Table of Contents

1 Protocol Description

Protocol version 49

1.1 Socket connection

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

The sensor acts as server.

IP: (camera IP see #IP.231)

Port: 1096 (Default)

1.2 Communication between Host and 3D camera sensor

Directly after establishing the connection the following communication is possible:

1. Commands from Host to Sensor
    - Settings Commands: Sets a parameter and gives an error message
    - Reading Commands: Gives the current value setting and range of the parameter

2. Responses from Sensor to Host
- Command Response after Setting Command
- Command Response after Reading Command
- Result Response during run mode

General it is possible to do the communication in ASCII or Binary format. The sensor starts in Binary Mode. Inside the camera menu or sensor setting file you can change the communication mode.

For an easy communication way, there is always a fixed HEADER size for every Host command. Depending on the communication format it is 4 times I32 values (16 Bytes) in Binary mode or 4 times 8 byte values (32 Bytes) in ASCII mode.

Example:

Host: CMD_STOP: (CommandID = 0, HostCounter = 123, Action = 1, Value = 0)

Sensor: Global Header: 100, CamCounter = 0, Size=4*8,

Meta Header: CommandID=0, MinVal=0, MaxVal=0, NewVal=0)

ASCII MODE: (s = SPACE)

Host: sssssss0sssss123sssssss1sssssss0

Sensor: sssss100sssssss0ssssss32sssssss0sssssss0sssssss0sssssss0

Binary MODE (little Endian):

Host: <0 (I32)> <123 (I32)> <1 (I32)> <0 (I32)>

Sensor: <100 (I32)> <0 (I32)> <32 (I32)> <0 (I32)> <0 (I32)> <0 (I32)> <0 (I32)>

1.3 Host Commands

1.3.1 Setting Command (Action = 0):

Description of the Command Header (Example Binary Mode):

1. <Cmd_ID (4Byte)>
2. <Host Counter (4 Byte)>
3. <Action (4 Byte)>
4. <Value (4 Byte)>

The Cmd_ID is the value corresponding to the command list (see next chapter).

The host counter is a global value defined by the host in the range of [0..99999999]. After reaching the max. value, the counter has to restart at 0. It is recommended to reset the value after each connection, meant to check if we are sending and receiving the command and the corresponding response.

The Action is always 0 in setting mode.

The Value will be used to set the new parameter value.

1.3.2 Reading Command (Action = 1):

Description of the Command Header (Example Binary Mode):

1. <Cmd_ID (4Byte)>
2. <Host Counter (4 Byte)>
3. <Action (4 Byte)>
4. <Value (4 Byte)>

The Cmd_ID is the value corresponding to the command list (see separate chapter).

The host counter is a global value defined by the host in the range of [0..99999999]. After reaching the max. value, the counter has to restart at 0. It is recommended to reset the value after each connection, meant to check if we are sending and receiving the command and the corresponding response.

The Action is always 1 (get) in reading mode.

The Value will be ignored at the reading command and is there in order to have always the same Header size.

1.4 Command List

Command name	**Nr.**	**Effect**
CMD_RESET	0	Stop image acquisition, reset host and camera counter and set NBR_LINES to 0
CMD_LASER_MODE	1	1: set laser on 0: set laser off
CMD_ROI_X	2	Set the ROI x starting point at X
CMD_ROI_Y	3	Set the ROI y starting point at X
CMD_ROI_DX	4	Set the horizontal ROI width received at X
CMD_ROI_DY	5	Set the vertical ROI width received at X
CMD_GAIN	6	Set the gain at X
CMD_SHUTTER	7	Set the shutter at X
CMD_TRIGGER	8	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)
CMD_MODE	9	-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
CMD_NBR_LINES	10	Set the number of frames that will be taken, if X equals -1 there will be endless capturing until CMD_RESET has been sent.
CMD_METHOD	11	Set the line extraction method: 0 center detection at pixel accuracy 1 BaryCenter detection 2 BaryCenter detection with speckle filter (recommended)
CMD_ALGORITHM	12	Set the Algorithm parameter 0 Fixed point algorithm (multiplied by 16 for sub pixel accuracy) 1 Floating point algorithm
CMD_THRESH	13	Set threshold for laser line detection
CMD_MINWIDTH	14	Set min. Laser Line Width
CMD_MAXWIDTH	15	Set max. Laser Line Width
CMD_EXPMODE	16	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)
CMD_LASERVAL	17	AUTO_EXPOSURE mode: Select Shutter time for an average laser brightness of laser value
CMD_MINSH	18	AUTO_EXPOSURE mode:

		Min. allowed Shutter time [µs]
CMD_MAXSH	19	AUTO_EXPOSURE mode: Max. allowed Shutter time [µs]
CMD_DOUBLE_SH1	20	DOUBLE_EXPOSURE mode: Shutter time 1 [µs] – main image shutter
CMD_DOUBLE_SH2	21	DOUBLE_EXPOSURE mode: Shutter time 2 [µs] – second image, if no laser detection in main image
CMD_FILTER	22	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)
CMD_ROI_X_MM	23	Read the ROI x starting point [mm]
CMD_ROI_Y_MM	24	Read the ROI y starting point at y
CMD_ROI_DX_MM	25	Set the horizontal ROI width received at X
CMD_ROI_DY_MM	26	Set the vertical ROI width received at X
CMD_LIB_VERSION	27	Get 3D Scanner Library Version
CMD_PLC_IN	28	Get all PLC inputs (Bit0 … Bit3)
CMD_PLC_OUT	29	Set all PLC outputs (Bit0 … Bit3)
CMD_ETHERNET_PACKAGE	30	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 !
CMD_AUTO_TRIG_PERIODE	31	Camera auto trigger mode in 1/10 frames per second 500 means 50 fps. See also CMD_AUTO_TRIG_ERR
CMD_AUTO_TRIG_ERR	32	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.
CMD_SAVE_SENSOR_DATA	33	Saves latest sensor settings in camera flash. Camera will start with this parameters
CMD_SEND_NOWAIT	34	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
CMD_SUBSAMPLE	35	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)
CMD_WORLD_ACCURACY	36	New sensors always have Kanatani calibration, which is faster and more accurate. This parameter is only used at Tsai calibration. 0: accurate float algorithm but slow 1: fast l40 algorithm with typical 0.1% inaccuracy but four times faster
CMD_SEND_INFO_PAR	37	Sends the parameters for lens calibration
CMD_READ_PRODUCT_FILE	38	reads the product parameter from the camera file "fd:/ProdPar.001"
CMD_CAMERA_SNR	39	Gets the serial number of the camera ID.
CMD_SET_PRODUCT_DATA	40	Set product parameter Host -> Cam. Use set mode (Action = 0) as Command Header and as X value the number of Bytes of the production file followed by the data itself.

CMD_SEND_PRODUCT_DATA	41	Send product parameter Cam -> Host. Use read mode (Action = 1) as Command Header. After the header size of 16 Bytes the product file data follows. The size of header and data are defined in the Command Header.
CMD_READ_SENSOR_FILE	42	reads the product parameter from the camera file "fd:/VC3DPar.001"
CMD_SET_SENSOR_DATA	43	Set sensor parameter Host -> Cam. Use set mode (Action = 0) as Command Header and as X value the number of Bytes of the production file followed by the data itself.
CMD_SEND_SENSOR_DATA	44	Send sensor parameter Cam -> Host. Use read mode (Action = 1) as Command Header. After the header size of 16 Bytes the product file data follows. The size of header and data are defined in the Command Header.
CMD_SAVE_PRODUCT_DATA	45	Saves latest product settings in camera flash. Camera will start with this parameters

1.5 Sensor Response

Each time the server (sensor) receives a command, it will send to the client an acknowledgment response containing the:

- Response Global Header
- Response Meta Header
- Response Data (optional)

1.5.1 Setting Response:

Whenever the Action is set to 0 (“set”) the following sensor response will transferred:

Response Global Header (Example Binary Mode):

1. <Response ID (4 Byte)>
2. <Cam Counter (4 Byte)>
3. <Meta Header and Data Size in Bytes (4 Byte)>

Response ID is always 101 after setting command.

