



SRTD
High-Voltage Supply
Controller Software

User Manual and Reference
v5.0a

Henk Boterenbrood
NIKHEF-H, Amsterdam
December, 1994

1. Introduction.....	1
2. Basic operation.....	1
3. Automatic trip recovery.....	3
4. Command/reply protocol.....	4
4.1 Controller operation examples	5
5. Calibration of the powersupplies	6
6. Controller commands.....	8
6.1 Commands for normal operation.....	9
6.1.1 CTR	9
6.1.2 DEE	10
6.1.3 DIS.....	10
6.1.4 ENA	10
6.1.5 HLP	10
6.1.6 LHX.....	11
6.1.7 RDA	11
6.1.8 RDD	11
6.1.9 RDU	12
6.1.10 RPS.....	12
6.1.11 RSA	12
6.1.12 RSE.....	13
6.1.13 RSS	13
6.1.14 RST.....	14
6.1.15 RVA	14
6.1.16 RVO	15
6.1.17 SCD	15
6.1.18 SCF.....	15
6.1.19 SDD.....	15
6.1.20 SDR	16
6.1.21 SMT.....	16
6.1.22 SPY.....	16
6.1.23 SSF.....	17
6.1.24 SVO	17
6.1.25 SVS	17
6.1.26 SWI.....	18
6.2 Commands for calibration.....	18
6.2.1 CAL.....	18
6.2.2 CAD	18
6.2.3 CAV	19
6.2.4 DEP	19
6.2.5 GDU	19
6.2.6 GVO	20
6.2.7 RPA.....	20
6.2.8 SDC/SDc.....	21
6.2.9 SDF/SDf.....	21
6.2.10 SVP	21
6.3 Specials.....	22
6.3.1 COM	22
7. Error numbers	22
8. Running code from EPROM or EEPROM.....	25
9. Default Settings.....	26

1. Introduction

This document is a reference and a user manual for the software running on the 80C552 microcontroller which forms the heart of the high-voltage powersupply module which was developed to supply high-voltage to the photomultiplier tubes (PMs) of the **Small-angle Rear Tracking Detector (SRTD)** which forms part of the **ZEUS** detector at DESY, Hamburg.

The 'normal' user should be aware that this document contains much more information than he needs to know or wants to know....

The microcontroller sets, monitors and controls the voltage and duty-cycle of 3 powersupplies, each of which will supply high-voltage to a number of PMs.

Additionally it sets and monitors an auxiliary powersupply which supplies the power to the high-voltage supplies.

In the following we will refer to the combination of controller, auxiliary supply and the high-voltage supplies as **HV-controller**.

2. Basic operation

The HV-controller is connected to the outside world by means of an RS232 serial connection operating at a baudrate of 9600.

Through this serial line the HV-controller can be controlled and the status of the powersupplies monitored. Also the initial calibration of each powersupply is performed under control of the user sitting at a terminal at the other end of the serial line.

The HV-controller behaves like a slave, which means it will start performing an action (e.g. switching-on a powersupply or reporting about the current status) only after receiving a command on the serial line.

All commands and messages to/from the HV-controller are in the form of ASCII-strings so that it is sufficient to connect a terminal to the RS232-line in order to operate the powersupplies, although for regular and continuous monitoring it will be more convenient to connect a hostcomputer to the RS232-line, to do the monitoring under software control.

It is possible to have up to 16 SRTD HV-controllers connected to one RS232 serial line; (a) hex-switch(es) on the controller-board can be set to uniquely identify an HV-controller; an HV-controller will only respond to commands addressed to it explicitly or to broadcasted commands (commands containing a wildcard address, see section 4); an arbitration scheme (hardware/software) prevents more than one controller at a time accessing the serial line for output.

The on-board 80C552 microcontroller controls the powersupply voltages by means of a set of Digital-to-Analog converters (DAC) with 6 bits resolution; in the case of the high-voltage supplies the voltage is set combining two DAC outputs, a coarse and a fine setting, resulting in a resolution close to 12 bits. The operating voltage ranges from 800V to 1200V, with a resolution of about 0.15 V. The accuracy of the set voltages is within ca. ± 1 V with the control process running and within ca. ± 2 V without it.

The auxiliary supply is set by one 6 bit DACs, and is set at a nominal voltage of about 80 V.

Each supply has an enable/disable input used by the controller to switch the supply on or off.

The voltage and duty-cycle of all powersupplies (the auxiliary supply and the high-voltage supplies) are measured using the micro-controller's on-chip Analog-to-Digital Converter (ADC) with a resolution of 10 bits (about 0.4 V of high-voltage resolution).

Because of the digital-to-analog and analog-to-digital conversions involved each powersupply has been calibrated so that it can be used in a meaningful way; the calibration constants are stored in on-board EEPROM.

