Software standard commands for SYSTEM 9100

REV.	By	Date	Description	Pages
1.00	DT	08.04.2008	Initial	All
1.01	DT	05.08.2008	ADRS added	7
1.02	DT	01-10-2008	Auto slewrate added Min/max & initial value	52,64,65
1.03	DT	14-01-2009	New error codes, ID, ESC<ID, val3 in ESC<Slopetime added Data error → DATA CONTENTS, Illegal request → Illegal command	

APP1. SW 1 Standard Commands

Following are the commands for the standard software listed in alphabetic order.
Please see the SW appendix for detail explanation of every command.

STANDARD COMMANDS. summary

AD X	Read value from an ADC channel.
ADR	Read the address of the MPS
ADR XXX	Write an address to a MPS
ADRS	Read the address of the MPS
ADRS XXX	Write an address to a MPS with address number as response
ANACTRL	Change to Analogue control
ASW	Enters answer mode
CMD	Read current control mode
CMDSTATE	Read current control state
CPUCTRL	Change to CPU DAC control
DA XXXXXX	Writes a value to an Digital to Analog converter. (Alternative WA command.)
ERRC	Coded error message.
ERRT	Text string error message.
F	Main Power OFF
ID	Read User configurable identification text field
LALL	Listen ALL.
LOC	Change to Local Control
LOCK	Lock the MPS in Local Control.
N	Main Power ON
NASW	No answer mode.
NERR	No error message
PO	Polarity status
PO +/-	Change to Normal polarity
POLOOL	Polarity Output Off Limit
PRINT	Reads internal user information about the MPS.
RA	Read the set value.
REM	Change to remote control.
RLOCK	Remote line only
RWR	Read Writing-Slewrate for polarity switch
RS	Reset interlocks.
S1	Read the internal status.
S1H	Read internal status in HEX format
S1FIRST	Read the interlock first catch status.
S1FIRSTH	Read the interlock first catch status in HEX format
S3	Read the internal extended status
S3H	Read internal extended status in HEX format
TD	Test DAC
UNLOCK	Unlock the MPS
VER	Reads the software version
WA XXXXXX	Writes a value to an Digital to Analog converter. (Alternative DA command.)
WR XXXXX	Writte-Slewrate for polarity switch

X is a number from 0 to 9 and Commands in quotation marks are optional.

Following are the commands for the standard software listed in alphabetic order.
Please see the SW appendix for detail explanation of every command.

Esc SET UP COMMANDS. summary

ESC<AD	Configures the AD converter scaling and routing (Output reading adjustment or output reading in % or Amps)
ESC<ADR	Configures the communication address setting (in RS422 mode)
ESC<ADSET	Auto Configures the scaling "gain" and Offset for an AD converter channel
ESC<AUX	Configures the special options
ESC<AUX2	Configures the special options 2
ESC<BAUD	Configures the Baud rate for the serial lines.
ESC<COLDBOOT	Configures the power up state. (Wake up position)
ESC<CPURESET	Hardware reset / CPU reset
ESC<DA	Configures The Digital to Analog converters. (Slew rate setting in A/sec or set value in Amps)
ESC<DASET	Auto Configures the scaling (gain) and Offset for a DA converter channel
ESC<ID	Identification field
ESC<LINE	Configures the protocol for the serial lines
ESC<POLDELAY	Configures the delay between Off and polarity switch change

X is a number from 0 to 9 and Commands in quotation marks are optional.

Programming:

The power supply communication protocol is build upon plain ASCII characters where each command or reply is delimited by a "Carriage Return" <CR> character. However a reply has a “Line Feed” <LF> character added before the <CR> for a friendlier display when using a terminal. <LF> characters on commands will be ignored.

Hint. Actually the protocol allows full control of the power supply from a "dumb" terminal. In case of a service- debug- situation a terminal can be used to tap the communication transfer by a simple parallel connection.

Hint: When debugging, the "ERRT" command enables error messages to be given as a read able text.

More commands may be transmitted in a chain but each single command must be trailed individually with the delimiter character <CR>. The power supply is able to execute up to 200 commands a second depending of the complexity of each command.

Ps. Issuing short commands faster than the time to transmit the answer eg. “S1" will overload the internal transmit buffer regardless of the selected baud rate.

All commands can be divided into three sections.

a) Directive commands. Eg. the "N" command that turns the power supply ON
b) Status commands . Eg. the "S1" that returns the power supply status
c) Set up commands. Eg. the “ESC”<BAUD sets up the baudrate

Status commands delivers always a reply whereas directive- and setup- commands only responds with an error message if the command couldn’t be understood or if the given parameters are incorrect. This feature is very useful when using RS485 protocol.

Answer scheme if set to “Always Answer” mode.

d)	Directive commands.	Answer:	- No answer - ERROR message - OK if set to always answer mode
e)	Status commands .	Answer:	- Data - ERROR message
f)	Set up commands.	Answer:	- No answer - ERROR message - OK if set to always answer mode.

Below is an example written in BASIC on how to turn ON the power supply and read the status without and with acceptance answer:

Turning the power supply ON and reading/evaluating the status with always answer disabled.

```
LPRINT "N"+CHR$(13)              :REM Turns the power supply on
LPRINT "S1"                      :REM Issues the status command
LINPUT S1$                       :REM Read the MPS reply
IF LEFT$(S1$,1) = CHR$(?)        :REM Is it an error message reply?
 GOTO ERROR_HANDLING             :REM Yes then go to error module
ENDIF
J=1
DO                               :REM evaluate status reply
 IF MID$(S1$,J,1)="!"
  GOSUB STATUS(J)_ACTIVE         :REM set this status bit active
 ELSE
  GOSUB STATUS(J)_ACTIVE         :REM set this status bit inactive
 ENDIF
 J=J+1
UNTIL J=24
```

Turning the power supply ON with always answer enabled

```
J=0 :ERROR$=”“
DO
 J=J+1                           :REM Counter for maximum attempts
 LPRINT "N"+CHR$(13)             :REM Turns the power supply on
 LINPUT RE$                      :REM Read the MPS reply with 0.1 Sec. time out
 IF LEFT$(RE$,1) = CHR$(?)       :REM Is it an error reply?
  ERROR$=RE$                     :REM Mark the error code
 ELSEIF RE$=”OK”                 :REM Is it a good reply
  BRAKE                          :REM then exit DO loop
 ELSEIF J=6                      :REM Try only six times
  IF LEFT$(ERROR$,1) = CHR$(?)   :REM Was it error reply?
   GOTO ERROR_HANDLING           :REM Yes then go to error module
  ELSEIF
   GOTO NO_COMMUNICATION         :REM Yes then go to “No answer” error module
  ENDIF
 ENDIF
UNTIL -1                         :REM loop endless
```

Ps. An ERROR message includes a "?BELL". (Bell = ASCII 7.)

AD - AD X

Command: AD'sp'ch'cr'

ch: ASCII digit 0 to 15

Example: AD 0
Syntax: AD'sp'0'cr'

Answer: 'val'lf''cr'
ch: ASCII digit 0 to 15
val: ch 0 to 5, 7 and 9, and 10 to 15 ASCII digit 000 to 999
ch 6 ASCII +/-00 to +/-99
ch 8 ASCII 00000 to 99999
(If a signed response is chosen will a sign be added to the front of the value.
See also 'ESC'< AD command for further information.)
or Error message

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

Description:

The AD command is used to read the different Analog to Digital converters. The AD channels and their response are described on the next column.

AD channel 6 differs from the other by containing a sign, plus or minus, before the value.

Note: It is possible to change the number of digits and the sign representation for each channel in the "Esc<" setup command. If an M-Panel is attached, please do not change the number of digits for the first 9 channels.

Over flow will be limited to a reading of all digits equal to 9. Under flow to 0 if unsigned format is used.

Nothing else is affected.

AD continued

CHANNEL	VALUE	UNITS	RESPONSE
0	Output current	(I/In)*100	"SDDD"
1	Ambient Temperature	DEg°C*100	"DDD"
2	Output Voltage	(V/Vn)*100	"SDDD"
3	Internal +15V sup.	V*10	"DDD"
4	Internal -15V sup.	Num.(V*10)	"DDD"
5	Internal +5V sup.	V*10	"DDD"
6	Delta Temperature	DEg°C*10	"SDD"
7	I set value.	V*1000	"SDDD"
8	Optional Iout (16 Bit).	(I/In)*99999	"SDDDDD"
9	Iout (16 Bit)(Crtl panel).	(I/In)*120	"SDDD"
10	Not used.		"DDD"
11	Output current display	(I/In)*1000	"SDDDD"
12	Output voltage display	(V/Vn)*1000	"SDDDD"
13	Not used.		"DDD"
14	Not used.		"DDD"
15	Not used.		"DDD"
16	Not used.		"DDD"
17	I set value.	A*100	"SDDD"
18	Spare.	X*100	"DDD"

Where D is a number from 0 to 9, and S is a sign character (either + or -).

The UNITS of AD 0 and AD 2 can be in AMPS and VOLTS if bit 4 (leaver 5 on dip switch) is set to 0 in the Auxiliary setup. The scaling factor for these two channels must be adjusted accordingly.

ADR - AdDRess (write)

Command: ADR address'cr'

address: ASCII digits 00 to 255 in decimal notation.

Example: ADR 23
Syntax: ADR 23'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

DATA CONTENTS, means that parameter format is incorrect or a non-digit character is found in the data field or the parameter is outside the specification.

Description:

The **ADR** command selects an actual power supply (unit) working RS422 multi drop mode. The previously addressed unit is automatically de-selected.

Please note, ADR=0 and ADR=255 are interpreted as always addressed. That is it will respond to any command in the line.

There is only one exception using the **ADR** command due to the **LALL** mode. When all connected units are in **LALL** mode, an **ADR** command given after the **LALL** command, will disable the **LALL** function.

Related commands: **ADRS** (write)

Affected commands: **LALL**

ADR - AdDRess (read)

Command: ADR'cr'

Answer: Address

or Error message

Example: Command: ADR
Syntax: ADR'cr'

Answer: address
Syntax: address'lf''cr'

Address: ASCII digits 000 to 255 in decimal notation.

Errors: **SYNTAX ERROR,** means wrong syntax.

Description:

The **ADR** command verifies the address of the addressed power supply (unit). The command returns the address of the addressed unit.
There is only one exception to the **ADR** command due to the **LALL** mode. When all connected units are in **LALL** mode, an **ADR** command given after the **LALL** command, will disable the **LALL** function. For the same reason, no answer will be generated, because the **LALL** mode has to be cancelled, before any answer can be generated. In this case, if you want to know the address of the addressed unit, the **ADR** command has to be repeated.

In cases where no answer is given, even after the second **ADR** command maybe a non-existing unit has been address, or the actual unit-address has been switched off. In that case just address another unit to verify the communication line and then re-address to the "dead" address for test.

Related commands: **ADRS** (read)

Affected commands: **LALL**

ADRS - AdDRessSpecial (write) from version 1.01

Command: ADRS address'cr'

address: ASCII digits 00 to 255 in decimal notation.

Example: ADRS 23
Syntax: ADRS 23'cr'

Answer: No answer, except errors

or address

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

DATA CONTENTS, means that parameter format is incorrect or a non-digit character is found in the data field or the parameter is outside the specification.

Description:

The **ADRS** command selects an actual power supply (unit) working RS422 multi drop mode. The previously addressed unit is automatically de-selected. Furthermore the selected MPS will respond its address instead of OK as command ADR does.

Please note, ADRS=0 and ADRS=255 are interpreted as always addressed. That is it will respond to any command in the line.

There is only one exception using the **ADRS** command due to the **LALL** mode. When all connected units are in **LALL** mode, an **ADRS** command given after the **LALL** command, will disable the **LALL** function.

Related commands: **ADR** (write)

Affected commands: **LALL**

ADRS - AdDRessSpecial (read) from version 1.01

Command: ADRS'cr'

Answer: Address

or Error message

Example: Command: ADRS
Syntax: ADRS'cr'

Answer: address
Syntax: address'lf''cr'

Address: ASCII digits 000 to 255 in decimal notation.

Errors: **SYNTAX ERROR,** means wrong syntax.

Description:

The **ADRS** command verifies the address of the addressed power supply (unit). The command returns the address of the addressed unit.
There is only one exception to the **ADRS** command due to the **LALL** mode. When all connected units are in **LALL** mode, an **ADRS** command given after the **LALL** command, will disable the **LALL** function. For the same reason, no answer will be generated, because the **LALL** mode has to be cancelled, before any answer can be generated. In this case, if you want to know the address of the addressed unit, the **ADRS** command has to be repeated.

