



SightLine
APPLICATIONS

EAN-Ethernet and Serial Communication

PN: EAN-Ethernet-and-Serial-Communication

9/8/2020

**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

© Copyright 2020, SightLine Applications, Inc. All Rights reserved. The SightLine Applications name and logo and all related product and service names, design marks and slogans are the trademarks, and service marks of SightLine Applications, Inc.

Before loading, downloading, installing, upgrading or using any Licensed Product of SightLine Applications, Inc., users must read and agree to the license terms and conditions outlined in the [End User License Agreement](#).

All data, specifications, and information contained in this publication are based on information that we believe is reliable at the time of printing. SightLine Applications, Inc. reserves the right to make changes without prior notice.

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.

i *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

1	Overview	1
1.1	Additional Support Documentation	1
1.2	SightLine Software Requirements.....	1
1.3	Third Party Software	1
1.4	Background.....	1
1.4.1	How does Passthrough Work.....	2
2	Configuring Serial Communication	3
3	Protocols	4
3.1	SLA Protocol - SightLine Command and Control.....	5
3.2	Aquarius and SLA Protocol	6
3.3	SLA Protocol No Telemetry	6
3.4	Port Not Used.....	6
3.5	TCP Passthrough.....	6
3.5.1	Serial Port Mapping	6
3.5.2	Defining TCP Passthrough Settings.....	6
3.6	Example Camera Control GUI Connection	8
3.6.1	Sony Control GUI.....	8
3.7	Raw Passthrough.....	9
3.8	KLV Passthrough.....	9
3.8.1	KLV Passthrough - Serial Port Example.....	10
3.8.2	KLV Passthrough - Ethernet Port Example.....	11
3.9	Serial-to-Serial Passthrough.....	12
3.9.1	Serial Passthrough Example - 1500-OEM	12
3.10	NMEA.....	15
3.10.1	Configure NMEA Input.....	16
3.10.2	Test NMEA Parsing.....	16
4	Max Length and Max Delay	17
5	Troubleshooting.....	17
5.1	Serial 1 to Serial 2 Test.....	18
5.2	NMEA Output Test in Tera Term.....	19



5.3 Additional Troubleshooting Tips 19

6 SightLine Command and Control 19

6.1 Additional Serial Port Configuration Notes..... 19

7 Questions and Additional Support..... 20

Appendix A - KLV Data 21

A1 Testing KLV Passthrough over Serial or Ethernet 21

Appendix B - Virtual Serial Port Emulator Software 23

Appendix C - Creating Custom Symlinks to Devices - 4000-OEM 26

List of Figures

Figure 1: Serial Passthrough Example..... 2

Figure 2: Serial Port Configuration Dialog 3

Figure 3: Serial Passthrough Settings Example..... 7

Figure 4: Apply New Settings Dialog - 3.01.xx and Earlier 7

Figure 5: Serial Port Settings - Raw Passthrough..... 9

Figure 6: TCP Passthrough Setup 10

Figure 7: Apply New Settings Dialog - 3.01.xx and Earlier 10

Figure 8: TCP Passthrough Setup 11

Figure 9: Apply New Settings Dialog - 3.01.xx and Earlier 11

Figure 10: SLADeCode Sample 12

Figure 11: Supply 3.3V to VIOSEL..... 13

Figure 12: Apply New Settings Dialog - 3.01.xx and Earlier 14

Figure 13: Configure Serial Ports 16

Figure 14: Apply New Settings Dialog - 3.01.xx and Earlier 16

Figure 15: Serial 1-to-Serial 2 On-Board Test 18

Figure 16: NMEA Test in Tera Term 19

Figure 17: Passthrough Test..... 20



List of Tables

Table 1: Summary of Field User Per Protocol 4
Table 2: Serial Port Mapping and Camera Names 6
Table 3: Current Metadata Data Values (0x13) 16
Table 4: Commands 19

Appendix Tables

Table A1: Sample KLV block (hex) 21



1 Overview

This document describes passthrough options for SightLine OEM hardware. It provides an example to configure and test using the Panel Plus application and other third-party software. While this document refers primarily to the 3000-OEM, the methods and features are similar for the 1500-OEM and 4000-OEM.

1.1 Additional Support Documentation

Additional Engineering Application Notes (EANs) can be found on the [Documentation](#) page of the SightLine Applications website.

The [Panel Plus User Guide](#) provides a complete overview of settings and dialog windows located in the Help menu of the Panel Plus application.

The Interface Command and Control ([IDD](#)) describes the native communications protocol used by the SightLine Applications product line. The IDD is also available as a PDF download on the [Documentation](#) page under Software Support Documentation.

1.2 SightLine Software Requirements

ⓘ IMPORTANT: The Panel Plus software version should match the firmware version running on the board. Firmware and Panel Plus software versions are available on the [Software Download](#) page.

1.3 Third Party Software

Information and links to virtual serial port emulator software and links to camera control software can also be found on SightLine website.

VSPE from [Eterlogic](#): Virtual serial port emulator used to create, debug, and test applications that use serial ports.

[Tera Term](#) (recommended) or [PuTTY](#): Terminal emulator programs used for debug output, or to issue commands on SightLine hardware.

Sony camera control software: A user Interface (UI) that allows camera configuration. Contact a Sony representative for GUI software used in this document.

1.4 Background

The serial passthrough capability provides a network bridge for serial communications to attached cameras or other payload devices. A host system (i.e., PC) communicates over a standard network to an attached camera to change settings such as aperture, exposure, focus, and zoom ([Figure 1](#)). It also allows data that has been received on these serial ports to be transmitted back to the host PC through the Ethernet connection. Other examples using serial ports might include autopilots, radios, pan-and-tilt turrets and more.

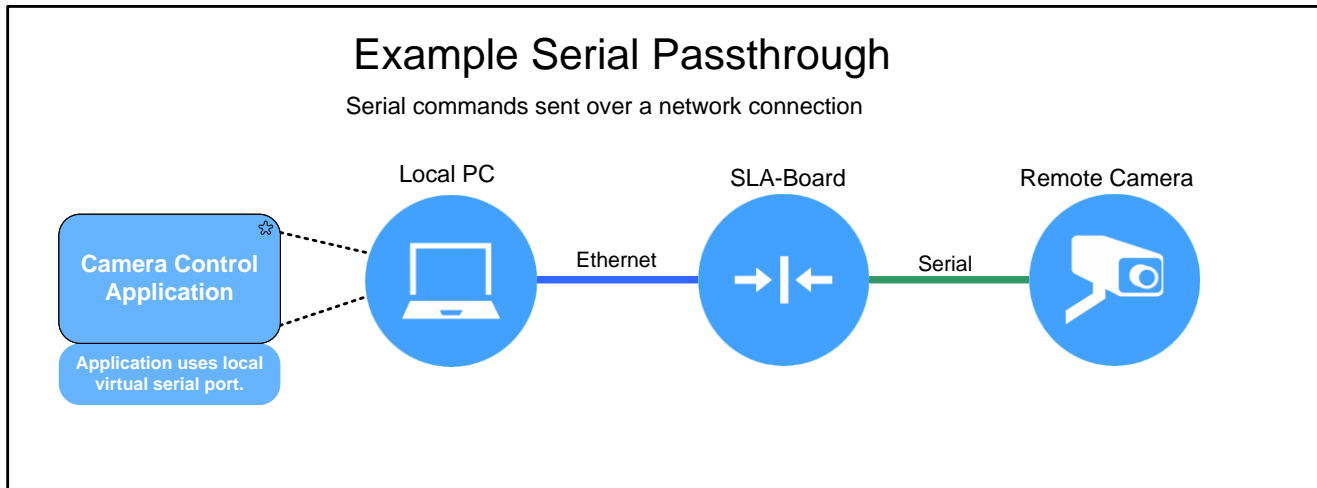


Figure 1: Serial Passthrough Example

1.4.1 How does Passthrough Work

To establish the network bridge, IP addresses and IP ports are defined on both the host PC and the SightLine board. Each IP address and port are mapped through software to a physical serial port on the board. Each serial port on the SightLine board can have its own Ethernet connection. The board receives data encapsulated in network packets and forwards those packets to the mapped serial connection. In the above example the host PC is running a virtual serial port application. To define all these elements for creating a passthrough configuration, see [Configuring Serial Communication](#).

Common Modes of operation:

- Ethernet to Serial: 2-way communication, where data is transferred to and from the serial port.
- Ethernet to Serial: 1-way communication where no reply from the serial device is expected. Examples include simple data transfer, KLV metadata, etc.
- Serial to Serial: 2-way communication



2 Configuring Serial Communication

The **Set Port Configuration (0x3E)** command is used to configure serial communication. The packet contains the following fields shown in the *Serial Port Settings* configuration dialog in Panel Plus for configuring the functionality:

Port: Index of the physical serial port being configured or a special Ethernet port configuration.

Port Settings: Standard serial port settings of baud rate, data bits, stop bits, and parity control how the data on the serial port is interpreted.

Max Length: Maximum number of bytes for serial input. When maximum is reached, any data received is sent to the destination.

Max Delay: Maximum timeout in milliseconds when reading from a port. When timeout is complete, any data received is sent to the destination port.

Protocol: See [Protocols](#).

Inbound Port: Local network port number where UDP/TCP packets are expected. Hardware opens a new socket to listen on this port. Must be greater than 0 (0 is reserved).

