EN

# ShapeDrive MLAS/MLBS

**3D Sensors**

Interface Description

Available as PDF version only
Status: 16.03.2022
Version: 1.5.1
www.wenglor.com

# Table of Contents


**1. Change Index ..... 5**

**2. Document Information ..... 5**

2.1 References ..... 5

**3. Introduction ..... 6**

3.1 System Requirements ..... 6

**4. Application Example ..... 7**

**5. Defined Data Types used in SDK ..... 9**

5.1 POINT 3D ..... 9

5.2 POINT XYZI ..... 9

5.3 Point XYZIC ..... 9

**6. SDK Description ..... 10**

6.1 Connecting ShapeDrive 3D Sensor ..... 10

6.2 Closing Connection ..... 10

6.3 Get Sensor Status ..... 10

6.4 Read Property Value ..... 11

6.5 Setup ShapeDrive 3D Sensor ..... 12

6.6 Get Point Cloud ..... 13

6.7 Get Camera Image ..... 14

**7. Write Data Commands ..... 15**

7.1 Sensor Database Path ..... 15

7.2 Load Sensor Database ..... 15

7.3 Load and Save Sensor Database ..... 16

7.4 Exposure Time ..... 16

7.5 Gain ..... 16

7.6 Auto Exposure ..... 17

7.7 Resize Factor ..... 17

7.7.1 Resize Factor Value ..... 18

7.8 Phase Image Prefilter ..... 18
7.8.1 Phase Image Prefilter Kernel Size ..... 19
7.9 Confidence Filter ..... 19
7.9.1 Confidence Filter Threshold ..... 20
7.10 Contrast Comparison Filter ..... 20
7.10.1 Contrast Comparison Filter Min Phase ..... 20
7.11 Fading Compensation Filter ..... 21
7.12 Outlier Filter ..... 21
7.12.1 Outlier Filter Kernel Size ..... 21
7.12.2 Outlier Filter Threshold ..... 22
7.12.3 Outlier Filter Substitute ..... 22
7.13 PMedian Filter ..... 22
7.13.1 PMedian Filter Kernel Size ..... 23
7.14 MSAL2 Filter ..... 23
7.14.1 MSAL2 Filter Kernel Size ..... 23
7.14.2 MSAL2 Filter Iteration ..... 24
7.14.3 MSAL2 Filter Reduction Factor ..... 24
7.15 SOR Filter ..... 24
7.15.1 SOR Filter Kernel Size ..... 25
7.15.2 SOR Filter Sigma ..... 25
7.15.3 SOR Filter Enable Smoothing ..... 25
7.15.4 SOR Filter Smoothing Threshold ..... 26
7.16 Trigger Software ..... 26
7.17 Acquisition Start ..... 26
7.18 Camera Mode ..... 27
7.19 Stack Acquisition Mode ..... 27
7.20 Set LED ..... 28
7.21 LED Power ..... 28
7.22 Image Rectification ..... 28
7.23 Bounding Box ..... 28
7.24 Saving Point Cloud to Disk ..... 29
7.24.1 Point Cloud Format ..... 29
7.24.2 Saving Point Cloud ..... 29
7.25 HDR Point Cloud ..... 29
7.25.1 Enable HDR Mode ..... 29

7.25.2 Number of Point Clouds used for HDR ..... 30
7.25.3 Setting Exposure Time ..... 30
7.25.4 Setting Exposure Gain ..... 31
7.26 Save Point Cloud in TIFF Format ..... 32
7.27 Save Image Stack to File ..... 32
7.28 Load Image Stack ..... 32
7.29 Save Settings ..... 33
7.30 Load Settings ..... 33
7.31 Save Acquisition State ..... 33
7.32 Load Acquisition State ..... 33
7.33 Settings Default ..... 33
7.34 Save HDR Image Stack ..... 33
7.35 Load HDR Image Stack ..... 34
7.36 Save Camera image to File ..... 34
7.37 Mask Image ..... 34
7.37.1 Enable Mask Image ..... 34
7.37.2 Mask Image Path ..... 34
7.38 Logging ..... 35
7.38.1 Enable Logging ..... 35
7.38.2 Logging Filename ..... 35

**8. Read Data Commands ..... 35**

**9. Offline Mode ..... 37**

# 1. Change Index

| Document version | Release Date | Description | SDK version |
|---|---|---|---|
| 1.0.0 | 12.03.2019 | Initial document | 1.1.0.44 |
| 1.1.0 | 07.08.2019 | • New set commands (section 6.8.17 ...6.8.25)<br>• New read commands "GetStacksPerSecond" and "GetIntrinsicCameraParameter" | 1.2.0.200 |
| 1.2.0 | 02.12.2019 | • Introducing new stack acquisition modes in FW 1.1.0 or higher<br>• Introducing SOR filter<br>• Bug fixes and stability improvements in the DLL | 1.3.0 |
| 1.3.0 | 23.06.2020 | • New set commands (section 7)<br>• Extension get commands (section 7)<br>• New read commands (section 8)<br>• Bugfixes and stability improvements<br>• Introducing offline mode for point cloud generation without connected sensor (section 9)<br>• Introducing new filter (fading compensation filter) | 1.4.0 |
| 1.3.1 | 06.07.2020 | • Some corrections | 1.4.0 |
| 1.4.0 | 27.01.2021 | • Improved performance and stability<br>• Introducing Confidence Filter<br>• Introducing Auto Exposure<br>• Improved MultiExposure (HDR) Mode<br>• Improved Fading Compensation | 1.5.0 |
| 1.5.0 | 28.02.2022 | • Extension of error codes | 1.5.0 |
| 1.5.1 | 16.03.2022 | • Bug fix: Connect to multiple ShapeDrive cameras through one host PC | 1.5.1 |

# 2. Document Information

## 2.1 References

| Document | Version |
|---|---|
| Operating_Instructions_ShapeDrive.pdf | 1.3.1 |

# 3. Introduction

This document describes the functions and the commands for using the library to realize custom application development for the ShapeDrive product series. The library is for users who want to create their own 3D applications using the ShapeDrive 3D Sensor series.

**NOTE!**
How to install and to operate the ShapeDrive 3D sensor please refer to the ShapeDrive operating instructions.

## 3.1 System Requirements

Applications development with the library requires a Microsoft operating system (WIN7, WIN10) or Linux (tested on Ubuntu 18.04).
The ShapeDrive product series requires a Gigabit Ethernet interface card and a 10 Gbit Ethernet interface card in addition.
On windows, the user should install the Spinnaker SDK (ShapeDrive_Essential_x.x.x.zip available to download from www.wenglor.com, please refer to chapter 5.4.3 in the operation instructions of the device). On Linux, the user shoud install the ShapeDrive_Essential_Linux_x.x.x (available to download from www.wenglor.com) ShapeDrive_Essential_Linux_x.x.x provides a README file for the instructions on how to install the necessary dependency on the linux machine.
The SDKs are available for download at www.wenglor.com in the product's separate download area.

# 4. Application Example

The ShapeDrive SDK has a demo application with the source code. The demo application is given as a mean to demonstrate the data transmission from the sensor to the application using the SDK functions.
Below is a screenshot from the ShapeDrive_SDK_Qt_Cpp.

**NOTE!**
Please check functionality with the demo client before start programming.

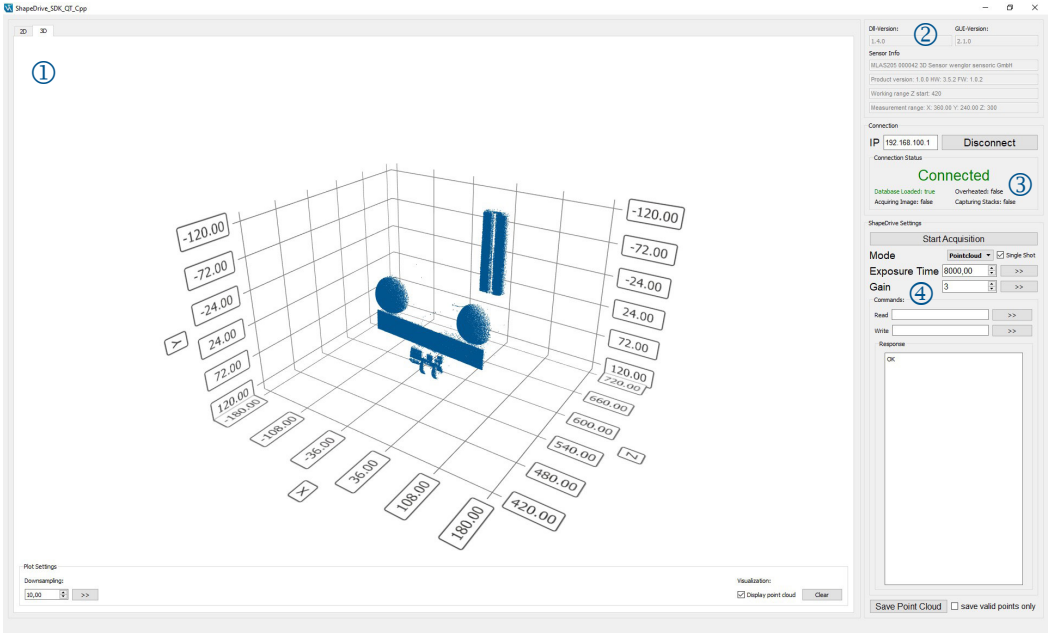



Fig. 1: Main window

① The window on the left shows the live picture or point cloud of the scanned object.

