Velodyne Lidar®

Alpha Prime User Manual

63-9483 Rev. 4

DRAFT

Trademarks

Velodyne™, HDL-32E™, HDL-64E™, VLP-16™, VLP-32™, Puck 32MR™, Puck LITE™, Puck Hi-Res™, Ultra Puck™, VLS-128™, Alpha Prime™, Alpha Puck™, Velarray™, and Velabit™ are trademarks of Velodyne Lidar, Inc. All other trademarks, service marks, and company names in this document or website are properties of their respective owners.

Disclaimer of Liability

The information contained in this document is subject to change without notice. Velodyne Lidar, Inc. shall not be liable for errors contained herein or for incidental or consequential damage in connection with the furnishing, performance, or use of this document or equipment supplied with it.

The materials and information contained herein are being provided by Velodyne Lidar, Inc. to its Customer solely for Customer's use for its internal business purposes. Velodyne Lidar, Inc. retains all right, title, interest in and copyrights to the materials and information herein. The materials and information contained herein constitute confidential information of Velodyne Lidar, Inc. and Customer shall not disclose or transfer any of these materials or information to any third party.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or stored in a database or retrieval system for any purpose without the express written permission of Velodyne Lidar, Inc., which reserves the right to make changes to this document at any time without notice and assumes no responsibility for its use. This document contains the most current information available at the time of publication. When new or revised information becomes available, this entire document will be updated and distributed to all registered users.

Limited Warranty

Except as specified below, products sold hereunder shall be free from defects in materials and workmanship and shall conform to Velodyne's published specifications or other specifications accepted in writing by Velodyne for a period of one (1) year from the date of shipment of the products. The foregoing warranty does not apply to any Garmin products, other products not manufactured by Velodyne or products that have been subject to misuse, neglect, or accident, or have been opened, dissembled, or altered in any way. Velodyne shall make the final determination as to whether its products are defective. Velodyne 's sole obligation for products failing to comply with this warranty shall be, at its option, to either repair, replace or issue credit for the nonconforming product where, within fourteen (14) days of the expiration of the warranty period, (i) Velodyne has received written notice of any nonconformity; (ii) after Velodyne's written authorization, Buyer has returned the nonconforming product to Velodyne at Buyer's expense; and (iii) Velodyne has determined that the product is nonconforming and that such nonconformity is not the result of improper installation, repair or other misuse. Velodyne will pay for return shipping for all equipment repaired or replaced under warranty and Buyer will pay all duties or taxes, if any, on all equipment repaired or replaced under warranty. THE FOREGOING WARRANTY AND REMEDIES ARE EXCLUSIVE AND MADE EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED, IMPLIED OR OTHERWISE, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. VELODYNE DOES NOT ASSUME OR AUTHORIZE ANY OTHER PERSON TO ASSUME FOR IT ANY OTHER LIABILITY IN CONNECTION WITH ITS PRODUCTS. This warranty is non-transferable.

Velodyne Lidar, Inc.
5521 Hellyer Ave
San Jose, CA 95138
Phone +1 408-465-2800

Release Notes

Sensor	Firmware	Release Date	Release Notes
Alpha Prime	5.2.3.0	2021-03-12	• ADDED: APF (Advanced Packet Format) support. • IMPROVED: 300m support (APF only). Repetition rate is reduced slightly (5%). • FIXED: Future upgrades will not require Factory re-calibration. • IMPROVED: Web server has better tolerance of bursty and high rate network traffic. • ADDED: PTP support. • ADDED: Selectable PTP or GPS time reference. • FIXED: 1-second jump back of SEC_OF_HOUR Timestamp. • FIXED: NMEA parser handling of empty fields. • FIXED: Occasional lack of image on startup. * NOTE: 5.2.3 firmware is not field upgradeable. Sensors with earlier versions must be RMA'd to be upgraded at the factory. • Improved characterization of GPS and PPS signals (see Chapter 7). • ADDED: Section on Multi-Sensor Support to VeloView appendix.

Table of Contents

List of Tables

List of Figures

List of Equations

Chapter 1 • About This Manual

1.1 Manual Scope

This manual provides descriptions and procedures supporting the installation, verification, operation, and diagnostic evaluation of the Alpha Prime sensor.

Throughout this manual, "VLS-128" and "Alpha Prime" are used interchangeably. The former is the product model, while the latter is the given name for marketing purposes.

1.2 Prerequisite Knowledge

This manual is written with the premise that the user has some basic engineering experience and general understanding of lidar technology. In addition, some familiarity with the configuration and operation of networking applications and equipment is recommended.

It is recommended that prior to installation or other procedures covered in this manual, the user fully reads and comprehends all information within this manual.

1.3 Audience

The user mentioned occasionally in this document is typically an engineer tasked with sensor integration for a project, a tech tasked with sensor upkeep, or data scientist looking to understand sensor output data.

1.4 Document Conventions

This document uses the following typographic conventions:

Table 1-1 Document Conventions

Convention	Description
Bold	Indicates text on a window, other than the window title, including menus, menu options, buttons, fields, and labels. Example: Click **OK**.
Italic	Indicates a variable, which is a placeholder for actual text provided by the user or system. Example: copy *source-file target-file* Note: Angled brackets (< >) are also used to indicate variables.
`screen/code`	Indicates text that is displayed on screen or entered by the user. Example: `# pairdisplay -g oradb`
[ ] square brackets	Indicates optional values. Example: [a | b] indicates that you can choose a, b, or nothing.
{ } braces	Indicates required or expected values. Example: { a | b } indicates that you must choose either a or b.
| vertical bar	Indicates that you have a choice between two or more options or arguments. Examples: [a | b] indicates that you can choose a, b, or nothing. { a | b } indicates that you must choose either a or b.

Note: Notes such as this indicate important information. They call attention to an operating procedure or practice which may enhance user interaction with the product. Notes may also be used to prevent information loss or product damage.

1.5 Acronyms

APF - Advanced Packet Format.

CRC - Cyclic Redundancy Check. CRC values are used to help provide data integrity.

MTU - Maximum Transmission Unit. The size of the largest packet that a network protocol can transmit.

INS - Inertial Navigation System.

PTP - Precision Time Protocol.

GM - PTP Grandmaster Clock.

FOV - Field Of View.

MLA - Multi-Laser Array, a proprietary component of Velodyne lidar directional sensors.

NIC - Network Information Card. Typically, this is the Ethernet card in a computer.

SNR - Signal to Noise Ratio.

TOH - Top Of the Hour.

TSN - Time-Sensitive Network.

Chapter 2 • Alpha Prime Overview

This chapter provides basic information on the sensor's hardware and software components.

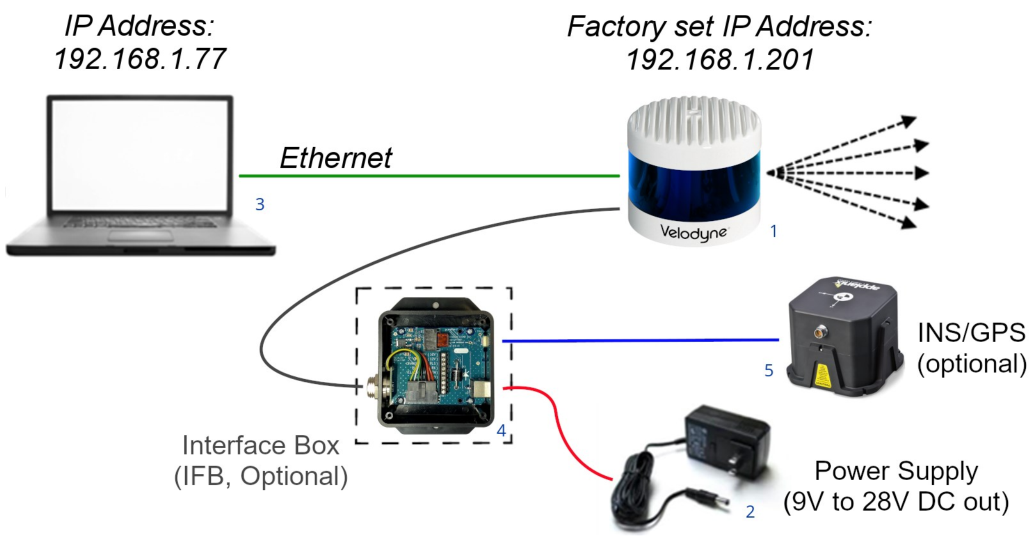
2.1 Product Overview

The Alpha Prime sensor uses 128 infra-red (IR) lasers paired with IR detectors to measure distances to objects. The device is mounted securely within a compact, weather-resistant housing. The assembly of laser/detector pairs spins rapidly within its fixed housing to scan the surrounding environment, firing each laser over 18,000 times per second, providing, in real-time, a rich set of 3D point data.

Advanced digital signal processing and waveform analysis provide highly accurate long-range sensing, as well as calibrated reflectivity data, enabling easy detection of retro-reflectors like street-signs, license plates, and lane markings.

Sensor data is made available to your application via Ethernet in real-time in the form of UDP data packets.

Figure 2-1 Example 3D Sensing System



.

Table 2-1 3D Sensing System Components

Item	Description
1	Alpha Prime Sensor
2	AC/DC Power Adapter or DC Power Supply for the sensor
3	Desktop/Laptop/Embedded Computer
4	Velodyne Interface Box (optional)
5	INS/GPS (optional)

Note: Optional - not included unless ordered: Garmin 18x LVC GPS Receiver (Velodyne P/N: **80-GPS18LVC**). When purchased through Velodyne Lidar, it comes pre-configured for use with any Velodyne sensor, making it a great reference GPS receiver. It plugs directly into the Interface Box's GPS Port. This device, when purchased anywhere else, does not work with our sensors out of the box, and is difficult to reprogram. *Factory Technical Support on page 108* if you are experiencing this problem.

2.2 Product Models

Sensor models can be compared at https://velodynelidar.com/product-comparison/.

To obtain a quote or make product or commercial inquiries, use the **Contact Sales** form at https://velodynelidar.com/contact-us/.

Data sheets for available models can be obtained under this page: https://velodynelidar.com/products/. Visit the sensor's product sub-page, click the *Resources* link, then download the product data sheet. Other assets may be available there. If you don't see something you're looking for, contact Sales. To request technical assets, however, use the **Contact Technical Support** form at https://velodynelidar.com/contact-us/.

2.3 Time of Flight

Velodyne Lidar sensors use time-of-flight (ToF) methodology.

When each IR laser pulse is emitted, its time-of-firing and direction are registered. As the laser pulse travels through air, its cross-section (aka beam spot) slowly, gradually increases its area over time due to beam divergence, until it hits a reflective surface which returns some of the energy. A small portion of that energy is received by the paired IR detector, which registers the time-of-acquisition and power received. The total flight time is halved then multiplied by the speed of light in air to determine the distance traveled from sensor to the reflective target.

2.4 Data Interpretation Requirements

- Desktop, Laptop or Embedded computer
- Advanced geo-referencing software application
  - GPS-Based
  - SLAM-Based
  - User Built
  - Purchased from System Integrator

For more software details, see *Converting PCAP Files to Point Cloud Formats on page 83*.

Note: Click the following link to view a list of Velodyne system integrators who can sell you imaging software or a complete system: https://velodynelidar.com/automated-with-velodyne/.

Chapter 3 • Safety Precautions

This chapter provides information necessary for the safe operation of Velodyne Lidar sensors.

Observe the following general safety precautions during all lidar sensor phases of operation. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of intended sensor usage and may impair the protection provided by the equipment. Velodyne Lidar, Inc. assumes no liability for failure to comply with these requirements.

3.1 Warning and Caution Definitions

3.1.1 Caution Hazard Alerts

⚠CAUTION

CAUTION indicates a potentially hazardous situation which may result in minor or moderate injury. It may also be used to alert against unsafe practices. The icon shown in the left column displays the specific concern; in this case, a hot surface.

3.2 Safety Overview

3.2.1 Electrical Safety

IMPORTANT: Read all installations instructions before powering up the sensor.

Note: The Alpha Prime sensor is not field serviceable. For servicing and repair, the equipment must be completely shut off, removed, packaged carefully, and shipped back to the manufacturer's facility with a completed RMA Form. See *Service and Maintenance on page 107* for details.

3.2.2 Mechanical Safety

⚠CAUTION

The Alpha Prime sensor contains a rapidly spinning assembly. Do not operate the Alpha Prime sensor without its cover firmly installed. The sensor does not contain user serviceable parts. It should not be opened in the field.

3.2.3 Laser Safety

This device complies with FDA performance standards for laser products except for deviations pursuant to Laser Notice No. 50, dated June 24, 2007.

Figure 3-1 Class 1 Laser

Note: The Alpha Prime sensor is a CLASS 1 LASER PRODUCT. The product fulfills the requirements of IEC 60825-1:2014 (Safety of Laser Products).

