GASERA ONE AK Protocol

Revision i

Revision history:

Date	Initials	Description
2018-01-02	AE	Examples changed
2018-01-16	AE	Added new SCOR command for updating component order in ACON command response. Description of ACON command updated correspondingly. Updated examples section.
2018-01-17	AE	Updated AK SCOR example.
2018-02-19	AE	Added note on MW version compatibility / support to each AK command description.
2018-05-31	AE	Minor fixes
2018-08-02	AE	Added descriptions of new AK commands STAT and AITR.
2018-08-16	AE	Added description for new SCON command (updated ACON).
2018-10-09	AE	Updated documentation for SCON / ACON / AITR commands (e.g., inlet field added to ACON which can be activated by SCON optional parameter).
2018-12-07	AE	Added new commands ANET / SNET for requesting / setting device network settings.
2018-12-19	AE	Added new command APAR for requesting value of Gasera ONE device parameter.
2019-01-1	AE	Added information about using AK-protocol over serial RS-232 connection.
2019-01-18	AE	Added new AK-command SONL for enabling / disabling of online measurement mode.
2019-03-07	AE	Added new AK-commands ACLK and STUN.
2019-04-24	JR	Document moved to EPDM
2019-05-28	JR	Accept all changes, modify SNET
2019-05-29	TV	Clarifications and small updates
2019-06-26	JF	Example for Realterm added
2019-08-07	JR	SocketTest v 3.0.0 example added
2021-03-28	AE	Added new AK-commands ATSP, ASYP, AMPS, ADEV, STST, and ASTR
2021-04-08	AE	Added new AK-command RDEV
2023-04-05	AE	Added info on new commands TRME and STDB

Document Information
Product name: GASERA ONE
Title: GASERA ONE AK Protocol
Release: 19.4.2023 8.47

Publisher
Gasera Ltd.
Web: http://www.gasera.fi
Email: contact@gasera.fi

Warranty Information
Information presented in this document are not guarantee declarations.

Table of contents

1. AK protocol

1.1. Introduction

GASERA ONE analyzers can be customized to support AK communication protocol. Using this protocol, an external device can communicate with the GASERA ONE. Supported remote actions include start or stop a measurement and request latest measurement results. Analyzer settings can be set only via user interface, see GASERA ONE User manual for details.

All requests are synchronous, so the client device sends an AK request to GASERA ONE and waits for the response. Once, the request is completed, and a response is sent to the client, the GASERA ONE is ready to accept a new request.

Supported commands and their availability in MW versions are defined in this document. MW is the internal MiddleWare of analyzers.

1.2. Connection to GASERA ONE

1.2.1. TCP/IP

The client device can connect to GASERA ONE by either over TCP/IP or RS232 serial connection.

When the client connects to GASERA ONE over TCP/IP, use TCP port 8888 and physical ethernet connection. Once a connection is established, GASERA ONE is ready to accept commands using AK protocol.

For testing, any standard TCP/IP socket client application can be used, for example SocketTest3 from https://sourceforge.net/projects/sockettest/.

1.2.2. RS232

The AK protocol can also be used over serial (RS232) connection (e.g., Pulse OEM models). To use serial connection, a suitable terminal application is required, for example:

- PuTTY (https://www.putty.org)
- RealTerm (https://sourceforge.net/projects/realterm/).

Connect the Gasera ONE using serial USB cable to a computer, and connect using serial settings:

- speed: 19200 bps
- data bits: 8
- stop bits: 1
- no flow control

Serial communication uses the same set of AK protocol format as over TCP/IP.

NOTE: In order to use AK with the demo mode, the analyzer needs to be logged in as Gasera Demo user!

