OptoFidelity Video Multimeter Control API

Version 1.1

OptoFidelity

OptoFidelity Oy • sales@optofidelity.com • www.optofidelity.com

Contents

1 Introduction

OptoFidelity Video Multimeter is device for measuring video playback quality. This document defines a protocol for controlling OptoFidelity Video Multimeter device from external device, e.g. PC. Control protocol enables measurement initialization and result data collection in automated measurement environment without using device manually.

Note that the actual set of available commands depends on the licenses of Video Multimeter features that are present on the device. Therefore, some of these may not apply.

2 Control protocol

Controlling device (PC) and Video Multimeter are connected with USB cable. Serial over USB (CDC ACM) protocol is used in control protocol communication. Communication parameters are defined in Chapter 4.

When USB cable is connected, selection window on Video Multimeter display appears. Selecting "Control Protocol" enables usage of control protocol.

Controlling device is Master and Video Multimeter is slave in connection. Only master can initialize communication and slave always sends response.

Master must wait for slave's response to command before sending new command. If slave does not send response, new command can be sent after *Response timeout*.

Slave sends response to command immediately. Maximum time for response is *Response Timeout.* If slave cannot send required information, e.g. data is not available, error codes are used.

Slave ignores commands that are received during command processing.

Video Multimeter can be operated normally when Control API is in use. Note that if Video Multimeter is controlled from touch screen between Control Protocol commands, device state may have been changed.

2.1 Framerate measurement

Framerate measurement using control API is presented in Figure 1. Framerate measurement can be initialized by starting Framerate application. Measurement contains three phases: calibration, measurement and result reading. Calibration is not necessary if device has been already calibrated.

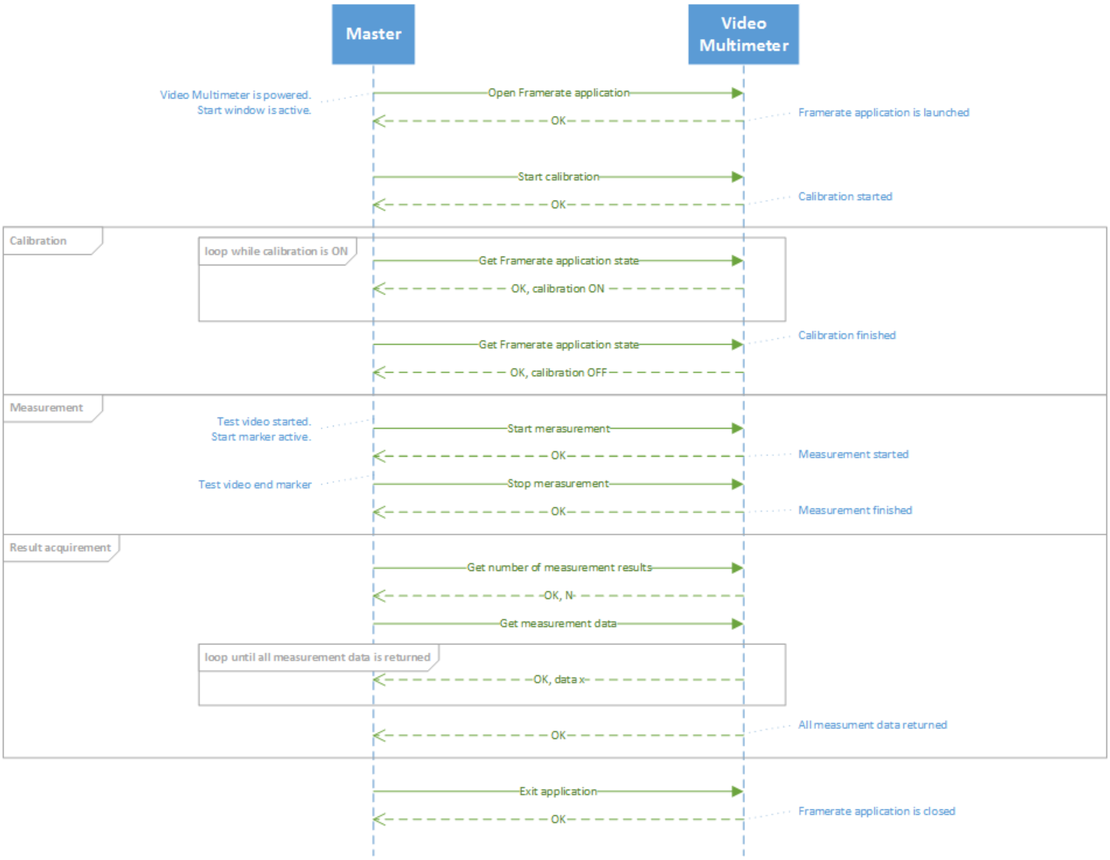

Figure 1. Framerate measurement using control API.

