SightLine APPLICATIONS

EAN-Metadata

PN: EAN-Metadata

3/21/2019

Contact:

Web: sightlineapplications.com

Sales: sales@sightlineapplications.com

Support: support@sightlineapplications.com

Phone: +1 (541) 716-5137

Export Controls

Exports of SightLine products are governed by the US Department of Commerce, Export Administration Regulations (EAR); classification is ECCN 4A994. The export summary sheet located on the support/documentation page of our website outlines customers responsibilities and applicable rules. SightLine Applications takes export controls seriously and works to stay compliant with all export rules.

Copyright and Use Agreement

Alerts

The following notifications are used throughout the document to help identify important safety and setup information to the user:

⚠ **CAUTION:** Alerts to a potential hazard that may result in personal injury, or an unsafe practice that causes damage to the equipment if not avoided.

ⓘ **IMPORTANT:** Identifies crucial information that is important to setup and configuration procedures.

Used to emphasize points or reminds the user of something. Supplementary information that aids in the use or understanding of the equipment or subject that is not critical to system use.

Contents

List of Figures

List of Tables

Appendix

1 Overview

This document describes commands used to set MISB metadata fields for SightLine products. SightLine products can support many simultaneous sources for metadata (serial, Ethernet). Sources can include GPS devices transmitting NMEA 0183 strings, raw KLV packets from gimbal, or directly set using SightLine's command and control protocol from an autopilot. KLV is then transmitted in an MPEG2-TS elementary stream along with H.264 video or by itself. Additionally, a limited set of Metadata can be selected for onscreen display.

This document also describes how to extract track box information from KLV metadata for user drawn track box overlays.

1.1 Associated Documents

EAN-Startup Guide 1500-OEM: Describes steps for connecting, configuring, and testing the 1500-OEM video processing board on the 1500-AB accessory board.

EAN-Startup Guide 3000-OEM: Describes steps for connecting, configuring, and testing the 3000-OEM video processing board on the 3000-IO interface board.

EAN-Ethernet and Serial Communication: Describes how to setup serial communications for cameras or other payload devices from SightLine hardware.

Interface Command and Control (IDD): Describes the native communications protocol used by the SightLine Applications product line. The IDD is also available as a local download on the Software Download page.

Panel Plus User Guide: A complete overview of settings and dialog windows in Panel Plus. Located in The Help menu of the Panel Plus application.

1.2 SightLine Software Requirements

ⓘ **IMPORTANT:** The Panel Plus software version should match the firmware version running on the board.

1.3 Application Bit Requirements

The functions described in this EAN require Application Bits (app bits) purchased from SightLine. App bits reside on the hardware unit and are enabled with a license file provided by SightLine at initial unit purchase or during a license upgrade process. License files use a hardware ID that is applicable to a specific hardware serial number. For questions and upgrade support contact SightLine Sales.

Table 1: Application Bits Requirement Table

Function	Minimum Software Version	Required Application Bit(s)
KLV Metadata Encoding	ALL	KLV. 0x0800
H.264 Video Compression	ALL	IP Encoding. 0x0004
Telemetry Output	ALL	Full Telemetry. 0x0040
Telemetry Output - Baseline	ALL	Base Telemetry. 0x8000

2 Related Commands

0x13	Set Metadata Data Values	Sets new metadata related to airframe and converts to KLV
0x14	Set Metadata Static Values	Sets new metadata that don't change very frequently
0x15	Set Metadata Frame Data Values	Sets metadata related to image and object inside image
0x62	Set Metadata Rate	Rate at which each element is injected into the stream (deprecated from firmware version 2.25, use 0x97 Set Tag Data Rate instead)
0x61	Set KLV Data	KLV blob data constructed by user
0x3B	Draw Object	Render text overlays automatically from Metadata
0x89	Set Appended Metadata	Set Application Specific Metadata (user data blob)
0x96	Set Tag Data	Set the data value of a KLV tag and KLV Subtag
0x97	Set Tag Data Rate	Set the Tag Data Rate for klv tags
0x98	Set Tag Source Selector	Set the tag source selector of a KLV Tag or range of tags.

2.1 Additional Related Commands

0x3E	Configure Communication Port	KLV Passthrough
0x76	Set Advanced Moving Target Indication Parameters	Control how many MTI tracks will be output in telemetry and KLV messages
0x29	Set Ethernet Display Parameters	Latest values are output in MPEG2-TS
0x60	Do SnapShot	images are created with EXIF metadata headers populated from MISB fields
0x90	Streaming Control	Enable / Disable streaming of KLV

3 MISB

Metadata is generated in accordance with MISB Standards in the following publications: ST 0102.10, ST 0601.11, ST 0603.2, ST 0604.3, and ST 0903.3. Publication standards are available on the MISB website.

Selectable KLV elements may be chosen from a superset of the Motion Imagery Sensor Minimum Metadata Set defined in publication ST 0902.3.

For conversion of values see *Table 1: UAS Datalink Local Metadata Set data elements* in publication ST 0601.1.

For non-string values, data encoding is big-endian as defined by publication ST 0603.2. For example, to encode Target Error Estimate CE90 – specified as a uint16 value – *identifier string length* should be set to 2 (byte 6 to the most significant byte of the value and byte 7 to the least significant byte of value).

Per publication ST 0601, version identifier fields for the UAS Local Data Set, Security Metadata Local Set, and Motion Imagery Track Metadata Local Set are emitted with elements of each local set:

- Implements UAS Datalink Local Set MISB ST 0601.7 (TAG 65 = 7)
- Implements Security Local Set MISB ST 0102.10 (TAG 22 = 10)
- Implements Video Moving Target Indicator and Track Local Set MISB ST 0903.3 (TAG 74, version = 3)

Values are calculated from platform angles, sensor angles, sensor position, frame center position, and slant range slant are supplied by **Set Metadata Values (0x13)** and Set Metadata Frame Data Values (0x15). All four corners are assumed to be at the same elevation as the frame center for these calculations.

Values are calculated from values supplied by **Set Metadata Values (0x13)** and **Set Metadata Frame Data Values (0x15)**, and by internal tracking position and box size. Target elevation is assumed to be the same as frame center elevation, and target latitude and longitude are calculated in a manner similar to the four corners. Track gate width and height are in pixels, derived directly from target box size.

