Streaming protocol

ArdeoNova Innovation Technologies, Inc

Revision date: **10/04/2018**

UDP streaming protocol uses separate UDP sockets for video streaming and profile streaming, by default video is streamed to port 13377 and profiles are streamed to 13378 port. Data frame layout looks like this:

HEADER	DESCRIPTION BLOCK	DATA BLOCK	FOOTER

Header block has fixed size of 16 bytes and has folowing structure:

Header marker	Version	Flags	Description size	Data size	Frame number
0xA5A5	2 bytes	2 bytes	2 bytes	4 bytes	4 bytes

All fields in header are in *big-endian* byte order, header marker is always *0xA5A5*, followed by *version field*.

Flags field contains frame flags.

Flags are:

```
#define FLAG_BINARY_DESCRIPTION      0x0001
#define FLAG_LE                      0x0002
```

Binary description flag determines format of description block, it is either binary or JSON.

If ***FLAG_LE*** is set then data block is in *little-endian* byte order.

Description size field contains size of ***description block*** in bytes.

Data size field contains size of ***data block*** in bytes.

Frame number field is a counter of sent frames.

Description block contains frame format info, dimensions and other. There is two formats for description block – JSON and binary, it is determined by ***FLAG_BINARY_DESCRIPTION*** in header flag field.

JSON description contains following fields:

```
#define FRAME_TYPE_FIELD            "type"
#define FRAME_FORMAT_FIELD          "format"
#define FRAME_WIDTH_FIELD           "width"
#define FRAME_HEIGHT_FIELD          "height"
#define FRAME_DENOM_FIELD           "denom"
#define PROFILE_COUNTER_FIELD       "pcounter"
#define MEASUR_COUNTER_FIELD        "mcounter"
#define GENERIC_COUNTERS_FIELD      "gcounters"
```

FRAME_TYPE field is an integer value and it determines what is stored in ***data block*** – video, or measurements.

FRAME_FORMAT field is an integer value that determines format of data block, values are specific to what is stored in data block.

FRAME_WIDTH integer, frame width.
FRAME_HEIGHT integer, frame height.
FRAME_DENOM integer, denominator for conversion to floating point types from integers.
PROFILE_COUNTER integer.
MEASUR_COUNTER integer.

Description layout in binary form

Binary description block is used when ***FLAG_BINARY_DESCRIPTION*** is set, binary description has following layout:

Frame type	Frame format	Frame width	Frame height	Frame denominator	Profile counter	Measurement counter
2 bytes integer	2 bytes integer	2 bytes integer	2 bytes integer	4 bytes integer	4 bytes integer	4 bytes integer

All values are in big-endian byte order.

Packet footer contains single field – packet checksum for a total of 2 bytes, footer size is not included in data size value of packet header. Algorithm used for calculation of checksum is CRC-16-CCITT(input is bit-reversed, value before final xor is bit-reversed) calculated over entire packet. Packet footer is always sent, but presence of checksum is optional, if no checksum were calculated, value of 0 is stored to checksum block.

Data transmission

Header is always sent as single 14-bytes datagram. Description block and data block are split into 1400 bytes datagrams, where first two bytes are the number of datagram, last block may have size less than 1400 bytes:

Header	Data chunk	Data chunk	...	Data chunk
14 bytes	1400 bytes	1400 bytes	...	<= 1000 bytes

Data chunk:

Block number	Data
2 bytes integer, big-endian byte order	1398 bytes

Numbers in data chunks allow to detect missing datagrams, and restore data in proper order.

Footer is sent as single footer-sized datagram.