

Software standard commands for SYSTEM 7000



	Written By
Name	Dat Tram
Date	09.06.2004
Signature	

REV.	By	Date	Description	Pages
1.0	DT	09.06.2004	Initial	All
1.0	DT	14.07.2004	Adding ESC<NFSMODE, STB, TYPE, ADCTRIG	36, 25, 10
1.0	DT	23-08-2004	Revise and add offset adjustment in ESC<DASET and ESC<ADSET	34, 30
1.0	DT	30-08-2004	Parameter L added in ESC<DASET. TD added	34, 26
1.1	DT	14-01-2005	Line-Turn-Around time added. From rev. 1.4 AD unit setting corrected	35 9
1.2	DT	07-09-2005	External interlocks reflect S1	23



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.

This issue is valid for software version BCP100

STANDARD COMMANDS. summary

AD X	Read value from an ADC channel.	STB	Standby (Main Power ON with clamped 'out4')
ADR	Read the address of the MPS.	TD	Test DAC
ADR XXX	Write an address to a MPS.	TYPE	AD type in use
ADCTRIG	Read the AD channel trig setup	UNLOCK	Unlock the MPS
CMD	Read current control mode.	VER	Reads the software version
CMDSTATE	Read current control state.	WA XXXXXX	Writes a value to an Digital to Analog converter. (Alternative DA command.)
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		
LALL	Listen ALL.		
LOC	Change to Local Control		
LOCK	Lock the MPS in Local Control.		
N	Main Power ON		
NERR	No error message		
PO	Polarity status		
PO +/-	Change to Normal polarity		
PRINT	Reads internal user information about the MPS.		
RA	Read the set value.		
REM	Change to remote control.		
RLOCK	Remote line only		
RS	Reset interlocks.		
S1	Read the internal status.		
S1H	Read internal status in HEX format		

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.

This issue is valid for software version BCP100

Esc SET UP COMMANDS. summary

ESC<AD	Configures the AD converter scaling and routing (Output reading adjustment or output reading in Amps)	ESC<DASET	Auto Configures the scaling (gain) and Offset for a DA converter channel.
ESC<ADR	Configures the communication address setting (in RS422 mode).	ESC<LINE	Configures the protocol for the serial lines.
ESC<ADSET	Auto Configures the scaling "gain" and Offset for an AD converter channel.	ESC<NFSMODE	Configures the ON/OFF/STANDBY mode operation
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		

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.)

