Table of Contents

1 Sensor Connection

Protocol version 5xx

1.1 TCP/IP Settings

The PC connects to the sensor with a TCP/IP connection.

The sensor acts as server.

IP: 192.168.0.65 (standard IP, if no camera #IP file)

Port: 1096 (Master) and 1097 (Slave) – Default Values

1.2 TCP/IP Functions

```
// high level functions
I32  TCP_NewConnection (U32 Sock, U32 *Alias);
I32 TCP_ReadCommand (LsParameter *LsPar, U32 *Alias, U8 *pCommand, I32 MinCommand, I32 MaxCommand);
I32 TCP_SendCommand (LsParameter *LsPar, U32 Alias, U8 *pCommand, I32 maxlen, enum SendMode Mode);

// low level functions
I32  TCP_Open            (I32 Port, U32 *NewSock);
I32  TCP_Accept          (U32 Sock, U32 *FromSock);
I32  TCP_Send            (U32 Sock, char *SendData, I32 SendBytes);
I32  TCP_Read            (U32 Sock, char *ReceiveData, I32 ReceiveBytes);
I32  TCP_Close           (U32 Sock);
```

2 Communication between Host and 3D camera sensor

Directly after establishing the TCP/IP connection the following communication is possible. We recommend to use the new STRING command for sensor settings.

1. Commands from Host (Client) to Sensor (Server)
    - STRING Commands
    - SET Commands
    - GET Commands

2. Responses from Sensor (Server) to Host (Client)
    - Command Response after STRING Command
    - Command Response after SET Command
    - Command Response after GET Command
    - Result Response during RUN mode

2.1 Global Header

For an easy communication way, there is a fixed **GLOBAL HEADER** of 32 Bytes for every Client or Server communication. The size and structure is always the same.

Size	Type	Description of Global Header	Value
4	U32	Synchronisation Bytes	0x10101010, 0x00001111, 0x01010101, 0x1111000 or 0xAA0F55F0
4	U32	Checksum (Global Header)	[0 .. 0x11111111]
4	U32	Checksum (Data)	[0 .. 0x11111111]
4	U32	Counter (0 = Reset Counter)	[0 .. 2^32-1]
4	I32	Data waiting time in ms (0 = Infinite)	[0 .. 10000]
4	I32	Reserved	
4	I32	Error	[-2^32 .. 2^32-1]
4	U32	Data Length N in Bytes (Without Global Header)	[0 .. 2^32-1]
		---------- END OF GLOBAL HEADER ----------	
N	U8	Data Bytes	[0 .. 2^8-1]

The Global Header is the leading part of every command. The following byte stream with the size of Data Length includes the main information.

The synchronisation Bytes are a fixed value for an easy 'start of header' recognition.

The Global Header Checksum starts from Counter until the end of DataLength. The Data checksum revers only to the Data bytes. Checksum is an easy XOR combination of all bytes.

Counter is a global value, which allows a simply check for missing commands. Every part (Client or Server) increases their individual counter. The opposite part can verify this value and recognize any lack. A counter value of 0 resets the counter of the opposite part. It is recommended to reset the value after each connection. After reaching the max. value, the counter restarts at 0.

The data waiting time works as a time out and can limit the waiting time for receiving the data bytes.

Error can be used for fast error messages like TimeOut, Checksum Error etc.

Data Length defines the following data size in bytes.

2.2 Host (Client) Data Bytes

The “Command Type” defines the action mode:

0: STRING (ASCII) Command (allows one or more commands at once)
1: GET (BINARY) command (ask for parameter value and range)
2: SET (BINARY) command (set parameter value)

The “Parameter ID” is the value corresponding to the parameter list (see separate chapter). For every “Parameter ID” you will find the possible “Command Types” in the table.

It is recommended to use the new STRING commands as it is more flexible and allows SENSOR and PRODUCT settings.

2.2.1 Host “STRING” - Command

Size	Type	Description of Meta Header	Value
4	U32	Command Type	[0]
4	U32	Parameter ID	[0 .. 2^32-1]
4	U32	Host Command ID	[0 .. 2^32-1]
4	U32	STRING Length N in Bytes (N=0, if no STRING Bytes)	[0 .. 2^32-1]
N	U8	STRING Bytes	[0 .. 2^8-1]

For valid “STRING” commands, please refer to the separate chapter. It is possible, to send one, more or all “STRING” parameters in one command. Every parameter setting must be terminated by Line Feed (0x0A) or NULL (0x00).