In cases where no answer is given, even after the second **ADRS** command maybe a non-existing unit has been address, or the actual unit-address has been switched off. In that case just address another unit to verify the communication line and then re-address to the "dead" address for test.

Related commands: **ADR** (read)

Affected commands: **LALL**

ANACTRL - ANAlog ConTRoL Set value (remote line only)

Command: **ANACTRL'cr'**

Example: ANACTRL
Syntax: ANACTRL'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is local-line.

Description:

The **ANACTRL** command is used to switch the I Set value to the analogue control input line instead of the internal CPU DAC control.

Nothing else is affected.

Related commands: **CPUCTRL**

ASW - AnSWer

Command: **ASW'sp'ch'cr'**

ch: ASCII digit 0 or 1 1=local; 0=remote

Example: ASW 0
Syntax: ASW' sp'0'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is local-line.

Description:

The **ASW** command is used to switch the remote line into an auto-answer mode in which some setup commands can generate an answer. The **ASW** mode is suppressed when the unit is in **LALL** mode. Commands which will generate an answer are listed in the following: **WA, PO+/-**

Nothing else is affected.

Related commands: **NASW**

CMD - CoMmanD line

Command: CMD'cr'

Answer: If line-in-command is remote line:
REM
Syntax: 'sp'REM'lf''cr'

or If line-in-command is local line:
LOC
Syntax: 'sp'LOC'lf''cr'

or Error message

Example: Command: CMD
Syntax: CMD'cr'

Answer: REM
Syntax: 'sp'REM'lf''cr'

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **CMD** command returns an answer about which line is the line-in-command (the line that may give commands, both channels can always read status). The command is used by the control panel to decide the status of the line-in-command indicator. From remote line it can be used to decide if anyone has changed the mode, (from the control panel). For example if an unexpected ILLEGAL COMMAND has been returned to a set command..

Nothing else is affected.

CMDSTATE - CoMmanD line STATE

Command: CMDSTATE'cr'

Answer: If line-in-command is in remote:
REMOTE
Syntax: REMOTE'lf''cr'

or If line-in-command is in local and the command is given from the
Syntax: LOCAL'lf''cr'

or If line-in-command is in local and the command is given from the local LOCKed from remote line:
Syntax: LOCK'lf''cr'

or Error message

Example: Command: CMDSTATE
Syntax: CMDSTATE'cr'

Answer: REMOTE
Syntax: REMOTE'lf''cr'

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **CMDSTATE** command is an extended command similar to **CMD** command. It returns answer about which line is the line-in-command. The answer is more detailed than in **CMD** and is constructed to be used between the controller and the IEEE-488 interface unit, during initializing.

Nothing else is affected.

Related commands: **CMD**

CPUCTRL - CPU ConTRoL Set value (remote line only)

Command: **CPUCTRL'cr'**

Example: ASW
Syntax: CPUCTRL'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is local-line.

Description:

The **CPUCTRL** command is used to switch the I Set value to the internal CPU DAC control instead of the analogue control input line.

Nothing else is affected.

Related commands: **ANACTRL**

Intentionally blank

DA - Write to DAC's

Command: DA'sp'ch'sp'val'cr'

ch: ASCII digit 0 to4
val: digits 000000 to ±999999 in PPM.

!!!! Please note, that the DA value always must be entered using the trailing zero notation regardless of the WA setting.

ch:4 val: digits 000000 to ±999999 in PPM..

Example: DA 0 480
Syntax: DA'sp'0'sp'480'cr'
Means a set value of 480 ppm

Answer: digits 000000 to ±999999 Depending on the DAC channel

Errors: **SYNTAX ERROR**, means a missing space between the command and the parameter or wrong syntax.

DATA CONTENTS, means that parameter format is incorrect, or a non-digit character found in the data field, or parameters is outside specification.

ILLEGAL COMMAND, Indicates that you are in a wrong command mode. Change to REMote or LOCal.

CHANGE IN PROGRESS, Indicates that the controller is in the middle of an internal sequence eg. a polar change. While this is running, it is a new DAC setup is not allowed.

VALUE IS LIMITED, means that value is above or below the set limits. If below min. limit, the value is automatic set to minimum and if above max limit, it is set to maximum. In both cases this error message is sent.

DA continued.

DAC OWNED BY _____ , (WA) Indicates who owns the DAC if new DAC setup is issued while DAC is running. While DAC is running, new DAC setup is not allowed. RAMP0 (Auto slewrate esc<slopetime), RAMP1(Arbitrary point), RAMP2(Equal time slot), slewrate, polarity switch and external interface can all own the DAC.

Description:

The DA command is a new alternative to the **WA** command.

DA 0 equals WA.	- Sets output current in ppm
DA 1	- Set current slewrate (Positive values only)
DA 2	- Set voltage slewrate (Positive values only)
DA 3 equals	- Optional
DA 4	- Sets output voltage in ppm (Positive values only)

Current slew rate is determined by following formula :

$$t_{SR} = \frac{26.4 * 10^3}{DA\ 1}$$ where DA 1 is from 0 to 250000 and tsr is time from 0% to 100%

Voltage slew rate is determined by following formula :

$$t_{SR} = \frac{27 * 10^3}{DA\ 1}$$ where DA 2 is from 0 to 250000 and tsr is time from 0% to 100%

DA continues on next page

DA continued

Description:

See also the DA description on the previous page.

The amendment gives the possibility to ad a sign to the set value for ch-0 when controlling a bipolar power supply (Attached to a bipolar DAC).

Function in Uni Polar mode. (Aux switch 8 =”0" Default)

- MPS without a polarity change over switch (motor).
  Any sign will be ignored.

- MPS with a polarity change over switch (motor).
  Giving a set value with the opposite sign than the present one will automatically initiate a polarity change over operation.

Function in Bipolar mode. (Aux switch 8 =”1")

- The set value will follow the signed value.

The ‘+’ & ‘-‘ led’s on the M-Panel will indicate the polarity status.

The ‘+’ & ‘-‘ buttons on the M-Panel will change the polarity status. (‘PO +’ or ‘PO - ‘ commands in remote control)

Reading the set value with the **DA 0** (without parameters) will automatically ad a minus sign to the value, if the output polarity is negative.

Depending on the polarity status may the PO status be affected.

Nothing else is affected.

Related commands: **RA, WA,**

DA - Read DAC’s

Command: DA'sp'ch'cr'

ch: ASCII digit 0 to4 (WA)

Example: DA 0
Syntax: DA'sp'0"cr'
Means read value of channel 0

Answer: digits 000000 to ±999999

Errors: **SYNTAX ERROR**, means a missing space between the command and the parameter or wrong syntax.

ILLEGAL COMMAND, Indicates that you are in a wrong command mode. Change to REMote or LOCal.

ERRC - ERRor in Code (remote line only)

Command: ERRC'cr'

Example: ERRC
Syntax: ERRC'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The command **ERRC** is used to put the controller into a mode, in which all errors will respond with a code number representing, which error was encountered.
This mode can chosen, when the controller(s) is (are) connected to a host computer, which is able to interpret the error message.

Nothing else is affected.

Related commands: **ERRT, NERR**

ERRC continued

CODE NO.	ERROR TEXT
0	EE Error buffer empty
1	Syntax
2	DATA CONTENTS,
3	Data length
4	Illegal command
5	Can not execute command
6	Status quo, no change
7	Change in progress
8	No data present
9	Local line, input buffer full
0	Remote line, input buffer full
11	NOT USED
12	Can not execute command
13	NOT USED
14	Datalog line, input buffer full
15	NOT USED
16	Program module not implemented
17	Busy
18	DAC owned by slewrate
19	DAC owned by polarity switch
20	DAC owned by Ramp 1
21	DAC owned by Ramp 2
22	DAC owned by External interface
23	DAC owned by Ramp0
24	Value is limited

ERRT - ERRor in Text (remote line only)

Command: ERRT'cr'

Example: ERRT
Syntax: ERRT'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The command **ERRC** puts the controller into a mode, in which all errors will respond with a text string representing, which error was encountered.
This mode is normally chosen, when the controller(s) is (are) connected to a low level host computer or terminal equipment.

Nothing else is affected.

Related commands: **ERRC, NERR**

F - ofF

Command: F'cr'

Example: F
Syntax: F'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR**, means wrong syntax

Description:

The **F** command is used for switching-off the power supply (main contactor).
If the OFF and RESET commands both are set to clear non active interlocks, it also clears these interlocks.

All settings are left unaffected.

Nothing else is affected.

Related commands: **N, RS**

ID - Identification read from ver 1.4

Command: ID 'cr'

Example: ID
Syntax: ID 'cr'

Answer: SYSTEM 9100
TYP 13 200A/50V

or Error message

Errors: **ILLEGAL COMMAND** means that line-in-command is wrong.

Description:

The **ID** command is used to read a user set able text string of maximum 64 characters. The information stored could be information about the power supply. The default txt is: (Ps. All characters are converted to upper case)

To write information in identification field, see ESC<ID command.

Related commands: **ESC<ID**

Affected commands: **NONE**

LALL - Listen ALL (remote line only)

Command: LALL'cr'

Example: LALL
Syntax: LALL'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: SYNTAX ERROR, means wrong syntax.

Description:

The **LALL** command puts all the connected controllers into a pseudo-addressed mode. This means, that all controllers will respond to any setup command regardless of its addressed state, except for the o**N** command.
No answers will be available.

The only way to disable the **LALL** mode is by using an **ADR** command, either for a new address or to read the last addressed controller.

Remark: Concerning the **ADR** read, the first access will not give any response at all, in this case a second **ADR** command has to be issued to get an answer.

Nothing else is affected.

Related commands: **ADR**

LOC - LOCal (line)

Command: LOC'cr'

Example: LOC
Syntax: LOC'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means wrong syntax.

Description:

The **LOC** command switches the line-in-command to the local line. The line-in-command can be locked to local-line by the **LOCK** command and released by the **UNLOCK** command.
If the change to local is done from the local-line (control panel), the line-in-command will automatically be **LOCK**ed to local, and can't be changed back from the remote line without releasing it with the **UNLOCK** command. A change to the remote line initiated from the control panel automatically releases the lock state.

Nothing else is affected.

Related commands: **REM, LOCK, UNLOCK**

Affected commands: **REM**

LOCK - LOCK (remote line only)

Command: LOCK'cr'

Example: LOCK
Syntax: LOCK'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means wrong syntax.

Description:

The **LOCK** command puts the controller into a mode, in which the line-in-command will be locked to the local line. The **LOCK** state is entered automatically, when shift to the local state is initiated from the control panel. From the remote line the **LOCK** state can only be entered by issuing the **LOCK** command.
The **LOCK** feature is to avoid remote access, when serviced and controlled locally through the control panel. The **UNLOCK** command from remote line is implemented for one reason only: to be able to shut down the entire system in an emergency situation. One should avoid using the **LOCK** and **UNLOCK** feature, from the remote line except in an emergency situation.

Nothing else is affected.

Related commands: **UNLOCK, (REM, LOC, RLOCK)**

N - oN

Command: N'cr'

Example: N
Syntax: N'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR**, means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

The **N** command switches on the power supply (main contact). All settings are left unaffected.
This command cannot be used in **LALL** mode.

Nothing else is affected.

Related commands: **F, RS**

NASW - No AnSWer

Command: **NASW'sp'ch'cr'**

ch: ASCII digit 0 or 1 1=local; 0=remote

Example: NASW 0
Syntax: NASW' sp'0'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR**, means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is l

Description:

The **NASW** command is used to cancel the auto-answer mode, in which some setup commands generates an answer.

Nothing else is affected.

Related commands: **ASW**

NERR - No ERRor (remote line only)

Command: NERR'cr'

Example: NERR
Syntax: NERR'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR**, means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The command **NERR** puts the controller into a mode, in which all errors only will respond with a "?" + "Bell" without showing what kind of error was encountered.
This mode is normally chosen, if one only wants to be kept informed about an error condition, but not interested in the type.

Nothing else is affected.

Related commands: **ERRC, ERRT**

PO - POlarity (read)

Command: PO'cr'

Answer: polarity

or Error message

Example: Command: PO
Syntax: PO'cr'

Answer: polarity
Syntax: polarity'lf''cr'

olarity: ASCII sign plus or minus. (+ or -)

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that l

Description:

The **PO** command verifies the actual output polarity of the power supply if a polarity reversal switch is attached or the power supply is in bipolar mode.
The command returns the polarity sign as an ASCII character.

If there is no polarity switch build-in, the returned polarity will be positive.

Related commands: **PO** (write), **WA** ±val, **DA 0** ±val, **DA 0**

Nothing else is affected.