In normal active operation (after the user has started the control and monitoring process) the controller takes samples of the duty-cycle and voltages of the active powersupplies, typically with a rate of 10 Hz; with a rate of typically 1 Hz averages are calculated and the voltage is adjusted if necessary and checks are made on the current settings and duty-cycles to see if everything still falls within preset safety and control margins; if this is not the case for a particular powersupply, the supply is switched off; this event is recorded in a statusword which the controller keeps for each powersupply.

The controller also records minimum and maximum values of duty-cycles and voltages measured which can be monitored by the user/host (not for the auxiliary powersupply).

Therefore the user/host should at regular intervals check the status of the powersupplies and take appropriate action whenever a supply is not functioning properly.

Settings and calibration parameters can be stored permanently in the on-board EEPROM, so that once stored they are used after every (power-up) reset.

The software allows for a new program to be downloaded if e.g. a new version of the software is available, but it might also be a totally different program, e.g. some testprogram. The program is stored in the onboard

EEPROM. If certain programming rules are obeyed it is possible to switch from one program to the other and back.

3. Automatic trip recovery

An automatic trip recovery feature has been added to the controller software:

if the control process is running and a HV-supply is switched off by the software (a 'trip') because of a value running out of bounds, it will be switched on again after a short period of time; if a number of consecutive trips occur shortly after one another the HV-supply will remain switched off permanently, until e.g. the user switches it on again explicitly.

The number of consecutive trips that causes the HV-supply to be switched off permanently can be set by the user; if this number is set to zero or to 1 the controller behaves as if the automatic trip recovery is not enabled.

The total number of trips during an 'enabled'-period for every HV-supply is stored by the controller and these 'trip-counters' can be inspected by the user.

After a trip the HV-supply is switched on again after 5 sample-periods; after the supply automatically has been switched on the usual control-delay period applies.

4. Command/reply protocol

A command to the HV-controller has the following syntax:

<s><n>[.<m>]<cmd>[<par>]<CR>

with

s = command tag ('S' for SRTD).

n = address of the HV-controller (SRTD: 0-9, A-F or *;
* meaning all HV-controllers connected to this RS232
line) this command applies to.

m = number of the powersupply on this HV-controller the
controller the command applies to (0, 1, 2, ... or *;
* meaning all high-voltage powersupplies; number 0 is the
auxiliary supply and for several commands has to be ad-
dressed separately); if the preceding period and **m** are not
present it is assumed to be *.

cmd = three-letter command mnemonic.

par = numeric parameter (in the form of an ASCII string),
but only if the command requires a parameter; if the com-
mand requires a parameter and the parameter is missing it
is assumed to be zero.

A command will always result in a reply, unless the HV-controller ad-
dress is not correct or a controller with the specified address is not con-
nected to the serial line; every HV-controller reads all command-lines, but
will ignore all lines that do not contain its address.

Lines are interpreted by the HV-controller on a line-by-line basis: only
after a carriage-return <CR> has been received the input is interpreted
and processed and an acknowledge or error message is returned.

A reply from the HV-controller has the following syntax:

<r><n>.<m><cmd>.[<par1>.<par2>...]<LF><CR>

with

r = reply tag ('s' for SRTD).

n = address of the HV-controller (SRTD: 0-9,A-F);
if all HV-controllers were addressed by means of a *-
wildcard every individual HV-controller will send a reply.

m = number of the powersupply on this HV-
controller the command applied to (0, 1, 2, ... or *; * mean-
ing all high-voltage powersupplies; number 0 is the auxil-
iary supply); if the preceding period and **m** were not pre-

sent in the command in the reply **m** will be a *-wildcard character.

cmd = the same three-letter command mnemonic if the command was processed successfully, "**ERR**" otherwise.

par1,

par2 = return value(s) (in the form of ASCII strings, separated by periods); the number of return values depends on the command given.

4.1 Controller operation examples

Assuming that all settings are stored in EEPROM, issuing the following commands to the HV-controller(s) are sufficient to switch on all power-supplies (the auxiliary supply (no. 0) has to be switched on separately) and start the measure-and-adjust process:

```
S*.0ENA  
S*ENA  
S*CTR1
```

Assuming that all voltage settings stored in EEPROM should be overruled by a new setting of 980V (for every HV-supply, but not for the auxiliary supply of course):

```
S*.0ENA  
S*SVO980  
S*ENA  
S*CTR1
```

Subsequently each HV-controller at regular intervals should at least be questioned about the status of its powersupplies:

```
S*RSS
```

or each controller separately (assuming we have 4 controllers, numbered 1 to 4):

```
S1RSS  
S2RSS  
S3RSS  
S4RSS
```

A full status of one particular powersupply can be requested by (for example):

```
S2.1RSA
```

5. Calibration of the powersupplies

When we calibrate a powersupply we determine the parameters that define the conversion functions between DAC-values and Volts, between Volts and ADC-values and between duty-cycle percentage and ADC-value. Each of these transfer functions is assumed to be linear, which in practice proves to be a valid assumption.