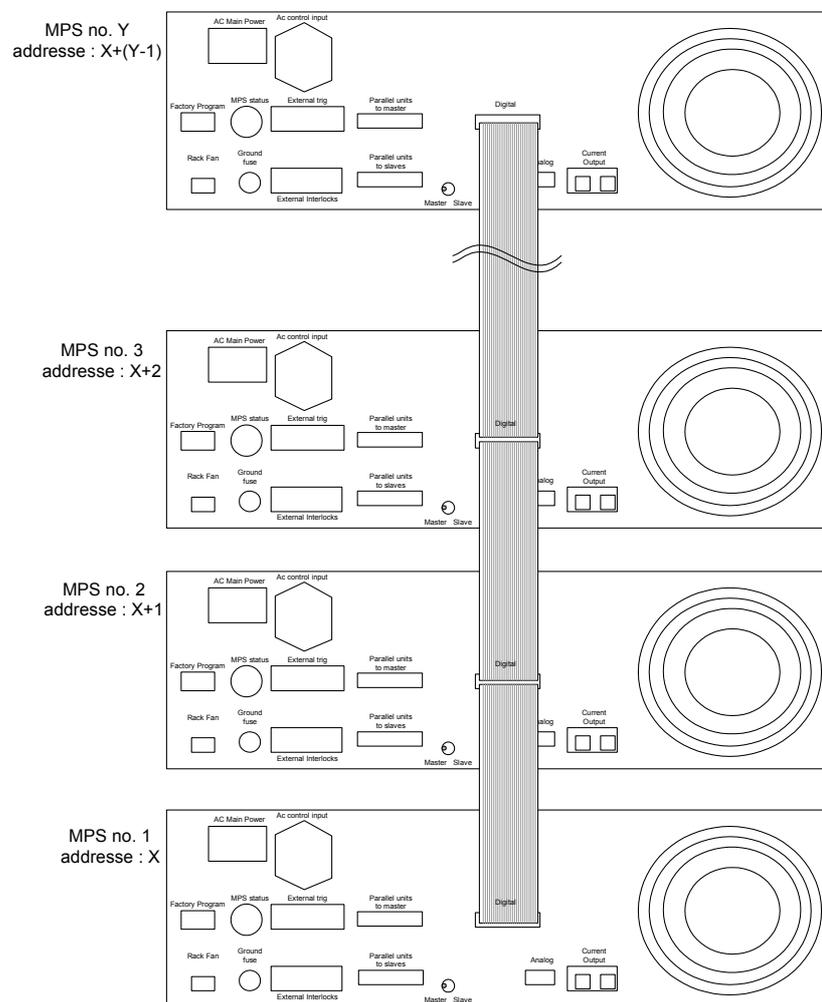


Multidrop configuration

Up to 32 SYSTEM 7000 units can be connected in a RS422 or RS485 multidrop configuration.

It is also possible to run a Master/Slave configuration within a multidrop system. All units must be connected in order to read the individual unit interlock status signals, but for controlling the system only the master connection is necessary. Trying to control one of the slaves will have no effect as the control commands are disabled on the slaves. To set up the Master/Slave configuration, see chapter 2.6.

- The Master / Slave slide switch located on the back of the supply must be placed in Master position for all masters.
- Address all units with a unique address. *See under ADR command on how to set up the unit address.*



X :Start address
Y : The last unit of MPS on the multidrop line

Example of multidrop configuration



When using several SYSTEM 7000 in RS422 multidrop configuration, each individual SYSTEM 7000 needs to have a unique address first.

When using a SYSTEM 7000 for the first time, the address of the specific MPS is undefined. This means that the system is not ready to run in multidrop mode as running in RS422 multidrop mode presumes that all the nodes have a unique address.

In order to address each individual SYSTEM 7000 for the first time, follow these steps below.

- Connect the serial cable to the “Digital” connector on the back of the SYSTEM 7000.
- Use “ESC<ADR” command for addressing the SYSTEM 7000 *. (See testscript for Multidrop below)
- Make a “ESC<CPURESET” in order for the address to take effect.
- Remove the serial cable from the “Digital” connector.
- Repeat steps 1 and 2 on another SYSTEM 7000 until all the SYSTEM 7000 have been addressed with a unique address.

When done addressing all the SYSTEM 7000, these can connect in RS422 multidrop mode.

Since all the addresses are known, calling a specific address is possible by using the command “ADR”. Changing address on a specific SYSTEM 7000 is also now possible in multidrop mode.

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



Software Profile Programming

SYSTEM 7000 is delivered with the software profile option.

With the ramp profile SW it is possible to down load and run a predefined ramp sequence that the output current must follow. The SYSTEM 7000 supports Equal timeslot method.

The examples below are shown for a uni-polar profile. For bipolar profiles, the output current may also be set as negative.

Equal time slot ramp profile method

With the "Equal time slot method" it is possible to download up to 512 current set values and a single time slot value, that will be used for all set values. Only one stack is available.

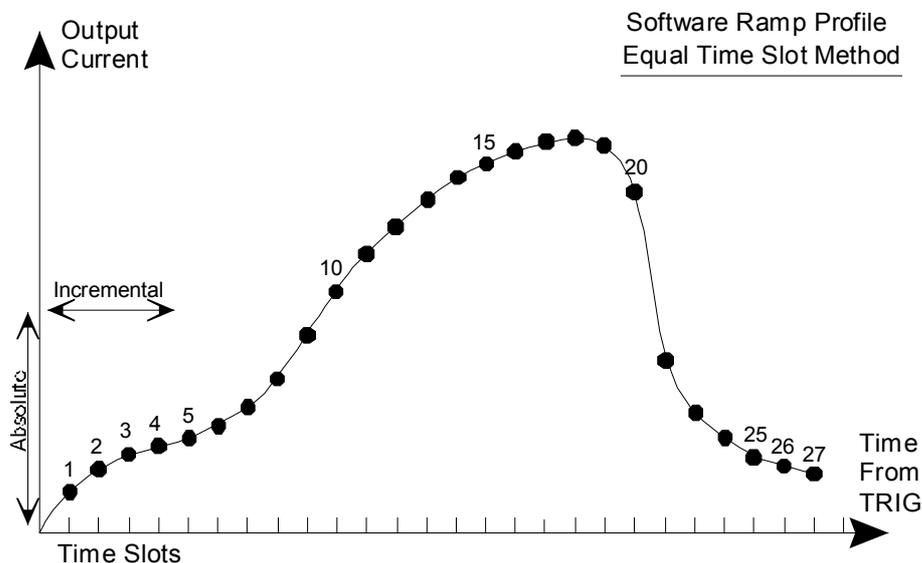
This profile method is especially useable for faster and more accurate curves fitting profiles e.g. as a function generator.

