLens1
 {
 :
   Formula =  ... + $A
 :
```

```
}
```

4.) After changing the config file you should check it with the program ConfigCheck as described in the next chapter.

3.3 ConfigCheck

One can check the syntax of a configuration file with the program “ConfigCheck”, which is included in the SpecsLab software package.

1. Start ConfigCheck found in the SPECS folder.
2. Load the appropriate config file. The name of the config file depends on your hard- and software and is only known at runtime of the software. Establishing a connection to the HSA using SpecsLab2, the HSA3500Juggler, or external programs using the SL2_AD.dll outputs the used config file to internal windows debug facilities. These can be read out with the free software tool DebugView available from http://technet.microsoft.com/en-us/sysinternals/bb896647.aspx Now open DebugView and connect to the HSA and look for the following line:
   000:00:02:023 0x0460 MAJOREVENT [HSA3500Analyzer::CCD-Phoibos-Hsa3500@HSA3500Analyzer] loading HSA configuration from "..\config\Hsa3500CCDConfig.123.hsa"
   where 123 denotes the hardware type of the HSA.
3. Check the config file without being connected.
4. If no error messages are printed, try to connect to the hardware by pressing "Connect". Now also no error messages should appear.

Note: ConfigCheck can not check for user input errors (wrong spline definitions, wrong forumulas, etc.). It can only check the syntax of the config file and that all the correct modules and DACs are referenced.

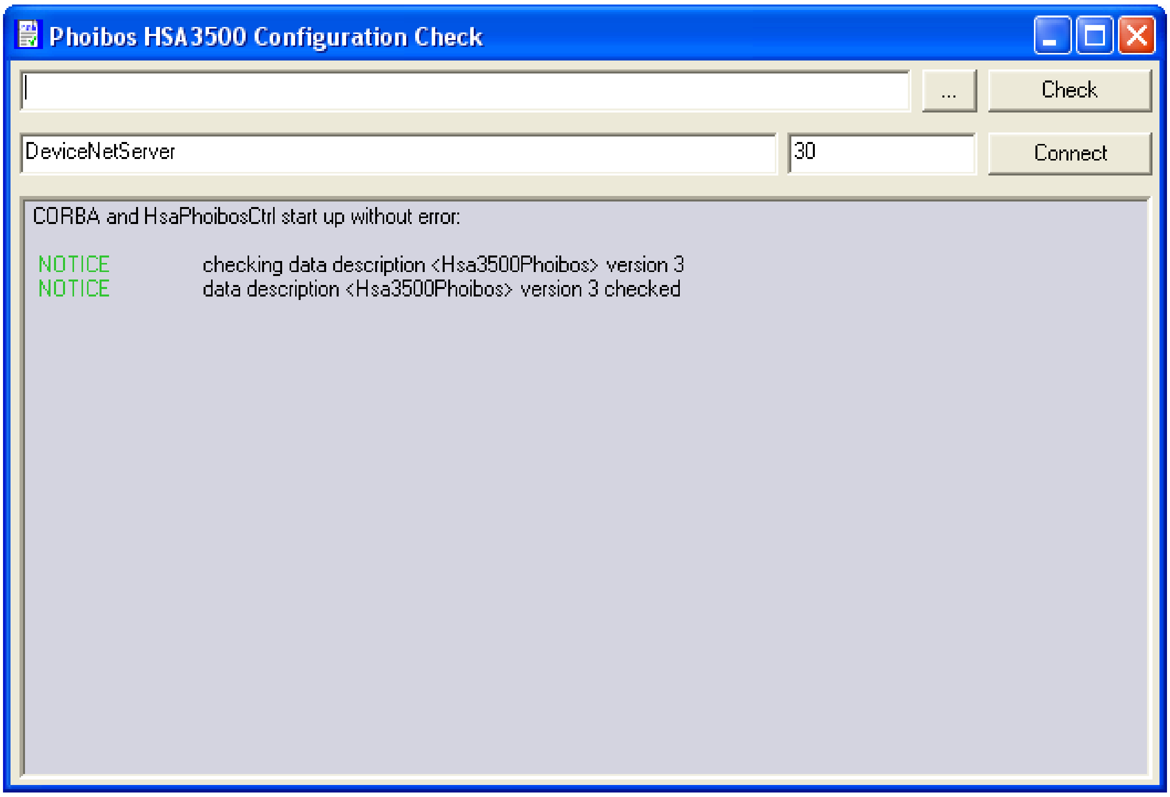


Figure 13: ConfigCheck startup screen

Chapter 4 Appendix

4.1 HSA modules

Every output of the module can be measured by the internal ADC. It is possible to measure the voltages and the total current from every module without a load. The ADC resolution is 12 Bit . The configuration for HSA modules is shown in following tables.

Tips:

- The RelayOut Bits have to be 1 so that the relay is closed and 0 for it to be open.
- Fast/Slow Bits are so encoded that 1 sets fast mode and 0 sets slow mode. Usually one wants to use the fast mode.
- For increased readability empty values in the operation mode tables denote binary 0.
- Bits which value do not matter are denoted by “x”.

Note: HSA3500 and HSA3500 plus do not have the identical ADC channels and can also not have same outputs at the connector. So please be careful and pay attention.