There are no controls or adjustments on the sensor itself that are user accessible.

Chapter 4 • Unboxing & Verification

This chapter provides the procedure to test and verify that your sensor is operating properly. Do this to check out a new sensor before permanently mounting it somewhere.

4.1 What's in the Box?

A standard Velodyne Alpha Prime sensor comes packaged in its own cardboard box, surrounded by foam padding.

Ensure all the components are present:

- Alpha Prime sensor in a protective shroud.
- Velodyne USB memory stick, containing:
  - User Manual
  - VeloView application installers for PC, Mac, and linux
  - Sensor sample data (i.e. pcap files)
  - Miscellaneous documents

4.1.1 Bulkhead Connectors

The Alpha Prime has two bulkhead connectors on the bottom of the sensor, enabling cabling to be hidden from public view while shielding them and the connectors from the elements. Placement of the connectors can be found in the *VLS-128 Mechanical Drawing on page 128*.

4.1.1.1 Ethernet Bulkhead Connector

Molex P/N 120341-0150 (datasheet)

4.1.1.2 Power+GPS Bulkhead Connector

Molex P/N 120244-0002 (datasheet)

4.1.2 Accessories

Interface boxes from other sensor types (e.g. VLP-32C) are not compatible with VLS-128 sensors. They are not interchangeable. In particular, note the different ratings for the fuses.

Optional components include:

- 2 m Gigabit Ethernet cable: CAT6A, Molex-to-RJ45, IP67, 8 conductor, 26 AWG, braided shielding
- 10 m Gigabit Ethernet cable: CAT6A, Molex-to-RJ45, IP67, 8 conductor, 26 AWG, braided shielding
- 2 m GPS+Power cable, Molex-to-Molex, 8 conductor (4 power, 4 signal), braided shielding, 22 AWG power, 26 AWG GPS
- 10 m GPS+Power cable, Molex-to-Molex, 8 conductor (4 power, 4 signal), braided shielding, 22 AWG power, 26 AWG GPS
- Interface Box with 7.5 A fuse and 3 m Molex power cable
- Standalone Interface Box with 7.5 A fuse and mating Molex connector

Each Interface box includes a 1.8 m AC power adapter.

Accessories are available for purchase either directly from Velodyne Sales or from any of our distributors.

4.1.3 Variants

Variants of the sensor exist, particularly with other connector types and/or cable lengths. Your sensor (or the type you are interested in) may vary from the standard configuration above. Contact Velodyne Sales for details.

4.2 Verification Procedure

The purpose of this procedure is to verify the sensor's basic functionality and get you started on your way to processing sensor data in (or from) the field. It involves one computer and one sensor in isolation at a workbench or desk. You'll need AC power. You won't need a GPS receiver.

Note: Due to the large volume of data produced by the sensor when scanning, users are cautioned against connecting it to a corporate network.

---

1. Unpack the sensor and its accessories, and place them on a workbench or desk. Ensure the sensor is upright with clear space around it.
2. Create a simple network setup with a test computer and the sensor in isolation. Follow the procedure in *Network Setup in Isolation below.*
3. Use the sensor's Web Interface to perform basic sensor configuration. Follow the procedure in *Access Sensor's Web Interface on page 29*.
4. Use VeloView (or other visualization software of your choice) to view data streaming from your sensor. Follow the procedure in *Visualize Live Sensor Data with VeloView on page 31*.

When finished, the sensor should be ready for more complicated usage scenarios.

4.2.1 Network Setup in Isolation

Your sensor's IP address comes from the factory set to its default value, **192.168.1.201**. This procedure prepares a computer to communicate directly with the sensor at that address.

Note: If using the computer's main Ethernet port, disconnect it from whatever network it's on. If using a secondary Ethernet interface, the primary network cannot be a 192.168.1 network. If it is, use the primary Ethernet interface instead.

---

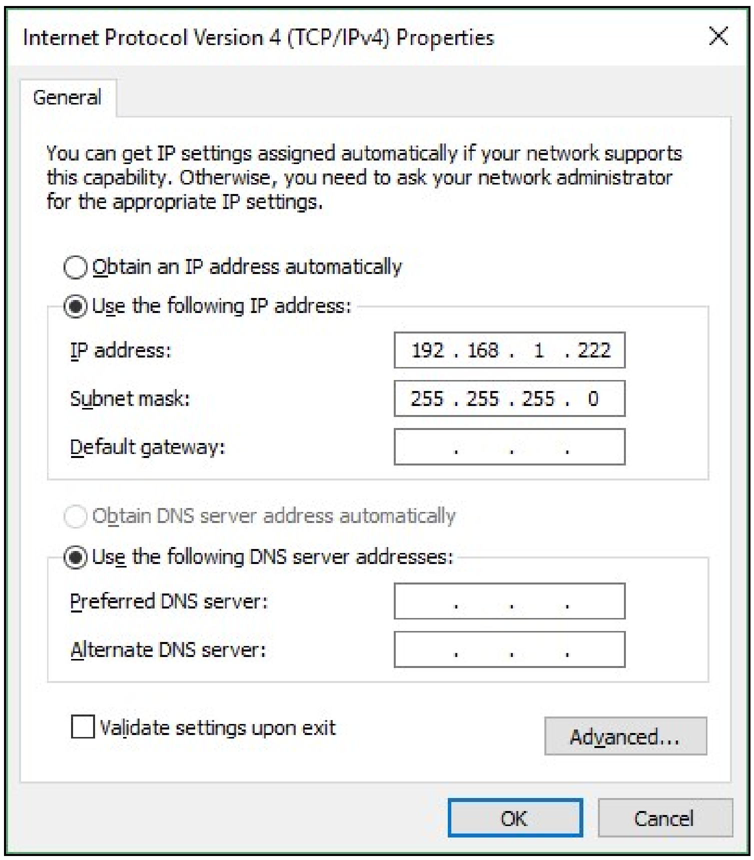
1. Open the computer's Network Connections page.
2. Open the applicable Ethernet adapter and make sure the interface is enabled.
3. Open Properties > **Internet Protocol Version 4 (TCP/IPv4)** (*Figure 4-1 below*).
4. Select the **Use the following IP address:** function.
5. Make up an IP address for the Ethernet port and enter it: 192.168.1.XXX.

   "XXX" may be any integer from 2 to 254 except 201.
6. Enter the subnet mask: 255.255.255.0.

   When using a Windows-based computer, you can press the TAB key and the subnet mask field will automatically populate with the default mask for the network class indicated by the IP address entered; which is 255.255.255.0 in this case.

   In some situations, notably on **Windows 11**, you may be asked instead to enter the *Subnet prefix length.* This is the count of '1' bits in the subnet mask. For a Class C network such as 192.168.1, the subnet mask length is 24; meaning, 24 bits in 255.255.255.0 are '1'. The subnet mask length for a Class A network is 8, and for Class B is 16.

Figure 4-1 Sensor Network Settings



7. Click **OK**. Gateway and DNS are not necessary when testing in isolation.

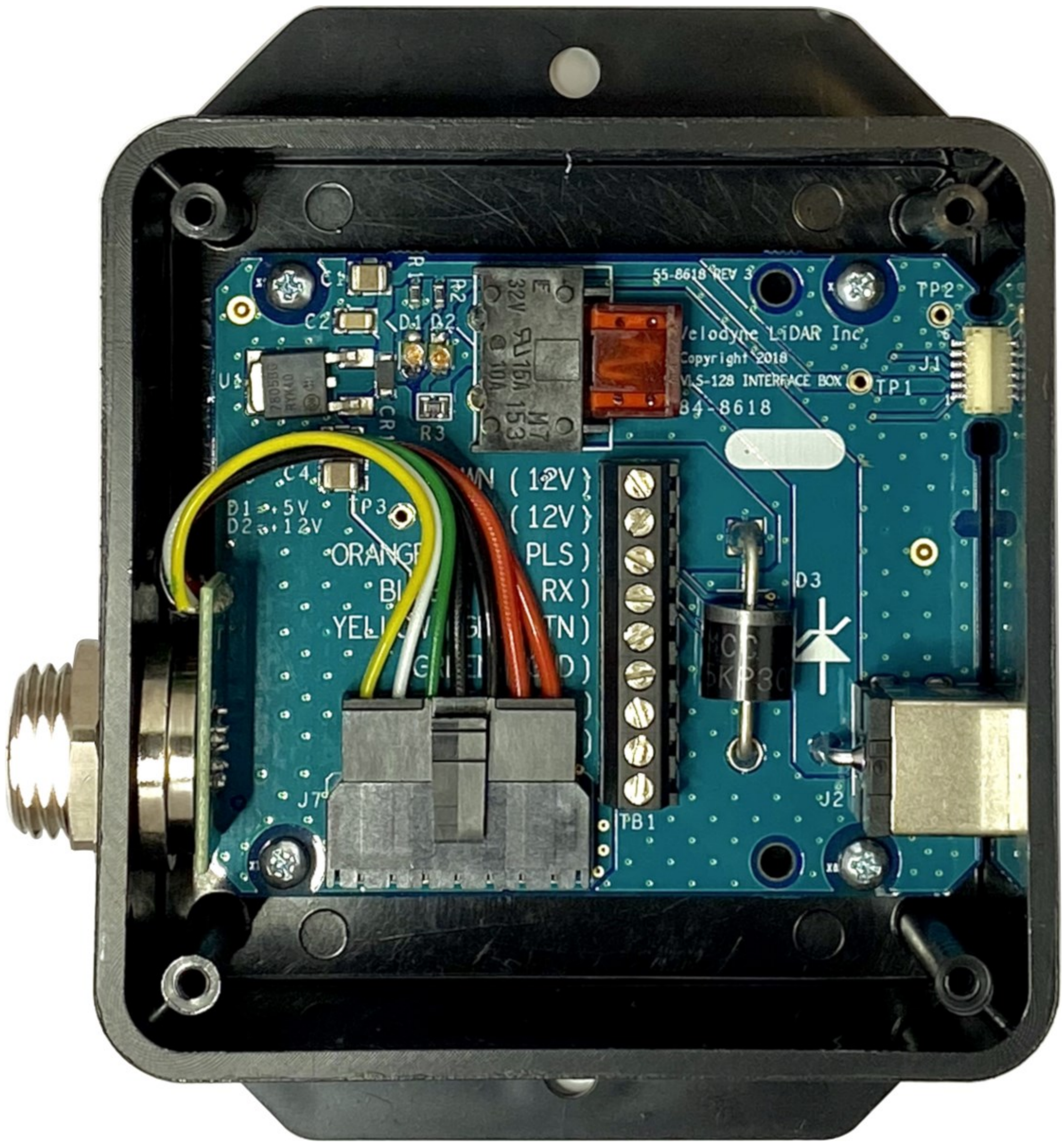
In some cases it may be necessary to disable the computer's firewall or configure it to allow UDP I/O on that Ethernet interface. How to do this is not covered here as the process varies widely.

4.2.2 Access Sensor's Web Interface

Now the computer is ready to connect to the sensor.

1. Plug the Ethernet cable into the computer and then plug its other end into the Ethernet port on the sensor's Interface Box (or equivalent if not using an Interface Box). *Figure 4-2 below* shows the Interface Box, its external ports, internal sensor terminal, and fuse.