To use the Equal time slot method at least the following steps must be preformed:

- Clear and set the stack "RAMPSET [parameter]"
- Program the stack "R [parameter]"
- Start the stack "RAMP [parameter]"
- Read the status of the running stack "RAMP"

PS. All values must be given as a floating point number scaled to "1.00000". That is; 2.5ms must be entered as 0.00250 and 19.54% output current as 0.1954.

The figure below shows an example of one ramp profile stack. (Ps. not all 512 points need to be programmed, empty entries will be ignored.)



The time slot must be given as a multiple of 2.5ms. Between 2.5ms to 1 second. Any value in-between will automatically be rounded according to formulae:

$$\{\text{time slot}\} = \text{frac}(\{\text{time}\} / 0.00250) * 0.00250$$

The SW will after the start command update the output current every 2.5ms. By means of interpolation regardless of the programmed time slot value:



The ramp can be initiated to run as a single shot “RAMP R”, auto iteratively (auto loop) by software command “RAMP R,L” or HW triggered auto armed “RAMP T,W”. For a full documentation on controlling the “Equal time slot method” please refers to *RAMP Software commands*.

If synchronization to an external event is required, it is possible to arm the ramp sequence first with the synchronization command “RAMP T”. A hardware signal on the trigger input X7 pin 1&2 (10 to 24V) or a “RAMP R” command will start the sequence.

If more power supplies have to be synchronized, one of the supplies has to be appointed as master. Connecting the master trig output X7 pin 3&4 to the other supplies trig input will start the other supplies when the master is triggered. A maximum skew of 2.5ms between the supplies may be expected. (an external 15V auxiliary supply is needed, as the trig output is an open collector and the trig input is an opto coupler input.)

External input and output triggers are located on the back of the SYSTEM 7000

Hint: When adding values to the ramp profile, the enter point (start) and exit point (stop) shall be the same in order for running the ramp sequence in loop. Otherwise there will be an unwanted level between the exit point for the first ramp and the enter point for the next ramp as the ramp is miscalculated. Besides, the output current should set to zero before start running the ramp.

In order to run ramp profile correctly, following steps must be performed

- Clear the stack by typing RAMPSET C
- Enter the data points, type R 0.xxxxx and press enter. Repeat this until the last data point has been entered.
- Save the data points by typing R S
- Set the output current to zero by typing WA 0 or DA 0,0
- Turn on the MPS by typing N
- Run the ramp in loop by typing RAMP R,L

x is a number from 0 to 9

Description of how to set up and run the ramp profile, refer to RAMP commands in page 37.

SW limits

The limits of the “Equal time slot method” ramp profile SW are:

- The set value must be given in a floating point representation normalized to 1.00000.
- The time slot may be between 0.0250 to 1 second given in a floating point representation normalized to 1.0000.
- Maximum numbers of stacks = 1
- Maximum number of time slots in a stack = 512

AD - AD X

Command: AD'sp'ch'cr'

ch: ASCII digit 0 to 16

Example: AD 0

Syntax: AD'sp'0'cr'

Answer: 'val'lf'cr'

ch: ASCII digit 0 to 16

val: ch 0, 2, 8 and 12, and 16 ASCII digit 00000 to 99999

val: ch 3,4,5 ASCII 000 to 999

(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 REQUEST 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 reads the different Analog to Digital converters. The AD channels and their response are described on the next column.

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 (12 bit)	I* 1000	"SDDDDDD"
1	Not used.		
2	Vout	V*100	"SDDDDDD"
3	Internal +15V sup.	V*10	"DDD"
4	Internal -15V sup.	Num.(V*10)	"DDD"
5	Internal +5V sup.	V*10	"DDD"
6	Not used.		
7	Not used.		
8	Output current (16 bit)	I*1000	"SDDDDDD"
9	Not used.		
10	Not used.		"DDD"
11	Not used.		"DDD"
12	Vout	V*100	"SDDDDDD"
13	Not used.		"DDD"
14	Not used.		"DDD"
15	Not used.		"DDD"
16	I set value.	A*100	"SDDDDDD"

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

ADCTRIG - AD Channel trig read

Command: ADCTRIG 'cr'

Example: ADCTRIG
Syntax: ADCTRIG 'cr'

Answer: 0 or 1 (0=continue mode 1=External trig)

or No answer, except errors

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

ILLEGAL REQUEST means that line-in-command is wrong.