PO {+/-} - POlarity (write)

Command: PO sign'cr'

sign: ASCII sign plus or minus. (+ or -)

Example: PO +
Syntax: PO +'cr'

Answer: no answer, except errors
or OK if autoanswer mode is set

Errors:

SYNTAX ERROR,	means wrong syntax.
ILLEGAL COMMAND,	means that line-in-command is wrong or no polarity switch build-in.
DATA CONTENTS,	means that the sign is neither plus nor minus.
STATUS QUO,	means that the desired polarity is already present.

Description:
The **PO** command changes polarity of the power supply, either if a polarity reversal switch is implemented or when using a bipolar DAC (on bipolar supplies).
Having a polarity reversal switch, the command starts an internal state-machine, that sets the set value to zero, waits until the current gets to zero, switches the supply OFF, changes the polarity, restores the set value and at last switches the supply on again. If the power supply was OFF, then it will stay OFF after the polarity change is over.
For bipolar supplies, the polarity will be changed without turning the power supply OFF first.
Preferred command is DA 0,±val. Using this command, it is possible to change to a different negative/positive value.

If no polarity switch is attached or if not set in bipolar mode, an illegal coomand error will be returned if issued.
Related commands: **PO** (read), **WA** ±val, **DA** ±val

Nothing else is affected.

POLOOL - POLarity Output Off Limit

Command: POLOOL'sp'val'cr'

Val: 000 to 100 in % of full scale

Example: POLOOL 1
Syntax: POLOOl'sp'1'cr'

Answer: 1'cr'
cr'

or Error message

Errors:

SYNTAX ERROR,	means wrong syntax.
ILLEGAL COMMAND,	means that line-in-command is wrong.

Description:

The **POLOOL** command specifies the Output current limit for MPS off in polarity switch operating mode. The Magnet Power Supply switches off if the output current is below this value, and the Polarity start to change polarity of the power supply. Default value is 1 (1 % of full scale).

The command can be used at the remote-line only.

Nothing else is affected.

Related commands: **PO, Esc<POLDELAY**

PRINT - PRINT

Command: PRINT'cr'

Example: PRINT
Syntax: PRINT'cr'

Answer: Two lines each containing 15 characters plus terminator as:
xxxxxxxxxxxxxxx'cr'
xxxxxxxxxxxxxxx'cr'

or Error message

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **PRINT** command returns internal information about the unit. The contents of these two lines may differ between the power supplies depending of the SW version.

The command can be used at the remote-line only.

Nothing else is affected.

RA - Read DAC

Command: RA'cr'

Answer: dac'lf''cr'

dac: digit 000000 to 999999

or Error message

Example: Command: RA
Syntax: R1'cr'

Answer: 004800
Syntax: 004800'lf''cr'

Errors: **SYNTAX ERROR**, means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **RA** command reads the currently numerical set value in ppm resolution. That is between 0 and 999999 from the regulation-DAC. Use the PO command to read the polarity status.

Preferred command is though DA 0. Using DA 0 will automatically deliver the present polarity status.

Nothing else is affected.

Related commands: **WA, DA**

REM - REMote (line)

Command: REM'cr'

Example: REM
Syntax: REM'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is locked to local-line. Unlock can be used to release this. --

Description:

The **REM** command switches the line-in-command to the remote operation. The line-in-command can be locked to remote-line by the **RLOCK**, command (given from the remote-line). The locked state can be released by a LOC command, also given from the remote-line. The Local-line cannot change the command-line if locked into remote.

Nothing else is affected.

Related commands: **LOC, LOCK, UNLOCK, RLOCK**

Affected commands: **LO**

RLOCK - Remote LOCK (remote line only)

Command: RLOCK'cr'

Example: RLOCK
Syntax: RLOCK'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR**, means wrong syntax.

ILLEGAL COMMAND, means that the line-in-command is either remote line or unlocked in local line.

Description:

The **RLOCK** command locks the line-in-command to the remote state. The **RLOCK** is similar to the function existing, when line-in-command is switched to local by the local line. When the **RLOCK** command is given from the remote line, it will inhibit the control panel to switch the line-in-command to local.
The **RLOCK** can only be switched off with the **REM** or **LOC** command.

Nothing else is affected.

Related commands: **(LOCK, REM, LOC, UNLOCK)**

RS - ReSet

Command: RS'cr'

Example: RS
Syntax: RS'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **RS** command clears all non pending interlock's.
If Reset and Off are combined will the O**F**F command also clear all the non pending interlock's.

Nothing else is affected.

Related commands: **F, N**

RWR - Read Writing-Slewrate for polarity switch

from ver 1.02

Command: RWR'cr'

Example: RWR
Syntax: RWR'cr'

Answer: value [1…..10000]

Errors: **SYNTAX ERROR,** means a missing space between the command and parameter or wrong syntax.

Description:

The **RWR** command read the set value for slewate used in polarity switch. The value range is from 1 to 10000 and is defined as 0.01% of full scale per 10ms i.e.

Slewrate time = (100/(value*0.01))*10ms

It means that if value=1 then slewrate time =100sec.

Slewrate time is in the range 10ms to 100sec.

Nothing else is affected.

Related commands: **WR**

S1 - Status 1

Command: S1'cr'

Answer: STATUS
Syntax: STATUS'lf''cr'
Where STATUS consists of 24 signs "." or "!" , each showing the status of a specific function, including all interlocks.

or Error message

Example: Command: S1
Syntax: S1'cr'

Answer: . ! ! . . . . . !
Syntax: . ! ! . . . . . ! 'lf''cr'

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **S1** command returns an answer about the internal status. The returned status line consists of a mixture of interlocks (latched indications) and status (transparent indications).

Spare bits can be assigned to special functions in some power supplies.

Each sign is explained separately at the right column.

Nothing else is affected.

Related commands: **SIH, S3,**

S1 continued

A typical example for an S1 status answer could be as follows:

"! . ! . ! ! . . . ! . . . . ! . . . . . ! ."
1. Character. 24. Character.

The interpretation of the individual characters, when the exclamation mark is shown are:

CHARACTER NO. CONTENTS

1 . . . . . MAIN POWER OFF (!=OFF .=ON)
2 . . . . . POLARITY NORMAL (!=Polarity Normal)
3 . . . . . POLARITY REVERSED (!=Polarity REVERSED)
4 . . . . . NOT USED
5 . . . . . CROWBAR ON (!=ON .=OFF)
6 . . . . . I-MODE (!=I-mode .=V-mode)
7 . . . . . != % , . = AMPS and VOLTS
8 . . . . . EXTERNAL INTERLOCK 0 (!=Interlock .=No interlock)
9 . . . . . NOT USED.
10 . . . . . SUM – INTERLOCK (!=Sum interlock .=No sum interlock)
11 . . . . . OVER VOLTAGE (OVP) (!=over voltage .=No over voltage)
12 . . . . . DC OVER CURRENT (OCP) (!=over current .=No over current)
13 . . . . . DC UNDERVOLTAGE (!=Fault .=OK)
14 . . . . . NOT USED
15 . . . . . PHASE FAILURE (AC LINE OK) (!=Fault .=OK)
16 . . . . . NOT USED
17 . . . . . EARTH LEAKAGE (!=Fault .=OK)
18 . . . . . FAN (!=Fault .=OK)
19 . . . . . MPS OVERTEMPERATURE (!=Fault .=OK)
20 . . . . . EXTERNAL INTERLOCK 1 (!=Interlock .=No interlock)
21 . . . . . EXTERNAL INTERLOCK 2 (!=Interlock .=No interlock)
22 . . . . . EXTERNAL INTERLOCK 3 (!=Interlock .=No interlock)
23 . . . . . MPS NOT READY (!=Not ready .=Ready)
24 . . . . . NOT USED

S1H - Status 1 in Hex

Command: S1H'cr'

Answer: STATUS
Syntax: STATUS'lf''cr'
Where STATUS consists of 6 ASCII Hex digits, each position showing the status of a specific function, including all interlocks.

Example: Command: S1H
Syntax: S1H'cr'

Answer: 600001
Syntax: 600001'lf''cr'

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **S1H** command returns an answer about the internal status in HEX format. The returned status line consists of a mixture of interlocks (latched indications) and status (transparent indications).

The HEX format is constructed from the 24 bit in the S1 status. These bits are divided into 6 nibbles and thereafter converted into six ASCII HEX digits. The individual bit placements in the HEX number are the same as for the S1 command. Please refer to the next column and to the S1 command for a detail bit definition.

Spare bits can be assigned to special functions in some power supplies.

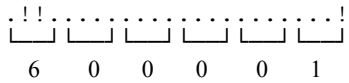
Each sign is explained separately under the S1 command.

Nothing else is affected.

Related commands: **S1**

S1H continued

HEX conversion examples.



will be represented as in HEX

└─┘ Equals one nibble.

S1FIRST - S1 first status

Command: S1FIRST'cr'

Answer: STATUS
Syntax: S1FIRST'lf''cr'
Where S1FIRST consists of 24 signs "." or "!", each position showing the status of a specific function, including all interlocks at the time an interlock occurred

or Error message

Example: Command: S1FIRST
Syntax: S1FIRST'cr'

Answer: . ! ! . . . . . !
Syntax: . ! ! . . . . . ! 'lf''cr'

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **S1FIRST** command returns an answer about the state of all internal status at the time the first interlock occurred including the first interlock. The returned status line consists of a mixture of interlocks (latched indications) and status (transparent indications).

Spare bits can be assigned to special functions in some power supplies.

Each sign is explained separately at the right column.

Nothing else is affected.

Related commands: **S1, S1FIRSTH**

S1FIRSTH - S1 first status in Hex

Command: S1FIRSTH'cr'

Answer: STATUS
Syntax: S1FIRSTH'lf''cr'
Where S1FIRST consists of 6 Hex digits, each position showing the status of a specific function, including all interlocks at the time an interlock occurred

or Error message

Example: Command: S1FIRSTH
Syntax: S1FIRSTH'cr'

Answer: 640001
Syntax: 640001 'lf''cr'

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **S1FIRSTH** command returns an answer about the state of all internal status at the time the first interlock occurred including the first interlock. The returned status line consists of a mixture of interlocks (latched indications) and status (transparent indications).

The HEX format is constructed from the 24 bit in the S1 status. These bits are divided into 6 nibbles and thereafter converted into six ASCII HEX digits. The individual bit placements in the HEX number are the same as for the S1 command. Please refer to the next column and to the S1 command for a detail bit definition.

Spare bits can be assigned to special functions in some power supplies.

Each sign is explained separately at the right column.

Nothing else is affected.

Related commands: **S1, S1FIRST**

S3 - Status 3

Command: S3'cr'

Answer: STATUS
Syntax: STATUS'lf''cr'
Where STATUS consists of 16 signs "." or "!" , each showing the status of a specific function, including all interlocks.

or Error message

Example: Command: S3
Syntax: S3'cr'

Answer: . ! ! . . . . . .
Syntax: . ! ! . . . . . 'lf''cr'

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **S3** command is used to return an answer about the extended internal status. The returned status line consists of a mixture of interlocks (latched indications) and status (transparent indications).

Spare bits can be assigned to special functions in some power supplies.

Each sign is explained separately at the right column.

Nothing else is affected.

Related commands: **S1, S3, S4**

S3 continued

A typical example for an S3 status answer could be as follows:

"! . ! . . . . . "
1. Character. 16. Character.

The interpretation of the individual characters, when the exclamation mark is shown are:

CHARACTER NO. CONTENTS.

1 . . . . . NOT USED
2 . . . . . NOT USED
3 . . . . . NOT USED
4 . . . . . NOT USED
5 . . . . . NOT USED
6 . . . . . NOT USED
7 . . . . . NOT USED
8 . . . . . NOT USED
9 . . . . . NOT USED
10 . . . . . NOT USED
11 . . . . . NOT USED
12 . . . . . NOT USED
13 . . . . . NOT USED
14 . . . . . NOT USED
15 . . . . . NOT USED
16 . . . . . INT/EXT INTERFACE (!=EXTERNAL (SPI board) .=INTERNAL (Control board))

S3H - Status 3 in Hex

Command: S3H'cr'

Answer: STATUS

Syntax: STATUS'lf''cr' Where STATUS consists of 4 ASCII Hex digits, each position showing the status of a specific function.

or Error message

Example: Command: S3H
Syntax: S3H'cr'

Answer: 6001
Syntax: 6001'lf''cr'

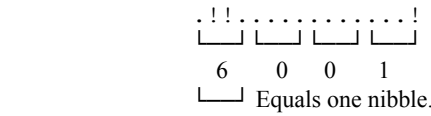
Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **S3H** returns an answer about internal status signals in HEX format.