2.2.2 Host “GET” - Command

Size	Type	Description of Meta Header	Value
4	U32	Command Type	[1]
4	U32	Parameter ID	[0 .. 2^32-1]
4	U32	Host Command ID	[0 .. 2^32-1]

2.2.3 Host “SET” - Command

Size	Type	Description of Meta Header	Value
4	U32	Command Type	[2]
4	U32	Parameter ID	[0 .. 2^32-1]
4	U32	Host Command ID	[0 .. 2^32-1]
4	I32	Parameter Value	[-2^32 .. 2^32-1]

2.3 Sensor (Server) Data Bytes

The “Command Type” defines the action mode:

0: STRING (ASCII) command response
1: GET (BINARY) command response
2: SET (BINARY) command response

The “Parameter ID” is the value corresponding to the parameter list (see separate chapter). For every “Parameter ID” you will find the possible “Command Types” in the table.

The “Error” value provides the result of the parameter handling (see separate chapter).

2.3.1 Sensor “STRING” - Response

Size	Type	Description of Meta Header	Value
4	U32	Command Type	[0]
4	U32	Parameter ID	[0 .. 2^32-1]
4	U32	Host Command ID Reply	[0 .. 2^32-1]
4	I32	Error	[-2^32 .. 2^32-1]
4	U32	STRING Length N in Bytes (N=0, if no STRING Bytes)	[0 .. 2^32-1]
N	U8	STRING Bytes	[0 .. 2^8-1]

2.3.2 Sensor “GET” - Response

Size	Type	Description of Meta Header	Value
4	U32	Command Type	[1]
4	U32	Parameter ID	[0 .. 2^32-1]
4	U32	Host Command ID Reply	[0 .. 2^32-1]
4	I32	Error	[-2^32 .. 2^32-1]
4	I32	Parameter Value	[-2^32 .. 2^32-1]
4	I32	Parameter Min Value	[-2^32 .. 2^32-1]
4	I32	Parameter Max Value	[-2^32 .. 2^32-1]

2.3.3 Sensor “SET” - Response

Size	Type	Description of Meta Header	Value
4	U32	Command Type	[2]
4	U32	Parameter ID	[0 .. 2^32-1]
4	U32	Host Command ID Reply	[0 .. 2^32-1]
4	I32	Error	[-2^32 .. 2^32-1]
4	I32	Parameter Value	[-2^32 .. 2^32-1]
4	I32	Parameter Min Value	[-2^32 .. 2^32-1]
4	I32	Parameter Max Value	[-2^32 .. 2^32-1]

2.4 Sensor (Server) Result Response Data Bytes

2.4.1 Server “DATA” - Result Response during RUN mode

Size	Type	Description of Meta Header	Value
4	U32	Command Type	[100]
4	I32	Data Mode	[1 .. 9]
4	U32	Sensor Result Counter	[0 .. 2^32-1]
8	U64	Image Time Stamp after Camera Boot [ms]	[0 .. 2^64-1]
N	U8	Data	[-2^8 .. 2^8-1]

The “Data” structure and size depends on the selected “Data Mode” (see separate chapter).

3 Parameter ID

Nearby all commands are usable as Read-Commands and Set-Commands. Some are only Read- or Set-Commands. Please refer to the list below. You will find Valid “command types” for every parameter in the table “CT” (0=STRING, 1=GET and 2=SET).

