

TENSORMETER TCP-Commands

1 Table of Content

| | | |
|------|---|----|
| 2 | General Description | 2 |
| 2.1 | Structure of a command sending via TCP | 2 |
| 2.2 | Command list | 3 |
| 3 | Commands..... | 4 |
| 3.1 | All Data [alld]..... | 4 |
| 3.2 | New Data [newd] | 7 |
| 3.3 | Select Channels [selc] | 8 |
| 3.4 | Clear Data [cldt] | 9 |
| 3.5 | Averaging Time [avgt]..... | 10 |
| 3.6 | Lock in Frequency [lfrq] | 11 |
| 3.7 | Voltage Amplitude [vamp]..... | 12 |
| 3.8 | Current Amplitude [camp] | 13 |
| 3.9 | Voltage DC [vodc]..... | 14 |
| 3.10 | Current DC [cudc]..... | 15 |
| 3.11 | Voltage Input Range [virg, viru, vird] | 16 |
| 3.12 | Voltage Output Range [vorg, voru, vord] | 17 |
| 3.13 | Current Range [crng, crup, crdn] | 18 |
| 3.14 | Series Resistance [sres, srup, srdn]..... | 19 |
| 3.15 | Voltage Protection [vpro] | 20 |
| 3.16 | Current Protection [cpro]..... | 21 |
| 3.17 | Switch Task [swit] | 22 |
| 3.18 | Analysis Mode [amod; mod?] | 25 |
| 3.19 | Control Mode [cmod] | 26 |
| 3.20 | Measure [meas] | 27 |
| 3.21 | Trigger [tcai, trig, trmo]..... | 28 |
| 3.22 | Pulse [puls, puar] | 30 |
| 3.23 | Reference Input [refe] | 32 |
| 3.24 | Set TCP-Data-Format [tcpa, tcpb] | 33 |
| 3.25 | Get All-Server-Settings [gass] | 34 |
| 3.26 | Exit [exit] | 35 |
| 3.27 | Identify [*IDN?]..... | 36 |
| 3.28 | Auto Update [auup]..... | 37 |

2 General Description

The whole data communication works via TCP/IP protocol.

The byte order is Big Endian.

The data are sent as **binary string** using the ASCII table and the extended ASCII table [Code Page 1252 Windows Latin 1 (ANSI)].

It is also possible to send data as **ASCII string** (i.e. clear text). This will require more TCP bandwidth and does not feature the fixed word length offered by binary. See Section 3.24 for more details.

All data are in SI units without prefix.

2.1 Structure of a TCP command

The commands are always concatenated strings of the following form:

[Length]+[Command]+[Data]

[Length] - 4 bytes, representing the l32 number of the byte length of [Command] + [Data]

[Command] - 4 bytes, the alphabetical command string

[Data] - Could represent anything like: no data, integers, doubles, strings, arrays...

| | [Length] | [Command] | [Data] |
|-----------------------------|---|-------------|-------------------------|
| Number of bytes | 4 Byte | 4 Byte | Zero to Many Bytes |
| Data Type | Integer (l32) | string | Command Dependent |
| Example | 12 | vamp | 1.243 (double) |
| Binary form (hex) | 00 00 00 0C | 76 61 6D 70 | 3F F3 E3 53 F7 CE D9 17 |
| Concatenated (hex) | 00 00 00 0C 76 61 6D 70 3F F3 E3 53 F7 CE D9 17 | | |
| Binary form (string) | vamp?óãS÷îù (-> unprintable characters) | | |

For better understanding, the binary numbers in the following command examples are given in the hexadecimal representation to avoid unprintable characters.

2.2 Command list

| | |
|---------------------------|--|
| [alld] | Send all data from the data array to the client |
| [amod] | Change the analysis mode |
| [avgt] | Send the averaging time |
| [auup] | Set the Auto Update On/OFF |
| [camp] | Set the current amplitude |
| [cldt] | Delete all data in the data array |
| [cmod] | Change the control mode |
| [cpro] | Set the value for current protection |
| [crdn] | Set the current range one level down |
| [crng] | Set the current range |
| [crup] | Set the current range one level up |
| [cudc] | Set the current DC value |
| [exit] | Terminates the server |
| [gass] | Request for all server settings |
| [lfrq] | Set a new value for the lock-in-frequency |
| [meas] | Start a measurement and set the no. of points to measure |
| [mod?] | If analysis mode "Auto", [mod?] sent the detected mode |
| [newd] | Send all new data to the client |
| [puar] | Send the pulse defining array |
| [puls] | Start a pulse/burst |
| [refe] | Set the reference input port On/OFF |
| [selc] | Set the channels the server will send to client |
| [srdn] | Set the series resistance one level down |
| [sres] | Set the series resistance |
| [srup] | Set the series resistance one level up |
| [swit] | Send a new array of switch states |
| [tcpa] | Set the TCP-Data-Format to ASCII |
| [tcpb] | Set the TCP-Data-Format to binary |
| [tcai] | Set the trigger connector as input |
| [trig] | Send a software trigger |
| [trmo] | Set the trigger Mode |
| [vamp] | Set the voltage amplitude |
| [vird] | Set the voltage input range one level down |
| [virg] | Set the voltage input range |
| [viru] | Set the voltage input range one level up |
| [vodc] | Set the voltage DC value |
| [vord] | Set the voltage output range one level down |
| [vorg] | Set the voltage output range |
| [voru] | Set the voltage output range one level up |
| [vpro] | Set the value for voltage protection |
| [*IDN?] | Returns a string that uniquely identifies the Device. |

3 Commands

3.1 All Data [alld]

If the Server receives the all data command, it will send the whole data array to the client.