Figure 4-2 Interface Box (power connection)



2. Connect power to the sensor's Interface Box.

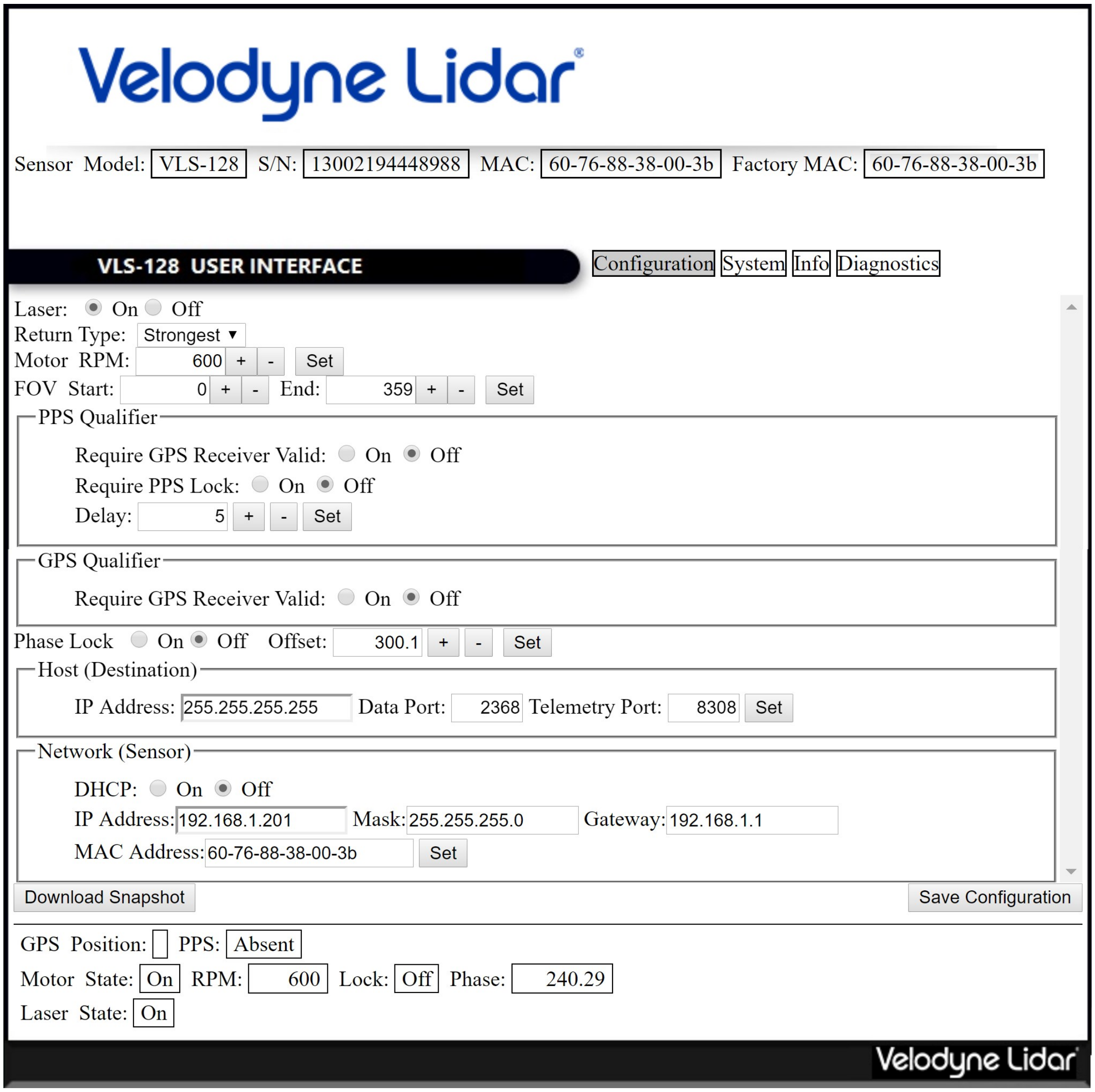
   When power is applied, two green LEDs in the Interface Box light up. The sensor begins scanning its environment and transmitting data approximately 30 seconds after power up.

3. On the computer, point a browser to http://192.168.1.201.

4. The sensor's Web Interface should appear (*Figure 4-3 below*).

   The Web Interface provides access to many of the sensor's control settings. See *Web Interface on page 85* for details.

Figure 4-3 Sample Web Interface Main Configuration Screen



4.2.3 Visualize Live Sensor Data with VeloView

Now that the computer can access the sensor's Web Interface, it's time to get a first look at the sensor's data.

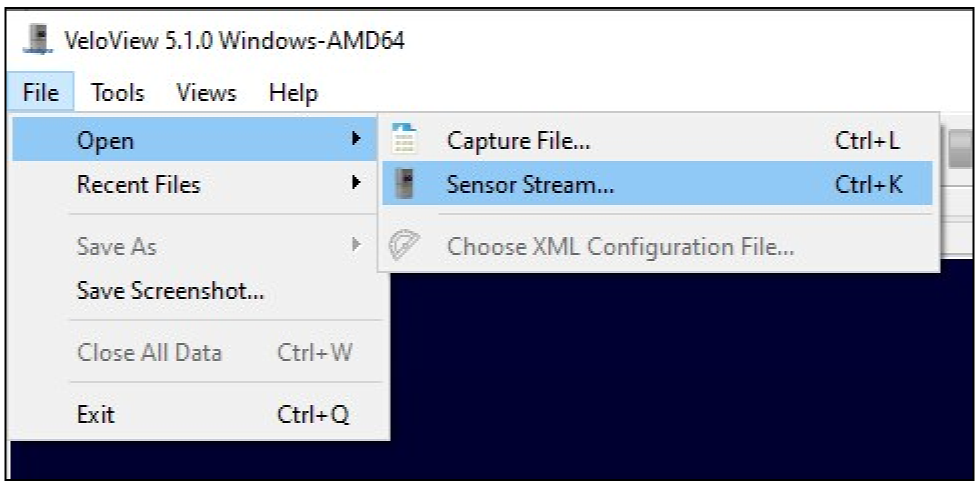
Note: VeloView is an open source visualization and 3D data recording application tailored for Velodyne Lidar sensors. Other visualization software (e.g. ROS, DSR and PCL) and middleware (e.g. Apollo, Autoware) can perform similar functions and may be used instead.

VeloView is documented in more detail in *VeloView on page 132*. If the application isn't already on the computer, perform the procedure detailed in *Install VeloView on page 133*. Older versions should be updated to at least the version installed by following the procedure.

4.2.3.1 VeloView Operation

1. Power-up the sensor.
2. Start the VeloView application.
3. Click on **File**->**Open** and select **Sensor Stream** (*Figure 4-4 below*).

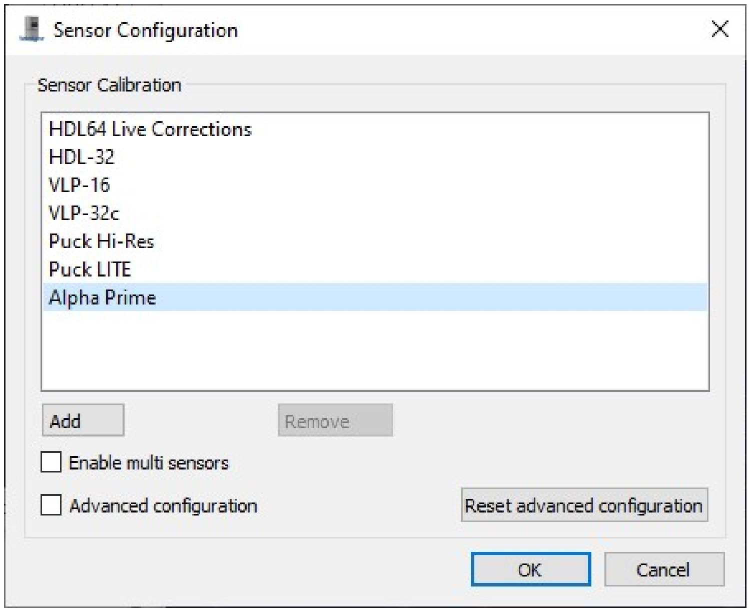
Figure 4-4 VeloView Open Sensor Stream



4. The Sensor Configuration dialog will appear (*Figure 4-5 on the next page*). Select the correct sensor type then click **OK**.

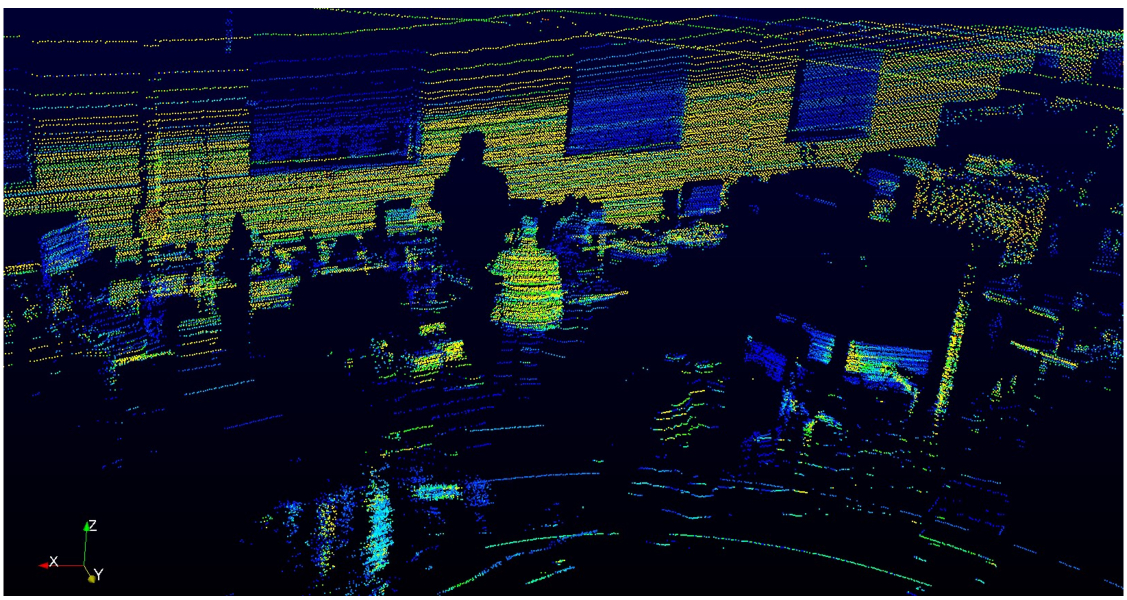
Depending on the version of VeloView you're using, you may see either 'Alpha Prime' or 'VeloView_VLS-128_FS1.xml' or both. They are identical xml files. Select either one.

Figure 4-5 VeloView Select Sensor Calibration



5. VeloView begins displaying the sensor data stream.

Figure 4-6 VeloView Sensor Stream Display



Above is an example of a VeloView screen in an office, workbench or lab scenario.

Chapter 5 • Installation & Integration

This chapter provides important information for integrating the Alpha Prime sensor into your application environment.

5.1 Overview

Ensure the sensor is functional first before beginning sensor integration. See *Verification Procedure on page 27*.

Common steps in installation and integration involve:

- Securely mounting the sensor to a vehicle, drone, robot, or other scanning platform
- Allowing for proper ventilation, providing thermal protection, and sensor encapsulation
- Connecting power to the sensor
- Connecting the sensor's Ethernet data output to a computer, switch, or network - see *Network Configuration on page 166*
- Optionally, connecting a GPS receiver or INS (Inertial Navigation System) - see *GPS, Pulse Per Second (PPS) and NMEA Sentence on page 45*

The typical sensor setup uses a standard computer or laptop connected to the sensor. However, it is recommended to use at least a 1 Gbps Ethernet adapter to accommodate the sensor data rate.

5.2 Mounting

The sensor base provides four M8 mounting holes and two precision locating holes for locator pins. The sensor may be mounted at any angle or orientation, though reliability should be best at 0° inclination (i.e. level to ground) as reductions to bearing life may occur at other orientations.

Accommodation must be made in the application mount for the two removable cables dropping down from the base of the sensor, plus room for tooling to secure or remove the cables.

Ensure the sensor is mounted securely to withstand vibration and shock without risk of detachment. The unit does not need shock proofing. The unit is designed to withstand automotive G-forces (i.e. 500 m/s^2 amplitude, 11 ms duration shock) and 3 G_{RMS} 5 Hz to 2,000 Hz vibration.

See *Mechanical Diagrams on page 126* for additional mounting details.

5.3 Encapsulation and Ventilation

For various reasons, you may wish to encapsulate the sensor, either wholly or partially. The working field of view of the enclosure should be highly transmissive of near-IR light at and near the 905 nm wavelength. Any moisture that enters the enclosure should have a way to drain.

The Alpha Prime generates a moderate amount of heat during normal operation. Strategies for managing heat in hot weather include employing a "thermal hat," exposing the sensor to moving air, and drawing heat from the sensor with a heat sink (e.g. aluminum plate(s)).

The sensor reports internal temperatures passively on its web interface. The same readings may be obtained programmatically via curl commands (i.e. http GET requests). See *Sensor Communication on page 85* for details. The sensor's operating temperature range can be found on its data sheet.

Do not operate the sensor without sufficient ambient air flow or cooling.

5.4 Connections

This section covers the sensor’s physical connections.

See *Network Considerations on page 167* before connecting one or more Velodyne Lidar sensors physically to your network. See *Ethernet and Network Setup on page 166* for instructions on how to configure the sensor's Ethernet connection.

5.4.1 Interface Box and Cable

The optional Alpha Prime Interface Box and Cable provides convenient connections for power and GPS inputs. It protects the sensor from power irregularities by incorporating a replaceable fuse and a reverse-current protection diode. When connected correctly to power, the diode allows current through to the sensor. If, however, the power and ground are switched, the diode blocks current flow, protecting the sensor.

Opening the Interface Box and making modifications inside, such as replacing the fuse or reverse-current protection diode, or repairing wires, is permitted.

For more information on the Interface Box, see *Interface Box Signals on page 44*.

5.4.2 Operation Without an Interface Box

If the sensor is used without a Velodyne Interface Box (meaning, a custom straight-through cable connects the sensor to source-power and signals), the user must provide sufficient reverse- and over-voltage protection circuitry as well as an equivalent fuse (or similar mechanism) to protect the sensor.

5.4.3 Power

The Alpha Prime does not have a power switch. It spins and operates whenever sufficient power is applied. In normal operation the sensor draws approximately 22 W of power. This can vary based on how much power the sensor deems necessary to use given its circumstances.