Description:

The **ADCTRIG** reads the analog-to-digital trigger mode. The mode is hardware selectable.

Note: External trig for ADC and external trig for ramp use the same port.
If external trig for ADC is selected, external trig for ramp can **not** be used.
If external trig for ramp is selected, external trig for ADC can **not** be used.

Nothing else is affected.

Intentionally blank

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 REQUEST means that line-in-command is wrong.
DATA ERROR 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: **ADR** (read)

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: **ADR** (write)

Affected commands: **LALL**

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 REQUEST 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 REQUEST** 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:
Syntax: REMOTE'lf'cr'

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

or If line-in-command is locked to remote :
Syntax: RLOCK'lf'cr'

or If line-in-command is locked to local :
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 REQUEST 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**

DA - Write to DAC's

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

ch: 0 val: digits ± 000000 to ± 999999 * in $10e-4$ Ampere.

* The max. value enter is depended on the value set in DAC.

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

ch 4, no change. See previous page.

Example: DA 0,-0480

Syntax: DA'sp'0','-0480'cr'

Means a set value of -480 in $10e-4$ Ampere

Answer: digits 000000 to ± 999999

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

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

ILLEGAL REQUEST 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

DA continued

Description:

When setting the DAC without prefixed sign will be accepted as positive. That is; if issuing "WA 20000" or "WA +20000" will become +20000 and issuing "WA -20000" will become -20000.

One 7-segment indicates the polarity status on the front-Panel display.

'-' minus will be indicated with minus sign

'+' positive, no indication (blank)

Use SET, UP/DOWN and LEFT/RIGHT buttons to change the polarity status in local control on the front-panel or 'PO +' or 'PO -' command 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 the PO status may be affected.

Related commands: **RA, WA**

ERRC - ERRor in Code (remote line only)

Command: ERRC'er'

Example: ERRC
Syntax: ERRC'er'

Answer: No answer, except errors

or OK if autoanswer mode is set.

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

ILLEGAL REQUEST 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 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

<u>CODE NO.</u>	<u>ERROR TEXT</u>
1	Command error
2	Data error
3	Data error
4	Illegal request
5	Ramp running
6	Status quo, no change
7	Change in progress – Command Delete
8	Stack is running
9	Stack is close
10	Data error
11	Stack is halted
12	PSU error
13	Not ready error
14	Syntax error
15	Stack is empty
16	MPS not on

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 REQUEST 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 - offF

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**

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 **oN** 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**

Intentionally blank

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 REQUEST 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**

NERR - No **ERR**or (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 REQUEST 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 REQUEST 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. (Bipolar operation ab SW version BCP100)

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 REQUEST means that line-in-command is wrong or no polarity switch build-in.
DATA ERROR 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, will the polarity 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 request error will be returned if issued.

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

Nothing else is affected.

PRINT - PRINT

Command: PRINT'cr'

Example: PRINT
Syntax: PRINT'cr'

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

or Error message

Errors: SYNTAX ERROR means wrong syntax.

ILLEGAL REQUEST 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 REQUEST means that line-in-command is wrong.

Description:

The **RA** command reads the currently numerical set value in 10e-4 amps. 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 REQUEST 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: **LOC**

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 REQUEST means that the line-in-command is either remote line or unlocked in local line.

COMMAND ALREADY ACTIVE means that the command has been given already and is still active.

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 REQUEST 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 **OFF** command also clear all the non pending interlock's.

Nothing else is affected.

Related commands: **F, N**

Intentionally blank

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 REQUEST 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**

S1 continued

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

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

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

CHARACTER NO. CONTENTS

1	Off	Status	High=!
2	Rem/Loc	Status	High=!
3	Spare Interlock (Ext intl. 4)	Interlock	High=!
4	Spare		Always=.
5	Spare		Always=.
6	Spare		Always=.
7	!= % , . = AMPS and VOLTS		
8	Spare INTERLOCK (Ext intl. 1)	Interlock	High=!
9	Standby	Status	High=!
10	SUM - INTERLOCK.	Interlock	High=!
11	DC OVER CURRENT (OCP)	Interlock	High=!
12	Over Voltage Protection	Interlock	High=!
13	On	Status	High=!
14	SPARE INTERLOCK (Ext intl. 2)	Interlock	High=!
15	Mains Failure	Interlock	High=!
16	Current Limit	Status	High=!
17	Earth Leakage Failure	Interlock	High=!
18	Converter Over Voltage	Interlock	High=!
19	MPS OVERTEMPERATURE.	Interlock	High=!
20	Spare		Always=.
21	Spare		Always=.
22	Spare Interlock (Ext intl. 3)	Interlock	High=!
23	MPS NOT READY	Status	High=!
24	MPS Fan Fault	Interlock	High=!

