SCU

ASCII Programming Interface

www.smaract.com

SmarAct
PERFECT MOTION

The information given in this document was carefully checked by our team and is constantly updated. Nevertheless, it is not possible to fully exclude the presence of errors. In order to always get the latest information, please contact our technical sales team.

SmarAct GmbH, Schuette-Lanz-Strasse 9, D-26135 Oldenburg
Phone: +49 (0) 441 - 800879-0, Telefax: +49 (0) 441 - 800879-21
Internet: www.smaract.com, E-Mail: info@smaract.com

Document Version: 2.2.5

TABLE OF CONTENTS

1 *INTRODUCTION*

This document describes the ASCII command set, which is used by Simple Control Units (SCUs) with RS232 interface.

1.1 RS232 Interface

Some notes regarding SCU with RS232 interface.

On power-up the device will initialize itself and wait for a valid RS232 signal (logic high) on its RX line. The device will answer with a logic high on its TX line and is then ready to receive commands.

The interface parameters are set to "8N1" (8 data bits, no parity, one stop bit). The factory default baud rate is set to 9600. Depending on the product version the baud rate may be configured by the user (see "`CB` – Configure RS232 Baud rate" command).

1.2 Command Format

Each command consists of an initial character (':', 0x3a), an ASCII string coding the actual command (hereafter referred to as *command string*) and a termination character (line feed, 0x0a). Empty strings (i.e. a line feed character that follows a colon character) are ignored. All characters between a line feed and a colon are also ignored.

The general command string format is:

<command><channelIndex>[<parameter><value>] ... [<parameter><value>]

For HCU-1D / CU-1D devices only channelindex 0 is valid. Using a different channelindex will return an "Invalid channel error" (Code 3).

Some parameters are required, some are optional. Optional parameters may be given in any order. If omitted they are treated as given with a default value of 0.

1.3 Answer Format

Some commands cause the device to send an answer, such as status information. Answers are again ASCII strings (hereafter referred to as *answer string*) that are preceded by a colon and terminated by a line feed character.

As with the commands the general answer string format is:

<answer><channelIndex>[<parameter><value>] ... [<parameter><value>]

In this case the channel index is the answer source. If the answer does not originate from a specific channel, then the channel index will be 255, indicating the device in general.

1.4 Command Summary

Table 1.1 summarizes the instruction set of the SCU device. Upper case letters mark a single ASCII character. An 'x' marks a parameter value.

The command and answer strings are described in more detail below (see chapter 2). Note that not all commands might be available, depending on your product version.

Table 1.1 – Initialization Commands

Command	Meaning	Answer	Meaning
Initialization Commands			
I	get device Identification	Ix	device Identification
GID	Get device ID number	IDx	device ID number
V	Get firmware Version	Vx.x.x	firmware Version
R	Device reset		
Configuration Commands			
AxAx	Set step Amplitude		
SSPxSxAxFx	Set Step Parameters		
SZx	Set Zero position		
SCLFxFx	Set Closed-Loop max Frequency		
GCLFx	Get Closed-Loop max Frequency	CLFxFx	current frequency
GSPx	Get Sensor Present	SPxP / SPxN	sensor Present / Not present
SSTxTx	Set Positioner Type		
GSTx	Get Positioner Type	STxTx	positioner type
SPAxAxFxBx	Set Positioner Alignment		
GPAx	Get Positioner Alignment	PAxAxFxBx	alignment, forward amplitude, backward amplitude
SSCxIxSx	Set Scale		
GSCx	Get Scale	SCxIxSx	inversion, offset
SSDxDx	Set Safe Direction		
GSDx	Get Safe Direction	SDxDx	direction
SCPxPxVx	Set Channel Property		
GCPxPx	Get Channel Property	CPxPxVx	channel Property
GSPRx	Get System Property	SPRxVx	system Property
SRCxRx	Set Report On Complete		
SERxEx	Set Error Report		
Movement Control Commands			
UxFxAxSx	move Up		
DxFxAxSx	move Down		
MPAxPxHx	Move to Position Absolute		
MPRxPxHx	Move to Position Relative		

Continued on next page

Table 1.1 – *Continued from previous page*

Command	Meaning	Answer	Meaning
MAAxAxRxHx	Move to Angle Absolute		
MARxAxRxHx	Move to Angle Relative		
MTRxHxZx	Move To Reference		
MESxDx	Move to End-Stop		
CSx	Calibrate Sensor		
Sx	Stop		
Positioner Feedback Commands			
GPx	Get Position	PxPx	Position
GAx	Get Angle	AxAxRx	Angle and Revolution
Mx	Get Movement Status	Mxx	Movement status
GPPKx	Get Physical Position Known	PPKxKx	Known status
Miscellaneous Commands			
Kx	Keep alive		
Ex	get Error status	Ex	Error status
CBx	Configure RS232 Baud rate.	CBx	Configured Baud rate

1.5 Errors

Whenever a command string cannot be processed an error is generated. There are two modes of error handling. The *auto error report mode* and the *non auto error report mode*, the latter being the default.

- In non auto error report mode commands that don't naturally invoke an answer return no answer string, but instead write the resulting error code to an internal register that may be read out via the E-command. If no error occurred a value of 0 is written to the register.
- In auto error report mode all commands return an answer string in any case. Query commands may return the normal answer string or an error string. Other commands always return an error string (with code 0 in case of no error). Note that in this mode the E-command always returns a value of 0, since errors are already implicitly reported and the internal error code register has been set to 0.

Returned error strings have the format Excode. The error code <code> indicates the cause of the failure. A code of 0 indicates no error and corresponds to an acknowledge in auto error report mode.

Please refer to section 4.2 for a list of error codes.

2 *COMMAND DESCRIPTION*

2.1 Initialization Commands

2.1.1 `I` – get device Identification

Description:
This command may be used to verify that you are communicating with the correct RS232 device. It returns a string that identifies the device.

Command String Format:
`I`

Parameters: none

Answer:
`I`<identification>

- <identification>: Device identification.

Example:
Command: `I`

Answer: `ISmarAct HCU-3D`

Requests the device identification. Depending on the model, the identification is one of:

- `SmarAct HCU-1D`
- `SmarAct HCU-3D`
- `SmarAct CU-1D`
- `SmarAct CU-3D`

2.1.2 GID – Get device ID number

Description:
This command may be used to physically identify a device. Each device has a unique ID which makes it possible to distinguish one from another. Useful when communicating with several devices.

Command String Format:
`GID`

Parameters: none

Answer:
`ID`<IDnumber>

- <IDnumber>: Decimal device ID number.

Example:
Command: `GID`

Answer: `ID1234567890`

Requests the device ID number, which is “1234567890”.

2.1.3 V – get firmware Version

Description:
This command may be used to retrieve the firmware version that is installed on the device.

Command String Format:
`V`

Parameters: none

Answer:
`V`<versionHigh>.<versionLow>.<versionBuild>

- <versionHigh>: High number of the firmware version
- <versionLow>: Low number of the firmware version
- <versionBuild>: Build number of the firmware version

Example:
Command: `V`

Answer: `V1.2.3`

Requests the version of the firmware on the device, which is “1.2.3”.

2.1.4 R – device Reset

Description:
When this command is sent to the device it will perform a reset. It has the same effect as a power-down/power-up cycle. No answer is sent before the reset. The device is ready to receive commands again after the TX line has gone logic high.

Command String Format:
`R`

Parameters: none

Answer:
none

Example:
Command: `R`

Initiates a reset of the device. It will be ready to receive commands again after the TX line has gone logic high.

2.2 Configuration Commands

2.2.1 A – set step Amplitude

Description:
This command sets the amplitude for the selected channel. An ongoing movement of this channel is implicitly stopped. If the channel index is set to 99 the command will be executed for all channels at once. Please note that this command is deprecated. The SSP command should be used instead.