② Display of library and GUI version and information about the used 3D sensor.

③ Sensor status. Please refer to EthernetScanner3D_GetSensorStatus (see section 6.3)

④ Read/Write ShapeDrive settings

| | |
|---|---|
| **Mode** | Possible options: LiveVideo (= 2D) or PointCloud (=3D) |
| **Exposure Time** | Setup the exposure time of the device |
| **Gain** | Setup the gain value of the device |
| **Commands** | Enter Read/Write commands and get the response of the executed command |

**NOTE!**
Please check the IP address of your 3D sensor and your network settings (check the reachability of the 3D sensor in the network by pinging the sensor's IP address using the ping command in the operating system console, e.g. "ping 192.168.100.1").

**NOTE!**
You can check the connection state of the 3D sensor through the web interface. Just type the IP address of the 3D sensor in a web browser and look at the "connected to" field on the right side of the web interface.

In order to get a point cloud from the sensor in a reliable way, the host application should send the following commands in the given sequence to the sensor to build a connection:

1. Build a connection to the sensor (see EthernetScanner3D_Connect)
2. Check the connection status to make sure that the sensor is connected and the sensor database is loaded (status value should be 3), see EthernetScanner3D_GetSensorStatus

**NOTE!**
Make sure that the sensor database is located in the working directory of the application. If this is not the case please refer to the ASCII command SetLoadSdb and SetSdbPath.

3. Setup the ShapeDrive sensor to 3D acquisition mode (see SetCameraMode=1 and EthernetScanner3D_WriteData)
4. Start the data acquisition (see SetAcquisitionStart=1 and EthernetScanner3D_WriteData)
5. Trigger the stack acquisition per software command (see SetTriggerSoftware=1 and EthernetScanner3D_WriteData)
6. Compute and read the point cloud from the sensor (see EthernetScanner3D_GetPointCloud)
7. Stop the acquisition (see SetAcquisitionStart=0 and EthernetScanner3D_WriteData).
8. Disconnect from sensor before ending the application (EthernetScanner3D_Disconnect).

**NOTE!**
You can download the sensor database either directly from the sensor web interface or using the command SetLoadSdbFromSensor.

# 5. Defined Data Types used in SDK

In the SDK following data types are defined:

## 5.1 POINT 3D

| Data type | POINT 3D |
|---|---|
| Data field 1 | double x : x coordinate of a 3D point in mm |
| Data field 2 | double y : y coordinate of a 3D point in mm |
| Data field 3 | double z : z coordinate of a 3D point in mm |
| Description | Contains the coordinates of a point in a 3D point cloud. |

## 5.2 POINT XYZI

| Data type | POINT XYZI |
|---|---|
| Data field 1 | POINT3D *pointcloud : Pointer to points contained in point cloud |
| Data field 2 | uint16_t *intensity : Pointer to the intensity value of each point in the point cloud |
| Description | Contains the coordinates and intensity of a point in a 3D point cloud. |

## 5.3 Point XYZIC

| Data type | POINT XYZIC |
|---|---|
| Data field 1 | POINT3D *pointcloud : Pointer to points contained in pointcloud |
| Data field 2 | unit16_t *intensity : Pointer to the intensity value of each point in the point cloud |
| Data field 3 | float *confidence : Pointer to the confidence value of each point in the point cloud |
| Description | Contains coordinates intensity and confidence value of a point in the point cloud |

# 6. SDK Description

## 6.1 Connecting ShapeDrive 3D Sensor

| | |
|---|---|
| **Command** | void* EthernetScanner3D_Connect(const char *chIP, const char * chPort, int iTimeOut) |
| **Parameter 1** | const char *chIP: The IP address of the sensor |
| **Parameter 2** | const char * chPort: The port number of the sensor (default value 32001) |
| **Parameter 3** | int iTimeOut: Timeout in [ms] before the operation is cancelled |
| **Response** | void* a handle to the sensor. A NULL pointer is returned in case of failure |
| **Description** | This function will create a connection to the ShapeDrive 3D Sensor. The function will return a handle to the sensor which will be used by other functions. |

## 6.2 Closing Connection

| | |
|---|---|
| **Command** | int EthernetScanner3D_Disconnect(void* pEthernetScanner3D) |
| **Parameter 1** | void* pEthernetScanner3D: The sensor handle |
| **Response** | The function returns:<br>• ETHERNETSCANNER3D_OK (0) in case of successful operation<br>• ETHERNETSCANNER3D_SENSORHANDLENULLPOINTER (-3) if the sensor handle is a null pointer<br>• ETHERNETSCANNER3D_UNKNOWNSENSORHANDLE (-2) if the pointer is not the sensor handle<br>• ETHERNETSCANNER3D_SENSORCOMMANDTIMEOUT (-6) if the timeout is reached |
| **Description** | This function closes the connection to the ShapeDrive 3D sensor. |

## 6.3 Get Sensor Status

| | |
|---|---|
| **Command** | int EthernetScanner3D_GetSensorStatus(void* pEthernetScanner3D, int *status) |
| **Parameter 1** | void *pEthernetScanner3D: The sensor handle |
| **Parameter 2** | int *status: A pointer to int variable which returns the sensor status bitwise decoded as follows:<br>bit0: SENSOR_CONNECTED<br>bit1: SENSOR_DATABASE_LOADED<br>bit2: SENSOR_ACQUISITION_STARTET<br>bit3: SENSOR_OVERHEATED<br>bit4: SENSOR_CAPTURING_STACKS<br>bit5: SENSOR_OFFLINE_MODE<br>bit6...bit31: not used |
| **Response** | ETHERNETSCANNER3D_OK (0)<br>ETHERNETSCANNER3D_UNKNOWNSENSORHANDLE (-2) |
| **Description** | Requests the current status of the ShapeDrive 3D Sensor. |

**NOTE!**
Starting from SDK version 1.4.0 and higher, the SDK validates the database in the working directory by comparing it with the database on the ShapeDrive 3D sensor. If the validation fails then bit1 (SENSOR_DATABASE_LOADED) in status will be reset (0).
Reason for a failed validation could be either the SDK does not find the database in the working directory, or the database in the working directory does not match the one on the device.

## 6.4 Read Property Value

| | |
|---|---|
| **Command** | int EthernetScanner3D_ReadData(void *pEthernetScanner3D, const char *chCommand, char *cRetBuf, int iRetBuf, int iTimeOut) |
| **Parameter 1** | void *pEthernetScanner3D: The sensor handle |
| **Parameter 2** | const char *chCommand: The command to read data |
| **Parameter 3** | char *cRetBuf: A pointer to the return buffer |
| **Parameter 4** | int iRetBuf: The buffer size of cRetBuf. A buffer size of 128 Byte × 1024 = 131072 Byte is a standard value. |
| **Parameter 5** | int iTimeOut: not used. Keep it (0) |
| **Response** | The function returns:<br>• ETHERNETSCANNER3D_OK (0) in case of successful operation<br>• ETHERNETSCANNER3D_SENSORHANDLENULLPOINTER (-3) if the sensor handle is a null pointer<br>• ETHERNETSCANNER3D_UNKNOWNSENSORHANDLE (-2) if the pointer is not the sensor handle<br>• ETHERNETSCANNER3D_ARGUMENTNULLPOINTER (-11) if one of the arguments is a null pointer<br>• ETHERNETSCANNER3D_RETURNBUFFERTOOSMALL (-13) if the return buffer is too small<br>• ETHERNETSCANNER3D_SENSORNOTCONNECTED (-4) if the sensor is not connected<br>• ETHERNETSCANNER3D_SENSORCOMMANDTIMEOUT (-6) if the timeout is reached<br>• ETHERNETSCANNER3D_UNKNOWNCOMMAND (-10) if the command does not exist |
| **Description** | This function is used to get property values from the SDK and the sensor. |