4.1.1 0xf1 – 3.5kV bipolar (HSA3500 module)

Channel	ADC name	Description	DAC Output Pin
0	N / A	N / A	N / A
1	Mon U0	Voltage Output 0	26
2	Mon U1	Relay Output 1	20
3	Mon U2	Relay Output 2	14
4	N / A	N / A	N / A
5	N / A	N / A	N / A
6	Mon I	Total current	
7	N / A	N / A	N / A

Table 26: HSA3500 0xf1 ADC mapping

Voltage range	Q2 Relay Out 2	Q1 Relay Out 1	Q0 Range	Hex value
-3500V to +3500V				0x00
		1		0x02
	1			0x04
	1	1		0x06
-1500V to +100V			1	0x01
		1	1	0x03
	1		1	0x05
	1	1	1	0x07

Table 27: HSA3500 0xf1 Operation Modes

4.1.2 0xe1 – 400V bipolar (HSA3500 module)

Channel	ADC name	Description	DAC Output Pin
0	N / A	N / A	N / A
1	Mon U0	Voltage Output 0	14
2	Mon I0	Current Output 0	14
3	Mon U1	Voltage Output 1	20
4	Mon I1	Current Output 1	20
5	Mon U2	Voltage Output 2	26
6	Mon I2	Current Output 2	26
7	N / A	N / A	N / A

Table 28: HSA3500 0xe1 ADC mapping

DAC	Voltage Range	Q5 Fast / Slow DAC2	Q4 Fast / Slow DAC1	Q3 Fast / Slow DAC0	Q2 Range DAC2	Q1 Range DAC1	Q0 Range DAC0	Hex value
0	-400V to +400V						1	0x01
				1			1	0x09
	-40V to +40V							0x00
				1				0x08
1	400V to +400V					1		0x02
			1			1		0x12
	-40V to +40V							0x00
			1					0x10
2	400V to +400V				1			0x04
		1			1			0x24
	-40V to +40V							0x00
		1						0x20

Table 29: HSA3500 0xe1 Operation Modes

4.1.3 0xd1 – 3.5kV unipolar (HSA3500 module)

Channel	ADC name	Description	DAC Output Pin
0	N / A	N / A	N / A
1	Mon U0	Voltage Output 0	26
2	Mon U1	Relay Output 1	20
3	Mon U2	Relay Output 2	14
4	N / A	N / A	N / A
5	N / A	N / A	N / A
6	Mon I	Total current	
7	N / A	N / A	N / A

Table 30: HSA3500 0xd1 ADC mapping

Voltage Range	Q2 Relay Out 2	Q1 Relay Out 1	Q0 Range	Hex
0V to +3500V			x	0x00
		1	x	0x02
	1		x	0x04
	1	1	x	0x06

Table 31: HSA3500 0xd1 Operation Modes

4.1.4 0xb1 – 150mA bipolar (HSA3500 module)

Channel	ADC name	Description	DAC Output Pin
0	N / A	N / A	N / A
1	Mon U	Output	26
2	Mon I	Output	26
3	N / A	N / A	N / A
4	N / A	N / A	N / A
5	N / A	N / A	N / A
6	N / A	N / A	N / A
7	N / A	N / A	N / A

Table 32: HSA3500 0xb1 ADC mapping

Note: No Operation Modes exist. The Operation Mode is defined by 0.

4.1.5 0x04 – 3.5kV Lens bipolar (HSA3500 plus module)

Channel	ADC name	Description	DAC Output Pin
0	Mon U1	Relay Output 1	20
1	Mon U2	Relay Output 2	14
2	Uerr		
3	Mon I0	Total current	
4	Mon U0	Voltage Output 0	26
5	N / A	N / A	N / A
6	N / A	N / A	N / A
7	N / A	N / A	N / A

Table 33: HSA3500 plus 0x04 ADC mapping

Module voltage range	Q3 Relay Out 2	Q2 Relay Out 1	Q1 Range 0	Q0 Range 1	Hex value
-3500V to +3500V			1	1	0x03
		1	1	1	0x07
	1		1	1	0x0B
	1	1	1	1	0x0F
-1500V to +1500V				1	0x01
		1		1	0x05
	1			1	0x09
	1	1		1	0x0D
-600V to +600V			1		0x02
		1	1		0x06
	1		1		0x0A
	1	1	1		0x0E
-150V to +150V					0x00
		1			0x04
	1				0x08
	1	1			0x0C

Table 34: HSA3500 plus 0x04 Operation Modes

4.1.6 0x14 – 3.5kV analyzer bipolar (HSA3500 plus module)

Channel	ADC name	Description	DAC Output Pin
0	Mon U1	Voltage Output 1	20
1	Mon U0	Voltage Output 0	26
2	Uerr		
3	Mon I	Total current	
4	Mon U2	Voltage Output 2	14
5	N / A	N / A	N / A
6	N / A	N / A	N / A
7	N / A	N / A	N / A

Table 35: HSA3500 plus 0x14 ADC mapping

Module voltage range	Q3 Relay Out 2	Q2 Relay Out 1	Q1 Range 0	Q0 Range 1	Hex value
-3500V to +3500V			1	1	0x03
		1	1	1	0x07
	1		1	1	0x0B
	1	1	1	1	0x0F
-1500V to +100V			1		0x02
		1	1		0x06
	1		1		0x0A
	1	1	1		0x0E

Table 36: HSA3500 plus 0x14 Operation Modes

4.1.7 0x24 – 400V bipolar (HSA3500 plus module)

Channel	ADC name	Description	DAC Output Pin
0	N / A	N / A	N / A
1	N / A	N / A	N / A
2	Mon U2	Voltage Output 2	14
3	Mon U1	Voltage Output 1	20
4	Mon U0	Voltage Output 0	26
5	Mon I0	Current Output 0	26
6	Mon I1	Current Output 1	20
7	Mon I2	Current Output 2	14

Table 37: HSA3500 plus 0x24 ADC mapping

DAC	Module voltage range	Q3 Range 10V	Q2 Range DAC0	Q1 Range DAC1	Q0 Range DAC2	Hex value
0	-400V to +400V				1	0x01
	-100V to +20V					0x00
1	400V to +400V			1		0x02
	-100V to +20V					0x00
2	400V to +400V		1			0x04
	-100V to +20V					0x00
0, 1, 2	-10V to +10V	1	x	x	x	0x08

Table 38: HSA3500 plus 0x24 Operation Modes

Notice:

If you use the 10V range, the high voltage stage at the module is off (voltage ripple is then very small). That means the range bits Q0, Q1, Q2 are don't care.

4.1.8 0x34 – 3.5kV unipolar (HSA3500 plus module)

Channel	ADC name	Description	DAC Output Pin
0	Mon U1	Voltage Output 1	20
1	N / A	N / A	N / A
2	Uerr		
3	Mon I	Total current	
4	Mon U0	Voltage Output 0	26
5	N / A	N / A	N / A
6	N / A	N / A	N / A
7	N / A	N / A	N / A

Table 39: HSA3500 plus 0x34 ADC mapping

Module	Q2 Relay Out 1	Q1 Range 0	Q0 Range 1	Hex
0V to +3500V		x	x	0x00
	1	x	x	0x04

Table 40: HSA3500 plus 0x34 Operation Modes

4.1.9 0x44 – 150mA bipolar (HSA3500plus module)

Channel	ADC name	Description	DAC Output Pin
0	Mon U	Output	26
1	Mon I	Output	26
2	N / A	N / A	N / A
3	N / A	N / A	N / A
4	N / A	N / A	N / A
5	N / A	N / A	N / A
6	N / A	N / A	N / A
7	N / A	N / A	N / A

Table 41: HSA3500plus 0x44 ADC mapping

Note: No Operation Modes exist. The Operation Mode is defined by 0. The 0x44 module is located on the main CPU.

4.2 Hardware type and backplane wiring

In order to be able to differentiate the various HSA, has SPECS defined a hardware type and a backplane wiring.