Command: [alld]

Data type: two dimensional array of double

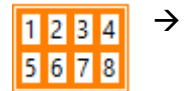
Data unit: column dependent (see [Column Value](#) below)

Whole TCP command:

| | | The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | |
|-----|-------------|---|----------------|-------------------------|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 04 | 61 6c 6c 64 | | 00 00 00 04 61 6c 6c 64 |

TCP feedback from server:

For example, the server sends an array with two rows and four columns, like this



| | | The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | |
|-----|-------------|---|----------------|---------------------------|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 4C | 61 6c 6c 64 | * | 00 00 00 4C 61 6c 6c 64 * |

* 00 00 00 02 00 00 00 04 3F F0 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00 40 08 00 00 00 00 00 00 00 40 10 00 00 00 00 00 00 00 40 14 00 00 00 00 00 00 00 40 18 00 00 00 00 00 00 00 40 1C 00 00 00 00 00 00 40 20 00 00 00 00 00 00 00

The whole TCP answer converted in hexadecimal string and for explanation with some separators |

00 00 00 4C | 61 6C 6C 64 | 00 00 00 02 | 00 00 00 04 | 3F F0 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00 | 40 08 00 00 00 00 00 00 | 40 10 00 00 00 00 00 00 | 40 14 00 00 00 00 00 00 | 40 18 00 00 00 00 00 00 | 40 1C 00 00 00 00 00 00 | 40 20 00 00 00 00 00 00

| Byte | Commend | Data type |
|--------|--|-----------|
| 1-4: | Represent the number of bytes (send as binary) in the answer starting from byte fife up to the end, in this case the number is 76 byte | int (I32) |
| 5-8: | Represent the command string with 4 characters (compare ASCII table) → alld | char |
| 9-12: | Represent the number of rows in the array | int (I32) |
| 13-16: | Represent the number of columns in the array | int (I32) |
| 17-48: | Represent the first row in the array (4 values with 8 bytes each) | double |
| 49-80: | Represent the second row in the array (4 values with 8 bytes each) | double |

The Values in the columns are the following:

| Time | Resistance | Current | Voltage | ... |
|-------------------------|-----------------|----------------|----------------|-----|
| 3601614296.275493620000 | -2.478374630472 | 0.000000351907 | 0.000000913021 | ... |
| 3601614297.275152210000 | 3.116247901954 | 0.000000351345 | 0.000001186151 | ... |
| 3601614298.276810640000 | -0.485871155480 | 0.000000352307 | 0.000000920412 | ... |

Column Value

| | |
|----|---|
| 1 | Time (time zone-independent number of seconds that have elapsed since 12:00 a.m. [midnight], Friday, January 1, 1904, Universal Time [01-01-1904 00:00:00].) |
| 2 | Resistance (Ohm) |
| 3 | Current-AC (A) |
| 4 | Voltage-Output-AC (V) |
| 5 | Voltage-Input-AC (V) |
| 6 | Current-DC (A) |
| 7 | Voltage-Output- DC (V) |
| 8 | Voltage-Input-DC (V) |
| 9 | Long H0. (Ohm) |
| 10 | Long H1 Re (Ohm) |
| 11 | Long H1 Im (Ohm) |
| 12 | Long H2 Re (Ohm) |
| 13 | Long H2 Im (Ohm) |
| 14 | Long H3 Re (Ohm) |
| 15 | Long H3 Im (Ohm) |
| 16 | Trans H0 (Ohm) |
| 17 | Trans H1 Re (Ohm) |
| 18 | Trans H1 Im (Ohm) |
| 19 | Trans H2 Re (Ohm) |
| 20 | Trans H2 Im (Ohm) |
| 21 | Trans H3 Re (Ohm) |
| 22 | Trans H3 Im (Ohm) |
| 23 | Switch state (U32) |
| 24 | Lock in Frequency (Hz) |
| 25 | Voltage Amplitude Setpoint (V) |
| 26 | Voltage DC Setpoint (V) |
| 27 | Current Amplitude Setpoint (A) |
| 28 | Current DC Setpoint (A) |
| 29 | Voltage Input Range (V) |
| 30 | Voltage Output Range (V) |
| 31 | Current Range (A) |
| 32 | Series Resistance (Ohm) |
| 33 | Input peak voltage Ch0 (V) |
| 34 | Input peak voltage Ch1 (V) |
| 35 | Input peak voltage Ch2 (V) |

- 36 Input peak voltage Ch3 (V)
- 37 Voltage protection (V)
- 38 Current protection (A)
- 39 Analysis Mode
- 40 Duration Waveform Segment (V)
- 41 LockQuality

3.2 New Data [newd]

The server sends only the data not yet sent. That means only the last rows of the data array will be sent. This is the standard acquisition mode for the TENSORMETER.

