KERN & Sohn GmbH
Ziegelei 1
D-72336 Balingen
E-Mail: info@kern-sohn.com
Tel: +49-[0]7433- 9933-0
Fax: +49-[0]7433-9933-149
Internet: www.kern-sohn.com

Reference manual KERN Communications Protocol (KCP)

KERN KCP

KCP versions: 1.1.1
Manual version 1.3.6 (2018-09)
GB

KCP-ZB-e-18136

GB

KERN KCP

Reference manual
KERN Communications Protocol (KCP)

Table of contents

1 BRIEF OUTLINE 4

1.1 Default interface communication parameters 4

1.2 Basic command and response format 4

1.3 Language conventions 4

1.4 Overview of basic commands 5

2 GENERAL 6

2.1 KCP Version 6

2.2 KCP Command Levels 6

2.3 Conventions in this manual 6

2.4 Default communication parameters 6
2.4.1 RS-232 / RS-485 6

2.5 Protocol structure 7
2.5.1 Encoding 7
2.5.2 Case sensitiveness 7
2.5.3 Commands 7
2.5.4 Responses 7
2.5.5 Examples 7

2.6 Command queue and timing 8
2.6.1 Command queue / sequence 8
2.6.2 Timeouts 8

2.7 Units 8

2.8 Message codes / Error codes 8

3 KCP COMMANDS – CATEGORY “DEVICE” (LEVEL 0) 9

4 KCP COMMANDS – CATEGORY “DEVICE DISPLAY” (LEVEL 1) 19

5 KCP COMMANDS – CATEGORY “WEIGHING BASIC” (LEVEL 0) 27

6 KCP COMMANDS – CATEGORY “WEIGHING ADVANCED” (LEVEL 1) 37

7 KCP COMMANDS – CATEGORY “WEIGHING ADJUSTMENT” (LEVEL 2) 46

8 KCP COMMANDS – CATEGORY “SENSORS” (LEVEL 2) 50

9 KCP COMMANDS – CATEGORY “EXTERNAL TERMINAL” (ETL) 55

10 KCP COMMANDS – CATEGORY “AXIS MOVEMENT” 57

11 KCP COMMANDS – LEVEL 2 (MODEL-SPECIFIC FEATURES) 66

12 KCP COMMANDS – CATEGORY “SERVICE” 71

13 KCP COMMAND INDEX 72

14 INTERN - TODOS .... FEHLER! TEXTMARKE NICHT DEFINIERT.

14.1 KDP .... Fehler! Textmarke nicht definiert.
14.1.1 Temperaturkorrektur einmessen "JT" .... **Fehler! Textmarke nicht definiert.**
14.1.2 Lesen/Schreiben der Kalibrierdaten: "JD" .... **Fehler! Textmarke nicht definiert.**
14.1.3 Kriechkorrektur einmessen "JM" .... **Fehler! Textmarke nicht definiert.**

14.2 Shimadzu Befehle aufnehmen? .... Fehler! Textmarke nicht definiert.

1 Brief outline

The KERN Communications Protocol (KCP) is a standardized interface command set for KERN balances and other instruments, which allows retrieving and controlling all relevant functions and functions of the device. KERN instruments featuring KCP are thus easily integrated with computers, industrial controllers and other digital systems.

This section gives an overview over the general command and response structure and lists the few basic commands required to handle the vast majority of applications.

1.1 Default interface communication parameters

By default, each KCP device comes preset to the following communication parameters. The applicable parameters depend on the type of communication interface:

Interfaces	**Parameters**			
RS-232 / RS-485 / Bluetooth SPP	Baud rate:	9600 baud/s	Data bits:	8 bits
	Parity:	none	Stop bits:	1 bit

1.2 Basic command and response format

KCP is based on simple ASCII-encoded text commands and responses. Every interaction consists of a command, possibly with arguments separated by spaces (symbol ␣) and terminated by Windows-style newline characters (*<CR><LF>)*:

Command		**Arguments**						**Terminator**	
<cmd>	␣	*<arg1>*	␣	*<arg2>*	␣	*<arg3>*	…	*<CR>*	*<LF>*

Correctly formatted commands are answered with a response containing the requested data including or following a confirmation of the following form:

Response		**Status**		**Data**	**Terminator**	
<cmd>	␣	A = accepted / acknowledge L = logical error / invalid parameter I = internal / technical error	␣	*command specific*	*<CR>*	*<LF>*

ES		Erroneous syntax or unknown command	*<CR>*	*<LF>*

Example: Command **"**Set indication unit to grams (g)" with response "accepted"

Command:	**U**	␣	**g**	*<CR>*	*<LF>*	➔	**Response:**	**U**	␣	**A**	*<CR>*	*<LF>*
dec:	85	32	103	13	10		dec:	85	32	65	13	10
hex:	55	20	67	0D	0A		hex:	55	20	41	0D	0A

Example: Command **"**Set indication unit to invalid unit" with response "logical error"

Command:	**U**	␣	**X**	*<CR>*	*<LF>*	➔	**Response:**	**U**	␣	**L**	*<CR>*	*<LF>*
dec:	85	32	88	13	10		dec:	85	32	76	13	10
hex:	55	20	58	0D	0A		hex:	55	20	4C	0D	0A

Example: Invalid command

Command:	**U**	␣	**g**	*<CR>*	*<LF>*	➔	**Response:**	**ES**	*<CR>*	*<LF>*
dec:	85	32	103	13	10		dec:	69 83	13	10
hex:	55	20	67	0D	0A		hex:	45 53	0D	0A

1.3 Language conventions

Throughout this manual, the following conventions are used for command and response syntax:

␣	Space symbol (dec 32, hex 20)
↓	Commands sent to the balance / measurement device.
↑	Responses of the balance / measurement device

1.4 Overview of basic commands

	Request stable indication (weighing or measured value) in the host unit (by default the current indication unit). Waits until indication fulfills the “stable” condition or until configured timeout is reached.	
↓	`S`	
↑	`S␣S␣␣␣␣␣100.00␣g`	Indication value is right aligned, 10 characters. Decimal sign is a point. The minus sign immediately precedes the numerical value – without leading zero. On multi-range devices, hidden trailing decimals are shown as spaces. Status “S” = current indication is stable Status “D” = current indication is unstable / dynamic
	`S␣S␣␣␣␣-100.00␣g`	
	`S␣S␣␣␣1152.05␣kg`	
	`S␣I`	In menu, currently executing another command or timeout reached.
	`S␣+` *or* `S␣-`	Overload or underload

	Request immediate indication in the host unit (by default the current indication unit) Immediately sends the current indication without waiting for stable conditions.	
↓	`SI`	
↑	`SI␣S␣␣␣␣␣100.00␣g`	*see description of command “S”*
	`SI␣D␣␣␣␣␣␣99.98␣g`	
	`SI␣I`	In menu, currently executing another command or timeout reached.
	`SI␣+` *or* `S␣-`	Overload or underload

	Zero indication	**Tare indication**	
↓	`Z`	`T`	
↑	`Z␣A`	`T␣A`	Zeroing/taring successful.
	`Z␣I`	`T␣I`	In menu, currently executing another command or timeout reached.
	`Z␣+` *or* `Z␣-`	`T␣+` *or* `T␣-`	Overload or underload; or zero range exceeded

	Query or set display and host unit			
↓	`U`	Query current display unit		
↑	`U␣A␣`*`<unit>`*	Current display unit is *`<unit>`*		
↓	`U␣`*`<unit>`*	Set current display and host unit.	**Units:**	g,kg,mg, lb,pcs,%, N,kN,TF,KLBF,…
↑	`U␣A`	Unit successfully set		
	`U␣I`	Invalid unit.		

	Set mode of indication (Peak or track mode)	
↓	`SIM`	Query current mode of indication
↑	`SIM␣A␣`*`<mode>`*	Current mode of indication is *`<mode>`*
↓	`SIM␣`*`<mode>`*	Set current mode of indication and reset the current peak value. *<mode>* is one of the following: • `T` = Track mode (indicate the current measurand) • `P` = Peak mode (only indicate the largest value +/-) • `P+` = Peak positive mode (only indicate the largest pos. value) • `P-` = Peak negative mode (only indicate the largest neg. value)
↑	`SIM␣A`	Mode successfully set, current peak value is zero.
	`SIM␣I`	Invalid *<mode>*

	Read measurement memory / reports Sends all available recorded data in a unspecified tabular form (separated by spaces)	
↓	`SMEM`	
↑	`SMEM␣A␣START`	Command understood, next lines will be the data in tabular form

	<header line>
	<data line 1>
	<data line 2>
	<data line 3>
		`Number Date       Time     Mode  Indication`
		`     1 2016-01-13 12:34:56 T       12.3456 N`
		`     2 2016-02-22 12:37:15 P+      12.3456 kN`
		`     3 2016-03-31 12:39:41 P-    -1234.56 N`

	`SMEM␣A␣END`	End of data

2 General

2.1 KCP Version

The KCP protocol is continuously being improved. With each new version, the KCP protocol version number is incremented. The number of the KCP version implemented in your particular device can be requested using the `I1` command.

Please make sure that you use the correct version of the KCP manual description (this document) for your device. If a command is only available in certain KCP versions, this will be mentioned in the section of the respective command.

2.2 KCP Command Levels

The KCP protocol commands are grouped in multiple levels. While Level 0 and Level 1 are available for all KCP devices, other levels may only be available with certain devices. Please refer to the individual chapter of each level for further details.

It is advised that you try to limit yourself to the lowest level of commands, that you can achieve your goals with. This allows you to connect a larger variety of KCP devices to your software without modifications.

2.3 Conventions in this manual

Throughout this manual, the following conventions are used for command and response syntax:

␣	Space symbol (dec 32, hex 20)
↓	Commands sent to the balance / measurement device.
↑	Responses of the balance / measurement device
«param»	Parameter name, the brackets (*« and »*) are not to be sent
[ ]	Optional parameter / expression

2.4 Default communication parameters

By default, each KCP device comes preset to certain communication parameters. The applicable parameters depend on the type of communication interface and are listed in the following paragraphs.