Hardware type	Backplane wiring	Number of modules	Type of HSA	Comments
1	A, B	9	HSA3500	Prototype
2	C	9	HSA3500	
3	C	10	HSA3500	
4	C	11	HSA3500	
5	C	11	HSA3500	
6			HSA3500	Internal only
7	D	9	HSA3500	Standard MCD (old)
8	D	10	HSA3500	
9	D	10	HSA3500	
10	D	11	HSA3500	
11	E		HSA3500	
12	F	12	HSA3500	
13	D	12	HSA3500	
14	I	10	HSA3500	Standard MCD
15	I	11	HSA3500	Standard CCD
17	K	12	HSA3500	Standard CCD + Defl
18	O	12	HSA3500	Standard MCD + 2x400V
19	P	13	HSA3500	Standard CCD + Defl + Screen
20	I	11	HSA3500	Standard MCD + 1x400V
21	I	11	HSA3500	Standard MCD + Screen
22	I	7	HSA3500	
23	R	13	HSA3500	Wide Angle
24	S	11	HSA3500	
25	T	12	HSA3500	
26	U	10	HSA3500	THEMIS
27	V	3	HSA3500	BIS

50	H	11	HSA3500	Spin
60	L	9	HSA15000	Multi HSA
61	M	10	HSA15000	Multi HSA
62	N	4	HSA15000	Multi HSA
100	Q, X1	9	HSA3500plus	Standard MCD
101	Q, X1	11	HSA3500plus	Standard CCD
102	X1	13	HSA3500plus	Standard CCD + Deflector
103	X2	13	HSA3500plus	HSA + PreLens