Command String Format:
A<channel>A<amplitude>

Parameters:

- <channel>: Zero-based channel index. If the index is set to 99 the command will be executed for all channels. Valid indices are 0, 1, 2 and 99.
- <amplitude>: Amplitude that the steps are performed with. Lower amplitude values result in a smaller step width. The parameter must be given in tenths of Volts and has a valid range of 150 to 1000. 0 corresponds to 0 V, 1000 to 100 V.

Answer:
none

Example:
Command: A2A1000

Sets the amplitude of channel 2 to 100 V. If the positioner is currently moving, it is stopped.

2.2.2 SSP – Set Step Parameters

Description:
This command sets the step parameters for a channel without performing an actual movement. It may be useful to preset the parameters and then use short command strings such as D0.

An ongoing movement of this channel is implicitly stopped. If the channel index is set to 99 the command will be executed for all channels at once.

Command String Format:
SSP<channel>S<steps>A<amplitude>F<frequency>

Parameters:

- <channel>: Zero-based channel index. If the index is set to 99 the command will be executed for all channels. Valid indices are 0, 1, 2 and 99.
- <steps>: Number of steps that are to be performed per burst. The valid range is 1 to 30000. The special value of 30000 indicates an unbounded move.
- <amplitude>: Amplitude that the steps are performed with. Lower amplitude values result in a smaller step width. The parameter must be given in tenths of Volts and has a valid range of 150 to 1000. 0 corresponds to 0 V, 1000 to 100 V.
- <frequency>: Frequency in Hz that the steps are performed with. The valid range is 1 to 18500.

Answer:
none

Example:
Command:

1. SSP2S10F500
2. D2
3. D2

Configures a burst of 10 steps at 500 Hz with an unchanged amplitude and then performs two bursts in down direction.

2.2.3 SZ – Set Zero position

Description:
For positioners that have a sensor installed, this command may be used to define the current position of the positioner as the zero position.

If the positioner "knows" its physical position (via the "MTR – Move To Reference" command) when sending this command, the position will be saved to non-volatile memory. On future power-ups it will recall its physical position automatically after a "MTR – Move To Reference" command. See section 2.6 "Defining Positions" for more information.

Command String Format:
SZ<channel>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.

Answer:
none

Example:
Command: SZ0

Defines the current position of positioner 0 as its zero position.

NOTICE
This feature is not available with all sensors. Refer to section 4.3 "Positioner Types" for further information.

2.2.4 SCLF – Set Closed-Loop max Frequency

Description:
For positioners that have a sensor installed, this command may be used to define the maximum frequency that the positioners are driven with when issuing closed-loop movement commands (e.g. "MPA – Move to Position Absolute"). This parameter may be set for each channel independently. Once set, all subsequent closed-loop commands will execute with the new setting.

Command String Format:
SCLF<channel>F<frequency>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.
- <frequency>: Defines the maximum driving frequency in Hz. The valid range is 1 to 18500. The default value is 5000.

Answer:
none

Example:
Command: SCLF0F4000

Sets the maximum closed-loop frequency of channel 0 to 4 kHz.

2.2.5 GCLF – Get Closed-Loop max Frequency

Description:
This is the inverse command to "SCLF – Set Closed-Loop max Frequency". It may be used to read out the current setting for the maximum driving frequency for closed-loop commands.

Command String Format:
GCLF<channel>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.

Answer:
CLF<channel>F<frequency>

- <channel>: Zero-based channel index.
- <frequency>: Maximum closed-loop driving frequency in Hz.

Example:
Command: GCLF0

Answer: CLF0F5000

Returns the maximum driving frequency that is currently set for channel 0, which is the default 5 kHz.

2.2.6 GSP – Get Sensor Present

Description:
This command may be used to check whether a sensor is present on a positioner or not.

Command String Format:
GSP<channel>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.

Answer:
SP<channel>P

in case a sensor is present or

SP<channel>N

in case no sensor is present.

- <channel>: Zero-based channel index.

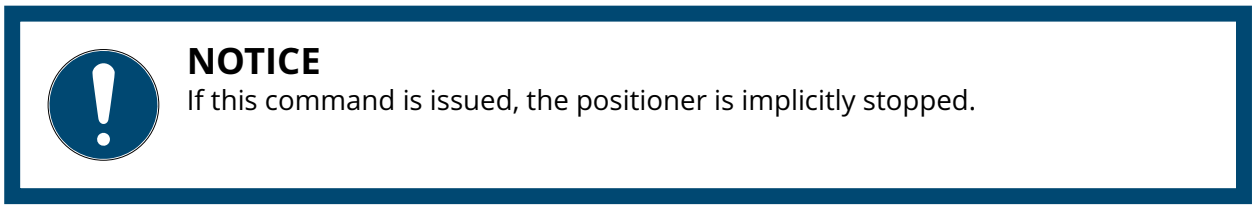
Example:
Command: GSP1

Answer: SP1P

Checks whether a sensor was detected on channel 1. In the example the answer is positive.

2.2.7 SST – Set Positioner Type (formerly Set Sensor Type)

Description:
When using positioners with integrated sensors, this command may be used to tell a channel what type of positioner is connected (e.g. linear or rotary). The type affects position calculation and functions that may be called for a channel (see for example "GP – Get Position" and "GA – Get Angle" commands).
Note that each channel stores this setting to non-volatile memory. Consequently, there is no need to call this function on every initialization.



For more information on positioner types, please refer to **??**.

Command String Format:
SST<channel>T<type>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.
- <type>: The type code of the positioner. See section 4.3 "Positioner Types" for a list of valid positioner types.

Answer:
none

Example:
Command: SST2T1

Sets the positioner type of channel 2 to 1.

2.2.8 GST – Get Positioner Type (formerly Get Sensor Type)

Description:
Inverse command to "SST – Set Positioner Type (formerly Set Sensor Type)". It may be used to check which positioner type is currently configured for a channel.

For more information on positioner types, please refer to **??**.

Command String Format:
GST<channel>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.

Answer:
ST<channel>T<type>

- <channel>: Zero-based channel index.
- <type>: Currently configured positioner type code. See section 4.3 "Positioner Types" for a list of positioner types.

Example:
Command: GST0

Answer: ST0T1

Returns the positioner type code that is currently configured for channel 0, which is 1 in this case.

2.2.9 SPA – Set Positioner Alignment

Description:
When using positioners with integrated sensors, this function is useful for positioners that are mounted vertically and carry high loads. It may be used to tell a channel how the positioner is physically aligned (horizontally or vertically). This setting affects only closed-loop control of a positioner. Open-loop commands are unaffected. Please refer to section 3.2 "Vertical Axis Configuration" for more information.

Note that each channel stores this setting to non-volatile memory. Consequently, there is no need to call this command on every initialization.

Command String Format:
SPA<channel>A<alignment>F<forwardAmplitude>B<backwardAmplitude>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.
- <alignment>: Alignment of the positioner. Must be either 0 (horizontal, default) or 1 (vertical).
- <forwardAmplitude>: Only valid for vertical alignment. Omit for horizontal alignment. Specifies the step amplitude that is used when the positioner is moved in forward direction while in closed-loop control. Lower amplitude values result in a smaller step width. The parameter must be given as an integer value in tenths of Volts. The valid range is 150 (15 V) to 1000 (100 V).
- <backwardAmplitude>: Only valid for vertical alignment. Omit for horizontal alignment. Specifies the step amplitude that is used when the positioner is moved in backward direction while in closed-loop control. Lower amplitude values result in a smaller step width. The parameter must be given as an integer value in tenths of Volts. The valid range is 150 (15 V) to 1000 (100 V).

Answer:
none

Example:
Command: SPA0A1F1000B600

Configures channel 0 for vertical positioning with a backward amplitude of 60 V.

2.2.10 GPA – Get Positioner Alignment

Description:
Inverse command to "SPA – Set Positioner Alignment". It may be used to check which positioner alignment is currently configured for a channel.