2.4.1 RS-232 / RS-485

Baud rate:	9600 baud/s
Data bits:	8 bits
Parity:	none
Stop bits:	1 bit

2.5 Protocol structure

KCP is based on simple ASCII-encoded text commands and responses.

2.5.1 Encoding

All characters and digits are encoded in ASCII – if not specified otherwise.

2.5.2 Case sensitiveness

The protocol is case sensitive. Commands and arguments should be written as described in this manual.

2.5.3 Commands

Every interaction consists of a command, possibly with arguments separated by spaces (symbol ␣) and terminated by Windows-style newline characters (*<CR><LF>)*:

Command		**Arguments**						**Terminator**	
<cmd>	␣	*<arg1>*	␣	*<arg2>*	␣	*<arg3>*	…	*<CR>*	*<LF>*

Commands should only be sent in uppercase letters.

2.5.4 Responses

Correctly formatted commands are answered with a response containing the requested data including or following a confirmation of the following form:

Response		**Status**		**Data**	**Terminator**	
<cmd>	␣	A = accepted / acknowledge L = logical error / invalid parameter I = internal / technical error	␣	*command specific*	*<CR>*	*<LF>*
ES		Erroneous syntax or unknown command			*<CR>*	*<LF>*

For commands that only execute actions on the device and do not return information required in your application, you can ignore the responses. However, to increase the reliability of your software, it is a good practice to read and evaluate the responses and act accordingly upon errors.

2.5.5 Examples

The following examples show some very basic interactions using the KCP protocol.

Example: Command **"**Set indication unit to grams (g)" with response "accepted"

Command:	**U**	␣	**g**	*<CR>*	*<LF>*		**Response:**	**U**	␣	**A**	*<CR>*	*<LF>*
dec:	85	32	103	13	10	➔	dec:	85	32	65	13	10
hex:	55	20	67	0D	0A		hex:	55	20	41	0D	0A

Example: Command **"**Set indication unit to invalid unit" with response "logical error"

Command:	**U**	␣	**X**	*<CR>*	*<LF>*		**Response:**	**U**	␣	**L**	*<CR>*	*<LF>*
dec:	85	32	88	13	10	➔	dec:	85	32	76	13	10
hex:	55	20	58	0D	0A		hex:	55	20	4C	0D	0A

Example: Invalid command

Command:	**U**	␣	**g**	*<CR>*	*<LF>*		**Response:**	**ES**	*<CR>*	*<LF>*
dec:	85	32	103	13	10	➔	dec:	69 83	13	10
hex:	55	20	67	0D	0A		hex:	45 53	0D	0A

2.6 Command queue and timing

2.6.1 Command queue / sequence

Ideally, the balance queues the data stream it receives and handles one command after the other. When this queue would overflow, the handshake mechanism of the underlying communication interface (e.g. RS-232 CTS/RTS or XON/XOFF handshake) prevents further data packages from the host computer. This allows the host computer to send whole scripts of commands to the balance.

Depending on the balance type (processor capabilities), this may not be possible. For maximum reliability, wait for the answer of a command before sending the next command – otherwise, for some balances, data could be corrupted or commands be missed.

2.6.2 Timeouts

There is no timeout between each character on a single command line (up to and including CR LF). An incomplete line will remain in the balance buffer without timeout until the line is completed. If the balance receive buffer is overflowing because there was no line end, the whole buffer is cleared.

This allows commands to be entered over a terminal software by a human user (one character at a time).

2.7 Units

All commands and responses in the KCP protocol use the following unit symbols:

Name	Symbol	Comment
Kilogram	kg	- no comment -
Ton	t	= 1000 kg
Gram	g	= 0.001 kg
Milligram	mg	= 0.000001 kg
Pound	lb	= 0.45359237 kg (lb. av. – Avoirdupois)
Pieces	pcs	requires piece weight
Percent	%	requires weight of 100%
Newton	N	Unit of force (where applicable)
Kilonewton	kN	= 1000 N
Ton-force	tf	= 9.80665 kN (weight of one ton due to standard gravity)
Pound-force	lbf / klbf	= 4.4482216152605 N (weight of one pount to standard gravity)

2.8 Message codes / Error codes

The following codes are used for errors and messages. In the protocol, the code number may be prefixed, e.g. “E1000”.

Code	Comment
0	no message
…	TODO

3 KCP commands – category "Device" (level 0)

`@`	Cancel
`I0`	List all implemented KCP commands
`I1`	Query KCP levels and KCP versions
`I2`	Query device information (type, capacity)
`I3`	Query device software version
`I4`	Query serial number
`I5`	Query software identification number
`IBIM`	Query/set balance external model number

English

@ – Cancel

Description

@ can be used to achieve the same effect as disconnecting and reconnecting the power supply, which empties the volatile memories. The purpose of this command is to initiate a command sequence.

Syntax

Command

`@`	Resets the device to the condition found after switching on, but without a zero setting being performed.

Responses

`I4␣A␣"`*«SNR»*`"`	Serial number is emitted; the device is ready for operation. (serial number may not available, then it is N/A)

Comments

- All commands awaiting responses are cancelled.
- If the device is on standby, it is switched on.
- The cancel command is always executed.
- The emitted serial number corresponds to the serial number of the terminal (if one is present), see [`I4`].

Examples

↓	`@`	Cancel
↑	`I4␣A␣"B021002593"`	Device is "reset", its serial number is B021002593

See also

➔	`I4` – Query serial number

`I0` – List all implemented KCP commands

Description

The `I0` command lists all commands implemented in the present software.

All level 0 commands are listed in alphabetical order before all commands of level 1 etc.

Syntax

Command

`I0`	Send list of all implemented KCP commands.

Responses

`I0˽B˽`*«Level»*`˽”`*«Command»*`”`
`I0˽B˽`*«Level»*`˽”`*«Command»*`”`
`I0˽B...`
`I0˽A˽`*«Level»*`˽”`*«Command»*`”`
	1st command implemented.
	2nd (next) command implemented.
	...
	Last command implemented.

`I0˽I`	Command understood but currently not executable (device is currently executing another command).

Parameters / Return values

Name	Type	Values	Meaning
`Level`	integer	Number of the KCP level where the command belongs to:	
		0	KCP level 0
		1	KCP level 1
		2	KCP level 2
		…	…
`Command`	string		KCP command

Comments

- If a terminal and a weigh module, weighing platform are being used, the command list of the terminal is output. If only a weigh module, platform is being used, the command list of the weigh module, platform is shown.
- If `I0` lists commands that cannot be found in the manual, these are reserved commands "for internal use" or "for future use", and should not be used or altered in any way.

Examples

↓	`I0`	Send list of commands
↑	`I0˽B˽0˽”I0”`	Level 0 command `I0` implemented
↑	`I0˽B...`	...
↑	`I0˽B˽0˽”@”`	Level 0 command `@` (cancel) implemented
↑	`I0˽B˽1˽”D”`	Level 0 command `D` implemented
↑	`I0˽B...`	...
↑	`I0˽A˽3˽”SM4”`	Level 3 command `SM4` implemented

See also

➔	`@` - cancel

I1, KCPV – Query KCP levels and KCP versions

Description

Query KCP level and versions.

Syntax

Command

`I1`	Query KCP level and KCP versions.

Responses

`I1␣A␣"«Level»"␣"«V0»"␣"«V1»"␣"«V2»"␣"«V3»"`	Current KCP level and KCP versions
`I1␣I`	Command understood but currently not executable

Parameters / Return values

Name	Type	Values	Meaning
`Level`	string	0	KCP level 0
		01	KCP level 0 and 1
		03	KCP level 0 and 3
		013	KCP level 0, 1 and 3
		…	…
`V0..V3`	string		KCP versions of the related level (0 to 3) (see cover page of this manual for the KCP version of these commands)

Examples

↓	`I1`	Query the current KCP level and version
↑	`I1␣A␣"123"␣"2.00"␣"2.20"␣"1.00"␣"1.50"`	Level 0-3 is implemented and the according version numbers are shown

KCPC – Query KCP categories

Description

Query KCP command categories.

Syntax

Command

`KCPC`	Query supported KCP command categories.

Responses

`KCPC˽B˽”«`*`CategoryName`*$_1$`»”`
`KCPC˽B˽”«`*`CategoryName`*$_2$`»”`
…
`KCPC˽A˽”«`*`CategoryName`*$_n$`»”`
	First supported KCP command category.
	Second supported KCP command category.
	…
	Last supported KCP command category.

`KCPC˽I`	Command understood but currently not executable

Parameters / Return values

Name	Type	Values	Meaning
`CategoryName`	string		KCP category internal name (see the separate chapters of this manual)

Examples

↓	`KCPC`	Query the current KCP level and version

↑	`KCPC˽B˽”Device”`
	`KCPC˽B˽”Counting”`
	`KCPC˽A˽”Weighing Basic”`	

I2, IBMT – Query device information (type, capacity)

Description

Use I2 to query information about the device (e.g. type and weighing capacity). The response is output as a whole string.

Syntax

Command

I2	Query of the device .

Responses

I2␣A␣”*«Type»*␣*«Capacity»*␣*«Unit»*”	Device/instrument type and capacity, with the correct number of digits depending on d.
I2␣I	Command understood but currently not executable (device is currently executing another command, e.g. taring).

Parameters / Return values

Name	Type	Values	Meaning
Type	string		Type of device / instrument
Capacity	string		Capacity of device / instrument
Unit	string		Weight unit

Comments

- With multi-range devices, the last decimal place is available only in the finer ranges.
- The number of characters of "text" depends on the device type and capacity.

Examples

↓	I2	Query of the device data
↑	I2␣A␣”GAT␣6K-4␣6000.00␣g”	Device type and capacity

I3 – Query device software version

Description

Provides the device software version(s).

Syntax

Command

I3	Query of the device software version.