104	X1	11	HSA3500plus	Standard MCD + Deflector
105	X4	10	HSA3500plus	THEMIS
106	X5	12	HSA3500plus	Wide Angle
107	X7	7	HSA3500plus	Wide Angle Pre-Lens +Defl
108	X7	5	HSA3500plus	Wide Angle Pre-Lens
109	X4	12	HSA3500plus	THEMIS + Defl
110	X9	13	HSA3500plus	Wide Angle IBM
150	X6	11	HSA3500plus	Spin
151	X8	13	HSA3500plus	SPIN + 90° Defl
160	Y1	9	HSA15000plus	Multi HSA
161	Y2	10	HSA15000plus	Multi HSA
162	Y3	3	HSA1500plus	Multi HSA

Table 42: Hardware and wiring types

4.2.1 Wiring I

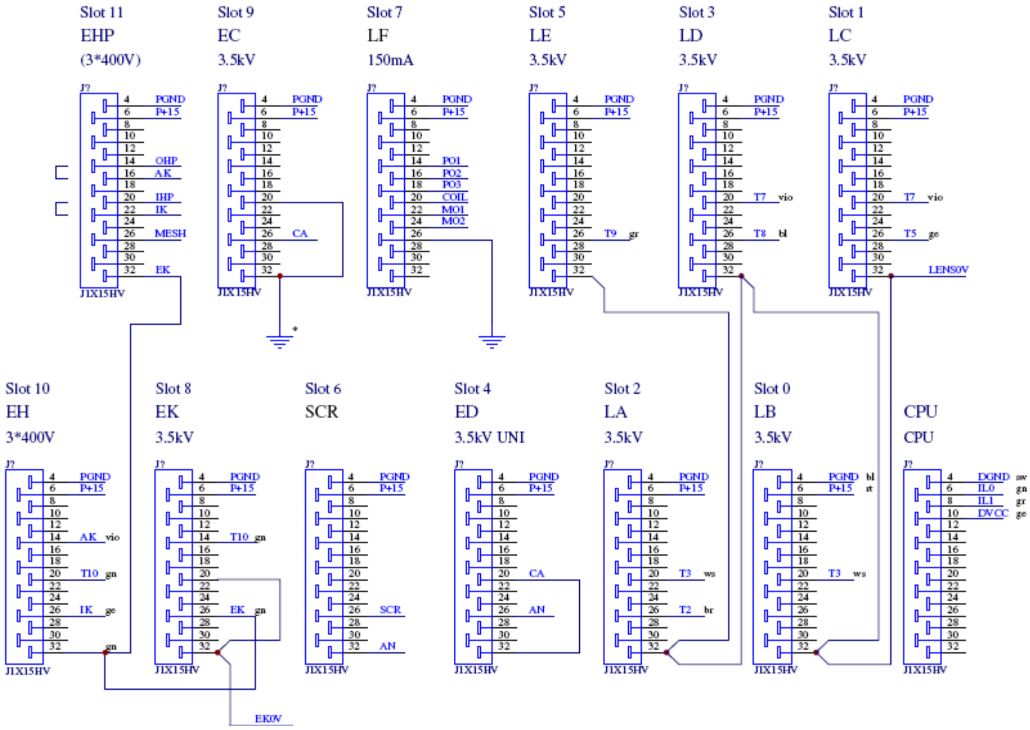


Figure 14: HSA3500 Wiring I

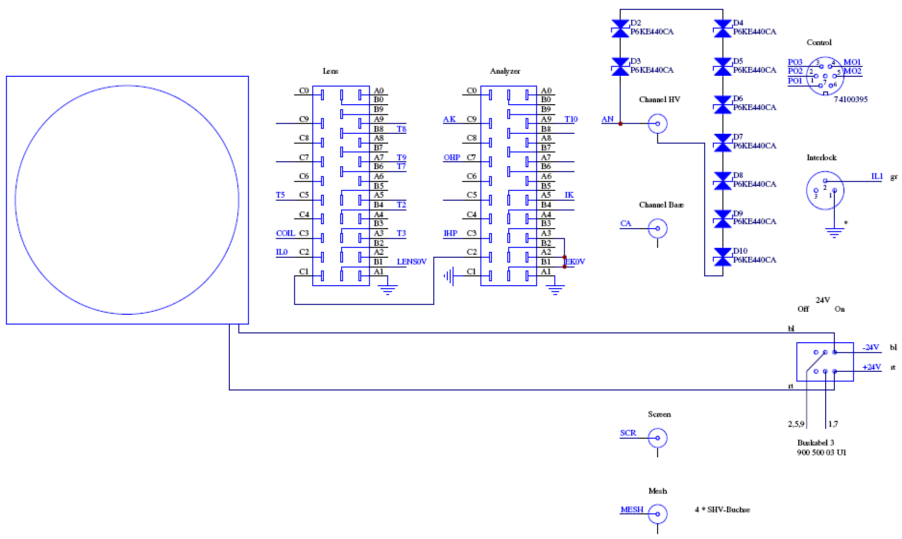


Figure 15: HSA3500 Wiring Backplane I

4.2.2 Wiring H

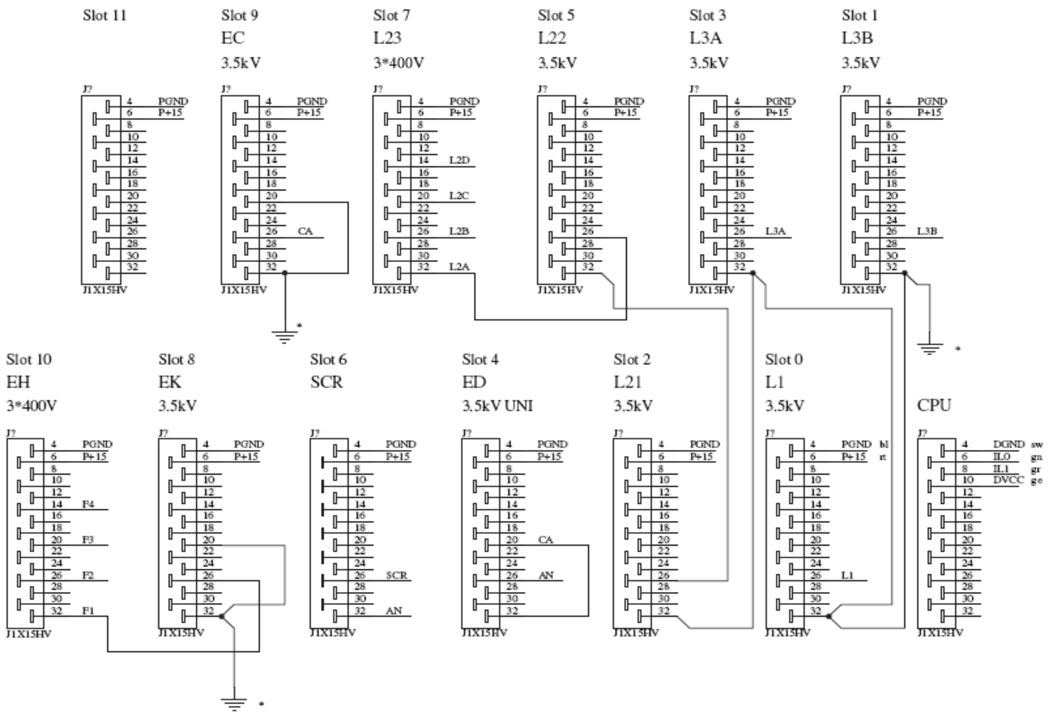


Figure 16: HSA3500 Wiring H

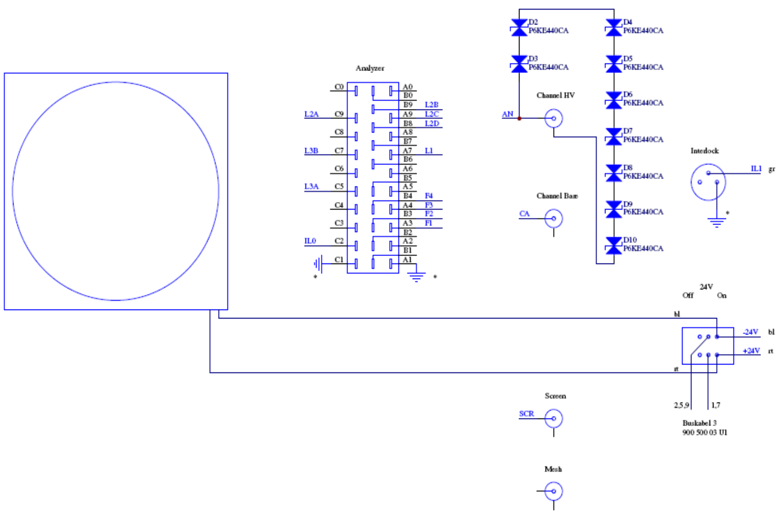


Figure 17: HSA3500 Wiring Backplane H

4.2.3 Wiring Q

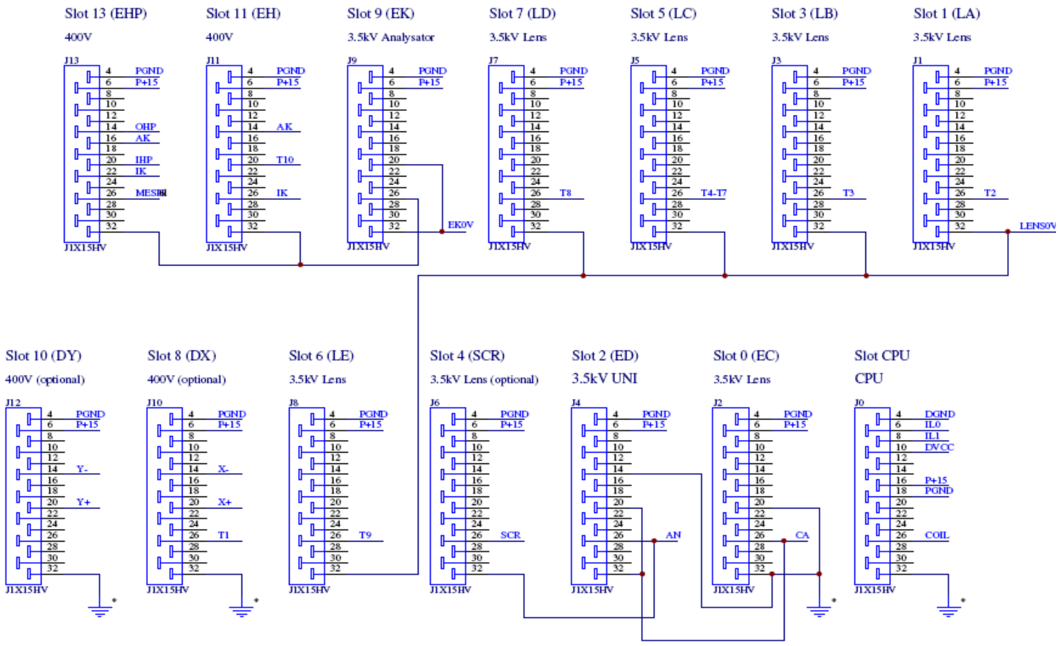


Figure 18: HSA3500 plus Wiring Q

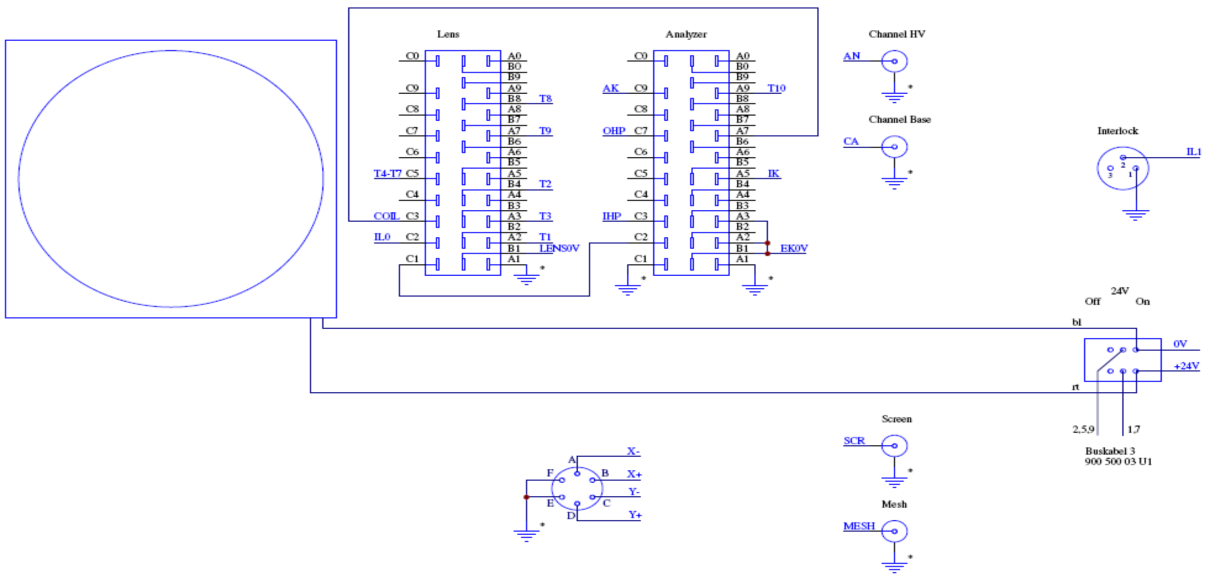