Command: [newd]

Data type: two dimensional array of double

Data unit: column dependent (see All Data [alld] command)

Whole TCP command:

| | | | | |
|------------|---------------|---|-----------------------|--------------------------|
| | | The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | |
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 04 | 6E 65 77 64 | | 00 00 00 04 6E 65 77 64 |

TCP feedback from server:

The format is equal to the [alld] command, but the response includes only those data lines which were acquired since the last New Data command.

3.3 Select Channels [selc]

With the [selc] command you can select the channels (columns of the data array) you will receive from the New Data command.

Command: [selc]

Data type: one dimensional array of integers (I32)

Data unit: no unit

Whole TCP command:

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|---|---|
| HEX | 00 00 00 14 | 73 65 6C 63 | 00 00 00 03 00 00 00 03 00 00 00 00 00 00 00 02 | 00 00 00 14 73 65 6C 63 00 00 00 03 00 00 00 03 00 00 00 00 00 00 00 02 |

The 4 Byte length corresponds to the sum of all Bytes from Command + Data

3
0
2

Example Data =

The whole TCP command converted in hexadecimal string:

00 00 00 14 73 65 6C 63 00 00 00 03 00 00 00 03 00 00 00 00 00 02

| Byte | Command | Data type |
|--------|--|-----------|
| 1-4: | Represent the number of bytes (send as binary) starting from byte five up to the end, in this case the number is 20 byte | int (I32) |
| 5-8: | Represent the command string with 4 characters (compare ASCII table) → selc | char |
| 9-12: | Represent the number of elements in the array | int (I32) |
| 13-16: | Represent the index of the first desired data column | int (I32) |
| 17-20: | Represent the index of the second desired data column | int (I32) |
| 21-24: | Represent the index of the third desired data column | int (I32) |

The server sends a feedback in the same format. If a value is outside the limits, a coerced value is returned.

This command is only used from the client side. When a client connects to the server, the server provides all data channels in ascending order by default. Using the Select Channels command, the client can define an array with any number of data column it likes, in arbitrary order.

3.4 Clear Data [cldt]

Delete all data (data array) at server side.

Command: [cldt]

Data type: no data

Data unit: no unit

Whole TCP command:

| |
|---|
| The 4 Byte length corresponds to the sum of all Bytes from Command + Data |
|---|

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------|-------------------------|
| HEX | 00 00 00 04 | 63 6C 64 74 | | 00 00 00 04 63 6C 64 74 |

The server sends a feedback in the same format.

3.5 Averaging Time [avgt]

Set the time between two switching operations. The value is sent in seconds.

Command: [avgt]

Data type: double

Data unit: Seconds [s]

Whole TCP command:

| | | The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | |
|-----|-------------|---|----------------------------|--|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 0C | 61 76 67 74 | 3F E0 00 00 00 00 00 00 | 00 00 00 08 6E 65 77 64 3F E0 00 00 00 00 00 00 |

Example Data = 0.5

The server sends a feedback in the same format. If the value is outside the limits, a coerced value is returned.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

3.6 Lockin Frequency [lfrq]

Set the lockin frequency in Hz.

Command: [lfrq]

Data type: double

Data unit: Hertz [Hz]

Whole TCP command:

| The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | | | |
|---|-------------|-------------|----------------------------|--|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 0C | 6C 66 72 71 | 40 36 80 00 00 00 00 00 | 00 00 00 0C 6C 66 72 71 40 36 80 00 00 00 00 00 |

Example Data = 22.5

The server sends a feedback in the same format. If the value is outside the limits, a coerced value is returned.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

3.7 Voltage Amplitude [vamp]

Set the voltage amplitude in V.

Command: [vamp]

Data type: double

Data unit: Voltage [V]

Whole TCP command:

| | | The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | |
|-----|-------------|---|----------------------------|--|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 0C | 76 61 6D 70 | 40 1D 4B C6 A7 EF 9D B2 | 00 00 00 0C 76 61 6D 70 40 1D 4B C6 A7 EF 9D B2 |

Example Data = 7.324

The server sends a feedback in the same format. If the value is outside the limits, a coerced value is returned.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

3.8 Current Amplitude [camp]

Set the current amplitude in A.

Command: [camp]

Data type: double

Data unit: Ampere [A]

Whole TCP command:

The 4 Byte length corresponds to the sum of all Bytes from Command + Data

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------------------|--|
| HEX | 00 00 00 0C | 63 61 6D 70 | 3F 60 62 4D D2 F1 A9 FC | 00 00 00 0C 63 61 6D 70 3F 60 62 4D D2 F1 A9 FC |

Example Data = 0.002

The server sends a feedback in the same format. If the value is outside the limits, a coerced value is returned.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

3.9 Voltage DC [vodc]

Set the voltage DC level in V.

Command: [vodc]

Data type: double

Data unit: Voltage [V]

Whole TCP command:

| | | The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | |
|-----|-------------|---|----------------------------|--|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 0C | 76 6F 64 63 | 40 1D 4B C6 A7 EF 9D B2 | 00 00 00 0C 76 6F 64 63 40 1D 4B C6 A7 EF 9D B2 |

Example Data = 7.324

The server sends a feedback in the same format. If the value is outside the limits, a coerced value is returned.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

3.10 Current DC [cudc]

Set the current DC level in A.