Values are calculated from internal tracking information. Multiple targets (either user designated or automatically generated) are indicated. Pixel coordinates are in display coordinates suitable for directly rendering over streamed digital video.

UTC is the number of microseconds which have elapsed since midnight (00:00:00), January 1, 1970 (1970-01-01T00:00:00Z)

Sample code for working with streaming MPEG2-TS data is available in the PC Code Support examples available on the SightLine Software Downloads page » System Control Support - Decoder and Command / Control.

SightLine recommends updating the Video Decoder library (SLA-Decode) to take advantage of constant improvements and new features.

4 Metadata

4.1 Standard Supported Metadata Fields

MISB Standard specifies ST 0601.11 (UAS Datalink local set) specifies 105 tags. Following tags are supported by the sightline as default. Support for any non-default tags from the MISB set of 105 tags can be added by extending the tag support by using the command Set Tag Data (0x96).

Table 2: Metadata Fields

Current Metadata Data Values (0x13)	TAG	Subtag	Notes	Default Value[1]
UTC time	2	NA	Microseconds	2011-11-30 10:00:00
Platform Heading Angle	5	NA		
Platform Pitch Angle	6	NA		
Platform Roll Angle	7	NA		
Sensor latitude	13	NA		
Sensor longitude	14	NA		
Sensor True Altitude	15	NA	See Note[2]	
Sensor horizontal field of view	16	NA		
Sensor vertical field of view	17	NA		
Sensor relative azimuth angle	18	NA		
Sensor relative elevation angle	19	NA		
Sensor relative roll angle	20	NA		
Set Metadata Static Values (0x14)	**TAG**	NA	**Notes**	**Default Value**
Mission Identifier	3	NA	127 characters max	
Platform Designation	10	NA	127 characters max	h.264/KLV Demonstration
Image Source Sensor	11	NA	127 characters max	Image Source Sensor
Image Coordinate System	12	NA		
Security: Classification	48	1		1 - Unclassified
Security: Classifying country coding method	48	2		12 -STANAG 1059
Security: Classifying country	48	3		
Security: SCI/SHI information	48	4		
Security: Caveats	48	5		
Security: Releasing Instructions	48	6		
Security: Object Country Coding Method	12	NA		
Security: Object Country	13	NA		12 -STANAG 10
Motion Imagery Core Identifier	94	NA		ST-1204
Platform Tail Number	4	NA	127 characters max	
Target Error Estimate CE90	45	NA		
Target Error Estimate LE90	46	NA		
Generic Flag Data	47	NA		See below
Platform Call Sign	59	NA		

1 Unless otherwise specified, default value will be set to zero (0).

(Metadata Fields table continued)

Set Metadata Frame Data Values (0x15)	TAG	Subtag	Notes	Default Value
Frame center latitude	23	NA		
Frame center longitude	24	NA		
Frame center elevation	25	NA	See Note[3]	
Target width	22	NA		
Slant range	21	NA		
User-supplied metadata flags		NA		
Target location specification mode latitude / row	40	NA		
Target location specification mode longitude / column	41	NA		
Target location specification mode elevation	42	NA		
Target track gate height	44	NA		
Target track gate width	43	NA		
Offset corner latitude points 1	26	NA		
Offset corner longitude points 1	27	NA		
Offset corner latitude points 2	28	NA		
Offset corner longitude points 2	29	NA		
Offset corner latitude points 3	30	NA		
Offset corner longitude points 3	31	NA		
Offset corner latitude points 4	32	NA		
Offset corner longitude points 4	33	NA		
VMTI LDS targets	74	NA	variable	
Sensor Ellipsoid Height	75	NA	See Note[2]	
Frame Center Height Above Ellipsoid	78		See Note[3]	
Set Appended Metadata (0x89)	**TAG**		**Notes**	**Default Value**
Application Specific Metadata Appending	100		127 characters max	

4.1.1 Multi Byte Data Order in SLA Command Messages (Endianness)

The SLA Command IDD specifies that all multi-byte fields are encoded with the Least Significant Byte (LSB) first, followed by MSB. This is true for the following KLV data messages as well. While the MISB specification requires big-endian format for multi-byte data fields, the SLA command requires little endian format for multi-byte data fields.

For example, a latitude value of 305419896 (0x12345678) will be sent in a **SetMetadataValues (0x13)** message in the order 78,56,34,12. This data will be output in the MPEG2-TS KLV stream in big-endian format as per the MISB standard.

4.1.2 Set KLV Data (0x61)

Set the user specified KLV blob data so it can be sent with MPEG2-TS stream. The KLV data will be sent along with the next H.264 frame.

[2] *[MISB 0601] For legacy systems, Tag 15 and Tag 75 | Tag 104 are allowed with preference for Tag 75 | Tag 104. For improved modeling accuracy use Sensor Ellipsoid Height (Tag 75) or Sensor Ellipsoid Height Extended (Tag 104).*

[3] *[MISB 0601] For legacy purposes, both MSL (Tag 25) and HAE (Tag 78) representations of Frame Center Elevation MAY appear in the same MISB ST 0601 packet. A single representation is preferred favoring the HAE version (Tag 78).*

The KLV blob approach does not touch the data. It is pushed out of the KLV stream, so the multi-byte data is in the big-endian format as required by MISB. See Appendix A for an example of this command.

When using this feature, consider disabling the transmission of the built-in KLV metadata unless KLV blob parsing is enabled as described in section 8. Set Frame step (Byte 12) to 0 using command Set Metadata Rate (0x62).

4.1.3 Set Metadata Rate (0x62)

This command is deprecated in v2.25. See Set Tag Data Rate (0x97). Sets the frame step rates at which the KLV metadata is output. This allows users to control the bandwidth used by the KLV elementary stream in accordance with ST 0902.6.

4.1.4 Metadata Overlays (0x84)

Enable / disable predefined overlay graphics for Metadata.

*Starting in 2.24 this has been replaced by more flexible features in **Draw Object (0x3B)**.*

4.1.5 Set Appended Metadata (0x89)

Set application specific appended metadata. For an example of generating this message see Appendix B.