For successful results, measurement must be started during test video start marker (white). Measurement must be stopped during test video end marker (white).

Video Multimeter does not send measurement data during measurement.

Measurement data can be required when measurement has been stopped.

Get measured data command returns results one result at a time. The oldest result is given first. When there are no more results available, response without result data is sent.

2.2 Camera framerate measurement

Camera Framerate and Dual Camera FPS measurements are controlled in a very similar way. Both measurement utilizes Camera FPS Target (see Figure 2) application. When the FPS target is calibrated and it is running, either Camera Framerate or Dual Camera FPS application is launched. The applications support similar commands (see Figure 3), but returned measurement data differs. The measurement data structure is described in the chapters **Error! Reference source not found.** and **Error! Reference source not found.**.

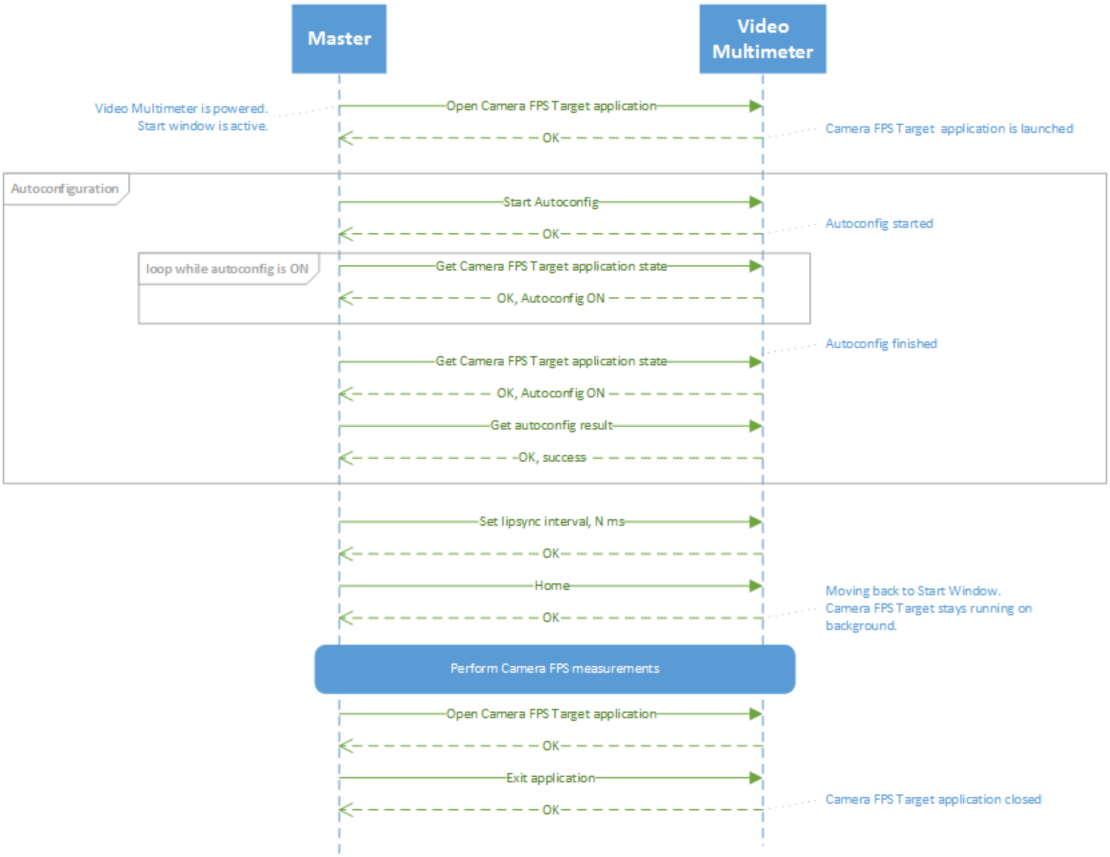
Master
Video Multimeter
Video Multimeter is powered. Start window is active.
Open Camera FPS Target application
OK
Camera FPS Target application is launched
Autoconfiguration
Start Autoconfig
OK
Autoconfig started
loop while autoconfig is ON
Get Camera FPS Target application state
OK, Autoconfig ON
Autoconfig finished
Get Camera FPS Target application state
OK, Autoconfig ON
Get autoconfig result
OK, success
Set lipsync interval, N ms
OK
Home
OK
Moving back to Start Window. Camera FPS Target stays running on background.
Perform Camera FPS measurements
Open Camera FPS Target application
OK
Exit application
OK
Camera FPS Target application closed

Figure 2. Camera FPS target application usage.