Parameter ID	ID	CT	Effect
RESET	0	0/2	Stop image acquisition, reset host and camera counter and set NBR_LINES to 0
LaserMode	1	1/2	1: set laser on 0: set laser off
ROI_X	2	1/2	Set the ROI x starting point at X
ROI_Y	3	1/2	Set the ROI y starting point at X
ROI_DX	4	1/2	Set the horizontal ROI width received at X
ROI_DY	5	1/2	Set the vertical ROI width received at X
GainVal	6	1/2	Set the gain at X
ShutterTime	7	1/2	Set the shutter at X
TriggerMode	8	1/2	0: set the line trigger off 1 (TRIG_RAIS): set the line trigger on raising value 2 (TRIG_FAL): set the line trigger on falling value -1: Auto trigger mode (camera trigger, see CMD_AUTO_TRIG_PERIODE) -2: Auto trigger mode (like -1, starting at a raising edge) -3: Auto trigger mode (like -1, starting at a falling edge)
DataMode	9	1/2	-1: sensor activ, no resutl transfer via TCP/IP 0: MODE_ STANDBY, no sensor activity 1: MODE_ TAKE_PIC 2: MODE_2D _POINTLIST 3: MODE_2D _POINTLIST_AND_LASER_IMG 4: MODE_2D _POINTLIST_MM (order x0 z0 / x1 z1 / x2 z2 / ...) 5: MODE_2D _POINTLIST_MM (order x0 x1 x2 ... / z0 z1 z2 ...) 6: MODE_BINARY_PIC 7: Not used any more 8: MODE_PRODUCT_RESULTS if sensor works in product mode 9: combination of Mode 4 and 8 (sends first Mode 4 than Mode 8) Look at the description below for more info about its meta Header
NbrLines	10	1/2	Set the number of scans that will be taken If X=-1 there will be endless capturing until CMD_RESET has been sent If X=0 there will be no scans and X=-2 will take the value from the other port
Methode	11	1/2	Set the line extraction method: 0 center detection at pixel accuracy 1 BaryCenter detection 2 BaryCenter detection with speckle filter (recommended)
Algorithm	12	1/2	Set the Algorithm parameter for PointList (DataMode 2, 3 and 7) 0 Fixed point algorithm (multiplied by 16 for sub pixel accuracy) 1 Floating point algorithm
RlcThresh	13	1/2	Set threshold for laser line detection
MinWidth	14	1/2	Set min. Laser Line Width
MaxWidth	15	1/2	Set max. Laser Line Width
ExposureMode	16	1/2	Set exposure mode: 0: FIX_EXPOSURE (use CMD_SHUTTER) 1: AUTO_EXPOSURE (overwrites CMD_SHUTTER [MINSH..MAXSH]) 2: DOUBLE_EXPOSURE (use CMD_DOUBLE_SH1 and 2)
AutoShutterVal	17	1/2	AUTO_EXPOSURE mode: Select Shutter time for an average laser brightness of laser value
AutoShutterMin	18	1/2	AUTO_EXPOSURE mode:

			Min. allowed Shutter time [µs]
AutoShutterMax	19	1/2	AUTO_EXPOSURE mode: Max. allowed Shutter time [µs]
AutoShutterFilter		0	Auto shutter mode to calculate laser brightness based on: 0: all valid laser points 1: 50% of median valid laser points 2: 50% of brightest valid laser points 3: 50% of darkest valid laser points 4: 50% of centre valid laser points 5: 50% of left valid laser points 6: 50% of right valid laser points
DoubleShutter1	20	1/2	DOUBLE_EXPOSURE mode: Shutter time 1 [µs] – main image shutter
DoubleShutter2	21	1/2	DOUBLE_EXPOSURE mode: Shutter time 2 [µs] – second image, if no laser detection in main image
ReflexionFilter	22	1/2	Laser Line Detection Filter Mode 0: No Filter >0 : skip n reflections from top (if more lines are in the image) <0 : skip n reflections from bottom (if more lines are in the image)
ROI_X_MM	23	1/2	Read the ROI x starting point [mm]
ROI_Y_MM	24	1/2	Read the ROI y starting point at y
ROI_DX_MM	25	1/2	Set the horizontal ROI width received at X
ROI_DY_MM	26	1/2	Set the vertical ROI width received at X
LIB_VERSION	27	1	Get 3D Scanner Library Version
PLC_IN	28	1	Get all PLC inputs (Bit0 … Bit3)
PLC_OUT	29	2	Set all PLC outputs (Bit0 … Bit3)
EthernetPackNr	30	1/2	Add n scan lines together in one Ethernet package. 0: write as much scan lines into camera memory and send the results at the end or if memory is full (about 11000 scans). This method is the fastest scan mode, as full processor power is available for scanning. But it will take more time at the end to transfer all data at once. Don't use this mode, if you scan permanently. 1: send scan line results directly after every scan (recommended) n: put n scan lines together in one Ethernet package. Recommended is max. of n=4 scan lines! Please have a look at CMD_SEND_NOWAIT !
AutoTriggerFPS	31	1/2	Camera auto trigger mode in 1/10 frames per second 500 means 50 fps. See also CMD_AUTO_TRIG_ERR
AUTO_TRIG_ERR	32	1/2	nr scans which are not in time during Autotrigger (allowed jitter < 50µs) >0 means, camera is not fast enough for selected scan speed, n lines are out of tolerance. Customer have to reset the value.
SAVE_SENSOR_FILE	33	0/1	Saves latest sensor settings in camera flash. Camera will start with this parameters
EthernetSendNoWait	34	1/2	0: send will wait until all data are transferred If host is not ready, it will lead to a delay of the next scan line 1: send as much data as possible (recommended) If host is not ready, it will temporally fill camera memory. If memory is full, it stops scanning, until all data are transferred
SubSample	35	1/2	0: full resolution for laser line detection 1: combine two lines for laser detection (fast laser line detection) 2: combine four lines for laser detection (fast laser line detection)
LOAD_PRODUCT_FILE	38	0/1	reads the product parameters from the camera file "fd:/ProdPar.001"
CameraSNR	39	1	Gets the serial number of the camera ID.
RECEIVE_PRODUCT_DATA	40	0	Set product parameter Host -> Cam from STRING.