The HEX format is constructed from the 16 bit in the S3 status. These bits are divided into 4 nibbles and thereafter converted into four ASCII HEX digits. The individual bit placements in the HEX number are the same as for the S3 command.



will be represented in HEX as

Each sign is explained separately under the S3 command. Spare bits can be assigned to special functions in some power supplies.

Nothing else is affected.

Related commands: **S3, S1H**

TD - Test DAC

Command: TD'sp'set'cr'

set : ASCII digit 0 to 9 and H and L

Example: TD 4
Syntax: TD'sp'4'cr'

Answer: No answer, except errors

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The command sets up a certain bit-pattern to the DAC output. The command is used for adjusting or verifying the functionality of the DAC. The different patterns are:

TD 0: Full negative voltage (-10V)
TD 1: 90% negative voltage (-9V)
TD 2: 50% negative voltage (-5V)
TD 3: 0 (0V)
TD 4: 50% positive voltage (+5V)
TD 5: 90% positive voltage (+9V)
TD 6: Full positive voltage (+10V)
TD 7: Reserved
TD 8: Reserved
TD 9: Not Used
TD H: Set DAC1 (LSB) (bit0-14 =0) and DAC2 (MSB) bit16=1, bit17-19=0
TD L: Set DAC1 (LSB) (bit0-14 =1) and DAC2 (MSB) bit16-19=0

Previous **WA, DA** and **RA** setting are affected. Make a new WA command to correct this.

TYPE - TYPE

Command: TYPE'cr'

Example: TYPE
Syntax: TYPE'cr'

Answer: T'sp'type'lf''cr'

type: ASCII digit 0 or 8
If 0, you will have no 16 bit ADC
If 8, you will have the 16 bit ADC

or Error message

Description:

The **TYPE** command returns a code, used by the control panel to determine the type of the AD-channel used for the current read back.. If a 0 is returned, it will use AD channel 0 to read-out a 3-digit current value (8 bit resolution). If an 8 is returned, it will use AD channel 8 to read-out a 5-digit current value (16 bit resolution).

!! This command can only be used at the control-line. !!

Nothing else is affected

Intentionally blank

UNLOCK - UNLOCK remote line only)

Command: UNLOCK'cr'

Example: UNLOCK
Syntax: UNLOCK'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR**, means wrong syntax.

ILLEGAL COMMAND, means that the line-in-command is either remote line or unlocked in local line.

Description:

The **UNLOCK** command releases the local **LOCK** state. The **LOCK** prevents access when controlled from the control panel. The **UNLOCK** command given from remote line can be used to remotely shut down the entire system in an emergency situation. One should normally avoid using the **UNLOCK** feature from the remote line except in an emergency situation.

Nothing else is affected.

Related commands: **LOCK, (REM, LOC, RLOCK)**

VER - VERsion

Command: VER'cr'

Example: VER
Syntax: VER'cr'

Answer: Three lines each containing 23 characters plus terminator as:
xxxxxxxxxxxxxxxxxxxxxxx'cr'
xxxxxxxxxxxxxxxxxxxxxxx'cr'
xxxxxxxxxxxxxxxxxxxxxxx'cr'

or Error message

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-.

Description:

The **VER** command returns internal information about the program and its version. The contents of these three lines depend on the program manufacturer, who programs this field with copyright notes, the actual version number and release date.
The command was designed primarily as a service command, to get information about the internal program.

The command can be used at the remote-line only.

Nothing else is affected.

WA - Write DAC in ppm

Command: WA'sp'dac'cr'

dac: digits 000000 to 999999 in PPM.

!!!! Please note, that either a leading zero or a trailing zero format can be used by the power supply depending of the initial setup mode. Default factory setting is leading zeroes. (Read leading or trailing as important zeroes)

Example: WA 0480
Syntax: WA'sp'0480'cr'
For leading zeroes this means 48000 ppm
For trailing zeroes this means 480 ppm

Answer: No answer, except errors.

or OK if autoanswer mode is set

Errors:

SYNTAX ERROR, means a missing space between the command and the parameter or wrong syntax.

DATA CONTENTS, means that parameter format is incorrect, or a non-digit character found in the data field, or parameters is outside specification.

ILLEGAL COMMAND, Indicates that you are in a wrong command mode. Change to REMote or LOCal.

CHANGE IN PROGRESS, Indicates that the controller is in the middle of an internal sequence eg. a polar change. While this is running, it is a new DAC setup is not allowed.

VALUE IS LIMITED, means that value is above or below the set limits. If below min. limit, the value is automatic set to minimum and if above max limit, it is set to maximum. In both cases this error message is sent.

WA continued.

DAC OWNED BY _____ , Indicates who owns the DAC if new DAC setup is issued while DAC is running. While DAC is running, new DAC setup is not allowed. RAMP0 (Auto slewrate esc<slopetime), RAMP1(Arbitrary point), RAMP2(Equal time slot), slewrate, polarity switch and external interface can all own the DAC.

.

Description:

The command **WA** is used to write a PPM value between 0 and 999999 to the regulation module DAC for output set current. Preferred command is though DA 0.

Nothing else is affected.

Related commands: **DA, RA**

WA - Write ±DAC in ppm

Command: WA'sp'dac'cr'

dac: digits ±000000 to ±999999 in PPM.

!!!! Please note, that either a leading zero or a trailing zero format can be used by the power supply depending of the initial setup mode. Default factory setting is leading zeroes. (Read leading or trailing as important zeroes)

Example: WA 0 -0480
Syntax: WA'sp'0'sp'-0480'cr'
For leading zeroes this means -48000 ppm
For trailing zeroes this means - 480 ppm

Answer: No answer, except errors.

or OK if autoanswer mode is set

Errors:

SYNTAX ERROR, means a missing space between the command and the parameter or wrong syntax.

DATA CONTENTS, means that parameter format is incorrect, or a non-digit character found in the data field, or parameters is outside specification.

ILLEGAL COMMAND, Indicates that you are in a wrong command mode. Change to REMote or LOCal.

CHANGE IN PROGRESS, Indicates that the controller is in the middle of an internal sequence eg. a polar change. While this is running, it is a new DAC setup is not allowed.

VALUE IS LIMITED, means that value is above or below the set limits. If below min. limit, the value is automatic set to minimum and if above max limit, it is set to maximum. In both cases this error message is sent.

DAC OWNED BY _____ , Indicates who owns the DAC if new DAC setup is issued while DAC is running. While DAC is running, new DAC setup is not allowed. RAMP0 (Auto slewrate esc<slopetime), RAMP1(Arbitrary point), RAMP2(Equal time slot), slewrate, polarity switch and external interface can all own the DAC.

(WA)

Description:

See also the WA description on the previous page.

The amendment from SW version SCY 1.xx gives the possibility to add a sign to the set value when controlling a bipolar power supply. No prefixed sign equals no change in the present polarity. That is; if the output current was -500000 and a “WA 600000" was issued then the output current will become -600000. If otherwise a “WA +600000" was issued then the polarity will be altered and the output current become +600000..

Function in Uni Polar mode. (Aux switch 8 =”0" Default)

- MPS without a polarity change over switch (motor).
  Any sign will be ignored.

- MPS with a polarity change over switch (motor).
  Giving a set value with the opposite sign than the present one will automatically initiate a polarity change over operation. .

Function in Bipolar mode. (Aux switch 8 =”1")

- The set value will follow the signed value.

The ‘+’ & ‘-‘ led’s on the M-Panel will indicate the polarity status.

The ‘+’ & ‘-‘ buttons on the M-Panel will change the polarity status. (‘PO +’ or ‘PO - ‘ commands in remote control)

Preferred command is though DA 0,val. Using DA 0 will automatically deliver the present polarity status.

Depending on the polarity status the PO status may be affected.
Related commands: **RA, DA**

WR - Write-Slewrate for polarity switch

Command: WR'sp'[value]'cr'

Value: Integer number between 1 to 10000

Example: WR
Syntax: WR'sp' 100 'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means a missing space between the command and parameter or wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **WR** command sets value for slewate used in polarity switch. The value range is from 1 to 10000 and is defined as 0.01% of full scale per 10ms i.e.

Slewrate time = (100/(value*0.01))*10ms

It means that if value=1 then slewrate time =100sec.

Slewrate time is in the range 10ms to 100sec.

Nothing else is affected.

Related commands: **RWR**

Intentionally blank

Esc<AD - AD setup

Command: 'Esc'<AD'sp'ch,scale_factor,no_Offset,digits,format,ch_reroute,Dot'cr'

ch:	ASCII digit 0 to 15
Scale_factor:	9 digit floating point value plus sign.
Offset:	5 digit floating point value plus sign.
No_digit:	1 to 6
format:	A (Absolute), D (Signed) or U (Unsigned)
ch_reroute:	ASCII digit 0 to 15
Dot*:	ASCII digit 1 to 3

*Dot is only valid for channel 11 and channel 12, dot-parameter will be ignored by other channels. Default value is 1 which means dot on second lsb LED segment. Value 3 means dot on msb LED segment.

Example: 'Esc'<AD 0,101.0025,0,3,A,0
Syntax: AD'sp'0,101.0025,0,3,a,0'cr'

or

'Esc'<AD 0,101.0025
Syntax: AD'sp'0,101,0025'cr'
Non entered parameters are not changed.

Answer: No answer except errors.
or OK if autoanswer mode is set

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means that the parameter format is incorrect or a non-digit character is found in the data field or a parameter is outside the specification.

Description:

The 'Esc'<AD command scales and/or reroutes any of the 15 AD channels. Gain adjustments are also preformed with this command.

All AD channels are internally normalized to 1 for full scale before multiplied with the scale factor positive or negative. This simplifies the scale factor to be nearly the same as the desired max output reading for most applications.
The number of digits returned is also programmable between 1 and 6. But not to disturb the operation of the M-Panel, please do not change the number of digits for the first 9 channels.

The number of digits also defines the max hold value during overflow. For example if no. of digits =3 then 999 is displayed as overflow.

On bipolar supplies a format parameter can issued to always return a positive value.
The "A" format converts the AD value to an absolute value. (-1 to 1)☐ (1 to 0 to 1)

The "D" format converts the AD value to an signed value. (-1 to 1)☐ (-1 to 1)

The "U" format converts the AD value to an unsigned value. (-1 to 1)☐ (0 to 1)

Rerouting a channel can for example be used to display the water flow on the M-Panel display instead of the Tesla. Following example show this:
'Esc'<AD 13,25,3,U,1
Issuing an {'Esc'<AD ch} without any parameters will return the present channel setting. The setting becomes first operational after a processor reset or a mode switch update. Nothing else is affected.

CHANNEL	VALUE	AD port	
0	Output current	16 bit ext.	3
1	Ambient Temperature	16 bit ext.	3
2	Output Voltage	16 bit ext.	3
3	Internal +15V sup.	10 bit int.	3
4	Internal -15V sup.	10 bit int.	3
5	Internal +5V sup.	10 bit int.	3
6	DeltaTemperature	10 bit int.	2
7	I set value	16 bit ext.	3
8	Optional Iout (16 Bit)	16 bit ext.	5
9	Iout (16 Bit)(Crtl panel).	mirror ch0 or ch8	3
10	Not used.		5
11	Output current for display	16 bit ext.	4
12	Output voltage for display	16 bit ext.	4
13	Not used.		3
14	Not used.		3
15	Not used.		3
16	Internal +5Vana.	10 bit int.	3
17	I set value	16 bit ext.	3
18	Spare	16 bit ext.	3

Esc<ADR - ADdRess setup write

Command: 'Esc'<ADR'sp'ch,address

ch: ASCII digit 0 to 1 1=local; 0=remote
Address: ASCII digit 0 to 255

Example: 'Esc'<ADR 0,10
Syntax: 'Esc'<ADR'sp'0,10'cr'

Answer: No answer except errors

or OK if autoanswer mode is set

Errors: **ILLEGAL COMMAND** means that line-in-command is wrong.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means that the parameter format is incorrect or a non-digit character is found in the data field or a parameter is outside the specification.

Description:

The 'Esc'<ADR command configures the communication line address.
The setting becomes first operational after a processor reset or a mode switch update.

Giving the 'Esc'<ADR'sp'ch, without any address will return the current setting.

Default address is 0 for both local and remote line.

Esc<ADR - ADdRess setup read

Command: 'Esc'<ADR'sp'ch

ch: ASCII digit 0 to 1 1=local; 0=remote

Example: 'Esc'<ADR 0
Syntax: 'Esc'<ADR'sp'0'cr'

Answer: 'ch,address

or Error message.

Errors: **ILLEGAL COMMAND** means that line-in-command is wrong.

SYNTAX ERROR, means a missing space between the command and and parameter or wrong syntax.

DATA CONTENTS, means that the parameter format is incorrect or a non-digit character is found in the data field or a parameter is outside the specification

Description:

The 'Esc'<ADR read command verifies the programmed line address.