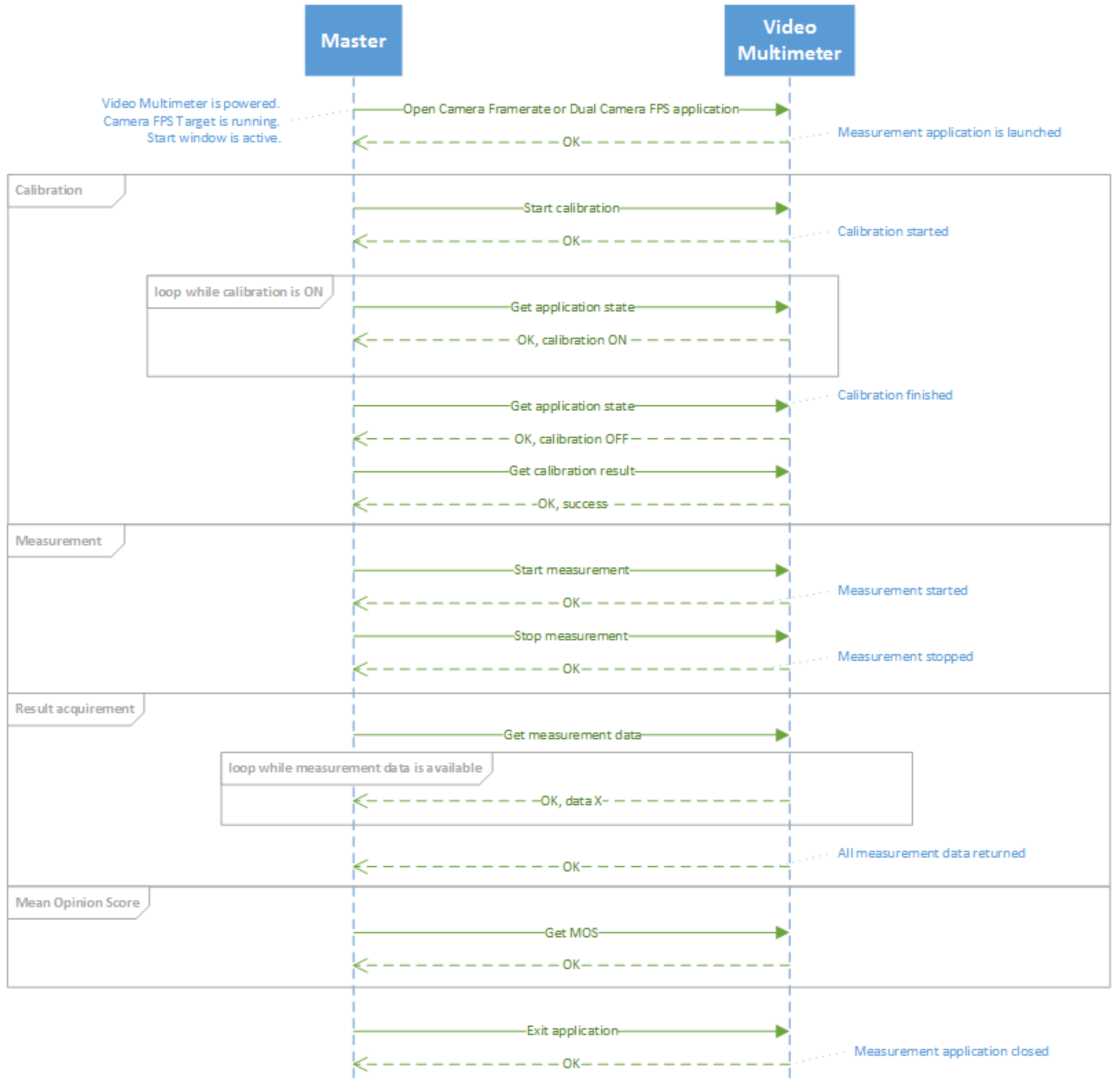

Figure 3. Camera framerate measurements.

3 Commands

Commands are human readable and they consist of Command Code, Command Parameters, Return value and Return Parameters. Fields are separated by spaces.

3.1 Return Values

Slave device sends response to all commands. Response must include Return value. Table 1 presents the return values.

Table 1. Return value codes

Return value code	**Description**
OK	Success
E1	Command not found
E2	Unsupported parameter
E3	Command not allowed
E4	No data available
E5	Unidentified error

3.2 General commands

General commands can be sent at any state.