SEND_PRODUCT_DATA	41	0	Send product parameter Cam -> Host as STRING.
LOAD_SENSOR_FILE	42	0/1	reads the product parameter from the camera file "fd:/VC3DPar.001"
RECEIVE_SENSOR_DATA	43	0	Set sensor parameter Host -> Cam from STRING.
SEND_SENSOR_DATA	44	0	Send sensor parameter Cam -> Host as STRING.
SAVE_PRODUCT_FILE	45	0/1	Saves latest product settings in camera flash. Camera will start with this parameters
SpeckleFilter	46	1/2	Low pass filter to improve laser speckles noise (3x3, 5x5, 7x7 or 9x9)
MedianFilter	47	1/2	Median filter to improve laser detection errors (3x1, 5x1, 7x1 or 9x1)

4 STRING parameters

All valid PARAMETER STRINGs and sensor depending parameter ranges will be transferred with the commands SEND_SENSOR_DATA or SEND_PRODUCT_DATA. For an easy telnet output, use the menu 'w'.

```
Sensor Parameter:
-----------------
LaserMode=0 Min=0 Max=1
ROI_X_MM=-32 Min=-67 Max=70
ROI_Z_MM=87 Min=88 Max=249
ROI_DX_MM=67 Min=1 Max=137
ROI_DZ_MM=10 Min=1 Max=161
GainVal=64 Min=0 Max=1023
ShutterTime=500 Min=5 Max=500000
TriggerMode=0 Min=-3 Max=2
DataMode=0 Min=-1 Max=9
NbrLines=0 Min=-2 Max=10000
Methode=2 Min=0 Max=2
Algorithm=0 Min=0 Max=1
RlcThresh=48 Min=1 Max=254
MinWidth=1 Min=1 Max=100
MaxWidth=30 Min=1 Max=100
ExposureMode=0 Min=0 Max=3
AutoShutterVal=128 Min=1 Max=254
AutoShutterFilter=0 Min=0 Max=6
AutoShutterMin=10 Min=5 Max=100000
AutoShutterMax=1000 Min=5 Max=100000
DoubleShutter1=100 Min=5 Max=100000
DoubleShutter2=500 Min=5 Max=100000
ReflexionFilter=0 Min=-8 Max=8
EthernetPackNr=1 Min=0 Max=100
AutoTriggerFPS=500 Min=1 Max=10000
AutoTriggerError=0 Min=0 Max=2147483647
EthernetSendNoWait=1 Min=0 Max=1
SubSample=0 Min=0 Max=2
MedianFilter=0 Min=0 Max=4
SpeckleFilter=3 Min=0 Max=4
SkipNoLaserPoints=1 Min=0 Max=1
CameraSNR=7900190 Min=7900190 Max=7900190
LibVersion=55 Min=55 Max=55

Product Parameter:
-----------------
AdjustSensorRoi=1 Min=0 Max=2
J00_ProductType=1 Min=0 Max=1000000
J00_TargetAngle=0.0 Min=0 Max=180
J00_AngleTolPos=5.0 Min=0 Max=180
J00_AngleTolNeg=5.0 Min=0 Max=180
```

```
J00_LineAlgoMode=1 Min=0 Max=1
J00_HistoLineFilter=2 Min=1 Max=9
J00_PhiMin=0.0 Min=0 Max=180
J00_PhiMax=180.0 Min=0 Max=180
J00_MinNrLaserPoints=8 Min=2 Max=200
J00_PolygonNr=2 Min=0 Max=16
J00_PolygonPoint_X00_MM=-67.0 Min=-67 Max=70
J00_PolygonPoint_Z00_MM=88.0 Min=88 Max=249
J00_PolygonPoint_X01_MM=70.0 Min=-67 Max=70
J00_PolygonPoint_Z01_MM=88.0 Min=88 Max=249
J01_ProductType=1 Min=0 Max=1000000
J01_TargetAngle=0.0 Min=0 Max=180
J01_AngleTolPos=5.0 Min=0 Max=180
J01_AngleTolNeg=5.0 Min=0 Max=180
J01_LineAlgoMode=0 Min=0 Max=1
J01_HistoLineFilter=2 Min=1 Max=9
J01_PhiMin=0.0 Min=0 Max=180
J01_PhiMax=180.0 Min=0 Max=180
J01_MinNrLaserPoints=8 Min=2 Max=200
J01_PolygonNr=2 Min=0 Max=16
J01_PolygonPoint_X00_MM=-67.0 Min=-67 Max=70
J01_PolygonPoint_Z00_MM=88.0 Min=88 Max=249
J01_PolygonPoint_X01_MM=70.0 Min=-67 Max=70
J01_PolygonPoint_Z01_MM=88.0 Min=88 Max=249
```