This binary message content uses tag 100. The content must be in MISB KLV metadata format. See below for content format. Following this format is required for this message to be decoded by a MISB standard ground station.

ST0601.11.pdf

See section 6.5.4 Segment LS/Amend LS within the UAS Datalink LS

ST1607.pdf

4.1.6 Extending Default Metadata Set Using Set Tag Data (0x96)

MISB Standard specifies ST 0601.11 (UAS Datalink local set) specifies 105 tags. A subset of tags are supported by the SightLine by default using the **Set Metadata Values (0x13)**, **Metadata Static Values (0x14)**, and **Set Metadata Frame Values (0x15)**. Support for any non-default tags from the MISB set of 105 tags can be added by extending the tag support by using the command Set Tag Data (0x96).

Set Tag Data (0x96) can be used to set any value to a KLV tag and subtag as specified in 0601.11. It can also be used to set values to one of the supported default KLV tags as well.

Set Tag Data (0x96) is only supported from 2.25 firmware. For an example of generating this message see Appendix C.

4.1.7 Set Tag Data Rate (0x97)

Sets the frame step rates at which KLV metadata is output in the MPEG2-TS stream. This allows users to control the bandwidth used by the KLV elementary stream in accordance with ST 0902.6. For an example of generating this message see Appendix C. Setting the FrameStep (rate) to zero for any tag implies that it will never be sent.

4.1.8 Metadata Tag Source Selector (0x98)

Select the source for a KLV tag or a range of KLV tags. Once a source is set for a KLV tag, only that source can set values for that tag. Attempts by any other sources to set the value for the same KLV tag are ignored. This allows multiplexing KLV values set by different sources. Source selector IDs are mutually exclusive and multiple sources to the same KLV tag cannot be set.

4.1.8.1 Valid Source Selector IDs:

Command ID	= 0x01	Used by KLV related SLA commands (0x13, 0x14, 0x15)
NMEA	= 0x02	Used by NMEA
VIDEOTRACK_INTERNAL	= 0x04	Used by VideoTrack internally to update KLV tags
KLV_BLOB	= 0x08	Used for KLV passthrough and **Set KLV Data (0x61)** to send the KLV blob mixed with KLVs from other sources
VBI	= 0x10	KLV set through VBI, internal source
TEST	= 0x20	For use with unit test, 3rd party apps. Internal any APP using SLA command and control for KLV should set selector as 1
GIMBAL	= 0x40	Reserved

See Appendix E for more details and example on setting the source multiplexer.

5 ISO and MISB References

MPEG2-TS: ISO/IEC 13818-1

KLV Metadata

MISB standards:

0102.10 Security Metadata Universal and Local Sets for Digital Motion Imagery

0601.7 UAS Datalink Local Set

0603.2 Common Time Reference for Digital Motion Imagery Using Coordinated Universal Time (UTC)

0604.3 Timestamps for Class 1/Class 2 Motion Imagery

0903.3 Video Moving Target Indicator and Track Metadata

Selectable KLV Elements

The Motion Imagery Sensor Minimum Metadata Set is defined in the Motion Imagery Sensor Minimum Metadata Set in publication ST 0902.3.

6 User Rendering of Track Box Overlays

This section references the Panel Plus software and requires a connection to the video processing board. See the 1500-OEM Startup Guide or the EAN-Startup Guide 3000-OEM for connection instructions.

It is possible to enable VMTI targets in the metadata stream and then extract this data from the resultant KLV encoded video for user rendering track boxes. This eliminates the need for a separate communication channel for receiving SLA Protocol messages for displaying track boxes.

To enable this feature:

1. Much of the KLV data is enabled by default, however the VMTI data is not. Panel Plus main menu » *Configure* » *KLV setup* » *Frame Values.* Check the *VMti Targets* box.

   Be sure to enable the frame step for VMti data to a non-zero value.

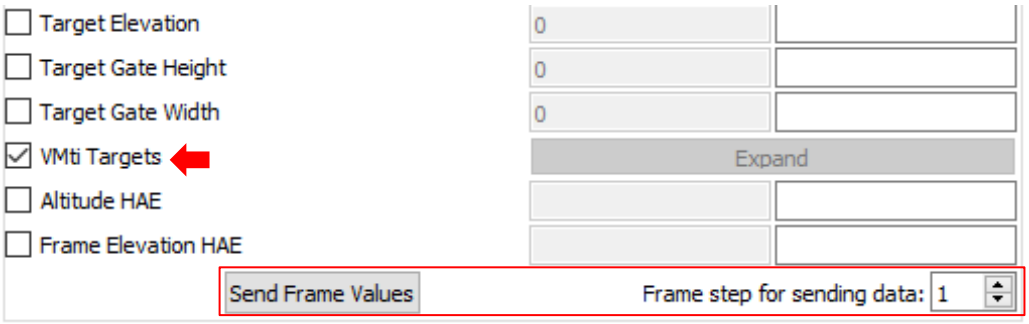
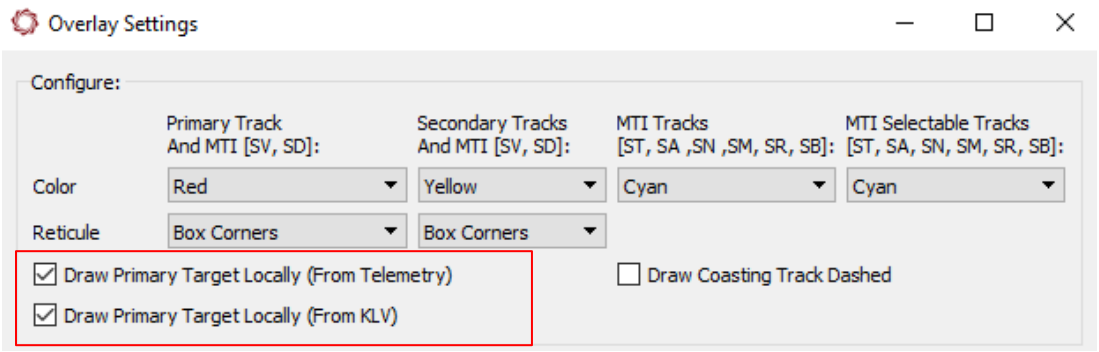


Figure 1: KLV Data Fields in Panel Plus

   Use the SightLine SLA-Decode library to register for a callback to provide the decoded image and the KLV data.