Responses

I3˽A˽”*«Software»*[˽*«TNR»*]” [˽”*«ApplicationSoftware»*”]	Device software version and type number.
I3˽I	Command understood but currently not executable (device is currently executing another command, e.g. taring).

Parameters / Return values

Name	Type	Values	Meaning
Software	string		(Legally relevant) software (firmware) version
TNR	string		Type number (number identifying the software configuration parameters used). Not sent, if software is not parametrizable / configurable to different types (most firmware).
ApplicationSoftware	string		(Not legally relevant) application soft (firmware) ware version, if available.

Comments

- Only the software version of the terminal software is issued.
- If no terminal is present, the bridge software is issued instead.

Examples

↓	I3	Query of the Software version number(s) and type definition number
↑	I3˽A˽”4.10”	4 .10: Software version number. No type number.
↑	I3˽A˽”4.10˽10.142”	4 .10: Software version number. 10.142: Type number.
↑	I3˽A˽”4.10˽10.142”˽”2.141”	4 .10: (Legally relevant) software version number. 10.142: Type number. 2.141: (Not legally relevant) application software number.

`I4` , `IBIS` – Query / set serial number

Description

Use `I4` to query the serial number of the device. In the case of devices, the serial number of the terminal is output.

Syntax

Command

`I4` `IBIS`	Query of the serial number.
`IBIS˽”`*«SNR»*`”`	Set the serial number (if allowed).

Responses

`I4˽A˽”`*«SNR»*`”`
`IBIS˽A˽”`*«SNR»*`”`
	Serial number.

`I4˽I` `IBIS˽I`	Command not understood, not executable at present.
`IBIS˽A`	The serial number is set successfully.

Parameters / Return values

Name	**Type**	**Values**	**Meaning**
`SNR`	string		Serial number

Comments

- Due to production / cost reasons, the serial number may not be available over KCP.
  Here, the answer is N/A.
- The serial number agrees with that on the model plate and is different for every device.
- The serial number can be used, for example, as a device address in a network solution.
- The device response to `I4` appears unsolicited after switching on and after the cancel command `@`.
- Only the serial number of the terminal is issued.
- If no terminal is present, the serial number of the bridge is issued instead.

Examples

↓	`I4`	Query of serial number
↑	`I4˽A˽”WX1712345”`	The serial number is: WX1712345

↓	`IBIS˽”WX1712345”`	Set serial number
↑	`IBIS˽A`	Serial number set.

↓	`IBIS`	Query of serial number
↑	`IBIS˽A˽”N/A”`	No serial number available.

See also

➔	`@` - cancel

I5 – Query software identification number

Description

Identical to I3.

`IBIM` – Query/set balance external model number

Description

Set the balance brand model number (external article number for clients/sale).

Syntax

Command

`IBIM`	Query external model number.
`IBIM˽"«ModelNumber»"`	Set external model number.

Responses

`IBIM˽A˽"«ModelNumber»"`	The external model number.
`IBIM˽A`	The external model number is set successfully.
`IBIM˽L`	Model number invalid (too short/long).

Parameters / Return values

Name	Type	Values	Meaning
`ModelNumber`	string		External model number (max. 31 characters).

English

Examples

↓	`IBIM˽"IFB 30K-2M"`	
↑	`IBIM˽A`	

4 KCP commands – category "Device Display" (level 1)

`D`	Display: Write text to display
`DM`	Query/set display mode
`DW`	Display: Show weight
`IBBS`	Query battery status
`K`	Keys: Set configuration
`PWR`	Power on/off

D – Display: Write text to display

Description

Use D to write text to the device display.

Syntax

Command

D␣"«*DisplayText*»"	Write text into the device display.

Responses

D␣A	Command understood and executed successfully: Text appears left-aligned in the device display marked by a symbol, e.g. *.
D␣I	Command understood but currently not executable.
D␣L	Command understood but not executable (incorrect parameter or device with no display).

Parameters / Return values

Name	Type	Values	Meaning
DisplayText	string		Text on the device display

Comments

- A symbol in the display, e.g. * indicates that the device is not displaying a weight value.
- The maximum number of characters of "text" visible in the display depends on the device type. If the maximum number of characters is exceeded, the text disappears on the right side.
- Quotation marks can be displayed as indicated

Examples

↓	D␣"HELLO"	Write "HELLO" into the device display
↑	D␣A	The full text HELLO appears in the device display

↓	D␣" "	Clear the device display
↑	D␣A	Device display cleared, marked by a symbol, e. g. *

See also

➔	DW – Display: Show weight

DM – Query / set display mode

Description

Describe the command in detail here.

Syntax

Command

`DM`	Query display mode.
`DM˽«DisplayMode»`	Set display mode.

Responses

`DM˽A˽«DisplayMode»`	Current display mode.
`DM˽A`	Display mode is set successfully.
`DM˽I`	Command understood but currently not executable.
`DM˽L`	Command understood but not executable (no display or incorrect parameter).

Parameters / Return values

Name	Type	Values	Meaning
`DisplayMode`	enum	DEF	Regular display mode.
		OFF	Display completely off (no segments)
		TXT	Display text defined in “D” only.

Comments

This command can be used (combined with `K`), to disable the balance indication. Useful when only the remote indicator displays relevant/corrent information.

Examples

↓	`DM`	Query current display mode.
↑	`DM˽A˽DEF`	The current, stable ("S") weight value is 100.00 g

↓	`DM˽OFF`	Switch display off.
↑	`DM˽A`	OK, display switched off.

↓	`DM˽TXT`
	`DM˽”Hello World”`
		Set display to display text only, display “Hello World”

↑	`DM˽A`	OK.

See also

➔	`D – Display text`

DW – Display: Show weight

Description

Writes the current weight value to the device display using the set unit. This command is used to reset the display after using the D command.

Syntax

Command

DW	Switch the main display to weight mode.

Responses

DW␣A	Command understood and executed successfully: Main display shows the current weight value.
DW␣I	Command understood but currently not executable.

Comments

- DW resets the device display following a [D] command.

Examples

↓	DW	Switch the main display to weight mode
↑	DW␣A	Main display shows the current weight value

See also

➔	D – Display: Write text to display

IBBS – Query battery status

Description

Query the current battery status.

Syntax

Command

`IBBS`	Retrieves the current battery capacity and charging status.

Responses

`IBBS˽A˽`*«BatteryIndicator»*`˽%˽`*«ChargeStatus»*	Current battery status.
`IBBS˽L`	No battery or no mechanism to retrieve battery charging status.

Parameters / Return values

Name	Type	Values	Meaning
`BatteryIndicator`	integer	0 - 100	Battery charge status in percent (how much battery capacity is left). If the device does not support providing the exact battery status, the following values will be sent: • 0: battery low • 100: battery charged sufficiently or no information about battery available
`ChargeStatus`	string	N	No external power, not charging, running from battery.
		C	Currently charging.
		F	Fully charged.

Comments

- This command is mainly used for displaying a battery indicator in remote devices.

K – Keys: Set configuration

Description

With the K command, the behavior of the terminal keys may be configured: first, the K command controls whether a key invokes its corresponding function or not and second, it configures whether an indication of which key has been pressed or released is sent to the host interface or not.

Using this functionality, an application running on a connected system (e.g. a PC or PLC) may make use of the device terminal to interact with the device operator.

Syntax

Command

K˽*«Mode»*	Set configuration.

Responses

K˽A[˽*«FunctionID»*]	Command understood and executed successfully. Mode 4: Function with *«FunctionID»* was invoked by pressing the corresponding key and executed successfully.
K˽I[˽*«FunctionID»*]	Command understood but currently not executable (device is actually in menu or input mode). Mode 4: Function with *«FunctionID»* by pressing the corresponding key, but it could not be successfully executed (e.g. calibration was aborted by user or a negative value was tared).
K˽L	Command understood but not executable (incorrect or no parameter).

Additional Responses in Mode 3:

K˽*«EventID»*˽*«KeyID»*	Key *«KeyID»* has issued an *«EventID»*.

Additional Responses in Mode 4:

K˽B˽*«FunctionID»*	Function with *«FunctionID»* was invoked and started; the execution needs time to complete.

Parameters / Return values

Name	Type	Values	Meaning
Mode	integer	1	Functions are executed, no indications are sent (factory setting)
		2	Functions are not executed, no indications are sent
		3	Functions are not executed, indications are sent
		4	Functions are executed, indications are sent
EventID	char	R	Key was pressed and held around 2 seconds
		C	Key was released(after being pressed shortly or for 2 second)
FunctionID	integer	0	Adjustment
		1	Tare
		2	Zero
		3	Data transfer to printing device
		4 ... 6	Reserved for future use
		7	Test
KeyID	integer	1	Home
		2	User profile
		3	Settings
		5	Zero
		7	Transfer
		8	Configure actual applications
		9	Applications

Comments

- `K␣1` is the factory setting (default value).
- `K␣1` active after device switched on and after the cancel command [`@`].
- Only one `K` mode is active at one time.

Examples

When a code with a long press is sent, new key commands will not be accepted.

↓	`K␣4`	Set mode 4: when a key is pressed, execute the corresponding function and send the function number as a response
↑	`K␣A`	Command executed successfully
↑	`K␣B␣1`	The taring function has been started → taring active
↑	`K␣A␣1`	Taring completed successfully
↑	`K␣B␣1`	The taring function has been started → taring active
↑	`K␣I␣1`	Taring not completed successfully, taring aborted (e.g. tried to tare a negative value)

PWR – Power on/off

Description

Switch device on, off or into standby.

Syntax

Command

`PWR`	Query current power state (if possible).
`PWR␣`*«On/Off»*	Set current power state.

Responses

`PWR␣A`	Device has been switched off successfully.

`PWR␣A`
`I4␣A␣"`*«SNR»*`"`
	Device has been switched on successfully.
	Serial number is sent after startup.