Command: [cudc]

Data type: double

Data unit: Ampere [A]

Whole TCP command:

The 4 Byte length corresponds to the sum of all Bytes from Command + Data

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------------------|--|
| HEX | 00 00 00 0C | 63 75 64 63 | 3F 60 62 4D D2 F1 A9 FC | 00 00 00 0C 63 75 64 63 3F 60 62 4D D2 F1 A9 FC |

Example Data = 0.002

The server sends a feedback in the same format. If the value is outside the limits, a coerced value is returned.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

3.11 Voltage Input Range [virg, viru, vird]

[virg] set the value for the voltage input range. Every positive value is possible. It will be assigned to the next higher range or will be coerced. If you send a zero (0.0) or negative value, the server goes into the Auto-Range-Mode. In the Auto-Range-Mode the server sends the ranges with a negative sign at the front like -0.5. To come back to the Standard-Mode just send a positive value for the range. (e.g. 0.8)

[viru] set the value for the voltage input range one level up.

[vird] set the value for the voltage input range one level down.

Commands: [virg]

[viru]

[vird]

Data type: [virg] double

[viru] no data

[vird] no data

Data unit: Voltage [V]

Whole TCP command for [virg]:

| The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | | | |
|---|-------------|-------------|----------------------------|--|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 0C | 76 69 72 67 | 40 00 00 00 00 00 00 00 | 00 00 00 0C 76 69 72 67 40 00 00 00 00 00 00 00 |

Example Data = 2

The server sends a feedback in the same format.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

Whole TCP command for [viru]:

| The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | | | |
|---|-------------|-------------|----------------|-------------------------|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 04 | 76 69 72 75 | | 00 00 00 04 76 69 72 75 |

The server sends a feedback in the format of the [virg] comand.

Whole TCP command for [vird]:

| The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | | | |
|---|-------------|-------------|----------------|-------------------------|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 04 | 76 69 72 64 | | 00 00 00 04 76 69 72 64 |

The server sends a feedback in the format of the [virg] command.

3.12 Voltage Output Range [vorg, voru, vord]

[vorg] set the value for the voltage output range. Every positive value is possible. It will be assigned to the next higher range or will be coerced. If you send a zero (0.0) or negative value, the server goes into the Auto-Range-Mode. In the Auto-Range-Mode the server sends the ranges with a negative sign at the front like -0.5. To come back to the Standard-Mode just send a positive value for the range. (e.g. 0.8)

[voru] set the value for the voltage output range one level up.

[vord] set the value for the voltage output range one level down.

Commands: [vorg]

[voru]

[vord]

Data type: [vorg] double

[voru] no data

[vord] no data

Data unit: Voltage [V]

Whole TCP command for [vorg]:

| The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | | | |
|---|-------------|-------------|----------------------------|---|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 0C | 76 6F 72 67 | 40 00 00 00 00 00 00 00 | 00 00 00 0C 76 6F 72 67 40 00 00 00 00 00 00 00 00 |

Example Data = 2

The server sends a feedback in the same format.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

Whole TCP command for [voru]:

| The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | | | |
|---|-------------|-------------|----------------|-------------------------|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 04 | 76 6F 72 75 | | 00 00 00 04 76 6F 72 75 |

The server sends a feedback in the format of the [vorg] command.

Whole TCP command for [vord]:

| The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | | | |
|---|-------------|-------------|----------------|-------------------------|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 04 | 76 6F 72 64 | | 00 00 00 04 76 6F 72 64 |

The server sends a feedback in the format of the [vorg] command.

3.13 Current Range [crng, crup, crdn]

[crng] set the value for the current range. Every positive value is possible. It will be assigned to the next higher range or will be coerced. If you send a zero (0.0) or negative value, the server goes into the Auto-Range-Mode. In the Auto-Range-Mode the server sends the ranges with a negative sign at the front like -0.5. To come back to the Standard-Mode just send a positive value for the range. (e.g. 0.8)

[crup] set the value for the current range one level up.

[crdn] set the value for the current range one level down.

Commands: [crng]

[crup]

[crdn]

Data type: [crng] double

[crup] no data

[crdn] no data

Data unit: Ampere [A]

Whole TCP command for [crng]:

| | | | | The 4 Byte length corresponds to the sum of all Bytes from Command + Data |
|-----|-------------|-------------|----------------------------|---|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 0C | 63 72 6E 67 | 3F B9 99 99 99 99 99 9A | 00 00 00 0C 63 72 6E 67 3F B9 99 99 99 99 99 9A |

Example Data = 0.1

The server sends a feedback in the same format.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

Whole TCP command for [crup]:

| | | | | The 4 Byte length corresponds to the sum of all Bytes from Command + Data |
|-----|-------------|-------------|----------------|---|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 04 | 63 72 75 70 | | 00 00 00 04 63 72 75 70 |

The server sends a feedback in the format of the [crng] command.