The voltage \mathbf{V} as a function of the coarse-setting DAC value $\mathbf{D}_{\text{coarse}}$ is:

$$\mathbf{V} = \mathbf{a}\mathbf{D}_{\text{coarse}} + \mathbf{b}$$

The additional voltage $\Delta \mathbf{V}$ added by the fine-setting DAC-value \mathbf{D}_{fine} is:

$$\Delta \mathbf{V} = \mathbf{a}'\mathbf{D}_{\text{fine}}$$

The ADC-value \mathbf{A}_V as a function of voltage \mathbf{V} is:

$$\mathbf{A}_V = \mathbf{c}\mathbf{V} + \mathbf{d}$$

and finally ADC-value \mathbf{A}_D as a function of duty-cycle value \mathbf{D} (as a percentage) is:

$$\mathbf{A}_D = \mathbf{e}\mathbf{D} + \mathbf{f}$$

In order to be able to calculate parameters \mathbf{a} , \mathbf{b} , \mathbf{a}' , \mathbf{c} , \mathbf{d} , \mathbf{e} and \mathbf{f} , it is sufficient to take 2 samples of each transfer function.

The accuracy of the set high voltage is dependent mainly on the accuracy and stability of the parameters describing the measured ADC-value-to-voltage conversion, i.e. parameters \mathbf{c} and \mathbf{d} .

The difference between the measured voltage and the requested voltage is used to adjust the current set voltage, independent of what the set voltage actually is according to the DAC-value-to-voltage settings on the two DACs; that explains why the accuracy of the set high voltage depends mainly on the accuracy and stability of parameters \mathbf{c} and \mathbf{d} .

Next follows an example of a complete calibration procedure for an HV-controller; the command lines to be typed by the user are in bold text type; see the chapter on calibration commands for a full explanation of the commands used; voltage values and calibration parameters shown are just examples; they will be different for each HV-controller calibrated.

Switch to calibration-mode:

S1CAL

s1.*.CAL.1

Calibration of the auxiliary supply voltage:

S1.0CAV

s1.0.CAV.25.0

S1GVO293

s1.0GVO.293.95.0

S1GVO675

Now calibration constants **a**, **b**, **a'**, **c** and **d** are calculated:

s1.0GVO.675.1265.5.0.11506.0

Calibration of HV-supply 1:

S1.1CAV

s1.1CAV.5.0

S1GVO7658

s1.1GVO.7658.95.0

S1GVO11553

s1.1GVO.11553.50.0

S1GVO9576

s1.1GVO.9576.50.100

S1GVO9662

Now calibration constants **a**, **b**, **a'**, **c** and **d** are calculated:

s1.1GVO.9662.8098.715.146.1771.-1204

The same voltage calibration procedure is to be followed for all other HV-supplies.

Calibration of a supply's duty-cycle:

S1.1CAD

s1.1CAD.

The supply is only enabled, the minimum voltage is generated (ca. 750 V).

S1GDU12

s1.1GDU.12

Now the load on the supply should be increased, e.g. by short-circuiting the HV-power output(s).

S1GDU44

Now calibration constants **e** and **f** are calculated:

s1.1GDU.44.6503.234

The same duty-cycle calibration procedure is to be followed for all other supplies.

Now save all parameters in EEPROM:

S1.0SVP

s1.0SVP.8098.715.146.1771.-1204.6503.234

S1.1SVP

s1.0SVP.8030.721.147.1756.-1201.6339.3171

S1.2SVP

s1.0SVP.8180.725.146.1758.-1209.6188.8977

etc. etc.

6. Controller commands

Below two lists of command mnemonics and their symbolic name are shown; the first table contains the commands used during normal operation, the second table contains the commands used when the powersupplies are being calibrated; in principle commands from this second group are only used once; calibration parameters are subsequently stored in the onboard EEPROM and are not expected to change (much) in time.

Mnem	Name
CTR	CONTROL_MODE
DEE	DELETE_EEPROM_PROGRAM
DIS	DISABLE_SUPPLY
ENA	ENABLE_SUPPLY
HLP	HELP
LHX	LOAD_INTELHEX
RDA	READ_DUTYCYCLE_ADC
RDD	READ_DC_DARKCURRENT
RDU	READ_DUTYCYCLE
RPS	READ_PROCESSOR_STATUS
RSA	READ_SUPPLY_ALL
RSE	READ_SETTINGS
RSS	READ_SUPPLY_STATUS
RST	RESET
RVA	READ_VOLTAGE_ADC
RVO	READ_VOLTAGE
SCD	SET_CONTROL_DELAY
SCF	SET_CONTROL_FREQUENCY
SDD	STORE_DC_DARKCURRENT
SDR	SET_DUTYCYCLE_RANGE
SMT	SET_MAX_TRIPS
SPY	SPY_MODE
SSF	SET_SAMPLE_FREQUENCY
SVO	SET_VOLTAGE
SVS	SAVE_SETTINGS
SWI	SWITCH_PROGRAM

Mnem	Name
CAL	CALIBRATION_MODE
CAD	CALIBRATE_DUTYCYCLE
CAV	CALIBRATE_VOLTAGE
DEP	DELETE_PARAMETERS
GDU	GET_DUTYCYCLE
GVO	GET_VOLTAGE
RPA	READ_PARAMETERS
SDC	SET_DAC_COARSE
SDF	SET_DAC_FINE
SVP	SAVE_PARAMETERS

A description and explanation of each command, with examples using SRTD HV-controller address 1 and powersupply address 2 follow.