`PWR␣I`	Command understood but currently not executable (device is currently executing another command, e.g. taring, or timeout as stability was not reached).
`PWR␣L`	Command understood but not executable (not capable of switching power states or incorrect/unsupported parameter).

Parameters / Return values

Name	Type	Values	Meaning
`On/Off`	integer	0	Switch to standby mode (lower power consumption).
		1	Switch device on.
		2	Switch device off completely (lowest power state).

Comments

- It depends on the device whether switching on from the lowest power state is possible using the `PWR` command.

5 KCP commands – category “Weighing Basic” (level 0)

The commands from Level 0 offer the very basic functions available for every basic weighing device.

S	Send stable indication (weight value / measured value)
SI	Send current indication immediately
SIR	Send current indication immediately and repeat
T	Tare
TI	Tare immediately
TZ	Tare or zero the balance (e.g. combined tare/zero button)
U	Query or set display and host unit
Z	Zero after stability
ZI	Zero immediately

`S` – Send stable weight value

Description

Use `S` to send a stable weight value, along with the unit.

Syntax

Command

`S`	Send the current stable net weight value.

Responses

`S˽S˽`*«WeightValue»*`˽`*«Unit»*	Current stable weight value in unit set.
`S˽I`	Command understood but currently not executable (device is currently executing another command, e.g. taring, or timeout as stability was not reached).
`S˽L`	Command understood but not executable (incorrect parameter).
`S˽+`	Device in overload range.
`S˽-`	Device in underload range.
`S˽S˽`*«ErrorCode»*	Code of error occurred

Parameters / Return values

Name	Type	Values	Meaning
`WeightValue`	float		Weight value
`Unit`	string		Currently displayed unit
`ErrorCode`	string		Code of error occurred

Comments

- The duration of the timeout depends on the device type.
- The weight value is formatted as a right aligned string with 10 characters including the decimal point.
- Preceding zeros are not shown except for the zero to the left of the decimal point.
- For multi-range or floating range balances, the decimal places at the end that are not displayed (in higher ranges) are shown as spaces.

Examples

↓	`S`	Send a stable weight value
↑	`S˽S˽˽˽˽˽100.00˽g`	The current, stable ("S") weight value is 100.00 g

↑	`S˽S˽˽˽˽˽200.00˽g`
	`S˽S˽˽˽˽˽200.0˽˽g` (d = 0.01/0.1g)
	
	`S˽S˽˽˽˽˽˽200.0˽g`
	`S˽S˽˽˽˽˽˽200.˽˽g` (d = 0.1/1g)
		In a higher range (for multi/floating-range balances), the last digit disappears and is replaced with a space.

↑	`S˽S˽˽˽˽˽200.˽˽˽g`	Theoretically, even two spaces could be missing with d=0.01/0.1/1g.
↑	`S˽S˽˽˽˽˽˽10000˽g`	When there is no decimal point, the value is still right aligned as described above.

SI – Send weight value immediately

Description

Use SI to immediately send the current weight value, along with the unit.

Syntax

Command

SI	Send the current net weight value, irrespective of device stability.

Responses

S␣S␣*«WeightValue»*␣*«Unit»*	Stable weight value in unit set
S␣D␣*«WeightValue»*␣*«Unit»*	Non-stable (dynamic) weight value in unit set
S␣I	Command understood but currently not executable (device is currently executing another command, e.g. taring).
S␣L	Command understood but not executable (incorrect parameter).
S␣+	Device in overload range.
S␣-	Device in underload range.
S␣S␣*«ErrorCode»*	Code of error occurred

Parameters / Return values

Name	Type	Values	Meaning
WeightValue	float		Weight value
Unit	string		Currently displayed unit
ErrorCode	string		Code of error occurred

Comments

- The device response to the command SI is the last internal weight value (stable or dynamic) before receipt of the command SI.
- The weight value is formatted as described in the comments of the S command.

Examples

↓	SI	Send current weight value
↑	S␣D␣␣␣␣␣129.07␣g	The weight value is unstable (dynamic, "D") and is currently 129.07 g

SIR – Send weight value immediately and repeat

Description

Use SIR to immediately send the current weight value, along with the unit, on a continuous basis.

Syntax

Command

SIR	Send the net weight values repeatedly, irrespective of device stability. The default time between transmissions is device dependent (typically around 15 Hz).
SIR␣*«TimeMsBetweenTransmissions»*	As above, setting the time between transmissions explicitly (in milliseconds).

Responses

S␣S␣*«WeightValue»*␣*«Unit»*	Stable weight value in unit set
S␣D␣*«WeightValue»*␣*«Unit»*	Non-stable (dynamic) weight value in unit set
S␣I	Command understood but currently not executable (device is currently executing another command, e.g. taring).
S␣L	Command understood but not executable (incorrect parameter).
S␣+	Device in overload range.
S␣-	Device in underload range.
S␣S␣*«ErrorCode»*	Code of error occurred

Parameters / Return values

Name	Type	Values	Meaning
TimeMsBetweenTransmissions	int		Time in milliseconds between repeated transmissions
WeightValue	float		Weight value
Unit	string		Currently displayed unit
ErrorCode	string		Code of error occurred

Comments

- SIR is overwritten by the commands S, SI, @ and hardware break and hence cancelled.
- The weight value is formatted as a right aligned string with 10 characters including the decimal point.

Examples

↓	SIR	Send current weight values at intervals
↑	S␣D␣␣␣␣␣129.07␣g	The device sends stable ("S") or unstable ("D") weight values at intervals
↑	S␣D␣␣␣␣␣129.08␣g	
↑	S␣S␣␣␣␣␣129.09␣g	
↑	S␣S␣␣␣␣␣129.09␣g	
↑	S␣D␣␣␣␣␣129.87␣g	
↑	S␣...	

T – Tare

Description

Use T to tare the device. The next stable weight value will be saved in the tare memory.

Command

T	Tare, i.e. store the next stable weight value as a new tare weight value.

Responses

T˽S˽*«TareWeightValue»*˽*«Unit»*	Taring successfully performed. The tare weight value returned corresponds to the weight change on the device in the unit set since the last zero setting.
T˽I	Command understood but currently not executable (device is currently executing another command, e.g. zero setting, or timeout as stability was not reached).
T˽L	Command understood but not executable (incorrect parameter).
T˽+	Upper limit of taring range exceeded.
T˽-	Lower limit of taring range exceeded.

Parameters / Return values

Name	Type	Values	Meaning
TareWeightValue	float		Weight value
Unit	string		Currently displayed unit

Comments

- The tare memory is overwritten by the new tare weight value.
- The duration of the timeout depends on the device type.
- Clearing tare memory: See TAC.
- The weight value is formatted as a right aligned string with 10 characters including the decimal point.

Examples

↓	T	Tare
↑	T˽S˽˽˽˽˽100.00˽g	The device is tared and has a value of 100.00 g in the tare memory

See also

➔	TAC – Clear tare value

`TI` – Tare immediately

Description

Use `TI` to tare the device immediately and independently of device stability.

Command

`TI`	Tare immediately, i.e. store the current weight value, which can be stable or non stable (dynamic), as are weight value.

Responses

`TI␣S␣`*«TareWeightValue»*`␣`*«Unit»*	Taring performed, stable tare value. The new tare value corresponds to the weight change on the device since the last zero setting.
`TI␣D␣`*«TareWeightValue»*`␣`*«Unit»*	Taring performed, non-stable (dynamic) tare value.
`TI␣I`	Command understood but currently not executable (device is currently executing another command, e.g. zero setting).
`TI␣L`	Command understood but not executable (e.g. certified version of the device).
`TI␣+`	Upper limit of taring range exceeded.
`TI␣-`	Lower limit of taring range exceeded.

Parameters / Return values

Name	Type	Values	Meaning
`TareWeightValue`	float		Tare Weight value
`Unit`	string		Currently displayed unit

Comments

- The tare memory will be overwritten by the new tare weight value.
- After a non-stable (dynamic) stored tare weight value, a stable weight value can be determined. However, the absolute value of the stable weight value determined in this manner is not accurate.
- The taring range is specified to the device type.
- The weight value is formatted as a right aligned string with 10 characters including the decimal point.
- The stored tare weight value is sent in the unit set

Examples

↓	`TI`	Tare immediately
↑	`TI␣D␣␣␣␣␣117.57␣g`	The tare memory holds a non-stable (dynamic) weight value

See also

➔	`TAC – Clear tare value`

TZ – Combined Tare/Zero

Description

Tare or zero the balance, depending on the current load (like a combined tare/zero button).

Syntax

Command

`TZ`	Tare or zero the balance.

Responses

`TZ˽A˽`*«TareOrZero»*`[˽`*«TareWeightValue»*`˽`*«Unit»*`]`	Balance tared or zeroed, depending on current load. If tared, the new tare value is being sent as parameter.
`TZ˽I`	Command understood but currently not executable (device is currently executing another command, e.g. zero setting, or timeout as stability was not reached).
`TZ˽L`	Command understood but not executable (incorrect parameter).
`TZ˽+`	Upper limits exceeded.
`TZ˽-`	Lower limits exceeded.

Parameters / Return values

Name	Type	Values	Meaning
`TareOrZero`	string	T	Tare operation executed
		Z	Zero operation executed
`TareWeightValue`	float		Tare weight value (only when tared)
`Unit`	string		Currently displayed unit

Examples

↓	`TZ`	Tare or zero.
↑	`TZ˽A˽Z`	Balance zeroed, tare value is cleared.

↓	`TZ`	Tare or zero.
↑	`TA˽A˽˽T˽˽˽˽100.00˽g`	Balance tared, the device has a value of 100.00 g in the tare memory

See also

→	`T – Tare`
→	`Z – Zero`
→	`TAC – Clear tare value`

`U` – Query / set display and host unit

Description

This command retrieves or sets both the display and the host unit.

The *display unit* is the unit displayed in the display of the indicator.

The *host unit* is the unit used to send weighing values to the *host* (remote device / computer).

Syntax

Command

`U`	Query the current display unit.
`U␣`*«UnitSymbol»*	Set the current display and host unit.