   The KLV data is returned in the structure defined in SLAKlv.h

2. Panel Plus main menu » *Configure* » *Overlays.*
3. Enable the following checkboxes shown in the *Overlay Settings* dialog.



4. To convert from VMTI to display coordinates, use a similar code that is shown in Figure 1 that gives the track box center, width and height.

   If the video is scaled in any way, it should be accounted for in the code.

```
if(m_Data.VMti.nTargets >= 1){
  s32 rowTop = m_Data.VMti.Target[0].BoundingBoxTopLeftPixelNumber / m_Data.VMti.FrameWidth;
  s32 colTop = m_Data.VMti.Target[0].BoundingBoxTopLeftPixelNumber - (rowTop*m_Data.VMti.FrameWidth);

  s32 rowBot = m_Data.VMti.Target[0].BoundingBoxBottomRightPixelNumber / m_Data.VMti.FrameWidth;
  s32 colBot = m_Data.VMti.Target[0].BoundingBoxBottomRightPixelNumber - (rowBot*m_Data.VMti.FrameWidth);

  width = colBot - colTop;
  height = rowBot - rowTop;
  row = ((rowTop+rowBot)>>1);
  col = ((colTop+colBot)>>1);
}
```

Figure 2: VMTI Data to Display Coordinates Sample Code

5. Save and activate the settings:
   a. Main menu » *Parameters* » *Save to Board*.
   b. Main menu » *Reset* » *Board*.
   c. Wait for the system to boot, and then reconnect to the board.

7 KLV Passthrough

See the EAN-Ethernet and Serial Communication document for more information.

Formatted KLV metadata is generated by an external processor. The KLV data is parsed for the Universal Key (16-bytes), the length field, and any remaining data. There is no other parsing or validation of the KLV data unless it is explicitly enabled through the selector.

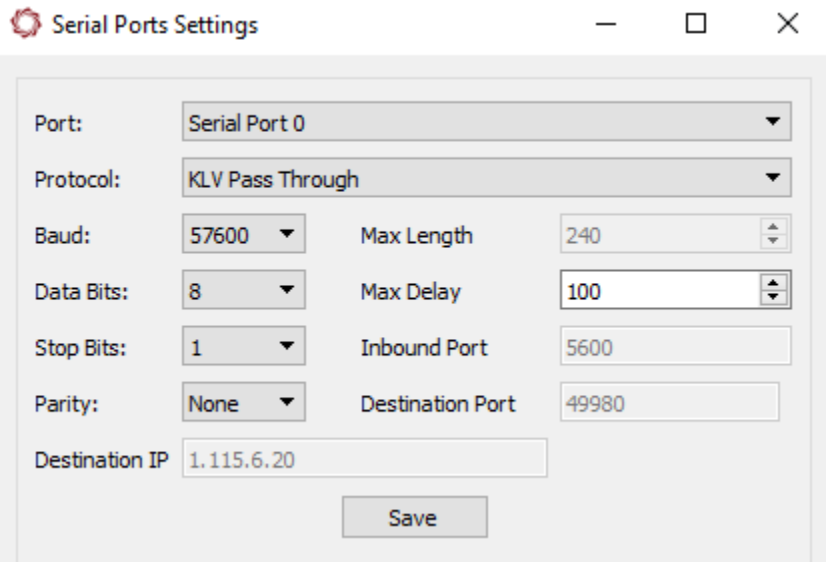
When using this mode, internally-generated KLV metadata is skipped for the frame that received the data on the serial port, and then reengages for the next frame except if parsing of KLV blob is enabled as described in section 8. When passing raw KLV data to the serial port, no additional SightLine headers or checksums are necessary.

As an alternative to sending raw KLV over the serial port or ethernet port, use the **Set KLV Data (0x61)** command from the SightLine Protocol, which allows the data to be sent over an Ethernet connection along with other SightLine commands.

7.1 KLV Passthrough - Serial Port Example

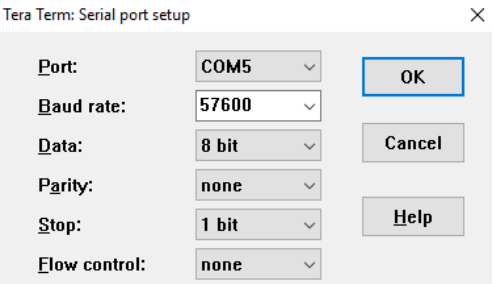
The Tera Term emulator program is used is this example.

1. Connect to the board through Panel Plus.
2. Configure *Serial Port 0* for KLV Passthrough using the settings shown.



3. Set *Max Delay* (default is 100 ms). This is the time the serial port waits to accumulate data.
4. Click *Save* to transmit the configuration to the hardware.
5. Click *Yes apply settings* to save the parameters and reboot the system. The parameters will take effect after the system reboot.

   The system is now ready to receive a block of KLV data over the serial port and send it out on the MPEG2-TS stream.
6. Open Tera Term and connect to COM port. Verify serial port settings.

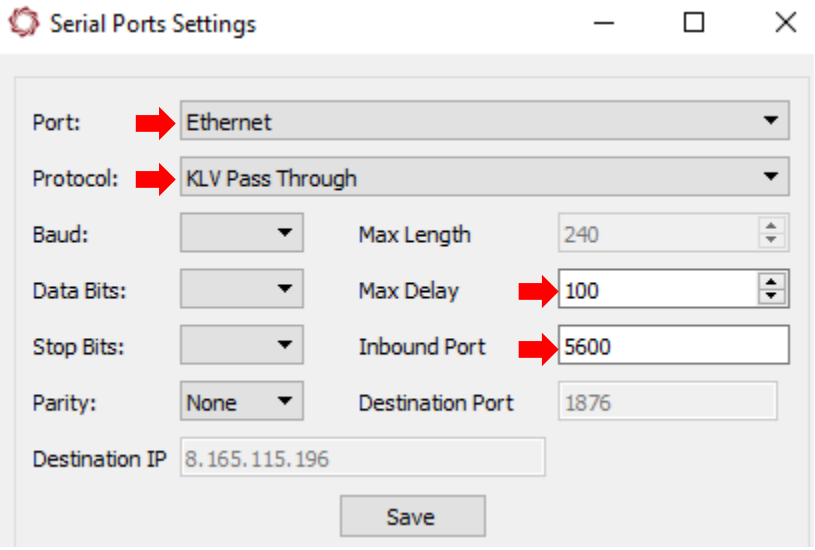


7. Copy the KLV blob and paste it in the Tera Term console (see Appendix).
8. Use the SLADecode sample from the SightLine web site to validate that the KLV passthrough is working (see Figure 3 below).