Command name	**Code**	**Command parameters**	**Return value**	**Return Parameters**
Home	HOME	-	OK/E2	-
Get time	GETTIME	-	OK	Time in format: dd.mm.yyyy hh:mm:ss for example: 17.03.2014 11:55:16
Set time	SETTIME	Time in format: dd.mm.yyyy hh:mm:ss	OK/E2	-
Get Battery level	GETBAT	-	OK/E2	Battery level percentage (0-100)

Reboot immediately	REBOOT	-	NO RESPONSE	-
Enable Watchdog	WATCHDOG	Maximum number of seconds between API commands	OK/E2/E3	-
Set config.ini parameter	SETCONFIG	SectionName SettingName SettingValue	OK/E2	-
Get config.ini parameter	GETCONFIG	SectionName SettingName	OK/E2	SettingValue

3.2.1 Home

This command returns to start window and leaves applications running on background.

```
HOME
OK
```

Command code

HOME

Parameters

None

Return values

OK	Success
E2	Parameters available

Return Parameters

None

Note

If Start Window is already active this command has no effect. This command can be used for getting device into a known state.

3.2.2 Get time

This command returns to start window and leaves applications running on background.

```
GETTIME
```

```
OK 12.08.2016 10:24:53
```

Command code

GETTIME

Parameters

None

Return values

OK	Success
E2	Parameters available

Return Parameters

Time in format:

dd.mm.yyyy hh:mm:ss

for example:

17.03.2014 11:55:16

Note

-

3.2.3 Set configuration parameter value

SETCONFIG sets the value of a parameter in /frm/config.ini file. The change is applied immediately and persists over reboot.

```
SETCONFIG BacklightPeriodDetector sequence_phase 3
OK
```

Command code

SETCONFIG

Parameters

INI file section (BacklightPeriodDetector in above example).

INI file parameter name (sequence_phase in above example).

Parameter value (3 in above example)

Return values

OK	Success
E2	Invalid number of parameters

Return Parameters

None

Note

-

3.2.4 Get configuration parameter value

GETCONFIG gets the value of a parameter in /frm/config.ini file.

```
GETCONFIG BacklightPeriodDetector sequence_phase
OK 3
```

Command code

GETCONFIG

Parameters

INI file section (BacklightPeriodDetector in above example).

INI file parameter name (sequence_phase in above example).

Return values

OK	Success
E2	Invalid number of parameters

Return Parameters

Value of the parameter requested

Note

-

3.3 Start Window commands

Start window commands can be sent when Video Multimeter is in Start Window state.

Command name	Code	Command parameters	Return value	Return Parameters
Get applications	GETAPPS	-	OK/E1/E2	List of available applications separated by space
Open application	OPEN	Application, one of the listed applications from previous command	OK/E1/E2/E3	-

3.3.1 Get applications

This command lists available applications.

```
GETAPPS
OK FRAMERATE SYSTEM_INFORMATION
```

Command code

GETAPPS

Parameters

None

Return values

OK	Success
E1	Not in Start Window
E2	Unknown parameter

Return Parameters

Applications codes.

FRAMERATE = Framerate application

CAMERA_FRAMERATE = Camera framerate application

CAMERA_FPS_TARGET = Camera FPS target application

SYSTEM_INFORMATION = System Information application

DUAL_CAMERA_FPS = Dual Camera FPS application

VR_MEASUREMENT = Measure VR displays application

Note

Command is available only when Start window is active.

3.3.2 Open application

This command opens application.

```
OPEN FRAMERATE
OK
```

Command code

OPEN

Parameters

Application code.

FRAMERATE = Framerate application

CAMERA_FRAMERATE = Camera framerate application (requires license)

CAMERA_FPS_TARGET = Camera FPS target application (requires license and RGB LED connected to sensor port)

SYSTEM_INFORMATION = System Information application

DUAL_CAMERA_FPS = Dual Camera FPS application

VR_MEASUREMENT = Measure VR displays application

Return values

OK	Success
E1	Not in Start Window
E2	Unknown parameter
E3	Too many applications open. Not enough free memory to open application. Missing sensor required for application.

Return Parameters

None

Note

Command is available only when Start window is active. Only three applications can be open at same time. Some applications require application specific license. Some applications require that there is a measurement or output device connected to sensor port.

3.4 General Application commands

General application command can be sent when any application is open.

Command name	Code	Command parameters	Return value	Return Parameters
Exit Application	EXIT	-	OK/E1/E2	-

3.4.1 Exit application

This command exits application and returns to start window.

```
EXIT
OK
```

Command code

EXIT

Parameters

None

Return values

OK	Success
E1	Not in any application
E2	Parameters available

Return Parameters

None

Note

If Start Window is active this command has no effect.

3.5 Framerate Application specific commands

Framerate application specific commands can be sent only if framerate application is open. Note that during measurement, most of the commands are disabled.