Esc<ADSET - Analog to Digital converter SETting

Command: 'Esc'<ADSET ch,val1,val2'cr'

ch: ASCII digit 0 to 15
val1: Z, F, B or T
val2: ASCII 00000 to 99999

Example: 'Esc'ADSET 8,F,999999
Syntax: 'Esc'ADSET 8,F,999999'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

DATA CONTENTS, means that parameter format is incorrect or a non-digit character is found in the data field or the parameter is outside the specification.

Description:

The **ADSET** command automatically Gain and Offset adjust a specific AD channel to display a certain value equals to "val2". If val2 is not given, zero will be read for the Offset - and the factory default for the gain value.

! Be aware, there is no syntax check on the values. Wrong values may give meaningless output readings. !

ADSET continued

val1 interpretation

Z: Offset adjustment (to Zero)
F: Gain adjustment (To full scale)
B: Restores the Offset value to the factory default. (Bottom)
T: Restores the Gaint value to the factory default. (Top)

Related commands: **'Esc'<DASET, 'Esc'<AD**

Affected commands: **NONE**

Esc<AUX- AUXiliary setup write

Command: 'Esc'<AUX'sp b1,b2,b3,b4,b5,b6,b7,b8

bx:: ASCII 0 or 1

Example: 'Esc'<AUX 0,0,1,1
Syntax: 'Esc'<AUX'sp'0,0,11'cr'

Current setting is kept for non entered bits.

Answer: No answer except errors.
or OK if autoanswer mode is set

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means that the parameter format is incorrect or a non-digit character is found in the data field or a parameter is outside the specification.

Description:

The 'Esc'<AUX command defines the different auxiliary setting
The setting becomes operational immediately after then saving in the EEPROM..

The bit definition is illustrated at the right of this page and on the dip switch setting chapter.

Setting of bit 16 and 17 is intended for Offset DAC use. That is a 16 bit setting between 88 and 96% (Bit16=0 & BIT 17=1) or between 92% and 100% output current (Bit16=1 & BIT 17=1). For linear DAC settings, please set bit 16 & 17 to transparent mode.

Using the "WA" command a leading zeroes or trailing zeroes input format can be chosen.
- For leading zeroes: "WA 123" equals "WA 000123"
- For trailing zeroes: "WA 123" equals "WA 123000"

Giving the 'Esc'<AUX without any parameter will return the current setting.

Esc<AUX continued

b1 :	DAC 16:	**0**: Transparent	1: = 1	0: = 0	1: = 1
b2 :	DAC 17:	**0**:	0: = 0	1: = 1	1: = 1
b3 :	Interlock clear	**0**:RS resets Interlocks		1:RS and OFF resets interlocks	
b4:	WA zeroes	**0**:WA uses trailing zeroes		1:WA uses leading zeroes	
b5:	Display Units	0:Display in V and A		**1**:Display in %	
b6:	SPI mode	**0**: HW detect*	0: HW detect*	1: SW ctrl ext	1: SW ctrl int
b7:	SPI mode	**0**:	1:	0:	1:
b8:	Uni/BI-polar (read only)	**0**:Uni polar DAC		1:Bipolar DAC	

Those in bold are the default setting.

*Hardware detection of the SPI mode. If SPI module is inserted, then the mode is external, see S3 command for status

Esc<AUX - AUXiliary setup read

Command: 'Esc'<AUX'

Example: 'Esc'<AUX
Syntax: 'Esc'<AUX'cr'

Answer: b1,b2,b3,b4,b5,b6,b7,b8

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

Description:

The 'Esc'<AUX read command is used to verify the programmed auxiliary bit setup.

Esc<AUX2- AUXiliary setup write

Command: 'Esc'<AUX2'sp b1,b2,b3,b4,b5,b6,b7,b8

bx:: ASCII 0 or 1

Example: 'Esc'<AUX2 0,0,1,1
Syntax: 'Esc'<AUX2'sp'0,0,11'cr'

Current setting is kept for non entered bits.

Answer: No answer except errors.
or OK if autoanswer mode is set

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means that the parameter format is incorrect or a non-digit character is found in the data field or a parameter is outside the specification.

Description:

The 'Esc'<AUX2 command defines the different auxiliary setting
The setting becomes operational immediately after then saving in the EEPROM..

The bit definition is illustrated at the right of this page and on the dip switch setting chapter.

Giving the 'Esc'<AUX2 without any parameter will return the current setting.

Esc<AUX2 continued

b1 :	Not Used	**0**:	1:
b2 :	Not Used	**0**:	1:
b3 :	*Polarity Switch	**0**:Disable	1:Enable
b4:	S-Curve form	**0**:½Cosinus	1:Trapezoid
b5:	Not Used	**0**:	1:
b6:	Not Used	**0**:	1:
b7:	Not Used	**0**:	1:
b8:	Not Used	**0**:	1:

*Polarity Switch is only for unipolar power supply

Those in bold are the default setting.

Esc<AUX2 - AUXiliary setup read

Command: 'Esc'<AUX2'

Example: 'Esc'<AUX2
Syntax: 'Esc'<AUX2'cr'

Answer: b1,b2,b3,b4,b5,b6,b7,b8

Errors: **ILLEGAL COMMAND** means that line-in-command is wrong.

Description:

The 'Esc'<AUX2 read command is used to verify the programmed auxiliary bit setup.

Esc<BAUD - BAUD rate setup write

Command: 'Esc'<BAUD'sp'ch,baud,parity, odd/even,no_bits,stop_bits

ch:	ASCII digit 0 or 1 1=local; 0=remote
baud	ASCII 1200, 2400, 9600, 19200, 38400, 57600, 76800, 115200
parity	ASCII digit 0 or 1 0: OFF 1: ON
odd/even	ASCII digit 0 or 1 0=odd; 1=even
no_bits:	ASCII digit 7 or 8 0=8; 1=7
Stop_bits	ASCII digit 0 or 1 0=1; 1=2

Example: 'Esc'<BAUD 0,9600,0,0,0,1
Syntax: 'Esc'<BAUD'sp'0,9600,0,0,0,1'cr'

Current setting are kept for non given settings

Answer: No answer except errors.

or OK if autoanswer mode is set

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means that the parameter format is incorrect or a non-digit character is found in the data field or a parameter is outside the specification.

Description:

The 'Esc'<BAUD command is used as UART HW setup for both channels..
The setting becomes first operational after a processor reset or a mode switch update. In remote line, this can be done by the command ESC<CPURESET.

Default setting is "9600Baud, No party, 8 Bits, 1 Stop bits" for local and "115200Baud, No party, 8 Bits, 1 Stop bits" for remote line.

Giving the 'Esc'<BAUD'sp'ch, without any parameter will return the current setting.

Esc<BAUD - BAUD rate setup read

Command: 'Esc'<BAUD"sp'ch

Example: 'Esc'<BAUD 0
Syntax: 'Esc'<BAUD'sp'0'cr'

Answer: 'ch,baud,parity, odd/even,no_bits,stop_bits

1,9600,0,0,0,0

ch:	ASCII digit 0 or 1 1=local; **0**=remote
baud	ASCII 9600*****, 19200, 38400, 57600, 76800, **115200***
parity	ASCII digit 0 or 1 **0**: OFF, 1: ON
odd/even	ASCII digit 0 or 1 **0**=odd; 1=even
no_bits:	ASCII digit 7 or 8 0=**8**; 1=7
Stop_bits	ASCII digit 0 or 1 **0**=1; 1=2

or Error message

Those in bold are the default setting. *=Default for local line is 9600 and remote line is 115200

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

Description:

The 'Esc'<BAUD read command verifies the programmed baud rate setup (UART HW setup).

Esc<COLDBOOT - Wake up mode setup write

Command: 'Esc'<COLDBOOT' 'sp'b1,b2,b3,b4,b5,b6,b7,b8

bx:: ASCII 0 or 1

Example: 'Esc'<COLDBOOT 1,1,1,0,0,0,1,0
Syntax: 'Esc'<CBOOT'sp'1,1,1,0,0,0,1,0'cr'

Answer: No answer except errors.

or OK if autoanswer mode is set

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means that the parameter format is incorrect or a non-digit character is found in the data field or a parameter is outside the specification.

Description:

The 'Esc'<COLDBOOT command defines the wake up control mode. That is which line has to be active after a reset and is the multi drop feature enabled or what error response should be returned if encountering an error.

The setting becomes operational immediately after then saving in the EEPROM..

The bit definitions are illustrated at the right of this page and on the dip switch setting chapter.

Giving the 'Esc'<COLDBOOT command without any parameter will return the current setting.

In order for the new settings to take effect, it needs a reset. This can be done by the command ESC<CPURESET

Esc<COLDBOOT continued

b1 :	Remote addressing:	0:Enabled	**1**:Disabled						
b2 :	Local addressing:	0:Enabled	**1**:Disabled						
b3 :	Default line in	0:Remote	**1**:Local						
b4:	Auto answer	**0**:Disabled	1:Enabled						
b5:	ERR response	**0**: ?♫ only	1: ?♫ and ERR code		0: ?♫ and ERR text		1: ?♫ and ERR text		
b6:	ERR response								
b7:	not used								
b8:	Control mode	**0**:CPU control	1:Analog control						

Those in bold are default setting.

Commands CPUCTRL and ANACTRL switches between CPU and analog control mode but will not effect setting in b8 as the change with CPUCTRL or ANACTRL is not saved in the EEPROM.
Notice, jumpers J502 and J503 have to be open (no jumper caps). Otherwise control mode setting with Esc<COLDBOOT has no effect. For detail about jumper settings, please refer to user manual for SYSTEM 9100

Esc<COLDBOOT - Wake up mode setup read

Command: 'Esc'<COLDBOOT'

Example: 'Esc'<COLDBOOT
Syntax: 'Esc'<COLDBOOT'cr'

Answer: b1,b2,b3,b4,b5,b6,b7,b8

or Error message

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

Description:

The 'Esc'<COLDBOOT read command verifies the programmed wake up bit setup.

Esc<CPURESET - Hardware reset/CPU reset

Command: 'Esc'<CPURESET'

Example: 'Esc'<CPURESET
Syntax: 'Esc'<CPURESET'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: SYNTAX ERROR, means that there are no running sequence HALT'ed or no sequence has been triggered.

Description:

The 'Esc'<CPURESET command simply perform a CPU reset.

Esc<DA - DA setup

Command: 'Esc'<DA'sp'ch',scale_factor,Offset,no_digits,format,ch_reroute'cr'

ch:	ASCII digit 0 to 4
Scale_factor:	9 digit floating point value plus sign.
Offset:	5 digit floating point value plus sign.
No_digit:	1 to 6
format:	A (Absolute), D (Signed) or U (Unsigned)
ch_reroute:	ASCII digit 0 to 15

Example: 'Esc'<DA 3,12345,0,3,U,3
Syntax: DA'sp'3,12345,0,3,U,3'cr'

or

'Esc'<DA 3,12345
Syntax: AD'sp'0,12345'cr'
Non entered parameters are left unchanged.

Answer: No answer except errors.

or OK if autoanswer mode is set

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means that the parameter format is incorrect or a non-digit character is found in the data field or a parameter is outside the specification.

Description:

The 'Esc'<DA command scales and/or reroutes any of the present four DA channels.
All DA channels are internally normalized to 1 for full scale before multiplied with the scale factor positive or negative. This simplifies the scale factor to be nearly the same as the desired max set value for most applications.
The number of digits for the set value is also programmable between 1 and 6.

DA continued

The number of digits also defines the max hold value during overflow. For example if no. of digits =3 then 999 is maximum allowable value.

On bipolar supplies a format parameter can be issued to always return a positive value.
The "A" format converts the AD value to an absolute value. (-1 to 1)☐ (1 to 0 to 1)

The "D" format converts the AD value to an signed value. (-1 to 1)☐ (-1 to 1)

The "U" format converts the AD value to an unsigned value. (-1 to 1)☐ (0 to 1)

This command can be used if the current set value has to be in amps (or milii Amps) or to change the absolute slew rate setting for the W3 (DA 3) command.

Issuing an {'Esc'<DA ch} without any parameters will return the present channel setting.

The setting becomes operational immediately after then saving in the FRAM

Nothing else is affected.

Below are listed the available DA channels.

CHANNEL	VALUE	DA channel spec.	Equal to
0	Output current	19 bit	WA range as default
1	Current slewrate	16 bit	
2	Voltage slewrate	16 bit	
3	Not used		
4	Output voltage	16 bit	

Esc<DASET - Digital to Analog converter SETting

Command: 'Esc'<DASET ch,val1,val2'cr'

ch: ASCII digit 0 to 4
val1: Z, F, B, T, L, M or I
val2: ASCII 00000 to 999999

Example: 'Esc'DASET 0,F,999999
Syntax: 'Esc'DASET 0,F,999999'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

DATA CONTENTS, means that parameter format is incorrect or a non-digit character is found in the data field or the parameter is outside the specification.