## 6.5 Setup ShapeDrive 3D Sensor

| | |
|---|---|
| **Command** | int EthernetScanner3D_WriteData(void *pEthernetScanner3D, const char *chCommand, int iTimeOut) |
| **Parameter 1** | void *pEthernetScanner3D: The sensor handle |
| **Parameter 2** | const char *chCommand: The command to write data |
| **Parameter 3** | int iTimeOut: not used. Keep it (0) |
| **Response** | The function returns:<br>• ETHERNETSCANNER3D_OK (0) in case of successful operation<br>• ETHERNETSCANNER3D_SENSORHANDLENULLPOINTER (-3) if the sensor handle is a null pointer<br>• ETHERNETSCANNER3D_UNKNOWNSENSORHANDLE (-2) if the pointer is not the sensor handle<br>• ETHERNETSCANNER3D_ARGUMENTNULLPOINTER (-11) if one of the arguments is a null pointer<br>• ETHERNETSCANNER3D_SENSORNOTCONNECTED (-4) if the sensor is not connected<br>• ETHERNETSCANNER3D_SENSORCOMMANDTIMEOUT (-6) if the timeout is reached<br>• ETHERNETSCANNER3D_NOIMAGESTACK (-101) if one or more images in the stack are missing (after writing the ASCII command SetTriggerSoftware=1)<br>• ETHERNETSCANNER3D_SENSOROVERHEATED (-5) if the sensor is overheated<br>• ETHERNETSCANNER3D_UNKNOWNCOMMAND (-10) if the command does not exist<br>• ETHERNETSCANNER3D_ARGUMENTOUTOFRANGE (-12) if the set value is not in the defined range (SetAcquisitionStart can only be used with parameter 0 or 1)<br>» Example: EthernetScanner3D_WriteData(scanner_, "SetAcquisitionStart=5" will return ETHERNETSCANNER3D_ARGUMENTOUTOFRANGE |
| **Description** | This function is used to set property values to the SDK and the sensor |

## 6.6 Get Point Cloud

| | |
|---|---|
| **Command** | int EthernetScanner3D_GetPointCloud(void *pEthernetScanner3D, void *pRetBuf, int iRetBuf, const char* chMode, int iTimeOut) |
| **Parameter 1** | void *pEthernetScanner3D: The sensor handle |
| **Parameter 2** | void *pRetBuf: A point to the return buffer containing the point cloud. The point cloud is in the format of the data type POINT_XYZI (s. section 5.2). |
| **Parameter 3** | int iRetBuf: The return buffer size in bytes. The minium size is given by the image size. To determine the image size use the read commands GetRoi1Width and GetRoi1Height.<br>The minimum size of the buffer is give by<br>iRetBuf=width × height × (sizeof(POINT3D) + sizeof(uint16_t)) |
| **Parameter 4** | const char* chMode: default (nullptr): PointcloudFormat XYZI (see section 5.2)<br>"XYZIC": PointcloudFormat PointXYZIC (see section 5.3) |
| **Parameter 5** | int iTimeOut: Timeout in [ms] before the operation is cancelled |
| **Response** | The function returns:<br>• ETHERNETSCANNER3D_OK (0) in case of successful operation<br>• ETHERNETSCANNER3D_SENSORHANDLENULLPOINTER (-3) if the sensor handle is a null pointer<br>• ETHERNETSCANNER3D_UNKNOWNSENSORHANDLE (-2) if the pointer is not the sensor handle<br>• ETHERNETSCANNER3D_ARGUMENTNULLPOINTER (-11) if one of the arguments is a null pointer<br>• ETHERNETSCANNER3D_RETURNBUFFERTOOSMALL (-13) if the return buffer is too small<br>• ETHERNETSCANNER3D_SENSORNOTCONNECTED (-4) if the sensor is not connected<br>• ETHERNETSCANNER3D_SENSORCOMMANDTIMEOUT (-6) if the timeout is reached<br>• ETHERNETSCANNER3D_IMAGESTACKNOTCOMPLETE (-102) if there is no or a corrupted image stack<br>• ETHERNETSCANNER3D_SENSORDATABASENOTLOADED (-105) if the sensor data base is not loaded<br>• ETHERNETSCANNER3D_WRONGACQUISITIONMODE (-32) if the sensor is in Live Camera Mode |
| **Description** | This function is used to get the point cloud data. The ShapeDrive 3D sensor should be in stack mode (3D mode, please refer to section 7.18). |

## 6.7 Get Camera Image

| | |
|---|---|
| **Command** | int EthernetScanner3D_GetCameraImage(void *pEthernetScanner3D, void *pRetBuf, int iRetBuf, const char* chMode, int iTimeOut) |
| **Parameter 1** | void *pEthernetScanner3D: The sensor handle |
| **Parameter 2** | void *pRetBuf: A pointer to the return buffer |
| **Parameter 3** | int iRetBuf: The return buffer size in bytes |
| **Parameter 4** | const char* chMode: Currently unused |
| **Parameter 5** | int iTimeOut: Timeout in [ms] before the operation is cancelled |
| **Response** | The function returns:<br>• ETHERNETSCANNER3D_OK (0) in case of successful operation<br>• ETHERNETSCANNER3D_SENSORHANDLENULLPOINTER (-3) if the sensor handle is a null pointer<br>• ETHERNETSCANNER3D_UNKNOWNSENSORHANDLE (-2) if the pointer is not the sensor handle<br>• ETHERNETSCANNER3D_ARGUMENTNULLPOINTER (-11) if one of the arguments is a null pointer<br>• ETHERNETSCANNER3D_RETURNBUFFERTOOSMALL (-13) if the return buffer is too small<br>• ETHERNETSCANNER3D_SENSORNOTCONNECTED (-4) if the sensor is not connected<br>• ETHERNETSCANNER3D_SENSORCOMMANDTIMEOUT (-6) if the timeout is reached<br>• ETHERNETSCANNER3D_WRONGACQUISITIONMODE (-32) if sensor is in ImageStack Mode |
| **Description** | This function is used to get the camera image data. The ShapeDrive 3D sensor should be in live picture mode (please refer to section 7.18). |

**NOTE!**
If the error code
ETHERNETSCANNER3D_UNHANDLEDERROR (-1000)
is returned, please contact support@wenglor.com

# 7. Write Data Commands

The commands are used together with the function EthernetScanner3D_WriteData (see section 6.5). If available the corresponding Get commands are listed below each Set command.

E.g. for many settings the following information can be requested:

- Current: Returns the current set value
- Default: Returns the default set value given by the factory setting
- Min: Returns the smallest allowed value for the command
- Max: Returns the largest allowed value for the command

*Examples:*

*To get minimum exposure time use e.g.:*

```
EthernetScanner3D_ReadData(device_, "GetExposureTimeMin", result, 100, 0)
```

*To set the exposure time use e.g.:*

```
EthernetScanner3D_WriteData(device_, "SetExposureTime=1000", 0)
```

**NOTE!**
To obtain the allowed value range for the command request "Min" and "Max" ensure that you set only valid parameters in the given range.

## 7.1 Sensor Database Path

| | |
|---|---|
| **Command** | SetSdbPath=path |
| **Parameter** | String with file path to database |
| **Description** | Sets the sensor database path if the database is locatedd outside the working directory of the application. |
| **Get Commands** | GetSdbPath |

## 7.2 Load Sensor Database

| | |
|---|---|
| **Command** | SetLoadSdb |
| **Parameter** | No parameter (in OfflineMode: filename or filepath+name of database) |
| **Description** | Searches sensor database of connected sensor by its name (e.g. SDB_MLAS101_000042.bin) in the working directory or the directory specified in section 7.1 and loads it to the SDK.<br>In OFFLINE MODE the function is used to initialize emulation of its corresponding sensor (see section 9). Therefore in this case the filename of the database has to be set as parameter (e.g. SetLoadSdb=C:\SDB_MLAS101_000042.bin). |
| **Get Commands** | --- |

**NOTE!**
Starting from SDK version 1.4.0 or higher, the SDK validates the database in the working directory by comparing it with the database on the ShapeDrive 3D sensor. In SDK version 1.5.0, if the validation fails then the function EthernetScanner3D_WriteData returns error code
ETHERNETSCANNNER3D_WRONGSENSORDATABASE (-106). If no corresponding database file was found ETHERNETSCANNNER3D_SENSORDATABASENOTLOADED (-105) is returned.
Reason for a failed validation could be either the SDK does not find the database in the working directory, or the database in the working directory does not match the one on the device.