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 REQUEST 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

6 0 0 0 0 1

Equals one nibble.

STB - Standby/ON

Command: STB'cr'

Example: STB
Syntax: STB'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set.

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

ILLEGAL REQUEST means that line-in-command is wrong.

Description:

The **STB** command switches on the power supply (main contact) and clamps 'out4'. This command is similar to N command but N command does not clamp 'out4'.

Note : This command is only valid for MODE 2 set in ESC<NFSMODE. STB have no effect on the other modes.

All settings are left unaffected.
This command cannot be used in **LALL** mode.

Nothing else is affected.

Related commands: **F, N**

TYPE - TYPE

Command: TYPE 'cr'

Example: TYPE
Syntax: TYPE 'cr'

Answer: T'sp'type'cr'

Type: ASCII digit 0 or 8
If 0, fast 12bit ADC is selected
If 8, standard 16bit ADC is selected

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

ILLEGAL REQUEST means that line-in-command is wrong.

Description:

The **type** command returns a code, used by the control panel to determine the type of the AD-channel implemented for the current read back.

Nothing else is affected.

TD - Test DAC

Command: TD'sp'set'cr'

set : ASCII digit 0 to 8

Example: TD 4

Syntax: TD'sp'4'cr'

Answer: No answer, except errors

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

ILLEGAL REQUEST 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

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

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 REQUEST 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:
xxxxxxxxxxxxxxxxxxxxxxxx'cr'
xxxxxxxxxxxxxxxxxxxxxxxx'cr'
xxxxxxxxxxxxxxxxxxxxxxxx'cr'

or Error message

Errors: SYNTAX ERROR means wrong syntax.

ILLEGAL REQUEST 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 10e-4 Ampere

Command: WA'sp'dac'cr'

dac: digits 000000 to 999999 in 10e-4 Ampere.

!!!! 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). Do not use comma in the value field for writing to the DAC for output set current.

Example: WA 0480

Syntax: WA'sp'0480'cr'

For leading zeroes this means 48000 in 10e-4 Ampere

For trailing zeroes this means 480 in 10e-4 Ampere

Answer: No answer, except errors.

or OK if auto answer mode is set.)

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

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

ILLEGAL REQUEST 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.

Description:

The command **WA** writes a value between 0 and 999999 in 10e-4 Ampere to the regulation module DAC for output set current. Preferred command is though DA 0.

Nothing else is affected.

Related commands: **DA, RA**

Esc<ADR - ADdRes 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 REQUEST** means that line-in-command is wrong.

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

DATA ERROR 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. In remote line, this can be done by the command ESC<CPURESET

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 - ADdRes 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 REQUEST** means that line-in-command is wrong.

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

DATA ERROR 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
val2: ASCII 00000 to 999999

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 REQUEST means that line-in-command is wrong.

DATA ERROR 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

F: Gain adjustment (To full scale)

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

Affected commands: **NONE**

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 REQUEST means that line-in-command is wrong.

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

DATA ERROR 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, 2 Stop bits" for both local and 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 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

or Error message

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

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

DATA ERROR 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.

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

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: ? and	1: ? and
b6:	ERR response	0: only	0: ERR code 1: ERR text
b7:	not used		
b8:	not used		

Those in bold are default setting.

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 REQUEST** means that line-in-command is wrong.

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

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 a missing space between the command and parameter or wrong syntax.

Description:

The command 'Esc'<CPURESET resets the CPU. This command is needed after changing settings in ESC<BAUD, ESC<ADR and ESC<COLDBOOT which need a hardware reset before the new settings take effect. Esc<CPURESET is a watchdog reset and takes 2 sec. before the system is reset.

Esc<DASET - Digital to Analog converter SETting

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

ch: ASCII digit 0 to 15
val1: Z, F or L *
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 a missing space between the command and parameter or wrong syntax.