7.2 KLV Passthrough - Ethernet Port Example

KLV Passthrough over ethernet port is supported only through serial port 15 (Ethernet). This port is reserved for Ethernet use.

1. Connect to the board through Panel Plus.
2. Configure serial port 15 (Ethernet) for KLV Passthrough using the settings shown.



3. Set *Max Delay* (default is 100 ms). This is the time the serial port waits to accumulate data.
4. Choose the inbound UDP port. Make sure to choose an unused UDP port.
5. Ignore all other parameters.
6. Click on *Save* to transmit parameters to the board.
7. Click *Yes apply settings* to save the parameters and reboot the system. The parameters will take effect after the system reboot. The system is now ready to receive a block of KLV data over the ethernet port and send it out on the MPEG2-TS stream.
8. Send a properly structured block of KLV data using a program which can send data over a UDP port. See Testing KLV Passthrough over Serial or Ethernet section in EAN-Ethernet and Serial Communication.
9. Use the SLADecode sample from the SightLine web site to validate that the KLV passthrough is working (Figure 3).

```
280 void myCallback(void *context, SLAImage *image, KLVData *klv, KLVData *recent)
281 {
282   CallbackContextType *p = (CallbackContextType*)context;
283
284   // Display the video frame (will blank for null)
285   if (klv && image)
286     p->display->Show(image, recent->Utctime);
287   else
288     p->display->Show(image, 0);
289
290   if(!image) {
291     p->decoder->StopSaving();
292     return;
293   }
294
295   if(klv && image) {
296     //    OutputDebugString(A2W("    Got KLV\n"));
297     static int count = 0;
298     count++;
```

Figure 3: SLADecode Sample

8 Parsing KLV BLOB from Set KLV Data (0x61) or KLV Passthrough

The KLV blob received through **Set Klv Data (0x61)** or KLV passthrough mode does not get validated. It is sent once and does not get mixed with the actual KLV packets that are received and configured through other methods.

The system can parse the received KLV blob, validate it, and then send it as part of the built in KLV. This feature is enabled by:

- setting the ENABLE_KLV_BLOB_PARSING system value to 1,
- configuring the KLV tags to receive through the KLV blob using the KLV Source Multiplexer command.

Users should configure the KLV Source Multiplexer (0x98) command to enable KLV passthrough for a set of tags. The rate for the tags can be set through Set Tag Data Rate or Set Metadata Rate.

Set the selector ID as KLV_BLOB for tags that need to receive data from the KLV blob. Tags received from FIP or Video Track should be set to the corresponding selector. The KLV Source Multiplexer chooses the KLV tag values for tags based on the selector, and then sets interleaving tags parsed from KLV blob with other KLV tags.

9 KLV Blob Examples

```
Length = 43
Contains UTC (Tag 2), Sensor Latitude (Tag 13), and Sensor Longitude (Tag 14)

06 0e 2b 34 02 0b 01 01 0e 01 03 01 01 00 00 00 1a 02 08 00 05 70 30 04 0e 42
eb 0d 04 3f f9 00 1b 0e 04 3f f9 00 1b 01 02 c9 a2
```

10 Questions and Additional Support

If you are still having issues and require additional support, please contact Technical Support. Additional support, documentation and Engineering Application Notes (EANs) can be found on the Support pages of the SightLine Applications website.

Appendix A - Set KLV Data (0x61) Example

This command is in firmware 2.24.10 and later.

The following example describes how to include a user specified KLV blob of data in the Set KLV Data (0x61) command.

Set KLV Data (0x61) packet lengths may exceed 127 bytes. SLA commands with packet lengths greater than 127 bytes use the 2-byte packet length format. See the Sightline Applications IDD for details.

To turn off other KLV metadata sources, use the following **Set MetaData Rate (0x62)** command:

- 3000-OEM = 51,AC,0D,62,00,00,00,00,00,00,00,00,82,00,C0
- 1500-OEM = 51,AC,0B,62,00,00,00,00,00,00,00,00,00,FA

There are some KLV metadata sources that are currently not disabled with these commands and can be ignored. See SLATagDataRate (0x97).

KLV Blob

The KLV blob in the example shown below is from sightline EAN-Ethernet and Serial Communication document and contains the following data:

- 16-byte universal key: 06 0e 2b 34 02 0b 01 01 0e 01 03 01 01 00 00 00
- 3-bytes of BER long form encoding for the length (0x183): 92 01 83
- 387 bytes of data (0x183)

The following symbols are used represent the KLV blob length and the klv blob:

