EN

ShapeDrive G4 MLAS/MLBS

3D Sensors

Interface Description

Subject to change without notice
Available as PDF file only
Revision level: 18.12.2023
Version: 1.2.0
www.wenglor.com

Table of Contents

1. Change Index of Operating Instructions

Version	Date	Description/Change	Library Version
1.0.0	27.04.2023	Initial version of the interface description	1.0.0
1.0.1	03.05.2023	Improving the document version 1.0.0	1.0.0
1.0.2	14.09.2023	Small corrections	1.0.0
1.1.0	20.11.2023	Introduce new functions: • Register/unregister point cloud callback function • Device network search Introduce new commands: • Contrast comparison filter • I/O configuration • SDK queue size Bugfixes: • Fixed point cloud buffer size could grow unlimitedly • General improvements in performance and stability	1.1.0
1.2.0	18.12.2023	Introduced new functionalities: • Greyscale HDR Mode • Reading Intrinsic and Extrinsic Camera Parameters • General improvements in performance and stability	1.2.0

2. 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 see operating instructions at ShapeDrive 3D Sensors.

2.1 System Requirements

Applications development with the library requires a Microsoft operating system (WIN10, WIN11) or Linux (tested on Ubuntu 20.04).
The ShapeDrive product series requires Ethernet adapter with 1 Gbit up to 10 Gbit (last one is recommended).

Executing any 3D processing application normally requires a powerfull PC. The more resources the PC has, the better and faster processing 3D data is.

3. Application Example

The ShapeDrive SDK provides a simple C++ example as a demonstration on how to use the SDK functions. Below you see a pseudo code for demonstration:

```
// build a connection to the 3D sensor
// see function Sensor3D_Connect()

// read 3D sensor confiugrations to create the point cloud buffer
// see functions Sensor3D_ReadData(GetPixelXMax) and Sensor3D_ReadData(GetPixelYMax)

// (optional) select the sensor 3D mode (point cloud or 2D image)
// see function Sensor3D_WriteData(SetSensorMode)

// (optional) select the trigger source (internal, software, or hardware)
// see function Sensor3D_WriteData(SetTriggerSource)

// (optional) set up the acquisiton time
// see function Sensor3D_WriteData(SetAcquisitionTime)

//(optional) setup the sensor mode and the related LED pattern
//see function Sensor3D_WriteData(SetSensorMode) and Sensor3D_WriteData(SetLEDPattern)

// start the acquisition
// see function Sensor3D_WriteData(SetAcquisitionStart)

// read out point cloud or 2d images
// see fuctions GetPointCloud(...) or GetCameraImage(...)

// stop the acquisition
// see function Sensor3D_WriteData(SetAcquisitionStop)

// disconnect from the sensor 3D
// see function Sensor3D_disconnect(...)
```

4. SDK Description

All the SDK functions are based on C function standard calls (_stdcall) and are compatible with all compilers that support C programming language. In fact, since the functions are based on C standard call, they can be used in a wide range of programming languages and development environments (C++, C#, Visual Basic, Matlab, Delphi, Labview, Python..etc.).

NOTE!
All the SDK functions are thread safe.

4.1 Connect to ShapeDrive 3D Sensor

Command	void * Sensor3D_Connect(const char * ip, int timeout)
Parameter 1	ip: IP address of the sensor
Parameter 2	timeout: Time in msec during the function tries to connect to sensor.
Response	If successful, the function returns a pointer to the sensor, otherwise it returns a null pointer.
Description	Establish new connection to sensor.

4.2 Disconnect from ShapeDrive 3D Sensor

Command	int Sensor3D_Disconnect(void * sensor)
Parameter 1	sensor: pointer to created sensor to disconnect from
Response	If successful, the function returns SENSOR3D_OK (0), otherwise it returns error code (see section 4.9).
Description	Disconnecting from sensor.

4.3 Get Sensor Status

Command	int Sensor3D_GetSensorStatus(void* sensor, int *status)
Parameter 1	sensor: pointer to created sensor
Parameter 2	status: pointer to status, which will be filled with status information.
Response	If successful, the function returns SENSOR3D_OK (0), otherwise it returns error code (see section 4.9).
Description	Reading current status of sensor. In case of success *status bits can have following values: bit0: Sensor connected bit1: reserved bit2: Acquisition startet bit3: Sensor overheated bit4...bit31: reserved

4.4 Get SDK Version

Command	int Sensor3D_GetVersion(char *buffer, int buffer_size)
Parameter 1	buffer: output buffer, which will be filled with version information.
Parameter 2	buffer_size: size of buffer in bytes.
Response	If successful, the function returns SENSOR3D_OK (0), otherwise it returns error code (see section 4.9).
Description	Reading the current version of SDK.