UDP Destination Address: IP address of host where Ethernet packets will be sent.

UDP Destination Port: Port number where remote host is listening. The hardware opens a new socket and sends data to the outbound destination IP address at the port number specified. The port number must be greater than 0 (0 is reserved).

UDP AttNav Port: (not shown in dialog) When the AttNav port number is set to a value other than zero, a network socket is created to listen on this port for ATTNV packets. The port defaults to 65100. Data received on this port is then sent to the defined serial port.

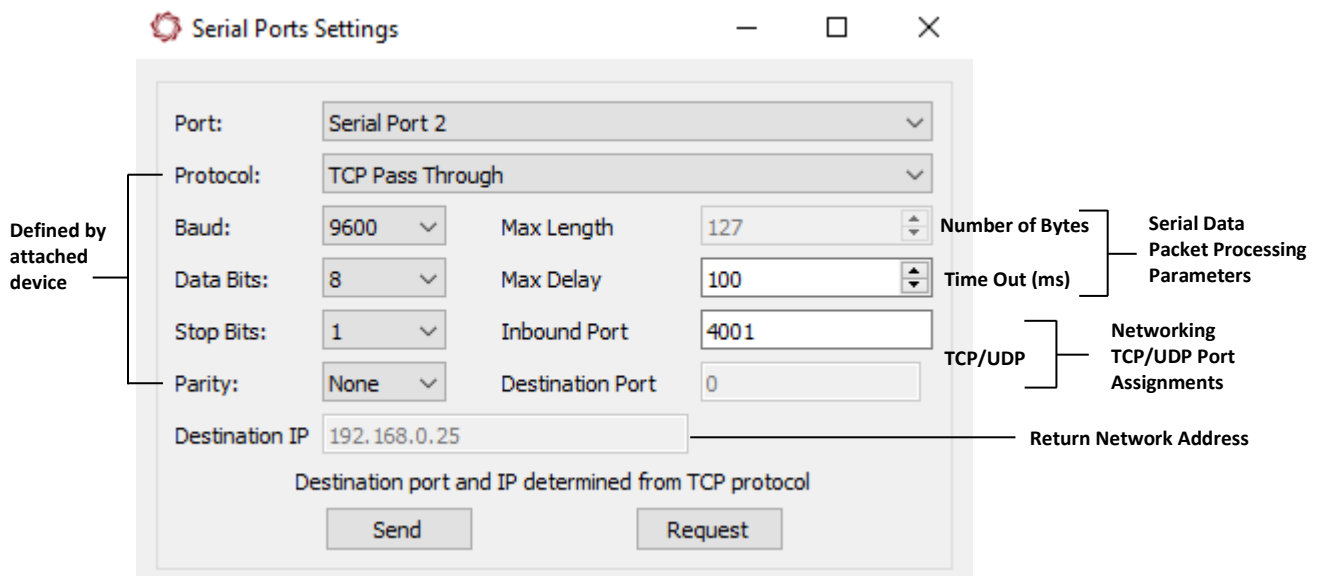


Figure 2: Serial Port Configuration Dialog




 *UDP AttNav Port only applies when the protocol type is set to Scan Eagle Aquarius packet parsing and SLA protocol (protocol type = 1). Any Aquarius data received on the serial port may be sent over Ethernet. It requires that the destination IP address and local outbound port number be non-zero. Any SightLine protocol will be handled locally and not passed through.*

Table 1: Summary of Field User Per Protocol

Type	Protocol	Baud Rate	Data Bits	Stop Bits	Parity	Max Delay	Max Packet Length*	Inbound Port	Destination IP	Destination IP Port	Destination Serial Port	attNavPort
0	SLA Protocol	YES	YES	YES	YES	NA	(A)	NA	NA	NA	NA	NA
1	Aquarius and SLA Protocol	YES	YES	YES	YES	YES	(B)	YES	YES	YES	NA	YES
2	SLA Protocol no telemetry	YES	YES	YES	YES	NA	(A)	NA	NA	NA	NA	NA
4	Port Not Used	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
5	TCP	YES	YES	YES	YES	YES	(C)	Yes	NA	NA	NA	NA
6	Raw	YES	YES	YES	YES	YES	(B)	YES	YES	YES	NA	NA
7	KLV	YES	YES	YES	YES	100ms	NO	NA	NA	NA	NA	NA
8	Serial-to-Serial	YES	YES	YES	YES	YES	(C)	NA	NA	YES	YES	NA
9	NMEA	YES	YES	YES	YES	YES	YES	NA	NA	NA	NA	NA
15	Ethernet port	NO	NO	NO	NO	NO	(A)	YES	YES	-	-	-

*Max Packet Length:

(A) No - internally 16KB is maximum

(B) Min of length and 16KB

(C) No - internally 2KB is maximum

3 Protocols

The SightLine hardware can be configured for command and control, various input protocols, or passthrough communication channels. Passthrough modes are useful to pass data to and from auxiliary equipment in the system.

The following protocol modes are available on the unit:

[SLA Protocol](#)

Communications protocol used by the SightLine Applications product line

[Aquarius and SLA Protocol](#)

Sightline protocol + special protocol simultaneously

[SLA Protocol no telemetry](#)

Ports using this protocol type will accept SightLine Protocol commands but will not generate any telemetry data.

[Port Not Used](#)

Disables the serial port so that it can be used by other onboard applications.

[TCP Passthrough](#)

Creates a TCP Socket for two-way communication

[Raw Passthrough](#)

Creates a UDP Socket for one-way communication

[KLV Passthrough](#)

Receives KLV metadata over serial ports or Ethernet and attaches that data to any MPEG2-TS streaming video

[Serial-to-Serial Passthrough](#)

Allows data to be sent between two serial ports

[NMEA](#)


Parses any inbound data as NMEA 0183 strings and converts them to Metadata



3.1 SLA Protocol - SightLine Command and Control

The default serial port settings are *57600*, *8*, *1*, and *None*. There is no hardware flow control. Only *Receive (RX)*, *Transmit (TX)*, and *Ground* are normally needed. These signals are from the SightLine hardware perspective, so *Receive* is the inbound line.

Each hardware platform is different. Check the appropriate Interface Control Document (ICD) to verify if the serial port is 3.3V TTL or RS-232C.

 *A NULL modem or DB-9 gender changer may be required to adapt the connection for the connected device.*

SightLine protocol uses the following terminology:

- **Command and Control:** Any request and response commands for configuring the system.
- **Telemetry:** Asynchronous output from the SightLine hardware providing data for tracking objects, steering a gimbal, landing an aircraft, etc.


Each of the serial ports can be used for command and control and telemetry. On the 1500-OEM, 3000-OEM, and 4000-OEM, *Serial Port 0* is set for command and control by default. The behavior can be overridden using the SLA protocol to select a different protocol (above).

Serial port communication is available at the same time as Ethernet command and control. When working with SightLine hardware, use only one method of communication for command and control at a time to avoid conflicts. Both Ethernet and serial communication methods can be used simultaneously if they are accomplishing different tasks.

For example:

- Ethernet is used to connect with SightLine hardware from a ground control station for controlling video streaming,
- another device is using serial communication to send KLV metadata,
- another serial port is streaming telemetry data for the tracking of a gimbal.

After three frames of acquisition, the system will send a **Version (0x40)** message to the client. This indicates the system is running and ready to receive commands.

 **IMPORTANT:** As new firmware is released, some command and control packets do change. Check the version packet upon receipt to verify the firmware matches the version of the command and control functions that were implemented.

By default, telemetry output is sent on every frame. Adjust the rate or disable telemetry output using the **Set Coordinate Reporting Mode (0x0B)** command.

Telemetry includes the frame-to-frame registration and confidence metric and primary target location. See the **Tracking Position (0x43)** message for more information.

All locations are in pixel coordinates. To use the coordinates in the control loop for gimbal steering, calibrate and translate them into position information.

When a serial port is configured for one of the passthrough modes in the next sections, telemetry and command and control is disabled for that serial port.

When telemetry output is enabled, all serial ports receive all telemetry messages. It is possible to isolate telemetry messages to one serial port using SLA Protocol No Telemetry instead.



3.2 Aquarius and SLA Protocol

Supports SLA protocol for command and control and passes through Aquarius commands to Ethernet IP address and destination. It also listens to ATTNV packets on the network (port 65100) and converts them to Aquarius packets for output on the Serial Port.

3.3 SLA Protocol No Telemetry

Same as SLA Protocol above, but can be used to eliminate asynchronous telemetry data from the system.

3.4 Port Not Used

Indicates that the serial port will not be used by the SightLine software but may be used by customer applications running on the ARM processors, or for debugging.

3.5 TCP Passthrough

The system can be used to create a TCP socket connection to a serial port. This allows bidirectional communication with external devices. Each serial port can be configured to accept a single client connection. These connections can be established concurrently. The example in the next section outlines configuring a 3000-OEM for passthrough for a connected Sony FCB EH-6300 camera.

3.5.1 Serial Port Mapping

The Sony interface board is connected to J3 on the 3000-OEM main board and serial 2 is mapped to it. For more details on the 3000-OEM board-to-board connectors, see the [ICD-3000-OEM](#) document.