The **sensor's data sheet** specifies the input voltage that should be supplied to the unit. The sensor will not operate outside that voltage range.

The power input jack on the Interface Box accommodates a 5.5 mm (O.D.) x 2.5 mm (I.D.) barrel connector. The center pin has positive (+) polarity.

For those creating a custom DC power cable that plugs into the power jack, it is recommended that you use 18 AWG (or thicker) power wires. This gauge has sufficient ampacity to accommodate the sensor at maximum power draw in all conditions.

Note: Before operating the sensor, ensure it is securely mounted and that power will be applied in the correct polarity.

If the sensor doesn't spin up when power is applied, check the fuse, check the sensor's web interface if Laser is On and Motor RPM is a valid value, check the input voltage, and make sure that the power source (battery, power inverter, or power rail) is providing sufficient current. If they check out and yet the problem persists, contact *Technical Support on page 108*.

5.4.3.1 Power Considerations

Velodyne Lidar sensors are robust systems and can handle operational variance. But too much variance can affect performance and lifetime of the sensor. One of the best understood and most controllable environmental inputs is power. The following items are guidelines for input power to Velodyne sensors.

1. Power: Typical power consumption when the Alpha Prime is stationary is approximately 22 W. Under certain conditions, power consumption will increase. To assure proper operations during those conditions the recommended power supply output (i.e. power in to the sensor) is 60 W continuous, 65 W peak.
2. Voltage: 12 V is recommended but the sensor can support 9-28V DC clean power. As with all electronic equipment, a large change in voltage levels in a short time can cause issues with power regulation. Voltage fluctuations must remain within ±100 mV. Voltage transients should be as short as possible and not exceed 35 V.
3. Current: The required input current is in the range of 2.5 to 7.5 A at 12 V, with the numbers varying at other voltages. The current level should be maintained to sustain the voltage change parameter stated above.
4. Connection: The power rail to the sensor should be isolated from any power spike generating equipment such as motors, incandescent lamps, etc.
5. Power Over Ethernet (POE): The Alpha Prime is **not** capable of taking power over Ethernet. **Do not attempt to operate the sensor while connected to a POE switch, POE injector, or POE/POE+ capable NIC** -- unless a POE splitter is used to separate Ethernet signals from DC power, which is then fed separately to the sensor. **The likely result of POE injection attempts will be a trip to the factory for the sensor to have its internal Ethernet hardware repaired.** (The lasers may still operate but there won't be any communication.) POE is best for lower power networked devices.

Chapter 6 • Key Features

6.1 Calibrated Reflectivity

The Alpha Prime measures reflectivity of an object independent of laser power and distances involved. Reflectivity values returned are based on laser calibration against NIST-calibrated reflectivity reference targets at the factory.

For each laser measurement, a reflectivity byte is returned in addition to distance. Reflectivity byte values are segmented into two ranges, allowing software to distinguish diffuse reflectors (e.g. tree trunks, clothing) in the low range from retroreflectors (e.g. road signs, license plates) in the high range.

A retroreflector reflects light back to its source with a minimum of scattering. The Alpha Prime provides its own light, with negligible separation between transmitting laser and receiving detector, so retroreflecting surfaces *pop* with reflected IR light compared to diffuse reflectors that tend to scatter reflected energy.

- Diffuse reflectors report values from 0 to 120 for reflectivities from 0% to 100%.
- Retroreflectors report values from 121 to 255, where 255 represents an ideal or saturated reflection.

Factors in the external environment may attenuate, compress or shift reported reflectivity values downward a bit. Great distances is one of them.

Note: When a laser pulse doesn't result in a measurement, such as when a laser is shot skyward, the distance will be 0 and reflectivity is usually 0. The key is the distance of 0, because *0 is a valid reflectivity value* (i.e. one step above noise). A distance value of 0 means there was no measurement.

6.2 Laser Return Modes

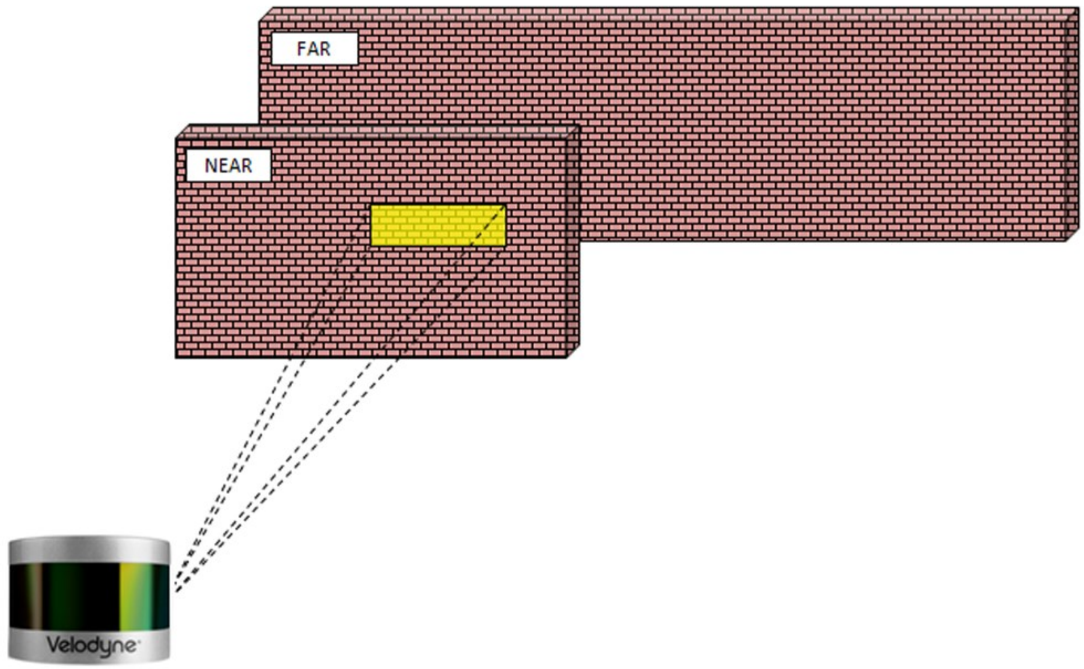
The Alpha Prime supports three laser return modes: **Strongest**, **Last**, and **Dual**. A sensor can be configured to handle laser returns in one of these ways interactively via the sensor's web interface (where the setting is called Return Type) or programmatically via curl command. (See *Configuration Screen on page 86* or *Sensor Control with curl on page 94* for additional information related to setting laser return modes.)

A laser return is a detection of a reflection -- as long as its distance is non-zero. Up to two returns per laser shot are supported by the Alpha Prime.

6.2.1 Single Return Modes: Strongest, Last

As shown in *Figure 6-1 below*, when a laser pulse hits a solid wall a single return or measurement is obtained. In this situation, the reading is considered both the strongest and the last return. (More on the nature of laser pulses emitted by your sensor, including the rectangular shape of the pulse, is covered in *Laser Pulse on page 149*.)

Figure 6-1 A Single Return



6.2.2 Multiple Returns

Multiple laser returns are possible from any single laser shot because of beam divergence. When a laser pulse leaves the sensor it slowly, gradually grows larger. A pulse can be big enough to hit multiple objects producing multiple reflections. Usually, the farther away a reflection starts, the weaker it is at the detector. Bright or retroreflective surfaces may flip that, however.

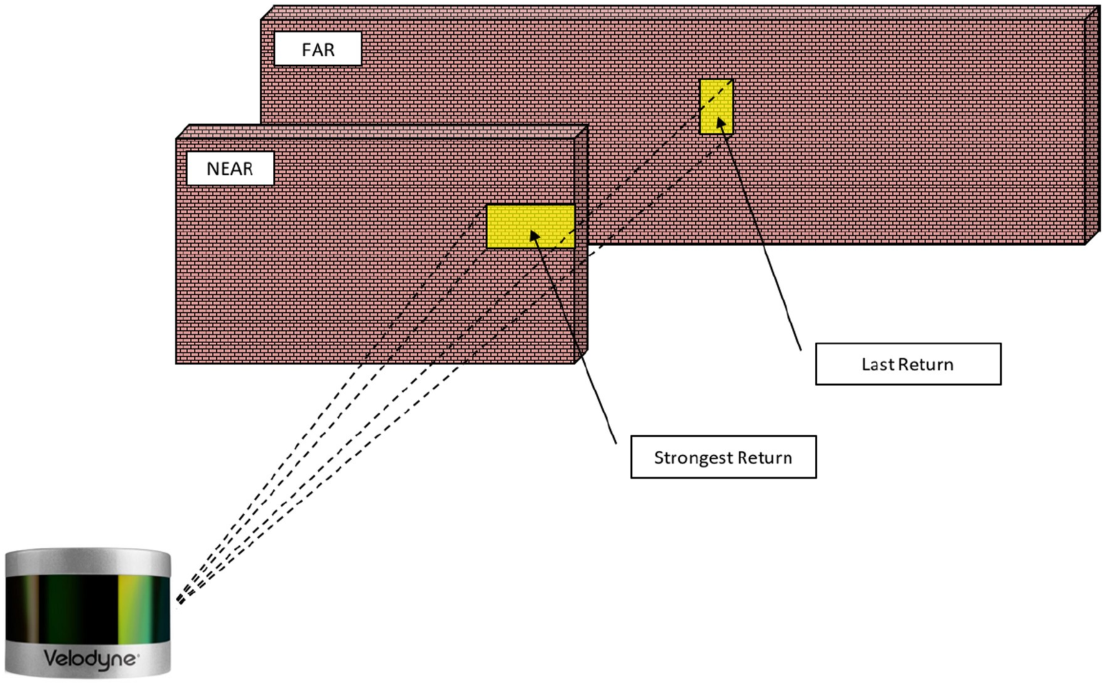
The Alpha Prime analyzes multiple returns and reports either the strongest return, the last return, or both returns, depending on the laser return mode setting. If the setting is Strongest and a pulse produces multiple returns, only the strongest reflection results in a measurement. Likewise, if the setting is Last, only the last (time-wise) reflection results in a measurement. This could be used to measure distances to the ground from the air.

6.2.3 Dual Return Mode

Figure 6-2 on the next page shows the majority of a laser pulse striking the near wall while the remainder hits the far wall. The Dual return mode setting allows you to obtain both measurements.

Note that the sensor records both returns only when the separation between the two objects is 1.5 meters or more.

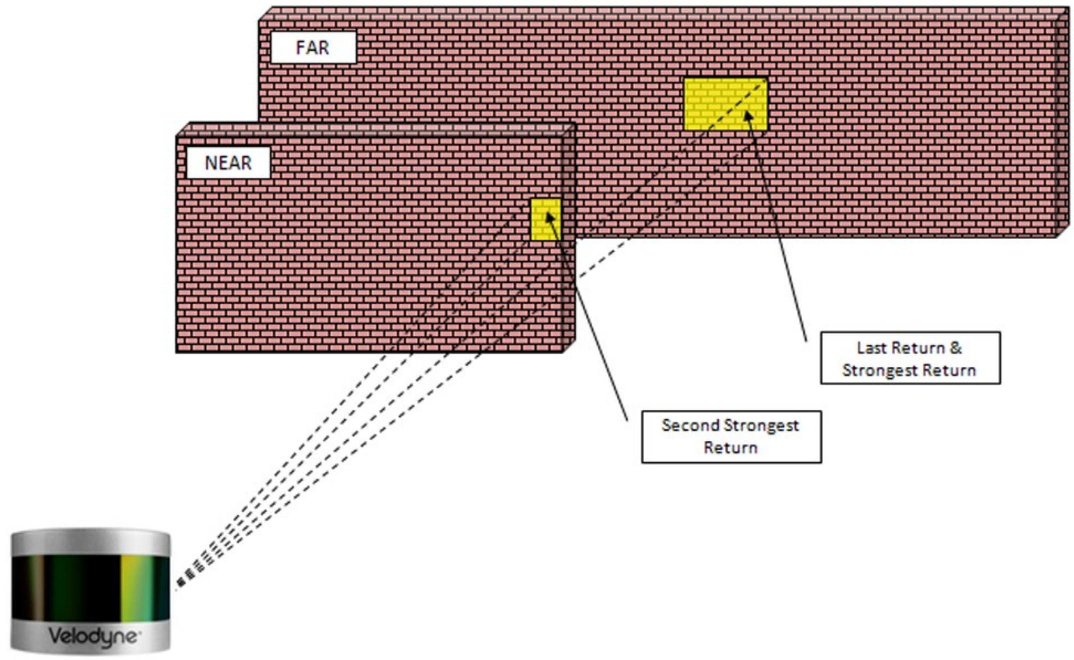
Figure 6-2 Dual Return with Last and Strongest Returns



As shown in *Figure 6-3 on the facing page*, in the event the strongest return is the last return, the second-strongest return is reported. The majority of the beam hits the far wall and is (in this case) the strongest return.