## 7.3 Load and Save Sensor Database

| | |
|---|---|
| **Command** | SetLoadSdbFromSensor |
| **Parameter** | No parameter |
| **Description** | Loads the sensor database from the sensor and saves it in the working directory on the hard disk. |
| **Get Commands** | --- |

## 7.4 Exposure Time

| | |
|---|---|
| **Command** | SetExposureTime=x |
| **Parameter** | To obtain the value range and the default use the corresponding Get commands. Only integer values are allowed. |
| **Description** | Exposure time is set in $\mu$s. |
| **Get Commands** | GetExposureTimeCurrent<br>GetExposureTimeDefault<br>GetExposureTimeMin<br>GetExposureTimeMax |

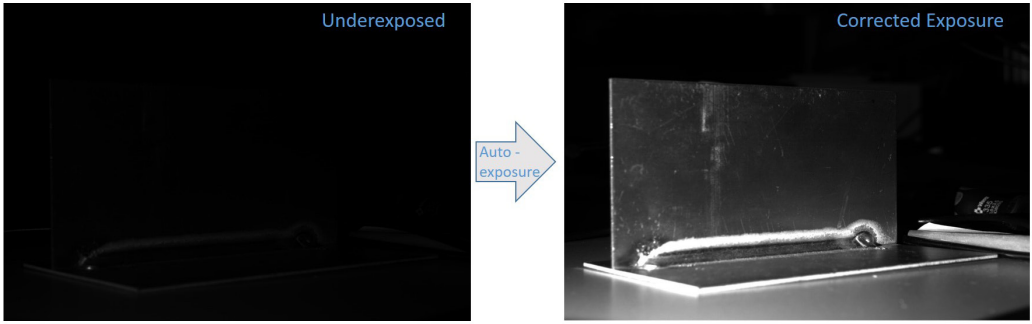
## 7.5 Gain

| | |
|---|---|
| **Command** | SetGain=x |
| **Parameter** | To obtain the value range and the default use the corresponding Get commands. Only integer values are allowed. |
| **Description** | Gain is set in dB |
| **Get Commands** | GetGainCurrent<br>GetGainDefault<br>GetGainMin<br>GetGainMax |

## 7.6 Auto Exposure

| Command | SetAutoExposure |
|---|---|
| Description | Computes and applies optimum exposure time and gain for the given scene. The adapted exposure time and gain can be obtained by the corresponding get commands for gain and exposure time. The auto exposure computation is only carried out once, immediately after the command was sent. If the scene has changed SetAutoExposure has to be called again. |

Example:




## 7.7 Resize Factor

| Command | SetSDKResizeFactor=x |
|---|---|
| Parameter | x: 0 or 1 |
| Description | 0: Disable resize factor<br>1: Enable resize factor<br>This switch reduces the number of pixels by a factor given by SetSDKResizeFactorValue (Pseudo binning). The reduction leads to a shorter precessing time calculating the 3D point cloud. |
| Get Commands | GetSDKResizeFactorCurrent<br>GetSDKResizeFactorDefault<br>GetSDKResizeFactorMin<br>GetSDKResizeFactorMax |

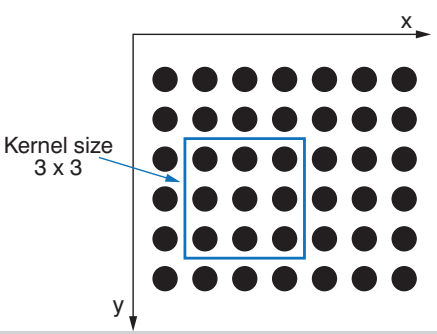
### 7.7.1 Resize Factor Value

| | |
|---|---|
| **Command** | SetSDKResizeFactorValue=x |
| **Parameter** | To obtain the value range and the default use the corresponding Get commands. Only integer values are allowed. |
| **Description** | If set to 2 only each 2nd point in X and Y direction is considered for further processing, i. e. the amount of data is reduced by a factor of 4. Correspondingly if set to 3, the amount of data is reduced by a factor of 9. |
| **Get Commands** | GetSDKResizeFactorValueCurrent<br>GetSDKResizeFactorValueDefault<br>GetSDKResizeFactorValueMin<br>GetSDKResizeFactorValueMax |

## 7.8 Phase Image Prefilter

| | |
|---|---|
| **Command** | SetSDKPhaseImagePrefilter=x |
| **Parameter** | x: 0 or 1 |
| **Description** | 0: Disable phase image prefilter<br>1: Enable phase image prefilter<br>This filter can be used to average out high frequency intensity noise as for highly textured surfaces like polished metal surfaces with fine scratches or grooves in the scene. Using this filter the 3D point cloud is more smoothed. If enabled it increases the processing time of the point cloud calculation. |
| **Get Commands** | GetSDKPhaseImagePrefilterCurrent<br>GetSDKPhaseImagePrefilterDefault<br>GetSDKPhaseImagePrefilterMin<br>GetSDKPhaseImagePrefilterMax |

### 7.8.1 Phase Image Prefilter Kernel Size

| | |
|---|---|
| **Command** | SetSDKPhaseImagePrefilterKernelSize=x |
| **Parameter** | To obtain the value range and the default value use the corresponding Get commands.<br>Only odd integer values are allowed. |
| **Description** | For example, a kernel size value of 3 creates a 3x3 average filter:<br> |
| **Get Commands** | GetSDKPhaseImagePrefilterKernelSizeCurrent<br>GetSDKPhaseImagePrefilterKernelSizeDefault<br>GetSDKPhaseImagePrefilterKernelSizeMin<br>GetSDKPhaseImagePrefilterKernelSizeMax |

## 7.9 Confidence Filter

| | |
|---|---|
| **Command** | SetSDKConfidenceFilter=x |
| **Parameter** | x: 0 or 1 |
| **Description** | 0: Disable confidence filter<br>1: Enable confidence filter<br>The confidence filter is used to filter out points with a confidence value below the threshold set by SetSDKConfidenceFilterThreshold. It is used to remove undesired points e.g. with artefacts or measurement errors due to environmental or geometric reasons. If enabled, a confidence value is computed for each pixel representing the reliability of the phase estimation. This can increase the pointcloud calculations processing time. |
| **Get Commands** | GetSDKConfidenceFilterCurrent<br>GetSDKConfidenceFilterDefault<br>GetSDKConfidenceFilterMin<br>GetSDKConfidenceFilterMax |

### 7.9.1 Confidence Filter Threshold

| | |
|---|---|
| **Command** | SetSDKConfidenceFilterThreshold=x |
| **Parameter** | Floating point value between 0 and 1 |
| **Description** | Threshold setting the minimum required confidence for a point to be considered valid. The value has the format of a probability between 0 and1 (e.g. 0.3). |
| **Get Commands** | GetSDKConfidenceFilterThresholdCurrent<br>GetSDKConfidenceFilterThresholdDefault<br>GetSDKConfidenceFilterThresholdMin<br>GetSDKConfidenceFilterThresholdMax |

## 7.10 Contrast Comparison Filter

| | |
|---|---|
| **Command** | SetSDKContrastComparisonFilter=x |
| **Parameter** | x: 0 or 1 |
| **Description** | 0: Disable contrast comparison filter<br>1: Enable contrast comparison filter<br>The contrast comparison filter compares for a given point the minimum and the maximum value of the phase images. If the contrast is below the setting in section 7.10.1 the point is rejected. If enabled it increases the processing time of the point cloud calculation. |
| **Get Commands** | GetSDKContrastComparisonFilterCurrent<br>GetSDKContrastComparisonFilterDefault<br>GetSDKContrastComparisonFilterMin<br>GetSDKContrastComparisonFilterMax |

### 7.10.1 Contrast Comparison Filter Min Phase

| | |
|---|---|
| **Command** | SetSDKContrastComparisonFilterMinPhase=x |
| **Parameter** | To obtain the allowed value range and the default value use the corresponding Get commands. Only integer values are allowed. |
| **Description** | Sets the value for the phase contrast. A value of 0.1 and above is recommended. |
| **Get Commands** | GetSDKContrastComparisonFilterMinPhaseCurrent<br>GetSDKContrastComparisonFilterMinPhaseDefault<br>GetSDKContrastComparisonFilterMinPhaseMin<br>GetSDKContrastComparisonFilterMinPhaseMax |

## 7.11 Fading Compensation Filter

| | |
|---|---|
| **Command** | SetSDKFadingCompensation=x |
| **Parameter** | x: 0 or 1 |
| **Description** | 0: Disable fading compensation filter<br>1: Enable fading compensation filter<br>The fading compensation is an algorithm to handle not stable light output of the LED during the image stack acquisition. |
| **Get Commands** | GetSDKFadingCompensationCurrent<br>GetSDKFadingCompensationDefault<br>GetSDKFadingCompensationMin<br>GetSDKFadingCompensationMax |

**NOTE!**
In the old ShapeDrive SDK version 1.3.0, the default value of fading compensation filter was 1. In ShapeDrive SDK version 1.4.0 and above, the default value is 0.
In the SDK version earlier than 1.4.0 the fading compensation was automatically activated. Because the fading compensation can lead to artefacts it is now deactivated by default and can be activated by the user.