Table 2: Serial Port Mapping and Camera Names

Connector	SLA Name (16/24bit)	SLA Name (8bit)
J3 (serial 2)	Camera 0 (Cam 0)	Camera 0 and Camera 1
J4 (serial 3)	Camera 2 (Cam 2)	Camera 2 and Camera 3

3.5.2 Defining TCP Passthrough Settings

1. Load the Panel Plus application and connect to the 3000-OEM board.
2. Click the *Comms* tab and enter the following settings in the *Passthrough Settings* dialog window (Figure 3):
 - *Port: Serial Port 2*
 - *Parity: None*
 - *Baud: 9600*
 - *Protocol: TCP Pass Through*
 - *Data Bits: 8*
 - *Max Delay: 100*
 - *Stop Bits: 1*
 - *Inbound Port: 4001*

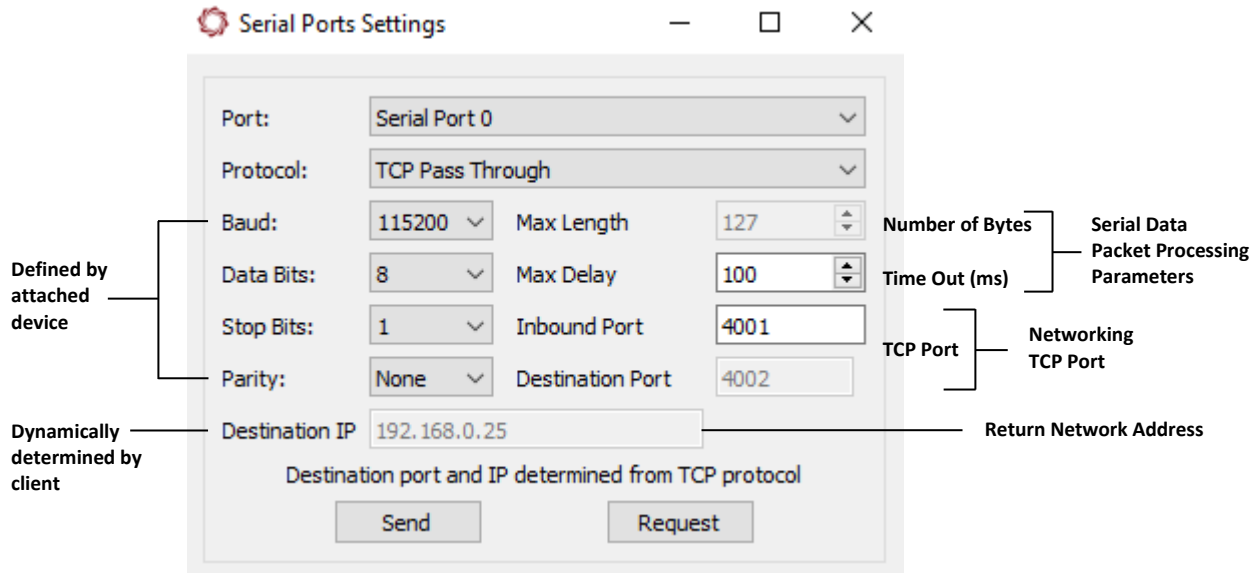


Figure 3: Serial Passthrough Settings Example

For TCP Passthrough, the client (PC) needs to know the server IP address and inbound port (SightLine hardware). The TCP protocol determines the rest of the information. The Destination IP and Destination Port of the client are implied / supplied by the connection socket.

3. After configuring the settings, changed fields will be highlighted in red. Click **Send**.
4. To save the configuration to the parameter file, from the Panel Plus main menu » *Parameters* » *Save to board*.

In 3.01.xx and earlier software versions, saving the Serial Port settings will prompt an additional dialog window. Some setting changes require the board to be restarted for the settings to take effect. In the *Apply New Settings* dialog window, select an option to save the port configuration

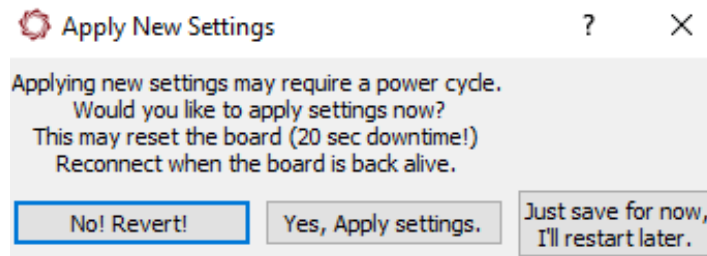


Figure 4: Apply New Settings Dialog - 3.01.xx and Earlier


Upon reboot the 3000-OEM will be ready to accept packets on port 4001 and will send that data to Serial Port 2. Any data received on Serial Port 2 will be sent to the client on the same TCP socket.

See the [Appendix](#) for steps on how to set up software to act as a virtual serial port on the host PC, where the camera GUI control application will run other serial control software.



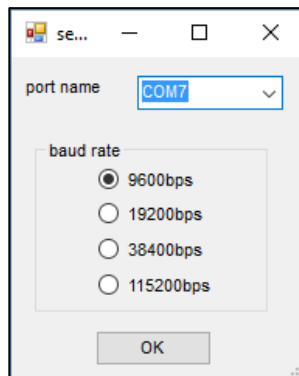
3.6 Example Camera Control GUI Connection

Many camera manufactures provide a User Interface (UI) that allows configuration of their camera. These UI's may assume a direct serial port connection to that camera. This section describes how to create a virtual serial port (COM PORT) that uses *TCP Passthrough* to connect to the camera connected to the SightLine hardware.

 *Contact a Sony representative for GUI software used in this example. Depending on the specific operating system of the host PC, Microsoft .NET framework V2.0 or higher may be required if not previously installed.*

3.6.1 Sony Control GUI

1. See the [Appendix](#) for steps on how to create a virtual com port for this example.
2. Start the Sony GUI camera control application.
3. For this example, select COM7 and a baud rate of 9600bps.



4. After connecting, the main application screen loads. The top right corner will display camera model, ROM version, baud rate, system, etc. To verify commands are sent through the serial passthrough configuration, change the zoom levels or white balance.

Model	FCB-EH6300
ROM Version	0200
Baud Rate	9600 bps
System	HD
Camera	0 (decimal)
VISCA No	1
<input type="checkbox"/> AutoUpdate On/Off	Wait
Refresh Once	
Command History	Clear Text



3.7 Raw Passthrough

Raw Passthrough creates one or more UDP sockets for communication with attached devices. The SightLine hardware listens on the inbound port for UDP packets from any IP address. The content of the packet is passed out the serial port. Data received on the serial port is sent as a UDP packet to the *Destination IP* address and *Destination Port*. The inbound data can be from a different IP address than the *Destination* address of the outbound data.

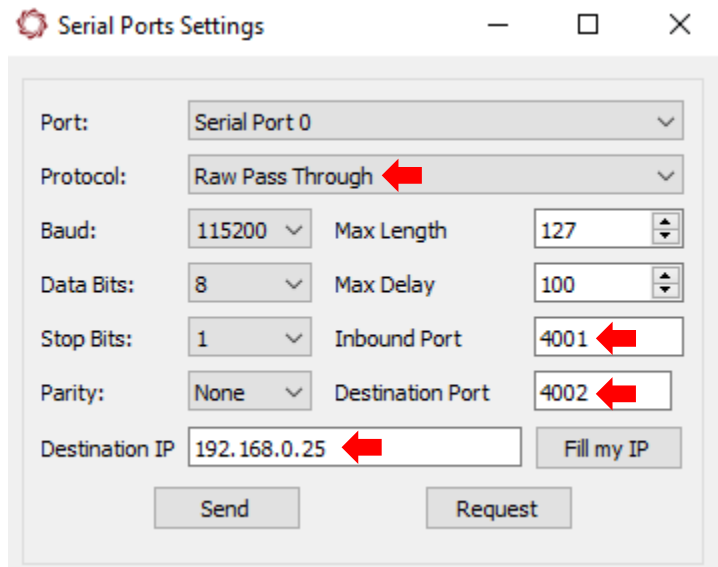


Figure 5: Serial Port Settings - Raw Passthrough

3.8 KLV Passthrough

KLV metadata received on the serial port or Ethernet port is injected into the MPEG2-TS stream along with compressed digital video. Formatted KLV metadata may be generated by an external processor. The KLV data is parsed for the universal key (16-bytes), the length field, and checksum, and reads the remaining data. There is no other parsing or validation of the KLV data. When passing raw KLV data to the serial port or network, no additional SightLine headers or checksums are necessary.

The system may be set in two possible configurations:

- (Default) - When using this mode, internally generated KLV metadata is skipped for the frame that received the data and then reengages for the next frame.
- KLV data can be parsed for individual fields and used to populate internal KLV values such as time, latitude and longitude, etc.

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



3.8.1 KLV Passthrough - Serial Port Example

The [Tera Term](#) emulator program is used in 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. After configuring the settings, changed fields will be highlighted in red. Click *Send*.
5. To save the configuration to the parameter file, from the Panel Plus main menu » *Parameters* » *Save to board*.