It is always possible to use a wildcard for the HV-controller address (except for the **HLP** command); for some commands it is not allowed to use a wildcard as a power-supply address and for several commands the use of a powersupply address is meaningless altogether and is then left out.

The cases where a command returns an error are not considered here; see section 7 for a description of possible error conditions.

6.1 Commands for normal operation

6.1.1 CTR

command: **S1CTR1** or **S1CTR0**

reply: **s1.*CTR.1** or **s1.*CTR.0**

Switches the control mode on or off; if the parameter in the command is 1 the control process is started, if it is 0 this process is stopped. Supplying a powersupply address is meaningless.

The control process measures and adjusts the HV-supplies that are enabled (see **ENA**) according to the set voltage, measure and control frequencies defined (see **SVO**, **SSF**, **SCF**).

If the powersupply still has to be set to the desired voltage when the control process is started the control process will wait a certain period of time as defined by the set control delay (see **SCD**) in order to let the voltage settle, before starting to regulate it.

6.1.2 DEE

command: **S1DEE**
reply: **s1DEE.**

Makes the program stored in EEPROM invalid; subsequent **SWI** commands do not work anymore, until a new program is downloaded.

6.1.3 DIS

command: **S1.2DIS**
reply: **s1.2DIS.**

Disables one powersupply or all HV-supplies if the *-wildcard is used; the DACs for this supply are set to zero.

6.1.4 ENA

command: **S1.2ENA**
reply: **s1.2ENA.**

Enables one powersupply or all HV-supplies if the *-wildcard is used. If the control process is active it will take care of setting the voltage to the required value. If the control process is not active the voltage will go to a value corresponding to a DAC value of zero or to a value previously set with the **SVO** command.

The auxiliary powersupply is not controlled by the control process and thus gets immediately set to its requested voltage when it is enabled or by the **SVO** command, independent of the control process.

6.1.5 HLP

command: **S1HLP**
reply: <version_number>
<list_of_mnemonics>
s1.*HLP.

The version number and the list of mnemonics are only produced when the controller is in 'spy'-mode (see **SPY**). Supplying a powersupply address is meaningless.

6.1.6 LHX

command: **S1LHX**
 <file_in_IntelHex_format>
reply: **s1.*LHX.9872**

After the command is given the controller will keep on reading lines and interpreting them as 'Intel-Hex' formatted lines until a so-called end-of-data line is read or if the first character of a line is 'Q' (to enable the user to escape from the command when e.g. a wrong file without an end-of-data line was downloaded).

The lines are parsed and written to RAM in binary form (the original file is in ASCII format). When the end-of-data line has been received the EEPROM is programmed, which might take several seconds depending on the size of the program. Then a reply as shown above is returned in which the number represents the number of bytes of RAM used to temporarily store the code file.

Supplying a powersupply address is meaningless.

If an error occurred somewhere in the process the reply will be:

S1.*ERR.<line_no>.<error_no>

in which <line_no> is the number of the line in the 'Intel-Hex' formatted file which led to the error and <error_no> is a number stating the type of error that occurred (see section 7).

6.1.7 RDA

command: **S1.2RDA**
reply: **s1.2RDA.485**

Reads the current value of the duty-cycle of a powersupply in ADC-units (a number between 0 and 1023); if the *-wildcard is used the reply contains the duty-cycle ADC-value of all HV powersupplies, e.g.:

s1.*RDA.876.53.485

6.1.8 RDD

command: **S1RDD**
reply: **s1RDD.109.110.111**

Reads the so-called duty-cycle dark-current values of all HV-powersupplies in ADC-units which were stored in EEPROM with command **SDD** (see there).

6.1.9 RDU

command: **S1.2RDU**
reply: **s1.2RDU.25**

Reads the current value of the duty-cycle of a powersupply and transforms it into a percentage (a number between 0 and 100); if the *-wildcard is used the reply will contain the duty-cycle percentages of all HV-powersupplies, e.g.:

s1.*RDU.87.13.25

6.1.10 RPS

command: **S1RPS**
reply: **s1.*RPS.50.0**

Returns the software version number and whether the code is running from EEPROM (1) or not (0). In the example above the version is **5.0** and the code is not running from EEPROM (but from EPROM). Supplying a powersupply address is meaningless.

6.1.11 RSA

command: **S1.2RSA**
reply: **s1.2RSA.0.1000.1000.1020.970.1070.109.108.111.109.0.0**

Returns current status (see **RSS**), set and measured values of one particular powersupply. A single powersupply address has to be provided; a wildcard is not allowed. The values returned are respectively:

- the supply status,
- last measured average voltage (V),
- requested voltage (V),
- last set voltage (V),
- minimum measured voltage (V),
- maximum measured voltage (V),
- last measured average duty-cycle (ADC-units),
- minimum measured duty-cycle (ADC-units),
- maximum measured duty-cycle (ADC-units),
- the currently valid darkcurrent duty-cycle (ADC-units),
- the trip counter of this supply
- and the last error status that occurred.