## 7.12 Outlier Filter

| | |
|---|---|
| **Command** | SetSDKOutlierFilter=x |
| **Parameter** | x: 0 or 1 |
| **Description** | 0: Disable outlier filter<br>1: Enable outlier filter<br>The outlier filter is used to remove/replace points which deviates significantly from the surrounding points. If enabled it increases the processing time of the point cloud calculation. |
| **Get Commands** | GetSDKOutlierFilterCurrent<br>GetSDKOutlierFilterDefault<br>GetSDKOutlierFilterMin<br>GetSDKOutlierFilterMax |

### 7.12.1 Outlier Filter Kernel Size

| | |
|---|---|
| **Command** | SetSDKOutlierFilterKernelSize=x |
| **Parameter** | To obtain the value range and the default value use the corresponding Get commands. Only odd integer values are allowed. |
| **Description** | Defines the kernel area in which to look for outliers, e.g. if set to 5, the area is 5x5 points. |
| **Get Commands** | GetSDKOutlierFilterKernelSizeCurrent<br>GetSDKOutlierFilterKernelSizeDefault<br>GetSDKOutlierFilterKernelSizeMin<br>GetSDKOutlierFilterKernelSizeMax |

### 7.12.2 Outlier Filter Threshold

| | |
|---|---|
| **Command** | SetSDKOutlierFilterThreshold=x |
| **Parameter** | To obtain the value range and the default use the corresponding Get command. Floating point values are allowed. |
| **Description** | Threshold value in mm. If the center point of the considered kernel area exceeds this value it is either labeled invalid or subsituted by the median value of the area (see Outlier Filter Substitute). |
| **Get Commands** | GetSDKOutlierFilterThresholdCurrent<br>GetSDKOutlierFilterThresholdDefault<br>GetSDKOutlierFilterThresholdMin<br>GetSDKOutlierFilterThresholdMax |

### 7.12.3 Outlier Filter Substitute

| | |
|---|---|
| **Command** | SetSDKOutlierFilterSubstitute=x |
| **Parameter** | x: 0 or 1 |
| **Description** | 0: Point is labeled invalid<br>1: Point is substituted by the median value of the area |
| **Get Commands** | GetSDKOutlierFilterSubstituteCurrent<br>GetSDKOutlierFilterSubstituteDefault<br>GetSDKOutlierFilterSubstituteMin<br>GetSDKOutlierFilterSubstituteMax |

## 7.13 PMedian Filter

| | |
|---|---|
| **Command** | SetSDKPMedianFilter=x |
| **Parameter** | x: 0 or 1 |
| **Description** | 0: Disable PMedian filter<br>1: Enable PMedian filter<br>Sorts the Z-value of all points in a range defined by SetSDKPMedianFilterKernelSize and replaces the focused point with the median value of the surrounding points. If enabled it increases the processing time of the point cloud calculation. |
| **Get Commands** | GetSDKPMedianFilterCurrent<br>GetSDKPMedianFilterDefault<br>GetSDKPMedianFilterMin<br>GetSDKPMedianFilterMax |

### 7.13.1 PMedian Filter Kernel Size

| | |
|---|---|
| **Command** | SetSDKPMedianFilterKernelSize=x |
| **Parameter** | To obtain the value range and the default value use the corresponding Get commands. Only odd integer values are allowed. |
| **Description** | For example, a kernel size value of 3 creates a 3x3 median filter. A typical value is 5. The value can be increased for surfaces with a high dynamic like twinkling surfaces. It smoothes the point cloud which smoothes also edges. |
| **Get Commands** | GetSDKPMedianFilterKernelSizeCurrent<br>GetSDKPMedianFilterKernelSizeDefault<br>GetSDKPMedianFilterKernelSizeMin<br>GetSDKPMedianFilterKernelSizeMax |

## 7.14 MSAL2 Filter

| | |
|---|---|
| **Command** | SetSDKMSAL2Filter=x |
| **Parameter** | x: 0 or 1 |
| **Description** | 0: Disable MSAL2 filter<br>1: Enable MSAL2 filter<br>This filter works on all coordinates and smoothes out noisy surfaces without removing points but instead shifting points to more likely positions. If enabled it increases the processing time of the point cloud calculation. |
| **Get Commands** | GetSDKMSAL2FilterCurrent<br>GetSDKMSAL2FilterDefault<br>GetSDKMSAL2FilterMin<br>GetSDKMSAL2FilterMax |

### 7.14.1 MSAL2 Filter Kernel Size

| | |
|---|---|
| **Command** | SetSDKMSAL2FilterKernelSize=x |
| **Parameter** | To obtain the value range and the default use the corresponding Get commands. Only odd integer values are allowed. |
| **Description** | Defines how many nearest neighbours are considered, typically it should be 9 [9...21]. |
| **Get Commands** | GetSDKMSAL2FilterKernelSizeCurrent<br>GetSDKMSAL2FilterKernelSizeDefault<br>GetSDKMSAL2FilterKernelSizeMin<br>GetSDKMSAL2FilterKernelSizeMax |

### 7.14.2 MSAL2 Filter Iteration

| | |
|---|---|
| **Command** | SetSDKMSAL2FilterIteration=x |
| **Parameter** | To obtain the value range and the default use the corresponding Get commands. Only integer values are allowed. |
| **Description** | Number of iteration. Typically, the more iterations, the smoother the result, however the runtime goes up with each iteration. A typical value is 4. |
| **Get Commands** | GetSDKMSAL2FilterIterationCurrent<br>GetSDKMSAL2FilterIterationDefault<br>GetSDKMSAL2FilterIterationMin<br>GetSDKMSAL2FilterIterationMax |

### 7.14.3 MSAL2 Filter Reduction Factor

| | |
|---|---|
| **Command** | SetSDKMSAL2FilterReductionFactor=x |
| **Parameter** | To obtain the value range and the default use the corresponding Get commands. Only integer values are allowed. |
| **Description** | Reduces the input point cloud size in order speed up filter runtime. E.g. if x=2 the point size is reduced by a factor of 4. |
| **Get Commands** | GetSDKMSAL2FilterReductionFactorCurrent<br>GetSDKMSAL2FilterReductionFactorDefault<br>GetSDKMSAL2FilterReductionFactorMin<br>GetSDKMSAL2FilterReductionFactorMax |

## 7.15 SOR Filter

| | |
|---|---|
| **Command** | SetSDKSORFilter=x |
| **Parameter** | x: 0 or 1 |
| **Description** | 0: Disable SOR filter<br>1: Enable SOR filter<br>The Statistical Outlier Removal (SOR) Filter removes sparse outliers. Therefore for each point the average point to neighbours distance within a defined kernel (see section 7.15.1) is computed. Assuming a Gaussian distribution all points with a mean distance above the threshold defined by the global distances mean and n times the standard deviation sigma (see section 7.15.2) are rejected. |
| **Get Commands** | GetSDKSORFilterCurrent<br>GetSDKSORFilterDefault<br>GetSDKSORFilterMin<br>GetSDKSORFilterMax |

### 7.15.1 SOR Filter Kernel Size

| | |
|---|---|
| **Command** | SetSDKSORFilterKernelSize=x |
| **Parameter** | To obtain the allowed value range and the default value use the corrensponding Get commands. Only odd integer values are allowed. |
| **Description** | Specifies the size of a quadratic kernel in pixels. The kernel determines how many neighbours of a point are used to compute its average point to neighbour distance. E.g. a kernel size value of 3 creates a 3x3 filter window. A typical value is 9. |
| **Get Commands** | GetSDKSORFilterKernelSizeCurrent<br>GetSDKSORFilterKernelSizeDefault<br>GetSDKSORFilterKernelSizeMin<br>GetSDKSORFilterKernelSizeMax |

### 7.15.2 SOR Filter Sigma

| | |
|---|---|
| **Command** | SetSDKSORFilterSigma=x |
| **Parameter** | To obtain the allowed value range and the default value use the corresponding Get commands. Only integer values are allowed. |
| **Description** | Multiplier which specifies how much times the standard deviation is accepted as a valid deviation from point clouds average point to neighbour distance. Points which have a higher deviation are rejected.<br>RejectionThreshold=avgDist+**x***StandardDeviation |
| **Get Commands** | GetSDKSORFilterSigmaCurrent<br>GetSDKSORFilterSigmaDefault<br>GetSDKSORFilterSigmaMin<br>GetSDKSORFilterSigmaMax |