- <klvblob length> = 16 + 3 + 387 = 406 (0x196) bytes
- <klvblob> =

Table A1: Raw KLV Blob

```
06 0e 2b 34 02 0b 01 01 0e 01 03 01 01 00 00 00 82 01 83 03 7f 49 73 73 75 65 20 73 70 65 65 64
69 6e 67 20 74 69 63 6b 65 74 73 20 2d 2d 20 74 68 69 73 20 69 73 20 61 20 6c 6f 6e 67 20 73 74
72 69 6e 67 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0a 7f 53 4c 41 20 53 75 70 65 72 20
45 61 67 6c 65 20 2d 2d 20 74 68 69 73 20 69 73 20 61 20 6c 6f 6e 67 20 73 74 72 69 6e 67 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0b 7f 41 6c 74 69 63 61 6d 20 27
32 37 20 2d 2d 20 74 68 69 73 20 69 73 20 61 20 6c 6f 6e 67 20 73 74 72 69 6e 67 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

SetKLVData Command Format

The following example describes how to create a 0x61 command using the following syntax:

51 ac <len> 61 82 00 <klv blob> <chk sum>

<len> will be a 2-byte length since it is > 127

<len> = 1(61) + 2(82 00) + <klvblob length> + 1(chksum)

<len> = 1 + 2 + 406 + 1 == 410 bytes (0x19A)

<displayId> = 82 00 (for 1500 use 82 00 since this is ignored) order is LSB MSB

(3000 only) Network Display ID (0x0002=Net0, 0x0080=Net1,0x0082=both) (see the IDD)

If <displayId> is changed from this value, the checksum will need to be recalculated.

Extended length bytes from the IDD:

yy=Upper 7 bits of <len> is 0x036 m

xx=Lower 7 bits of <len> is 0x1A, with upper bit set (to indicate extended message) == 0x9A

The SetKLVData message will be: 51 ac 9a 03 61 82 00 <KLV Blob> cksum

The cksum is calculated on data starting with message ID (including 61). For this example, the cksum will be 80.

The total length of the message (including the 51 ac 9a 03 and cksum) will be 414 bytes.

The 414 bytes can be sent to the SightLine hardware and included in the network video stream through Panel Plus (header and checksum have already been added). From the main menu go to *File* » *Send Command.* Copy and paste command bytes into the *Send Raw Message* field.

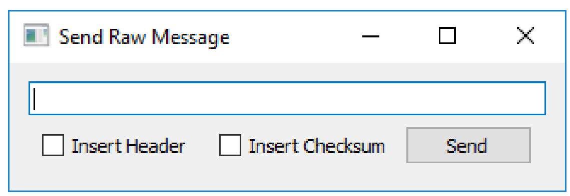


Figure A1: Send Command

If the parameters change you can use Panel Plus to calculate the checksum. Omit the final 80 in the data below, and then check the *Insert Checksum* checkbox in the *Send Raw Message* window.