5 Parameter Error List

Error Name	Error Number	Effect
ERR_NONE	0	success
ERR_COUNTER	-3002	Host Counter Error (missing command)
ERR_PARM	-3003	Illegal parameter value or type (out of range)
ERR_COMMAND	-3004	Not a valid Parameter ID
ERR_PARM_NAME	-3009	Did not find a valid parameter name in STRING command
ERR_PARM_RANGE_MIN	-3016	One or more parameter are out of range (below min)
ERR_PARM_RANGE_MAX	-3017	One or more parameter are out of range (above max)
ERR_CMD_STRG_LEN	-3010	Bad string length at STRING command
ERR_CMD_TYPE	-3011	Not a valid Command Type
ERR_CMD_DATA_LEN	-3012	Command length does not fit to received bytes

6 Measure Response (Data Mode)

Once the server has been initialized and in measure mode, it will send back to the client a result response that will contain the data depending on the selected MODE.

The Mode_ID is an enum value corresponding to the commands asked by the client, the counter has the same role as in the Command Header.

Data Mode	Effect	Sub Header (Bytes)	Sub Header	Data	Data size
1	grey image	8	<width(4)> <height(4)>	copy of the image memory in U8	Width * height * sizeof (U8)
2	laser line profile (Sensor values)	16	<Data_Type(4)> <Number_of_Point(4)> <StartX(4)> <dX(4)>	If Data_Type=0, data array coded in I16. If Data_type=1, data array in floating point	<Number_of_Point> * sizeof (I16 or float)
3	laser line profile and laser brightness	16	<Data_Type(4)> <Number_of_Point(4)> <StartX(4)> <dX(4)>	If Data_Type=0, data array coded in I16. If Data_type=1, data array in floating point	<Number_of_Point> * sizeof (I16 or float) + <Number_of_Point> * sizeof(U8)
4 5	laser line profile (world coordinates in mm)	4	<Number_of_Point(4)>	4: x0z0x1z1... 5: x0x1...z0z1...	<Number_of_Point> * 2 (x, y) * sizeof (float)
6	binary image	8	<width(4)> <height(4)>	copy of the image memory in U8	Width * height * sizeof (U8)
7	laser line profile, laser brightness and product results	16	<Data_Type(4)> <Number_of_Point(4)> <StartX(4)> <dX(4)>	If Data_Type=0, data array coded in I16. If Data_type=1, data array in floating point	<Number_of_Point> * sizeof (I16 or float) + <Number_of_Point> * sizeof(U8)
8	product results	4	<Data String Size(4)>	<1> AngleMeasure	Depending on product type different ASCII data sets will be send.
9	laser line profile in mm and product results	8	<Number_of_Point(4)> <Data String Size(4)>	<1> AngleMeasure	<Number_of_Point> * 2 (x, y) * sizeof (float) + Depending on product type different ASCII data sets will be send.

