APP2. 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 formats and further detail description.
These commands are optionally available.

SW RAMP PROFILE COMMANDS "Arbitrary point method". summary

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

X is a number from 0 to 9

SW RAMP PROFILE COMMANDS for "Equal time slot method". summary

Available from SW version SCC107

R value	Writes data to the stack.
RAMP [RSHTN],[LWB]	Controls the stack operation.
RR	Read ramp status
RAMPSET Time,Multiplicant, [LWN],C	Configures the ramp operation.

SW RAMP PROFILE COMMANDS for "Auto slew rate method". summary

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

CONT - CONTinue sequence

Command: CONT'cr'

Example: CONT
Syntax: CONT'cr'

Answer: No answer, except errors

or OK if autoanswer mode is set. (From SW version SCS110)

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. (From SW version SCS110)

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. (From SW version SCS110)

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. (From SW version SCS110)

Errors:

SYNTAX ERROR,	means that there are no sequence running. I.e. that it already has been HALT'ed or no sequence has been triggered.
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 are 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. (From SW version SCS110)

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" From version SCS107

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

Value: Floating point number between 0.0000000 to 1.0000000
H: Halts the stack execution (Same as pause). (From SW version SCS111)
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. (From SW version SCS110)

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 is used to write 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. 3 is minimum. "**R H**" is a Halt point where the stack execution will be paused. The execution can be restarted trough HW or SW. Please refer to the RAMP command A new stack may be down loaded during execution for 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" From version SCS107

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' From version SCS107

R: Run. Starts the stack execution.
S: Stop. Stops the stack execution.
H: Halt. Halts the stack execution.
T: Trig. Starts the stack execution if in wait mode. A HW trig pulse is also generated for synchronizing connected slaves. If not in Wait mode will the stack be transferred to the wait mode first (eg. "RAMP T,W") and the second "RAMP T" will start the execution.

L: Loop. Loops the stack. After issuing the last point will the stack automatically be restarted.
W: Wait. Waits for a HW trigger signal or a "RAMP T" command to start stack execution.
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/SW trig or starts the execution if already in the wait mode.
Syntax: RAMP'sp'T,W'cr'
Or
RAMP R,L Runs (starts) the stack in a loop operation
Syntax: RAMP'sp'R,L'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.
ERR_CANNOT_EXECUTE_CMD, means A stack is already running.

Description:
The **RAMP** (write) command is used to set the running condition of the "Equal time slot" stack. I is also possible to change the running condition [LWN] that was prior set with the RAMPSET command.

Related commands: **RAMPSET, R**

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

From version SCS107

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.
RAMP H N Stack is halted in normal mode

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

Description:

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

Related commands: **RAMPSET, R**

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

From version SCS107

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. (From SW version SCS109)
Minimum resolution is one µ-step (=78µS.)
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. 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

From version SCS107

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.

Trigger delay value is given from SW version SCS109.

Related commands: **RAMP, R**

RR - Read ramp/stack status

From version SCS113

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. (From SW version SCS110)

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 specifification.

RSP continued

Description:

The **RSP** command is used to read a set of parameters from a given position in a given sequence-stack.

No pointers are affected.

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

RWSP - Reset Read Sequence Pointer

Command: RWSP'sp'stack'cr'

stack: ASCII digit 0 to 15

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

Answer: No answer, except errors

or OK if autoanswer mode is set. (From SW version SCS110)

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 is used to read 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 a 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. (From SW version SCS110)

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. (From SW version SCS110)

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 (latest from SW version SCC113)
From SW version SCS 114 will the STOP command work with all ramp profile executions.

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 frozen 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. (From SW version SCS110)

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.

ERR_CANNOT_EXECUTE_CMD, means A stack is already running.

Description:

The **SYNC** command is used to arm 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.
ERR_CANNOT_EXECUTE_CMD,	means A stack is already running.

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 is used to write 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. (From SW version SCS110)

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 is used to load 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 version SCC 113)

Command: 'esc'<SLOPETIME'sp'val1','val2',val3'

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 SCC114)

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

Answer: No answer except errors.

or OK if autoanswer mode is set. (From SW version SCS110)

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'<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" command that immediately thereafter will be executed.
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 micro steps interpolation in-between.

Related commands: **RR**

Esc<SLOPETIME - (read)

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 SCC114)

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