Responses

`U␣A␣`*«UnitSymbol»*	Returns the currently set display unit symbol.
`U␣A`	Unit successfully set.
`U␣I`	Unit symbol invalid or required factors not set (see below).

Parameters / Return values

Name	Type	Values	Meaning
`UnitSymbol`	string	see [*Units*]	Symbol of the unit to set.

Comments

- For certain units (e.g. percent, pieces, free factor, …), before using this command, the corresponding factor has to be set using KCP commands or the balance keyboard.

Examples

↓	`U`	Query unit
↑	`U␣A␣g`	The current unit is gram (g).
↑	`U␣A␣kg`	The current unit is kilogram (kg).

↓	`U␣g`	Set the units to gram (g).
↑	`U␣A`	The unit is set now.

↓	`U␣%`	Set the unit to percent (%).
↑	`U␣I`	Invalid action, because the weight of 100% was not set before.

`Z` – Zero (after stability)

Description

Use `Z` to set a new zero; all weight values (including the tare weight) will be measured relative to this zero. After zeroing has taken place, the following values apply: tare weight = 0; net weight (= gross weight) = 0.

Syntax

Command

`Z`	Zero the device.

Responses

`Z␣A`	Zero setting successfully performed. Gross, net and tare = 0.
`Z␣I`	Command understood but currently not executable (device is currently executing another command, e.g. taring, or timeout as stability was not reached).
`Z␣+`	Upper limit of zero setting range exceeded.
`Z␣-`	Lower limit of zero setting range exceeded.

Comments

- The tare memory is cleared after zero setting.
- The zero point determined during switching on is not influenced by this command, the measurement ranges remain unchanged.
- The duration of the timeout depends on the device type.

Examples

↓	`Z`	Zero
↑	`Z␣A`	Zero setting performed

`ZI` – Zero immediately

Description

Use `ZI` to set a new zero immediately, regardless of device stability. All weight values (including the tare weight) will be measured relative to this zero. After zeroing has taken place, the following values apply: tare weight = 0; net weight (= gross weight) = 0.

Syntax

Command

`ZI`	Zero the device immediately regardless the stability of device..

Responses

`ZI␣D`	Re-zero performed under non-stable (dynamic) conditions.
`ZI␣S`	Re-zero performed under stable conditions.
`ZI␣I`	Command understood but currently not executable (device is currently executing another command, e.g. taring).
`ZI␣+`	Upper limit of zero setting range exceeded.
`ZI␣-`	Lower limit of zero setting range exceeded.

Comments

- The tare memory is cleared after zero setting.
- This command is not supported by approved devices.
- The zero point determined during switching on is not influenced by this command, the measurement ranges remain unchanged.

Examples

🡻	`ZI`	Zero immediately
🡹	`ZI␣D`	Re-zero performed under non-stable (dynamic) conditions

6 KCP commands – category “Weighing Advanced” (level 1)

The commands from Level 1 are available for all more advanced weighing instruments.

IBRL	List of balance range information
IBRT	Query balance ranges type
IVERS	Query/set verification state
SR	Send weight value on weight change (send and repeat)
TA	Query/preset tare weight value
TAI	Query/preset (internal) tare weight value
TAC	Clear tare value

`IBRL` / BalanceRangesList – List of balance range information

Description

Query balance range information.

Syntax

Command

IBRL	Query balance range information.

Responses

IBRL˽B˽*«RangeNr»*˽*«Max»*˽*«Unit»*˽*«d»*˽*«Unit»* [˽*«Min»* ˽*«Unit»* ˽*«e»* ˽*«Unit»*] IBRL˽B˽*«RangeNr»*˽*«Max»*˽*«Unit»*˽*«d»*˽*«Unit»* [˽*«Min»* ˽*«Unit»* ˽*«e»* ˽*«Unit»*] … IBRL˽A˽*«RangeNr»*˽*«Max»*˽*«Unit»*˽*«d»*˽*«Unit»* [˽*«Min»* ˽*«Unit»* ˽*«e»* ˽*«Unit»*]	Information about first range (if multiple). Information about second range (if multiple). Information about last range. (Min and e are optional when it is not verifyable)
IBRL˽I	Command understood but currently not executable.

Parameters / Return values

Name	Type	Values	Meaning
RangeNr	float	0,1,2,3,…	Number of the range
Max	float		Max (capacity of this range)
d	float		d (readout)
Min	float		Min (minimum verification value)
e	float		e (verification interval)
Unit	string	see *Units*	Unit for the corresponding value.

IBRT – Query balance ranges type

Description

Query the type of the balance ranges. This defines the way, the balance switches between ranges (if multiple).

Syntax

Command

IBRT	Query balance range type.

Responses

IBRT␣A␣«*BalanceRangeType*»	Answer with balance range type.

Parameters / Return values

Name	Type	Values	Meaning
BalanceRangeType	string	SR	Single range
		MR	Multi range
		FR	Floating range

English

IVERS – Query/set verification state

Description

Query or set current state of verification.

Syntax

Command

IVERS	Query current verification state.
IVERS␣*«VerificationState»*	Set current verification state.

Responses

IVERS␣A␣*«VerificationState»*	Current verification state.
IVERS␣A	Verification state set successfully.
IVERS␣I	Command understood but currently not executable (device is currently executing another command, e.g. taring, or timeout as stability was not reached).
IVERS␣L	Command understood but not executable (incorrect parameter.

Parameters / Return values

Name	Type	Values	Meaning
VerificationState	bool	0	not in verified mode or no type approval / verification not possible.
		1	in verified mode

SR – Send weight value on weight change (send and repeat)

Description

Use SR to send the current weight values following a predefined minimum change in weight and on a continuous basis. The weight value is sent, along with the unit.

Command

SR	Send the current stable weight value and then continuously after every weight change. If no preset value is entered, the weight change must be at least 12.5% of the last stable weight value, minimum = 30 digit.
SR␣*«PresentValue»*␣*«Unit»*	Send the current stable weight value and then continuously after every weight change greater or equal to the preset value a non-stable (dynamic) value followed by the next stable value, range = 1 digit to maximal capacity.

Responses

S␣S␣*«WeightValue»*␣*«Unit»*	Current, stable weight value in unit set, 1st weight change.
S␣D␣*«WeightValue»*␣*«Unit»*	Dynamic weight value in unit set.
S␣S␣*«WeightValue»*␣*«Unit»*	Next stable weight value in unit set.
S␣I	Command understood but currently not executable (device is currently executing another command, e.g. zero setting, or timeout as stability was not reached).
S␣L	Command understood but not executable (incorrect parameter).
S␣+	Device in overload range.
S␣-	Device in underload range.
S␣S␣*«ErrorCode»*	Code of error occurred

Parameters / Return values

Name	Type	Values	Meaning
WeightValue	float		Weight value
Unit	string		Unit, only available units permitted
ErrorCode	string		Code of error occurred

Comments

- SR is overwritten by the commands S, SI, @ and hardware break and hence cancelled.
- If, following a non-stable (dynamic) weight value, stability has not been reached within the timeout interval, the response S␣I is sent and then a non-stable weight value. Timeout then starts again from the beginning.
- The preset value can be entered in any by the device accepted unit.
- The weight value is formatted as a right aligned string with 10 characters including the decimal point.

Examples

↓	`SR˽10.00˽g`	Send the current stable weight value followed by every load change of 10 g
↑	`S˽S˽˽˽˽˽100.00˽g`	Device stable
↑	`S˽D˽˽˽˽˽115.23˽g`	100.00 g loaded
↑	`S˽S˽˽˽˽˽200.00˽g`	Device again stable

See also

➔	`S – Send stable weight value`
➔	`SI – Send weight value immediately`
➔	`SIR – Send weight value immediately and repeat`

English

TA – Query/preset tare weight value

Description

Use TA to query the current tare value or preset a known tare value.

Command

TA	Query of the current tare weight value (rounded).
TA˽*«TarePresentValue»*˽*«Unit»*	Preset of a tare value.

Responses

TA˽A˽*«TareWeightValue»*˽*«Unit»*	Query current tare weight value in tare memory, in unit set.
TA˽I	Command understood but currently not executable (device is currently executing another command, e.g. zero setting, or timeout as stability was not reached).
TA˽L	Command understood but not executable (incorrect parameter).

Parameters / Return values

Name	Type	Values	Meaning
TareWeightValue	float		Rounded tare weight value
Unit	string		Currently displayed unit

Comments

- The tare memory will be overwritten by the preset tare weight value.
- The inputted tare value will be automatically rounded by the device to the current readability.
- The taring range is specified to the device type.
- The weight value is formatted as a right aligned string with 10 characters including the decimal point.

Examples

↓	TA˽100.00˽g	Preset a tare weight of 100 g
↑	TA˽A˽˽˽˽˽100.00˽g	The device has a value of 100.00 g in the tare memory

See also

➔	TAC – Clear tare value

TAI – Query/preset tare weight value (internal, not rounded)

Description

Use TAI to query the current, unrounded tare value or preset a known exact tare value.

Command

TAI	Query of the current tare weight value in the internal resolution (not rounded).
TAI˽*«TarePresentValue»*˽*«Unit»*	Preset of a tare value.

Responses

TAI˽A˽*«TareWeightValue»*˽*«Unit»*	Query current tare weight value in tare memory, in unit set.
TAI˽I	Command understood but currently not executable (device is currently executing another command, e.g. zero setting, or timeout as stability was not reached).
TAI˽L	Command understood but not executable (incorrect parameter).

Parameters / Return values

Name	Type	Values	Meaning
TareWeightValue	float		Exact tare weight value
Unit	string		Currently displayed unit

Comments

- The tare memory will be overwritten by the preset tare weight value.
- The inputted tare value will **not** be rounded to the current readability.
- The internal resolution and taring range is specified to the device type. Typically, the internal resolution is 5-10 times higher than the resolution of the TA command.