4.5 Read Data

Command	int Sensor3D_ReadData(void* sensor, const char * feature_name, char * buffer, int buffer_size, int reserved)
Parameter 1	sensor: pointer to created sensor
Parameter 2	feature_name: required ASCII command (see section 6)
Parameter 3	buffer: output buffer, which will be filled with response data
Parameter 4	buffer_size: size of buffer in bytes
Parameter 5	reserved: not used, use zero
Response	If successful, the function returns SENSOR3D_OK (0), otherwise it returns error code (see section 4.9).
Description	Reading SDK or firmware properties by using predefined ASCII commands.

4.6 Write Data

Command	int Sensor3D_WriteData(void* sensor, const char * command)
Parameter 1	sensor: pointer to created sensor
Parameter 2	command: the ASCII command to be sent to the 3D sensor (see section 5)
Response	If successful, the function returns SENSOR3D_OK (0), otherwise it returns error code (see section 4.9).
Description	Writing SDK or firmware properties by using predefined ASCII commands.

NOTE!

The function Sensor3D_WriteData will return SENSOR3D_BUSY if the user sends a command during sensor acquisition.
The following commands are allowed to be sent to the sensor 3D during acquisition:

- SetGain
- SetExposureTime
- SetLEDPower
- SetTriggerSoftware
- SetAquisitionStop

4.7 Get Point Cloud

Command	int Sensor3D_GetPointCloud(void* sensor, void * buffer, int buffer_size, int * number_of_points, void* roi, int timeout)
Parameter 1	sensor: pointer to created sensor
Parameter 2	buffer: output buffer, which will be filled with point cloud data. The buffer should have a specific data format as illustrated below.
Parameter 3	buffer_size: size of buffer in bytes. The example below shows how the user can compute the size of the buffer.
Parameter 4	number_of_points: output argument used to return the number of copied points inside the buffer.
Parameter 5	roi: returns the current ROI settings of the sensor 3D. The ROI setting is defined using a structure as shown below.
Parameter 6	timeout: timeout for scanning one point cloud
Response	If successful, the function returns SENSOR3D_OK (0), otherwise it returns error code (see section 4.9).
Description	Reading point cloud. **NOTE!** The sensor should be in point cloud mode, see command SetSensorMode=4.

Data format of point cloud:

```
// Point structs
typedef struct {
        double x;
        double y;
        double z;
} POINT3D;

struct POINT_CLOUD {
        POINT3D* point;
        uint16_t* intensity;
        uint16_t* confidence;
        POINT_CLOUD() {
                point = nullptr;
                intensity = nullptr;
                confidence = nullptr;
                number_of_points = 0;
        }
};
```

- POINT3D* point : a pointer to a buffer in which the function copies the point cloud data. The user should manage creating and cleaning the buffer. If the pointer is nullptr, the function will skip copying the point cloud data into the buffer.
- unit16_t intensity : a pointer to a buffer in which the function copies the intensity data. The user should manage creating and cleaning the buffer. If the pointer is nullptr, the function will skip copying the intensity data into the buffer.
- uint16_t confidence : a pointer to a buffer in which the function copies the confidence data. The user should manage creating and cleaning the buffer. If the pointer is nullptr, the function will skip copying the confidence data into the buffer.