6.1.12 RSE

command: **S1RSE**

reply: **s1.*RSE.1.100.10.3.5.80.1000.1000.1000.3**

Returns the current settings, being:

- controlprocess on/off (0=off, 1=on),
- the sample frequency (Hz*10),
- the frequency of regulating (Hz*10),
- the control delay (s),
- the duty-cycle safety margin (in ADC-units),
- the required voltages for all supplies (the HV-supplies and the auxiliary supply) in Volts
- and the number of consecutive trips after which a HV-supply will be switched off permanently (no automatic trip recovery anymore).

Supplying a powersupply address is meaningless.

Note that the frequencies are multiplied by a factor of 10. In the example above the frequencies are 10 Hz and 1 Hz respectively and the control delay is 3 seconds; the control process will wait 3 seconds in order to let the voltage settle before starting to regulate the supply voltage (if the control-process is active).

6.1.13 RSS

command: **S1RSS**

reply: **s1.*RSS.0.3.1.0.0.0.0.0**

Returns the current on/off-status of all supplies (the first 4 numbers in this reply; the auxiliary supply and 3 HV-supplies respectively) and 4 trip counters (one for each supply; the counter of the auxiliary supply --the first number-- will always be zero, because it is never automatically switched off). Supplying a powersupply address is meaningless.

Per supply a byte describes whether the supply is switched on or off and if switched off, whether there was an abnormal situation that caused it to be switched off, where set bits have the following meaning:

0x01	disabled
0x02	disabled due to measured duty-cycle out of allowed range
0x04	disabled due to measured voltage out of allowed range
0x08	disabled due to set voltage out of allowed range
0x10	disabled due to set voltage out of absolute supply range
0x20	disabled due to on-board powerfailure
0x40	error occurred in setting a DAC belonging to this supply (I ² C-bus)
0x80	not defined

In the example at the beginning of this paragraph the reply shows that HV-supply 3 and the auxiliary supply are switched on, supply 1 is switched off because its duty-cycle went outside its safety margin (and it tripped twice), and supply 2 is switched off (no abnormal cause).

6.1.14 RST

command: **S1RST**
 reply: none

Causes the controller to be reset by its watchdog timer; parameters, constants etc. are restored to their default or EEPROM-stored values. (If running from EEPROM there will be a switch back to the program in EPROM.)

6.1.15 RVA

command: **S1.2RVA**
 reply: **s1.2RVA.485**

Reads the current value of the voltage of a powersupply in ADC-units (a number between 0 and 1023); if the *-wildcard is used the reply will contain the voltage ADC-value of all HV-powersupplies, e.g.:
s1.*RDA.876.53.485

6.1.16 RVO

command: **S1.2RVO**
reply: **s1.2RVO.1025**

Reads the current value of the voltage of a powersupply; if the *-wildcard is used the reply will contain the voltages of all HV-powersupplies, e.g.: **s1.*RVO.1023.1024.1021**

6.1.17 SCD

command: **S1SCD5**
reply: **s1.*SCD.5**

Sets the control delay for all HV powersupplies to the requested value. The control delay is the delay in seconds taken into account by the control process when the supply is switched on or when the set voltage is changed; this in order to let the voltage settle first before starting to regulate it. Supplying a powersupply address is meaningless.

6.1.18 SCF

command: **S1SCF20**
reply: **s1.*SCF.20**

Sets the frequency with which averages are calculated for the voltage and duty-cycle of active powersupplies and the frequency with which these quantities are checked and/or regulated. The required frequency should be multiplied by a factor of 10. In the example above the control frequency is set to 2 Hz.

Supplying a powersupply address is meaningless; the control frequency applies to all HV-supplies controlled by this HV-controller.

6.1.19 SDD

command: **S1.2SDD109**
reply: **s1.2SDD.109**

Stores the so-called duty-cycle dark-current values of a particular HV-powersupply in EEPROM. This value is the duty-cycle (in ADC-units) of the powersupply with PM-tubes installed in place, but in rest (in the 'dark'); it is different for different set voltages, but must be stored for one particular voltage when the HV-supply with PM-tubes is installed; later on during breaks in the experiment checks at this voltage can be per-

formed by the user to see if this 'dark-current' value changed significantly, which could point to a faulty PM-tube or a light-leak.

At controller start-up for each supply the dark-current value stored in EEPROM is used to determine the safe range of duty-cycle; command **SDD** immediately results in a new safe range for a running control process; the control process will measure the deviations in the duty-cycle and switch off the supply if these exceed this range of stored-dark-current \pm margin, with 'margin' as set by command **SDR** (or was stored in EEPROM as part of the saved settings).