1.3. AK protocol format as used by the GASERA ONE

1) The AK request format (Client to GASERA ONE) is as follow:

   <STX><BLANK><FUNC_BYTE1><FUNC_BYTE2><FUNC_BYTE3><FUNC_BYTE4><BLANK>K<channel><BLANK><DATA_BYTE1><DATA_BYTE2>..<DATA_BYTE_N><ETX> (STX=ASCII 02, ETX=ASCII 03)

2) The AK response format (GASERA ONE to Client) is as follow:

   <STX><BLANK><FUNC_BYTE1><FUNC_BYTE2><FUNC_BYTE3><FUNC_BYTE4><BLANK>
   <error_status_byte><BLANK><DATA_BYTE1><DATA_BYTE2>..<DATA_BYTE_N><ETX> (STX=ASCII 02, ETX=ASCII 03)

   In the above, BLANK refers to single space. The error status byte in the response is set to 0 on success; in case of request processing error the error status is set to 1.

   Supported commands are given in the next section.

1.4. Supported commands

1.4.1. Ask Current Device Status – ASTS

Available from MW ver. 1.3.9

a. format *ASTS* K0 - request the current device status
   i. response *ASTS* <errorstatus> <device_status> (errorstatus: 0=no errors, 1=error)

Possible values for device_status:
- 0 – device initializing
- 1 – initialization error
- 2 – device idle state
- 3 – device self-test in progress
- 4 – malfunction
- 5 – measurement in progress
- 6 – calibration in progress
- 7 – canceling measurement
- 8 – laserscan in progress

1.4.2. Ask Active Errors – AERR

Available from MW ver. 1.3.9

a. format *AERR* K0 - ask the current list of active error codes (empty data part if no errors)
   ii. response (success) *AERR* 0 <err_code1> <err_code2> <err_codeN> ..
   iii. response (request error) *AERR 1*

1.4.3. Ask Task List – ATSK

Available from MW ver. 1.3.9

b. format *ATSK* K0 - request the current list of measurement tasks
   iv. response (success) *ATSK* 0 <taskid> <taskname> <taskid> <taskname> ..
   v. response (error) *ATSK 1*

1.4.4. Start New Measurement – STAM

Available from MW ver. 1.3.9

c. format *STAM* K0 <taskId> - start measurement based on task with specified id (id of tasks is retrieved from response to *Ask Task List* request)
   vi. response *STAM* <errorstatus> (0=no errors, 1=error)

1.4.5. Stop Current Measurement – STPM

Available from MW ver. 1.3.9

d. format *STPM* K0
   vii. response *STPM* <errorstatus> (0=no errors, 1=error)

1.4.6. Get Last Measurement Results (concentrations) – ACON

Available from MW ver. 1.3.9

e. format *ACON* K0
   viii. response (success): *ACON* 0 <timestamp> <cas> <conc_ppm> <timestamp> <cas> <conc_ppm> ..
   ix. response (error) *ACON* 1

If MW database version is 10 or earlier, the timestamp is the time and date values when the measurement result was calculated. If MW database version is newer 10, the timestamp corresponds to the sampling time or time of the gas exchange.

Timestamp is defined using internal clock of GASERA ONE unit. It can be changed only via user interface.

The timestamp is in Linux (epoch) format, i.e., number of seconds elapsed since 1.1.1970.

Formatting of the ACON command response can be changed using the SCON command.

1.4.7. Set measurement component (CAS) order – SCOR

Available from MW ver. 1.6.0

The order of data elements is based on CAS codes of gas components. The order can be changed by using the SCOR command. If the SCOR command has not been invoked, the elements are ordered automatically by GASERA ONE.

f. format *SCOR* K0 <CAS_1> <CAS_2> … <CAS_n>
   x. response (success): *SCOR* 0
   xi. response (error) *SCOR* 1

Set the order of in which component CAS elements are returned when asking for the last measurement results using the ACON command, i.e., ACON data elements are ordered by CAS in the order specified by SCOR command.

The SCOR command is active until the device is reboot, i.e., after device reboot the command needs to be re-invoked prior to starting measurement.

