VEOBOX

CONTROL PROTOCOL FOR EXTERNAL SYSTEMS

30/09/2015

VEO-LABS

14, rue du Patis Tatelin Batiment D 35700 RENNES - France

E-mail : info@veo-labs.com

www.veo-labs.com

Table of contents

General

This document provides information to control the VEOBOX through external equipment.

First part describes all the supported communication transports.

Second part explains the general guidelines on messages formatting.

Finally, this document lists all the supported commands, and related responses that should be received.

Communication transports

Communication with the VEOBOX can be done through TCP/IP link or RS-232 link.

Both links can work together at the same time as long as each one is properly configured.

TCP/IP

IP settings must be configured using the Veobox Web interface. (Menu : Settings > IP control)
The listening port can be configured in the same place.

Messages are sent as TCP packets.

RS-232

Serial line as the following configuration which is not editable:

- Baudrate: 9600
- Byte size: 8
- Parity: None
- Stop bits: 1
- Flow control: No

Since RS-232 is not a message-based transport, messages are separated by a newline character: `\n`

Messages formatting

As a general rule, all messages use ASCII coding.

The protocol uses the simple request-response paradigm.

Response is only emitted back on the transport from which the request has been received and not on all transports.

Sending multiple requests without synchronously waiting for the response is not recommended, because responses will be returned in an undefined order, and there is no mean to identify the corresponding original request.

Requests

Requests sent by the client can be prepended or appended by blank characters such as tabulation, newline or spaces. These blank characters are ignored.

In the same way, tokens such as command name or arguments can be separated by any (possibly multiple) blank characters.

Example:

```
  \t  command   value \n
```

will be treated the same way as:

```
command value
```

Warning: When using RS-232 transport, all requests MUST be ended by a newline character `\n` which is the messages separation character. Thereby, this character MUST NOT be used inside a request to separate tokens.

Responses

Responses are always terminated by a newline character `\n` which makes it easy to control the system using generic tools such as `telnet`.

When the response contains multiple tokens, they are separated by a single space character.

When the response contains keyword/value tuples, keyword and value are separated by a single = character. There is no space character before or after this = character.

Example:

```
keyword1=value1 keyword2=value2\n
```

Commands list

1.1 Encoder status

Command: `status`

Return value: `[starting|started|stopping|stopped|error|unknown]`

Return the current status of the encoder.

Status	Description
`starting`	The recorder is starting a new recording
`started`	The recorder is currently recording
`stopping`	The recorder is finalizing the recording
`stopped`	The recorder is ready
`error`	A fatal error occurred in last recording. The recorder is ready now.
`unknown`	The recorder cannot be reached

1.2 Start recording

Command: `start [1-5]`

Return value: `success=[true|false]`

Start a new recording.

A preset number must be provided which corresponds to the preset to use. This preset number must be between 1 and 5. Presets are pre-configured using the Veobox Web interface. (Menu settings > IP control)

Return false if the recorder is not ready or if an error occurred.
Return true if the command is accepted.

1.3 Stop recording

Command: `stop`

Return value: `success=[true|false]`

Stop the currently running recording.

Return false if no recording is running or if an error occurred.
Return true if the command is accepted.

1.4 Get information on video input

Command: `video.info [camera|slides]`

Return value:

```
supported=[true|false]
timings=[<width>x<height>[i|p]@<fps>|null]
```

Examples:

```
supported=true timings=1920x1080p@60

supported=false timings=320x240i@30

supported=false timings=null
```

Return information on the specified video input.

`timings` contains the video input detected timings: resolution, interleaving and framerate.

`supported` is `true` when the input can be used for recording.
`supported` is `false` when the input cannot be used for recording. In this case, a start command will most likely lead to an error status.