### 7.15.3 SOR Filter Enable Smoothing

| | |
|---|---|
| **Command** | SetSDKSORFilterSmoothingEnabled=x |
| **Parameter** | x: 0 or 1 |
| **Description** | 0: Disable smoothing<br>1: Enable smoothing<br>Enables or disables smoothing of point cloud.<br>If enabled points which have a Z distance below the smoothing threshold will be set to the kernels average Z value (smoothing has no effect on the rejection carried out by the SOR filter). |
| **Get Commands** | GetSDKSORFilterSmoothingEnabledCurrent<br>GetSDKSORFilterSmoothingEnabledDefault<br>GetSDKSORFilterSmoothingEnabledMin<br>GetSDKSORFilterSmoothingEnabledMax |

### 7.15.4 SOR Filter Smoothing Threshold

| | |
|---|---|
| **Command** | SetSDKSORFilterSmoothingThreshold=x |
| **Parameter** | To obtain the value range and the default value use the corresponding Get commands. Floating point values are allowed. |
| **Description** | Smoothing threshold which specifies the maximum deviation from the kernels average Z value in Z direction in mm. Points with deviations below this threshold are set to the kernels average Z value. |
| **Get Commands** | GetSDKSORFilterSmoothingThresholdCurrent<br>GetSDKSORFilterSmoothingThresholdDefault<br>GetSDKSORFilterSmoothingThresholdMin<br>GetSDKSORFilterSmoothingThresholdMax |

## 7.16 Trigger Software

| | |
|---|---|
| **Command** | SetTriggerSoftware=x |
| **Parameter** | x: 1 |
| **Description** | Needed to start a 3D measurement. |
| **Response** | ETHERNETSCANNER3D_FIFOFULL (-100) if the number of already acquired but not yet processed image stacks is higher than 5. |
| **Get Commands** | --- |

## 7.17 Acquisition Start

| | |
|---|---|
| **Command** | SetAcquisitionStart=x |
| **Parameter** | x: 0 or 1 |
| **Description** | 0: Stop acquisition<br>1: Start acquisition<br>If Camera Mode (see section 7.18) is set to 0 (Live picture) the measurement starts immediately. If set to 1 (Stack) SetTriggerSoftware is needed. |
| **Response** | Stop acquisition:<br>ETHERNETSCANNER3D_ACQUISITIONALREADYSTOPPED ( -31) if the acquisition has already been stopped.<br>Start acquisition:<br>ETHERNETSCANNER3D_ACQUISITIONALREADYSTARTED ( -30) if the acquisition has already been started. |
| **Get Commands** | --- |

## 7.18 Camera Mode

| | | | |
|---|---|---|---|
| **Command** | SetCameraMode=x | | |
| **Parameter** | x: 0 or1 | **Default:** | 1 |
| **Description** | 0: Live picture<br>1: Stack (mode for 3D measurement) | | |
| **Get Commands** | GetCameraModeCurrent<br>GetCameraModeDefault<br>GetCameraModeMin<br>GetCameraModeMax | | |

## 7.19 Stack Acquisition Mode

| | | | |
|---|---|---|---|
| **Command** | SetStackAcquisitionMode=x | | |
| **Parameter** | x: 1 ... 6 | **Default:** | 1 |
| **Description** | In the old FW 1.0.x there are only two supported modes:<br>1: Mode 1: (28 images per stack, maximum exposure time 20000 µs)<br>2: Mode 2: (16 images per stack, maximum exposure time 20000 µs)<br>In the new FW 1.1.x or higher there are six different stack acquisition modes:<br>1: Mode 1: (28 images per stack, maximum exposure time 20000 µs for MLxS1xx with 5 MPix camera and 26000 µs for MLxS2xx with 12 MPix camera)<br>2: Mode 2: (16 images per stack, maximum exposure time 20000 µs for MLxS1xx with 5 MPix camera and 26000 µs for MLxS2xx with 12 MPix camera)<br>3: Mode 3: (28 images per stack, maximum exposure time 40000 µs)<br>4: Mode 4: (16 images per stack, maximum exposure time 40000 µs)<br>5: Mode 5: (28 images per stack, maximum exposure time 75000 µs)<br>6: Mode 6: (16 images per stack, maximum exposure time 75000 µs) | | |
| **Get Commands** | GetStackAcquisitionModeCurrent<br>GetStackAcquisitionModeDefault<br>GetStackAcquisitionModeMin<br>GetStackAcquisitionModeMax | | |

**NOTE!**
The command SetStackAcquisitionMode is a blocking command. The execution of the command could take up to 8 sec. (e.g. changing the stack acquisition mode from mode 1 to mode 5). During the execution of the command, the ShapeDrive 3D sensor does not process any furrther commands until it finishes processing the SetStackAcquisitionMode command. The ShapeDrive 3D Sensor will ignore executing the command if the device is already in the same stack acquisition mode.
It is recommended to call this command right before calling the command SetAcquisition-Start.

## 7.20 Set LED

| | | | |
|---|---|---|---|
| **Command** | SetLED=x | | |
| **Parameter** | x: 0 or1 | **Default:** | 1 |
| **Description** | 0: Disable LED<br>1: Enable LED<br>Enables or disables the LED light temporarily while acquiring camera images. The lighting is reset to enabled after acquisition has been started again. It has to be called every time after camera acquisition is started using SetAcquisitionStart in order to disable lighting. | | |
| **Get Commands** | --- | | |

## 7.21 LED Power

| | |
|---|---|
| **Command** | SetLEDPower=x |
| **Parameter** | To obtain the value range and the default use the corresponding Get commands. Only integer values are allowed. |
| **Description** | The command sets the LED power. |
| **Get Commands** | GetLEDPowerCurrent<br>GetLEDPowerDefault<br>GetLEDPowerMin<br>GetLEDPowerMax |

## 7.22 Image Rectification

| | | | |
|---|---|---|---|
| **Command** | SetCameraImageDistortion=x | | |
| **Parameter** | x: normal or undistorted | **Default:** | normal |
| **Description** | The 2D camera image is rectified if undistorted is set. | | |
| **Get Commands** | GetCameraImageDistortion | | |

## 7.23 Bounding Box

| | | | |
|---|---|---|---|
| **Command** | SetBoundingBoxEnabled=x | | |
| **Parameter** | x: 0 or 1 | **Default:** | 1 |
| **Description** | 0: Bounding box disabled<br>1: Bounding box enabled<br>Option to remove bounding box which limits the measurement volume to the volume with the best calibration (is equal to the data sheet volume). Points outside of the measurement volume given in the data sheet are less accurate. | | |
| **Get Commands** | GetBoundingBoxEnabledCurrent<br>GetBoundingBoxEnabledDefault<br>GetBoundingBoxEnabledMin<br>GetBoundingBoxEnabledMax | | |

## 7.24 Saving Point Cloud to Disk

### 7.24.1 Point Cloud Format

| | | | |
|---|---|---|---|
| **Command** | SetPointCloudFormat=x | | |
| **Parameter** | x: X, Y, Z, I, C, NN | **Default:** | XYZ |
| **Description** | X: x-coordinate in mm<br>Y: y-coordinate in mm<br>Z: z-coordinate in mm<br>I: Intensity<br>C: Confidence of each point<br>NN: No NAN (Not a number) values<br>The output ASCII format of the point cloud can be configured by combining the letters to arbitary combinations like XYZICNN or XYZ. | | |
| **Get Commands** | GetPointCloudFormat | | |

### 7.24.2 Saving Point Cloud

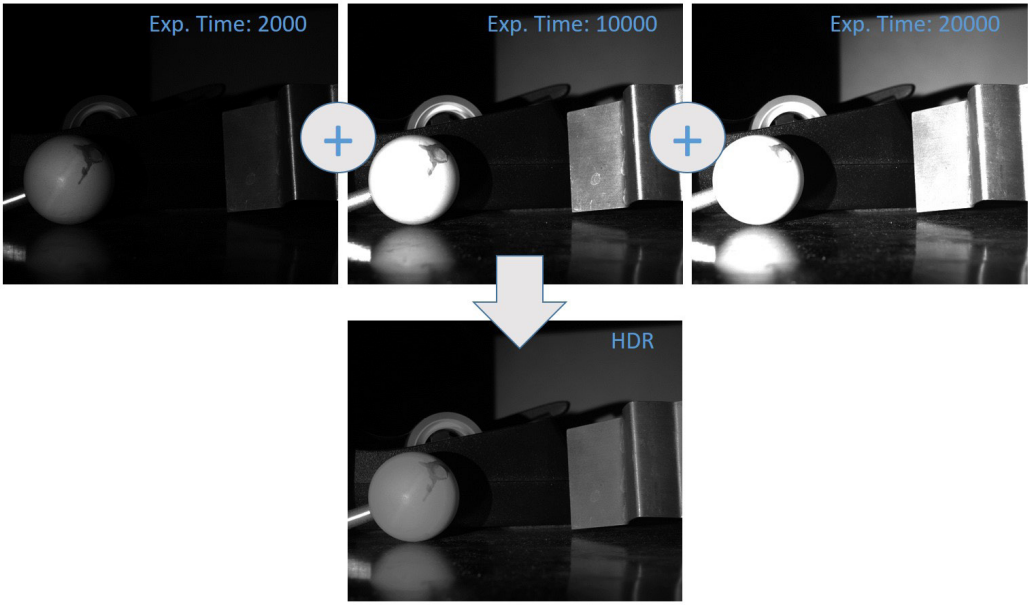
| | |
|---|---|
| **Command** | SavePointCloud=filename |
| **Parameter** | Filename |
| **Description** | Filename of the pointcloud. |
| **Get Commands** | --- |