```
51 ac 9a 03 61 82 00 06 0e 2b 34 02 0b 01 01 0e 01 03 01 01 00 00 00 82 01 83 03 7f 49 73 73 75 65
20 73 70 65 65 64 69 6e 67 20 74 69 63 6b 65 74 73 20 2d 2d 20 74 68 69 73 20 69 73 20 61 20 6c 6f
6e 67 20 73 74 72 69 6e 67 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0a 7f 53 4c 41 20 53 75 70 65
72 20 45 61 67 6c 65 20 2d 2d 20 74 68 69 73 20 69 73 20 61 20 6c 6f 6e 67 20 73 74 72 69 6e 67 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0b 7f 41 6c 74 69 63 61 6d 20 27 32 37
20 2d 2d 20 74 68 69 73 20 69 73 20 61 20 6c 6f 6e 67 20 73 74 72 69 6e 67 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 80
```

Appendix B - Set Appended Metadata (0x89) Example

This command is in firmware 2.24.10 and later. The following example describes how to create a 0x89 command using the following syntax:

51 ac <len> 89 <dataLen> <binaryData> <displayId> <chk sum>

Binary data will be embedded into KLV tag 100.

This example assumed 5 bytes of binary data (max is 127):

<binaryData> = 01 02 03 04 05

<dataLen> = 5

<displayId> = 82 00 (for 1500 use 82 00 since this is ignored) order is LSB MSB

(3000 only) Network Display ID (0x0002=Net0, 0x0080=Net1,0x0082=both) (see the IDD)

<len> = 1(89) + 1(X = data Len) + X(data Len) + 2(displayId) + 1(cksum)

1 + 1 + 5 + 2 + 1 = 10 = 0xA

Example command bytes: 51 AC 0A 89 05 01 02 03 04 05 82 00 3E

These bytes can be sent to the SightLine ardware and included in the network video stream through Panel Plus (header and checksum have already been added). Main menu *File* » *Send Command.* Copy and paste command bytes into the *Send Raw Message* field.

In 2.24.10 software the SLADecoder will decode the appended metadata and make the data available in the KLVData *struct.*

Appendix C - Set Tag Data (0x96) Example

This command is in firmware 2.25 and later. The following example describes how to create a 0x96 command using the following syntax:

51 ac <len> 96 <reserved1> <reserved2> <tagId> <tagSubId><reservedInternal ><displayId><klvdata> <chk sum>

<tagId> and <tagSubId> are the tags and subtags as specified in MISB Specification 0601.11. For the tags that do not have sub tags (for example platform heading angle) <tagSubId> value is ignored.

<ReservedInternal> should be set to 0

<Klvdata> is for the form

{ unsigned char tagLen, unsigned char tagVal[255] }

For example, if type of the tag is uin16 then <tagLen> will be 2 followed by first two bytes of <tagVal> filled with the <tagdata>. If the type of the tag is variable in length or if it is a string (similar to image coordinate system tag) then <tagLen> will be the length of the valid <tagVal> in the <tagVal> buffer.

Example 1:

Command bytes: 0x51 0xAC 0x0D 0x96 0x00 0x00 0x26 0x00 0x00 0x00 0x02 0x00 0x02 0x40 0x01 0xe2

Packet length	0x0D
Reserved1	0x00
Reserved2	0x00
Tag	0x26 (Density Altitude)
SubTag	0x00
ReservedInternal	0x00 0x00 (2 bytes - should be set to 0)
displayId	0x02 0x00 (2 bytes)
tagLen in klvData	0x02 (Density Altitude is of type uint16)
tagVal	0x40 0x01 (corresponds to a value of 320 in big-endian)
checksum	0xe2

Example 2:

Command bytes: 0x51 0xAC 0x1D 0x96 0x00 0x00 0x46 0x00 0x00 0x00 0x02 0x00 0x12 0x53 0x69 0x67 0x68 0x74 0x6C 0x69 0x6E 0x65 0x20 0x73 0x79 0x73 0x74 0x65 0x6D 0x73 0x00 0x74

Packet Length	0x1D
Reserved1	0x00
Reserved2	0x00
Tag	0x46 (Alternate platform name)
Sub Tag	0x00
ReservedInternal	0x00 0x00 (2bytes should be set to 0)
Display id	0x02 0x00 (2bytes for display id)
TagLen in klvData	0x12
tagVal	0x53 0x69 0x67 0x68 0x74 0x6C 0x69 0x6E 0x65 0x20 0x73 0x79 0x73 0x74 0x65 0x6D 0x73 0x00 (Corresponds to string Sightline systems.”)
Checksum	0x74

Bytes can be sent to the SightLine hardware and included in the network video stream through Panel Plus (header and checksum have already been added). From the main menu, go to *File* » *Send Command.* Copy and paste command bytes into the *Send Raw Message* field.

Appendix D - Set Tag Data Rate (0x97) Example

This command is in firmware 2.25 and later. The following example describes how to create a 0x97 command using the following syntax:

51 ac <len> 97 <reserved1> <reserved2> <mode> <tagId1><tagId2> <frameStep> <dispId><checksum>

<mode> is used to set rate for a range of tags.

0 Setting the rate of a single KLV tag. <tagId1> is used. <tagId2> is ignored in this case.

1 Setting the rate of a range of KLV tags to the frameStep value (tagId1, tagId1 +1, ….. , tagId2)), where <tagId1> is the start of the range and <tagId2> is the end of the range.

Example 1:

Command bytes: 0x51 0xAC 0x0B 0x97 0x00 0x00 0x01 0x02 0x69 0x01 0x00 0x02 0x00 0x27

Packet length	0x0B
Reserved1	0x00
Reserved2	0x00
mode	1
tagId1	0x02
tagId2	0x69 (105 max tag in misb 0601.11)
frameStep	0x01 0x00
displayId	0x02 0x00 (2 bytes)
checksum	0x27

All the supported tags in the range 0x02 to 0x69 will be sent at the rate of 1. KLV data for the supported tags will be sent in every frame.

Appendix E - Metadata Tag Source Selector (0x98) Example

This command is in firmware 2.25 and later. The following example describes how to create a **Tag Source Selector 0x98** command using the following syntax:

51 AC 0B 98 <reserved1> <reserved2> <mode> <tagId1><tagId2> <selector source> <dispId> <checksum>