Figure 19:HSA3500 plus Wiring Backplane Q1

4.2.4 Wiring X1

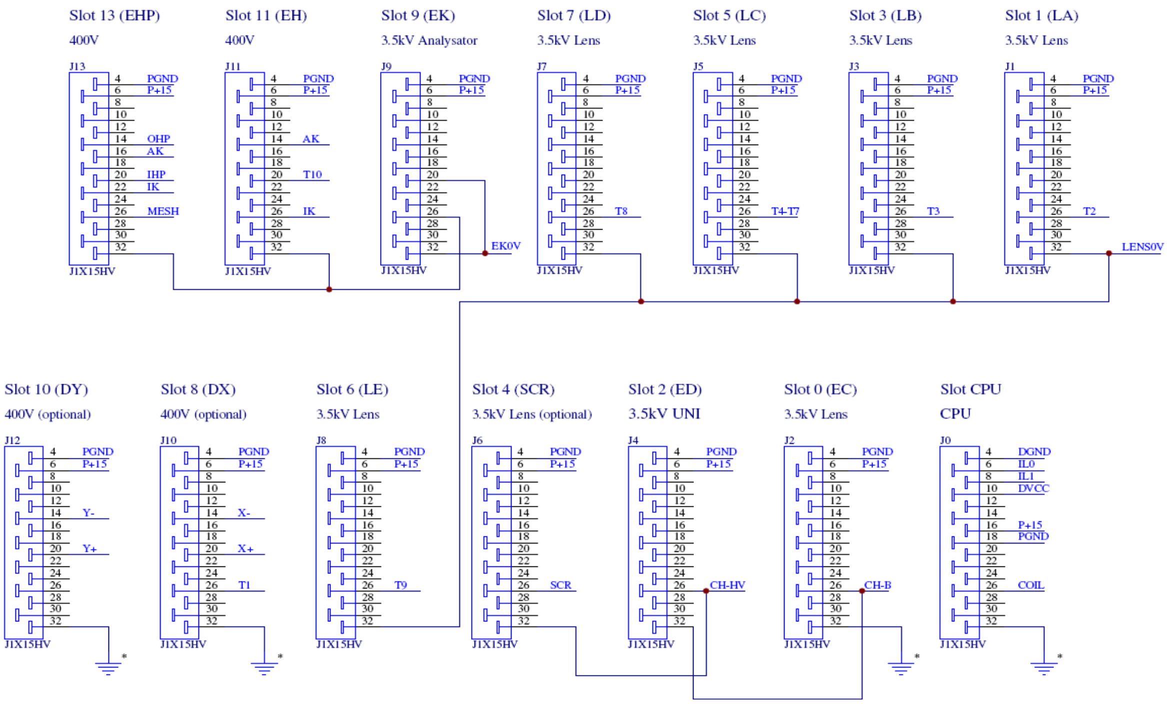


Figure 20: HSA3500 plus Wiring X1

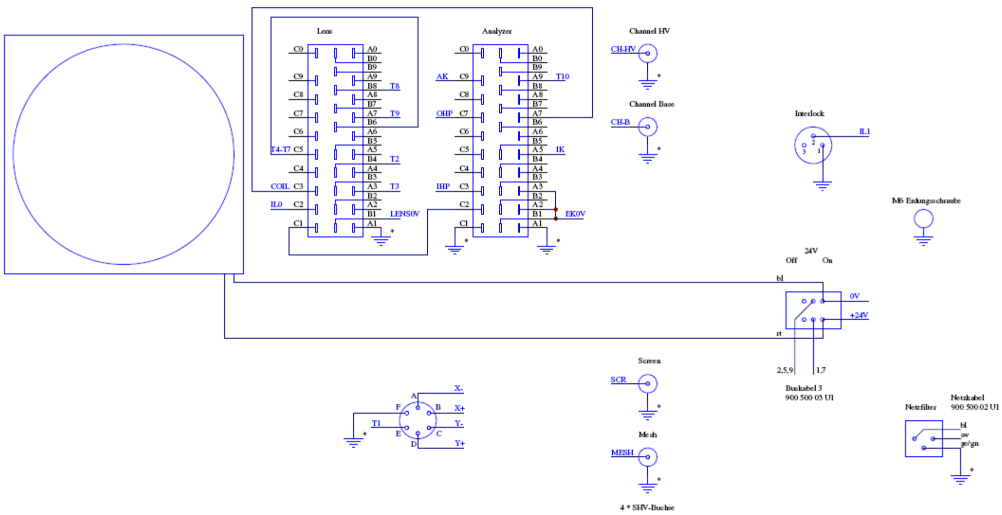


Figure 21: HSA3500 plus Wiring Backplane X1

4.2.5 Wiring X4

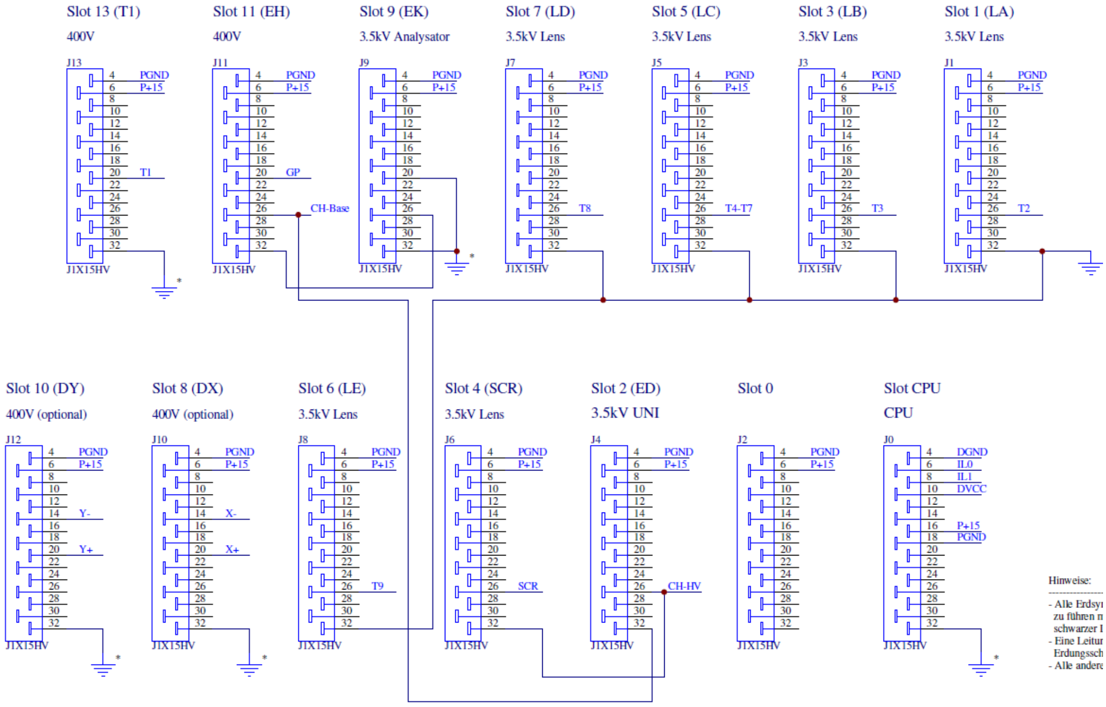


Figure 22: HSA3500 plus Wiring X4

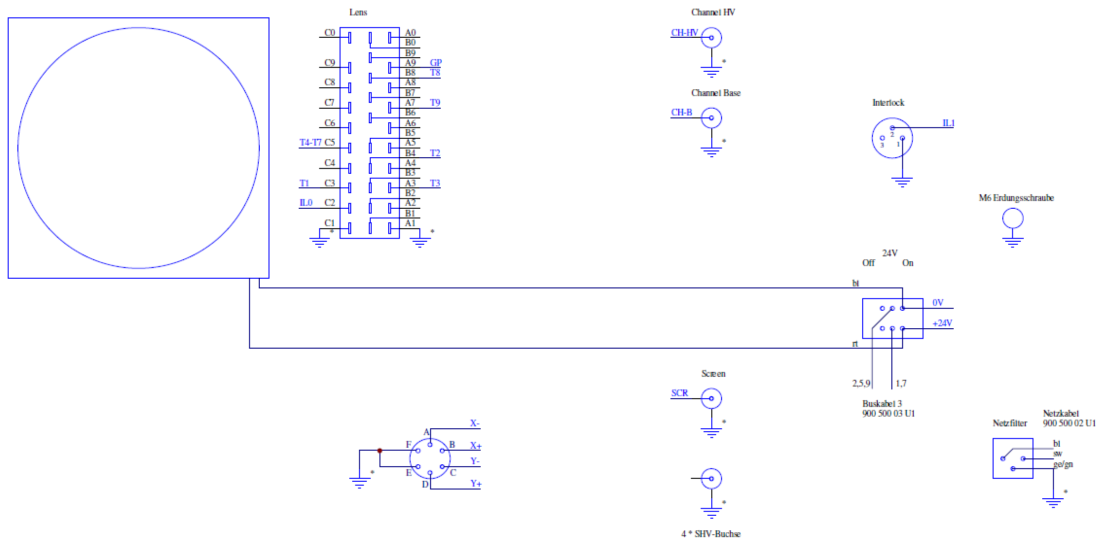


Figure 23: HSA3500 plus Wiring Backplane X4

4.3 The Filterbox cable

The filterbox cable is the connection between the HSA and the PHOIBOS analyzer.
It consists on one end of two cables which are labelled “Analyzer” / “Lens” and on the other there is a 12-Pin feedthrough. It is shown in figure 24.

4.3.1 Interlock connections

For all HSA hardware types there are safety interlocks in the cable so that the voltages are only applied if the cable is connected.
These interlocks are as follows:

–Both Connectors: C1 and C2 have to be connected to ground.