Command name	Code	Command parameters	Return value	Return Parameters
Get Framerate state	GETSTATE	-	OK/E1/E2	calib 0/1 meas 0/1
Start calibration	STARTCAL	-	OK/E1/E2/E3	-
Stop calibration	STOPCAL	-	OK/E1/E2/E3	-
Get marker type	GETM	-	OK/E1/E2/E3	RGB/BW/Any
Set marker type	SETM	RGB/BW/Any	OK/E1/E2/E3	-
Start Measurement	STARTMEAS	-	OK/E1/E2/E3	-
Stop measurement	STOPMEAS	-	OK/E1/E2/E3	-
Get number of measurement data	GETN	-	OK/E1/E2/E3	0-N
Get measurement data	GETDATA	-	OK/E1/E2/E3/E4	Measurement data in format: timestamp(us); frame

				time(us); color; dropped frames; lipsync value(ms, optional)
Get stats	GETS	-	OK/E2/E3	Statistics data in TDB format.
Save	SAVE	-	OK/E1/E2/E3/E4	-
Get calibration values	GETCAL		OK/E1/E2	val1 val2 val3 val4 val5 val6 val7 val8 val9 val10
Set calibration values	SETCAL	val1 val2 val3 val4 val5 val6 val7 val8 val9 val10	OK/E1/E2/E3	-
Retrieve MOS results	GETMOS		OK/E1/E2/E3	Composite, Jerkiness, Jitter, Dropped frames, Lipsync Delay, Lipsync Jitter

3.5.1 Get Framerate state

This command gets state of Framerate application. Example of command and response is presented below. Response tells that calibration is not ongoing and measurement is active.

```
GETSTATE
OK calib 0 meas 1
```

Command code

GETSTATE

Parameters

None

Return values

OK	Success
E1	Not in Framerate application.
E2	Parameters available

Return Parameters

calib	Informs that next parameter is calibration state
meas	Informs that next parameter is measurement state
0	State not active
1	State active

Note

-

3.5.2 Start calibration

This command starts calibration.

```
STARTCAL
OK
```

Command code

STARTCAL

Parameters

None

Return values

OK	Success
E1	Not in Framerate application.
E2	Parameters available
E3	Trying to restart calibration.

Return Parameters

None

Note

-

3.5.3 Stop calibration

Stop calibration.

```
STOPCAL
OK
```

Command code

STOPCAL

Parameters

None

Return values

OK	Success
E1	Not in Framerate application.
E2	Parameters available
E3	Calibration not ongoing.

Return Parameters

None

Note

Command is targeted for use in situations where calibration jams. This command is allowed only when calibration is ongoing.

3.5.4 Start Measurement

Start Framerate measurement.

```
STARTMEAS
OK
```

Command code

STARTMEAS

Parameters

None

Return values

OK	Success
E1	Not in Framerate application.
E2	Parameters available
E3	Measurement already active.

Return Parameters

None

Note

Command is disabled during measurement

3.5.5 Stop Measurement

Stop Framerate measurement.

```
STOPMEAS
```

```
OK
```

Command code

STOPMEAS

Parameters

None

Return values

OK	Success
E1	Not in Framerate application.
E2	Parameters available
E3	Measurement is not active.

Return Parameters

None

Note

Command is disabled when measurement is not ongoing.

3.5.6 Get number of measurement data

Get number of measurement results.

```
GETN
OK 5
```

Command code

GETN

Parameters

None

Return values

OK	Success
E1	Not in Framerate application.
E2	Erroneous parameter
E3	Measurement ongoing.

Return Parameters

Number of result items received from previous measurement.

Note

Command is disabled during measurement.

3.5.7 Get measurement data

Gets result data from previous measurement. Results are returned one-by-one in pre-defined format. Response without data field, indicates end of data.

```
GETDATA
OK   19038000; 34000; g;    79
OK   19072000; 82000; c;    79
OK   19154000;    -1; b;    80
OK   19154000; 51000; p;    80
OK   19205000; 34000; k;    80;  -116
OK
```

Command code

GETDATA

Parameters

None

Return values

OK	Success
E1	Not in Framerate application.
E2	Erroneous parameter
E3	Measurement ongoing.
E4	No data available from previous measurement.

Return Parameters

Result data item in format: timestamp(us); frame time(us); color; dropped frames; Lipsync (ms, optional).

Timestamp is proportional to beginning of measurement. It is presented in us. Note that dropped frame timestamp is same as timestamp of next valid frame.

Frame duration is presented in us. Value of -1 indicates dropped frame.

Color values are yellow(y), green(g), cyan(c), blue(b), purple(p), red(r), black(k).

Dropped frames field indicates total number of dropped frames.