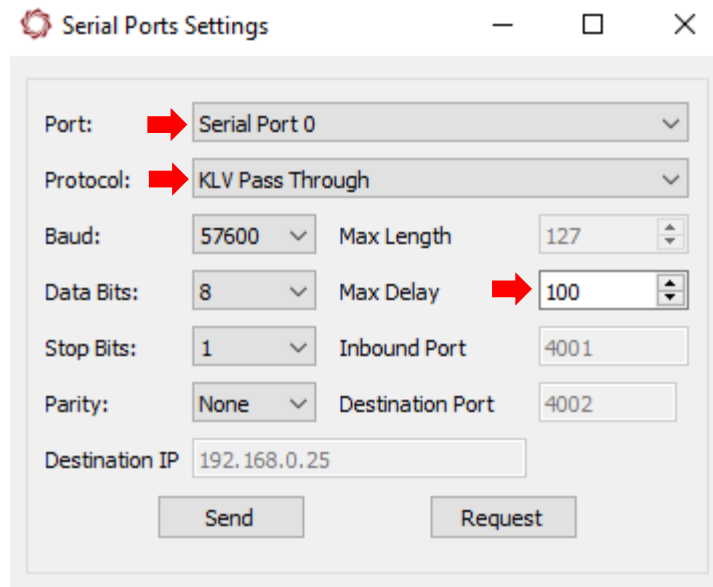


Figure 6: TCP Passthrough Setup

In 3.01.xx and earlier software versions, saving the Serial Port settings will prompt an additional dialog window. Some setting changes require the board to be restarted for the settings to take effect. In the Apply New Settings dialog window, select an option to save the port configuration.

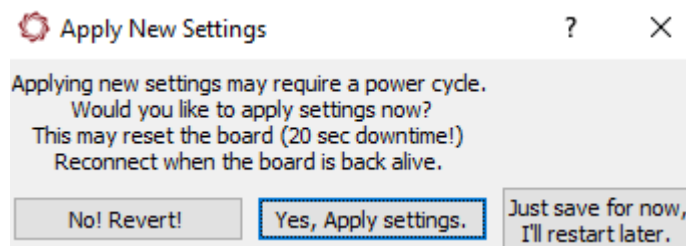


Figure 7: Apply New Settings Dialog - 3.01.xx and Earlier

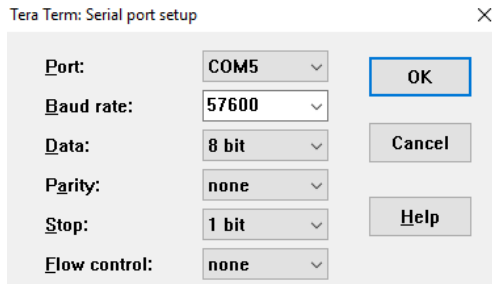
The system is now ready to receive a block of KLV data over the serial port and send it out on the MPEG2-TS stream.

With firmware release 2.25.xx, the KLV blob can be parsed for individual metadata fields. The KLV input sources must be explicitly enabled.

1. From the main menu in Panel Plus » *Configure* » *KLV Setup*.
2. Select *Tag Sources* tab.
3. For each tag to be extracted from the KLV blob, check the ID (TAG ID) box and select the KLV radio button.
4. When complete, click *Send*.



- Open Tera Term and connect to COM port. Verify serial port settings.



- Copy the KLV blob and paste it in the Tera Term console (see [Appendix](#)).
- Use the SLADeCode sample from the SightLine web site to validate that the KLV passthrough is working (see [Figure 10](#)).

3.8.2 KLV Passthrough - Ethernet Port Example

Receives binary KLV metadata over Ethernet and attaches that data to any MPEG2-TS streaming video.

- Configure Port to Ethernet (protocol type 15) and protocol for KLV Passthrough using the settings shown.
- Set *Max Delay* (default is 100 ms). This is the time the serial port waits to accumulate data.
- Choose the inbound UDP port. Make sure to choose an unused port.
- Ignore all other parameters.
- After configuring the settings, changed fields will be highlighted in red. Click *Send*.

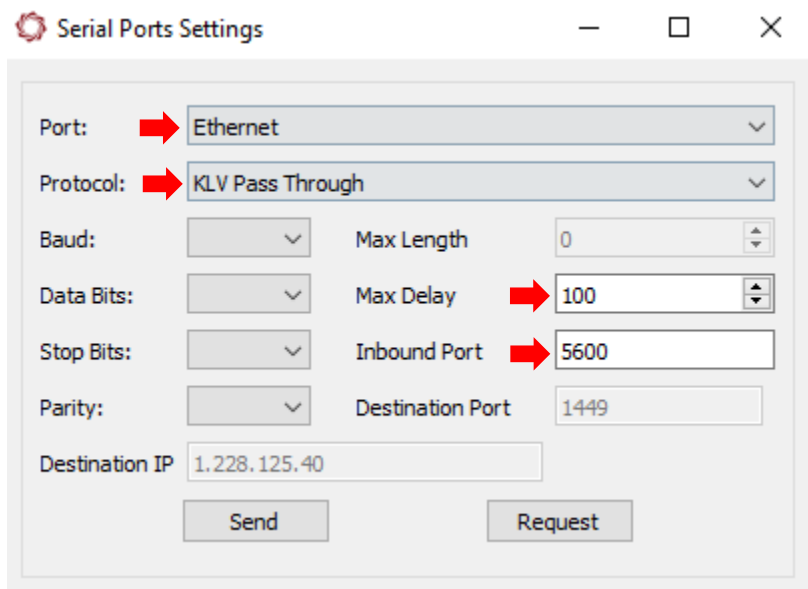


Figure 8: TCP Passthrough Setup

- To save the configuration to the parameter file, from the Panel Plus main menu » *Parameters* » *Save to board*.

In 3.01.xx and earlier software versions, saving the Serial Port settings will prompt an additional dialog window. Some setting changes require the board to be restarted for the settings to take effect. In the Apply New Settings dialog window, select an option to save the port configuration.

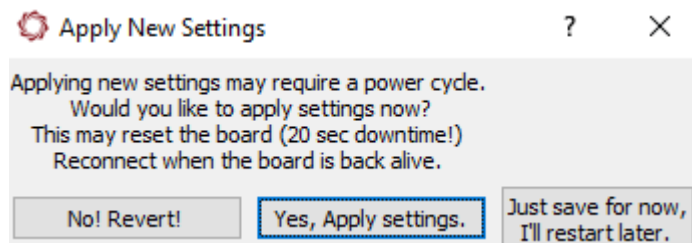


Figure 9: Apply New Settings Dialog - 3.01.xx and Earlier



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.

Send a properly structured block of KLV data using a program that can send data over UDP port. See Testing KLV Passthrough over Serial or Ethernet in the [Appendix](#).