Examples

↓	TAI˽100.123˽g	Preset a tare weight of 100.123 g (even when d=0.01g).
↑	TAI˽A˽100.123˽g	The device has a value of 100.123 g in the tare memory

See also

➔	TAC – Clear tare value

TAC – Clear tare value

Description

Use TAC to clear the tare memory.

Command

TAC	Clear tare value.

Responses

TAC␣A	Tare value cleared, 0 is in the tare memory.
TAC␣I	Command understood but currently not executable (device is currently executing another command, e.g. zero setting).
TAC␣L	Command understood but not executable (incorrect parameter).

Examples

↓	TAC	Clear tare value
↑	TAC␣A	Tare value cleared, 0 is in the tare memory

See also

➔	T – Tare
➔	TI – Tare immediately
➔	TA – Query/preset tare weight value
➔	TC – Tare or tare immediately after timeout

7 KCP commands – category “Weighing Adjustment” (level 2)

C3	Start adjustment with internal weight
I54	External adjustment loads
M19	Query/set adjustment weight (with error)

C3 – Start adjustment with internal weight

Description

You can use C3 to start an internal adjustment procedure.

Syntax

Command

C3	Start the internal adjustment.

First Responses

C3˽B	The adjustment procedure has been started. Wait for second response.
C3˽I	Adjustment cannot be performed at present as another operation is taking place. No second response follows.
C3˽L	Adjustment operation not possible (e.g. no internal weight). No second response follows.

Further Responses

C3˽A	Adjustment has been completed successfully.
C3˽I	The adjustment was aborted as, e.g. stability not attained or the procedure was aborted with the C key.

Comments

- Commands sent to the balance during the adjustment operation may not processed and responded to in the appropriate manner until the adjustment is at an end.

Examples

↓	C3	Start internal adjustment.
↑	C3˽B	Started.
↑	C3˽A	Completed successfully.

I54 – External adjustment loads

Description

This command lists the available loads for external adjustment.

Syntax

Command

`I54`	Query list of adjustment loads.

Responses

I54␣A␣*«Min»*␣*«Max»*␣*«Increment»*␣*«Unit»*	Adjustment loads are between “Min” and “Max”, with increments of “increment”.
I54␣B␣*«Load$_1$»*␣*«Unit»* I54␣B␣*«Load$_2$»*␣*«Unit»* … I54␣A␣*«Load$_n$»*␣*«Unit»*	Adjustment loads are the listed loads (1 to n).

Parameters / Return values

Name	Type	Values	Meaning
`Min`	float		Smallest load to use for external adjustment.
`Max`	float		Biggest load to use for external adjustment.
`Increment`	float		Load increment between Min and Max.
`Unit`	enum	see *Units*	Unit in which the previous values are given.

Examples

↓	`I54`	Query.
↑	I54␣A␣1000.0␣3000.0␣750.0␣g	In the case of external adjustment, the loads for selection are 1000 g, 1750 g, 2500 g and 3000 g.

`M19` – Query/set adjustment weight (with error)

Description

You can use `M19` to set the currently used external adjustment weight, or to query the current weight setting.

Syntax

Command

`M19`	Query the current adjustment weight.
`M19˽«Value»˽«Unit»`	Set the adjustment weight.

Responses

`M19˽A˽«Value»˽«Unit»`	Current adjustment weight.
`M19˽A`	Command understood and executed successfully.
`M19˽I`	Command understood but currently not executable (device is currently executing another command, e.g. taring, or timeout as stability was not reached).
`M19˽L`	Command understood but not executable (incorrect parameter).

Parameters / Return values

Name	Type	Values	Meaning
`Value`	float		Value of the adjustment weight, balance specific limitation
`Unit`	string		

Examples

↓	`M19`	Query current adjustment weight.
↑	`M19˽A˽100.123˽g`	The current adjustment weight value is 100 g, with a deviation of 0.123 g.

See also

➔	`I54 – Query adjustment loads`

8 KCP commands – category “Sensors” (level 2)

`SENSL`	Query list of sensors in the device
`SENSI`	Query detailed sensor information
`SENSQ`	Query current sensor value

SENSL / SensorList – Query list of sensors in the device

Description

Lists the available sensors and basic information about each sensor.

Syntax

Command

SENSL	

Responses

SENSL˽B˽*«ID»*˽*«Type»*˽*«Description»*˽*«CurrentValue»*˽*«Unit»* SENSL˽B˽*«ID»*˽*«Type»*˽*«Description»*˽*«CurrentValue»*˽*«Unit»* ... SENSL˽A˽*«ID»*˽*«Type»*˽*«Description»*˽*«CurrentValue»*˽*«Unit»*	First sensor. Second sensor. Last sensor.
SENSL˽L	No sensors.

Parameters / Return values

Name	Type	Values	Meaning
ID	int	0, 1, …	Sensor unique ID
Type	string	T	Temperature sensor
		H	Relative humidity sensor
		P	Air pressure sensor
		BL	Bubble level status sensor (ok, not ok)
		BLPos	Bubble level position (X, Y)
		DL	Door limit switch
Description	string		Internal description of the sensor, e.g. LoadCellTemp, CaseTemp, EnvTemp, EnvHumi, EnvPres, BubbleLevelStatus, BubbleLevelPosition, DoorLimitSwitch
CurrentValue	float		Current value of the sensor
Unit	string		Unit of the value of the sensor

Examples

↓	SENSL	Send a list of the available sensors.
↑	SENSL˽B˽0˽T˽EnvTemp˽22.00˽°C SENSL˽B˽1˽H˽EnvHumi˽50.00˽% SENSL˽B˽2˽P˽EnvPres˽106545.00˽Pa SENSL˽A˽3˽P˽EnvPres˽256512.00˽Pa	The current available sensor list.

See also

➔	SENSI – Query detailed sensor information
➔	SENSQ – Query current sensor value

SENSI / SensorInfo – Query detailed sensor information.

Description

Queries detail information about a specific sensor.

Syntax

Command

SENSI␣*«SensorID»*	

Responses

SENSI␣B␣*«SensorID»*␣*«ParKey»*␣*«ParValue»*[␣*«ParValueUnit»*] SENSI␣B␣*«SensorID»*␣*«ParKey»*␣*«ParValue»*[␣*«ParValueUnit»*] ... SENSI␣A␣*«SensorID»*␣*«ParKey»*␣*«ParValue»*[␣*«ParValueUnit»*]	First parameter. Second parameter. … Last parameter.
SENSI␣L	No sensors.

Parameters / Return values

Name	Type	Values	Meaning
SensorID	int	0, 1, …	Sensor unique ID
ParKey	string		Name/internal key of the parameter. See comments for possible parameter keys.
ParValue	-		Current value of the parameter
ParValueUnit	string		Unit of the value of the parameter, if applicable.

Examples

↓	SENSI␣1	Query detailed information of sensor 1.
↑	SENSI␣B␣1␣max␣1000␣N SENSI␣B␣1␣min␣10␣N … SENSI␣A␣1␣d␣0.1␣N	Sensor max. capacity is 1000 N. Sensor min. load is 10 N. … Sensor readability is 0.1 N.

Comments

The following parameters are defined at the moment:

Parameter name	Sensor types	Description
max	-	Maximum capacity
min	-	Minimum load (relative uncertainty limit)
d	-	Readability
manufacturer	-	Manufacturer name
typenumber	-	Type number

See also

➔	SENSI – Query detailed sensor information
➔	SENSQ – Query current sensor value

`SENSQ` / SensorQuery – Query current sensor value.

Description

Use this command to query a specific sensor value.

Syntax

Command

`SENSQ␣`*«SensorID»*	Query specific sensor value.

Responses

`SENSQ␣A␣`*«SensorValue»*`␣`*«Unit»*	Current sensor value in unit actually set under host unit.
`SENSQ␣I`	Command understood but currently not executable (device is currently executing another command, e.g. taring, or timeout as stability was not reached).
`SENSQ␣L`	Command understood but not executable (incorrect parameter).

Parameters / Return values

Name	Type	Values	Meaning
`SensorID`	integer	0, 1, 2, …	Sensor ID
`SensorValue`	float		Sensor value
`Unit`	string		Currently displayed unit

Examples

↓	`SENSQ␣0`	Send a sensor value from sensor 0
↑	`SENSQ␣A␣22.00␣°C`	The current sensor value is 22.00 °C

↓	`SENSQ␣1`	Send a sensor value from sensor 1
↑	`SENSQ␣A␣50.00␣%`	The current sensor value is 50.00 %

See also

➔	`SENSL – Query list of sensors in the device`
➔	`SENSI – Query detailed sensor information`

SENSQR / SensorQueryRepeat – Query current sensor values and repeat.

Description

Use this command to query a certain list of sensors at a regular interval.

This command is used for high frequency transmission of sensors.

Syntax

Command

SENSQR␣«*SensorIDs*»[␣«*Interval*»]	Query specific sensor value.
SENSQR	Stops current SENSQR transmission.

Responses

SENSQR␣A␣«*SensorValue*»␣«*Unit*»	Current sensor value in unit actually set under host unit.
SENSQR␣I	Command understood but currently not executable (device is currently executing another command).
SENSQR␣L	Command understood but not executable (incorrect parameter, unknown sensors, interval out of allowed value range).

Parameters / Return values

Name	Type	Values	Meaning
SensorIDs	string		Comma separated list of sensor IDs
Interval	integer		Interval at which the sensor values are sent, in milliseconds between the values. Optional, default value is 100ms.

Examples

↓	SENSQR␣0,1,5␣10 … SENSQR	Retrieve values from sensors 0, 1 and 5 at an interval of 10ms = 100 Hz … Stop transmission.
↑	SENSQR␣B␣22.00␣253.35␣-53.51 SENSQR␣B␣22.10␣252.26␣-53.51 SENSQR␣B␣22.21␣251.31␣-53.52 … SENSQR␣A␣25.81␣249.89␣-53.55	The current sensor values. Transmission ended.

Comments

Alternative: Transfer as byte encoded floating point values (shorter transmission and less encoding), if the above is not possible?