Lipsync value is available only if there was corresponding audio marker detected for frame. Typically Lipsync value is reported every 1-10 seconds. Lipsync value indicates audio and video timing. Positive value indicates that audio is late, negative value indicates that audio is early.

For example:

OK 19205000; 34000; k; 80; -116

This indicates black frame with length of 34ms at timestamp 19205000us from the beginning of measurement. 80 dropped frames detected during measurement. Audio is detected to be 116ms early compared to video signal.

Note

Command is disabled during measurement.

3.5.8 Save measurement results

This command saves measurement data to Video Multimeter internal memory.

```
SAVE
OK
```

Command code

SAVE

Parameters

None

Return values

OK	Success
E1	Not in Framerate application.
E2	Parameters available
E3	Measurement active.
E4	Data was already saved. No data available for saving.

Return Parameters

None

Note

Command is disabled during measurement. Data cannot be saved before measurement is completed.

3.5.9 Get calibration values

This command gets RGB calibration values.

```
GETCAL
OK 100 40 20 280 320 30 130 130 190 0
```

Command code

GETCAL

Parameters

None

Return values

OK	Success
E1	Not in Framerate application.
E2	Parameters available.

Return Parameters

List of calibration values. Ten integer values and last value must be either 0 or 1.

Note

Command is disabled during measurement.

3.5.10 Set calibration values

This command sets RGB calibration values.

```
SETCAL 100 40 20 280 320 30 130 130 190 0
OK
```

Command code

SETCAL

Parameters

RGB calibration values. Ten integer values.

Return values

OK	Success
E1	Not in Framerate application.
E2	Incorrect parameters
E3	Calibration is active

Return Parameters

None

Note

Command is disabled during calibration.

3.5.11Get measurement statistics

The command returns the summary of the measurement data from the previous measurement. Example of command and response is presented below.

```
GETMEASSTATS
OK 34.4 ms;13.1 ms;0.2 s; 5.4 ms;4.5 ms
```

Command code

GETMEASSTATS

Parameters

None

Return values

OK	Success
E1	Not in Framerate application.
E2	Parameters available
E3	Measurement or calibration active. Measurement data is not saved.
E4	No data available from the previous measurement.

Return Parameters

A list of statistical values from the previous measurement:

1. Average frame interval
2. Standard deviation of the frame intervals
3. Duration lost due to the dropped frames
4. Average latency of audio markers (compared to the video stream)
5. Standard deviation of the audio latency

3.5.12Get MOS (Mean Opinion Score) results

This command retrieves the mean opinion score results for the latest measurement.

```
GETMOS
OK 4.8 4.5 5.0 5.0 NaN NaN
```

Command code

GETMOS

Parameters

None

Return values

OK	Success
E1	Not in Framerate application.
E2	Incorrect parameters
E3	MOS is not allowed

Return Parameters

List of MOS scores:

1. Composite (weighted average of all scores)
2. Jerkiness (frame interval)
3. Jitter (frame interval variance)
4. Dropped frames (interval between dropped frames)
5. LipsyncDelay (average lipsync delay)
6. LipsyncJitter (variance of lipsync)

Each value is in range 1.0 to 5.0, or NaN for lipsync if no lipsync measurement was performed.

3.6 Measure VR Display specific commands

3.6.1 Get Measure VR displays application state

This command gets state of Measure VR displays application. Example of command and response is presented below. Response tells that measurement is active.

```
GETSTATE

OK meas 1
```

Command code

GETSTATE

Parameters

None

Return values

OK	Success
E1	Not in Measure VR displays application.
E2	Parameters available

Return Parameters

meas	Informs that next parameter is measurement state
0	State not active
1	State active

Note

-

3.6.2 Start measurement

Start Measure VR displays measurement.

```
STARTMEAS
OK
```

Command code

STARTMEAS

Parameters

None

Return values

OK	Success
E1	Not in Dual Camera FPS application.
E2	Parameters available
E3	Measurement is already active.

Return Parameters

None

Note

Command is disabled when measurement is ongoing.

3.6.3 Stop measurement

Stop Measure VR displays measurement.

```
STOPMEAS
OK
```

Command code

STOPMEAS

Parameters

None

Return values

OK	Success
E1	Not in Measure VR displays application.
E2	Parameters available
E3	Measurement was not active.

Return Parameters

None

Note

Command is disabled when measurement is not ongoing.