VALUE IS LIMITED, means that value is above or below the set limits. If below min. limit, the value is automatic set to minimum and if above max limit, it is set to maximum. In both cases this error message is sent.

Description:

The **DASET** command is used to automatically Gain and Offset adjust a specific DA channel to given value "val2". If val2 is not provided, zero will be taken for the Offset - and all nines for the Gain adjustment.
! Be aware, there is no syntax check on the values. Wrong values may give meaningless output readings. !

Related commands: **'Esc'<DASET, 'Esc'<DA**

Affected commands: **NONE**

DASET continued

val1 interpretation

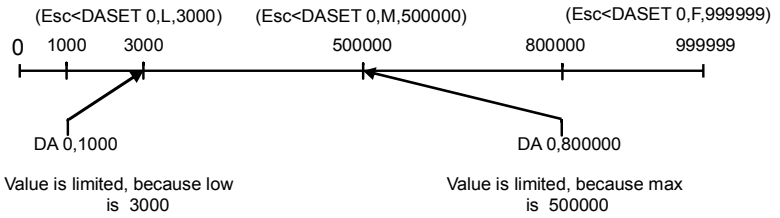
Z: Offset adjustment (to Zero)
F: Gain adjustment (To full scale)
B: Restores the Offset value to the factory default. (Bottom)
T: Restores the Gain value to the factory default. (Top)
L: Low(min) value *
M: Max value
I : Initial value after reset

* Be aware when performing the full scale adjustment 'F', the set value (DA 0,xxxxxx) can not be set higher than the max value (val2, set by the 'M' setting). That is, perform the full scale adjustment before the limit setting.
Initial value has to be higher than low set value

Example on limit setting

Esc<DASET 0,F,999999
Esc<DASET 0,M,500000
Esc<DASET 0,L,3000

DA 0,900000 ' → Value will be limited and set to 800000 with Error message "value is limited"
DA 0,1000 ' → Value will be limited and set to 3000 with Error message "value is limited"
Esc<DASET 0,I,1000 ' → Error message "value is limited". Initial value has to be higher than low set



Related commands: 'Esc'<DASET, 'Esc'<AD

Affected commands: NONE

Esc<DASET - Digital to Analog converter SETting read

Command: 'Esc'<DASET ch'cr'

ch: ASCII digit 0 to 4

Example: 'Esc'DASET 0
Syntax: 'Esc'DASET 0'cr'

Answer: val1,val2 ,val3 ,val4, val5
val1 = 000000 to ±999999 (Fullscale)
val2 = 000000 to ±999999 (Zero offset)
val3 = 000000 to ±999999 (Low/min value)
val4 = 000000 to +999999 (Max value)
val5= 000000 to +999999 (Initial value)

or

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

DATA CONTENTS, means that parameter format is incorrect or a non-digit character is found in the data field or the parameter is outside the specification.

Description:

The **ESC<DASET** read command is used to read the minimum, maximum and initial value set in ESC<DASET for a specific channel. Possible channels are 0,1,2 and 4.

Related commands: **'Esc'<DASET, 'Esc'<DA**

Affected commands: **NONE**

Intentionally blank

Esc<ID - Identification write from ver 1.4

Command: 'Esc'<ID val'cr'

val: ASCII character (up to 64 characters)
\r is reserved word and will generate a carriage return with line feed ('cr' 'lf')

Example: 'Esc'ID SYSTEM 9100 typ 13 200A/50V
Syntax: 'Esc'ID SYSTEM 9100\r typ 13 200A/50V 'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means wrong syntax.

ILLEGAL COMMAND, means that line-in-command is wrong.

DATA CONTENTS, means that parameter format is incorrect or a non-digit character is found in the data field or the parameter is outside the specification.

Description:

Description:
The **'esc'<ID** command is used to enter a user defined text string of maximum 64 characters in length that will be stored in the EEPROM.

Ps. - All characters are converted to upper case regardless if they are typed in lower case
- If the input string is larger than 64 characters, the whole string will be ignored and an error message "data length" will be generated.
- Issuing the 'esc'ID command without parameter will return the ID field in the same way as the ID command alone.

Related commands: **ID**
Affected commands: **NONE**

Esc<ID - Identification read from ver 1.4

Command: 'Esc'<ID 'cr'

Example: 'Esc'ID
Syntax: 'Esc'ID 'cr'

Answer: SYSTEM 9100
TYP 13 200A/50V

or Error message

Errors: **ILLEGAL COMMAND** means that line-in-command is wrong.

Description:

The **ESC<ID** command is used to read a user set able text string of maximum 64 characters. The information stored could be information about the power supply. The default txt is:
(Ps. All characters are converted to upper case)

Related commands: **ID**

Affected commands: **NONE**

Esc<LINE - Serial LINE working mode setup write

Command: 'Esc'<LINE' 'sp'ch',b1,b2,b3,b4,b5,b6,b7,b8

ch: ASCII digit 0 or 1 1=local; 0=remote
bx:: ASCII 0 or 1

Example: 'Esc'<LINE 0,0,0,0,0,1,0,0
Syntax: 'Esc'<LINE'sp'ch',0,0,0,0,0,1,0,0'cr'

Current setting are kept for non entered bits.

Answer: No answer except errors.

or OK if autoanswer mode is set

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means that the parameter format is incorrect or a non-digit character is found in the data field or a parameter is outside the specification.

Description:

The 'Esc'<LINE command configures the working protocol of the serial lines.

The setting becomes first operational after a processor reset or a mode switch update.

The bit definitions are illustrated at the right of this page and on the dip switch setting chapter.

Giving the 'Esc'<LINE sp'ch', command without any parameter will return the current setting.

Esc<LINE continued

b1:	RS485 communication:	0: Disabled			1: Enabled			
b2:	Line turn around	0: no	**1**:	1ms	0:	2ms	1:	3ms
b3:	Line turn around	0: delay	**0**:	delay	1:	delay	1:	delay
b4:	'OK' Answer Mode:	0: Disabled			1: Enabled			
b5:	BOOT character: 0:	"FF"			1: "R" (remote) "L"(local)			
b6:	ACK/NACK Protocol:	0: Disabled			1: Enabled			
b7:	XON/XOFF Protocol:	0: Disabled			1: Enabled			
b8:	ACK/NACK error code	0: Not included			1: Included			

Those in bold are the default setting.

Esc<LINE - Serial LINE working mode setup read.

Command: 'Esc'<LINE' 'sp'ch',

Example: 'Esc'<LINE' 'sp'ch',
Syntax: 'Esc'<LINE' 'sp'ch"cr'

Answer: LINE' 'sp'ch',b1,b2,b3,b4,b5,b6,b7,b8

or Error message

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

Description:

The 'Esc'<LINE read command verifies the programmed wake up bit setup.

Esc<POLDELAY - Polarity DELAY setup write

Command: 'Esc'<POLDELAY'sp'val'

val: ASCII digit 0 to 255 in 100msec. Steps

Example: 'Esc'<POLDELAY 50
Syntax: 'Esc'<POLDELAY'sp'50'cr'

Answer: No answer except errors.

or OK if autoanswer mode is set

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means that the parameter format is incorrect or a non-digit character is found in the data field or a parameter is outside the specification.

Description:

The 'Esc'<POLDELAY command sets a time delay between the OFF state and the activation of the polarity change over switch. This is for letting the rest energy in the magnet to decay.
The time delay is only inserted, if the power supply was ON before invoking the POL +/- command. That is, when just changing the polarity in the power OFF mode, no time delay will be inserted..

Giving the 'Esc'<POLDELAY without any value will return the current setting.

Esc<POLDELAY - Polarity DELAY setup read

Command: 'Esc'<POLDELAY'

Example: 'Esc'<POLDELAY
Syntax: 'Esc'<POLDELAY'cr'

Answer: pulse time value 0 to 255 in 100msec. Steps

or Error message

Errors: **ILLEGAL COMMAND,** means that line-in-command is wrong.

Description:

The 'Esc'<POLDELAY read command verifies the programmed polarity pulse delay time.

APP1. SW 2 Ramp Profile Commands

Following are the commands for the software driven "RAMP PROFILE" listed in alphabetic order.
Please see the SW appendix for parameter format and further detail description.
These commands are optionally available.

SW RAMP PROFILE COMMANDS "Arbitrary point method". Summary

CONT	Continue sequence operation	RSP	Read sequence position
CSS	Clear sequence stack and pointers	RWSP XX	Reset write pointer
FAST	Fast sequence timing	S2	Read sequence status
HALT	Halt sequence operation	SLOW	Slow sequence timing
MULT	Reads the multiplying factor for DAC scaling	SPEED	Read sequence timing
MULT	Writes a multiplying factor for DAC scaling	STOP	Stop sequence executing
R	Write data to the stack	SYNC	Synchronization of sequence
RR	Read ramp status	TS	Trig sequence
RRSP	Reset read sequence pointer	WSA	Write sequence and auto increment
RSA	Read sequence and auto increment	WSP	Write Sequence position

SW RAMP PROFILE COMMANDS "Equal Time slot method". Summary

R	Write data to the stack	RAMPSET	Configure the ramp operation
RAMP	Control the stack operation	RR	Read ramp status

SW RAMP PROFILE COMMANDS "Auto slew rate method". Summary

Esc<SLOPETIME	Set slew rate time for auto slew rate ramp profile execution	RR	Read ramp status
		STOP	Stop the running auto slew rate execution

X is a number from 0 to 9 and Commands in quotation marks are optional.

CONT - CONTinue sequence

Command: CONT'cr'

Example: CONT
Syntax: CONT'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means that there are no running sequence HALT'ed or no sequence has been triggered.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **CONT** command releases the momentary stop, and continues the executing the stack sequence, from the frozen point.

Related commands: **STOP, HALT**

CSS - Clear Sequence Stack

Command: CSS 'sp 'stack 'cr'

stack: ASCII digit 0 to 15

Example: CSS 2
Syntax: CSS'sp'2'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **STACK FRAME ERROR,**
means missing space and stack number after command or stack number outside specified

STACK IS RUNNING, means attempt to clear a running stack.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax

DATA CONTENTS, means parameters outside specified or by use of a non-digit character as parameter, which, in case, can produce a double-error, as it will translate a non-digit character as a zero.

Description:

The command **CSS** resets the whole stack and sets the write-sequence pointer to the first position.

No other command, except **WSA**, affects the auto-increment write pointer.

Related commands: **WSA**

FAST - FAST sequence timing

Command: FAST'sp'stack'cr'

stack: ASCII digit 0 to 15

Example: FAST 0
Syntax: FAST'sp'0'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **STACK FRAME ERROR,** means missing space and stack number, after command or stack number outside specified.

STACK IS RUNNING, means attempt to set timing to a running stack.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means parameters outside specified or by use of a non-digit character as parameter.

Description:

The **FAST** command sets the time unit to 0.1 second and affects all time parameters in a given stack.

This command gives a time range, for each step, from 0.1 sec. to 6553.5 sec.

This command does not affect any parameters, only the speed-setting.

Related commands: **SLOW, SPEED**

Affected commands: **WSA, WSP, RSA, RSP**

HALT - HALT sequence

Command: HALT'cr'

Example: HALT
Syntax: HALT'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means that there are no sequence running. I.e. that

ILLEGAL COMMAND, means that line-in-command is wrong

Description:

The **HALT** command puts the running sequence into a momentary stop. The sequence can at a later state be continued or stopped.

In **HALT** mode all timing and calculations is frozen. The DAC output remains with the present value, when **HALT** was given.

In **HALT** mode the sequence can be terminated by using the **STOP** command without restarting it with **CONT**.

Related commands: **STOP, CONT**

MULT - MULTiply factor (read)

Command: MULT'sp'stack'cr'

stack: ASCII digit 0 to 15

Example: MULT 1
Syntax: MULT'sp'1'cr'

Answer: MULT'sp'stack,factor'lf''cr'

stack: ASCII digit 0 to 3
factor: ASCII digit 000000 to 999999 in PPM

Errors: **STACK FRAME ERROR,** means missing space and stack number, after command or stack number outside specified.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means parameters outside specified or by use of a non-digit character as parameter.

Description:

The **MULT** (read) command returns actual scale-factor, in which all set values in a stack are multiplied (global attenuator).

A **MULT**iply value at 000000 means a disabled function.

Command, **MULT** (read), does not affects its parameter.