Command String Format:
GPA<channel>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.

Answer:
PA<channel>A0

in case of horizontal alignment, or

PA<channel>A1F<forwardAmplitude>B<backwardAmplitude>

in case of vertical alignment.

- <channel>: Zero-based channel index.
- <forwardAmplitude>: Currently configured amplitude for forward movement given in 0.1 V.
- <backwardAmplitude>: Currently configured amplitude for backward movement given in 0.1 V.

Example:
Command: GPA0

Answer: PA0A0

Returns the alignment that is currently configured for channel 0, which is horizontal in this case.

2.2.11 SSC – Set Scale

Description:
When using positioners with integrated sensors this command may be used to configure the logical scale for a channel. The logical scale can be inverted, thus altering the positive and negative moving direction of the positioner. Furthermore the measuring scale of a positioner can be shifted using a given offset. Please see sarefsec:DefiningPositions for more information.

When calling this function the setting is saved to non-volatile memory. Therefore, the scale does not have to be configured in each session. Note though that if the scale inversion is changed for positioner types that are referenced via end stop then the sensor must be calibrated again for the "MTR – Move To Reference" command to work properly (see section 4.3 "Positioner Types").

CAUTION
Please note that only the logical scale of the positioner will be inverted when the inverted value has changed. Parameters like the *SafeDirection* will not be altered. Thus the positioner will move in the opposite direction when e.g. calling "MTR – Move To Reference" with the same parameters prior to the inversion change.

Command String Format:
SSC<channel>I<inversion>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.
- <inversion>: Sets the scale inversion. Must be either 0 (normal) or 1 (inverted).
- <scale>: Sets the desired scale shift relative to the physical scale of the positioner. The value is given in nanometers for linear positioners and in micro degrees for rotary positioners.

Answer:
none

Example:
Command: SSC0I0S1000000

Configures the scale offset of channel 0 to +1 mm relative to the physical scale.

2.2.12 GSC – Get Scale

Description:
Inverse command to "SSC – Set Scale". It may be used to check the currently configured scale offset and inversion of a channel.

CAUTION
Please note that only the logical scale of the positioner will be inverted when the inverted value has changed. Parameters like the *SafeDirection* will not be altered. Thus the positioner will move in the opposite direction when e.g. calling MTR with the same parameters prior to the inversion change.

Command String Format:
GSC<channel>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.

Answer:
SC<channel>I<inversion>

- <channel>: Zero-based channel index.
- <inversion>: Currently configured scale inversion. Will be either 0 (normal) or 1 (inverted).
- <scale>: Currently configured scale offset.

Example:
Command: GSC0

Answer: SC0I1S0

Returns the scale inversion and offset that is currently configured for channel 0. In this case the scale is inverted and the offset is 0.

2.2.13 SSD – Set Safe Direction

Description:
When using positioners with integrated sensors this command may be used to configure the safe direction for a channel. The safe direction is the direction in which the positioner may safely move while calibrating (see "CS – Calibrate Sensor" command) or moving to the reference position (see "MTR – Move To Reference" command).

When calling this function the setting is saved to non-volatile memory. Therefore, the safe direction does not have to be configured in each session. Note though that if the safe direction is changed for positioner types that are referenced via end stop then the sensor must be calibrated again for the "MTR – Move To Reference" command to work properly.

Command String Format:
SSD<channel>D<direction>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.
- <direction>: Safe direction. Must be either 0 (backward) or 1 (forward).

Answer:
none

Example:
Command: SSD0D1

Configures the safe direction for channel 0 to forward.

NOTICE
This feature is not available with all sensors. Refer to section 4.3 "Positioner Types" for further information.

2.2.14 GSD – Get Safe Direction

Description:
Inverse command to "SSD – Set Safe Direction". It may be used to check which safe direction is currently configured for a channel.

Command String Format:
GSD<channel>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.

Answer:
SD<channel>D<direction>

- <channel>: Zero-based channel index.
- <direction>: Currently configured safe direction. Will be either 0 (backward) or 1 (forward).

Example:
Command: GSD0

Answer: SD0D1

Returns the safe direction that is currently configured for channel 0, which is forward in this case.

NOTICE
This feature is not available with all sensors. Refer to section 4.3 "Positioner Types" for further information.

2.2.15 SCP – Set Channel Property

Description:
This command can be used to configure channel specific properties. For a list of available channel properties refer to subsection 4.4.2 “Channel Properties”.

Command String Format:
SCP<channel>P<property>V<value>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.
- <property>: Defines the property to be set.
- <value>: The value to be set depending on the selected property.

Answer:
none

Example:
Command: SCP0P3V500

Sets the target reached threshold for channel 0 to 500 nm. Thus the target position is now considered as reached within a range of ±500 nm.

2.2.16 GCP – Get Channel Property

Description:
Inverse command to "SCP – Set Channel Property". This command may be used to retrieve the property of a channel. For a list of available channel properties refer to subsection 4.4.2 "Channel Properties".

Command String Format:
GCP<channel>P<property>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.
- <property>: Defines the property to be retrieved.

Answer:
CP<channel>P<property>V<value>

Example:
Command: GCP0P3

Answer: CP0P3V300

Retrieves a target reached threshold of 300 nm for channel 0.

2.2.17 GSPR – Get System Property

Description:
This command can be used to retrieve the current value of a given system property. For a list of available system properties refer to subsection 4.4.1 “System Properties”.

Command String Format:
GSPR<property>

Parameters:

- <property>: Index of the desired property.

Answer:
SPR<property>V<value>

Example:
Command: GSPR1

Answer: SPR1V28

Returns a current system temperature of 28 °C.

2.2.18 SRC – Set Report on Complete

Description:
This command may be used to instruct a channel to send a string when a movement command has completed. The string has the format: C<channel>.

Command String Format:
SRC<channel>R<report>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.
- <report>: Enables (1) or disables (0, default) sending of the string.

Answer:
none

Example:
Command: SRC0R1

Enables reporting of movement completion for channel 0. A completed movement on this channel will now be reported by the string C0.

2.2.19 SER – Set Error Report

Description:
This command can be used to enable or disable sending of strings when a channel error occurs.

The generated error string has the following format: CE<channel>E<error>.

Command String Format:
SER<channel>E<error>R<report>

Parameters:

- <channel>: Zero-based channel index of the device. Valid indices are 0, 1 and 2.
- <error>: Code of the error to be enabled / disabled. (currently configurable errors below)

Code (<error>)	Meaning
21	End Stop Reached
22	Targeting Timeout

- <report>: Enables (1) or disables (0, default) sending of the string.

Answer:
none

Example:
Command: SER0E21R1

Enables reporting if a movement is aborted because an End Stop on channel 0 was reached. An End Stop on this channel will now be reported by the following string: CE0E21.

2.3 Movement Control Commands

2.3.1 U – move Up

Description:
This command causes the selected channel to perform a movement in “up” direction with the given frequency, amplitude and number of steps. If the channel index is set to 99 the command will be executed for all channels at once.

The F, A and S parameters may be omitted in which case the last value for the selected channel is used. If omitted on the first call, default values are used. These are 1 kHz for the frequency, 100 V for the amplitude and “unlimited” for the number of steps.

Command String Format:
U<channel>F<frequency>A<amplitude>S<steps>

Parameters:

- <channel>: Zero-based channel index. If the index is set to 99 the command will be executed for all channels. Valid indices are 0, 1, 2 and 99.
- <frequency>: Frequency in Hz that the steps are performed with. The valid range is 1 to 18500.
- <amplitude>: Amplitude that the steps are performed with. Lower amplitude values result in a smaller step width. The parameter must be given in tenths of Volt and has a valid range of 150 to 1000. 0 corresponds to 0 V, 1000 to 100 V.
- <steps>: Number of steps to be performed. The valid range is 1 to 30000. The special value of 30000 indicates an unbounded move.