Example on how to create the point cloud buffer (in C++):

```
POINT_CLOUD buffer;
int nrPixels = camera_max_width * camera__max_height;
buffer.point = new POINT3D[nrPixels];
buffer.intensity = new unsigned short[nrPixels];
int buffer_size = (sizeof(POINT3D) + sizeof(unsigned short)) * nrPixels;
```

NOTE!
ShapeDrive FW 1.0.0 does not support confidence values.

```
struct ROI {
        uint32_t offset_x;
        uint32_t offset_y;
        uint32_t width;
        uint32_t height;
        }
```

4.8 Get Camera Image

Command	int Sensor3D_GetCameraImage(void* sensor, void * buffer, int buffer_size, void * roi, int timeout)
Parameter 1	sensor: pointer to created sensor
Parameter 2	buffer: output buffer, which will be filled with camera data.
Parameter 3	size: size of buffer in bytes
Parameter 4	roi: returns the current ROI settings of the sensor 3D.
Parameter 5	timeout: max. waiting time in msec until the camera data are read.
Response	If successful, the function returns SENSOR3D_OK (0), otherwise it returns error code (see section 4.9).
Description	Reading camera image as byte array out of sensor in 8 bit format. **NOTE!** The sensor should be in live image mode, see command SetSensorMode=5.

4.9 Register Point Cloud Callback

Command	int Sensor3D_RegisterPclCallback(void* sensor, void (*pcl_callback_func)(void* context),void* context)
Parameter 1	sensor: pointer to created sensor
Parameter 2	pcl_callback_func: function pointer to a function of type void with an argument of type void*
Parameter 3	context: context of the called function which will be passed as argument in pcl_callback_func. (e.g. 'this' pointer pointing to the context object of a member function)
Response	If successful, the function returns SENSOR3D_OK (0), otherwise it returns error code (see section 4.11).
Description	Registering a function which is automatically called when a point cloud is received.

NOTE!
The callback function is executed from the SDK's point clouds processing thread. Callback functions with long execution time might slow down point cloud acquisition process. Try to only read out point cloud in callback function and move further more time consuming processing to an extra thread.

4.10 Unregister Point Cloud Callback

Command	int Sensor3D_UnregisterPclCallback(void* sensor)
Parameter 1	sensor: pointer to created sensor
Description	Description: Unregistering the point cloud callback function. If successful, the function returns SENSOR3D_OK (0), otherwise it returns error code (see section 4.12)

4.11 List Devices

Command	int Sensor3D_ListDevices(char* buffer, int size)
Parameter 1	buffer: output buffer, which will be filled with list of ShapeDrive Sensors in Network
Parameter 2	size: size of buffer in bytes
Response	If successful, the function returns SENSOR3D_OK (0), otherwise it returns error code (see section 4.11).
Description	Listing info of all ShapeDrive devices in the network as JSON response. The function returns the error code SENSOR3D_SOCKETCOMMUNICATIONERROR (-106) if the port 33002 is used by other application on the host PC (e.g. Wenglor Discovery Tool).

Example Response for 2 Sensors:


```
[{
“deviceDetails”:{
        “articleNumber”:”MLAS112”,
        “authentication”:false,
        “capabilities”:26,
        “deviceStatus”:{“info”:[“Running”],
        “state”:”ACTIVE”,”value”:0},
        “deviceType”:”3D Sensor”,
        “deviceUrl”:”https://www.wenglor.com/product/MLAS112”,
        “firmwareVersion”:”1.2.0,
        “hardwareVersion”:”1.0.1”,
        “mac”:”54:4A:05:0A:3A:31”,
        “manufactureDate”:”29-11-2022 06:23:39”,
        “manufacturer”:”wenglor sensoric GmbH”,
        “network”:{
                “gateway”:”0.0.0.0”,
                “ip”:”192.168.100.1”,
                “mask”:”255.255.255.0”,”method”:”PERSISTENT_IP”,”nic_name”:””},
                “serialNumber”:”001004”,
                “userDefinedName”:”3D Sensor”
        }
}]
```


4.12 Error Codes

Error code definition	Value	Description
SENSOR3D_OK	0	No error.
SENSOR3D_INVALIDSENSORHANDLE	-1	Sensor handle doesn’t exists in the list of opened 3D sensors.
SENSOR3D_SENSORNOTCONNECTED	-2	Sensor is disconnected.
SENSOR3D_TIMEOUT	-3	Operation timeout.
SENSOR3D_CONFIGURATIONERROR	-4	Failed to read the configurations from the 3D sensor.
SENSOR3D_ARGUMENTNULLPOINTER	-101	Argument used in SDK function is null.
SENSOR3D_ARGUMENTOUTOFRANGE	-102	Argument used in SDK function is out of possible range. The range is described in XML settings which can be read out of SDK.
SENSOR3D_RETURNBUFFERTOOSMALL	-103	The size of the SDK result is bigger than the size of buffer used as argument in SDK function.
SENSOR3D_COMMANDNOTFOUND	-104	The ASCII command could not be found
SENSOR3D_SOCKETCOMMUNICA-TIONERROR	-106	Error in socket communication
SENSOR3D_BUSY	-201	The command cannot be processed. The device is in Acquisition state.
SENSOR3D_DATASTREAMERROR	-301	Failed to read data stream format from the sensor

5. Write Data Commands

The commands are used together with the function Sensor3D_WriteData (see section 4.6).