<mode> is used to set rate for a range of tags.

0 Setting the selector of a single KLV tag. <tagId1> is used. <tagId2> is ignored in this case.

1 Setting the source of a range of KLV tags to the selector value (tagId1, tagId1 +1, ..., tagId2), where <tagId1> is the start of the range and <tagId2> is the end of the range.

Example 1:

Command bytes: 0x51 0xAC 0x0B 0x98 0x00 0x00 0x01 0x02 0x69 0x01 0x00 0x02 0x00 0x59

Packet length	0x0B
Reserved1	0x00
Reserved2	0x00
Mode	1
tagId1	0x02 – start at tag 02
tagId2	0x69 (105 - max tag supported now in 0601.11) - end at tag 105
source	0x01 0x00 - value corresponding to FIP
displayId	0x02 0x00 (2 bytes)
checksum	0x59

The command shown above will set the entire range of KLV to SightLine Command and Control. **Set Metadata Values (0x13)**, **Set Metadata Static Values (0x14)**, and **Set Metadata Frame Values (0x15)** commands use FIP and can be used to set the values for KLV tags.

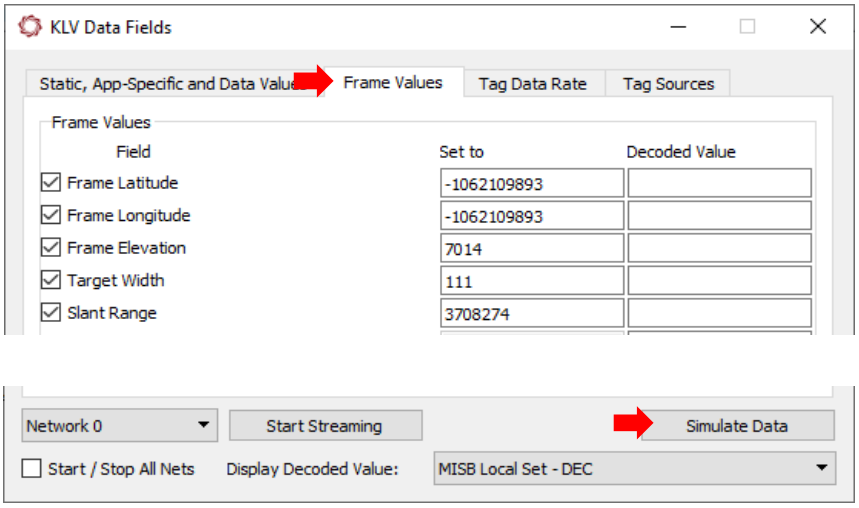
Appendix F - Validate SightLine KLV Streams

GV3.0 is a third-party tool that can be used to validate SightLine KLV streams. It is a licensed product and requires registration. If you do not have a license, you can request one via the License Request Form on their Downloads page. A specific government contract number is helpful.

SightLine has identified some instability issues with this application. If you are having issues and require additional support, please contact SightLine Technical Support.

Connect to an OEM board over an Ethernet connection using the Panel Plus application. See the EAN-Startup Guide 1500-OEM or the EAN-Startup Guide 3000-OEM connection instructions. In the example below the 3000-OEM board is used.

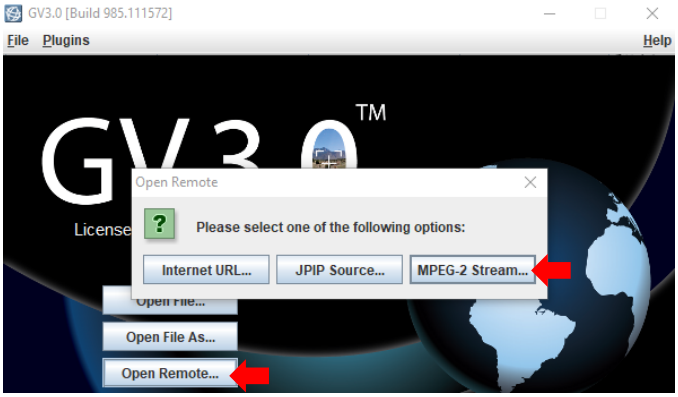
1. Set up a 3000-OEM to send simulated KLV data from Panel Plus to the OEM. From Panel Plus go to main menu » *Configure* » *KLV Setup.*
2. In the *KLV Data Fields* window click the *Frame Values* tab. Click *Simulate Data*.



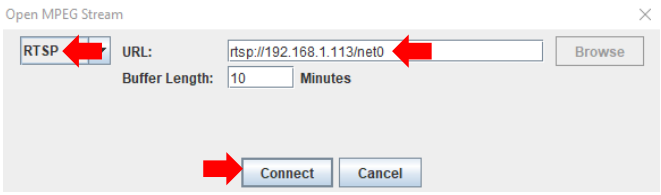
3. From the *KLV Data Fields* window, the returned data appears in the *Decoded Value* column on the *Frame Values* tab and the *Static, App-Specific and Data Values* tab.

Set to	Decoded Value
1549327714000000	1549327713985870
20206	20206
18021	18021
-7208	-7208
1020480690	1020480690
1020480690	1020480690
17184	17184
13471	13471
9830	9830
1717986918	1717986918
524940447	524940447
3770026849	3770026849

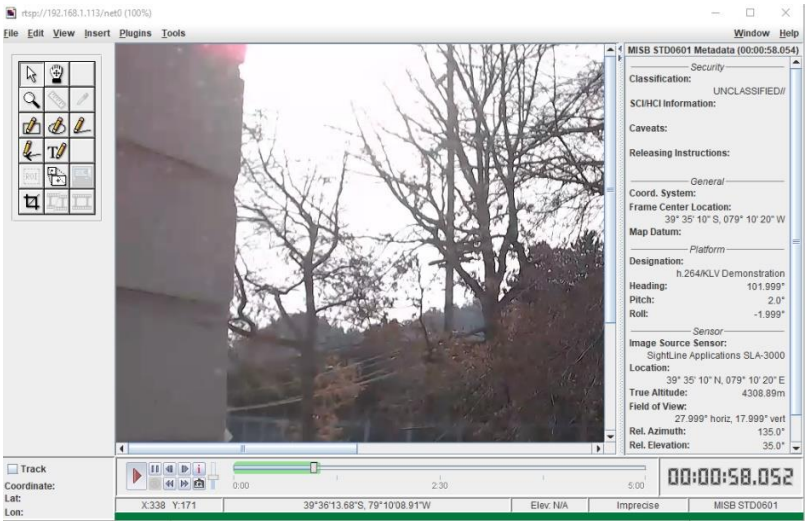
4. Install the GV3.0 application. During installation, select all plugins.
5. From Panel Plus plus click the *Compression* tab. From one of the RTP choices select a *Codec / Transport* that includes *TS*. Click *Send*.
6. Open the GV3.0 application. Select *Open Remote* » *MPEG-2Stream*.



7. From the *Open MPEG Stream* dialog window, select *RTSP* and enter the network address of the 3000-OEM board. Click *Connect*.



8. After a short delay verify video and metadata are displaying from the GV3.0 main screen.

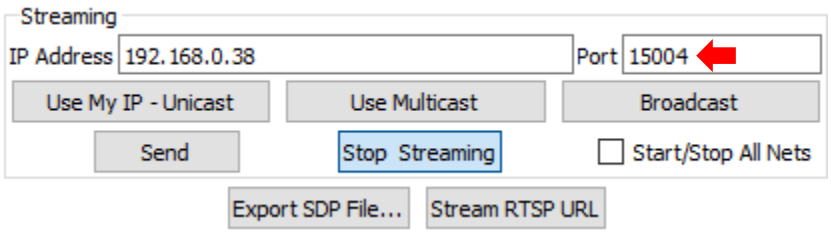


To stop streaming to GV3.0 go to File » Close. To exit GV3.0 go to File » Exit.

***IMPORTANT:** Closing the GV3.0 application using the close window icon (X) in the upper right corner can lock up the program.*

Viewing telemetry Data:

1. Start the GV3.0 application. Go to *Open Remote* » MPEG-*2 Stream*.
2. From the *Open MPEG Stream* dialog window, select *Unicast* and enter the port number found in the *Compression* tab in Panel Plus.



In some cases, streaming to the same PC in Panel Plus may need to be turned off to clear the port for the GV3.0 application. On the Multi Camera tab select the network that the PC and GV3.0 application are not running on.

The GV3.0 application will open and display the video stream and the KLV data that is being sent by Panel Plus and the Panel Plus KLV data simulator.

3. From the GV3.0 main menu » *Plugins* » *Alternate KLV Metadata Viewer* to open a window showing KLV metadata. Use the *Show Key as* drop down menu at the bottom of the window to show keys by *Name* or *UL Key* number.

Show Key as: Name

KLV (Std0601) Metadata (00:11:13.500)

Key	Value
unix time stamp	01:10:22.081 Z 05 Feb...
platform heading angle	103.99816891737238
platform pitch angle	3.999755851924192
platform roll angle	-3.999450666829432
platform designation	h.264/KLV Demonstrat...
image source sensor	SightLine Applications...
sensor latitude	34.562938676477856
sensor longitude	69.12587735295571
sensor true altitude	4310.711833371482
sensor horizontal field ...	29.998626688029297
sensor vertical field of ...	19.998168917372396
sensor relative azimut...	136.99999999650754
sensor relative elevati...	36.999999991152436
sensor relative roll ang...	323.00000010128133
slant range	4310.999066641321
target width	10.986495765621424
frame center latitude	-34.56293867647785

Show Key as: UL Key

KLV (Std0601) Metadata (00:12:10.300)

Key	Value
02	1549329079006180
05	111.99633783474479
06	11.999877925962096
07	-11.999877925962096
0A	h.264/KLV Demonstrat...
0B	SightLine Applications...
0D	34.205904805197335
0E	68.41180961039467
0F	4318.910505836576
10	37.9995422293431
11	27.999084458686198
12	145.00000002444722
13	45.00000002095476
14	315.00000007334165
15	4318.999127559131
16	18.921187151903563
17	-34.20590480519734