Answer:
none

Example:
Command: U99F5A500

Moves all positioners in positive direction with a step frequency of 5 Hz, an amplitude of 50 V and the corresponding last number of steps. If a positioner is currently moving, the previous movement command is overridden.

2.3.2 D – move Down

Description:
This command causes the selected channel to perform a movement in “down” direction with the given frequency, amplitude and number of steps. If the channel index is set to 99 the command will be executed for all channels at once.

The F, A and S parameters may be omitted in which case the last value for the selected channel is used. If omitted on the first call, default values are used. These are 1 kHz for the frequency, 100 V for the amplitude and “unlimited” for the number of steps.

Command String Format:
D<channel>F<frequency>A<amplitude>S<steps>

Parameters:

- <channel>: Zero-based channel index. If the index is set to 99 the command will be executed for all channels. Valid indices are 0, 1, 2 and 99.
- <frequency>: Frequency in Hz that the steps are performed with. The valid range is 1 to 18500.
- <amplitude>: Amplitude that the steps are performed with. Lower amplitude values result in a smaller step width. The parameter must be given in tenths of Volt and has a valid range of 150 to 1000. 0 corresponds to 0 V, 1000 to 100 V.
- <steps>: Number of steps to be performed. The valid range is 1 to 30000. The special value of 30000 indicates an unbounded move.

Answer:
none

Example:
Command: D1F5

Moves the positioner connected to channel 1 in negative direction with a step frequency of 5 Hz and the amplitude and the number of steps that channel 1 was last driven with. If the positioner is currently moving, the previous movement command is overridden.

2.3.3 MPA – Move to Position Absolute

Description:
Instructs a positioner to move to a specific position via closed-loop control. This command is only executable by a positioner that has a sensor integrated.

While executing the command the positioner will have a movement status of `T` (targeting). If a hold time was specified the movement status will change to `H` (holding) after having reached the target position. After the hold time has passed, the movement status will be `S` (stopped).

Command String Format:
`MPA`<channel>`P`<position>`H`<holdTime>

Parameters:

- <channel>: Zero-based channel index. Valid indices are `0`, `1` and `2`.
- <position>: Absolute position to move to in µm.
- <holdTime>: Time in ms that the target position is to be held. The valid range is `0` to `60000`. A value of `60000` represents infinity.

Answer:
none

Example:
Command: `MPA0P1000H1000`

Moves the positioner connected to channel 0 to the position 1 mm which is then held for one second.

2.3.4 MPR – Move to Position Relative

Description:
Instructs a positioner to move to a position via closed-loop control, which is given in relation to the current position. This command is only executable by a positioner that has a sensor integrated.

While executing the command the positioner will have a movement status of `T` (targeting). If a hold time was specified the movement status will change to `H` (holding) after having reached the target position. After the hold time has passed, the movement status will be `S` (stopped).

Command String Format:
`MPR`<channel>`P`<position>`H`<holdTime>

Parameters:

- <channel>: Zero-based channel index. Valid indices are `0`, `1` and `2`.
- <position>: Relative position to move to in µm.
- <holdTime>: Time in ms that the target position is to be held. The valid range is `0` to `60000`. A value of `60000` represents infinity.

Answer:
none

Example:
Command: `MPR2P500`

Moves the positioner connected to channel 2500 µm in positive direction. This position is not held.

2.3.5 MAA – Move to Angle Absolute

Description:
Instructs a positioner to turn to a specific angle via closed-loop control. This command is only executable by a positioner that has a sensor integrated. Additionally, the command is only executable if the addressed channel is configured to be of type rotary (see "SST – Set Positioner Type (formerly Set Sensor Type)" command). A linear channel will return an error. See also "MPA – Move to Position Absolute" command.

While executing the command the positioner will have a movement status of T (targeting). If a hold time was specified the movement status will change to H (holding) after having reached the target position. After the hold time has passed, the movement status will be S (stopped).

Command String Format:
MAA<channel>A<angle>R<revolution>H<holdTime>

Parameters:

- <channel>: Zero-based channel index. Valid indices are 0, 1 and 2.
- <angle>: Absolute angle to turn to in m°.
- <revolution>: Reserved for future use. Omit or set to 0.
- <holdTime>: Time in ms that the target position is to be held. The valid range is 0 to 60000. A value of 60000 represents infinity.

Answer:
none

Example:
Command: MAA0A500

Turns the positioner connected to channel 0 to the angle 0.5°. The angle is not actively held.

2.3.6 MAR – Move to Angle Relative

Description:
Instructs a positioner to move to an angle via closed-loop control, which is given in relation to the current angle. This command is only executable by a positioner that has a sensor integrated. Additionally, the command is only executable if the addressed channel is configured to be of type rotary (see "SST – Set Positioner Type (formerly Set Sensor Type)" command). A linear channel will return an error. See also "MPA – Move to Position Absolute" command.

While executing the command the positioner will have a movement status of T (targeting). If a hold time was specified the movement status will change to H (holding) after having reached the target position. After the hold time has passed, the movement status will be S (stopped).

Command String Format:
MAR<channel>A<angle>R<revolution>H<holdTime>

Parameters:

- <channel>: Zero-based channel index. Valid indices are 0, 1 and 2.
- <angle>: Relative angle to move to in m°.
- <revolution>: Reserved for future use. Omit or set to 0.
- <holdTime>: Time in ms that the target position is to be held. The valid range is 0 to 60000. A value of 60000 represents infinity.

Answer:
none

Example:
Command: MAR1A-1500

Moves the positioner connected to channel 1 1.5° in negative direction. This position is not held.

2.3.7 MTR – Move To Reference

Description:
For positioners that are equipped with sensor feedback, this command may be used to move the positioner to a known physical position of the positioner. Some positioner types are equipped with a physical reference mark, others are referenced via a mechanical end stop (see section 4.3 "Positioner Types"). For all types you must issue "CS – Calibrate Sensor" before the positioner can be properly referenced. In which direction the movement starts is implicitly controlled by the "safe direction" of the positioner (see "SSD – Set Safe Direction").

If the auto zero flag is set, the current position resp. angle is set to zero after the reference position has been reached. Otherwise the position is set according to the information stored in non-volatile memory of the last referencing or "SSC – Set Scale" command. See section 2.6 "Defining Positions" for more information.

As a safety precaution, make sure that the positioner has enough room to move without damaging any other equipment.

While executing the command the positioner will have a movement status of R (referencing). If a hold time was specified the movement status will change to H (holding) after having reached the reference position. After the hold time has passed, the movement status will be S (stopped).

If this command was successful, the physical position of the positioner becomes known. See "GPPK – Get Physical Position Known".

Command String Format:
MTR<channel>H<holdTime>Z<autoZero>

Parameters:

- <channel>: Zero-based channel index. Valid indices are 0, 1 and 2.
- <holdTime>: Time in ms that the target position is to be held for. The valid range is 0 to 60000. A value of 60000 represents infinity.
- <autoZero>: Must be either 0 (no auto-zero) or 1 (auto-zero). The latter will reset the current position to zero upon reaching the reference position.

Answer:
none

Example:
Command: MTR1H0Z1

Moves the positioner connected to channel 1 to the reference position. Sets the position to zero thereafter.

NOTICE
This feature is not available with all sensors. Refer to section 4.3 "Positioner Types" for further information.

2.3.8 MES – Move to End-Stop

Description:
Instructs a positioner to move to an end stop in a given direction. This command is only executable by a positioner that has a sensor integrated.

While executing the command the positioner has a movement status code T (targeting). After the positioner has reached the end stop, the movement status code changes to S (stopped).

Command String Format:
MES<channel>D<direction>

Parameters:

- <channel>: Zero-based channel index. Valid indices are 0, 1 and 2.
- <direction>: Direction in which the positioner shall move. The symbol can be one of the following:
  U: for the up / positive direction
  D: for the down / negative direction

Answer:
none

Example:
Command: MES1DU

Moves the positioner connected to channel 1 to the end stop in positive direction.