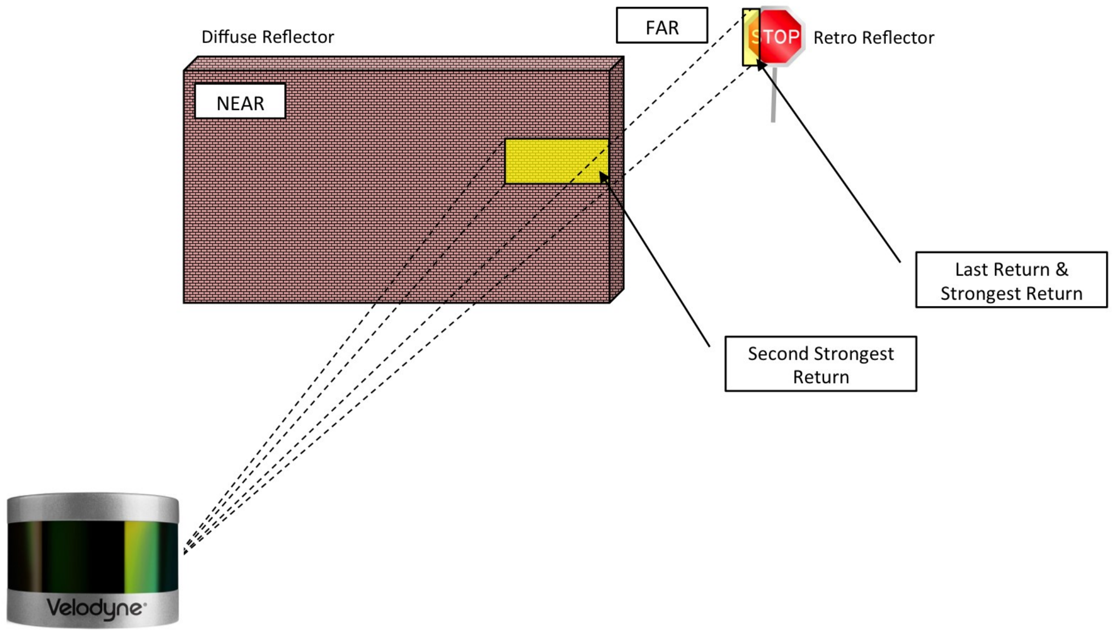
It's entirely possible, however, that the far wall might be far enough away that despite reflecting the majority of the beam, the return from the near wall would be the strongest return.

Figure 6-3 Dual Return with Second Strongest Return



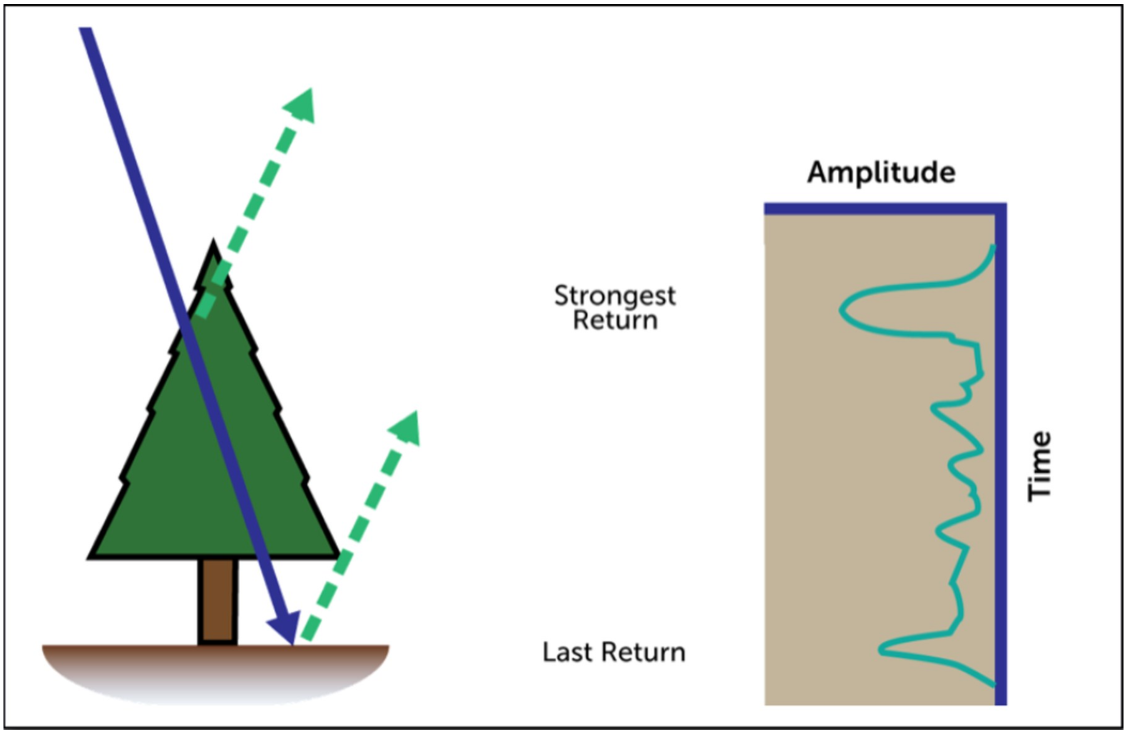
It's also possible that a small portion of the beam clips a retro-reflector and returns more energy than the majority of the beam, as in *Figure 6-4 on the next page*.

Figure 6-4 Dual Return with Far Retro-Reflector



The dual return function is often used in forestry applications where measurement of the height of trees is desired. *Figure 6-5 on the facing page* indicates a sample response when the laser pulse initially hits the upper canopy, penetrates it, and eventually hits the ground, producing multiple returns. Which laser return mode would be best in this situation?

Figure 6-5 Forestry Application Multiple Returns



Note: In dual return mode, the data rate of the sensor doubles.

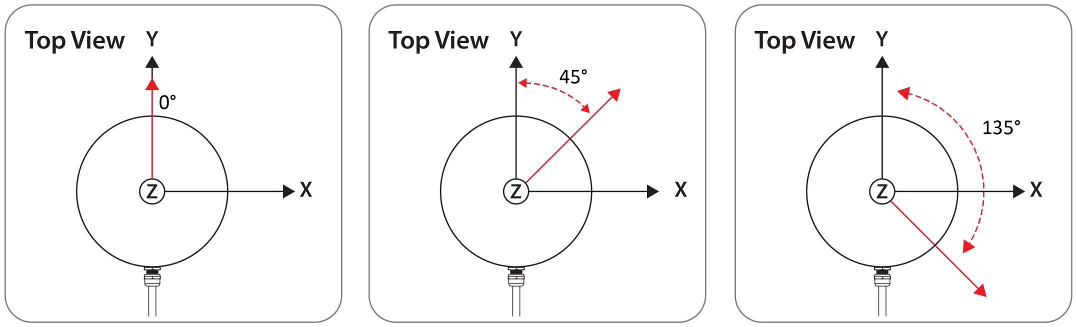
6.3 Phase Locking Multiple Sensors

When using multiple Velodyne Lidar sensors in proximity to one another, users may observe interference between them due to one sensor picking up a reflection intended for another. To minimize this interference, the sensor provides a phase-locking feature that enables the user to control where the laser firings overlap.

The Phase Lock feature can be used to synchronize the relative rotational position of multiple sensors based on the PPS signal and relative sensor orientation. To operate correctly, the PPS signal must be present and locked. Phase locking works by offsetting the firing position based on the rising edge of the PPS signal.

Note: For phase lock to work correctly, the sensor's RPM setting must be set to a multiple of 60 RPM between 300 RPM and 1200 RPM (inclusive). However, for this release, RPM is fixed at 600 due to a hardware issue while a remedy is sought.

Figure 6-6 Phase Locking Example



Phase offset at 0° Phase offset at 45° Phase offset at 135°

The red arrows shown in *Figure 6-6 above* indicate the firing direction of the sensor's laser at the moment it receives the rising edge of the PPS signal.

Additional information for phase locking multiple sensors is located in *Phase Lock on page 159*.

6.4 Power Stepping

Velodyne lidar sensors employ transmit power stepping.

Depending on conditions, the sensor increases or decreases transmit power in an effort to maximize the signal to noise ratio. Getting readings from objects at great distances often requires the higher power settings, whereas highly reflective white, diffuse objects close by require the lowest power settings. The sensor makes thousands of decisions per second regarding power stepping. It handles each channel independently from the others.

Chapter 7 • Sensor Inputs

This chapter covers sensor input requirements and functionality, including power, PPS, and Ethernet. It also covers scenarios for obtaining GPS input.

7.1 Power Requirements

Power requirements are detailed in Chapter 5: *Power on page 34*.

7.2 Interface Box Signals

The Interface Box is described in *Interface Box and Cable on page 34*.

Figure 7-1 Interface Box

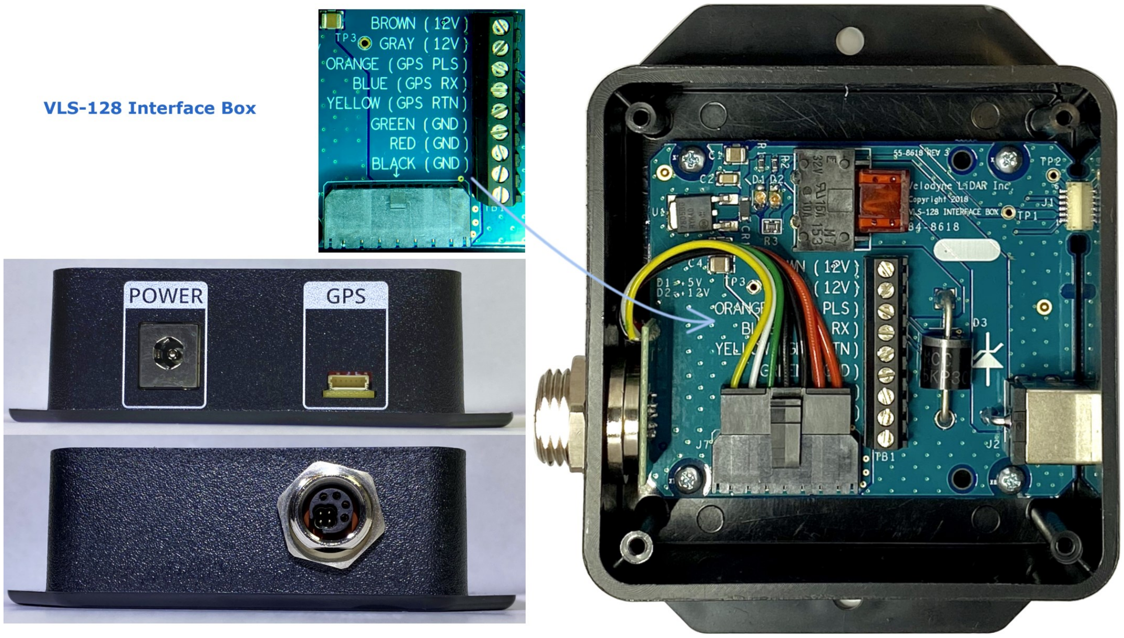
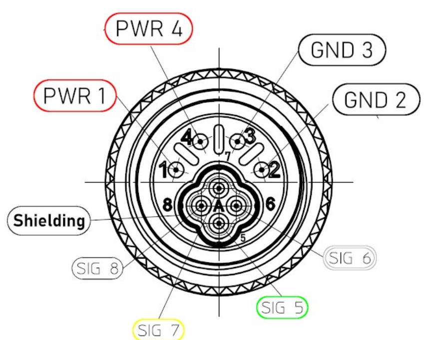


Figure 7-2 Interface Box Signals



M12 (CHT) Bulkhead Connector & Pigtail Cable					IFB Terminal Block	
PIN	**AWG**	**Wire Color**	**Signal**		**Label**	**Pin**
1	18	RED	POWER IN	←	(12V)	1
4	18	RED	POWER IN	←	(12V)	2
7	24	YELLOW	GPS PPS	←	(GPS PLS)	3
6	24	WHITE	GPS DATA IN	←	(GPS RX)	4
5	24	GREEN	GPS GROUND	→	(GPS RTN)	5
3	18	BLACK	GROUND RETURN	→	(GND)	6
2	18	BLACK	GROUND RETURN	→	(GND)	7
8	24		unused		(GND)	8
			Shielding		Shielding (GND)	9

See *Wiring Diagrams on page 130* for more.

7.3 Ethernet Interface

Your sensor's primary interface is its Ethernet interface. All command and control occurs over it, and all sensor data is transmitted over it.

The sensor's Ethernet cable connects to any standard 1000BASE-T Ethernet NIC or switch with MDI or AUTO MDIX capability.

See *Ethernet and Network Setup on page 166* for more.