5.1 Acquisition Start

Command	SetAcquisitionStart
Parameter	No parameter
Description	Starts data acquisition of the 3D Sensor. Depending on the tirgger source (see SetTriggerSource) and the sensor enable (see SetSensorEnable), the 3D sensor will send the acquired data without stop until it receives the command SetAcquisitionStop.

5.2 Acquisition Stop

Command	SetAcquisitionStop
Parameter	No parameter
Description	Stops the data acquisition of the 3D sensor.

5.3 SDK Queue Size

Command	SetSDKQueueSize=x		
Parameter	x	Default:	5
Description	Values for x: Maximum number of point clouds buffered in SDK Sets the maximum size of the SDK's queue buffer for point clouds. (maximum number of point clouds stored).		
Get Commands	GetSDKQueueSize GetSDKQueueSizeDefault		

NOTE!
If more point clouds are received than read out via GetPointCloud these are stored in the SDK's internal queue. On GetPointCloud the oldest stored cloud is returned (FIFO principle) and removed from the buffer. If the buffer exceeds its size limit, the newest received cloud is written to the buffer and the oldest one stored is discarded.

NOTE!
On each AcquisitionStart the queue will be reset.

5.4 Trigger Source

Command	SetTriggerSource=x		
Parameter	x = 0, 1, 2, 3, 4, 5	Default:	0
Description	Sets the trigger source of the 3D sensor. 0: Internal trigger: The sensor triggers itself. The trigger rate is defined by the command SetAcquisitionTime 1: Software trigger: The 3D sensor is triggered when it receives the software command SetTriggerSoftware. 2 ... 5: Hardware trigger (2=I/O1 ... 5=I/O4): The 3D sensor is triggered when it receives a signal on the selected digital I/O.		
Get Commands	GetTriggerSource GetTriggerSourceMin GetTriggerSourceMax GetTriggerSourceDefault		

Command	SetTriggerSoftware
Description	It triggers the 3D sensor to do an acqusition. The trigger source of the 3D sensor should be in software trigger (see SetTriggerSource).

5.5 Enable Sensor

Command	SetSensorEnable=x		
Parameter	x = 0, 1, 2, 3, 4	Default:	0
Description	Activates the I/O data acquisition of the 3D sensor. 0: Off (disables the global data acquisition, data acquisition enabled) 1...4: Referes to line1..line4. If x = 1 then the 3D Sensor will acquire data only if the I/O line1 is active.		
Get Commands	GetSensorEnable GetSensorEnableMin GetSensorEnableMax GetSensorEnableDefault		

NOTE!
For details about the relationship between I/O data acquisition, the data acquisition commands (SetAcquistionStart and SetAcquisitionStop) and the trigger source of the 3D Sensor check operating instructions at ShapeDrive 3D Sensors.

5.6 Set Sensor Mode

Command	SetSensorMode=x		
Parameter	x = 4 or 5	Default:	4
Description	Configure the 3D sensor to generate point cloud or 2D images. 4: 3D point cloud 5: 2D camera images The output rate is defined by the command SetAcquisitionTime.		
Get Commands	GetSensorMode GetSensorModeDefault		

5.7 Set HDR Mode

Command	SetHDRMode=x		
Parameter	x = 0 or 1	Default:	0
Description	Sets the sensor's HDR mode. 0: disabled (single exposure) 1: Greyscale HDR Acquires an exposure bracket around the currently set exposure time and calculates a HDR image which is output as the pointcloud's intensity image (only has effect on the pointcloud's texture, not on the pointcloud itself). **NOTE!** Greyscale HDR is availabel in ShapeDrive FW version 1.2.0 or higher.		
Get Commands	GetHDRMode		