Related commands: **MULT** (Write)

Affected commands: **WSA, WSP, RSA, RSP**

MULT- MULTiply factor (write)

Command: MULT'sp'stack,factor'cr'

stack: ASCII digit 0 to 15
factor: ASCII digit 000000 to 999999 in PPM

Please note, that either a leading zero or a trailing zero format can be used by the power supply depending of the initial setup mode. Default factory setting is leading zeroes. (Read leading or trailing as important zeroes)

Example: MULT 1,750000
Syntax: MULT'sp'1,750000'cr'

Answer: No answer, except errors
or OK if autoanswer mode is set

Errors: **STACK FRAME ERROR,** means stack number outside specified.

STACK IS RUNNING, means attempt to set factor in a running stack.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means parameters outside specified or by use of a non-digit character as parameter.

Description:

The **MULT** (write) command is used as a scale factor, in which all set values in a stack are multiplied. The time parameters are not affected.
This command is used as a general "volume control", which allows to scale all DAC parameters in a given stack.
A **MULT**iply value at 000000 disables the function.

No other commands, except **MULT**, affects this parameter.

Related commands: **MULT** (read)

Affected commands: **WSA, WSP, RSA, RSP**

R - Writes data to the stack "Equal time slot mode"

Command: R'sp'[value],[S]'cr'

Value: Floating point number between 0.0000000 to 1.0000000
S: Ends the writing.

Example: R 0.123456 equals 123456ppm output current.
Syntax: R'sp'0.123456'cr'
Or
R S
Syntax: R'sp'S'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means a missing space between the command and parameters or wrong syntax.

DATA CONTENTS, means that parameter format incorrect or a non-digit character found in data field or parameters outside specification.

Description:

The **R** (write) command writes a data points to the "Equal time slot" stack. All data must entered sequently until "**R S**" states the end of data. When first started entering data no other commands are allowed in-between. If issued any way, they will be discarded. Up to 1000 values may be entered, but only 3 is minimum.

A new bunch of data may be down loaded during a stack execution as next run preparation.

Before using **R** command, use the **RAMPSET C** command to empty the stack.

Related commands: **RAMPSET, RAMP**

R - Reads data from the stack "Equal time slot mode"

Command: R'cr'

Example: R
Syntax: R'cr'

Answer: All stack data will be dumped.

Example: R 0.0
R 0.123456
R 0.879
R 0.3
R 0.0
R S

Errors: **SYNTAX ERROR,** means wrong syntax.

Description:

The **R** (read)command is used to dump all data from the "Equal time slot" stack.
When first started this dump command no other commands are allowed before the whole stack has been read. If issued any way, they will be discarded.

Related commands: **RAMPSET, RAMP**

RAMP - Controls the stack execution "Equal time slot mode"

Command: RAMP'sp'[RNSHT],[LWN]'cr'

R:	Run.	Starts the stack execution
S:	Stop.	Stops the stack execution
H:	Halt.	Halts the stack execution
T:	Trig.	Prepares the stack for a HW trigger start

L:	Loop.	Loop the stack. After the last data is put through, the execution will automatically restart from the beginning.
W:	Wait.	After a stack run the process will wait for a HW trigger signal before a new run will be restarted.
N:	Normal.	Resets all control parameters to normal (one shoot). If the stack is running, the execution will stop after the last data point.

Example: RAMP T,W Prepares the stack for HW trig start and runs the stack once. HW triggering is needed for a new run.
Syntax: RAMP'sp'T,W'cr'
Or
RAMP R,L Runs (starts) the stack in a loop operation
Syntax: RAMP'sp'R,L'cr'

RAMP , N Change to normal run-mode while ramp is running.
Syntax: RAMP'sp',N'cr'

Answer: No answer, except errors

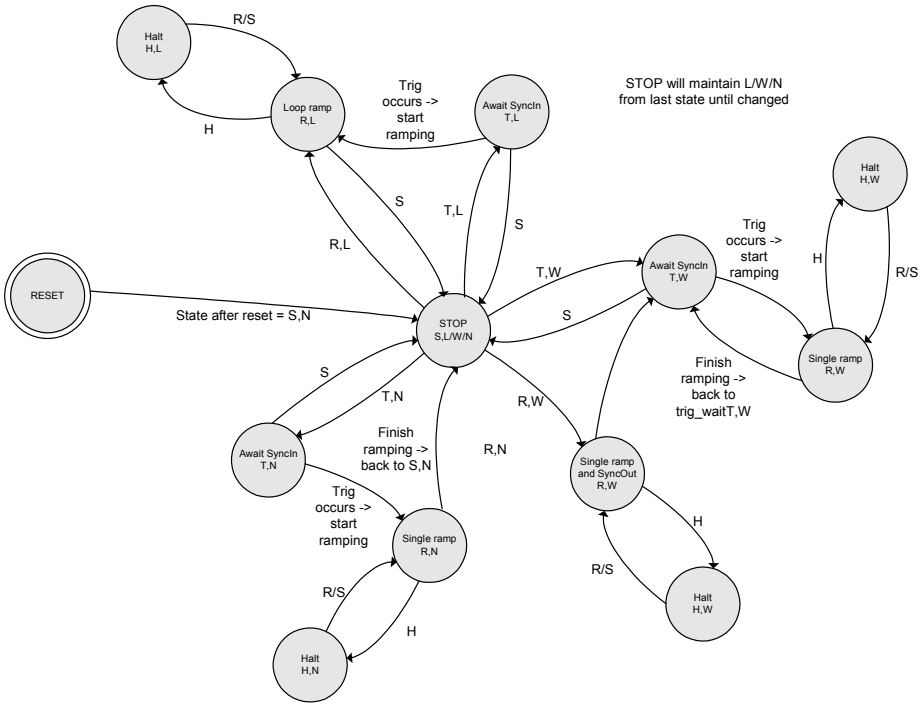
Errors: **SYNTAX ERROR,** means a missing space between the command and parameters or wrong syntax.
DATA CONTENTS, means that parameter format incorrect.

RAMP continued

Description:
The **RAMP** (write) command configures the running condition of the "Equal time slot" stack. It is also possible to change the running condition [LWN] that was prior set with the RAMPSET command. When ramp is running and the run-status needs to be change, use following command to change the run-status: RAMP ,[LWN] remember to omit the first parameter.
If RAMP S is issued, then the DAC output will freeze to its present value.

Related commands: **RAMPSET, R**

State machine:



RAMP - Reads the stack control "Equal time slot mode"

Command: RAMP'cr'

Example: RAMP
Syntax: RAMP'cr'

Answer: Stack control present setting

Example:	RAMP T W	Stack is on HW trigger loop mode
	RAMP R L	Stack is running in loop mode
	RAMP S N	Stack is stopped in normal mode
	RAMP H W	Stack is halted in a trigger wait mode.

Errors: **DATA PARM ERROR,** means that parameter format incorrect or wrong syntax

Description:

The **RAMP** (read) command reads the present status of the "Equal time slot" stack.

Related commands: **RAMPSET, R**

Intentionally blank

RAMPSET - SETing up the stack parameter "Equal time slot mode"

Command: RAMPSET'sp'[Time,[Mult],[TrDly]],[LWN],C'cr'

Time: Time. Sets the time slot value in 0.00125 second steps.
Mult: Mult. Sets the multiplicate factor. (Gain control)
TrDly: Delay Sets a delay time in seconds from the trig time to the stack execution.
Minimum resolution is one µ-step (=1.25mS.)
Loop. Loop the stack. After the last data is put through will the execution automatically restart from the beginning.
W: Wait. After a stack run will the process wait for a HW trigger signal before a new run will be restarted.
N: Normal. Set ramp execution to normal. (No loops)

Example: RAMPSET C Clears the stack and parameters (1.0sec, *1.0, N=true)
Syntax: RAMPSET'sp'C'cr'
Or
RAMPSET 0.01,0.98657,0.001,W

sets time, multiplicand, trigger delay & in wait looping

RAMPSET 0 Sets time to max resolution=0.00125sec. multiplicand=1 & N=true

RAMPSET ,,0.98657 Sets only the multiplicand other parameters left unchanged.

RAMPSET Sets parameters to normal excecution.

Answer: No answer, except errors

Errors: **SYNTAX ERROR,** means a missing space between the command and parameters or wrong syntax.

DATA CONTENTS, means that parameter format incorrect or a non-digit character found in data field or parameters outside specification.

RAMPSET Continued.

Description:

The **RAMPSET** (write) command is used to set up the stack controlling parameters, that is: time slot value, multiplicand (gain factor), a time delay before the stack starts running and running condition. Execution can not be started with this command, use RAMP S to start. Changing the TrDly only needs empty parameters for preceding parameters.
Eg. RAMPSET „0.001 Default trigger delay is 0.

Related commands: **RAMP, R**

RAMPSET - Reading the SETup stack parameter

Command: RAMPSET‘cr'

Example: RAMPSET
Syntax: RAMPSET'cr'

Answer: Stack control present setting
Syntax RAMPSET’sp’Time,Mult,TrDly [LWN],No. of entries

Example: RAMPSET 1.0,1.0,0 N 0 Empty stack
RAMPSET 0.01,0,75,0.1,L,512 10ms. Time slot, gain=0.75, 10mS Trigger delay, Looping mode and 512 data entries.

Errors: **SYNTAX ERROR,** means wrong syntax.

Description:

The **RAMPSET** (read) command is used to read the setting parameters of the stack, the running condition and hoe many entries there are in the present stack.

Related commands: **RAMP, R**

RR - Read ramp/stack status from ver 1.02

Command: RR

Example: RR
Syntax: RR'cr'

Answer: Ramp/stack status R, S or H

Example: S Stack is stopped
R Stack is running
H Stack Halted: waiting for trig, in polarity change state or in a brake point state.

Errors: **SYNTAX ERROR,** means wrong syntax.

Description:
The **RR** is a general purpose command that works with all three stack execution modes telling the present state of the stack. (Arbitrary Ramp Profile method, Equal Time Slot method and Auto Ramp feature)
The **RR** command is specially designed for the Auto Ramp feature to see, when the desired new current set value has been reached.

Related commands: **‘esc’<SLOPETIME**

RRSP- Reset Read Sequence Pointer

Command: RRSP'sp'stack'cr'

stack: ASCII digit 0 to 15

Example: RRSP 2
Syntax: RRSP'sp'2'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **STACK FRAME ERROR,**
means missing space and stack number after command or stack number outside specified.

SYNTAX ERROR,
means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS,
means parameters outside specification or a non-digit character, which also can produce a double-error, as it will translate a non-digit character to a zero.

Description:

The **RRSP** command resets the read-sequence pointer used by the **RSA** command to its first position.

No other command, except **RSA**, affects the "auto increment read pointer".

Related commands: **RSA**

Intentionally blank

RSA - Read Sequence and Auto increment

Command: RSA'sp'stack'cr'

stack: ASCII digit 0 to 15

Example: RSA 0
Syntax: RSA'sp'0'cr'

Answer: If actual position contains data then:
SP'sp'stack,posit,start,stop,time'lf''cr'

stack:	ASCII digit 0 to 15	
posit:	ASCII digits 00 to 15	
start:	ASCII digits 000000 to 999999	in PPM
stop:	ASCII digits 000000 to 999999	in PPM
time:	ASCII digits 00001 to 65535	in time units *)

or If actual position does not contain data (time=zero) then:
SP'sp'stack,posit,EMPTY'lf''cr'

or If actual position attempts to pass last position then:
'bell'?'sp'STACK NO LONGER'lf''cr'

Errors: **STACK FRAME ERROR,** stack pointed to, outside specified.

SYNTAX ERROR, means a missing space between the command and parameters or wrong syntax.

DATA CONTENTS, means that parameter format incorrect or a non-digit character found in data field or parameters outside specification.

RSA continued

Description:

The **RSA** command is used to read a sequence from sequence-stack without using an absolute position. Each **RSA** command increments the position-pointer to the next position, until last position.

Before using **RSA** command, use the **RSP** command to start reading from a given position.

No other command, except **RRSP**, affects the “auto increment read pointer”.

Related commands: **RRSP, RSP, WSA, WSP**

RSP - Read Sequence Position

Command: RSP'sp'stack,posit'cr'

stack: ASCII digit 0 to 3
posit: ASCII digit 00 to 15

Example: RSP 0
Syntax: RSP'sp'0'cr'

Answer: If pointed position contains data then:
SP'sp'stack,posit,start,stop,time'lf''cr'

stack:	ASCII digit 0 to 3	
posit:	ASCII digits 00 to 15	
start:	ASCII digits 000000 to 999999	in PPM
stop:	ASCII digits 000000 to 999999	in PPM
time:	ASCII digits 00001 to 65535	in time units *)

or If pointed position does not contain data (time=zero) then:
SP'sp'stack,posit,EMTY'lf''cr'

or If pointed position attempts to pass last position then:
'bell'?'sp'STACK NO LONGER'lf''cr'

Errors: **STACK FRAME ERROR,** stack pointed to, outside specified.