3.6.4 Get measurement data

GETDATA command returns the results of the previous measurement. The results are returned line-by-line in the pre-defined format. The response without data fields indicates the end of data. There are also comment lines starting with the character # which contain information about the measurement and data format.

```
GETDATA
OK   # Recorded at 2018-01-26 10:06:10 using OptoFidelity Video
Multimeter
OK   # Frame start (us); M2P Latency (ms); Latency accuracy (ms);
Backlight on time (us); Backlight period (us);
OK           0;    43;     4;  5121; 16850;
OK       16833;    43;     4;  5121; 16850;
OK       33700;    43;     3;  5121; 16850;
OK       50550;    43;     3;  5151; 16850;
OK       67393;    41;     3;  5121; 16850;
OK       84285;    40;     3;  5121; 16850;
OK
```

Command code

GETDATA

Parameters

None

Return values

OK	Success
E1	Not in Measure VR displays application.
E2	Erroneous parameter
E3	Measurement ongoing.
E4	No data available from previous measurement.

Return Parameters

Measurement data consists of the following columns:

- Capture time (microseconds from the start of the measurement)
- Motion-to-photon latency (milliseconds)
- Latency accuracy (milliseconds)
- Duration (in microseconds) that the display backlight was switched on. The value reflects the pixel persistence of the display.

- Display backlight period (in microseconds). The display refresh rate is 10e6 divided by the value of this field.

Note

Command is disabled during measurement.

3.6.5 Save measurement data

This command saves measurement data to Video Multimeter internal memory.

```
SAVE
OK
```

Command code

SAVE

Parameters

None

Return values

OK	Success
E1	Not in Measure VR displays application.
E2	Parameters available
E3	Measurement active.
E4	Data was already saved. No data available for saving.

Return Parameters

None

Note

Command is disabled during measurement. Data cannot be saved before measurement is completed.

3.6.6 Get raw encoder data

GETENCDATA command returns the raw encoder values from the previous measurement. The command is like the GETDATA command, the only difference being the data returned.

```
GETENCDATA
OK    # Recorded at 2018-01-26 10:06:10 using OptoFidelity Video Multimeter
OK    # Time (us); Encoder X;
OK    4604;  2179;
OK    9969;  2182;
OK    15081;  2185;
OK
```

Return Parameters

Measurement data consists of the following columns:

- Capture time (microseconds from the start of the measurement)
- Encoder position (raw number of encoder pulses)

3.6.7 Get raw optical flow data

GETOFDATA command returns the raw optical flow values from the previous measurement. The command is like the GETDATA command, the only difference being the data returned.

```
GETOFDATA
OK    # Recorded at 2018-01-26 10:06:10 using OptoFidelity Video
Multimeter
OK    # Time (us); Optical flow X; Optical flow Y; Surface quality;
OK    6125;   282;   227;   255;
OK    17216;   286;   226;   255;
OK    28306;   287;   227;   255;
OK
```

Return Parameters

Measurement data consists of the following columns:

- Capture time (microseconds from the start of the measurement)
- Optical flow X position (camera pixels)
- Optical flow Y position (camera pixels)
- Surface quality (0-255, the “goodness” of the current frame for optical flow calculation)

3.7 Video Transfer Performance specific commands

Video Transfer Performance specific commands can be sent only if Video Transfer Performance application is open.

Command name	Code	Command parameters	Return value	Return parameters
Get application state	GETSTATE	-	OK/E1/E2	calib 0 meas 0/1
Start measurement	STARTMEAS	-	OK/E1/E2/E3	-
Stop measurement	STOPMEAS	-	OK/E1/E2/E3	-
Get measurement results	GETDATA	-	OK/E1/E2/E3/E4	Measurement data in CSV format.
Save results to file	SAVE	-	OK/E1/E2/E3/E4	-
Get Mean Opinion Score values	GETMOS	-	OK/E1/E2/E3/E4	MOS components separated by space.

3.7.1 Get application state

The GETSTATE command checks the current state of the application. Because Video Transfer Performance application requires no color calibration, the "calib" value in result is always 0, while the "meas" value indicates whether a measurement is running.

```
GETSTATE
OK calib 0 meas 0
```

Command code

GETSTATE

Parameters

None

Return values

OK	Success
E1 UNKNOWN_COMMAND	Video Transfer Performance application is not active.
E2 PARAMETER_ERROR	Too many parameters.

Return parameters

calib 0	Always 0 (no calibration required)
meas 0/1	Measurement running (1) or not running (0)

3.7.2 Start measurement

Start Video Transfer Performance measurement.

STARTMEAS
OK

Command code
STARTMEAS

Parameters
None

Return values

OK	Success
E1 UNKNOWN_COMMAND	Video Transfer Performance application is not active.
E2 PARAMETER_ERROR	Too many parameters.
E3 NOT_ALLOWED	Measurement is already running.

Return parameters

None

3.7.3 Stop measurement

Stop the running Video Transfer Performance measurement.