6.1.20 SDR

command: **S1SDR10**
reply: **s1.*SDR.10**

Sets the margin within which the duty-cycle of each HV-supply has to fall; the parameter value is in ADC-units because a percentage appeared to be a bit too coarse; the capacity of the HV-supplies is such that only about 0.5 percent fluctuation in the duty-cycle results when a PM-tube is exposed to a lightsource (and the supply should be switched off then!).

6.1.21 SMT

command: **S1SMT2**
reply: **s1.*SMT.2**

Sets the maximum number of consecutive trips; if a HV-supply trips again within a certain period of time, the supply is disabled and is not switched on again automatically.

Two trips are called 'consecutive' if at the first control-loop after the previous trip the supply trips again. Setting the number to 0 or 1 will cause the supply NOT to be switched on automatically after a trip at all.

6.1.22 SPY

command: **S1SPY1** or **S1SPY0**
reply: **s1.*SPY.1** or **s1.*SPY.0**

Switches the so-called 'spy' mode on or off; if the parameter in the command is 1 the spy-mode is switched on, if it is 0 the spy-mode is switched off. In 'spy'-mode the output produced by the HV-controller will be a bit more informative. This is useful when a user is directly connected to the HV-controller with his terminal. Default spy-mode is off.

6.1.23 SSF

command: **S1SSF100**
reply: **s1.*SSF.100**

Sets the frequency with which the voltage and the duty-cycle of the active supplies are sampled. The required frequency should be multiplied by a factor of 10. In the example above the sample frequency is set to 10 Hz.

Supplying a powersupply address is meaningless; the sample frequency applies to all supplies controlled by this HV-controller, including the auxiliary supply.

6.1.24 SVO

command: **S1.2SVO1120**
reply: **s1.2SVO.1120**

Sets the required voltage for the stated powersupply to the stated value (in Volts). The DACs for this supply are set to the proper values unless the control process is active and the supply is a HV-supply, then the control process will take care of setting the voltage. A *-wildcard for the powersupply address is allowed; then the voltage setting applies to all HV-supplies.

6.1.25 SVS

command: **S1SVS**
reply: **s1SVS.100.10.3.5.70.1000.1000.1000.3**

Saves the current general settings in EEPROM for permanent storage, so that from now on they will be the current values after every subsequent (power-up) reset.

The command returns the saved current settings, being the sample frequency (Hz*10), the frequency of regulating (Hz*10), the control delay (s), the duty-cycle safety margin (in ADC-units), the required voltages for all supplies (the auxiliary supply and the HV-supplies) in Volts, and finally the maximum number of trips in a row before a supply remains switched off permanently. Supplying a powersupply address is meaningless.

Note that the frequencies are multiplied by a factor of 10. In the above example the actual frequencies are 10 Hz and 1 Hz respectively.

6.1.26 SWI

command: **S1SWI**
reply: none

If a valid program is present in EEPROM or when the present program is running from EEPROM this command will switch to the start of the program in EEPROM or EPROM respectively. See also section 8.

6.2 Commands for calibration

6.2.1 CAL

command: **S1CAL1** or **S1CAL0**
reply: **s1.*CAL.1** or **s1.*CAL.0**

Switches the calibration mode on or off; if the parameter in the reply is 1 calibration mode is switched on, if it is 0 calibration mode is switched off.

The HV-controller has to be in calibration mode to be able to perform the commands described in the next sections (i.e. **CAD**, **CAV**, **DEP**, **GDU**, **GVO**, **RPA**, **SDC**, **SDF**, **SVP**)

6.2.2 CAD

command: **S1.2CAD**
reply: **s1.2CAD.**

Notifies the HV-controller that a duty-cycle calibration of a particular supply is to take place. It will now expect two **GDU** inputs with duty-cycle percentages so that it can calculate the ADC-to-duty-cycle parameters after the second **GDU** input.

The powersupply voltage should be set (**SVO** or **SDC**) to an appropriate value.

A single powersupply address has to be provided; a wildcard is not allowed.

6.2.3 CAV

command: **S1.2CAV**
reply: **s1.2CAV.25.0**

Notifies the HV-controller that a voltage calibration of a particular supply is to take place. The powersupply is enabled and the supply is set to its first calibration setting; the two numbers in the reply state what this setting is for the DACs of this powersupply: in the example above the coarse-setting DAC is set to 25% of its full scale and the fine-setting DAC to 0% of its full scale.

The controller will now expect a number of **GVO** inputs with voltages in Volts so that it can calculate the DAC-to-Volts and ADC-to-Volts parameters after the last **GVO** input.

A single powersupply address has to be provided; a wildcard is not allowed.

6.2.4 DEP

command: **S1.2DEP**
reply: **s1.2DEP.**

Deletes the calibration parameters stored in EEPROM for the stated supply; after the next reset the default parameters will be used for this supply.