1.4.8. Set concentration settings (format) – SCON

Available from MW ver. 1.7.3

g. format *SCON* K0 <format bits> (0/1 (show time) 0/1 (show CAS) 0/1 (show concentration) 0/1 (show gas inlet)
   xii. response (success): *SCON* 0
   xiii. response (error) *SCON* 1

Set the format of the ACON response. The format is specified as bit pattern. e.g., command SCON K0 0 0 1 0 means show only concentration values in ppm.
The command is active until device is rebooted, or SCON is called with new bit pattern value. The last bit (inlet) is optional and requires MW version 1.8.2 or newer.

1.4.9. Get Measurement Status (Phase) – AMST

Available from MW ver. 1.5.0

h. format *AMST* K0
   i. response *AMST* <errorstatus> <measurement status> (errorstatus: 0=no errors, 1=error)

Possible values of measurement status:
0 – none (device is idle)
1 – gas exchange in progress
2 – sample integration (measurement) in progress
3 – sample analysis in progress
4 – laser tuning in progress

1.4.10. Get Device Name – ANAM

Available from MW ver. 1.5.0

i. format *ANAM* K0
   ii. response *ANAM* <errorstatus> <device name string> (errorstatus: 0=no errors, 1=error)

Returns the device name string (if set). Empty value if not set.

1.4.11. Start new measurement (by task name) – STAT

Available from MW ver. 1.7.3

j. format *STAT* K0 <task name>
   iii. response *STAT* <errorstatus> (errorstatus: 0=no errors, 1=error)

Start a new measurement based on the specified task. Alternative to STAM, but parameter is task name instead of internal task id.

1.4.12. Get current measurement iteration number – AITR

Available from MW ver. 1.7.2

k. format *AITR* K0
   iv. response *AITR* <errorstatus> <iteration number> (errorstatus: 0=no errors, 1=error)

Request the current measurement iteration number. In MW version 1.7.2 or 1.7.3, the returned value is the internal (global) measurement iteration number from the internal database. In MW version 1.8.2 or newer, AITR returns the zero-based

iteration number within the currently in-progress measurement (starts from zero when a new measurement is started).

1.4.13. Get current device network settings – ANET

Available from MW ver. 1.8.7

l. format *ANET* K0
   v. response *ANET* <errorstatus> <useDHCP> <ip address> <netmask> <gateway>

   errorstatus: 0=no errors, 1=error
   usDHCP: 0/1 (DHCP not in use / in use)
   ip address: current IP address (NO_IP if not available)
   netmask: netmask IP (NO_NETMASK if not available)
   gateway: gateway IP (NO_GW if not available)

Request the current network settings for the device.

1.4.14. Set device network settings – SNET

Available from MW ver. 1.8.7

m. format *SNET* K0 <useDHCP> <ip address> <netmask> <gateway>
   vi. response *SNET* <errorstatus> (errorstatus: 0=no errors, 1=error)

Request to set the device network settings. All parameters are required. For example, to set device to use DHCP, set useDHCP parameter to 1, ip to NO_IP, netmask to NO_NETMASK and gateway to NO_GW.

In case IP should not be set, set ip address to NO_IP.
In case netmask should not be set, set netmask to NO_NETMASK.
In case gateway should not be set, set gatway to NO_GW.

NOTE: To make network setting changes effective, restart of the analyzer is required.

1.4.15. Get device parameter value – APAR

Available from MW ver. 1.8.9

n. format *APAR* K0 <parameter name>
   vii. response *APAR* <errorstatus> <parameter value> (errorstatus: 0=no errors, 1=error)

Request value of device parameter, such as cell temperature or heater power. The response contains the parameter value only on success (errorstatus 0). The command returns error status 1, for example, in case the specified parameter name does not exist. Parameter name is case insensitive.

Examples of parameter names: *CELLTEMP* (temperature of PA gas cell), *VAISALACO2VALUE* (CO2 value provided by Vaisala sensor, if assembled to GASERA ONE analyzer)

1.4.16. Set Online Measurement Mode – SONL

Available from MW ver. 2.0.1

o. format *SONL* K0 <enable (1) / disable (0)>
viii. response *SONL* <errorstatus> (errorstatus: 0=no errors, 1=error)

Set device online measurement mode. In case, online mode is enabled (1), no measurement results are stored in the device internal database in order to save space.