## 7.25 HDR Point Cloud

### 7.25.1 Enable HDR Mode

| | | | |
|---|---|---|---|
| **Command** | SetMultiExposureEnabled=x | | |
| **Parameter** | x: 0 or 1 | **Default:** | 0 |
| **Description** | 0: HDR disabled<br>1 HDR enabled<br>Enables/disables the HDR (High Dynamic Range) mode which generates a pointcloud based on merging multiple exposures into a high dynamic range image stack. HDR mode makes it possible to generate high quality pointclouds of objects yielding very high contrasts not capturable with a single exposure. | | |
| **Get Commands** | GetMultiExposureEnabledCurrent<br>GetMultiExposureEnabledDefault<br>GetMultiExposureEnabledMin<br>GetMultiExposureEnabledMax | | |

Example:




### 7.25.2 Number of Point Clouds used for HDR

| | | | |
|---|---|---|---|
| **Command** | SetMultiExposureNumber=x | | |
| **Parameter** | x: 1,2 ... 5 (only integer values) | **Default:** | 1 |
| **Description** | Number of used point clouds for HDR point cloud. | | |
| **Get Commands** | GetMultiExposureNumberCurrent<br>GetMultiExposureNumberDefault<br>GetMultiExposureNumberMin<br>GetMultiExposureNumberMax | | |

### 7.25.3 Setting Exposure Time

| | |
|---|---|
| **Command** | SetMultiExposureTime1=x |
| **Parameter** | To obtain the value range and the default use the corresponding Get command. Only integer values are allowed. |
| **Description** | Exposure time is set in $\mu$s for first point cloud in HDR mode. |
| **GetCommands** | GetMultiExposureTime1 |

| Command | SetMultiExposureTime2=x |
|---|---|
| Parameter | To obtain the value range and the default use the corresponding Get command. Only integer values are allowed. |
| Description | Exposure time is set in µs for second point cloud in HDR mode (only needed if number >1). |
| Get Commands | GetMultiExposureTime2 |

| Command | SetMultiExposureTime3=x |
|---|---|
| Parameter | To obtain the value range and the default use the corresponding Get command. Only integer values are allowed. |
| Description | Exposure time is set in µs for third point cloud in HDR mode (only needed if number >2). |
| Get Commands | GetMultiExposureTime3 |

| Command | SetMultiExposureTime4=x |
|---|---|
| Parameter | To obtain the value range and the default use the corresponding Get command. Only integer values are allowed. |
| Description | Exposure time is set in µs for fourth point cloud in HDR mode (only needed if number >3). |
| Get Commands | GetMultiExposureTime4 |

| Command | SetMultiExposureTime5=x |
|---|---|
| Parameter | To obtain the value range and the default use the corresponding Get command. Only integer values are allowed. |
| Description | Exposure time is set in µs for fifth point cloud in HDR mode (only needed if number >4). |
| Get Commands | GetMultiExposureTime5 |

### 7.25.4 Setting Exposure Gain

| Command | SetMultiExposureGain1=x |
|---|---|
| Parameter | To obtain the value range and the default use the corresponding Get command. Only integer values are allowed. |
| Description | Gain is set in dB for first point cloud in HDR mode. |
| Get Commands | GetMultiExposureGain1 |

| Command | SetMultiExposureGain2=x |
|---|---|
| Parameter | To obtain the value range and the default use the corresponding Get command. Only integer values are allowed. |
| Description | Gain is set in dB for second point cloud in HDR mode (only needed if number >1). |
| Get Commands | GetMultiExposureGain2 |

| Command | SetMultiExposureGain3=x |
|---|---|
| Parameter | To obtain the value range and the default use the corresponding Get command. Only integer values are allowed. |
| Description | Gain is set in dB for third point cloud in HDR mode (only needed if number >2). |
| Get Commands | GetMultiExposureGain3 |

| Command | SetMultiExposureGain4=x |
|---|---|
| Parameter | To obtain the value range and the default use the corresponding Get command. Only integer values are allowed. |
| Description | Gain is set in dB for fourth point cloud in HDR mode (only needed if number >3). |
| Get Commands | GetMultiExposureGain4 |

| Command | SetMultiExposureGain5=x |
|---|---|
| Parameter | To obtain the value range and the default use the corresponding Get command. Only integer values are allowed. |
| Description | Gain is set in dB for fifth point cloud in HDR mode (only needed if number >4). |
| Get Commands | GetMultiExposureGain5 |

## 7.26 Save Point Cloud in TIFF Format

| Command | SaveTIFF=filename.tiff |
|---|---|
| Parameter | Filename |
| Description | Saving point cloud in TIFF format, 3-layer tiff (x, y, z). |
| Get Commands | --- |

## 7.27 Save Image Stack to File

| Command | SetSaveImageStack=filename |
|---|---|
| Parameter | Filename |
| Description | Saving image stack to file with format .bin for debugging (support). |
| Get Commands | --- |

## 7.28 Load Image Stack

| Command | SetLoadImageStack=filename |
|---|---|
| Parameter | Filename or filepath + name |
| Description | Loading image stack from binary file to SDK ImageStack. |
| Get Commands | --- |

## 7.29 Save Settings

| | |
|---|---|
| **Command** | SetSaveSettings=filename |
| **Parameter** | Filename or filepath + name |
| **Description** | Saving current user settings to a .set file. |
| **Get Commands** | --- |

## 7.30 Load Settings

| | |
|---|---|
| **Command** | SetLoadSettings=filename |
| **Parameter** | Filename or filepath + name |
| **Description** | Loading user settings from .set file. |
| **Get Commands** | --- |

## 7.31 Save Acquisition State

| | |
|---|---|
| **Command** | SetSaveAcquisitionState=filename |
| **Parameter** | Filename or filepath + name |
| **Description** | Saving current image stack and user settings to .sta file, e.g. to reconstruct sensor's point cloud generation with given stack and settings at a later time. |
| **Get Commands** | --- |

## 7.32 Load Acquisition State

| | |
|---|---|
| **Command** | SetLoadAcquisitionState=filename |
| **Parameter** | Filename or filepath + name |
| **Description** | Loading image stack and user settings from .sta file, eg. to reconstruct sensor's point cloud generation with given stack and settings. |
| **Get Commands** | --- |

## 7.33 Settings Default

| | |
|---|---|
| **Command** | SetSettingsDefault |
| **Description** | Resets all user settings to their default values. |
| **Get Commands** | --- |

## 7.34 Save HDR Image Stack

| | |
|---|---|
| **Command** | SetSaveHDRImageStack=filename |
| **Parameter** | Filename or filepath + name |
| **Description** | Saving current multi exposure image stacks (see section 7.25) to binary file. |
| **Get Commands** | --- |

## 7.35 Load HDR Image Stack

| | |
|---|---|
| **Command** | SetLoadHDRImageStack=filename |
| **Parameter** | Filename or filepath + name |
| **Description** | Loading multi exposure image stacks (see section 7.25) from binary file to SDK image stack. |
| **Get Commands** | --- |

## 7.36 Save Camera image to File

| | |
|---|---|
| **Command** | SaveCameraImage=filename |
| **Parameter** | Filename |
| **Description** | Saving camera image to file with format .bmp \| .png \| .jpg. The format is configured via used filename extension (e.g. test.bmp is format BMP and test.png is format PNG). |
| **Get Commands** | --- |

## 7.37 Mask Image

### 7.37.1 Enable Mask Image

| | |
|---|---|
| **Command** | SetMaskImageEnabled=x |
| **Parameter** | x: 0 or 1 |
| **Description** | Enables the use of a custom mask image. To set path refer to section 7.37 to define ROI. |
| **Get Commands** | GetMaskImageEnabledCurrent<br>GetMaskImageEnabledDefault<br>GetMaskImageEnabledMin<br>GetMaskImageEnabledMax |