Use the SLADecode sample from the SightLine web site to validate that the KLV passthrough is working (Figure 10), other 3rd party programs such as GV3.0, etc.

```

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 10: SLADecode Sample

3.9 Serial-to-Serial Passthrough

Serial-to-Serial sends any input from one serial port to another serial port specified in Destination port number. Maximum packet length, inbound port, destination IP and AttNav are ignored. By configuring two ports to output to each other, the two ports behave as if they are connected.

3.9.1 Serial Passthrough Example - 1500-OEM

In the following example, the 1500-OEM will be configured to pass data from Serial 0 to Serial 1. The configuration will then be tested using two serial ports on out PC, and two separate instances of Tera Term.

1. Hardware bench setup:
 - a. Attach the 1500-OEM board to the 1500-AB accessory board using the board connectors.
 - b. Attach the Molex 14-pin cable (SLA-CAB-1514) to the 1500-OEM connector and the 1500-AB connector.

Additional serial ports are available using the IO port header pins on the 1500-AB board.

2. Power the 1500-OEM serial ports:

Supply 3.3V to (VIOSEL) to use these additional serial ports using a jumper connector on the pins shown in Figure 11. Connecting VIOSEL to 3.3V enables Serial Port 1 and Serial Port 2.

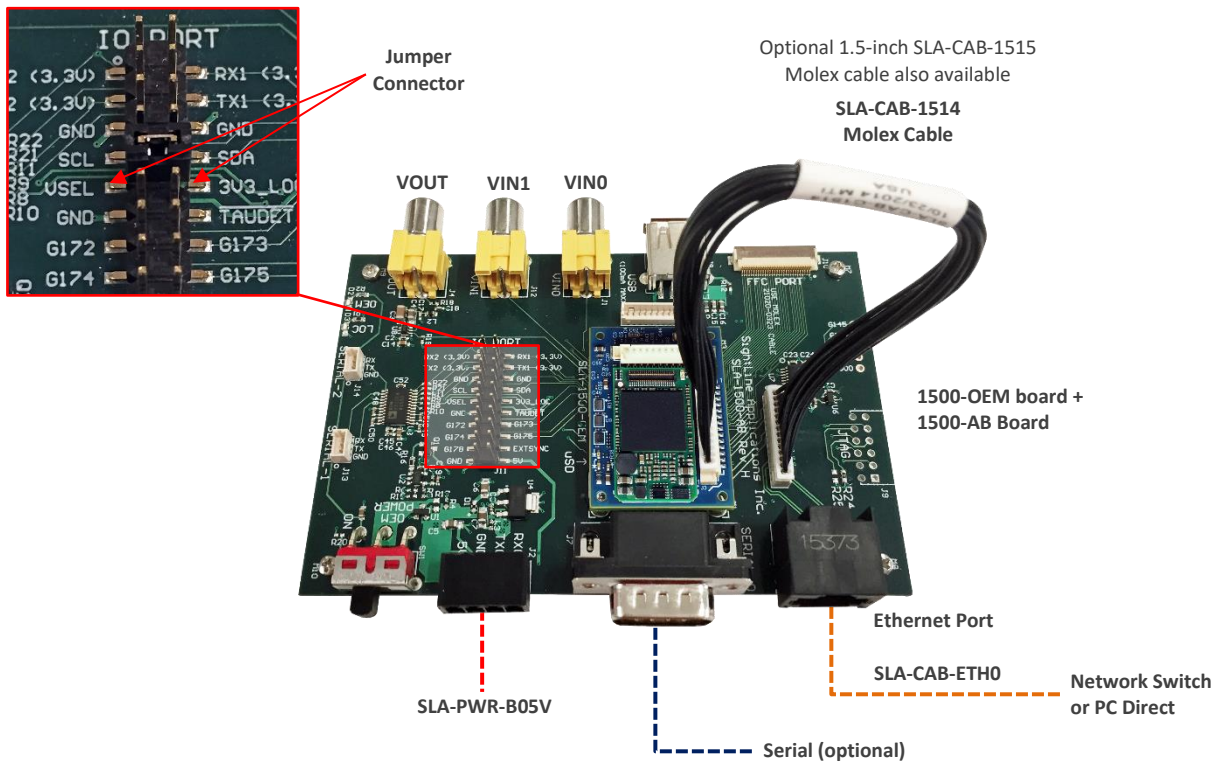
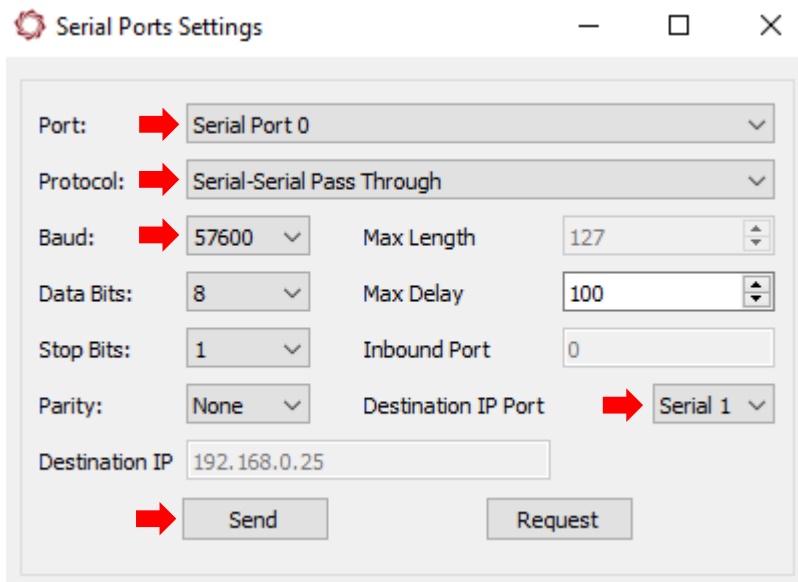


Figure 11: Supply 3.3V to VIOSEL

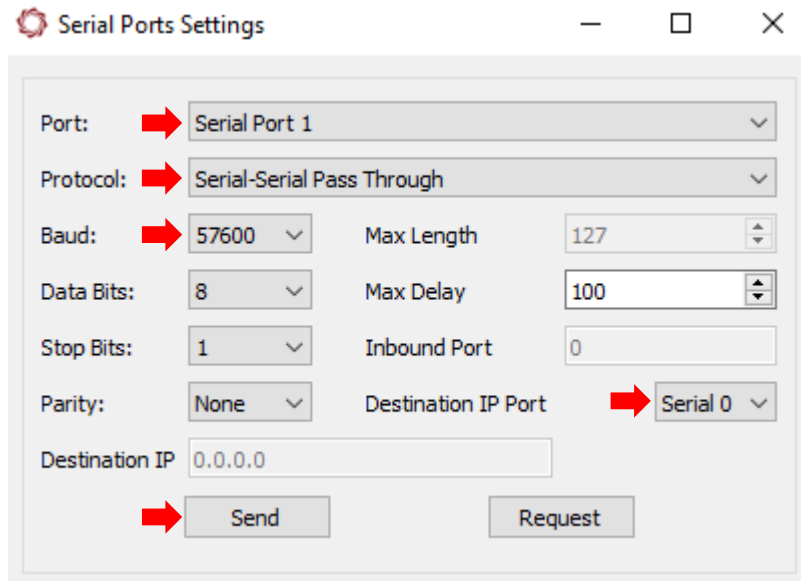
3. Panel Plus setup:

- a. In Panel-Plus, configure Serial Port 0 as shown.





b. Configure Serial Port 1 as shown.



4. After configuring the settings, changed fields will be highlighted in red. Click *Send*.
5. To save the configuration to the parameter file, from the Panel Plus main menu » *Parameters* » *Save to board*.

In 3.01.xx and earlier software versions, saving the Serial Port settings will prompt an additional dialog window. Some setting changes require the board to be restarted for the settings to take effect. In the Apply New Settings dialog window, select an option to save the port configuration.

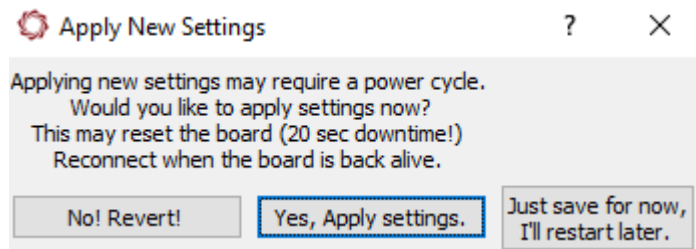
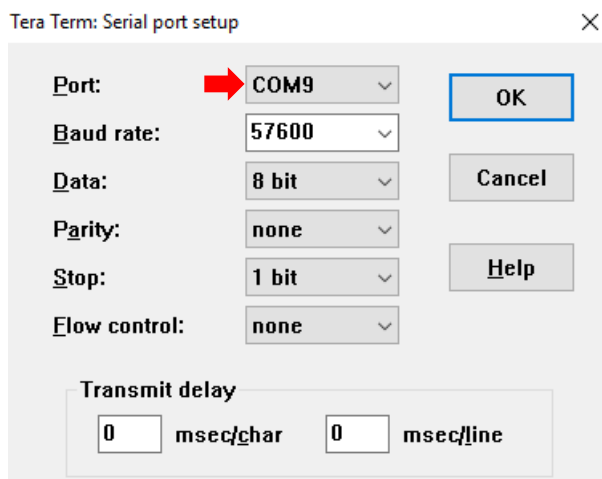


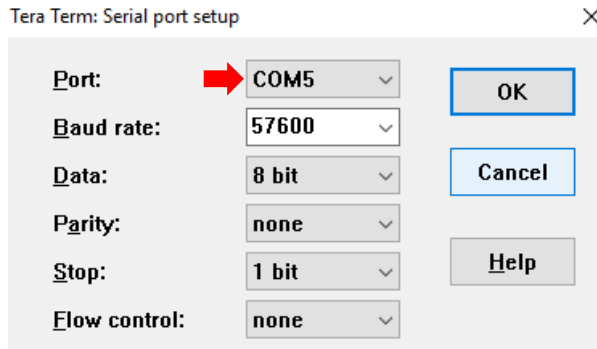
Figure 12: Apply New Settings Dialog - 3.01.xx and Earlier

6. Tera Term setup:
 - a. Use Tera Term to open serial port connections to a COM port on the PC. The port should be connected physically to SLA-1500-AB at Serial Port 0. The example below uses COM9 to Serial 0.





- b. Open another instance of Tera Term to open serial port connections to another COM port on the PC. This port should be connected physically to SLA-1500-AB at Serial Port 1. The example uses COM5 to Serial 1.



7. Testing serial-to-serial passthrough. The serial-to-serial passthrough test sends text from the serial port of the PC through the serial port of the 1500-OEM and back into another serial port of the PC.
 - a. In the COM9 Tera Term window, type any text, i.e., TEST. Text should display in COM5 of the Tera Term window.
 - b. In the COM5 Tera Term window, type any text. Text should display in COM9 of the Tera Term window.

3.10 NMEA

A serial port can be configured to receive NMEA 0183 compatible strings. The external device (GPS), which generates the strings, is connected to the specified serial port. The serial port is configured for the compatible baud rate and other settings to match the external device (all Ethernet settings are ignored and can be set to zeros). Strings are parsed and converted to appropriate Metadata values. All other data is ignored.

ⓘ IMPORTANT: KLV licensing is required for this feature to work. The licensing can be verified on the *Connect* tab of Panel Plus.

NMEA notes:

- NEMO mode listens for NMEA 0183 data over the serial port.
- Default settings: *4800, 8, 1, None* (other settings can be used).
- Very simple NMEA parsing of \$GPGGA and \$GPRMC.
- Serial Port 1 is used in the example below. Other serial ports may be used.
- SightLine hardware can be used in passthrough mode to allow configuration of a GPS.
- NMEA mode converts known fields into comparable MISB Metadata (below).
- The user can continue to set other Metadata fields using **Set Metadata Values (0x13)** commands or other metadata functions.
- Use the correct enable bits to avoid conflict with the fields set by NMEA.



3.10.1 Configure NMEA Input

The example assumes a GPS device is connected to Serial Port 1.

1. From the Panel Plus main menu *Configure » Serial Ports*.
2. Select Serial Port 1 and configure the baud rate and other settings to match the attached GPS.
3. After configuring the settings, changed fields will be highlighted in red. Click *Send*.