1.4.17. Get current device date and time – ACLK

Available from MW ver. 2.0.2

p. format *ACLK* K0
ix. response *ACLK* <errorstatus> <date/time string>

Return the current device date and time in the format: YYYY-mm-ddThh:mm:ss. Device uses internally UTC time zone. Time can be set via user interface only.

1.4.18. Set Laser Tuning Interval – STUN

Available from MW ver. 2.0.2

q. format *STUN* K0 <interval value>
x. response *STUN* <errorstatus> (errorstatus: 0=no errors, 1=error)

Set the interval / frequency value on-demand for laser tuning for laser-type devices (QCL/DFB) (overriding default value from GASERA ONE internal setting in init). Possible values:

- K0 = 0 – do not tune
- K0 = 1 – tune every iteration
- K0 > 1 – tune every Nth iteration.

The value is used until a new STUN command is received, or a new measurement is started (on new measurement start, the default value from init is assigned).

1.4.19. Get Measurement Task Parameters – ATSP

Available from MW ver. 2.4.0

r. format *ATSP* K0 <measurement task id>
   xi. response *ATSP* <errorstatus>
   <CompCAS1,CompCAS2,..,CompCASn>
   <TargetPressure> <Flush Time Bypass> <Flush Time Cell>
   <Cell Flush Cycles>
   (errorstatus: 0=no errors, 1=error)

Retrieve the list of parameters for the specified measurement task: list of measured component CAS, target pressure, flush time bypass, flush time cell, and cell flush cycles.

1.4.20. Get System Parameters List – ASYP

Available from MW ver. 2.4.0

s. format *ASYP* K0
   xii. response *ASYP* <errorstatus>
   <param1_name>,<param1_value>,<param1_min>,<param1_max>,<param1_unit> <param2_name>,…
   (errorstatus: 0=no errors, 1=error)

Retrieve the list of device system parameters (parameter name, value, min. value, max. value and unit).

1.4.21. Get Multi-Point Sampler Parameters – AMPS

Available from MW ver. 2.4.0

t. format *AMPS* K0
   xiii. response *AMPS* <errorstatus> <inlet-id> <active(0/1)> <byPassTimeSecs> <inlet_id>…

   (errorstatus: 0=no errors, 1=error, 2=request ok / MPS not connected)

Retrieve the inlet configuration for the Multi-Point Sampler (if available / connected).

1.4.22. Get Device Information – ADEV

Available from MW ver. 2.4.0

u. format *ADEV* K0
xiv. response *ADEV* <errorstatus>
<Manufacturer> <Serial Number> <Device Name>
<Firmware version>

(errorstatus: 0=no errors, 1=error)

Retrieve general device information, such as manufacturer, serial number, and firmware version. Response fields are surrounded by double quotes. In case some field in the response is not available, an empty string is returned.

1.4.23. Start Device Self-Test – STST

Available from MW ver. 2.4.0

v. format *STST* K0
xv. response *STST* <errorstatus>

(errorstatus: 0=no errors, 1=error)

Request to start device self-test routine. Returned errorstatus field indicates request status. The self-test routine result / state can be subsequently queried with the AK-command ASTR (Get device self-test result).

1.4.24. Get Device Self-Test Result – ASTR

Available from MW ver. 2.4.0

w. format *ASTR* K0
xvi. response *ASTR* <errorstatus> <self-test state/result>

(errorstatus: 0=no errors, 1=error)

Request device self-test result / state. If the request is processed successfully, error status returns 0, otherwise 1. The self-test state/result field is interpreted as follows:

-2 = test result N/A (e.g., self-test routine has not been started)
-1 = test in progress
0 = self-test failed
1 = self-test completed successfully

1.4.25. Reboot device – RDEV

Available from MW ver. 2.4.0

x. format *RDEV* K0
xvii. response *RDEV* <errorstatus>

(errorstatus: 0=no errors, 1=error)

Request to reboot the measurement device. After the success status is received, device reboots after approx. 2 seconds. NOTE: use caution when using this command, as the device is rebooted in any state (e.g., even though a measurement is in progress).