6.2.5 GDU

command: **S1GDU34**
reply: **s1.2GDU.34**
or
s1.2GDU.34.10230.0

Tells the HV-controller the current duty-cycle as a percentage. The controller will measure the corresponding ADC-value and when this was the last step in the calibration it will calculate the ADC-to-dutycycle conversion parameters **e** and **f**; these parameters are returned in the reply as shown above in the second reply; they are multiplied by 1000 and displayed as an integer, although inside the program they are reals; so the approximate values are:

e = 10.230
f = 0.000

There is no need to provide a powersupply address; the address provided previously by the **CAD**-command is used.

6.2.6 GVO

command: **S1GVO10402**
reply: **s1.2GVO.10402.75.25**
or
s1.2GVO.10402.12543.700.103.1277.-708

Tells the HV-controller the current voltage in tenths(!) of a Volts (in the example above 1040.2 V). The controller will measure the corresponding ADC-value and when this was the last step in the calibration it will calculate the ADC-to-Volts conversion parameters **a**, **b** and **a'** and the Volts-to-ADC conversion parameters **c** and **d**; after calculating the conversion parameters they are returned in the reply as shown above in the second reply, displayed as integers although inside the program they are represented by reals; parameters **a** and **a'** are multiplied by 1000 before they are put in the reply. The values in the example above are thus (approximately):

a = 12.543
b = 700.0
a' = 0.103
c = 1.277
d = -708.0

If the GVO-command is not the final calibration step, the reply will contain --apart from the voltage entered by the user-- the next settings of the coarse- and fine-setting DAC, in percentages of the full DAC-ranges of the powersupply being calibrated, in the example above 75% and 25% respectively.

There is no need to provide a powersupply address; the address provided previously by the **CAV**-command is used.

6.2.7 RPA

command: **S1.2RPA**
reply: **s1.2RPA.12543.700.543.1277.-708.9950.800**

Reads the currently used values of the calibration parameters that are used in the linear conversions of DAC and ADC units to Volts and percentages. A single powersupply address has to be provided; a wildcard is not allowed.

The numbers contained in the reply are parameters **a**, **b**, **a'**, **c**, **d**, **e** and **f** respectively, where **a**, **a'**, **c**, **e** and **f** are multiplied by a factor 1000.0. All

these parameters inside the controller program are real numbers, but in the reply to **RPA** they are truncated to an integer.

In the example above the parameters are thus (approximately):

a = 12.543
b = 700.0
a' = 0.543
c = 1.277
d = -708.0
e = 9.950
f = 0.800

6.2.8 SDC/SDc

command: **S1.2SDC60**
reply: **s1.2SDC.60**

Sets the DAC which does the coarse setting for the stated powersupply to the stated value, which is a percentage of the full DAC range. A *-wildcard for the powersupply address is allowed. (Command **SDc** takes as parameter the DAC-value to be set (0-63)).

6.2.9 SDF/SDf

command: **S1.2SDF60**
reply: **s1.2SDF.60**

Sets the DAC which does the fine setting for the stated powersupply to the stated value, which is a percentage of the full DAC range. A *-wildcard for the powersupply address is allowed. (Command **SDc** takes as parameter the DAC-value to be set (0-63)).

6.2.10 SVP

command: **S1.2SVP**
reply: **s1.2SVP.**

Stores the calibration parameters for the stated powersupply in EEPROM for permanent storage, so that they will be used from now on after every (power-up) reset.

The reply returns the stored values in the same way as described above in the section on command **RPA**.

6.3 Specials

6.3.1 COM

command: **S1COM1** or **S1COM0**

reply: **s1.*COM.1** or **s1.*COM.0**

Enables (1) or disables (0) the use of the RTS and CTS handshake lines during RS232 serial output, which is necessary in case a so-called COM-box is installed to multiplex the serial lines of several HV-controllers onto one RS232 line. Default the software assumes a COM-box is present; if this is not the case (e.g. when testing an individual HV-controller through a terminal connected directly to the HV-controller) the first command after a reset should be the **COM** command, otherwise the software will hang up in the serial output.

7. Error numbers

If the reply of the HV-controller to a command contains the same command mnemonic it means that the command has been processed correctly. If however the reply contains the mnemonic **ERR** something went wrong. The number in the reply following the **ERR**-mnemonic identifies the nature of the error.

Sometimes the **ERR**-reply contains more numbers; the error identifier number is always the last number of the reply.

Below a list of the error identifiers and their symbolic name is shown.