4. To save the configuration to the parameter file, from the Panel Plus main menu » *Parameters » Save to board*.

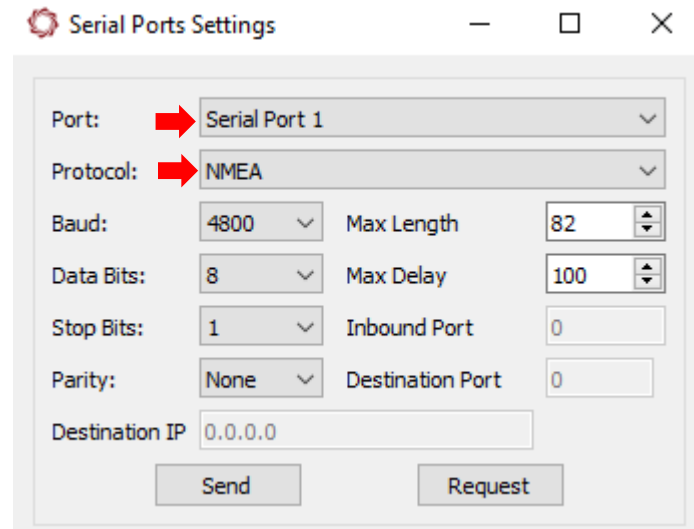


Figure 13: Configure Serial Ports

In 3.01.xx and earlier software versions, saving the Serial Port settings will prompt an additional dialog window. Some setting changes require the board to be restarted for the settings to take effect. In the Apply New Settings dialog window, select an option to save the port configuration.

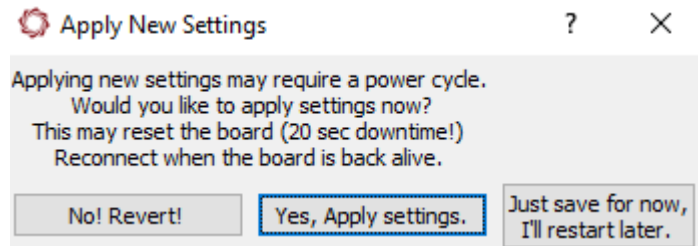


Figure 14: Apply New Settings Dialog - 3.01.xx and Earlier

3.10.2 Test NMEA Parsing

You can validate that the system is reading the NMEA strings and converting them to Metadata. The example assumes that video streaming is enabled. From the Panel Plus main menu *Configure » KLV Setup*. The Latitude and Longitude fields should be updating from the GPS device.

NMEA Parsing notes:

- See the Troubleshooting section below for instructions on how to see the raw NMEA data coming into the serial port.
- The SightLine hardware can be configured for TCP passthrough when setting up a GPS device. Once complete, the serial port for NMEA parsing can be configured.
- The Metadata fields shown in [Table 3](#) can be updated from the NMEA inputs. Use the Tag Source Selector (0x98) to set NMEA as the source for these MISB Tags.

Table 3: Current Metadata Data Values (0x13)

Description	Tag	NMEA
UTC time	2	GPGGA, GPRMC
Platform Heading Angle	5	PASHR, GPRMC
Sensor latitude	13	GPGGA, GPRMC
Sensor longitude	14	GPGGA, GPRMC
Sensor True Altitude	15	GPGGA
Platform Ground Speed	56	GPRMC



4 Max Length and Max Delay

The *Max Length (bytes)* and *Max Delay (ms)* act as a form of software flow control for serial port communication. *Max Length* indicates that the software should stop reading the serial port after a fixed number of bytes are received. The software then analyzes the bytes to see if a complete message has been received. The message is determined by the protocol selected in the Set Port Configuration (0x3e) command in the [IDD](#).

For example, a complete message can have a NMEA 0183 sentence and an SLA command and control packet, or others. The software will continue to internally cache the bytes received until a fully formed command can be determined. A fully formed command is one that complies with a known header, length, and checksum, or whatever rules apply to that protocol.

The *Max Delay* indicates how long in milliseconds the system will wait for the sender to complete transmission of characters before timing out. Below are two examples of how they might be used.

Example 1: Sender transmits the SLA command for **Get Version (0x00)** (5 bytes).

1. The *Max Length* is set to 18 bytes.
2. The 5ms timeout set by *Max Delay* forces the receiver to timeout, reads the serial port buffer, and then executes the command.

Example 2: Sender transmits a command that is 42 bytes long.

1. The receiver reads the first 18-bytes as set by *Max Length*.
2. The bytes are analyzed (does not make a fully formed command).
3. The receiver reads the next 18-bytes. It appends it to the first 18-bytes for a total of 36-bytes (is not a fully formed command).
4. The *Max Delay* times out, and the remaining 6-bytes are read. These 6-bytes are appended to the 36-bytes for a total of 42 bytes.
5. The bytes are analyzed. A complete command can now be executed.

Not every protocol uses Max Delay or Max Length. Most commands for serial port communication use these values.

5 Troubleshooting

There are many things that can go wrong with serial port communication. Issues such as incorrect voltage levels, grounding, crossing transmit and receive signals, bad cables, and incorrect settings can all be diagnosed with simple testing techniques. This section describes a troubleshooting test to validate that the SightLine hardware is functioning as expected.

ⓘ IMPORTANT: It is important to start with a known working configuration and then change one variable at a time until the root cause of the problem is found.

It may be necessary to either stop the VideoTrack1500 process or configure the serial port as port not used before beginning.



5.1 Serial 1 to Serial 2 Test

A jumper must be in place across VIOSEL and 3v3_LOC as shown in *Figure 15*. Connecting VIOSEL to 3.3V enables Serial Port 1 and Serial Port 2.

1. Using two terminal windows (both windows using SSH), connect Serial-1 to Serial-2 as shown in *Figure 15*.
2. Apply power to the 1500-OEM.

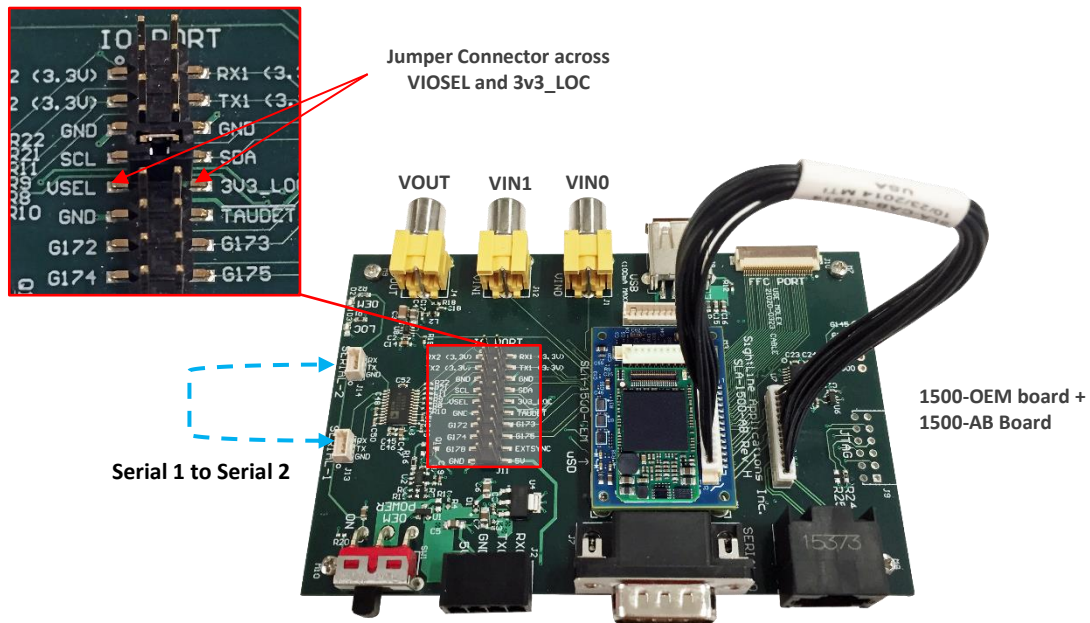


Figure 15: Serial 1-to-Serial 2 On-Board Test

3. Configure the serial ports for the same baud rate.