2.3.9 CS – Calibrate Sensor

Description:
This command may be used to increase the accuracy of the position calculation. It is only executable by a positioner that has an integrated sensor.

This command should be called once for each channel if the mechanical setup changes (different positioners connected to different channels). The calibration data will be saved to non-volatile memory. If the mechanical setup is unchanged, it is not necessary to call this function on each initialization, but newly connected positioners have to be calibrated in order to ensure proper operation.

The calibration procedure changes depending on the sensor (absolute sensor or sensor with reference mark) and reference type (end stop or mark).

The calibration consists of up to two phases. During the first phase the positioner will perform a movement of up to several millimeters in one direction and then in the other direction. The user must ensure, that the positioner is not near a mechanical end stop during this first phase. Otherwise the calibration might fail and lead to unexpected behavior when executing closed-loop commands.

During the second phase positioners that are referenced via a mechanical end stop section 4.3 “Positioner Types” are moved to the end stop. The direction is implicitly given by the “safe direction” which is configured via the “SSD – Set Safe Direction” command. Once the end stop has been calibrated during this second phase you may use the “MTR – Move To Reference” command to accurately move to the reference position.

The calibration takes a few seconds to complete. During this time the positioner the movement status code C (calibrating). After the calibration has finished, it will change to S (stopped).

Command String Format:
CS<channel>

Parameters:

- <channel>: Zero-based channel index. Valid indices are 0, 1 and 2.

Answer:
none

Example:
Command: CS0

Calibrates the sensor on channel 0.

2.3.10 S – Stop

Description:
This command stops any ongoing activity of the positioner on the selected channel. If the channel index is set to `99` the command will be executed for all channels at once.

Command String Format:
`S`<channel>

Parameters:

- <channel>: Zero-based channel index. If the index is set to `99` the command will be executed for all channels. Valid indices are `0`, `1`, `2` and `99`.

Answer:
none

Example:
Command: `S99`

Stops all positioners.

2.4 Positioner Feedback Commands

2.4.1 GP – Get Position

Description:
Returns the current position of a linear positioner. This command is only executable by a positioner that has an integrated sensor.

Command String Format:
`GP`<channel>

Parameters:

- <channel>: Zero-based channel index. Valid indices are `0`, `1` and `2`.

Answer:
`P`<channel>`P`<position>

- <channel>: Zero-based channel index.
- <position>: The current position of the positioner in µm.

Example:
Command: `GP0`
Answer: `P0P-13.5`

Requests the current position of channel 0. The answer indicates that the positioner is currently at position −13.5 µm.

2.4.2 GA – Get Angle

Description:
Returns the current angle of a rotary positioner. This command is only executable by a positioner that has an integrated sensor.

Command String Format:
GA<channel>

Parameters:

- <channel>: Zero-based channel index. Valid indices are 0, 1 and 2.

Answer:
A<channel>A<angle>R<revolution>

- <channel>: Zero-based channel index.
- <angle>: The current angle of the positioner in m°.
- <revolution>: Reserved for future use.

Example:
Command: GA0
Answer: A0A-187.2R0

Requests the current angle of channel 0.The answer indicates that the positioner is currently at angle −187.2 m°.

2.4.3 M – Movement status query

Description:
This command may be sent to retrieve the current movement status of a channel.

Command String Format:
M<channel>

Parameters:

- <channel>: Zero-based channel. If the index is set to 99 the command will be executed for all channels. Valid indices are 0, 1, 2 and 99.

Answer:
M<channel><status>

- <channel>: Zero-based channel index.
- <status>: A letter that indicates the status of the channel. See section 4.1 “Channel Status Codes” for a list of status codes.

If no error occurs the answer string contains the channel and a letter indicating the status of the channel. If the channel index was set to 99, then three separate answer strings are generated, one for each channel.

Example:
Command: M0
Answer: M0S

Requests the movement status of channel 0. The answer indicates that positioner 0 is currently stopped.

2.4.4 GPPK – Get Physical Position Known

Description:
Returns whether the positioner “knows” its physical position. After a power-up the physical position is unknown and the current position is implicitly assumed to be the zero position. After the reference position was reached by calling “MTR – Move To Reference” the physical position becomes known.

This function can be useful if the software application restarts and connects to a system that has stayed online. If the physical position is already known, traveling to the reference mark again may be omitted.

Command String Format:
GPPK<channel>

Parameters:

- <channel>: Zero-based channel. Valid indices are 0, 1 and 2.

Answer:
PPK<channel>K<known>

- <channel>: Zero-based channel index.
- <known>: Status. Will be either 0 (unknown) or 1 (known).

Example:
Command: GPPK0
Answer: PPK0K1

Requests the physical position known status of channel 0. The answer indicates that positioner 0 currently “knows” its position.

2.5 Miscellaneous Commands

2.5.1 K – Keep-alive

Description:
A keep-alive timer is provided to stop all positioners in case no command or keep-alive command is sent in predefined intervals.

The K-command has two functions:

1. Keep-alive
   If no parameter is given, it causes the internal timeout counter to be reset. If a timeout occurs, all positioners are immediately stopped.
2. Set keep-alive timeout
   If a number is given as a parameter, then the timeout is set to this value given in ms. A value of 0 disables the timeout functionality.

If the timeout is enabled all (successful) commands implicitly reset the timeout counter.

It is highly recommended to use this feature when using unbounded moves.

Command String Format:
Keep-alive:

K

Set Keep-alive timeout:

K<timeout>

Parameters:

- <timeout>: Timeout delay in ms. The valid range for the timeout delay is 100 to 60000. A value of 0 (default) is also valid and disables the timeout functionality.

Answer:
none

Example:
Command: K5000

Sets the timeout delay to 5 s. Subsequent movement commands are halted after 5 s if no new command is received.

Command: K

Keeps the movement alive. Having no effect otherwise.

2.5.2 E – Error status

Description:
An internal register is provided which keeps the error code of the last command.

The E-command has two functions.

1. Get error status
   If no parameter is given, then an answer string contains the code of the last error that occurred. After this, the internal error code register is reset to zero (= no error).
2. Set auto error report mode
   A single digit may be given as a parameter to configure the device to auto error report mode or non auto error report mode. The default configuration is non auto error report mode. See section 4.2 “Error Codes” for more information.

Command String Format:
Get error status:
E

Set auto error report mode:
E<mode>

Parameters:

- <mode>: A single digit will configure the device to
  0: non auto error report mode,
  1: auto error report mode.

Answer:
In case that the E-command has been used without parameters the error code of the previous command is returned:

E<code>

In case that the E-command has been used with parameters the returned error codes indicates whether setting the auto error report mode was successful or not.

E<code>

Parameters:

- <code>: Error code. See the section 4.2 “Error Codes” for a list of error codes and their meanings.

Example:
Command: E
Answer: E17

Requests the error code of the last command. The answer (error code 17) shows that an invalid parameter was given on the last command.

Command: E1
Answer: E0

Sets the current mode to be the auto error report mode. The answer shows that this mode setting has been successful.

2.5.3 CB – Configure RS232 Baud rate

Description:
Sets the baud rate of the RS232 interface to the specified value. The setting is stored to EEPROM (non-volatile memory) and loaded on future power-ups. Note that a change to this setting does not take effect until the next system reset. After configuring the interface, either do a power-down/power-up cycle or send a reset command to make the changes take effect.

Command String Format:
CB<baudrate>

Parameters:

- <baudrate>: Baud rate of the RS232 interface. The valid range is 9600 to 500000.

Answer:
The answer string contains the baud rate that was effectively configured and reflects the closest value that the internal baud rate generator is able to produce. For standard baud rates the error is small enough for a stable communication.

CB<baudrate>

- <baudrate>: Baud rate of the RS232 interface that was effectively configured.

