EyeVision

Documentation
Image Transfer Protocol

Protocol version V.6

EVT Eye Vision Technology GmbH

Haid-und-Neu-Str. 7
D-76131 Karlsruhe
Germany

Tel. +49 721 626 905 82
FAX: +49 721 626 905 96

Copyright and Notices

EVT EyeVision Technology
Haid-und-Neu-Str. 7
76131 Karlsruhe
Germany

* EVT in short form means the EVT GmbH and or the EVT AG.

Table of Contents

Outline

The EyeVision camera is sending two or more UDP packets, if images are sent to a client. Basically the data stream contains a header packet and at least one data packet for the image content (depending on the size of the image, the selected region of interest, color depth, compression, …)

Header-UDP-Packet:

The Header-UDP-Packet is a sequence of binary and ASCII data blocks. It conforms to a sequence “HSI”, where “H” corresponds to a binary header, “S” to a character string which is built with a lot of substrings and “I” to a binary image data block.
The binary header ”H“ is 28 bytes long and is built in the following way:

Byte-Offset	Length(Bytes)	Data Type	Field	Content
0	2	Short Int	Packet number	Contains the UDP packet number belonging to the transmission. Each header packet has got the number 1.
2	2	Short Int	Number of UDP packets for image data	Number of image UDP packets. The transmission is done via Low-Byte/High-Byte (little Endian).
4	4	Long Int	Image number	During transmission the images are numbered This field contains the number of the image which is transmitted (little Endian).
8	4	Long Int	Number of overlay packets	Number of UDP packets for transmitting the symbolic overlay. The overlay packets are transmitted after the image packets. The packet numbers (s. first field) are counted up. Without overlay this value is 0.
12	1	Char	Protocol version	Version number of the transmission protocol which the camera us using. Actual value is 5.
13	7	String	Identifier	“EVTACP” as constant (NULL terminated)
20	4	Long Int	Packet size	Reference value of the UDP packet in bytes (little Endian)
24	4	Long Int	Number of attempts	Indicates the number of transmission attempts of the current packet. Thanks to this entry a packet can be explicitly identified with the analysis of the network traffic.

The character string S contains the meta-information of the transmitted image and of the EyeVision camera, which is sending the information. In the following way:

> “S1\n;S2\n;S3\n;S4\n;S5\n;L1;L2;L3;L4;L5;L6;L7;L8;L9;L10;L11;L12;L13;L14;L15;L16;L17;
> L18;L19;L20;L21;L22;L23;L24;L25;L26;L27;L28;L29;”

S1 to S5 are character strings followed by a Return-char “\n” and “;” as separator. L1 to L29 are numbers in ASCII separated with the separator “;”. After L30 follows a ”;“ and a binary null character for termination of the character string.

The field content in protocol version 6 is described here:

Field name	Content	Range of values	Description
S1	Camera name	Character string, not containing ";"	User-defined name of the EyeVision-camera (shown e.g. in the tool EyeControl)
S2	Inspection program name	Character string, not containing ";"	User-defined name of the activated inspection program.
S3	Global String (text string)	Character string, not containing ";"	The camera transmits the generated global string, which is built via the „abc" tool or with other commands generating a global string (named Text string or SendString).
S4	Timestamp	Character string as follows: "YYYY_MM_DD_HH_ MM_SS_MSMSMS"	Timestamp of the image. Year (4-digits), month (2-digits), Day (2-digits), hours (2-digits), minutes (2-digits), seconds (2-digits), milliseconds (3-digits)
S5	Camera-IP	Character string „xxx.xxx.xxx.xxx"	Camera IP address in ASCII.
L1	Image number	Integer (unsigned long) as ASCII-character string	Number of the actual image transmission.
L2	Image width (for transmission)	Integer "1" .. max. image width	Image width, which was selected by the user for transmission (in pixel).
L3	Image height (for transmission)	Integer "1" .. max. image height	Image height, which was selected by the user for transmission (in pixel).
L4	Color depth (for transmission)	Integer "1" or "3"	Color depth, which was selected by the user for transmission (in bytes). "1" = Greyscale image "3" = RGB-color image
L5	X-Position ROI	Integer "0" .. max. image width	The user can specify a region of interest (ROI) for transmission. The ROI is selected with the upper left point (X,Y) and the width "B" and the height "H". L5 contains the x value (in pixel). If no ROI is selected then the whole image is sent. L5 contains then the value 0.
L6	Y-Position ROI	Integer "0" .. max. image height	The user can specify a region of interest (ROI) for transmission. The ROI is selected with the upper left point (X,Y) and the width "B" and the height "H". L6 contains the y value (in pixel). If no ROI is selected then the whole image is sent. L6 contains then the value 0.
L7	Width ROI or rather original image	Integer "1" .. max. image width	The user can specify a region of interest (ROI) for transmission. The ROI is selected with the upper left point (X,Y) and the width "B" and the height "H". L7 contains the B value (in pixel). If no ROI is selected then the whole image is sent. L7 contains then the value 0.
L8	Height ROI or rather original image	Integer "1" .. max. image height	The user can specify a region of interest (ROI) for transmission. The ROI is selected with the upper left point (X,Y) and the width "B" and the height "H". L8 contains the H value (in pixel). If no ROI is selected then the whole image is sent. L8 contains then the value 0.
L9	Color depth original image	Integer "1" or "3"	Color depth, which is available on the camera (in bytes). "1" = Greyscale image "3" = RGB-color image
L10	Image section	Integer "1", "2" or "3"	"1" = Transmission complete image (s. L7 and L8)