```
stty -F /dev/ttyO1 57600
```

```
stty -F /dev/ttyO2 57600
```

4. In one window, read from the serial port using *cat*.

```
cat /dev/ttyO2
```

5. In the other window, send some data using *echo*.

```
echo "hello" > /dev/ttyO1
```

6. Test confirms that serial 1 can transmit and serial 2 can receive.

This test can be repeated for serial port 0.



5.2 NMEA Output Test in Tera Term

The following method in Tera Term can be used to test NMEA output from the GPS device.

```
DM-37x# cat /dev/tty01
:A,4532.26164,N,12240.29287,W,7.00,,,,,*4C
$GPUTG,,,,,,M*33
$GPGGA,,4532.26164,N,12240.29287,W,7.00,,,,,*4C
$GPUTG,,,,,,M*33
$GPGGA,,4532.26164,N,12240.29287,W,7.00,,,,,*4C
$GPUTG,,,,,,M*33
```

Figure 16: NMEA Test in Tera Term

5.3 Additional Troubleshooting Tips

If the RX and TX lines are swapped use a NULL modem. Verify the baud rate, data bits, and other serial settings match for each serial port.

6 SightLine Command and Control

This section describes passthrough and other communication related commands.

Table 4: Commands

Communication Commands	Description
Configure Communication Port (0x3E)	Set up serial and Ethernet ports
Get Port Configuration (0x3F)	Request the current settings.
Current Port Configuration (0x53)	Reply message containing the settings for the current port.
Set Network Parameters (0x1C)	Configure network settings such as IP Address, etc.
Get Network Parameters (0x1D)	Request network settings such as IP Address, etc.
Current Network Parameters (0x49)	Reports network settings such as IP address, etc.
Command Passthrough (0x3D)	Method to passthrough data using SLA command and control packets.
Save Parameters (0x25)	
Reset (0x01)	

6.1 Additional Serial Port Configuration Notes

Some changes that are made to the serial port configuration require that the parameters be saved, and the hardware restarted. Reset can be done by cycling power to the board or by sending the soft reset command to the board.

Serial Passthrough (0x3E):

Byte 5 through 8 (serial port settings) require the settings to be saved and a reboot to take effect. Parameters 9 through 21 can be changed without having to reboot. The IP address and inbound / outbound ports can be changed at run time.

The serial port settings (baud rate, etc.) should only need to be set once and will stay fixed for the life of the design. However, the destination IP address of the remote device may not be known until much later, so changing this at run time is useful.




See SightLine hardware specific ICDs for more information on specific serial ports that are available for passthrough.

In some cases, a UDP protocol is preferred over TCP, specifically when lost packets are not critical to the operation of the system. The setup for UDP passthrough will be identical to the example above except for the TCP Passthrough protocol used. Because it is UDP-packet based, there is no guarantee against lost packets.

Another way to utilize the serial port is to send a specific command to the SightLine hardware using the **Command Passthrough (0x3D)** message, which includes the content the user wishes to emit on the serial port. This achieves a one-way communication only, as the data included as a payload within the command to the SightLine hardware is transmitted on the serial port, but there is no equivalent command to receive data and send it back.

When utilizing more than one serial port on the SightLine hardware, it is important to select a different set of IP ports for each serial port configured. An Ethernet connection is also serving the control of the SightLine hardware itself on the default ports 14001 and 14002. Do not select these numbers as IP ports. Digital video is sent to the host PC on default ports 15004 or 5004, which should also be avoided. A more complete list is available in the [IDD](#).

 *Firewalls or other system administration applications on the host PC might be preventing certain network ports from being used.*

 *When using Panel Plus, connect a scope or other device to the TX serial pins and test the setup of the serial passthrough by sending characters to the SLA board.*

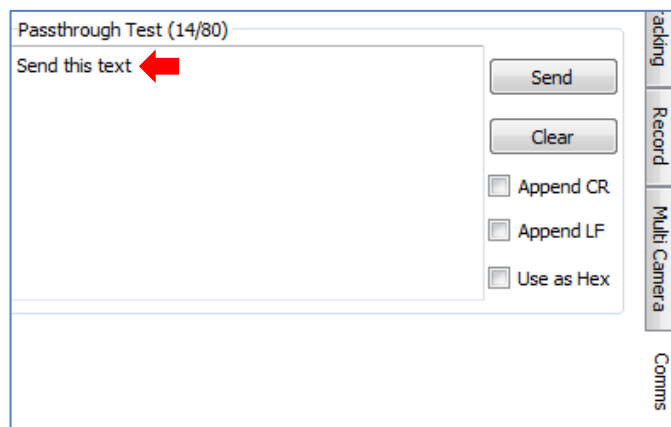


Figure 17: Passthrough Test

7 Questions and Additional Support

For questions and additional support, please contact [Technical Support](#). Additional support documentation and Engineering Application Notes (EANs) can be found on the [Documentation](#) page of the SightLine Applications website.



Appendix A - KLV Data

A sample block of KLV data is shown in [Table A1](#).

- Starts with a 16-byte universal key (060e2b....)
- 3-bytes of BER long form length encoding (...82006a)
- 106 (0x006a) bytes of data
- 2-byte checksum (0x2031)

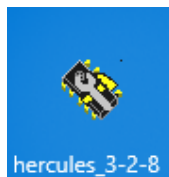