5.8 Set Acquisition Time

Command	SetAcquisitionTime=x
Parameter	To obtain the value range and the default see the corresponding Get commands. Only integer values are allowed.
Description	x is set in μs Determines the interval of acquiring point cloud or 2D camera images, see the command SetSensorMode.
Get Commands	GetAcquisitionTime GetAcquisitionTimeDefault GetAcquisitionTimeMin GetAcquisitionTimeMax

5.9 Set Exposure Time Limit

Command	SetExposureTimeLimit=x
Parameter	To obtain the value range and the default see the corresponding Get commands. Only integer values are allowed.
Description	x is set in μs Sets the exposure time limit. The LED projector frequency will be adjusted according to the new limit. Processing the command might take up to 5 seconds.
Get Commands	GetExposureTimeLimit GetExposureTimeLimitDefault GetExposureTimeLimitMin GetExposureTimeLimitMax

5.10 Set Exposure Time

Command	SetExposureTime=x
Parameter	To obtain the value range and the default see the corresponding Get commands. Only integer values are allowed.
Description	x is set in μs Sets the exposure time of the built-in camera chip of the 3D Sensor. The exposure time should be equal or less than the Exposure Time Limit. The exposure time is set larger than the limit, then it will be adjusted internally to the limit.
Get Commands	GetExposureTime GetExposureTimeDefault GetExposureTimeMin GetExposureTimeMax

5.11 Set LED Pattern

Command	SetLEDPattern=x		
Parameter	x = 16, 28 or solid	Default:	28
Description	Defines the LED projection pattern. x = 16 or 28: Used for 3D point cloud (SetSensorMode=4). It defines the number of images used for every 3D capture to generate the point cloud. A higher value means more accuracy in the point cloud, however the 3D capture time will be slower. x=solid: Used for 2D camera images (SetSensorMode=5). The user should switch the LED pattern to solid in 2D camera image.		

NOTE!
The built-in camera chip of the 3D sensor is controlled internally by the LED projector. The frequency of the LED projector depends on the value of the ExposureTimeLimit which in turns influence the trigger rate of the built-in camera.
The relationship between ExposureTimeLimit, AcquisitionTime and the number of images per point cloud is as follows:

AcquisitionTime > ExposureTimeLimit × number of acquired images per point cloud.

5.12 Set Gain

Command	SetGain=x		
Parameter	To obtain the value range and the default see the corresponding Get commands. Only integer values are allowed.		
Description	Sets the gain of the build-in camera chip of the 3D sensor.		
Get Commands	GetGain GetGainDefault GetGainMin GetGainMax		

5.13 Set Subsampling

Command	SetSubSampling=x		
Parameter	x = 0 or 1	Default:	0
Description	Activates the subsampling in the built-in camera chip of the 3D sensor.		
Get Commands	GetSubSampling GetSubSamplingMin GetSubSamplingMax GetSubSamplingDefault		

5.14 Set LED Power

Command	SetLEDPower=x
Parameter	To obtain the value range and the default see the corresponding Get commands. Only integer values are allowed.
Description	x is set in % Sets the brightness of the sensor's LED projector. **NOTE!** Raising LED power can be used to compensate lower exposure times and so decrease the overall acquisition time.
Get Commands	GetLEDPower GetLEDPowerDefault GetLEDPowerMin GetLEDPowerMax

5.15 Set User LED

Command	SetUserLED=x		
Parameter	x = 0, 1, 2, 3	Default:	0
Description	Sets User LED colour. 0: Off 1: Green 2: Red 3: Orange		
Get Commands	GetUserLED GetUserLEDMin GetUserLEDMax GetUserLEDDefault		

5.16 Reboot Sensor

Command	SetReboot
Parameter	No parameter
Description	Reboots the 3D sensor per software command

5.17 Activate/Deactivate LED Projector

Command	SetLEDActivate=x		
Parameter	x = 0 or 1	Default:	1
Description	This command activates/deactivates built-in LED projector of the 3D Sensor. 0: deactivate 1: activate		
Get Commands	GetLEDActivate GetLEDActivateMin GetLEDActivateMax GetLEDActivateDefault		

5.18 Extended Measuring Range

Command	SetEnableExtendedMeasuringRange=x		
Parameter	x = 0 or 1	Default:	0
Description	Activates/deactivates the extended working area. The extended working area is different between different devices of the same 3D sensor type. 0: deactivate 1: activate		
Get Commands	GetEnableExtendedMeasuringRange		

NOTE!