STOPMEAS
OK

Command code
STOPMEAS

Parameters
None

Return values

OK	Success
E1 UNKNOWN_COMMAND	Video Transfer Performance application is not active.
E2 PARAMETER_ERROR	Too many parameters.
E3 NOT_ALLOWED	Measurement is not running.

Return parameters

None

3.7.4 Get measurement results

Retrieves data from the latest measurement. Results are in CSV format, with columns as follows:

# Timestamp (us); Frame length (us); Color; Lipsync (ms); Latency (ms)

Example usage:

```
GETDATA
OK 0; 17000; #1a4507; ; 10
OK 17000; 33000; #064c10; ; 10
OK 50000; 15000; #05310f; ; 10
OK 65000; 18000; #043419; ; 10
OK 83000; 17000; #031718; ; 10
OK
```

The response is terminated by an OK line without any data fields.

Command code
GETDATA

Parameters
None

Return values

OK	Success
E1 UNKNOWN_COMMAND	Video Transfer Performance application is not active.
E2 PARAMETER_ERROR	Too many parameters.
E3 NOT_ALLOWED	Measurement has not been stopped yet.
E4 NO_DATA	No measurement has been done, or previous measurement had 0 rows of data.

Return parameters

Result data columns are:

Column	Measurement unit	Description
1: Timestamp	microseconds (µs)	Start of the frame display, running timestamp from the start of the measurement.
2: Frame length	microseconds (µs)	Length of the frame display, i.e. the difference between timestamps of this and next row.
3: Color	#RGB	Detected color of the frame, used only for diagnostic purposes.
4: Lipsync	milliseconds (ms)	Detected lipsync value, or empty if no lipsync marker occurred during this frame.
5: Latency	milliseconds (ms)	Detected latency value.

3.7.5 Save measurement results

Save the measurement results to Video Multimeter internal memory card. Note that GETDATA command can be used also without saving the results.

SAVE
OK

Command code
SAVE

Parameters
None

Return values

OK	Success
E1 UNKNOWN_COMMAND	Video Transfer Performance application is not active.
E2 PARAMETER_ERROR	Too many parameters.
E3 NOT_ALLOWED	Measurement is still running, or there was a problem writing to the memory card.
E4 NO_DATA	No measurement has been done, or previous measurement was already saved.

Return parameters

None

3.7.6 Get Mean Opinion Score values

Returns the Mean Opinion Score values calculated from the previous measurement.

GETMOS
OK 4.8 5.0 4.1 NaN NaN 5.0 5.0

Command code

GETMOS

Parameters

None

Return values

OK	Success
E1 UNKNOWN_COMMAND	Video Transfer Performance application is not active.
E2 PARAMETER_ERROR	Too many parameters.
E3 NOT_ALLOWED	Measurement is still running, or MOS option is not available.
E4 NO_DATA	No measurement has been done, or previous measurement had 0 rows of data.

Return parameters

List of MOS scores:

1. Composite score
2. Display FPS jerkiness
3. Display FPS jitter
4. Lipsync delay
5. Lipsync jitter
6. Latency delay
7. Latency jitter

Each value is in range 1.0 to 5.0, or NaN if no data is available to compute the score.

4 Protocol parameter values

Table 2. Values for Protocol parameters

Name	**Value**
Serial connection Baud rate	115200
Serial connection Data bits	8
Serial connection Stop bits	1
Serial connection Parity	None
Serial connection Flow control	XON/XOFF
Response timeout	TBD

5 Change history

Ver.	Status	Date	Author	Remarks
0.1	Draft	23.01.2014	RJ	First draft
0.2	Draft	17.03.2014	RJ	Draft updated according to first demo. Modification to GETD command. Green indicates commands that are implemented.
0.3	Draft	16.5.2014	RJ	Updates to data acquisition according to Lipsync implementation
0.4	Draft	26.5.2015	RJ	Adds GETCAL and SETCAL commands. Removes GETV/SETV commands.
0.5	Draft	14.1.2016	JPA	Add GETMOS command.
0.6	Draft	22.8.2016	RJ	Add Camera Framerate and Camera FPS target application specific commands. Adds BACK command. Command renaming.
0.7	Draft	4.11.2016	JL	Added Dual Camera FPS and GETMEASSTATS commands
0.8	Draft	14.2.2017	JL	VR Measurement extensions
0.9	Draft	29.1.2018	SK	VR Measurement updates, GETCONFIG and SETCONFIG commands.
1.0	Draft	23.4.2018	TB	General commands list: REBOOT, WATCHDOG, SETCONFIG, GETCONFIG
1.1	Draft	24.10.2018	TB	Video Transfer Performance specific commands added, Camera FPS removed