Table A1: Sample KLV block (hex)

06	0e	2b	34	02	0b	01	01	0e	01	03	01	01	00	00	00	82	00	6a	02	08	00	05	88	dd	3f	ff	f7	f8
41	01	09	05	02	01	bb	06	02	0c	86	07	02	ff	8d	08	01	00	0d	04	40	ff	da	d1	0e	04	a9	95	51
de	0f	02	0d	86	10	02	4f	ed	11	02	35	49	12	04	00	02	5f	e1	13	04	00	00	d2	ca	14	04	00	00
00	00	17	04	80	00	00	00	18	04	80	00	00	00	19	02	00	00	39	04	ff	ff	ff	ff	15	04	ff	ff	ff
ff	16	02	ff	ff	01	02	20	31																				

A1 Testing KLV Passthrough over Serial or Ethernet

This example uses the Hercules SETUP utility by the HW-group. The PC has a serial port on COM 9. The PC serial port is connected to target serial port 2.

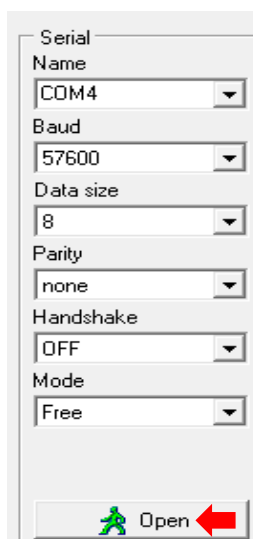
1. Open the Hercules software.



2. KLV passthrough over serial:
 - a. Click the *Serial* tab.

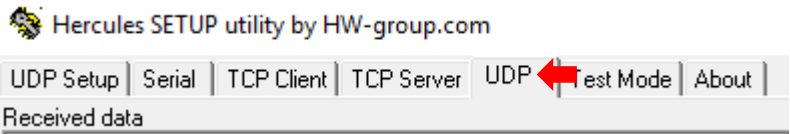


- b. Configure the serial port settings as shown and click *Open*.

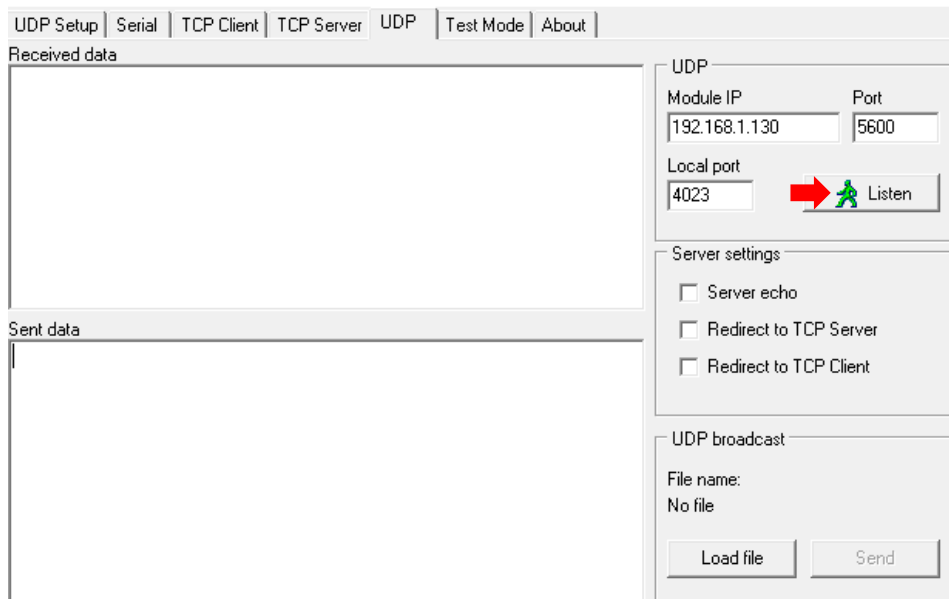




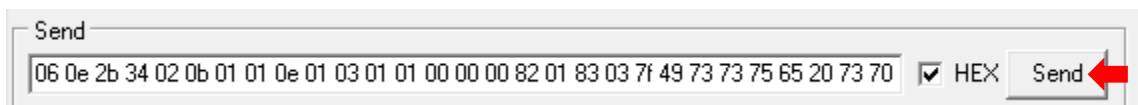
3. KLV passthrough over Ethernet:
 - a. Click the *UDP* tab.



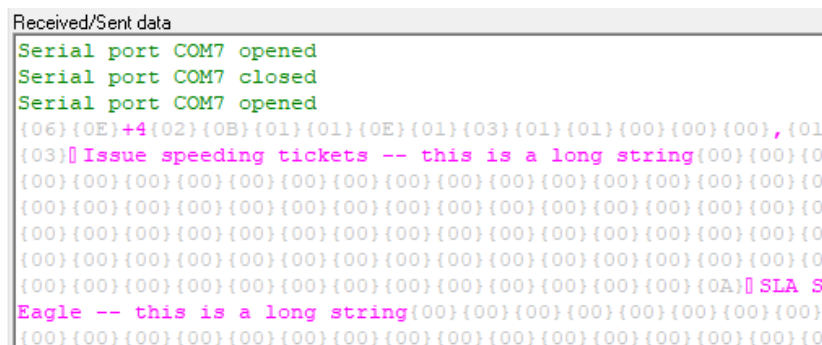
- b. Choose the same inbound port used in configuring KLV passthrough over ethernet using panel plus. In this example it is 5600.
 - c. Click *Listen*.



4. Copy and paste the sample KLV blob data above into the *Send* field. Click *Send*.




5. Verify the output data was sent. The KLV data should be received by the target and sent in the MPEG-2 TS stream.



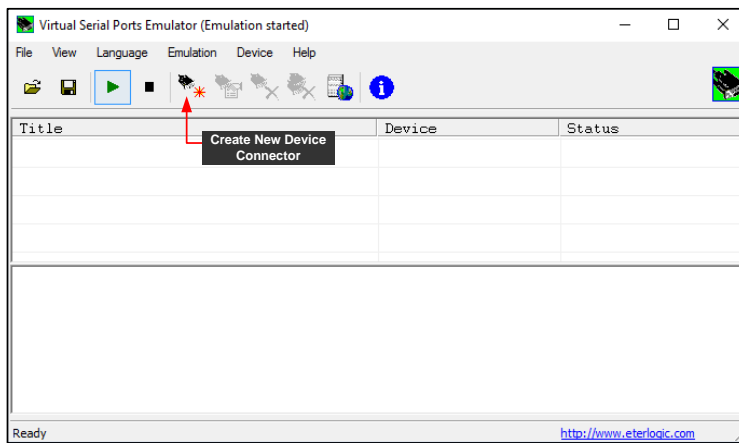


Appendix B - Virtual Serial Port Emulator Software

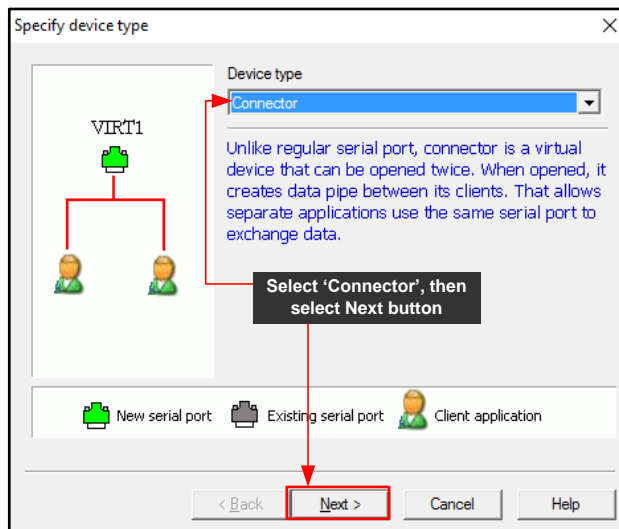
This example shows how to create two virtual devices for connecting to the camera serially over the network with the camera GUI software using the Virtual Serial Port Emulator from [Eterlogic](http://www.eterlogic.com).

 Review the physical serial ports on the host PC. Create virtual serial ports that are not already assigned or in use on the current system. Since these numbers are typically COM1 to COM4, it is important to assign higher numbers. For this example, COM7 is used.

1. Open the Virtual Serial Port Emulator application.
2. To create a connector, select the *Create Device* icon from the main menu.

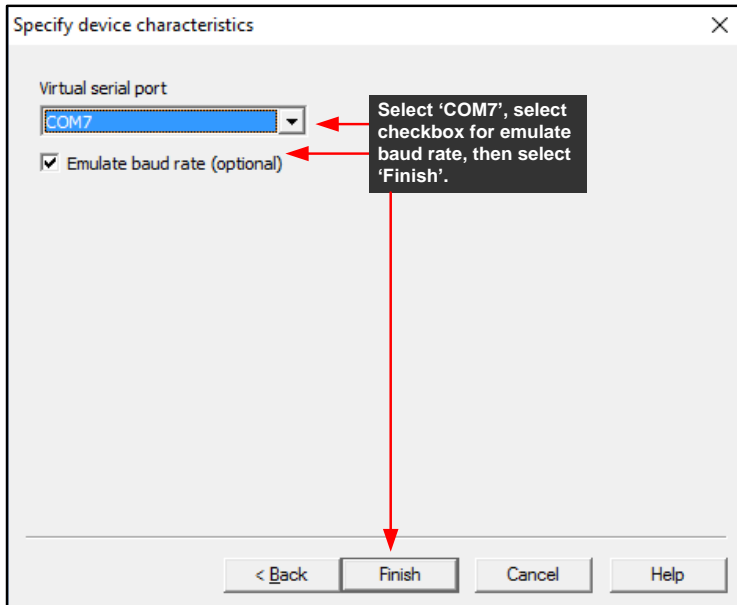


3. Select the *Connector* as the device type, and then select *Next*.

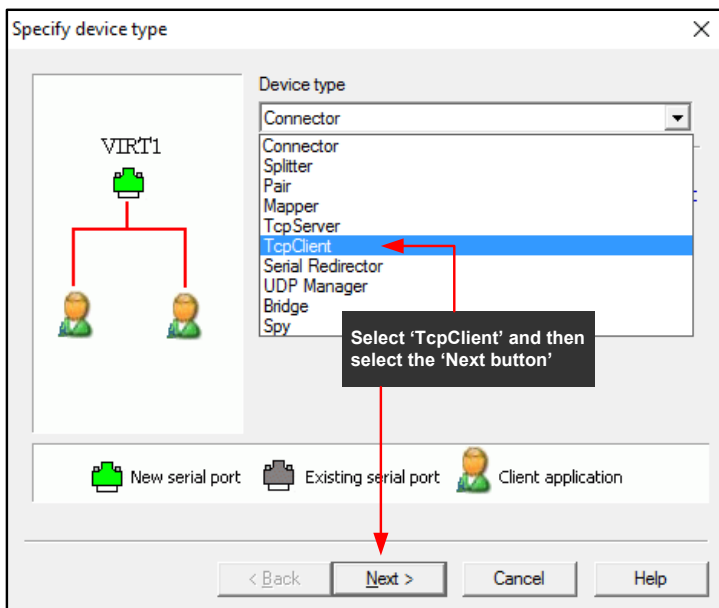




4. Select *COM7* from the dropdown menu. Select the checkbox for *Emulate baud rate*, and then select *Finish*.



5. Create another device and select *TcpClient* from the device type dropdown and select *Next*.



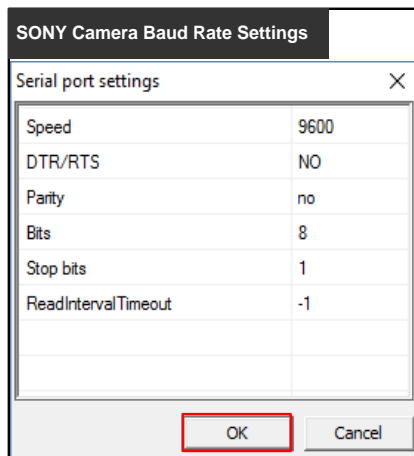


6. Type in the following device elements. For this example, use the values configured in the [Serial Passthrough Example 1500-OEM](#).

- Remote TCP host (SightLine board IP): 192.168.0.117
- Remote TCP port (SightLine board defined port): 4001
- Source serial port is the VSPE connector that was defined for this configuration: COM7

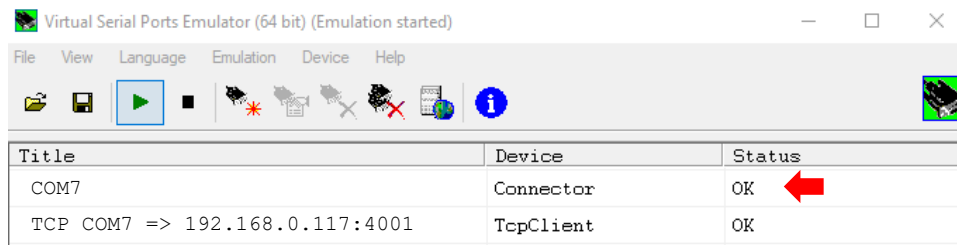
DTR/RTS - leave checked (default).

7. Select the *Settings* button under the selected port. Enter the camera baud rate settings, and then select *OK*.



8. Click *Finish* from the device characteristics settings window to start the emulation. This completes the virtual connection setup and allows connection to the camera over the network with the camera GUI.

9. In the Virtual Serial port Emulator dialog window verify the virtual port status.





Appendix C - Creating Custom Symlinks to Devices - 4000-OEM

Version 3.01.xx software includes the udev device manager for naming and creating custom symlinks to devices that use virtual serial ports. For more information on writing udev rules see the following [article](#).

This process requires [Tera Term](#) (recommended) or another SSH client to connect to SightLine OEM hardware.

Example:

The FLIR Boson 640 with a USB interface board presents a CDC ACM virtual serial port to the 4000-OEM system when connected. Using udev rules, this port can be renamed to *ttySL0* and then configured to perform serial passthrough.

1. Use Tera Term to establish an SSH session to the OEM hardware.
2. Find a port to use for serial passthrough. From the terminal window enter:

```
#slroot @ ~
SD> ls /dev | grep ttyACM
ttyACM0
```

3. Create a udev rule to make a symlink for the *ttyACM0* port. From the terminal window enter:

```
SD> echo `KERNEL=="ttyACM*", SYMLINK+="ttySL0"` >
/etc/udev/rules.d/new.rules
```

4. Power cycle the board for the new rule to take effect.
5. Verify the new rule works. From the terminal window enter:

```
SD> ls /dev | grep ttySL
ttySL0
```

6. Configure TCP Passthrough to */dev/ttySL0*, following the steps in [TCP Passthrough](#) and [Appendix B](#).
7. Connect to the camera with the FLIR Boson control application.