There is no guarantee that the extended working range of one device will cover the same area of another device belonging to the same device type.
Accuracy and geometric correctness cannot be ensured outside the calibrated measuring range.

5.19 Min Measuring Range Z

Command	SetZmin=z		
Parameter	z = Start of working range Z...end of working range Z	Default:	Start of working range Z
Description	Reduces the working range of the 3D sensor in Z. The value will be truncated if it exceeds the limits.		
Get Command	GetZmin		

5.20 Max Measuring Range Z

Command	SetZmax=z		
Parameter	z = Start of working range Z...end of working range Z	Default:	End of working range Z
Description	Reduces the working range of the 3D Sensor in Z. The value will be truncated if it exceeds the limits.		
Get Command	GetZmax		

NOTE!

The value for Z min should be less than Z max.

5.21 Contrast Comparison Filter

Command	SetContrastComparisonFilterMinPhase=x
Parameter	To obtain the value range and the default see the corresponding Get commands. Only integer values are allowed.
Description	Sets the filter threshold of the contrast comparison filter where 1 is the highest possible contrast and 0 is no contrast at all. Points with a phase contrast below the threshold are filtered out.
Get Commands	GetContrastComparisonFilterMinPhase GetContrastComparisonFilterMinPhaseMin GetContrastComparisonFilterMinPhaseMax GetContrastComparisonFilterMinPhaseDefault

5.22 Digital I/O Line Mode

NOTE!

Digital IO Line Mode commads are available in ShapeDrive FW version 1.1.0 or higher.

Command	SetEA1LineMode=x		
Parameter	x = 0 or 1	Default:	0
Description	Sets the sensors digital I/O as input or output 0: Input 1: Output		
Get Commands	GetEA1LineMode GetEA1LineModeMin GetEA1LineModeMax GetEA1LineModeDefault		

Command	SetEA2LineMode=x		
Parameter	x = 0 or 1	Default:	0
Description	Sets the sensors digital I/O as input or output 0: Input 1: Output		
Get Commands	GetEA2LineMode GetEA2LineModeMin GetEA2LineModeMax GetEA2LineModeDefault		

Command	SetEA3LineMode=x		
Parameter	x = 0 or 1	Default:	1
Description	Sets the sensors digital I/O as input or output 0: Input 1: Output		
Get Commands	GetEA3LineMode GetEA3LineModeMin GetEA3LineModeMax GetEA3LineModeDefault		

Command	SetEA4LineMode=x		
Parameter	x = 0 or 1	Default:	0
Description	Sets the sensors digital I/O as input or output 0: Input 1: Output		
Get Commands	GetEA4LineMode GetEA4LineModeMin GetEA4LineModeMax GetEA4LineModeDefault		

5.23 I/O Line Source

NOTE!

I/O Line Source commads are available in ShapeDrive FW version 1.1.0 or higher.

Command	SetEA1LineSource=x
Parameter	x = 0, 1 or 2
Description	Sets the line source for the Sensors I/O 0: FrameActive 1: SensorBusy 2: Timer0
Get Commands	GetEA1LineSource GetEA1LineSourceMin GetEA1LineSourceMax GetEA1LineSourceDefault

Command	SetEA2LineSource=x
Parameter	x = 0, 1 or 2
Description	Sets the line source for the Sensors I/O 0: FrameActive 1: SensorBusy 2: Timer0
Get Commands	GetEA2LineSource GetEA2LineSourceMin GetEA2LineSourceMax GetEA2LineSourceDefault

Command	SetEA3LineSource=x
Parameter	x = 0, 1 or 2
Description	Sets the line source for the Sensors I/O 0: FrameActive 1: SensorBusy 2: Timer0
Get Commands	GetEA3LineSource GetEA3LineSourceMin GetEA3LineSourceMax GetEA3LineSourceDefault

Command	SetEA4LineSource=x
Parameter	x = 0, 1 or 2
Description	Sets the line source for the Sensors I/O 0: FrameActive 1: SensorBusy 2: Timer0
Get Commands	GetEA4LineSource GetEA4LineSourceMin GetEA4LineSourceMax GetEA4LineSourceDefault

5.24 Timer0 Configuration

NOTE!

Timer0Configuration commads are available in ShapeDrive FW version 1.1.0 or higher.