Whole TCP command for [crdn]:

| | | | | The 4 Byte length corresponds to the sum of all Bytes from Command + Data |
|-----|-------------|-------------|----------------|---|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 04 | 63 72 64 6E | | 00 00 00 04 63 72 64 6E |

The server sends a feedback in the format of the [crng] command.

3.14 Series Resistance [sres, srup, srdn]

[sres] set the value for the series resistance. Every positive value is possible. It will be assigned to the next higher resistance or will be coerced. If you send a zero (0.0) or negative value, the server goes into the Auto-Range-Mode. In the Auto-Range-Mode the server sends the ranges with a negative sign at the front like -0.5. To come back to the Standard-Mode just send a positive value for the range. (e.g. 0.8)

[srup] set the value for the series resistance one level up.

[srdn] set the value for the series resistance one level down.

Commands: [sres]

[srup]

[srdn]

Data type: [sres] double

[srup] no data

[srdn] no data

Data unit: Ohm [Ω]

Whole TCP command for [sres]:

| The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | | | |
|---|-------------|-------------|----------------------------|---|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 0C | 73 72 65 73 | 40 10 00 00 00 00 00 00 | 00 00 00 0C 73 72 65 73 40 10 00 00 00 00 00 00 00 |

Example Data = 4

The server sends a feedback in the same format.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

Whole TCP command for [srup]:

| The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | | | |
|---|-------------|-------------|----------------|-------------------------|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 04 | 73 72 75 70 | | 00 00 00 04 73 72 75 70 |

The server sends a feedback in the format of the [srng] command.

Whole TCP command for [srdn]:

| The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | | | |
|---|-------------|-------------|----------------|-------------------------|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 04 | 73 72 64 6E | | 00 00 00 04 73 72 64 6E |

The server sends a feedback in the format of the [srng] command.

3.15 Voltage Protection [vpro]

Set the voltage protection level in V.

Command: [vpro]

Data type: double

Data unit: Voltage [V]

Whole TCP command:

The 4 Byte length corresponds to the sum of all Bytes from Command + Data

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------------------|--|
| HEX | 00 00 00 0C | 76 70 72 6F | 40 1D 4B C6 A7 EF 9D B2 | 00 00 00 0C 76 70 72 6F 40 1D 4B C6 A7 EF 9D B2 |

Example Data = 7.324

The server sends a feedback in the same format. If the value is outside the limits, a coerced value is returned.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

3.16 Current Protection [cpro]

Set the current protection level in A.

Command: [cpro]

Data type: double

Data unit: Ampere [A]

Whole TCP command:

The 4 Byte length corresponds to the sum of all Bytes from Command + Data

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------------------|--|
| HEX | 00 00 00 0C | 63 70 72 6F | 3F 60 62 4D D2 F1 A9 FC | 00 00 00 0C 63 70 72 6F 3F 60 62 4D D2 F1 A9 FC |

Example Data = 0.002

The server sends a feedback in the same format. If the value is outside the limits, a coerced value is returned.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

3.17 Switch Task [swit]

Switch task sends a one dimensional array of U32 values. Each value represents a switch state of the TENSORMETER.

Command: [swit]

Data type: one dimensional array of uint (U32)

Data unit: no unit

Whole TCP command:

For example, sending an array with two values, like this →



The 4 Byte length corresponds to the sum of all Bytes from Command + Data

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|--|--|
| HEX | 00 00 00 10 | 73 77 69 74 | 00 00 00 02 00 00 00 00 00 00 00 01 | 00 00 00 10 73 77 69 74 00 00 00 02 00 00 00 00 00 00 01 |

The whole TCP command converted in hexadecimal string:

00 00 00 10 73 77 69 74 00 00 00 02 00 00 00 00 00 00 01

| Byte | Comment | Data type |
|--------|--|------------|
| 1-4: | Represent the number of bytes (send as binary) starting from byte five up to the end, in this case the number is 16 byte | int (I32) |
| 5-8: | Represent the command string with 4 characters (compare ASCII table) → swit | char |
| 9-12: | Represent the number of elements in the array | int (I32) |
| 13-16: | Represent the first state of the array (32 switch bits = 4 bytes) | uint (U32) |
| 17-20: | Represent the second state of the array (32 switch bits = 4 bytes) | uint (U32) |

The server sends a feedback in the same format.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

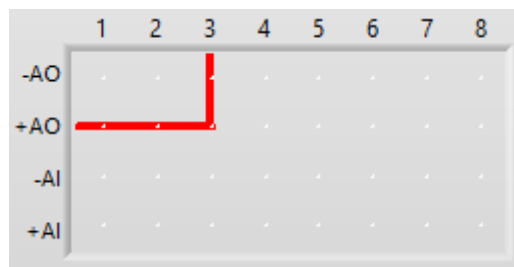
The switch state is calculated in the following way: → see next pages

The switch state is a 32 bit unsigned integer value. Each bit represents a possible connection from an internal analog input or analog output to a BNC connector at the front panel of the TENSORMETER.