SYNTAX ERROR, means a missing space between the command and parameters or wrong syntax.

DATA CONTENTS, means that parameter format incorrect, or a non-digit character found in data field, or parameters outside specification

RSP continued

Description:

The **RSP** command reads a set of parameters from a given position in a given sequence-stack.

No pointers are affected.

Related commands: **RSA, WSA, WSP**

RWSP - Reset Write-Sequence Pointer

Command: RWSP'sp'stack'cr'

stack: ASCII digit 0 to 15

Example: RWSP 2
Syntax: RWSP'sp'2'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **STACK FRAME ERROR,** means missing space and stack number after command or stack number outside specified.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means the parameter is outside the specification or a non-digit character, which also can produce a double-error, as it will translate a non-digit character to a zero.

Description:

The **RWSP** command resets the Write-sequence pointer used by the **WSA** command to its first position.

No other command, except **WSA**, affects the "auto increment read pointer".

Related commands: **WSA, RRSP**

S2 - Status 2

Command: S2'cr'

Example: S2
Syntax: S2'cr'

Answer: If sequence is running then:
Rstack,posit'lf''cr'

or If sequence is halt'ed then:
Hstack,posit'lf''cr'

or If sequence is stopped or has not been started after reset, then:
Sstack,posit'lf''cr'

or If sequence prepared to run (armed) through a SYNC command:
Pstack,posit'lf''cr'

stack: ASCII digit 0 to 15 or X *)
posit: ASCII digits 00 to 15

*)After a reset or a power-up (before any sequence has been triggered) the stack is not yet been specified and will then return as X. The same answer is possible after a **STOP** command is executed.

Errors: **SYNTAX ERROR,** means wrong syntax.

Description:

The **S2** command reads back information about the internal status of the ramp profile sequence state machine. It return's information which stack is running, if it is halted or the present position. The position will always be the actual (last) executed position.
If the sequence has been stopped or finished its execution, the information about the stack, is lost and the returned answer will be expressed as an X.
The command **S2** can be executed at any time.

Nothing else is affected.

SLOW - SLOW sequence timing

Command: SLOW'sp'stack'cr'

stack: ASCII digit 0 to 15

Example: SLOW 0
Syntax: SLOW'sp'0'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **STACK FRAME ERROR,** means missing space or stack number, or stack number outside specification.

STACK IS RUNNING, means attempt to set timing to a running stack.

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means parameters outside specification or use of a non-digit character as parameter.

Description:

The command **SLOW** sets the time unit *) to 1 second and affects all time parameters in a given stack.

This command gives a time range, for each step, from 1 sec. to 65535 sec.

This command does not affect any parameters, only the speed-setting.

Related commands: **FAST, SPEED**

Affected commands: **WSA, WSP, RSA, RSP**

SPEED- SPEED (read sequence timing)

Command: SPEED'sp'stack'cr'

stack: ASCII digit 0 to 15

Example: SPEED 0
Syntax: SPEED'sp'0'cr

Answer: If stack is running in FAST mode then:
SPEED'sp'0,FAST'lf'cr'

or If stack is running in SLOW mode then:
SPEED'sp'0,SLOW'lf'cr

Errors: **STACK FRAME ERROR,** means stack number outside specified

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax.

DATA CONTENTS, means that the parameter is outside specification or a non-digit character used as parameter

Description:

The **SPEED** command returns the actual speed-mode set to a given stack.

This command does not affect any parameters.

Related commands: **FAST, SLOW**

STOP - STOP sequence

Command: **STOP'cr'**

Example: STOP
Syntax: STOP'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **SYNTAX ERROR,** means that there are no running sequence to stop.

ILLEGAL COMMAND, means that line-in-command is wrong.

Description:

The **STOP** command works with the "Arbitrary point method" and the Auto Ramp feature (from ver. 1.02)

The **STOP** command terminates the sequence execution. !! The sequence can not be continued.!! If desired, please use the **HALT** command instead.

After the **STOP** command is given, the DAC output will freeze to its present value.

Related commands: **HALT, CONT**

SYNC - SYNChronisation

Command: SYNC'sp'stack,dly'cr'

stack: ASCII digit 0 to 15
dly: ASCII digit 0 to 10000000 in µ seconds

Example: SYNC 2,1000
Syntax: SYNC'sp'2,1000'cr

Answer: No answer, except errors

or OK if autoanswer mode is set

Errors: **STACK FRAME ERROR,** means stack number outside specified

SYNTAX ERROR, means a missing space between the command and parameter or wrong syntax

DATA CONTENTS, means parameters are outside specification or due to a non-digit character as parameter

Description:

The **SYNC** command arms a stack so it can be synchronized to an external event or signal. If the **SYNC** command is followed by a **TS** command a sync signal is given to other supplies. Plug 33 is used for the HW input and output sync signals. A start delay 'dly' can be issued to adjust the skew between the connected power supplies or events.

This command does not affect any parameters.

Related commands: **TS,**

TS - Trig Sequence

Command: **TS'sp'stack'cr'**

stack: ASCII digit 0 to 15

Example: TS 0
Syntax: TS'sp'0'cr'

Answer: No answer, except errors

Errors:

STACK NO LONGER,	means that stack is empty, eg, as after a CSS command or power-up/hard-reset,
STACK IS RUNNING,	means attempt to Trig a running stack.
SYNTAX ERROR,	means a missing space between the command and parameter or wrong syntax.
DATA CONTENTS,	means parameters outside specification or a non-digit character. The later case can produce a double error, as it will translate a non-digit character as a zero.

Description:

The **TS** command starts executing the given sequence from the first position.

Related commands: **HALT, STOP, CONT**

Intentionally blank

WSA - Write Sequence and Auto-increment

Command: WSA' sp' stack, start, stop, time 'cr'

stack:	ASCII digit 0 to 15	
start:	ASCII digits 000000 to 999999	in PPM
stop:	ASCII digits 000000 to 999999	in PPM
time:	ASCII digits 00001 to 65535	in time units *)

Leading zero's can be omitted in all parameters
*) See FAST and SLOW commands. If time is set to zero, position will be marked EMPTY!

Example WSA 0,000,450050,00225
Syntax: WSA'sp'0,000000,450050,00225'cr'
(most readable syntax but slow)

or WSA 0,0,450050,225
Syntax: WSA'sp'0,0,450050,225'cr
(recommendable syntax medium speed)

or WSA ,,450050,225
Syntax: WSA'sp',,45050,225'cr'
(fast syntax)

Answer: No answer, except errors

STACK NO LONGER, means attempt to write to a position beyond stack.

WSA continued

Errors: **STACK FRAME ERROR,** stack pointed to, outside specification

STACK IS RUNNING, means attempt to write to a running stack.

SYNTAX ERROR, means a missing space between the command and the parameter or wrong syntax.

DATA CONTENTS, means that the parameter format is incorrect or a non-digit character is found in the data field or a parameter is outside the specification.

STACK NO LONGER, means attempt to write to a position beyond stack

Description:

The **WSA** command writes a sequence-stack without using absolute position. Each **WSA** command automatically increment the position pointer to the next position.

Before using **WSA** command, use the command **CSS** to empty the stack and reset pointer.

No other command, except **CSS**, affects the “auto increment write pointer”.

Related commands: **CSS, WSP, RSA, RSP**

WSP - Write Sequence Position

Command: WSP'sp'stack,posit,start,stop,time'cr'

stack: ASCII digit 0 to 15
posit: ASCII digits 00 to 15
start: ASCII digits 000000 to 999999 in PPM
stop: ASCII digits 000000 to 999999 in PPM
time: ASCII digits 00001 to 65535 in time units *)

Leading zero's can be omitted in all parameters
*) See FAST and SLOW commands. If time is set to zero, position will be marked EMPTY!

Example: WSP 0,09,000000,450050,00225 (most readable syntax but slow)
Syntax: WSP'sp'0,09,000000,450050,00225'cr'

or WSP 0,9,0,450050,225 (recommendable syntax medium speed)
Syntax: WSP'sp'0,0,450050,225'cr

or WSP ,9,,450050,225 (very fast syntax but not recommended)
Syntax: WSP'sp',,45050,225'cr'
(very fast syntax but not recommended)

Answer: No answer, except errors

Or OK if autoanswer mode is set

WSP continued

Errors:

STACK FRAME ERROR, stack pointed to, outside specification.

STACK IS RUNNING, means attempt to write to a running stack.

SYNTAX ERROR, means a missing space between the command and the parameter or wrong syntax.

DATA CONTENTS, means that the parameter format is incorrect or a non-digit character is found in the data field or a parameter is outside the specification.

STACK NO LONGER, means attempt to write to a position beyond stack.

Description:

The **WSP** command loads a set of parameters into a given position of a given sequence-stack. No pointers are affected.

Related commands: **WSA, RSA, RSP**

Esc<SLOPETIME - SW generated auto slew rate setup write

from ver 1.02

Command: 'esc'<SLOPETIME'sp'val1','val2', 'val3' 'cr'
val1: ASCII digits in floating point representation 0.005 -1000 & 0
Positive slew rate time from 0 to 100%
val2: ASCII digits in floating point representation 0.005 -1000 & 0
Negative slew rate time from 0 to 100%
Val3: ASCII digits in floating point representation 0.005 -1000 & 0
Minimum time a slew rate run may take. (from ver 1.4)

Example: 'esc'<SLOPETIME 1.125,0.750,0.2
Syntax: 'Esc'<SLOPETIME'sp'1.125,0.750,0.2'cr'

Answer: No answer except errors.
or OK if autoanswer mode is set

Errors:

ILLEGAL COMMAND	means that line-in-command is wrong.
SYNTAX ERROR,	means a missing space between the command and parameter or wrong syntax.
DATA CONTENTS,	means that the parameter format is incorrect or a nondigit character is found in the data field or a parameter is outside the specification.

Description:
The 'Esc'<SLOPETIME command is used to set the desired slew rate times for the positive and the negative slew rates with the parameter 'value 1 & 2' respectively. The slope time is to be seen as the time it takes the power supply to go from 0 to 100%. If both the values are equal to zero the auto slew rate feature will be disabled. If only one of the values are equal to zero (or not given) the "zero" value will be set equal to the other non zero value.
The auto slew rate feature will automatically generate an internal SW ramp profile after receiving a "DA 0,xxxxxx" or "WA xxxxxx" command.
Val3 is the minimum time a slew rate run may take. Val3 must be less than or equal to the smallest value of val1 or val2.

The setting becomes operational immediately after then saving into the EEPROM.

Esc<SLOPETIME continued

Definition of slew rate sign:
For unipolar supplies:
Positive slew rates are defined when ramping from a small absolute value to a higher absolute value. (| Istart | < | Istop |)
Negative slew rates are defined when ramping from one absolute value to a lower absolute value. (| Istart | > | Istop |)
For power supplies equipped with a "polarity change over switch", the supply will first ramp down to zero (negative slew rate), make the polarity change, thereafter run up to the desired output current with a positive slew rate.
For bipolar supplies:
Positive slew rates are defined when ramping from a lower value to a higher value. (Istart < Istop)
Negative slew rates are defined when ramping from a higher value to a lower value. (Istart > Istop)
Ramp will continue across zero.

Definition of slew rate time:
The slew rate time is defined as the time it will take the power supply to go from zero to 100%.
That is:
Ramp amplitude: | Istart - Istop |
Ramp time: (| Istart% - Istop% |)*('Slew Rate value'/100)
'Slew Rate' = 'val1' or val2'

Ramp Profile shape:
The ramp profile will follow a ½ sinus or a square shape depending on b4 in AUX2 setup.
(Other shapes can be achieved on request)
The ramp profile are split into 80 linear lines with HW 1.25ms interpolation in between.

Related commands: **RR**

Esc<SLOPETIME - (read)

from ver 1.02

Command: 'esc'<SLOPETIME'

Example: 'esc'<SLOPETIME
Syntax: 'Esc'<SLOPETIME'cr'

Answer: Slope time val1 (for positive ramp), val2 (for negative ramp)

val1: Floating point ASCII digits 0.005 to 1000
val2: Floating point ASCII digits 0.005 to 1000
val3: Floating point ASCII digits 0.005 to 1000 (from ver 1.4)

or Error message

Errors: **ILLEGAL COMMAND** means that line-in-command is wrong.

Description:

The slope time is to be seen as the time it will take the power supply to go from 0 to 100%. Val3 is the minimum time a slew rate run may take.

Intentionally blank