Example:
Command: CB57600
Answer: CB57142

Configures the RS232 baud rate to 57 600Bd. The baud rate will be set to 57 142Bd (0.8 % error) after the next reset.

2.6 Defining Positions

Since position calculation is done on an incremental basis, the SCU controller has no way of knowing the physical position of a positioner after a system power-up. It simply assumes its starting position as the zero position.

However, in many applications it is convenient to define a certain physical position as the zero position. The "SZ – Set Zero position" command may be used for this purpose. It defines the current position to be zero. Furthermore if a system "knows" its physical position after being referenced the scale and thus the retrieved position can be shifted using the "SSC – Set Scale" command.

2.6.1 Reference Marks

The physical position of a positioner has to be determined by some external method and then configured to the system. Moreover, this procedure must be done on every system power-up.

To overcome this inconvenience the "MTR – Move To Reference" command may be used to move a positioner to a known physical position in an automated fashion. Depending on the product option of your positioner it may be equipped with a single reference mark or with multiple reference marks. Some positioners do not have a physical reference mark, but are rather referenced via a mechanical end stop. The different search algorithms are outlined in the following:

- Single reference Mark: In this case the reference mark (which is usually located near the middle of the travel range) is used to determine the physical position. The positioner starts to move in the configured safe direction (see "SSD – Set Safe Direction"). As soon as the reference mark has been detected the positioner stops and the search is successful. Should the positioner detect an end stop, the search direction is reversed and the search continues. If a second end stop is detected before the reference mark is found the search will abort unsuccessfully (When activating the auto error report mode, an error will be generated). For these types of positioners (linear but also rotary) the physical measuring scale is defined such that the zero position is located on the reference mark.
- Distance coded Reference Marks: In this case the distance between any two neighboring reference marks is measured in order to determine the physical position. The positioner starts to move in the configured safe direction (see "SSD – Set Safe Direction" command). When the first reference mark has been detected, the current position is stored and the search continues in the same direction for a second mark. When the second mark has been detected, the positioner stops and the search is successful. The distance between the two reference marks is calculated to determine the physical position. Should the positioner detect an end stop, the search direction is reversed and the process is repeated. If a second end stop is detected before two reference marks have been found the search will abort unsuccessfully (When activating the auto error report mode, an error will be generated). For these types of positioners the physical measuring scale is defined such that half the length of the positioner is near the middle position of the slider (The physical zero position is unreachable, since it lies outside the travel range).
- End Stops: In this case a mechanical end stop is used as a known physical position. The positioner will move in the safe direction (see "SSD – Set Safe Direction" command) until it detects an end stop. The sensor signals are then used to align the position to the reference

position with high repeat accuracy. For these types of positioners the physical measuring scale is defined such that the zero position lies near the mechanical end stop that is used for referencing. Note that the scale therefore depends on the Safe Direction setting.

Note that the positioner must be calibrated via the "CS – Calibrate Sensor" command to ensure a proper reference detection. When the command has completed successfully the system knows the physical position of the positioner (see "GPPK – Get Physical Position Known" command).

2.6.2 Shifting the Measuring Scale

The physical measuring scale is fix for each positioner and cannot be changed. However, the SCU controller uses a logical measuring scale when calculating positions to result in position values returned by "GP – Get Position" (or "GA – Get Angle"). The logical measuring scale may be shifted or inverted by the user so that the controller returns a desired value at a certain physical position.

The relation between the physical and the logical scale is defined by two parameters. The offset value, which represents the shift and the inversion value, which inverts the count direction of the logical scale relative to the physical scale. The default value of the offset and the inversion is zero which makes the physical and the logical scale identical.

There are two methods to modify the offset value:

- Sending a "SZ – Set Zero position" command sets the offset implicitly by shifting the logical scale so that the current position equals the zero position.
- Sending a "SSC – Set Scale" command sets the offset explicitly and the current position will have a value that reflects the new scale shift and direction.

The offset and inversion value is stored in non-volatile memory. Once it is configured you only need to send a "MTR – Move To Reference" command to restore your settings on future power-ups.

NOTICE

The behavior of the system when sending a "SZ – Set Zero position" or "SSC – Set Scale" command differs slightly depending on whether the physical position is known or not (see "GPPK – Get Physical Position Known"). When the physical position is unknown sending a "SZ – Set Zero position" will not update the offset value in the non-volatile memory. Likewise, sending a "SSC – Set Scale" will have no immediate effect on the values returned by "GP – Get Position". The following table summarizes the behavior.

	physical position is known		physical position is unknown	
	"SSC – Set Scale"	"SZ – Set Zero position"	"SSC – Set Scale"	"SZ – Set Zero position"
offset value is written to non-volatile memory	yes	yes	yes	no
command has immediate effect on position values	yes	yes	no	yes

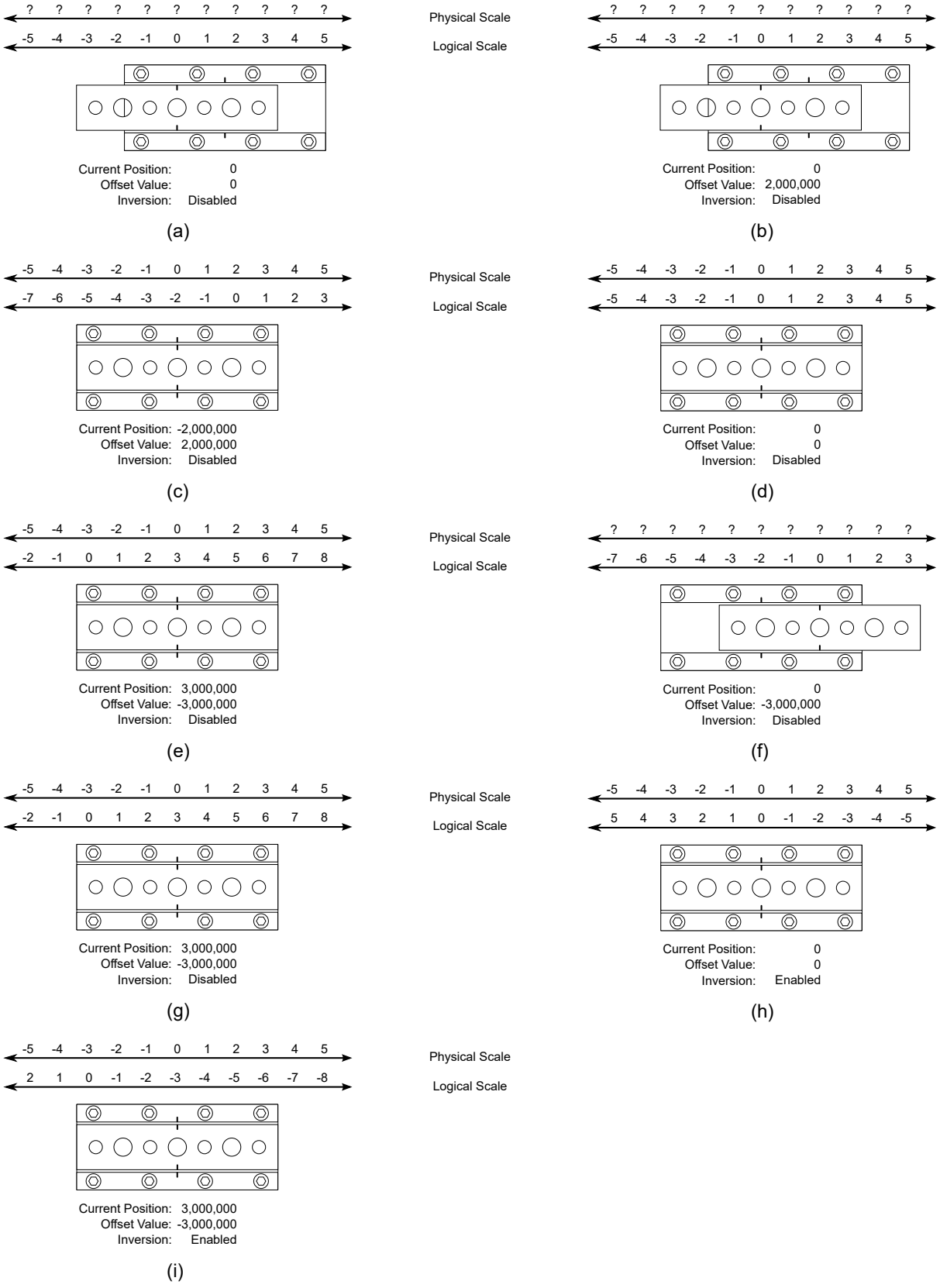


Figure 2.1: Scale shifting example

2.6.3 Example

To further demonstrate the behavior of the system in different situations the figure below shows an example of an SLC positioner with a single reference mark. The small markings indicate the location of the reference mark. When the markings overlap the positioner is on the mark. The code section below shows the corresponding command sequence.

