Video Gauge™ V6.1 - Socket Comms protocol

Guidance for the Socket Comms feature connecting with Video Gauge™

September 2023

The Socket Comms feature allows measurement data to be received in real-time from the Video Gauge system via a network socket. It is also possible to send control commands to the Video Gauge to perform remote control of certain functions. Socket Comms consists of a 'Data Stream channel' for receiving the streamed measurement data and a 'Command channel' for sending the control commands.

Data Stream Channel

The Video Gauge™ software will listen for incoming connections on port 1234 by default. You can telnet into this port to see the data stream. The protocol of the data stream is defined below.

The data stream consists of a sequence of lines. Each line consists of a number of tab-delimited ("\t") items. Each line is terminated with a new-line carriage-return pair ("\n\r"). (Note – it is acceptable for an item to contain space characters.)
The first item in a line is a 'command' that defines the meaning of the remaining items on the line.

The available commands are:

VERSION
ENCODING
HEADINGS
DATA

Command descriptions (Protocol 1)

Command	Description	When it is sent
VERSION ###	The command is followed by an integer value that specifies the version number of the stream protocol. The current protocol is 1.	This command is only sent once. It is always the first command and is sent immediately after the stream is connected.
ENCODING ascii | binary	Specifies whether data values will be stream in ascii or binary format. By default, data will be sent in ascii format.	Initially sent after VERSION but before first DATA. Will be sent again if the encoding changes (i.e., Video Gauge™ user changes the encoding via the option dialogue).

HEADINGS ### heading1 hading2...	### Specifies the number of columns in the data. This is followed by the heading for each column.	Initially sent after VERSION but before the first DATA. Will be sent again if the measurements configured in Video Gauge™ change (e.g. Video Gauge™ user adds an extra strain gauge).
DATA value1 value2...	The actual data values. The number of values will equal the number of columns specified by the 'HEADINGS' command. The format depends on whether the encoding is ascii or binary: **Ascii** Each data value is converted to ascii using "%#g" printf format specifier. The data values are tab-delimited. If a data value is not valid (for example if a target has been lost), the ascii string "invalid" will be sent. **Binary** Each data value is sent as a 64 bit (8 byte) 'floating-point' number (C double type). This is followed by a single byte 'bool' value that indicates if the value is valid (false: invalid, true: valid). Each value follows on immediately from the previous- there is no separator.	Will be sent immediately after each image has been analysed by Video Gauge™.

Sample stream

In the following example each tab character ("\t") has been highlighted in blue. The protocol version is 1 and the data values are encoded as ascii. There are 3 data columns with headings: "Time", "Strain 1" and "Strain 2".

```
VERSION     1
ENCODING    ascii

HEADINGS    3       Time        Strain 1        Strain 2

DATA                48.6950     0.000000        0.000000
DATA                48.7620     0.00372214      -0.000522473
DATA                48.8280     0.00281814      -0.00025387
```

Interfacing your application with Socket Comms data stream

Your application needs to perform the following tasks:

1. Open a network connection to port 1234 on the Video Gauge™ PC (probably local host).
2. Parse the PROTOCOL command for the protocol version number.
3. Parse the ENCONDING commands for the encoding type: binary or ascii (ascii is default).
4. Parse the HEADINGS command for the list of column headings. This provides your application with a list of the measurements currently defined within Video Gauge™. If the user adds/edits/ deletes a measurement within Video Gauge™ a new HEADINGS command will be sent, and you must listen for these to ensure you maintain an up to date list of measurements.

   You will probably use this list of measurements to populate a 'drop-down' list that enables your application users to select which of the Video Gauge™ measurements are to be read into your application.

5. Parse the DATA commands to read the actual measurement values into your application.

Your application will probably use a sperate thread that opens the network connection, reads in a line at a time, parses the line and then notifies your application accordingly.

Command Channel

Sending commands to Video Gauge™ is only supported on version 4 of the software or higher. The software will listen for incoming connection on port 1235. You can use telnet to connect to this port and send commands (commands are sent to ASCII text). The available commands and descriptions are listed below.

Command	Description
mode measurements	Switch to measurements mode
mode test	Switch to testing mode
mode review	Switch to results review mode
test start numframes=<n>	Start a test (must be in test mode). Optionally stop it after n frames are captured if specified
test stop	Stop a test (must be in test mode)
test next	Move on to next test (must be in test mode)
test new using cameras {blank | from_template <filename> | using_current_settings [no_preset_selection] [preset_id <id>]}	Starts a new test (and switches to measurements mode). You can start a new blank test, a test from a template or a test with the current test settings. Optionally you can disable the pre-set selection popup window from appearing to prevent manual intervention being needed when a new test is started. You can also optionally provide the preset ID that you would like to start the new test with.
set archive on	Turn on saving of video and DAQ data to disk during test.
set archive off	Turn off saving of video and DAQ data to disk during the test
set archive pause	Pause saving of video data to disk during the test
set archive resume	Resume saving of video data to disk during the test

set notifications on	Enable sending of notification messages. Notification messages have the format. notification new <n>:<n>:<n> <cid> {<cname>}{<message header>}{<message content>}{ <message footer>} where<n>:<n>:<n> is the message id, <cid> is the category id and cname is the category name. When turned on, all current notifications are sent.
set notifications off	Disable sending of notification messages
notification dismiss <n>:<n>:<n>	Dismiss notification with id <n>:<n>:<n>
notification dismiss all	Dismiss all current notifications
set status on	Enable sending of status change messages Status change messages have the format "status <fromStatus>:<toStatus>" where <fromStatus> and <toStatus> can be one of "init-from-archive", "tracking", "recording", "recording-paused-video-archive", "reviewing" or "exporting-video". When turned on, a status message is sent with both the from and to fields set to the current status.
set status off	Disable sending of status change messages
get burstframecount [min|max|channel <n>]	Get the minimum frames captured across all channels, the maximum frames captured across all channels or the number of frames captured on channel n
save withvideo <filename>	Save the test results together with the video file & DAQ data (requires that archive has been set to on)
save withoutvideo <filename>	Save the test results without the video file or DAQ data
export rawdata <filename> starttime=<mintime> endtime=<maxtime> decimationtype= [nthframe|targetfrequency] decimationvalue=<n>	Exports a CSV or a TXT file containing the test data. The values that are exported depend on what is set up in the test template. Optionally specify a minimum and maximum time, and data decimation settings.