1.4.26. Send measurement trigger – TRME

Available since MW ver. 2.6.0

y. format *TRME* K0
xviii. response *TRME* <errorstatus>

(errorstatus: 0=no errors, 1=error)

When the device is in interval measurement mode, and waiting for external event, generates the trigger which causes the measurement progress to continue.

The triggers, which are sent outside of the waiting state, are ignored.

When the interval period is set to zero, the instrument will trigger only from external trigger.

When the interval period is finite and longer than the measurement iteration, the wait can be interrupted by sending TRME command. This, however, will not influence the internal timer which will continue to trigger with the original settings.

In example, if the interval is enabled with period of 60s, the interval timer will work with this interval. If TRME is used to start 2nd measurement at 55s, the 3rd measurement will be start still at 120s making the interval between 2nd and 3rd measurement to be 65s.

1.4.27. Set background compensation measurement interval – STDB

Available since MW ver. 2.6.0

z. format *STDB* K0 <interval value>
xix. response *STDB* <errorstatus>

(errorstatus: 0=no errors, 1=error)

Set the interval / frequency value on-demand for when background compensation measurement should be performed (overriding value for measurement task parameters). Possible values:

- value = 0 – no dynamic background measurement
- value = 1 – dynamic background measurement every iteration
- value > 1 – dynamic background measurement every Nth iteration

The value is used until a new STDB command is received, or a new measurement is started (on new measurement start, the default value from measurement task settings is used).

1.5. Examples of using the AK protocol

1.5.1. Example requests and responses

Examples of typical usage of the AK communication protocol are presented in this section. Screen shot of examples is given in the next section. Numbers within brackets in example requests and responses refer to line numbers in screen shot in the next section.

We can query the current device state using the following command

Request: <STX> ASTS K0 <ETX> [9]
Response: <STX> ASTS 0 5<ETX> [10], which indicates device is measuring.

To start a measurement with the GASERA ONE, a measurement task must be selected. Measurement tasks are created using the user interface of the GASERA ONE.

A list of measurement tasks can be requested with the *Ask Task List* command. The response will contain the task list (each data element is composed of the measurement task name and its internal task id). To start a measurement, we need to know the task id.

Request: <*STX*> ATSK K0 <*ETX*> *[1]*
Example response: <*STX*> ATSK 0 7 Calibration task 11 TEST<*ETX*> *[2]*
In the above example two tasks are returned, Calibration task with task id 11, and TEST with task id 11.

Request: <*STX*> SCOR K0 74-82-8 124-38-9 7732-18-5 630-08-0 10024-97-2 7664-41-7 7446-09-5<*ETX*>

Example response: <*STX*> SCOR 0 <*ETX*>

The above request specifies the order of components (CAS) when requesting last measurement results using ACON, i.e., 74-82-8 (CH4), 124-38-9 (CO2), 7732-18-5 (H2O), 630-08-0 (CO), 10024-97-2 (N2O), 7664-41-7 (NH3) and 7446-09-5 (SO2).

If the user wants to start a measurement using the above TEST task, the *Start New Measurement* command should be sent.

Request: <*STX*> STAM K0 11 <*ETX*> *[6]*
Example response: <*STX*> STAM 0 <*ETX*> *[7]* (0 status indicates success, 1 indicates some error condition)

Current measurement results can be requested using the *Get Last Measurement Results* command.

Request: <*STX*> ACON K0 <*ETX*> [15]
Example response:
<*STX*> ACON 0 1511865967 74-82-8 0.919439 1511865967 124-38-9 435.765 1511865967 7732-18-5 7125.4 1511865967 630-08-0 0 1511865967 10024-97-2 0 1511865967 7664-41-7 0.0044561 1511865967 7446-09-5 0<*ETX*> *[16]*

Using the timestamp field (time in Unix epoch), it is possible to determine when results have been updated. Successive calls will return equal timestamps and result values until the next measurement is complete and new results are available from the analyzer. In the above example response, we can interpret that the component with CAS value *74-82-8 (CH4)* has concentration value *0.919439 ppm* (at Unix epoch time *1511865967*) etc.