7.4 GPS, Pulse Per Second (PPS) and NMEA Sentence

Your sensor can synchronize data with precise GPS-supplied time. Synchronizing to a GPS-supplied Pulse-Per-Second (PPS) signal provides the ability to compute the exact firing moment of each data point as required by some geo-referencing applications. See *Time Synchronization on page 155* for details on GPS time synchronization and how important it is for associating sensor data with the sensor's environment.

To utilize these features, configure your GPS/INS device to issue a PPS signal in conjunction with a once-per-second NMEA GPRMC or GPGGA sentence. No other NMEA sentence is accepted by the sensor.

Note: The GPRMC record may be configured for HHMMSS, HHMMSS.s, HHMMSS.ss, and HHMMSS.sss formats.

7.4.1 GPS Input Signals

The serial data output from the GPS/INS is connected to the sensor's Interface Box via the screw terminal labeled: "GPS RX."

The PPS output from the GPS/INS is connected to the sensor's Interface Box via the screw terminal labeled: "GPS PLS."

The ground signal from the GPS/INS is connected to the sensor's Interface Box via the screw terminal labeled: "GPS RTN." This ground is the reference signal the GPS RX and GPS PLS are measured against.

Note: You can use the provided GPS port on the Interface Box if using the Velodyne GPS Receiver (P/N 80-GPS18LVC). See *GPS Port (J1) on page 48* for details.

7.4.2 Electrical Requirements

- "High" voltage must be greater than 3.0 V and less than 5.0 V.
- "Low" voltage must be less than 1.2 V, and should be greater than or equal to 0 V; however, it is allowed to go as low as -5.0 V.
- The GPS/INS unit must be capable of supplying at least 2 mA of current in the "High" state.

7.4.3 Timing and Polarity Requirements

The PPS synchronization pulse and either GPRMC or GPGGA sentence should be received in alternating fashion and not overlap or coincide. The PPS synchronization pulse width is not critical (typical lengths are between 10 µs and 200 ms). The 300 ms minimum gap after receiving the NMEA message gives the sensor enough time to extract and process time data from the message so it is ready to set the time when the leading edge of the next PPS arrives. If no PPS arrives in time then the time information extracted from the NMEA message eventually expires.

Figure 7-3 Synchronizing PPS with NMEA Message

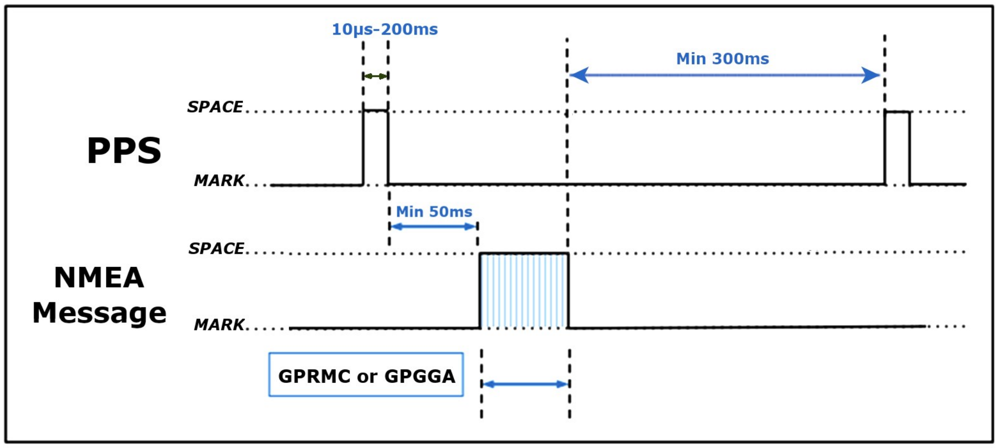


Figure 7-4 below is an example of a PPS sync pulse followed too closely by an NMEA message. The minimum 50 ms gap requirement between trailing edge of PPS and leading edge of NMEA message is not met.

Figure 7-4 PPS Signal Followed Too Closely by NMEA Message

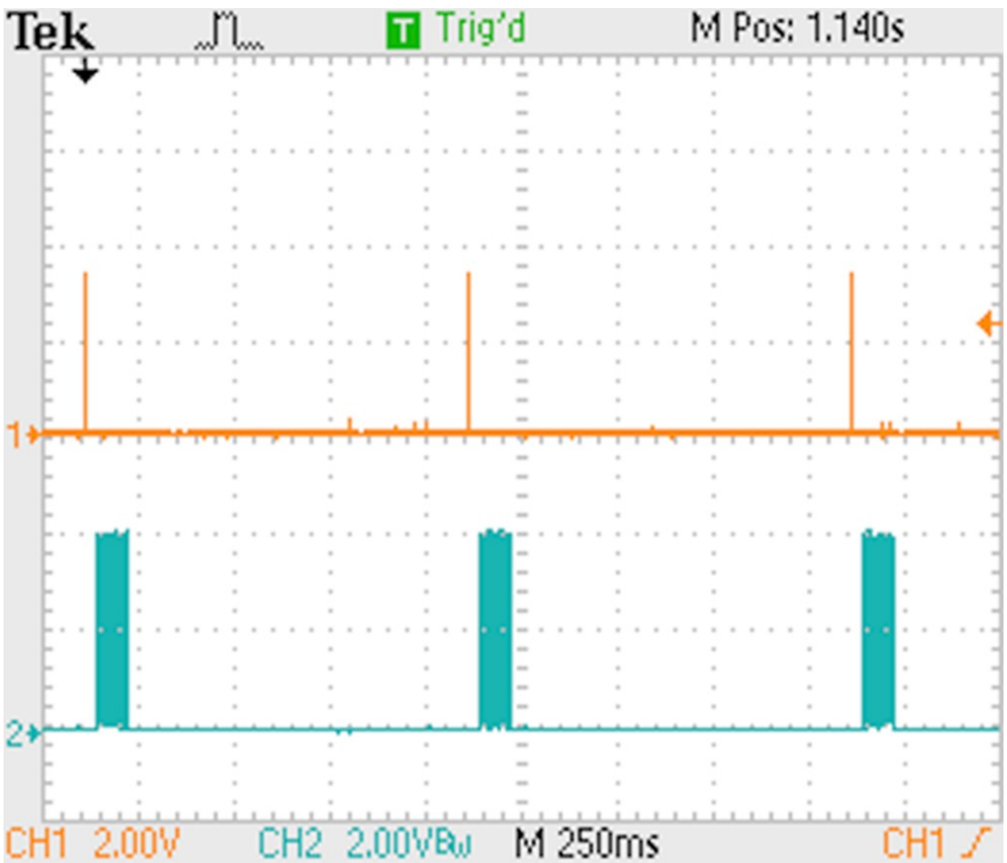
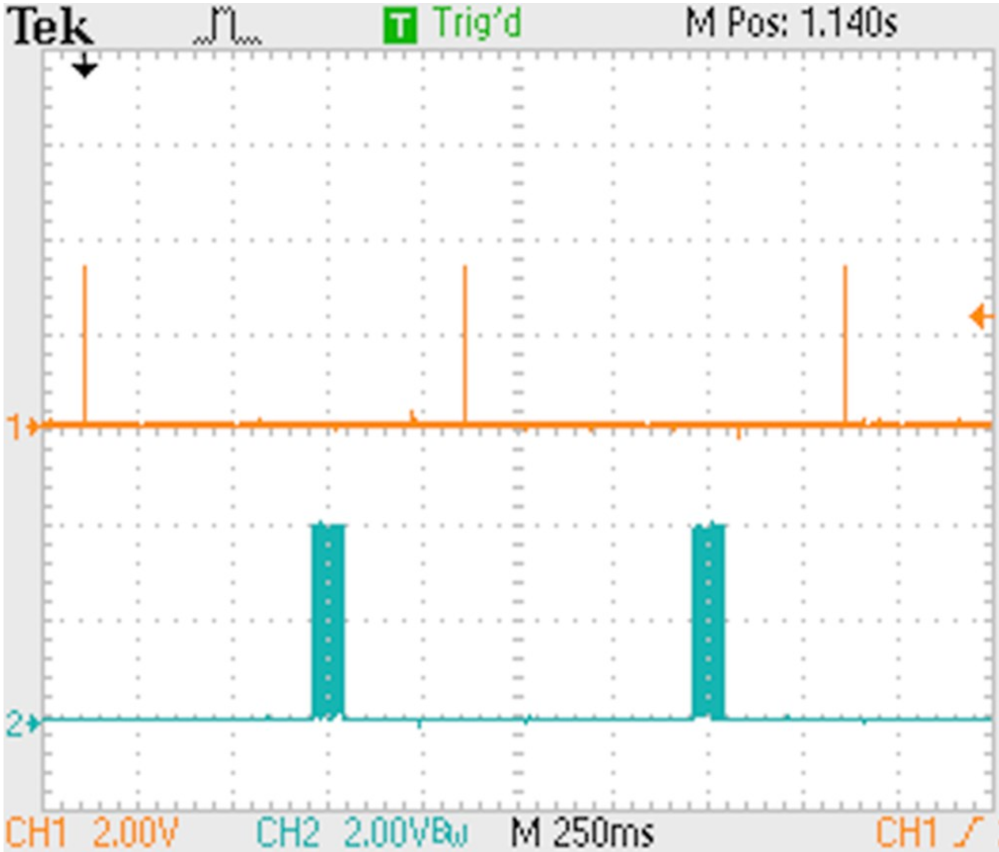


Figure 7-5 below is an example of an NMEA message falling in the sweet spot between two PPS signals. The minimum 50 ms gap requirement between trailing edge of PPS and leading edge of NMEA message is met, and the minimum 300 ms gap requirement between trailing edge of the NMEA message and leading edge of PPS is met.

Figure 7-5 NMEA Message Leading PPS Signal by 320 ms



7.4.4 Sample Serial Transmission

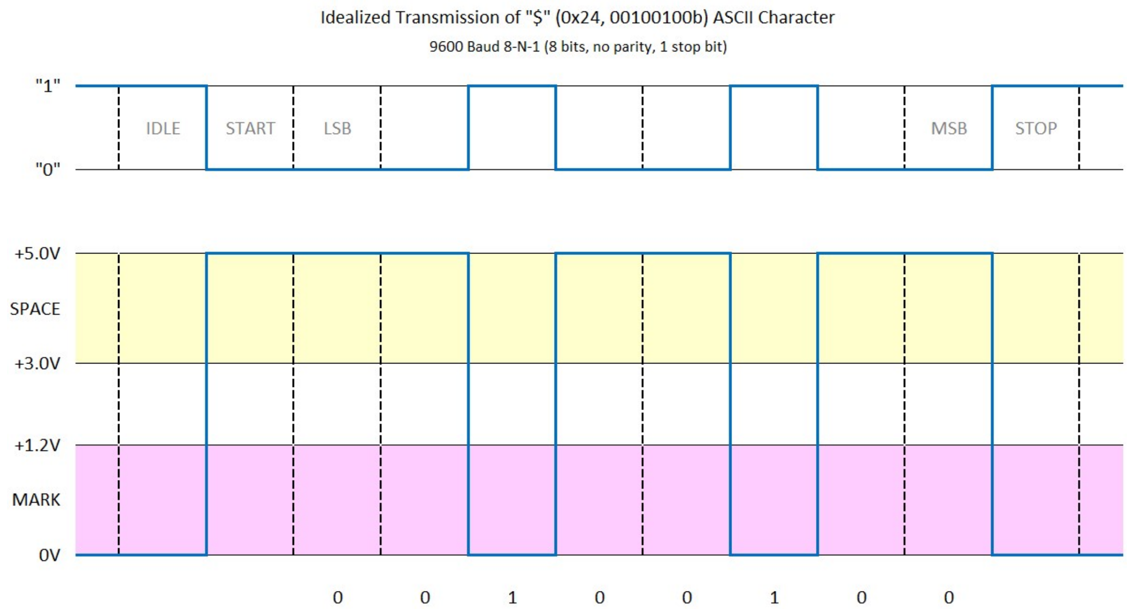
The serial connection for the NMEA message is TTL serial, or more commonly termed TTL RS-232. It is not True RS-232.

- Low voltages are marks and represent a logical 1.
- High voltages are spaces and represent a logical 0.

The serial line idle state (MARK) is a low voltage indicating a logical 1. When the start bit is asserted, the positive voltage will be asserted representing a logical 0.

As an example, an idealized sample transmission of an ASCII "$" character is shown in *7.4.4 above*. Note that the binary value of the '$' character, 00100100b ('b' suffix indicates the value is binary), is transmitted from least-significant bit to most-significant bit, interpreting the diagram from left to right (i.e. the time axis).

Figure 7-6 Sample Serial Transmission of $ Character



7.4.5 GPS Port (J1)

The OEM Part Number for the physical GPS Port is **SM06B-SRSS-TB(LF)(SN)**: Connector Header Surface Mount, Right Angle 6 position 0.039" (1.00mm) Male.