1. (a) – The system is powered up. The physical position is unknown and the current position is assumed to be 0. The offset value is in the default setting.
2. `SSC0I0S2000000`
3. (b) – The offset value in the non-volatile memory has been set to +2mm. Since the physical position is unknown, the current position could not be updated implicitly.
4. `MTR0H0Z0`
5. (c) – The positioner has moved to the reference mark. The physical position is now known and the position value has been updated to reflect the configured offset between the physical and logical scale.
6. `SZ0`
7. (d) – The current position has been set to zero. Since the physical position is known, the offset value in the non-volatile memory was updated implicitly (in this case set to zero as we have not moved after the last step).
8. `SSC0I0S-3000000`
9. (e) – The offset value in the non-volatile memory has been set to -3mm. Since the physical position is known, the current position was updated implicitly.
10. (f) – The system was shut down, the positioner was moved externally to some random location and the system is then powered up again. The physical position is unknown and the positioner assumes its current position as 0 again.
11. `MTR0H0Z0`
12. (g) – The positioner has moved to the reference mark. The physical position is now known and the position value has been updated to reflect the configured offset between the physical and the logical scale.
13. `SSC0I1S0`
14. (h) – The offset and inversion value in the non-volatile memory have been set to 0mm and 1 (inverted). Thus the counting direction of the scale got inverted. Since the physical position is known, the current position was updated implicitly.
15. `SSC0I1S-3000000`
16. (i) – The offset value was changed and both values were stored inside the non-volatile memory. Since the physical position is known, the current position was updated implicitly.

3 *GENERAL BEHAVIOR*

Once the system is initialized after power up and the TX line has gone logic high it is able to receive commands at any time. The execution of previous commands does not affect the reception of new commands.

All channels may be driven at the same time. Each channel is independent of the others and has its own parameters. A movement command for a positioner that is already moving overwrites the previous command.

3.1 Restrictions

If a movement command changes the amplitude for a channel, the actual movement is delayed by several milliseconds. If the positioner was previously moving, it is stopped to adjust the amplitude and then the new movement is executed. This may result in a non-smooth speed change when overwriting previous movement commands. Therefore it is recommended to change the amplitude only infrequently. The set amplitude command may be useful to change the amplitude when a positioner is currently stopped. All subsequent movement commands that omit the amplitude parameter (or supply the same value) are driven with the new amplitude.

Driving multiple channels at the same time with slightly different frequencies may result in unstable frequencies due to interference effects.

3.2 Vertical Axis Configuration

In systems that use a positioner as a vertical axis you may experience inaccurate closed-loop positioning if the positioner has to lift fairly heavy weights. When moving upwards the positioner has to overcome gravity and therefore moves slower. Conversely when moving downwards gravity potentially pushes the positioner beyond the target position. For these situations a special mode of operation can optimize the behavior.

The SPA command may be used to configure a channel for vertical positioning. In this mode closed-loop commands are processed slightly different. Two parameters must be specified to configure the mode: a forward amplitude and a backward amplitude. These values define which step amplitudes are used when moving in either direction. Which direction points upwards and which downwards depends on which way the positioner is mounted. Generally you will want to use a full amplitude (`1000` = 100 V) when moving upwards and a smaller amplitude when moving downwards. The exact value of the small amplitude may vary depending on the weight to be lifted. A larger weight usually requires smaller amplitudes in down direction. A good rule of thumb is to choose a value where the up and down movements of a fixed travel distance take about the same time to execute.

As an example you would call `SPA0A1F1000B500` to use an amplitude of 50 Volts for the down direction. If the positioner is mounted the other way around, call `SPA0A1F500B1000` instead.

Note that the vertical mode is not necessarily needed for all z-axes. If the weight that is to be lifted is small compared to the capability of the positioner, then it is generally sufficient to leave the default horizontal mode unchanged.

Also note that the alignment setting does not affect open-loop control of the positioner, since you already have direct control of the used step amplitudes.

4 *APPENDIX*

4.1 Channel Status Codes

Table 4.1 lists the movement status codes of positioners that are returned by the "M – Movement status query" command.

Table 4.1 – Movement status codes

Code	Meaning	Description
S	Stopped	The positioner is currently not performing active movement (see "S – Stop" command).
A	Amplitude setting	The amplitude for an open-loop movement command ("U – move Up" and "D – move Down") is being set.
M	Moving	The positioner is performing a (open-loop) movement (see movement commands "U – move Up" and "D – move Down").
T	Targeting	The positioner is executing a closed-loop movement towards a given target position (see "MPA – Move to Position Absolute", "MPR – Move to Position Relative", "MES – Move to End-Stop" commands).
H	Holding	The positioner is holding its current target position (see closed-loop commands: "MPA – Move to Position Absolute" and "MPR – Move to Position Relative").
C	Calibrating	The positioner is busy calibrating its sensor (see "CS – Calibrate Sensor" command).
R	Referencing	The positioner is moving to a reference position (see "MTR – Move To Reference" command).

4.2 Error Codes

An error answer string has the general format E<code>. Table 4.2 lists the error codes and their meanings.

Table 4.2 – Error Codes

Code	Meaning	Description
0	No error	Indicates that no error occurred and therefore corresponds to an acknowledge.
1	Parse error	The command could not be processed due to a parse error.
2	Unknown command error	The command given is not known to the system.
3	Invalid channel error	The channel index given is invalid and the command cannot be processed.
4	Invalid mode error	The parameter that defines the mode for automatic error reporting is not valid, such that the mode change cannot be processed.
13	Syntax error	The command could not be processed due to a syntax error.
15	Overflow error	A number value given was too large to be processed.
17	Invalid parameter error	A parameter that was given with the command was invalid.
18	Missing parameter error	A parameter was omitted where it was required.
19	No Sensor Present Error	This error occurs if a command was given that requires sensor feedback, but the addressed positioner has none attached.
20	Wrong Positioner Type Error	Some commands are only executable for certain positioner types. For example, issuing an MAA command to a linear positioner leads to this error.
21	End Stop Reached Error	This error occurs if the last closed-loop movement command of a channel was aborted due to end stop detection. Reporting of this error can be enabled or disabled for each channel.
22	Targeting Timeout Error	This error occurs if the last closed-loop movement command of a channel was aborted due to not terminating after a fixed timeout. Reporting of this error can be enabled or disabled for each channel.
23	HV Range Error	If the voltage operating the positioners is out of range, all movement will be stopped and only a power cycle sets the device back into operating mode. This may be caused e.g. by defective cabling.