–Analyzer Connector: A2 and A3 have to be connected to B1.

4.3.2 Pin Assignment

Figure 24 shows the pin assignment of the 12 PIN UHV feedthrough connector.
The relation between the two 30-pin and the 12-pin connector is as given in table 43 and 44.

12 Pin	30-Pin Lens
2	B4
3	A3
5	C5
6	B6
7	B8
8	A7
12	C3

Table 43: Lens connector

12 Pin	30-Pin Lens
1	C3
4	C7
9	A9
10	A5
11	C9

Table 44: Analyzer connector

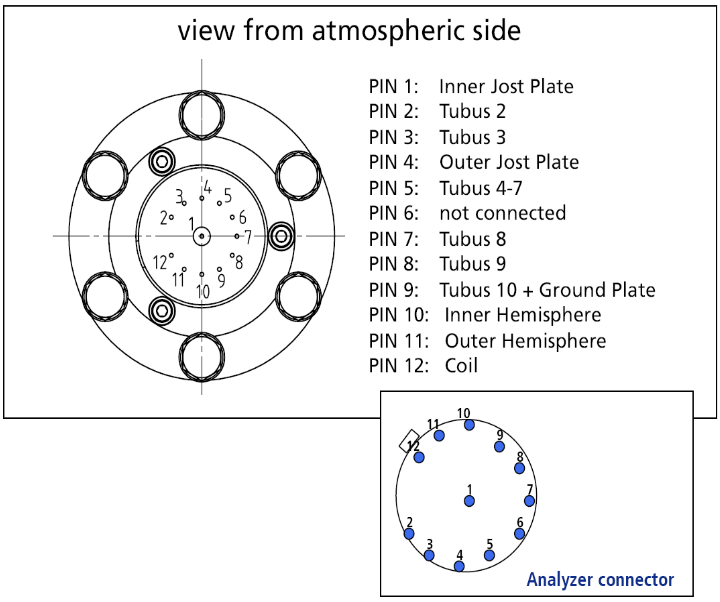


Figure 24: Schematics of the 12-pin analyzer feedthrough

4.4 HSA3500 Tool

Warning!!

When using this program, it is possible for the unit to switch on dangerous High Voltage. This voltage can also be present at the connector terminals of the back plate of the Unit (HV Meter Connector, HV BNC Connector). It is therefore very important to not connect cables or make any contact (touching or otherwise) whatsoever with the connector jacks during this process. The only exception to this is the Interlock short circuit plug.