For example if the bit number 10 (Bit 10) is high, the BNC number 3 (BNC 3) connector is connected with the internal plus analog output (+AO) of the TENSORMETER.

| | BNC 1 | BNC 2 | BNC 3 | BNC 4 | BNC 5 | BNC 6 | BNC 7 | BNC 8 |
|------------|-------|-------|--------|--------|--------|--------|--------|--------|
| -AO | Bit 1 | Bit 5 | Bit 9 | Bit 13 | Bit 17 | Bit 21 | Bit 25 | Bit 29 |
| +AO | Bit 2 | Bit 6 | Bit 10 | Bit 14 | Bit 18 | Bit 22 | Bit 26 | Bit 30 |
| -AI | Bit 3 | Bit 7 | Bit 11 | Bit 15 | Bit 19 | Bit 23 | Bit 27 | Bit 31 |
| +AI | Bit 4 | Bit 8 | Bit 12 | Bit 16 | Bit 20 | Bit 24 | Bit 28 | Bit 32 |

BNC = BNC connector at the front panel of the TENSORMETER; AO = analog output; AI = analog input

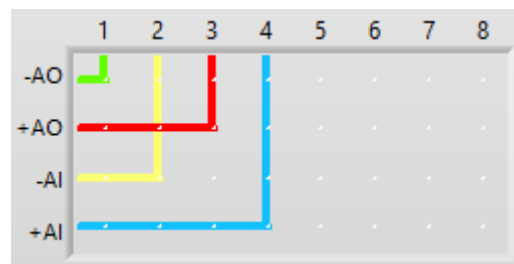


If only the bit number 10 (Bit 10) is high the switch state value (U32) will be 512.

| Bit 32 - 29 (BNC 8) | Bit 28 - 25 (BNC 7) | Bit 24 - 21 (BNC 6) | Bit 20 - 17 (BNC 5) | Bit 16 - 13 (BNC 4) | Bit 12 - 9 (BNC 3) | Bit 8 - 5 (BNC 2) | Bit 4 - 1 (BNC 1) |
|---|------------------------|------------------------|------------------------|------------------------|-----------------------|----------------------|----------------------|
| 0000 | 0000 | 0000 | 0000 | 0000 | 0010 | 0000 | 0000 |
| 0000 0000 0000 0000 0000 0010 0000 0000 | | | | | | | (binary) |
| 00 00 02 00 | | | | | | | (hex) |
| 512 | | | | | | | (dec) |

Second example: If the bits with the number 1; 7; 10 and 16 are high, the switch state value (U32) is 33345.

| | BNC 1 | BNC 2 | BNC 3 | BNC 4 | BNC 5 | BNC 6 | BNC 7 | BNC 8 |
|------------|-------|-------|--------|--------|--------|--------|--------|--------|
| -AO | Bit 1 | Bit 5 | Bit 9 | Bit 13 | Bit 17 | Bit 21 | Bit 25 | Bit 29 |
| +AO | Bit 2 | Bit 6 | Bit 10 | Bit 14 | Bit 18 | Bit 22 | Bit 26 | Bit 30 |
| -AI | Bit 3 | Bit 7 | Bit 11 | Bit 15 | Bit 19 | Bit 23 | Bit 27 | Bit 31 |
| +AI | Bit 4 | Bit 8 | Bit 12 | Bit 16 | Bit 20 | Bit 24 | Bit 28 | Bit 32 |



| Bit 32 - 29 (BNC 8) | Bit 28 - 25 (BNC 7) | Bit 24 - 21 (BNC 6) | Bit 20 - 17 (BNC 5) | Bit 16 - 13 (BNC 4) | Bit 12 - 9 (BNC 3) | Bit 8 - 5 (BNC 2) | Bit 4 - 1 (BNC 1) |
|---|------------------------|------------------------|------------------------|------------------------|-----------------------|----------------------|----------------------|
| 0000 | 0000 | 0000 | 0000 | 1000 | 0010 | 0100 | 0001 |
| 0000 0000 0000 0000 1000 0010 0100 0001 | | | | | | | (binary) |
| 00 00 82 41 | | | | | | | (hex) |
| 33345 | | | | | | | (dec) |

3.18 Analysis Mode [amod; mod?]

Set the value for analysis mode. If the analysis mode is 0 (Auto) the server sends the detected mode via the [mod?] command. (e.g. Client receives *mod?3* , that means the detected analysis mode on the server is *Van der Pauw*).

Commands: [amod]
[mod?] (only sent from server)

Data type: unsigned integer (U16)

Data unit: no unit

Modes: 0 → Auto
1 → Kelvin
2 → Zero-Offset-Hall
3 → Van-der-Pauw
4 → Ratiometric
5 → Differential

Whole TCP command:

The 4 Byte length corresponds to the sum of all Bytes from Command + Data

| | Length | Command | Data (example) | Whole TCP command |
|------------|-------------|-------------|----------------|-------------------------------|
| HEX | 00 00 00 06 | 61 6D 6F 64 | 00 02 | 00 00 00 06 61 6D 6F 64 00 02 |

Example Data = 2 (Sets the Zero-Offset-Hall mode)

The server sends a feedback in the same format.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

3.19 Control Mode [cmod]

Set the value for the control mode.