Continued on next page

Table 4.2 – *Continued from previous page*

Code	Meaning	Description
24	Temperature Overheat Error	If a system becomes too hot for any reason, all movement is stopped. Furthermore the operating mode is paused and automatically resumed when the system has cooled down.
25	Calibration Failed Error	This error occurs if the last calibration command of a channel was aborted due to a mechanically blocked positioner. Note that this might as well be the case when calibrating near an end stop.
26	Referencing Failed Error	This error occurs if the last move-to-reference command of a channel was aborted. If this happens, the system has not been able to detect the desired mark(s). A possible reason might be an incorrect positioner type setting.
27	Not Processable Error	The given command cannot be processed (at the moment). This might occur when issuing a SZ – Set Zero command while the positioner is still moving.

4.3 Positioner Types

The following table lists the currently available positioner types that may be configured with the "SST – Set Positioner Type (formerly Set Sensor Type)" command.

NOTICE
Depending on hardware and firmware configuration, some positioner types may not be available and result in an invalid parameter error.

The reference type indicates the way the positioner is referenced with the "MTR – Move To Reference" command. Positioners with 'mark' are referenced via a physical reference mark that is typically located near the middle of the complete travel range. Positioners with 'end stop' are referenced via a mechanical end stop. Positioners with 'none' cannot be referenced.

NOTICE
On SCU-3CLM, SCU-3CM and SCU-1DM hardware, only positioner types with reference type "end stop" or "none" may be selected.

Table 4.3 – Positioner Types

Symbol	Type Code	Positioner Series	Comment	Reference Type
M	1	SLCxxxxm, SLxxxxm	linear positioners with microsensor	end stop
GC	4	SR1910m	rotary positioner with microsensor, no reference mark, end stops	end stop
GD	5	SGO60.5m	goniometer with 60.5 mm radius	end stop
GE	6	SGO77.5m	goniometer with 77.5 mm radius	end stop
RA	7	SFWxxxxam	rotary positioner with absolute sensory	none
GF	8	SR1209m	rotary positioner with microsensor	end stop
RB	9	SR1910m	rotary positioner with microsensor, no reference mark, no end stops (unlimited rotation)	none
SR36M	10	SR3610m, SR4011m	rotary positioner	none
SR36ME	11	SR3610m, SR4011m	rotary positioner with end stops	end stop

Continued on next page

Table 4.3 – *Continued from previous page*

Symbol	Type Code	Positioner Series	Comment	Reference Type
SR50M	12	SR5018m	rotary positioner	none
SR50ME	13	SR5018m	rotary positioner with end stops	end stop
MM50	14		linear positioner with magnetic encoder	end stop
G935M	15	SGO93.5me	goniometer with 93.5 mm radius	end stop
MD	16	SLCxxxxdme	linear positioner with microsensor and double piezo	end stop
TT254	17	STT25.4m	no longer supported	none
LC	18	SLCxxxxlc	linear positioner with improved micro sensor	mark[1]
LR	19	SRxxxxl	rotary positioner with improved micro sensor	mark
LCD	20	SLCxxxxdlc	like LC, but with large actuator	mark[1]
L	21	SLCxxxxl	linear positioner with improved micro sensor	mark
LD	22	SLCxxxxdl	like L, but with large actuator	mark
LE	23	SLCxxxxl	linear positioner with improved micro sensor	end stop
LED	24	SLCxxxxdl	like LE, but with large actuator	end stop
SL...S1I1E1	25	SLxxxxS1I1E1	Linear positioner, inductive sensor with 293 nm resolution	end stop
SL...D1I1E1	26	SLxxxxD1I1E1	like SL...S1I1E1, but with large actuator	end stop
SL...S1I2E2	27	SLxxxxS1I2E2	like SL...S1I1E1, but with 73 nm resolution	end stop
SL...D1I2E2	28	SLxxxxD1I2E2	like SL...S1I2E2, but with large actuator	end stop
ST...S1I1E2	29	STxxxxS1I1E2	25.4 mm mirror tip-tilt axis, inductive sensor with 73 µ° resolution	end stop
SG...D1L1S	30	SGxxxxD1L1S	Goniometer, 60.5 mm radius, double piezo element	mark
SG...D1L1E	31	SGxxxxD1L1E	Goniometer, 60.5 mm radius, double piezo element	end stop
SG...D1L2S	32	SGxxxxD1L2S	Goniometer, 77.5 mm radius, double piezo element	mark

Continued on next page

Table 4.3 – *Continued from previous page*

Symbol	Type Code	Positioner Series	Comment	Reference Type
SG...D1L2E	33	SGxxxxD1L2E	Goniometer, 77.5 mm radius, double piezo element	end stop
SG...D1M1E	34	SGxxxxD1M1E	Goniometer, 60.5 mm radius, double piezo element	end stop
SG...D1M2E	35	SGxxxxD1M2E	Goniometer, 77.5 mm radius, double piezo element	end stop
SI...S1L1S	36	SIxxxxS1L1S	Iris-Diaphragm, 21.42 mm radius, single piezo element	mark

[1] These positioners are equipped with multiple reference marks. The positioner will only have to move a few millimeters to know its physical position. See also section 2.6 “Defining Positions”.

4.4 Property Keys

4.4.1 System Properties

Table 4.4 – System Properties

Code	Name	Description
1	Internal System Temperature	The current internal temperature, given in °C.
2	Internal 100 V Voltage	The current internal 100 V voltage, given in 0.1 V.
3	Internal Hardware Version Code	The internal version code of the underlying hardware. Note that this code does not necessarily correspond to the version printed on the PCB.

4.4.2 Channel Properties

Table 4.5 – Channel Properties

Code	Name	Description
3	Target Reached Threshold	Defines a range around the target position (so the total width of the range is double the value configured). Whenever a closed-loop movement command is given and the current position is within this range, the target position is considered as reached and the movement command completes. The value is given in nm. The default value depends on the configured positioner type. This a volatile property.
5	Proportional Gain Base	Defines the proportional gain base value used for the selected channels closed-loop movements. The valid range of values is from `1` up to `127`. To allow finer grained proportional gain values, the gain divisor (see below) can be used to divide the selected base value. The default value depends on the configured positioner type. This is a volatile property.

Continued on next page

Table 4.5 – *Continued from previous page*

Code	Name	Description
6	Proportional Gain Divisor	Defines the proportional gain divisor value. The proportional base value will therefore be divided by powers of 2 for each given integer. The valid integer range is from `0` up to `10`, resulting in divisors from 1 to 1024. The corresponding formula is: proportional_gain = $\frac{\text{gain\_base}}{2^{\text{gain\_divisor}}}$ The default value depends on the configured positioner type. This is a volatile property.
21	Advanced Stepping Mode Enabled	Enables an advanced stepping algorithm for positioning. In this mode, the position will be further optimized after reaching the Target Reached Threshold by conventional means. This will improve the accuracy for most positioners, at the cost of increased positioning time.

Sales partner / Contacts

Headquarters

SmarAct GmbH

Schuette-Lanz-Strasse 9
26135 Oldenburg
Germany

T: +49 441 – 800 87 90
Email: info@smaract.com
www.smaract.com

France

SmarAct GmbH

Schuette-Lanz-Strasse 9
26135 Oldenburg
Germany

T: +49 441 – 80 08 79 956
Email: nicoul@smaract.com
www.smaract.com

Israel

Trico Israel Ltd.

P.O.Box 6172
46150 Herzeliya
Israel

T: +972 9 – 950 60 74

www.trico.co.il

Japan

Physix Technology Inc.

Ichikawa-Business-Plaza
4-2-5 Minami-yawata,
Ichikawa-shi
272-0023 Chiba
Japan
T/F: +81 47 – 370 86 00
Email: info@physix-tech.com
www.physix-tech.com

South Korea

SEUM Tronics

Room 502, 534 Seobusaet-gil
Geumcheon-Gu
08505 Seoul
Korea

T: +82 2 868 – 10 02
Email: hslee@seumtronics.com
www.seumtronics.com

USA

SmarAct Inc.

2140 Shattuck Ave., Suite 1103
94704 Berkeley, CA
United States of America

T: +1 415 – 766 90 06
Email: info@smaract.com
www.smaract.com