Error	Name
200	EEPROM_TIMEOUT
220	CALIB_VOLT_ZERO
221	CALIB_DAC_ZERO
222	CALIB_ADC_ZERO
223	CALIB_DUTY_ZERO
230	HEX_CHECKSUM_ERROR
231	HEX_NO_CODE
232	HEX_INCOMPLETE
233	HEX_TOO_BIG
240	I2C_TIMEOUT
241	I2C_STATUS
248	LINE_TOO_LONG
249	LINE_INCOMPLETE
250	ERROR_IN_ADDRESS
251	ERROR_IN_PARAMETER
252	PAR_OUT_OF_RANGE
253	ILLEGAL_CMD
254	UNKNOWN_CMD

A description of each error number follows:

- **EEPROM_TIMEOUT**
A timeout occurred on programming data --like settings, parameters or code bytes of a downloaded program-- in the EEPROM.
- **CALIB_VOLT_ZERO**
Sensible voltage calibration parameters could not be calculated because there is a zero voltage difference between different calibration steps.
- **CALIB_DAC_ZERO**
Sensible voltage calibration parameters could not be calculated because there is a zero DAC-setting difference between different calibration steps.
- **CALIB_ADC_ZERO**
Sensible voltage calibration parameters could not be calculated because there is a zero ADC-reading difference between different calibration steps.
- **CALIB_DUTY_ZERO**

Sensible calibration parameters could not be calculated because there is a zero duty-cycle difference between the two calibration steps.

- **HEX_CHECKSUM_ERROR**

A checksum error was detected in a line of the downloaded Intel-hex format file.

- **HEX_NO_CODE**

The user issued the command to switch to the program in EEPROM, but no valid program is present in EEPROM.

- **HEX_INCOMPLETE**

The downloaded Intel-Hex format file did not have a proper end-of-data line.

- **I2C_TIMEOUT**

A timeout occurred on receiving an acknowledge on the I²C-bus.

- **I2C_STATUS**

An incorrect status was detected during an I²C-bus operation.

- **LINE_TOO_LONG**

The (command)line received by the controller is longer than the controller can store (the line length is limited to 50 characters including <CR>).

- **LINE_INCOMPLETE**

Somehow the <CR> of the (command)line is lost.

- **ERROR_IN_ADDRESS**

The address of the addressed supply/supplies in the commandline is incorrect (for the given command).

- **ERROR_IN_PARAMETER**

A parameter in the command line is incorrect.

- **PAR_OUT_OF_RANGE**

A parameter in the commandline is outside its valid range.

- **ILLEGAL_CMD**

The command given in the commandline is valid but illegal in this mode/version of the program.

- **UNKNOWN_CMD**

The command given does not occur in the list of available commands.

8. Running code from EPROM or EEPROM

If a version of the HV-supply controller software is produced it is important to know whether the code will be stored in EPROM (32 KByte) or EEPROM (32 KByte), in other words: whether the code is to be the default running code when the controller is switched on or whether the code is downloaded by the user, and then run by issuing the command **SWI**.

Note that downloading needs to be done only once, but to run this code after switching on the controller the **SWI**-command has to be given first to switch from the default program in EPROM to the version that was previously downloaded to EEPROM.

In *global.h* when running from EPROM the following definition has to be present (to protect against switching to an 'empty' EEPROM, or downloading code to EEPROM when the running program itself is already stored in EEPROM, overwriting itself):

```
#define RUNNING_EEPROM 0
```

and when running from EEPROM:

```
#define RUNNING_EEPROM 1
```

In order to be able to make a distinction between versions, it is recommended to change the **version** number in *hv_main.c* to a unique number. In the example below the version is set to **a.b**.

```
char version = 0xab;
```

The current version number can then be obtained using command **RPS**.

Module *switch.s03* should contain the following assembler code to be able to switch from EPROM to EEPROM (Note: in case of a 32 KByte EPROM or EEPROM the code bytes have to be put in the HEX-file by hand at address 07FFEh !!):

```
PUBLIC switch_program  
ASEG  
ORG 0FFFEH ; 32K EPROM mirrored in 08000H-0FFFFH  
PIO DEFINE 0C0H ; 80C552 port P4  
switch_program:  
CPL PIO.5  
END
```

and the following code to be able to switch from EEPROM to EPROM:

```
PUBLIC switch_program  
ASEG  
ORG 07FFEh ; EEPROM not mirrored in 08000H-0FFFFH  
PIO DEFINE 0C0H ; 80C552 port P4  
switch_program:  
CPL PIO.5  
END
```

If the above directions are followed the user can (safely) switch from the program (version) in EPROM to the one in EEPROM and back.

9. Default Settings

When no settings are ever saved the default settings are:

sample-frequency:	10.0	Hz
control-frequency:	1.0	Hz
control delay:	3	s
requested high-voltage:	1000	V
auxiliary supply voltage:	80	V
range of allowed duty-cycle	duty-cycle =	darkcurrent ADC-count ± 10 (ca. 0.5%)

and:

maximum number of trips:	1
--------------------------	---

and '-hardwired' in the code-:

minimum high-voltage to set:	800	V
maximum high-voltage to set:	1200	V
minimum sample-frequency:	1.0	Hz
maximum sample-frequency:	20.0	Hz
minimum control-frequency:	0.1	Hz
maximum control-frequency:	10.0	Hz
range of absolute 'dark-current'	duty-cycle:	(-->not implemented<--)
range of no adjustment to voltage:	measured voltage =	requested voltage ± 0.3 V
range of allowed measured voltage:	measured voltage =	required ADC-count ± 50 (ca. 20V)
range of allowed set voltage:	set voltage =	requested voltage ± 50 V