The program can typically be run using a batch file, named something like "tool-xxx.bat", where xxx represents the type of CAN-bus card in the PC. For example "tool-isa.bat" for an ISA-card in the PC (older PC's), or "tool-pci.bat" for PCI cards. A log file is not written at this time. The output of the program is always given with a leading '#' symbol, to indicate that this output is related to the program/HSA3500.

The program will attempt to make a CAN-bus connection with the Unit. If this should fail, the program will terminate. If successful, the CAN-bus LED on the front plate of the Unit will light up (green) and stay lit. At this same time, the program will provide information about this connection on-screen, for example:

```
#Connected to Unit 0f:28:f7:f5:05:00:00
#Firmware Version: 0x0103
#Status: 0x01
#UpTime: 1331.266 Seconds
#
#HSA3500Tool. Enter ,help' for a List of Commands.
#
#>
```

In the last row, a "greater than" symbol will appear as a prompt. The program is now ready to accept user input. The following text is provided as an example session: At the prompt the command "list" is given, and the resulting output is a list of the Modules that have been found / recognized. At the prompt the command “errors" is given, errors if exist.

```
#>list
#Slot 0: Type=0xf1 State=0x0020 Error=0x00
#Slot 1: Type=0xf1 State=0x0020 Error=0x00
#Slot 2: Type=0xf1 State=0x0020 Error=0x00
#Slot 3: Type=0xf1 State=0x0020 Error=0x00
#Slot 4: Type=0xf1 State=0x0020 Error=0x00
```

```
#Slot 5: Type=0xf1 State=0x0020 Error=0x00
#Slot 6: Type=0xf1 State=0x0020 Error=0x00
#Slot 7: Nothing detected
#Slot 8: Type=0xf1 State=0x0020 Error=0x00
#Slot 9: Nothing detected
#Slot 10: Type=0xe1 State=0x0020 Error=0x00
#Slot 11: Nothing detected
#Slot 12: Nothing detected
#Slot 13: Nothing detected
#Slot 14: Nothing detected
#Slot 15: Nothing detected
#>
```

```
#>errors
#General=0 Cmd=0x00 ErrCode=0x00 FatalErr=0x00,0x00
#Total Error Counter: 0
#Fatal Error Code: 0x00
#Setup Error Code: 0x00
#EventCounter[0] = 0
#EventCounter[1] = 0
#EventCounter[2] = 0
#EventCounter[3] = 0
#EventCounter[4] = 0
#EventCounter[5] = 0
#EventCounter[6] = 0
#>
```

The example below shows the output for the following actions:

- Setting a mode
- the High Voltage is turned on
- the Voltage Monitor of the Analog-to-Digital Converter of Module 4 is read out
- the DAC of Module 4 is set to -50000 then 10000, each time reading the value at the ADC

If you connect a High Voltage Probe (Voltmeter designed for High Voltages) to the BIAS terminal on the Unit's back plate, it is possible to measure the High Voltage. It goes without saying that you should be very careful when working with High Voltages!

```
#>mode ,,,,0
#>hvon
#>adc 4 1
#ADC (4,1) = -96 [0xffa0]
#>dac 4 0 -50000
#>adc 4 1
#ADC (4,1) = -3184 [0xf390]
#>dac 4 0 100000
#>adc 4 1
#ADC (4,1) = 6080 [0x17c0]
#>
```

If a voltmeter is connected at the BIAS terminal at the back plate of the Unit, you should get a measurement of approximately -710 Volts. The LEDs on the front panel should be lit as follows: Power = on, CAN = green, HV = on and all of the other LEDs should not be lit at all.

If you now pull out one of the interlock plugs, the Voltage should immediately shut off and the Interlock LED on the front panel should start blinking. If you now connect this plug again, the Voltage should be present and the Interlock LED should go out.

The command 'help' will show a list of available commands (see below):

```
#>help
#Commands:
# adc <slot> <channel> // slot=0..15 channel=0..12
# dac <slot> <chn> <val> // slot=0..15 channel=0..2
# // val=-524288..524287
# calibrate <num> // Calibrate DACs <num> times
```

```
# echo <Text> // Prints <Text>
# errors // Show Error Counters
# exit // Terminate Program
# help // Show this Text
# hvoff // Disable HV
# hvon // Enable HV
# list // List Module Slot Contents
# mode <Mode-Def> // Define and set Mode
# // Enter ,help mode' for
# // Help on <Mode-Def>
# pmode <mode> // Set predefined Mode
# // mode=0..109
# sleep <time> // Sleep <time> Seconds
# wait // Wait for <RETURN> Key
#>
```

The exit command will cause the program to disconnect the CAN-bus connection to the Unit, and the program will then terminate.

4.4.1 Commands Explained

adc <slot> <channel>

Reads out the ADC `<channel>` of Module `<slot>`. The channel assignments are described in the section

dac <slot> <chn> <val>

Set the `<channel>` of Module `<slot>` of the DAC to the value `<val>`. The DACs are signed 20-bit converters, meaning the first bit represents whether the value is positive or negative. Said another way, the value range possible with these 20 bits is -2^{19} to $2^{19} - 1$, or in decimal: -524288 to 524287. This command is only possible when the Unit is set in operation mode (via "mode") and the High Voltage is on (via "hvon")

`calibrate <num>`

Hardware-calibrate all of the DACs of the Unit. Specify how many times `<num>` the calibration procedure should be run. Each run requires 0.4 seconds. A typical value here would be 30.

`echo <Text>`

displays the string of text `<Text>` on screen. This command is typically used when running automated scripts.

`errors`

returns a list of error codes, which are described in the section

`exit`

ends the program.

`help [mode]`

provides a short description of the command's syntax. "help mode" will describe the "mode" command.

`hvoff`

Turns off the High Voltage.

`hvon`

Turns on the high voltage. This can only occur once an operation mode has been defined with "mode".

`list`

provides a list of detected modules and their respective states. For more information please see section $$

`mode <Mode-Def>`

Sends the mode's definition to the Unit and activates this mode. `<Mode-Def>` is a comma-delimited list of Mode definitions per Module. These definitions consist of an optional exclamation mark (!) and a hexadecimal number. The Hexadecimal number corresponds directly to the status of the relays of the module. The functions of the relays themselves are explained in the appendix. When an exclamation point is in front of the Hexadecimal number, the High Voltage generation for this module will remain switched off. Otherwise it will be turned on.