Commands: [cmod]

Data type: unsigned integer (U16)

Data unit: no unit

Modes: 0 → No Protection

1 → Protect Overvoltage/Overcurrent

Whole TCP command:

| | | The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | |
|-----|-------------|---|----------------|-------------------------------|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 06 | 63 6D 6F 64 | 00 01 | 00 00 00 06 63 6D 6F 64 00 01 |

Example Data = 1 (Sets the "Protect Overvoltage/Overcurrent" mode)

The server sends a feedback in the same format.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

3.20 Measure [meas]

Start a measurement and set the number of points to measure before going into idle state. When idling, the voltage output will remain on, the switch state will remain in the most recent configuration, but the input signal is ignored.

Commands: [meas]

Data type: signed integer (I32)

Data unit: no unit

Whole TCP command:

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------|--|
| HEX | 00 00 00 08 | 6D 65 61 73 | 00 00 00 02 | 00 00 00 06 6D 65 61 73 00 00 00 02 |

The 4 Byte length corresponds to the sum of all Bytes from Command + Data

Example Data = 2

The server sends a feedback in the same format.

A value of -1 will never decay, indicating continuous measurements. This is also the default state of the device.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

3.21 Trigger [tcai, trig, trmo]

[tcai] "trigger-connector-as-input" sets the BNC Trigger connector to output trigger signals when set to true (1). The default state is false (0), which set the BNC Trigger connector as input. [trmo] set the value for the hardware trigger mode and [trig] sends a software trigger.

Commands: [tcai]

[trmo]

[trig]

Data type: [tcai] boolean

[trmo] unsigned integer (U16)

[trig] no data

Data unit: no unit

Modes:

| Mode | Description | Start conditions | Stop conditions | Stop by Trigger | Stop by averaging time |
|------|---|------------------|-----------------|-----------------|------------------------|
| 1 | [START]: switch from low to high [STOP]: switch from high to low or after average time | | | | |
| 2 | [START]: switch from high to low [STOP]: switch from low to high or after average time | | | | |
| 3 | [START]: switch from low to high [STOP]: switch from low to high or after average time | | | | |
| 4 | [START]: switch from high to low [STOP]: switch from high to low or after average time | | | | |
| 5 | [START]: by level change [STOP]: by the next level change (is also the start for the next measurement) or after average time | | | | |
| 6 | [START]: by pulse [STOP]: by the next pulse (is also the start for the next measurement) or after average time | | | | |

Whole TCP command for [tcai]:

The 4 Byte length corresponds to the sum of all Bytes from Command + Data

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------|----------------------------|
| HEX | 00 00 00 05 | 74 63 61 69 | 01 | 00 00 00 05 74 63 61 69 01 |

Example Data = 1

The server sends a feedback in the same format.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

Whole TCP command for [trmo]:

The 4 Byte length corresponds to the sum of all Bytes from Command + Data

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------|-------------------------------|
| HEX | 00 00 00 06 | 74 72 6D 6F | 00 01 | 00 00 00 06 74 72 6D 6F 00 01 |

Example Data = 1

The server sends a feedback in the same format.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

Whole TCP command for [trig]:

The 4 Byte length corresponds to the sum of all Bytes from Command + Data

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------|-------------------------|
| HEX | 00 00 00 04 | 74 72 69 67 | | 00 00 00 04 74 72 69 67 |

There is no feedback to this command.

3.22 Pulse [puls, puar]

The [puar] command defines what to output, once a pulse is triggered. This can be either a single pulse, a burst of pulses or an arbitrary waveform. The [puls] command triggers the output. If the device is currently measuring, the pulse will be queued and output once the measurement is done.

Command: [puls]

[puar]

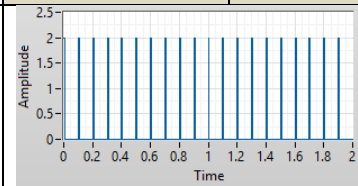
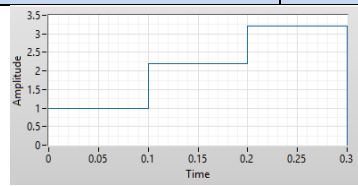
Data type: [puls] no data

[puar] one dimensional array of double

Data unit: mixed

The Array: The Array can represent two different types of descriptions for a Pulse/Burst.

| Array - Element | Mode 1 | Example for Mode 1 | Mode 2 | Example for Mode 2 |
|-----------------|-------------------|--------------------|-------------------|--------------------|
| 0 | mode 1 | 1 | mode 2 | 2 |
| 1 | Number of periods | 20 | Number of periods | 0.1 |
| 2 | phase [°] | 0 | Phase [°] | 0 |
| 3 | DC-Offset [V] | 2 | DC-Offset [V] | 1 |
| 4 | pulse height [V] | 0 | Sampling time [s] | 0.0001 |
| 5 | pulse width [s] | 0.0005 | value 1 [V] | 0 |
| 6 | Period [s] | 0.1 | value 2 [V] | 1.2 |
| 7 | | | value ... [V] | 2.2 |

Whole TCP command for [puls]:

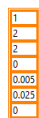
The 4 Byte length corresponds to the sum of all Bytes from Command + Data

| HEX | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------|-------------------------|
| | 00 00 00 04 | 70 75 6C 73 | | 00 00 00 04 70 75 6C 73 |