The OEM Part Number for the mating female connector is SHR-06V-S-B. However, it does not include contacts. Instead, use of Part Number A06SR06SR30K152A (JUMPER 06SR-3S - 06SR-3S 6": 6 Position Cable Assembly Rectangular Socket to Socket, Reversed 0.500' (152.40mm, 6.00")) is recommended if making a custom cable. Other lengths are available. Velodyne Lidar does not sell these parts.

Table 7-1 GPS Port Signals

PIN	Signal
1	GPS PPS
2	+5 V (supplied to GPS)
3	GND
4	GPS RECEIVE
5	GND
6	GPS TRANSMIT

7.4.6 GPS Connection Scenarios

Depending on the user’s application, the source of the NMEA message (and PPS) can be a GPS/INS receiver or a substitute device such as a laptop or microcomputer.

There are three common connection scenarios: connecting to a sensor from a Velodyne-sourced Garmin 18x LVC GPS receiver, connecting to it from a computer's serial port, and connecting directly from a microcomputer’s UART. The first connection is via the Interface Box's GPS port, the other two are via the screw terminals inside the Interface Box.

7.4.6.1 Connecting a Garmin 18x LVC GPS Receiver

As an option, Velodyne Lidar offers a Garmin 18x LVC GPS Receiver (P/N 80-GPS18LVC) pre-configured (per *Figure 7-7 below*) for optimized operation with your sensor. GPRMC is configured by default but GPGGA is also supported.

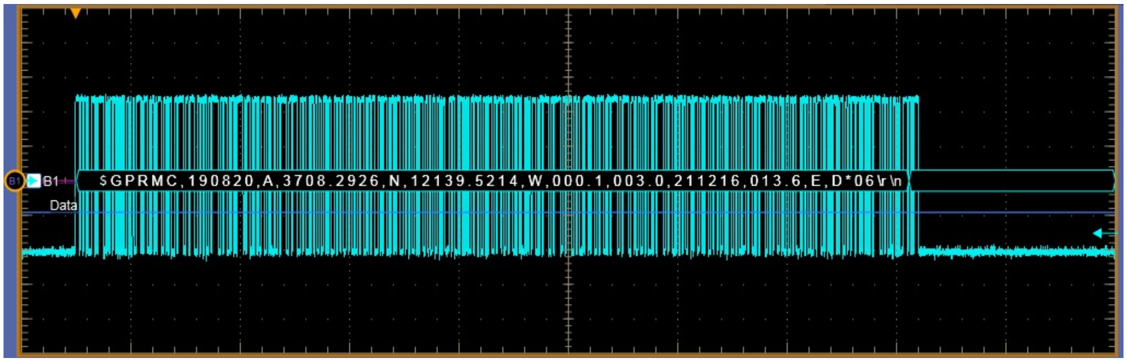
Figure 7-7 Garmin 18x LVC Configuration

GPS SETTING	VALUE
Baud Rate	9600
Power Save Mode	Off
NMEA Sentences	GPRMC*
NMEA 2.30 Mode	On
NMEA Output Time	1 sec
PPS Mode	1 Hz
PPS Length	100 ms
PPS Auto Off	Off
Fix Mode	Automatic
DGPS Mode	WAAS Only
Differential Mode	Automatic
Dead Reckon Time	30 sec
Position Averaging	Off

* Or GPGGA

The receiver plugs directly into the Interface Box’s GPS port and is used to synchronize your sensor's timestamp with precision GPS time. The signals from the Garmin receiver will be similar to those shown in *Chapter 7 • on page 43* where the signal is normally low and zeros are represented by the high voltage.

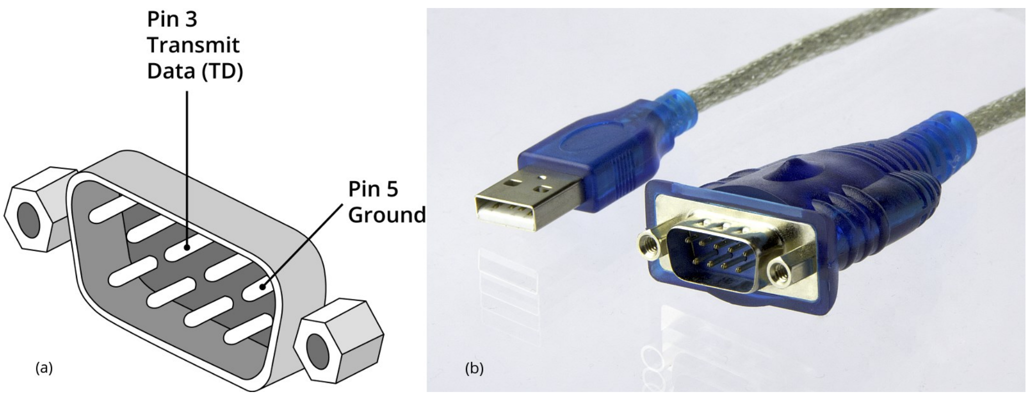
Figure 7-8 Sample NMEA Message



7.4.6.2 Connecting to a computer's serial port

In some situations, you may wish to source NMEA messages from a computer instead of a GPS receiver.

Figure 7-9 DB9 Pin-outs (DTE) and USB-to-Serial Adapter



An example of a standard DB9 serial port is shown in *Figure 7-9 above* (a). Because modern computers use USB ports, a USB to Serial Adapter, as shown in *Figure 7-9 above* (b), will be required. The DB9 connector on the adapter provides a signal with the proper polarity and compatible voltage levels to connect directly to the sensor's Interface Box.

After connecting the USB to Serial Adapter to your computer and wiring up a mating DB9 connector (not pictured), you can complete the connection to your sensor. Remove the cover from the Interface Box and make the following connections to the terminal screw strip in the Interface Box.

- DB9 pin 3 to GPS RX
- DB9 pin 5 connects to GPS RTN

7.4.6.3 Connecting to a microcomputer's UART

In other situations, you may wish to source NMEA messages from a microcomputer such as a Raspberry Pi or Arduino, instead. The native signal coming from the microcomputer's UART will have incorrect polarity. In this instance, invert the signal using a 7404 hex inverter chip or equivalent circuitry. To connect to the sensor, remove the cover from the Interface Box and connect the appropriate leads to the GPS RX and GPS RTN connections on the terminal strip.

Figure 7-10 on the facing page shows a signal directly from a Raspberry Pi UART output and *Figure 7-11 on the facing page* shows the same output inverted into a signal compatible with your Velodyne sensor.

Figure 7-10 Signal Directly from UART (incorrect polarity)

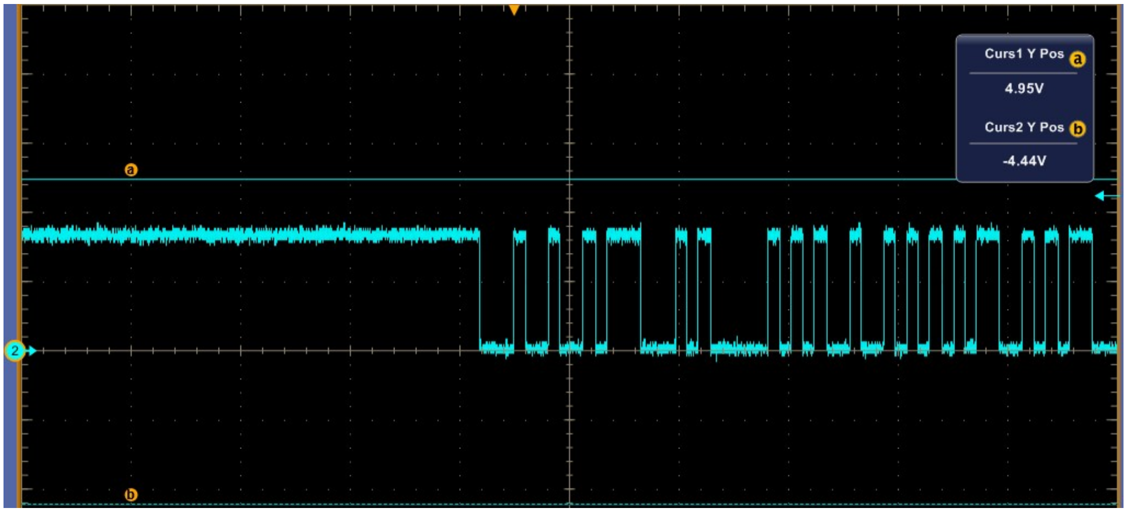
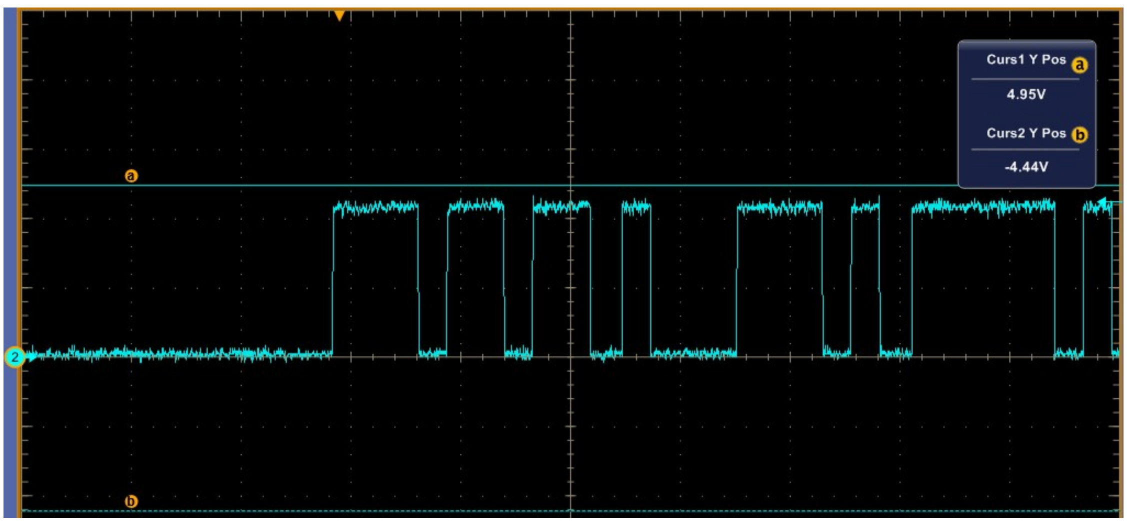


Figure 7-11 Inverted Signal from UART (correct polarity)



7.4.7 NMEA Message Formats

The sensor accepts GPRMC sentences by default. GPGGA may be substituted in place of GPRMC.

The sensor accepts both pre- and post-NMEA 0183 Version 2.3 sentence structures. The difference is that in version 2.3 a Fix Mode Indicator field was added after Magnetic Variation and before the Checksum.

NMEA messages are typically terminated with sentence delimiter <CR><LF> (HEX 0D0A).

7.4.7.1 Pre-NMEA Version 2.3 Message Format

Table 7-2 on the next page provides a description of the pre-NMEA 0183 Version 2.3 message format contents shown below.

```
$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A
```

Table 7-2 Pre-NMEA 0183 Version 2.3 Message Format

Value	Description
$GPRMC	Recommended Minimum sentence
123519	Fix taken at 12:35:19 UTC
A	Receiver status: A = Active, V = Void
4807.038,N	Latitude 48 deg 07.038' N
01131.000,E	Longitude 11 deg 31.000' E
022.4	Speed over the ground (knots)
084.4	Track made good (degrees True)
230394	23rd of March 1994
003.1,W	Magnetic Variation
*6A	* followed by 2-byte Checksum

7.4.7.2 NMEA 0183 Version 2.3 Message Format

Table 7-3 below provides a description of the post-NMEA 0183 Version 2.3 message format contents shown below.

```
$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W,A*07
```

Table 7-3 Post-NMEA 0183 Version 2.3 Message Format

Value	Description
$GPRMC	Recommended Minimum sentence
123519	Fix taken at 12:35:19 UTC
A	Receiver status: A = Active, V = Void
4807.038,N	Latitude 48 deg 07.038' N
01131.000,E	Longitude 11 deg 31.000' E
022.4	Speed over the ground (knots)
084.4	Track made good (degrees True)
230394	23rd of March 1994

Value	Description
003.1,W	Magnetic Variation
A	Fix Mode Indicator: A = Autonomous, D = Differential, E = Estimated, N = Not valid, S = Simulator. Sometimes there can be a null value as well.
*07	* followed by 2-byte Checksum

Note: The Receiver Status (aka Validity) field in the GPRMC message ('A' or 'V') should be checked to ensure the GPS receiver is actively positioning and is providing trustworthy UTC (Coordinated Universal Time) and position updates. If status is Void, which usually occurs when the GPS receiver is searching for satellites, GPS position should be ignored. In many instances, the GPS receiver, if its status is Void, will continue to provide a PPS signal based on the GPS receiver's internal clock. More on this can be found in *Time Synchronization on page 155*.