			“2“ = Transmission static ROI (s. L5-L8) “3“ = Transmission dynamic ROI (s. L5 – L8)
L11	Image resolution	Integer “1“, “2“ or “3“	“1“ = original resolution (each pixel) “2“ = 1/4 resolution (each second pixel in X- and Y-direction) “3“ = 1/16 resolution (each fourth pixel in X- and Y-direction) is sent.
L12	Rewritable receiver	Integer “0“ or “1“	“0“ = Receiver can not be overwritten via RemoteRequest “1“ = Receiver can temporarily be overwritten via RemoteRequest.
L13	Time of transmission	Integer “1“ or “2“	“1“ = The image transfer command has done the image transfer immediately. “2“ = The image transfer was initialized at the end of the inspection program. The image buffer is cached.
L14	Image buffer	Integer “1“ or “2“	“1“ = A IM (Image Memory) is transmitted. The number of the IM depends on the setting in the inspection program in the image transfer icon (s. L16). „2“ = (not used at the moment, s. L15).
L15	NOK image storage number	Integer “0“ to ??? (not specified)	Number of the NOK image storage starting with “0“. If NOK image storage is not used this value is ”0“.
L16	IM-number	Integer ”0“ to number IMs	Number of the transmitted image Memory starting with “0”. If NOK image storage is used this value is ”0“.
L17	Image format	Integer “1“ or “2“	Color depth, which was selected by the user for transmission (in bytes). “1“ = Greyscale image “3“ = RGB-color image (redundant to L4).
L18	Compression for transmission	Integer “1“ or “2“	User-defined compression method: ”1“ = No Compression ”2“ = JPEG-Compression
L19	Transmission graphical overlay	Integer “1“, “2“ or “3”	“1“ = no graphical overlay “2“ = graphical overlay imbedded into the image (not implemented) “3“ = graphical overlay is sent symbolic in the header packet (not implemented).
L20	Line number Image transfer	Integer “1“ to max. line number of the inspection program.	Line number of the triggering image transfer icon in the inspection program. This information is used in order to distinguish the reason or origin of the image transfer, when more than one image transfer is used in the inspection program.
L21	Internal transmission condition	Integer “1“, “2“ or “3”	“1“ = send image if the logic of the inspection program is (OK). “2“ = send image if the logic of the inspection program is (NOK). “3“ = send image always
L22	External transmission condition	Integer “1“ or “3”	“1“ = send image always. “2“ = send image only after a RemoteRequest. The request can be sent permanently or just one time. With a request the receiver for the image transfer can be temporarily replaced (s. L12).
L23	Good Counter	Integer (unsigned long) as ASCII-character string	Number of cycles in the inspection program with the global result “OK” at the end.

L24	Bad Counter	Integer (unsigned long) as ASCII-character string	Number of cycles in the inspection program with the global result “NOK” at the end.
L25	Cycle Time	Integer (unsigned long) as ASCII-character string	Cycle time of the last execution of the inspection program in milliseconds.
L26	UDP-Packet size	Integer (unsigned long) as ASCII-character string	Selected UDP packet size for image and overlay transmission. The packet size between EyeView and camera is selected dynamically depending on the number of the lost packets.
L27	PConLan-Port	Integer (unsigned long) as ASCII-character string	Internal data: Socket-Port of camera process, where EyeView can adapt the UDP packet size. This data are not relevant for the end user.
L28	Vertical Pitch	Integer (signed long) as ASCII-character string	Additional information of the ordering of the lines in the memory. Not used at the moment.
L29	Image Memory Type Descriptor	Integer (unsigned long) as ASCII-character string	Not used at the moment.
L30	Result	Integer (signed long) as ASCII-character string	Contains a result, which is transmitted with an image.

The binary image data block I contains the first 28 bytes of the image buffer and if sent (L19 contains the value 3) the first 28 bytes of the overlay description. The corresponding data blocks have to be prefixed after receiving and analysis of the following UDP packets the image data or the overlay data, in order to get a valid image or a valid overlay illustration:

Byte-Offset	Length (Bytes)	Data Type	Field	Content
0	28	Char	Image data	This 28 bytes represent the beginning of the transmitted image buffer in the remaining UDP packets. If this is the first gray value, a color value or the begin of a jpeg block depends on the transmission settings.
28	28	Char	Overlay-data	If L19 contains value 3, then the camera sends also the overlay data in symbolic form. In this case the image data block contains also the first 28 bytes of the overlay data.

Why do we transmit image and/or overlay data already in the Header?
Speed optimization!
Thus it is possible to do the transmission of a complete image and/or overlay without any internal copy actions on the smart camera and it is still possible to separate the data blocks in different UDP packets.