The server sends a feedback in the same format.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

Whole TCP command for [puar]:



For example, the server send an array, like this →

The 4 Byte length corresponds to the sum of all Bytes from Command + Data

| HEX | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------|---------------------------|
| | 00 00 00 40 | 70 75 61 72 | * | 00 00 00 40 70 75 61 72 * |

*00 00 00 07 3F F0 00 00 00 00 00 00 40 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3F 74 7A E1 47 AE 14 7B 3F 99 99 99 99 99 99 9A 0000 0000 0000 0000

The whole TCP answer converted in hexadecimal string and for explanation with some separators |

00 00 00 40 | 70 75 61 72 | 00 00 00 07 | 3F F0 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00 | 3F 74 7A E1 47 AE 14 7B | 3F 99 99 99 99 99 99 9A | 00 00 00 00 00 00 00 00

| Byte | Comment | Data type |
|---------|--|-----------|
| 1-4: | Represent the number of bytes (send as binary) in the answer starting from byte five up to the end, in this case the number is 64 byte | int (I32) |
| 5-8: | Represent the command string with 4 characters (compare ASCII table)→ puar | char |
| 9-12 | Represent the number of elements in the array | int (I32) |
| 13-20 | Represent the first element in the array | double |
| 21-28 | Represent the second element in the array | double |
| 29..... | and so on | ... |

The server sends a feedback in the same format.

This command can be used bidirectionally. If a client is connected, the client can also listen for this command to update the client user interface when the value was changed at the server side.

3.23 Reference Input [refe]

Set the reference input to ON or OFF (1 or 0). If the parameter is set to ON, the set value for the lockin frequency will be ignored and the device will instead phase lock onto the strongest frequency component present at the Reference Port. When set to OFF, the Reference Port will output a buffered duplicate of the excitation waveform. This can be loaded with up to 10 mA of current and does not influence the measurement.

Command: [refe]

Data type: boolean

Data unit: no unit

The 4 Byte length corresponds to the sum of all Bytes from Command + Data

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------|----------------------------|
| HEX | 00 00 00 05 | 72 65 66 65 | 01 | 00 00 00 05 72 65 66 65 01 |

Example Data = 1

The server sends a feedback in the same format.

3.24 Set TCP-Data-Format [tcpa, tcpb]

[tcpa] command set the TCP-Data-Format to ASCII and [tcpb] set the TCP-Data-Format to binary.

Command: [tcpa]

[tcpb]

Data type: no data

Data unit: no unit

Whole TCP command for [tcpa]:

| | | The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | |
|-----|-------------|---|----------------|-------------------------|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 04 | 74 63 70 61 | | 00 00 00 04 74 63 70 61 |

Whole TCP command for [tcpb]:

| | | The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | |
|-----|-------------|---|----------------|-------------------------|
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 04 | 74 63 70 62 | | 00 00 00 04 74 63 70 62 |

The server sends a feedback in the same format.

3.25 Get-All-Server-Settings [gass]

[gass] command request all server settings. After the [gass] command, the server starts to send all setup parameter.

Command: [gass]

Data type: no data

Data unit: no unit

Whole TCP command for [gass]:

| | | | | |
|------------|---------------|---|-----------------------|--------------------------|
| | | The 4 Byte length corresponds to the sum of all Bytes from Command + Data | | |
| | Length | Command | Data (example) | Whole TCP command |
| HEX | 00 00 00 04 | 67 61 73 73 | | 00 00 00 04 67 61 73 73 |

The server sends a feedback in the same format, followed by all set parameters. The individual parameters are sent in their respective formats.

3.26 Exit [exit]

Terminates the server.

Command: [exit]

Data type: no data

Data unit: no unit

Whole TCP command:

| |
|---|
| The 4 Byte length corresponds to the sum of all Bytes from Command + Data |
|---|

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------|-------------------------|
| HEX | 00 00 00 04 | 65 78 69 74 | | 00 00 00 04 65 78 69 74 |

There is no feedback to this command.

3.27 Identify [*IDN?]

Identify the device.

Command: [*IDN?]

Data type: no data

Data unit: no unit

Whole TCP command:

| |
|---|
| The 4 Byte length corresponds to the sum of all Bytes from Command + Data |
|---|

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------------|----------------|----------------------------|
| HEX | 00 00 00 04 | 2A 49 44 4E 3F | | 00 00 00 04 2A 49 44 4E 3F |

The server returns a string that uniquely identifies the Device.

3.28 Auto Update [auup]

Set the auto update to ON or OFF (1 or 0). If the parameter is set to ON, the server send all changes at the server side to the client. If the auto update is OFF the server answers only of the command sent from the server. Some of the commands generate a multi command answer.

Command: [auup]

Data type: boolean

Data unit: no unit

Whole TCP command:

The 4 Byte length corresponds to the sum of all Bytes from Command + Data

| | Length | Command | Data (example) | Whole TCP command |
|-----|-------------|-------------|----------------|----------------------------|
| HEX | 00 00 00 04 | 61 75 75 70 | 01 | 00 00 00 04 61 75 75 70 01 |

Example Data = 1

The server sends a feedback in the same format.