The cam counter is a global value defined by the sensor in the range of [0..99999999]. After reaching the max. value, the counter will be reseted to 0.

Meta Header Size is 8 or more, if additional data are transferred.

Response Meta Header (Example Binary Mode):

1. <Cmd_ID (4Byte)>
2. <Error Nr. (4 Byte)>

Response Data

Optional binary data string.

Error Name	Error Number	Effect
ERR_NONE	0	success
ERR_ERROR	-1	Common error
ERR_COUNTER	-2	Host Counter Error (missing command)
ERR_PARM	-3	Illegal parameter value or type (out of range)
ERR_COMMAND	-4	Command not available

1.5.2 Reading Response:

Whenever the Action is set to 1 (“get”) the following sensor response will transferred:

Response Global Header (Example Binary Mode):

1. <Response ID (4 Byte)>
2. <Cam Counter (4 Byte)>
3. <Meta Header and Data Size in Byte (4 Byte)>

Response ID is always 100 after reading command.

The cam counter is a global value defined by the sensor in the range of [0..99999999]. After reaching the max. value, the counter will be reseted to 0.

Meta Header Size is 16 or more, if additional data are available.

Response Meta Header (Example Binary Mode):

1. <Cmd_ID (4Byte)>
2. <Min Value(4 Byte)>
3. <Max Value(4 Byte)>
4. <Current Value(4 Byte)>

Response Data

Optional binary data string.

The Header will return the actual command number, the minimum value, the maximum value and the current value, for every command that support the getdata function. The Getdata command is sent instead of the acknowledgment, if the command is interpreted as a “getting-command”.

1.5.3 Result Response during RUN time:

Once the server has received the command, it will send back to the client a result response that will contain the data asked depending on the selected MODE. It will have the following format:

Response Global Header (Example Binary Mode):

1. <MODE (4 Byte)>
2. <Cam Counter (4 Byte)>
3. <Meta Header and Data Size in Bytes (4 Byte)>

Mode ID is described in the next chapter.

The cam counter is a global value defined by the sensor in the range of [0..99999999]. After reaching the max. value, the counter will be reseted to 0.

Meta Header Size in Bytes.

Response Meta Header and Response Data (Example Binary 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.

Mode	Effect	Meta Header (Bytes)	Meta Header	Data	Data size
1	capturing and sending an image	8	<width(4)> <height(4)>	copy of the image memory in U8	Width * height * sizeof (U8)
2	Send a 2D laser line profile	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	Send a 2D laser line profile and laser image information	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	Send a 2D laser line profile in mm	8	<Number_of_Point(4)> <LineCounter(4)>	4: x0z0x1z1... 5: x0x1...z0z1...	<Number_of_Point> * 2 (x, y) * sizeof (float)
6	capturing and sending an binary image	8	<width(4)> <height(4)>	copy of the image memory in U8	Width * height * sizeof (U8)
7	Send a 2D laser line profile, laser image information 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	Send product results	8	<LineCounter(4)> <Data String Size(4)>	<1> AngleMeasure	Depending on product type different ASCII data sets will be send.
9	Send 2D laser line profile in mm and product results	12	<Number_of_Point(4)> <LineCounter(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 State Meta Header:

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 Meta Header: 8 Bytes
- Description of the Meta 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 Meta Header: 16 Bytes
- Description of the Meta 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 Meta Header: 16 Bytes
- Description of the Meta 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 Meta Header: 8 Bytes
  - Description of the Meta 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“ .

2 New Version Changes

Version 49:

- no change to version 48

Version 50:

- command CMD_MODE number 9 implemented.
- command CMD_CMD_SAVE_PRODUCT_DATA implemented.