S	E	N	S	Q	R	␣	B	␣	🟨	🟨	␣	🟧	🟧	🟧	🟧	🟩	🟩	🟩	🟩	🟦	🟦	🟦	🟦	🟫	🟫	🟫	🟫	…	CR	LF
Header									Len			Sensor 1				Sensor 2				Sensor 3				Sensor 4					Newline	

See also

➔	SENSL – Query list of sensors in the device
➔	SENSI – Query detailed sensor information

9 KCP commands – category “External Terminal” (ETL)

`ETLS` – Start continuous status transmission

Description

Starts sending relevant balance status information continuously with a high repetition rate.

Syntax

Command

`ETLS[␣«`*`On/Off`*`»]`	Start/stop status transmission.

Responses

`ETLS␣A`	Acknowledged
	One line with the following information (repeated), separated by spaces:
`ETLS␣…␣…␣…␣…␣…␣…`	
`Nr`	Sequential 16-bit number, round-robin for hash code, 0 if not used
`NetInternalValue`	Internal net weighing value before rounding, but after all corrections.
`NetRoundedValue`	Net weighing value rounded to correct digits. Missing trailing digits (multi range balances) are replaced by underscore (_). E##### in case of errors in the weighing system (see *Message Codes*).
`GrossRoundedValue`	Gross weighing value rounded to correct digits. Missing trailing digits (multi range balances) are replaced by underscore (_). E##### in case of errors in the weighing system (see *Message Codes*).
`TareRoundedValue`	Tare value rounded to correct digits. Missing trailing digits (multi range balances) are replaced by underscore (_). E##### in case of errors in the weighing system (see *Message Codes*).
`Unit`	Current display unit.
`InsignificantDigits`	Number of insignificant digits. Examples: 0 (no insignificant digit, e=d) 1 (last digit is insignificant = grey digit) 2 (last two digits are insignificant = grey digits) 3 not possible 12.2340 g
`ActiveRangeNo`	Number of active range (0 … n-1)
`IndicatorFlags`	Encoded as characters 0 S Z N T P C when true, else missing. Meaning: • 0: Zero indicator • C: Internal calibration necessary • N: Net indicator • P: Print allowed • S: Stability indicator • Z: Zero allowed • T: Tare allowed Example: “ST” means stable and tare allowed, zero and print not allowed
`DetailStatusCode`	Message code for current (most important) status, see *Message Codes*. Used for current state, like initial zero error, or other errors. 0 = currently no message

`Hash`	Hash code for integrity of this line (0 if not used)

Parameters / Return values

Name	Type	Values	Meaning
`On/Off`	bool	0, 1	Enable/disable status transmission

Comments

This command is specifically designed for use with remote display indicators like EasyTouch tablets.

- The sending frequency depends on the balance capabilities, but should be at least 10 Hz for a fast display refresh rate.

Examples

↓	`ETLS`	Enables the ETL status transmission.
↓	`ETLS˽1`	Enables the ETL status transmission.

↑	`ETLS˽A`
	`ETLS˽…`
	`ETLS˽…`
	`ETLS˽…`
	…
		Transmission started.
		Status line
		Another status line
		Another status line

10 KCP commands – category “Axis movement”

At the moment, these commands are used mainly for test stands with one axis, but are designed generic to allow movement of any type and number of (linear or rotational) axis.

`AXISHOME`	Home the specified axis.
`AXISLIST`	List all available axes
`AXISMOVE`	Move the specified axis
`AXISPOS`	Retrieve the actual axis position
`AXISPOSRATE`	Set axis position reporting mode and interval
`AXISRAMP`	Set the acceleration and deceleration of an axis
`AXISSPEED`	Set the default speed of an axis

AXISMOVE – Move the specified axis

Description

Use AXISMOVE to move the specified axis to a position absolute or relative to the current position. The maximum movement speed can be specified optionally; otherwise the default axis speed and ramp parameters are used.

Syntax

Command

AXISMOVE␣*«AxisName»*␣*«PosSpecType»*␣*«Pos/Dist»*[␣*«Speed»*]	Move the specified axis.

Responses

AXISMOVE␣B AXISMOVE␣B␣*«AxisName»*␣*«CurPos»*␣*«CurSpeed»* AXISMOVE␣B␣*«AxisName»*␣*«CurPos»*␣*«CurSpeed»* … AXISMOVE␣A	Axis started to move Axis is moving, current absolute position and speed is reported regularly. … Movement completed.
AXISMOVE␣I	Command understood but currently not executable (e.g. device not homed, device is currently executing another command)

Parameters / Return values

Name	Type	Values	Meaning
AxisName	string		Name of the axis (Example “X”, “Y”, “Z”, or other, as given by the AXISLIST command)
PosSpecType	enum	BY	Moving relative from the current position. Negative values mean the axis should move in the opposite direction.
		TO	Moving to an absolute position (only after homing the system).
Pos/Dist	float		In BY mode the value is the distance to move from the current position. Can be interrupted when the axis triggers a limit switch. Example: +100 means move 100 units up/right, -100 means move 100 units down/left from the current position.
			In TO mode, this parameter specifies the absolute target position. The relative movement distance is known to the system, because the system was homed at a specified zero. Example: TO 100 means to move to position 100.
Speed	float		Axis maximum travel speed.
CurPos	float		The current absolute position of the axis (0 if not available).
CurSpeed	float		The speed the axis is currently moving at.

Comments

- Speed default depends on the device (standard: 20 mm/min).
- Default distance units are “mm” for a linear axis and degrees for a rotational axis.
- Default speed units are “mm/min” for a linear axis and degrees/s for a rotational axis.
- The given speed is the maximum speed reached after completing ramp-up and ramp-down (if applicable). See the AXISRAMP command for details.
- The interval of reporting the current position and speed depends on the device hardware capability; see the AXISPOSRATE command on how to configure rate and reporting mode.

Examples

↓	`AXISMOVE˽X˽BY˽100`	Move X axis by 100 units from current position
↑	`AXISMOVE˽B˽X`	Command understood, axis X started moving
↑	`AXISMOVE˽B˽X˽115˽5`	Axis X is currently at absolute position 115, moving with speed 5.
↑	`AXISMOVE˽B˽X˽120˽10`	Axis X is currently at absolute position 120, moving with speed 10.
↑	`AXISMOVE˽B˽X˽125˽20`	…
↑	`AXISMOVE˽B˽X˽130˽20`	…
↑	…	
↑	`AXISMOVE˽A˽X`	Movement of axis X completed successfully

AXISLIST – List all available axes

Description

List all available and configured axes of the device.

Syntax

Command

AXISLIST	List the available axes.

Responses

AXISLIST␣B␣*«Name»*␣*«Type»*␣*«Desc»*␣*«CurPos»*␣*«Unit»* AXISLIST␣B␣*«Name»*␣*«Type»*␣*«Desc»*␣*«CurPos»*␣*«Unit»* ... AXISLIST␣A␣*«Name»*␣*«Type»*␣*«Desc»*␣*«CurPos»*␣*«Unit»*	First axis. Second axis. Last axis.
AXISLIST␣I	Command understood but currently not executable.

Parameters / Return values

Name	Type	Values	Meaning
Name	string	X, Y, …	Internal name / ID of the axis
Type	enum	LIN	Linear axis
		ROT	Rotational axis
Description	string		Internal description of the sensor, e.g. MainAxis
CurPos	float		Current absolute position of the axis (0 if not homed)
Unit	string		Unit of axis position (usually mm or degrees)

Comments

- Most current devices only have one axis.

Examples

↓	AXISLIST	List all axis
↑	AXISLIST␣B␣X␣LIN␣SpindleZ␣51␣mm AXISLIST␣A␣Y␣LIN␣TableX␣151.3␣mm	First axis: Linear, ID X, descr. “SpindleZ” Second axis: Linear, ID Y, descr. “TableX”

AXISHOME – Retrieve current axis position

Description

AXISHOME brings the specified axis to its default home position (usually by slowly moving to a reference switch or scale). Most axes require homing to calculate absolute positions, e.g. after power loss or reset of the device.

Syntax

Command

AXISHOME␣*«AxisName»*[␣*«Speed»*]	Homes the specified axis.

Responses

AXISHOME␣B␣*«AxisName»* AXISHOME␣A␣*«AxisName»*	Homing started, axis begins to move (if not already at home position). Homing successfully completed.
AXISHOME␣I	Command understood but currently not executable.
AXISHOME␣L	Command understood but not executable (unknown axis or impossible speed value).

Parameters / Return values

Name	**Type**	**Values**	**Meaning**
AxisName	string		Name of the axis (Example "X", "Y", "Z", or other, as given by the AXISLIST command)
Speed	float		Optional homing speed value (overrides default).

Examples

↓	AXISHOME␣X	Home axis X.
↑	AXISHOME␣B␣X AXISHOME␣A␣X	Understood, axis X starts moving

↓	AXISHOME␣Y	Home axis Y.
↑	AXISHOME␣A␣Y	Axis Y at home position already.

AXISPOS – Retrieve current axis position

Description

AXISPOS queries the current position of the specified axis.

Syntax

Command

AXISPOS␣*«AxisName»*	Query position of axis.

Responses

AXISPOS␣A␣*«AxisName»*␣*«CurPos»*	Current position of axis.
AXISPOS␣I	Command understood but currently not executable.
AXISPOS␣L	Command understood but not executable (unknown axis or axis not homed).

Parameters / Return values

Name	Type	Values	Meaning
AxisName	string		Name of the axis (Example “X”, “Y”, “Z”, or other, as given by the AXISLIST command)
CurPos	float		Current absolute position of the axis.

Examples

↓	AXISPOS␣X	Query position of axis X.
↑	AXISPOS␣A␣X␣13.22	Axis X is at 13.22.

AXISSPEED – Configure the default speed of an axis

Description

This command configures the default speed of the specified axis (if no specific speed given in the move command).