ILLEGAL REQUEST means that line-in-command is wrong.

DATA ERROR 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 **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 settings. !

DASET continued

val1 interpretation

Z: Offset adjustment (to Zero)
F: Gain adjustment (To full scale)
L: Set value limit (val2) *

* Be aware when performing the full scale adjustment 'F', the set value can not be set higher than the limit value (set by the 'L' setting). That is, perform the full scale adjustment before the limit setting.

Related commands: **'Esc'<ADSET**

Affected commands: **NONE**

Esc<LINE - Serial LINE working mode setup write.

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

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 is kept for non entered bits.

Answer: No answer except errors.

or OK if autoanswer mode is set.

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

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

DATA ERROR 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: Not used.

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: Not used.

b8: Not used.

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 REQUEST** means that line-in-command is wrong.

Description:

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

Esc<NFSMODE - ON/OFF/STANDBY Mode setup write

Command: 'Esc'<NFSMODE'sp'mode
mode:: ASCII 0,1 or 2

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

Current setting are kept for non entered bits.

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

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

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

DATA ERROR 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'<NFSMODE command sets up the ON/OFF/STANDBY mode operation. The setting becomes operational immediately after then saving in the non volatile RAM.

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

'Mode' definition

Mode=0: Direct ON & OFF Default mode
Mode=1: Stand by mode 1
Mode=2: Stand by mode 2

See document MSYS7000a.doc at chapter 3 for further information about the working principia of the modes.

Esc<NFSMODE - ON/OFF/STANDBY Mode setup read

Command: 'Esc'<NFSMODE'

Example: 'Esc'<NFSMODE
Syntax: 'Esc'<NFSMODE'cr'

Answer: NFSMODE'sp'mode'cr'

Mode: ASCII digit 0 to 2
Mode=0: Direct ON & OFF
Mode=1: Stand by mode 1
Mode=2: Stand by mode 2

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

Description:

The 'Esc'<NFSMODE command reads the ON/OFF/STANDBY mode setup.



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 a viable.

SW RAMP PROFILE COMMANDS “Equal Time slot method”. Summary

Available from SW version BCP100

R Write data to the stack.

RAMP Control the stack operation

RAMPSET Configure the ramp operation

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, Stack is closed

or OK if autoanswer mode is set.

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

DATA ERROR 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 IS CLOSED Stack is closed. The stack has been terminated with R S.

Description:

The **R** (write) command writes a data points to the “Equal time slot” stack. All data must enter 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 500 values may be entered, but only 3 is minimum.

In order to write new data to the stack, the stack must be empty first by command **RAMPSET C**

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 0.000000
R 0.123456
R 0.879000
R 0.300000
R 0.000000

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

Description:

The **R** (read) command dumps 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 (pauses) 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 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.	Resets all control parameters to normal (one shoot). If the stack is running, will the execution 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: **DATA ERROR** means that parameter format incorrect or 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 IS EMPTY means that the stack is empty. Running the stack is not possible.

STACK IS RUNNING means that the stack is running
MPS NOT ON means that MPS is not turned on.

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 stopping a running ramp with RAMP S, the ramp should stop completely and not be started until giving news commands to prepare for trig or start the ramp.

Related commands: **RAMPSET, R**

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

Description:
 The **RAMP** (read) command reads the present status of the “Equal time slot” stack.

Related commands: **RAMPSET, R**

RAMPSET - SETing up the stack parameter “Equal time slot mode”

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

Time: Time. Sets the time slot value in 0.0025 second steps.
 Mult: Mult. Sets the multiplicand factor. (Gain control)
 TrDly: Delay. Predefined as 0 (zero)

L: 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. Normal. Set ramp execution to normal. (No loops)
 C: Clear. Clear the stack and parameters (1.0sec,*1.0, N=true)

Example: RAMPSET 0.01,0.98657,0,W sets time, multiplicand, trigger delay & in wait looping
 RAMPSET 0 Sets time to max resolution=0.0025sec. multiplicand=1 & N=true
 RAMPSET ,0.98657 Sets only the multiplicand other parameters left unchanged.
 RAMPSET Sets parameters to normal execution.
 RAMPSET C Clear the stack and parameters

Answer: No answer, except errors

Errors: **DATA ERROR** means that parameter format incorrect or 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.

RAMPSET Continued.

Description:

The **RAMPSET** (write) command sets 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.

Related commands: **RAMP, R**