Image-UDP-Packets:

The image UDP packets have got all the same structure HB, where “H“ stands for a 28 bytes long binary header followed by the binary block ”B“, which contains data of the image buffer. The length of such an UDP packet is max. 60.000 bytes and at least 1500 bytes. In each packet max. 59976 image data bytes can be transmitted.

The structure of the binary header “H” is similar to the UDP-header-packet:

Byte-Offset	Length (Bytes)	Data Type	Field	Content

0	2	Short Int	Packet number	Contains the UDP packet number belonging to the transmission. Each image packet contains numbers starting with 2 and are enumerated.
2	2	Short Int	Number of UDP packets for image data	Number of image UDP packets. The transmission is done via Low-Byte/High-Byte (little Endian).
4	4	Long Int	Image number	During transmission the images are numbered This field contains the number of the image which is transmitted (little Endian).
8	4	Long Int	Image data offset	This number contains the offset (in bytes) of the actual transmitted image UDP packet of the image data in the whole camera image. With this value the image-data bytes can be copied to the right place in the target image. The first offset is starting with 13, because the bytes with the numbers 0 to 12 have been transmitted already in the header packet.
12	1	Char	Protocol version	Version number of the transmission protocol which the camera us using. Actual value is 5.
13	7	String	Identifier	„“EVTACP” as constant (NULL terminated)
20	4	Long Int	Packet size	Reference value of the UDP packet in bytes (little Endian)
24	4	Long Int	Number of attempts	Indicates the number of transmission attempts of the current packet. Thanks to this entry a packet can be explicitly identified with the analysis of the network traffic.

The data in the binary block “B” depend on the transmission. The transmission can be done line-by-line of a grayscale image with one byte per pixel, or it can be a RGB representation with three bytes per pixel, or it can be a compressed HPEG of a greyscale image, …
In the case of the transmission of a JPEG compressed greyscale image the binary block contains before the JPEG format an unsigned long (4 bytes, little Endian) with the total-length of the compressed JPEG in bytes. The actual length of the UDP-packet is known by the use of the UDP transmission. If for the transmission of an image more than one UDP-packet is necessary (this is the case for complete images without compression) then the last data of an image are shared equally to the last two UDP packets, in order to avoid too small packet sizes.

Overlay-UDP-Packets:

The overlay-UDP-packets have got all the same structure HB, where “H“ stands for a 28 bytes long binary header followed by the binary block ”B“, which is containing data of the overlay. The length of such an UDP packet is max. 60.000 bytes and at least 1500 bytes. In each packet max. 59976 image data bytes can be transmitted. Usually the overlay needs only a few kilobytes.
The structure of the binary header “H” is similar to the UDP-header-packet:

Byte-Offset	Length (Bytes)	Data Type	Field	Content
0	2	Short Int	Packet number	Contains the UDP packet number belonging to the transmission. Each image packet contains numbers starting with (2 + number of image data packets) and are enumerated. An overlay-packets can be identified due to the number (> 1 + number of image data packets).
2	2	Short Int	Number of UDP packets for image data	Number of image UDP packets(!). The transmission is done via Low-Byte/High-Byte (little Endian). The number of overlay-UDP-packets is only transmitted initially in the first header-packet.
4	4	Long Int	Image number	During transmission the images are numbered This field contains the number of the image which is transmitted (little Endian).

8	4	Long Int	Overlay data offset	This number contains the offset (in bytes) of the actual transmitted overlay UDP packet of the overlay data in the whole overlay structure. With this value the overlay-data bytes can be copied to the right place. The first offset is starting with 13, because the bytes with the numbers 0 to 12 have been transmitted already in the header packet.
12	1	Char	Protocol version	Version number of the transmission protocol which the camera us using. Actual value is 5.
13	7	String	Identifier	“EVTACP” as constant (NULL terminated)
20	4	Long Int	Packet size	Reference value of the UDP packet in bytes (little Endian)
24	4	Long Int	Number of attempts	Indicates the number of transmission attempts of the current packet. Thanks to this entry a packet can be explicitly identified with the analysis of the network traffic.

The data in the binary block “B” depend on the transmission. The structure of the binary data of the structure of the overlay is not published here.
The actual length of the UDP-packet is known by the use of the UDP transmission. If for the transmission of an overlay more than one UDP-packet is necessary (rarely the case!) then the last data of an overlay are shared equally to the last two UDP packets, in order to avoid too small packet sizes.

Receive Images

In order to execute the reception of images, the UDP data-stream has to be scanned for a header-UPD-packet (packet number = 1). Then all packets with the same image number should be collected (s. first 28 header-byes). The sequence can not be guaranteed in the UDP transmission. Individual UDP-packets might get lost during transmission.

Starting with the first 28 image-bytes out of the UDP-header-packet the image-data (out of the image-UDP-packets) should be put together depending on the image-data offsets, in order to get a valid image buffer with the selected transfer format. If overlay-data are also sent then they can be identified with the overlay-header-description and can be ignored.