### 7.37.2 Mask Image Path

| | |
|---|---|
| **Command** | SetMaskImagePath=filename |
| **Parameter** | Filename |
| **Description** | Filename of a black/white image in uncompressed bmp format. The resolution should be the same as the camera resolution. The image is used as mask to define the region of interest. White areas are evaluated and black not. Using this option decreases the computation time of the point cloud.<br>If a path is set the mask image is enabled automatically. To disable it use SetMaskImageEnabled=0. |
| **Get Commands** | GetMaskImagePath |

## 7.38 Logging

### 7.38.1 Enable Logging

| | | | |
|---|---|---|---|
| **Command** | SetFunctionLoggingEnabled=x | | |
| **Parameter** | x: 0 or 1 | **Default:** | 0 |
| **Description** | 0: Logging disabled<br>1: Logging enabled<br>Enables/disables logging function of SDK. | | |
| **Get Commands** | GetFunctionLoggingEnabled | | |

### 7.38.2 Logging Filename

| | |
|---|---|
| **Command** | SetFunctionLogFilename=filename |
| **Parameter** | Filename or path+filename |
| **Description** | Filename of logging file on disk. |
| **Get Commands** | GetFunctionLogFilename |

# 8. Read Data Commands

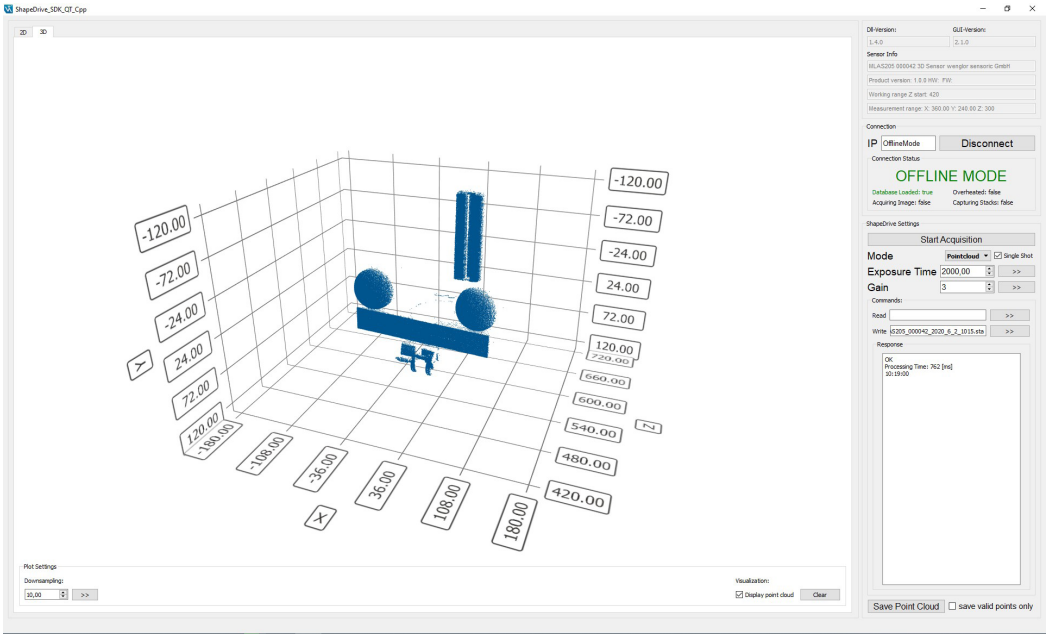
The commands are used together with the function EthernetScanner3D_ReadData.
The table below shows a list of read data commands which have no corresponding Set command.

| Command | Description | Remarks |
|---|---|---|
| GetXmlDescriptor | Gets user settings in XML format | |
| GetSdbLoaded | Returns if sensor database is loaded | |
| GetTemperature | Gets the temperature of the sensor in °C | |
| GetGetCommands | Returns list of available GetCommands | |
| GetSetCommands | Returns list of available SetCommands | |
| GetFirmwareVersion | Gets the firmware version | |
| GetHardwareVersion | Gets the hardware version | |
| GetMacAddress | Gets the Mac address of the sensor | |
| GetOrderNumber | Gets the order number of the sensor (e.g. MLAS204) | |
| GetSerialNumber | Gets the serial number of the sensor (e.g. 001012) | |
| GetProducer | Gets the producer of the sensor | |
| GetSdbDownloadProgress | Gets the progress of database download from the sensor | |
| GetCameraSerialNumber | Returns serial number of camera | |
| GetMeasuringRangeX | Returns the width of the measurement volume in X direction | |
| GetMeasuringRangeY | Returns the height of measurement volume in Y direction | |

| | | |
|---|---|---|
| GetMeasuringRangeZ | Returns the depth of measurement volume in Z direction | |
| GetWorkingRangeZ | Returns the distance in Z direction where the measurement volume starts | |
| GetRoi1Width | Gets the width of the region of interest | |
| GetRoi1Height | Gets the height of the region of interest | |
| ListDevices | Gets general information about the sensor, e.g.:<br><LIMA DIR="ReplyOk"<br>CMD="DEVICE_SCANWENGLOR"<br>ETHADDR="54:4a:05:0a:0d:0d"<br>IPADDR="192.168.100.1"<br>TCP_PORT="32001"<br>ARTICLENUMBER="MLAS101"<br>DEVICENAME="" /> | Can be used without being connected to sensor by passing nullptr instead of sensor handle |
| GetVersion | Gets the library version | |
| GetStacksPerSecond | Gets the frequency of image stacks acquisition. | |
| GetIntrinsicCamera Parameter | Gets the intrinsic camera parameters, e.g.:<br><FocalLength><br><first>3491.739746</first><br><second>3506.620361</second><br></FocalLength><br><PrincipalPoint><br><first>1263.248535</first><br><second>1045.453857</second><br></PrincipalPoint><br><Alpha>0</Alpha><br><Distortion><br><k1>-0.042916</k1><br><k2>0.317504</k2><br><k3>0.000000</k3><br><p1>0.001303</p1><br><p2>0.000327</p2><br></Distortion><br></Camera><br>See for parameter definitions:<br>https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html | |

# 9. Offline Mode

The offline mode enables the user to test the effects of different filter settings on point cloud generation given a prior saved image stack without having a sensor connected. It can also be used to reconstruct the internal state (settings and aquired image data) of a ShapeDrive sensor e.g. for support/analysis purposes.




To generate data files that can be used in offline mode an image stack needs to be acquired with a connected ShapeDrive sensor. It is possible to save either image stack SetSaveImageStack (see section 7.27) or image stack with user settings SetSaveAcquisitionState (see section 7.31).

In order to start the offline mode, the IP argument in the EthernetScanner3D_Connect(..) (see section 6.1) has to be set to "OfflineMode" or "0.0.0.0". In the SDK demo this state is displayed as "OFFLINE MODE" in the connection status display and the bit5 in sensor status (see function EthernetScanner3D_GetSensorStatus(..)) is set to 1.

In offline mode a specific sensor is emulated by loading its sensor database. This is done by using the command SetLoadSdb=<database name> (see section 7.2). In this case the name or path+name of the desired database has to be passed explicitly. In the SDK demo successful initialization is displayed by showing the emulated sensors info in the "Sensor Info" display and "Database Loaded: true" in the "Connection Status" display.

After that, image stacks or aquisition states of the emulated sensor can be loaded using the commands SetLoadImageStack (see section 7.28) or SetLoadAcquisitionState (see section 7.32).
After a stack is loaded the point cloud can be generated equally to the regular operation mode with the function EthernetScanner3D_GetPointCloud(..) (see section 6.6). Using the function EthernetScanner3D_GetCameraImage(..) (see section 6.7) the sequence of images in the loaded image stack is returned. Filter settings affecting the point cloud generation can be changed as required using their corresponding commands.

**Summary:**

Preparation:

1. Save ImageStack acquired with connected sensor by using commands SetSaveImageStack (see section 7.27) or SetSaveAcquisitionState (see section 7.31).

Use offline mode:

2. Connect passing "OfflineMode" or "0.0.0.0" as IP in EthernetScanner3D_Connect(..) (see section 6.1).
3. Load database of the sensor that should be emulated by passing the databases name or path+name with the command SetLoadSdb=<database name> (see section 7.2).
4. Load image stack or acquisition state acquired with emulated sensor with the commands SetLoadImageStack (see section 7.28) or SetLoadAcquisitionState (see section 7.32).
5. Compute and read the point cloud or stack images using the function EthernetScanner3D_GetPointCloud or EthernetScanner3D_GetCameraImage (see section 6.6 and 6.7).