The measurement can be stopped at any time using the *Stop Current Measurement* command. Stopping the measurement corresponds to similar action triggered from user interface of GASERA ONE.

Request: <*STX*> STPM K0 <*ETX*> *[18]*
Example response: <*STX*> STPM 0 <*ETX*> [19]

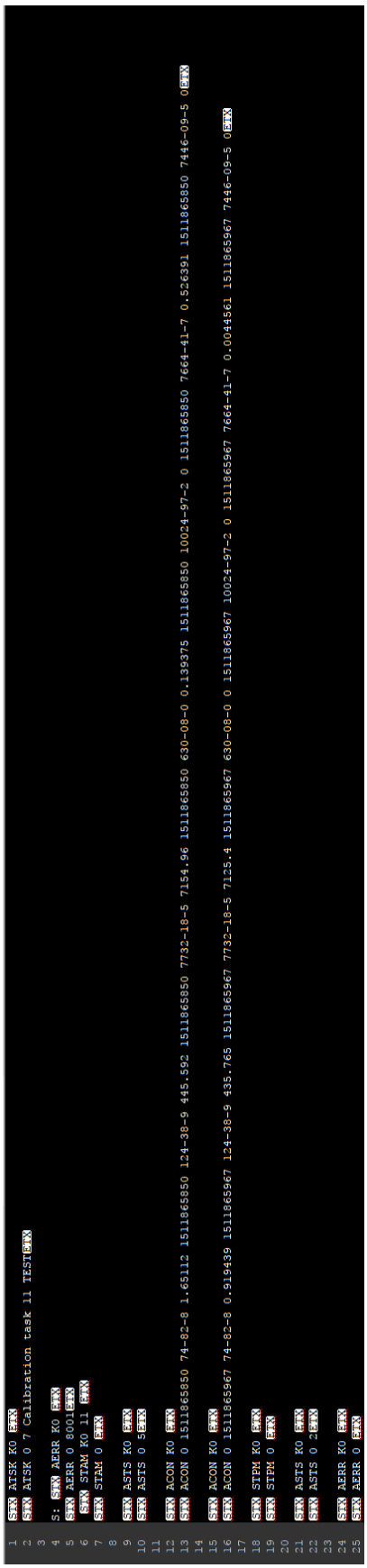
Device error state can be requested by using the following AK request:

Request: <*STX*> AERR K0 <*ETX*> [4]
Example response: <*STX*> *AERR 0 8001*<*ETX*> *[5]*

This indicates that error code 8001 is currently active. List of most common error codes is in the GASERA ONE User manual. The complete error code list is in GASERA ONE Service manual.

1.5.2. Example of AK communication

Below is the screen shot of example requests and responses given in the previous section.