Command

AXISRAMP␣*«AxisName»*	Query parameters.
AXISRAMP␣*«AxisName»*␣*«Speed»*	Set the default axis speed

Responses

AXISRAMP␣A␣*«AxisName»*␣*«Speed»*	Returns the current parameter for the axis
AXISRAMP␣A	New default speed for the axis set
AXISRAMP␣I	Command understood but currently not executable (device is actually in menu or input mode).
AXISRAMP␣L	Command understood but not executable (incorrect or no parameter).

Parameters / Return values

Name	Type	Values	Meaning
AxisName	string		Name of the axis (Example “X”, “Y”, “Z”, or other, as given by the AXISLIST command)
Speed	float		Axis default speed

Comments

- Speed value limits depend on the axis and motor capacity.

Examples

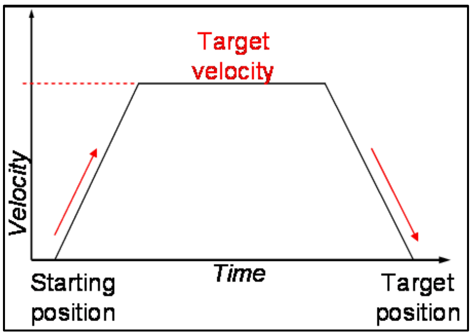
↓	AXISSPEED␣X␣2	Set the default speed for axis X to 2 mm/s per second.
↑	AXISSPEED␣A	Command executed successfully

↓	AXISSPEED␣X	Query parameters.
↑	AXISSPEED␣A␣X␣2	Axis X default speed is 2.

`AXISRAMP` – Configure the acceleration and deceleration of an axis

Description

This command configures the acceleration and deceleration ramp slopes of the specified axis.



Command

`AXISRAMP␣`*«AxisName»*	Query parameters.
`AXISRAMP␣`*«AxisName»*`␣`*«Acc»*`[␣`*«Dec»*`]`	Set the acceleration and deceleration values for the movement of the axis

Responses

`AXISRAMP␣A␣`*«AxisName»*`␣`*«Acc»*`␣`*«Dec»*	Show the current parameter for the ramp
`AXISRAMP␣A`	New ramp parameters for the axis configured
`AXISRAMP␣I`	Command understood but currently not executable (device is actually in menu or input mode).
`AXISRAMP␣L`	Command understood but not executable (incorrect or no parameter).

Parameters / Return values

Name	Type	Values	Meaning
`AxisName`	string		Name of the axis (Example “X”, “Y”, “Z”, or other, as given by the `AXISLIST` command)
`Acc`	float		Value for the acceleration ramp in mm/s²
`Dec`	float		Value for the deceleration ramp in mm/s². If not specified, the value of *`Acc`* will be used.

Comments

- Standard ramps depend on the axis and motor capacity.
- Ramp is implemented for smoother start and stop of the axis, in particular on heavy loads.

Examples

↓	`AXISRAMP␣X␣2`	Set the acceleration and deceleration rate for axis X to 2 mm/s per second.
↑	`AXISRAMP␣A`	Command executed successfully

↓	`AXISRAMP␣X`	Query parameters.
↑	`AXISRAMP␣A␣X␣2`	Axis X ramp parameter is 2 mm/s per second.

AXISPOSRATE – Set axis position reporting mode and interval

Description

During AXISMOVE commands, the device continuously sends the current position and speed. The AXISPOSRATE command configures the interval, at which these updates are sent.

Syntax

Command

AXISPOSRATE˽*«AxisName»*	Query current parameters.
AXISPOSRATE˽*«AxisName»*˽*«Interval»*	Set new parameters.

Responses

AXISPOSRATE˽A˽*«AxisName»*˽*«Interval»*	Current interval of axis.
AXISPOSRATE˽A	New parameters stored successfully.
AXISPOSRATE˽I	Command understood but currently not executable (device is actually in menu or input mode).
AXISPOSRATE˽L	Command understood but not executable (incor-rect or no parameter).

Parameters / Return values

Name	Type	Values	Meaning
AxisName	string		Name of the axis (Example “X”, “Y”, “Z”, or other, as given by the AXISLIST command)
Interval	float	0.0 - 100	Sample rate in values per second (Hz). If value is 0, no updates are sent.

Comments

- To disable the regular transmission of values during axis movement, set Interval to 0.
- Typical, device dependent default value is 5 Hz.

Examples

↓	AXISPOSRATE˽X˽2	Set the sample rate for the position of the axis to 2 Hz (samples per second)
↑	AXISPOSRATE˽A	Command executed successfully

↓	AXISPOSRATE˽X	Query parameters.
↑	AXISPOSRATE˽A˽X˽2	Axis X sample rate is 2 Hz (samples per second)

↓	AXISPOSRATE˽X˽0	Disable auto-send of position.
↑	AXISPOSRATE˽A˽X˽0	Command executed successfully

11 KCP commands – Level 2 (model-specific features)

The commands from Level 2 are available for certain instruments.

Attention: In future versions of KCP, these commands are split into more categories.

`PCTW`	Percent weighing: Query/set 100% weight
`PW`	Piece counting: Query/set piece weight
`SIM`	Set mode of indication (Peak or track mode)
`SMEM`	Read measurement memory / reports

`PCTW` – Percent weighing: Query/set 100% weight

Description

Use this command to set or query the reference value for percent weighing.

Syntax

Command

`PCTW`	Queries the weight corresponding to 100%.
`PCTW˽«WeightValue»˽«Unit»`	Sets the weight corresponding to 100%.

Responses

`PCTW˽A˽«WeightValue»˽«Unit»`	Current 100% reference weight with unit.
`PCTW˽A`	100% reference weight is set.
`PCTW˽I`	Command understood but currently not executable.
`PCTW˽L`	Command understood but not executable (incorrect weight).

Parameters / Return values

Name	Type	Values	Meaning
`WeightValue`	float		100% reference weight – numerical value
`Unit`	string		100% reference weight – unit string

Comments

The balance automatically chances to the percent weighing mode.

Examples

↓	`PCTW`	
↑	`PCTW˽A˽100.00˽g`	The current 100% weight value is 100.00 g.

↓	`PCTW˽100.00˽g`	Set the current 100% weight value to 100.00 g.
↑	`PCTW˽A`	OK

`PW` – Piece counting: Query / set piece weight

Description

Use this command to set or query the piece weight value for piece counting.

Syntax

Command

`PW`	Queries the current piece weight.
`PW˽«WeightValue»˽«Unit»`	Sets the current piece weight.

Responses

`PW˽A˽«WeightValue»˽«Unit»`	Returns the current piece weight.
`PW˽A`	Current piece weight is set.
`PW˽I`	Command understood but currently not executable.
`PW˽L`	Command understood but not executable (incorrect weight).

Parameters / Return values

Name	Type	Values	Meaning
`WeightValue`	float		Piece weight – numerical value
`Unit`	string		Piece weight – unit string

Comments

The balance automatically chances to the piece counting mode.

Examples

↓	`PW`	
↑	`PW˽A˽1.2345˽g`	The current piece weight is 100.00 g.

↓	`PW˽1.2345˽g`	Set the current piece weight to 100.00 g.
↑	`PW˽A`	OK

SIM – Set mode of indication (peak or track mode)

Description

Queries or sets the current mode of indication and resets the current peak value.

Syntax

Command

`SIM`	Query current mode of indication.
`SIM˽`*«Mode»*	Set current mode of indication and reset the current peak value.

Responses

`SIM˽`*«Mode»*	Current mode of indication.
`SIM˽A`	Mode successfully set, current peak value is zero.
`SIM˽I`	Invalid mode.

Parameters / Return values

Name	Type	Values	Meaning
`Mode`	string	T	Track mode: indicate the current measurand
		P	Peak mode: only indicate the largest value +/-
		P+	Peak positive mode: only indicate the largest pos. value
		P-	Peak negative mode: only indicate the largest neg. value

Examples

↓	`SIM`	Query current mode of indication.
↑	`SIM˽T`	Current mode of indication is track mode.
↑	`SIM˽P+`	Current mode of indication is peak positive mode.

↓	`SIM˽P+`	Set current mode of indication to peak positive.
↑	`SIM˽A`	OK, peak value reset.

↓	`SIM˽XYZ`	Invalid mode.
↑	`SIM˽I`	Error.

`SMEM` – Read measurement memory / reports

Description

Sends all available recorded data in a unspecified tabular form (separated by spaces).

Syntax

Command

`SMEM`	Request recorded data.

Responses

`SMEM˽A˽START`
`<header line>`
`<data line 1>`
`<data line 2>`
`<data line 3>`
`SMEM˽A˽END`
	For better human readability, the data values in the columns are right or left-aligned, depending on the field type.
	Values/data strings with spaces shall be quoted/escaped as defined.

English

Examples

↓	`SMEM`	

↑	`SMEM A START`
	`Number Date       Time     Mode  Indication`
	`     1 2016-01-13 12:34:56 T      12.3456 N`
	`     2 2016-02-22 12:37:15 P+     12.3456 kN`
	`     3 2016-03-31 12:39:41 P-    -1234.56 N`
	`SMEM A END`	

12 KCP commands – category “Service”

Not publically documented.

13 KCP command index

This page lists all the page numbers, relevant information to a command can be found.

@

@ 10

C

C3 47

D

D 20
DM 21
DW 22

E

ETLS 54

I

I0 11
I1 12
I2 14
I3 15
I4 16
I5 17
I54 48
IBIS 16
IBMT 14
IBRL 38
IBRT 39

J

JDF 64
JG 58
JNA 60
JNG 62
JNK 61
JNM 63
JS 59
JZ 57

K

K 24
KCPC 13
KCPV 12

S

S 5, 28
SI 5, 29
SIM 5, 68
SIR 30
SJ 56
SJR 56
SMEM 5, 69
SR 41

T

T 31
TA 43
TAC 45
TAI 44
TI 32
TZ 33

U

U 5, 34

Z

Z 5, 35
ZI 36