5.24.1 Timer0 Source

Command	SetTimer0Source=x
Parameter	x = 0 or 1
Description	Description: Sets the trigger source of Timer0 0: FrameStart 1: FrameEnd
Get Commands	GetTimer0Source GetTimer0SourceMin GetTimer0SourceMax GetTimer0SourceDefault

5.24.2 Timer0 Duration

Command	SetTimer0Duration=x
Parameter	To obtain the value range and the default see the corresponding Get commands. Only integer values are allowed.
Description	Sets the Timer0 line's pulse duration in usec.
Get Commands	GetTimer0Duration GetTimer0DurationMin GetTimer0DurationMax GetTimer0DurationDefault

5.24.3 Timer0 Delay

Command	SetTimer0Delay=x
Parameter	To obtain the value range and the default see the corresponding Get commands. Only integer values are allowed.
Description	Sets the Timer0 line's delay in usec.
Get Commands	GetTimer0Delay GetTimer0DelayMin GetTimer0DelayMax GetTimer0DelayDefault

NOTE!

For details about the sensor's trigger behaviour and configuration check the operating instructions at ShapeDrive 3D Sensors.

6. Read Data Commands

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

Command	Description
GetROI1WidthX	Reading amount of pixels of camera in X direction
GetROI1HeightY	Reading amount of pixels of camera in Y direction
GetMeasuringRangeX	Reading the length of the sensors calibrated measuring range in X direction [mm]
GetMeasuringRangeY	Reading the length of the sensors calibrated measuring range in Y direction [mm]
GetMeasuringRangeZ	Reading the length of the sensors calibrated measuring range in Z direction [mm] (Zstart-Zend)
GetZstart	Reading the start of the sensors calibrated measuring range in Z direction [mm]
GetZend	Reading the end of the sensors calibrated measuring range in Z direction [mm]
GetPixelXMax	Reading camera width in pixel
GetPixelYMax	Reading camera height in pixel
GetOrderNumber	Reading sensor's order number
GetSerialNumber	Reading sensor's serial number
GetDescription	Reading sensor's description
GetProducer	Reading sensor's producer info
GetProductVersion	Reading sensor's product version
GetHardwareVersion	Reading sensor's hw version
GetFirmwareVersion	Reading sensor's fw version
GetSDKQueueLoad	Reading current SDK FIFO loading factor in percent 0-100
GetSDKQueueSize	Reading current SDK FIFO size
GetXmlDescriptor	xml from device
GetOperatingTemperature	Reading temperature of internal body of the 3D sensor
GetIntrinsicCameraParameters	Reading the sensor's intrinsic camera parameters (available in FW 1.2.0 or higher)
GetExtrinsicCameraParameters	Reading the sensor's extrinsic camera parameters (available in FW 1.2.0 or higher)

Example for Camera parameters format:

Intrinsic:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<intrinsic_camera_parameter>
   <FocalLength>
            <first>4618.431998128899977</first>    <second>4619.793930338903920</second>
   </FocalLength>
   <PrincipalPoint>
      <first>1199.812538251492242</first>
      <second>1023.536277863465330</second>
   </PrincipalPoint>
   <Alpha>0.000000000000000</Alpha>
   <Distortion>
      <k1>-0.190502758875769</k1>
      <k2>0.274045242136254</k2>
      <k3>0.000000000000000</k3>
      <p1>-0.000169400134632</p1>
      <p2>-0.000070347409649</p2>
   </Distortion>
</intrinsic_camera_parameter>
```

Extrinsic:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<extrinsic_camera_parameter>
   <T00>0.000000000</T00>
   <T10>0.000000000</T10>
   <T20>0.000000000</T20>
   <T30>0.000000000</T30>
   <T01>0.000000000</T01>
   <T11>0.000000000</T11>
   <T21>0.000000000</T21>
   <T31>0.000000000</T31>
   <T02>0.000000000</T02>
   <T12>0.000000000</T12>
   <T22>0.000000000</T22>
   <T32>0.000000000</T32>
   <T03>0.000000000</T03>
   <T13>0.000000000</T13>
   <T23>0.000000000</T23>
   <T33>0.000000000</T33>
</extrinsic_camera_parameter>
```

For a detailed description of parameters and underlying models see:
https://docs.opencv.org/4.8.0/d9/d0c/group__calib3d.html