Turning off the High Voltage is a good idea when you only want to use the relays of one and only one module. The assignment of individual definitions to a corresponding module number is carried out via the position of that assignment in the list, beginning with 0. Modules that are not required or are not present should have an empty definition. The following examples can explain this paragraph better:

```
mode 1
// Module 0, Mode 1, HV on

mode !1
//   Module 0, Mode 1, HV off

mode , , 0, !7, , 1
// Module 2 Mode 0 HV on;  Module 3 Mode 7 HV off; Module 5 Mode 1
HV on
```

`pmode <mode>`

Sets mode `<mode>`, whose definition exists in NVRAM. `<mode>` is a decimal number between 0 and 109. This option is not longer supported.

`sleep <time>`

Pauses the Program's execution for `<time>` seconds. This command is usually only required for use in scripts / batch files.

`wait`

Pauses the program's execution until the user presses the ENTER key. This command typically only used in scripts.

4.4.2 Error Messages of the Diagnostic and Programming

Errors can occur at various points in the software system:

- In the firmware of the Unit
- In the CAN-bus interface layer of the PC software
- In the application itself (the diagnostic tool)

The error messages of the Application are typically text-based and self-explanatory (for example: "Voltage difference too small/large"). Therefore the error messages of the Unit and of the CAN-bus layer will now be discussed:

The error messages have the format (printf notation):

```
General=%d Cmd=0x%02x ErrCode = 0x%02x FatalErr = 0x%02x, 0x%02x
```

Above,

`<General>` is the Error Message of the CAN/Devicenet layer of the PC,

`<Cmd>` is the Firmware command that failed,

`<ErrCode>` is the error code of the Unit, which relates to the command that failed

`<FatalErr = 0x00, 0x00>` means that a fatal error did not occur.

In Detail,

`<General>` - Messages

0: No error

1: Invalid parameter when loading the DeviceNet layer. e.g. incorrect I/O address of the CAN-card, incorrect name of the DLL driver. The parameters are normally in the batch file that the actual program runs. See the call parameters of the respective program for more information.

2: Corba error while loading the DeviceNet Layer. Either the software was incorrectly installed, or a driver has crashed. In this case, rebooting the PC is recommended, then trying again.

3: Internal error while establishing the DeviceNet connection. If this occurs, reboot the PC and try again.

4: A Devicenet connection to the unit already exists. It may be that another program (running in another window) established this connection. If unsure, please reboot the PC and try again to establish a connection.

5: A Connection to the Unit does not exist (Internal Error). This error normally occurs as a result of bugs that exist in the software trying to access the unit.

6: Timeout while waiting for connection to the device. This error occurs when the unit doesn't respond to a command sent via CAN-bus within the expected amount of time. This occurs most often due to an error in the Devicenet allocation which occurs at the program's start / initialization.

Other possible causes:

- CAN-Card is defective
- CAN cable connections are incorrect
- Incorrect IRQ of the CAN-card in the PC driver has been set, or the PC does not support that IRQ (for example, because it has been disabled in the BIOS)
- The Unit is not turned on
- The Unit is defective and can't allocate resources. Note this error can also occur while the Unit is in operation. (For example, when the Unit "crashes")

7: Invalid response from the Unit.

8: Error message from the Unit. This error message will be seen when the program receives an unexpected message or error from the Unit. In these situations, the fields `<Cmd>`, `<ErrCode>`, and `<FatalErr>` can provide more information about the nature of the problem.

9: At program runtime a "Duplicate MAC ID Message" was received from the Unit.

Possible causes:

- The Unit was turned off and then on again while the program was running.
- Unit rebooted as a result of some other error, typically a defective hardware component.

SPECS

EEC Declaration of Conformity

We - SPECS GmbH - herewith declare that the products defined below meet the basic requirements regarding safety and health of the relevant EC directives by design, type and versions which are brought into circulation by us.

In case of any product changes made without our approval, this declaration will be void.

Designation of the products:	Hemispherical Electron Analyzer	
Types: *100 and 150*	Analyzer:	*PHOIBOS*
	Power Supply:	*HSA 3500*
	Detector:	*Channeltron-Detector with PCU 300 counting electronics*

The products conform to the following directives:

- EC Directive on Low-Voltages (73/23/EWG)
- EC EMC Directive (89/336/EWG)

Applied harmonized standards:

DIN EN ISO 55011	08.03	Industrial, scientific and medical (ISM) radio-frequency equipment - Radio disturbance characteristics - Limits and methods of measurement.
DIN EN ISO 55022	09.03	Information technology equipment - Radio disturbance characteristics - Limits and methods of measurement.
DIN EN ISO 61000	94-98	Electromagnetic compatibility (EMC) - Part 4-2 + A1, 4-3 + A1, 4-4, 4-5, 4-6, 4-8 and 4-11.
DIN EN ISO 61010-1	08.02	Safety requirements for electrical equipment for measurement, control, and laboratory use - Part 1: General requirements.
DIN EN ISO 61326-1	05.04	Electrical equipment for measurement, control and laboratory use - EMC requirements - Part 1 : General requirements.

Applied national standards and technical specifications:

DIN 31 001	04.83	Sicherheitsgerechtes Gestalten technischer Erzeugnisse; Schutzeinrichtungen; Begriffe, Sicherheitsabstände für Erwachsene und Kinder.

Berlin, 9.7.2009

Oliver Schaff

Figure 25: HSA 3500 CE Conformity Declaration

SPECS®

EEC Declaration of Conformity

We - SPECS GmbH - herewith declare that the products defined below meet the basic requirements regarding safety and health of the relevant EC directives by design, type and versions which are brought into circulation by us.

In case of any product changes made without our approval, this declaration will be void.

Designation of the products: Power Supply HSA3500plus

The products conform to the following directives:

- EC Directive on Low-Voltages (73/23/EWG)
- EC EMC Directive (89/336/EWG)

Applied harmonized standards:

DIN EN ISO 55011	08.03	Industrial, scientific and medical (ISM) radio-frequency equipment - Radio disturbance characteristics - Limits and methods of measurement.
DIN EN ISO 61000	94-98	Electromagnetic compatibility (EMC) - Part 4-2 + A1, 4-3 + A1, 4-4, 4-5, 4-6, 4-8 and 4-11.
DIN EN ISO 61010-1	08.02	Safety requirements for electrical equipment for measurement, control, and laboratory use - Part 1: General requirements.
DIN EN ISO 61326-1	05.04	Electrical equipment for measurement, control and laboratory use - EMC requirements - Part 1 : General requirements.

Applied national standards and technical specifications:

DIN 31 001	04.83	Sicherheitsgerechtes Gestalten technischer Erzeugnisse; Schutzeinrichtungen; Begriffe, Sicherheitsabstände für Erwachsene und Kinder.

Berlin, 7.1.2009

Dr. Oliver Schaff
Head of R&D Division

Figure 26: HSA 3500 plus CE Conformity Declaration

Chapter LF List of Figures

Chapter LT

List of Tables

SPECS®