Description of the main Modes:

This chapter describes the states and the data that will be returned from the camera in the corresponding state.

After the camera has been set to a special state with the command CMD_MODE it may return data.

Mode 1 (MODE_TAKE_PIC):

Capture an image then send the array containing the header and the image.

- Size of the Sub Header: 8 Bytes
- Description of the Sub Header: <width (4 Bytes)> <height (4 Bytes) >
- Data: The data will be an exact copy of the image memory with size <width> *<height>, coded in U8.
- The number of images that will be returned depends on the previously sent command CMD_NBR_LINES. If this value was -1 the sending will be endless.
  Image transfer always can be stopped if there is a command „CMD_RESET“ .

Mode 2 (MODE_2D_POINTLIST):

Capture a frame and extract the laser line points in pixel coordinates, then send the array containing the header and the laser line points.

- Size of the Sub Header: 16 Bytes
- Description of the Sub Header:

- the data sent are I16 points (integer coded in 2 bytes):
- <Data_Type = 0 (4 Bytes)> <Number_of_Point(4 Bytes)> <StartX(4 Bytes)> <dX (4 Bytes)>

- the data are floating points (4 Byte float values) :
  <Data_Type = 1 (4 Bytes)> <Number_of_Point(4 Bytes)> <StartX(4 Bytes)> <dX (4 Bytes)>

- Data:

For each profile the Z-Position in pixels will be returned in either integer points or floating points.
The floating values are always in little endian format.

- Extracting laser line with the method set by CMD_METHOD and the type of point set by CMD_ALGORITHM

- The number of frames that will be returned depends on the previously sent command CMD_NBR_LINES. If this value was -1 the sending will be endless.
  Frame transfer always can be stopped if there is a command „CMD_RESET“ .

Mode 3 (MODE_2D_POINTLIST_AND_LASER_IMG):

Capture a frame, extract the laser line points in pixel coordinates, then send the array containing the header and the laser line points. Additionally, send an array representing the laser line information : each value of the array will be the sum of each column of the laser line image.

- Size of the Sub Header: 16 Bytes
- Description of the Sub Header:
  - For I16 data:
    <Data_Type = 0 (4Bytes)> <Number_of_Point (4 Bytes)> <StartX (4 Bytes)> <dX (4Bytes)>

  - For floating point data:
    <Data_Type = 1 (4Bytes)> <Number_of_Point (4 Bytes)> <StartX (4 Bytes)> <dX (4 Bytes)>
- Data:
  - For each profile the Y-Position in pixels will be returned in either short points or floating points followed by the laser line image information. The starting point and the shift in X can be used to go directly to some point in the data array.
  - As the laser line image and the 2D PointList have the same number of points, the data size is : Number_of_Point *sizeof(type) + Number_of_Point * sizeof(U8))
  - The array representing it will be sent in U8, the floating values are always in little endian format.

- Extracting laser line with the method set by CMD_METHOD and the type of point set by CMD_ALGORITHM
- The number of frames that will be returned depends on the previously sent command CMD_NBR_LINES. If this value was -1 the sending will be endless.
  Frame transfer always can be stopped if there is a command „CMD_RESET“
-

Mode 4 (MODE_2D_POINTLIST_MM): capture a frame and extract the laser line points in mm coordinates, then send the array containing the header and the laser line points.

- Size of the Sub Header: 8 Bytes
- Description of the Sub Header:
  - <Number_of_Point(4 Bytes)> <LineCounter(4 Bytes)>
  - the data are floating points (4 Byte float values) :
    <x> <y>
- Extracting laser line with the method set by CMD_METHOD and the type of point set by CMD_ALGORITHM
- The number of frames that will be returned depends on the previously sent command CMD_NBR_LINES. If this value was -1 the sending will be endless.
  Frame transfer always can be stopped if there is a command „CMD_RESET“ .
-

7 New Version Changes