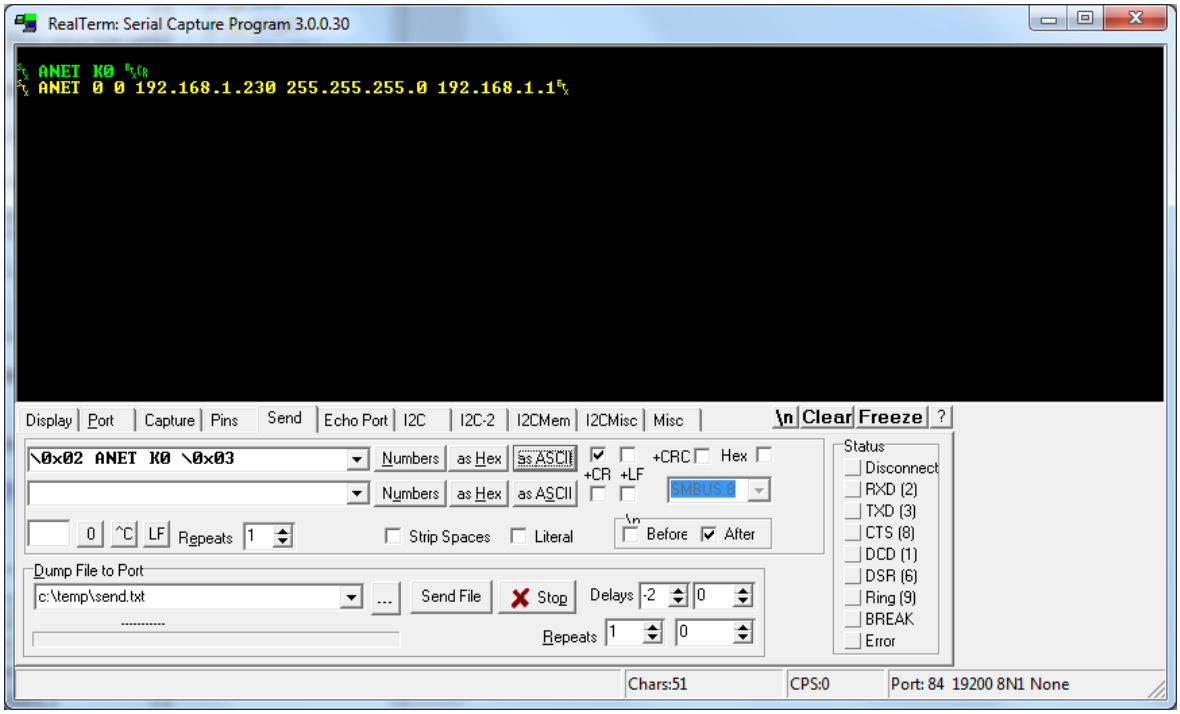

SEN ATSK KO CRC
SEN ATSK 0 7 Calibration task 11 TEST CRC
S: SEN AERR KO CRC
SEN AERR 0 8001 CRC
SEN STAM KO 11 CRC
SEN STAM 0 CRC
SEN ASTS KO CRC
SEN ASTS 0 5 CRC
SEN ACON KO CRC
SEN ACON 0 1511865850 74-82-8 1.65112 1511865850 124-38-9 445.592 1511865850 7732-18-5 7154.96 1511865850 630-08-0 0.139375 1511865850 10024-97-2 0 1511865850 7664-41-7 0.526391 1511865850 7446-09-5 0 CRC
SEN ACON KO CRC
SEN ACON 0 1511865967 74-82-8 0.919439 1511865967 124-38-9 435.765 1511865967 7732-18-5 7125.4 1511865967 630-08-0 0 1511865967 10024-97-2 0 1511865967 7664-41-7 0.0044561 1511865967 7446-09-5 0 CRC
SEN STPM KO CRC
SEN STPM 0 CRC
SEN ASTS KO CRC
SEN ASTS 0 2 CRC
SEN AERR KO CRC
SEN AERR 0 CRC

1.5.3. Example for Realterm

Connect the Gasera ONE as explained in the section 1.2.2 RS232.
Send as ascii (see figure below) e.g. request:

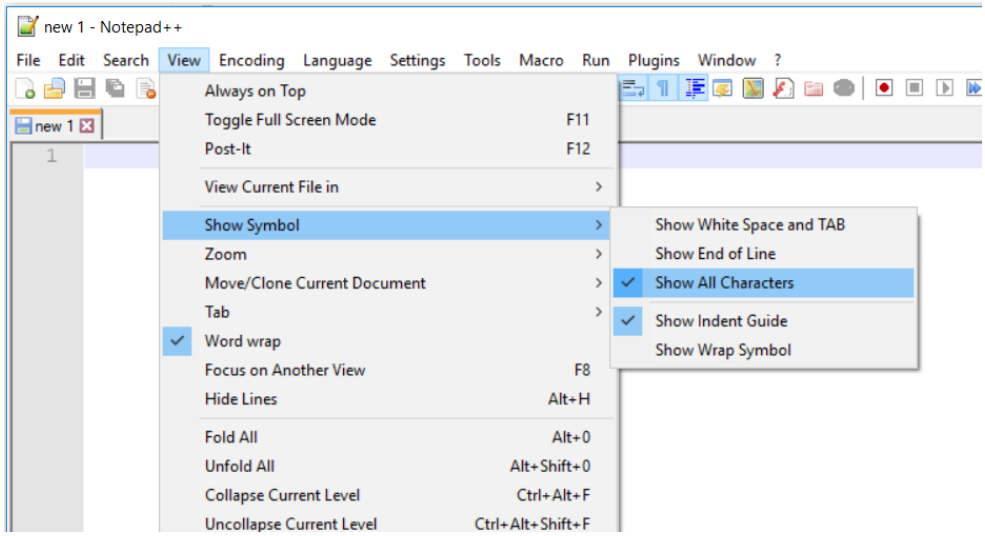
\0x02 ANET K0 \0x03

Where \0x02 is for <STX> and \0x03 for <ETX>.

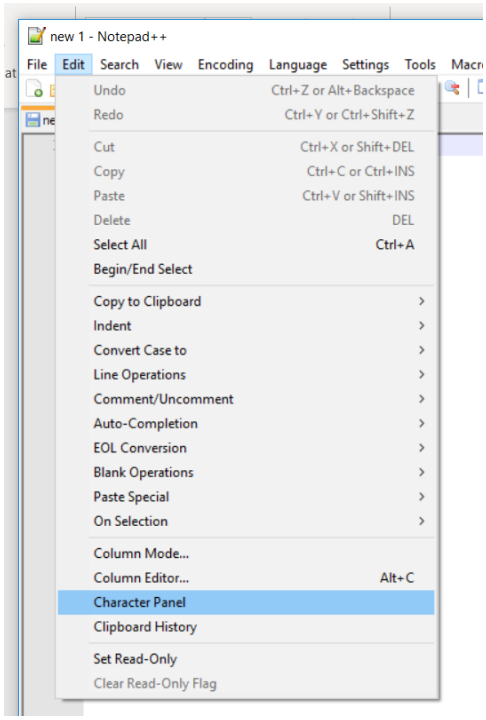


1.5.4. Example with Sockettest v 3.0.0

NOTE !!! STX and ETX are ascii characters 0x002 and 0x003 which means you have to have special tools to type them into the send string. One way to do this is to use Notepad++. First choose View->Show symbol-> Show All characters.

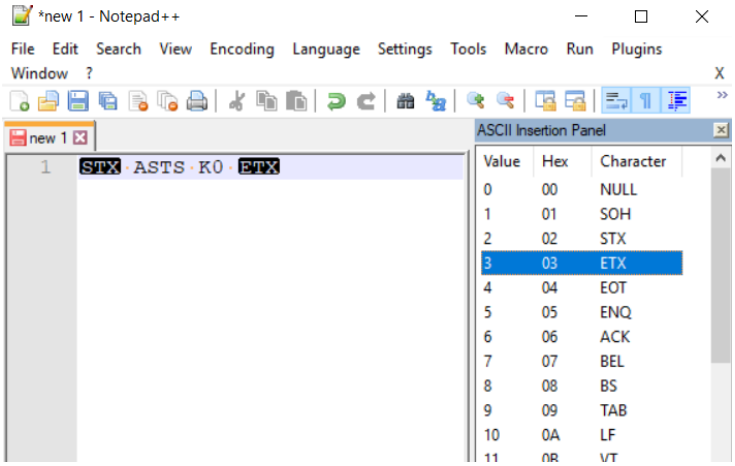


Then choose Edit->Character Panel



Then type the requested string, copy it and paste it to the send box of the Sockettest v 3.0.0.

First, type the string using the ASCII character panel for the STX and ETX and the copy the line.



When you have connected to the Gasera One with correct IP address and with the port 8888, paste the string to the send field. The STX and ETX characters are shown as boxes. Then press send and Gasera One responses.