7.4.8 Accepting NMEA 0183 Messages Via Ethernet

Your sensor can accept NMEA 0183 sentences over Ethernet. Three methods are supported:

1. A host opens a TCP connection to the sensor’s port 10110 and transmits NMEA sentences on the socket.
2. A host transmits NMEA sentences in UDP packets to the sensor’s UDP port 10110.
3. A host transmits NMEA sentences in UDP packets to a broadcast address on port 10110 with an NMEA sentence.

1 and 2 require knowledge of the sensor’s IP address. 3 does not, but it may load devices on the network that don’t want the packets.

Note: The supported NMEA sentence and syntax are exactly the same as on the wired (serial) GPS interface. Supported sentence delimiter sequences include <CR><LF> (HEX 0D0A, the standard), <CR> by itself, and <LF> by itself.

The support follows the ad-hoc but widely supported NMEA over Ethernet approach utilized by the OpenCPN project. Additional information can be found at the following web sites:

- http://stripydog.blogspot.com/2015/03/nmea-0183-over-ip-unwritten-rules-for.html
- https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:supplementary_software:nmea_instruments

Chapter 8 • Sensor Operation

This chapter provides details on operational elements such as RPM, the laser firing sequence, point density, and how to determine throughput rate and angular resolution.

8.1 Firing Sequence

Details of the sensor's firing sequence are specified in *Discreet Point Timing Calculation (HDL) on page 67*. The pattern of firing in terms of time is illustrated in *Figure 9-8 on page 69*.

8.2 Throughput Calculations

Some terms used here are defined in *Legacy Packet Format (HDL) on page 59*. The *Legacy (HDL) Data Packet Structure on page 62* and *Legacy Position Packet Structure on page 65* are defined in Chapter 9.

8.2.1 Data Packet Rate

In single return mode, up to 3 firing sequences can fit in a data packet: 3 x 58.5688 µs = 0.1757064 ms is the accumulation delay per packet. Whereas, only one firing cycle can fit in a data packet in dual return mode: 1 x 58.5688 µs = 0.0585688 ms is the accumulation delay per dual return data packet. (Note that 58.5688 µs is an *average*. See *Figure 9-8 on page 69* for details.)

Single return mode: 1 packet/0.1757064 ms ≈ 5691.3 packets/second.

Dual return mode: 1 packet/0.0585688 ms ≈ 17073.9 packets/second.

Single return mode: 1248 bytes/packet * 5691.3 packets/second ≈ 7102742.4 bytes/second

Dual return mode: 1248 bytes/packet * 17073.9 packets/second ≈ 21308227.2 bytes/second

8.2.2 Position Packet Rate

Position packets arrive at a rate more or less independent of data packet rate. (If both a position packet and a data packet are ready to be sent by the sensor, the position packet is delayed while the data packet is sent.)

Position packets typically arrive at a rate of about 94 position packets/second.

554 bytes/position packet * 94 packets/second = 52076 bytes/second

8.2.3 Total Packet Rate

Total packet rate is approximately 5691.3 data packets/second + 94 position packets/second ≈ 5785.3 packets/second, approximately. Remember, this is an average.

Single return mode: Summing 7102742.4 bytes/second (data) and 52076 bytes/second (position) yields 7154818.4 bytes/second.

Dual return mode: Summing 21308227.2 bytes/second (data) and 52076 bytes/second (position) yields 21360303.2 bytes/second.

8.2.4 Laser Measurements Per Second

One data point is equivalent to one laser measurement. It is possible to obtain multiple laser measurements from a single laser shot, depending on the Return Mode setting.

8.2.4.1 Measurements per Second in Single Return Mode

In Single Return Mode (either Strongest or Last), each laser firing may result in a single data point (if it reflects off something), which is a measurement of distance to target and its reflectivity.

128 data points/firing sequence * 3 firing sequences/packet = 384 data points/packet

And now we compute the maximum number of laser measurements per second.

384 data points/packet * 5691.3 packets/second ≈ 2185459.2 laser measurements per second

8.2.4.2 Measurements per Second in Dual Return Mode

In Dual Return Mode, each laser firing may result in 0, 1 or 2 data points (depending on what the laser pulse reflects off of), each of which is a measurement of distance to target and its reflectivity.

256 data points/firing sequence * 1 firing sequence/packet = 256 data points/packet

And now we compute the maximum number of laser measurements per second.

256 data points/packet * 17073.9 packets/second = 4370918.4 laser measurements per second, approximately

8.3 Rotation Speed (RPM)

The sensor's motor can normally be set to rotate between 300 RPM and 1200 RPM, inclusive, in increments of 60 RPM (i.e. 300, 360, 420, 480, … 1140, 1200).

(Note that Max RPM was set to 600 for certain units during the 5.2.3 firmware era due to an rpm-related hardware issue, so the valid range was from 300 to 600 RPM, in increments of 60, for those units. Sensors made after that period did not have the restriction.)

Note: If the RPM setting is not evenly divisible by 60, neither motor speed control nor phase lock functions will function properly.

The user can set this parameter using the sensor's Web Interface or curl commands. See *Configuration Screen on page 86* or *Sensor Control with curl on page 94* for more on setting rotation speed.

Note: In VeloView, one rotation is referred to as a single "frame" of data, beginning and ending at approximately 0° azimuth. The number of frames per second of data generated depends entirely on the RPM setting, e.g. 600 RPM / (60 s/min) = 10 frames per second.

Choice of RPM is up to the user and may depend on the application. For example, an autonomous vehicle may wish to increase rotation speed so it could more quickly identify frame-to-frame variations in the environment, such as a child running across a street. Conversely, a defined-space monitoring solution might use a lower RPM to increase the detail in each frame.

8.3.1 Horizontal Angular (Azimuth) Resolution

Though the firing timing of the sensor is, on average, consistently close to 58.5688 µs (55.275 µs for pre-5.2.3 firmware) per firing sequence, it is still mainly the speed of rotation that dictates the angular resolution of the sensor. The faster it spins, the coarser the measurements in a full revolution. The slower it spins, the higher the resolution.

An example calculation for 600 RPM is given in *Equation 8-1 below*.

Equation 8-1 Azimuth Resolution at 600 RPM

$$\text{Azimuth}_{\text{Resolution}} = 600\,\frac{rev}{min} \times \frac{1}{60}\,\frac{min}{s} \times 360\,\frac{^\circ}{rev} \times \frac{55.275\times10^{-6}\,s}{firing\ cycle}$$

$$\text{Azimuth}_{\text{Resolution}} = 0.19899^\circ / firing\ cycle$$

By plugging in a different RPM in the equation above you can calculate the azimuthal resolution for any rotation speed.

Table 8-1 Rotation Speed vs Resolution

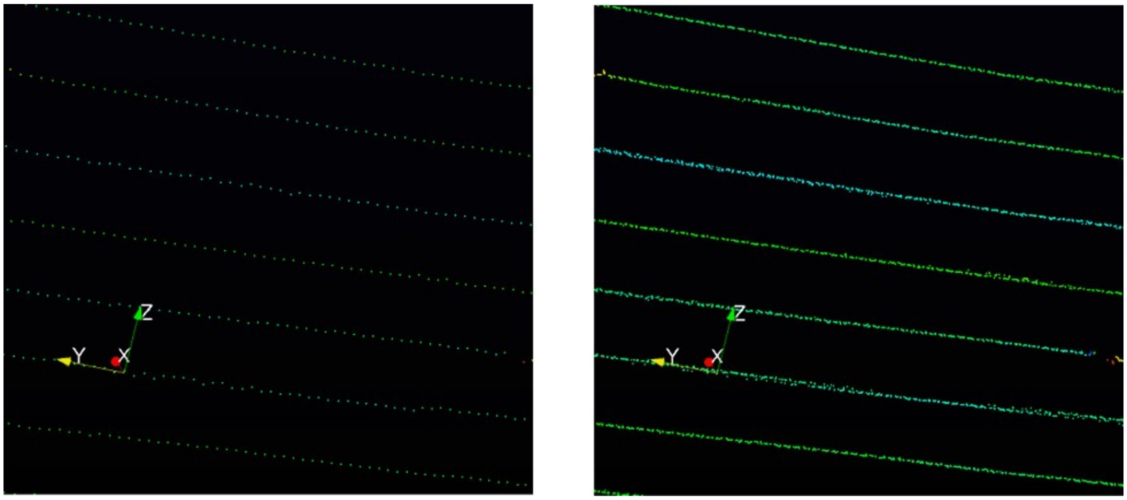
RPM	Resolution
300	0.099495°
600	0.19899°
900	0.298485°
1200	0.39798°

8.3.2 Rotation Speed Fluctuation and Point Density

The sensor uses a feedback control function to maintain its rotational speed within +/-0.5% RPM of its configured setting. This small variation in speed produces a small change in the azimuthal gaps with every revolution. Yet *the sensor fires lasers based on time, not azimuth*. Consequently, over time, the sensor naturally "fills in the gaps" between prior laser firings.

A data set from a stationary sensor illustrates the effect of this strategy in *Figure 8-1 on the facing page*. On the left is a single frame of data. On the right is the same frame and the nine preceding frames overlaid on each other.

Figure 8-1 Point Density Example



Chapter 9 • Sensor Data

This chapter provides detailed information about sensor data characteristics.

As of firmware 5.2.3, the Alpha Prime supports two packet formats: *Legacy Packet Format (HDL) on the facing page*, and *Alpha Prime Advanced Packet Format (APF) on page 74*. You may select on the sensor's web interface which one you want the sensor to generate (see *Configuration Screen on page 86*). APF supports distances up to 300 m while HDL is limited to 262.14 m.

9.1 Sensor Origin and Frame of Reference

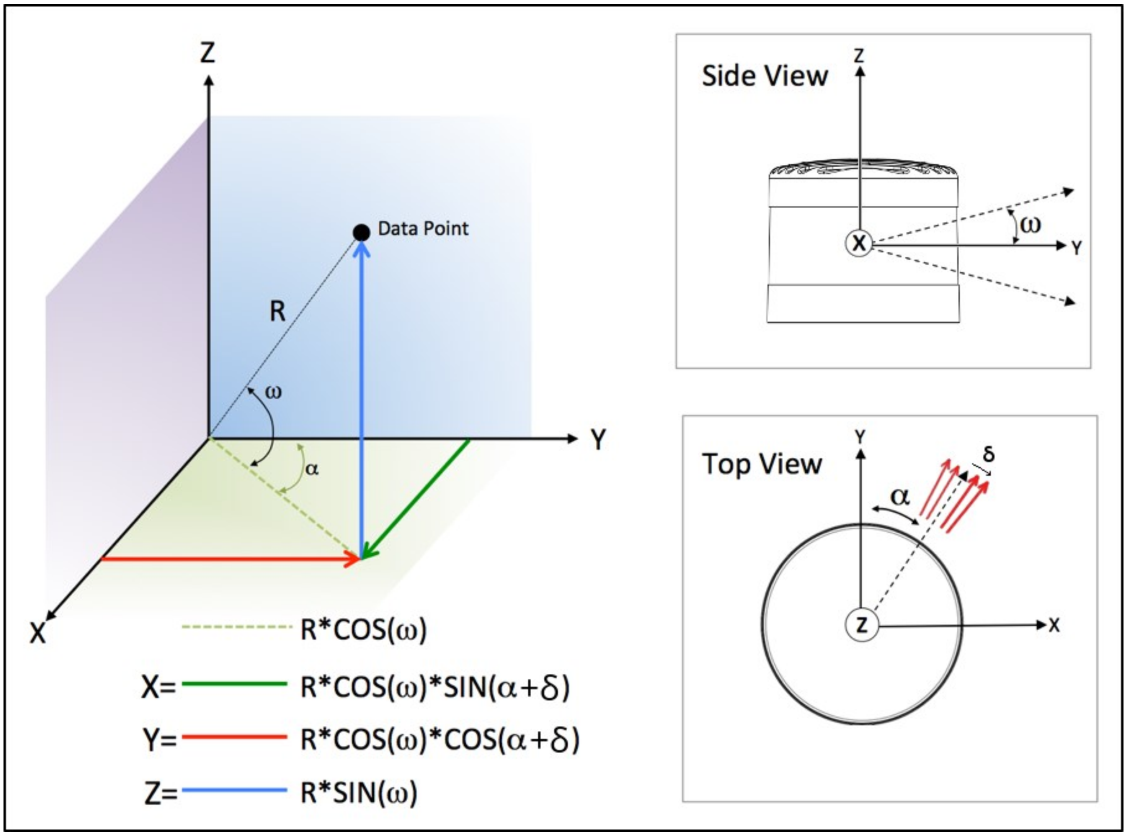
The sensor reports distances relative to its origin in spherical coordinates (radius r, elevation ω, azimuth α).

Sensor data origin (0,0,0) is 66.11 mm (2.603 inches) above the sensor base, on the center axis, as shown in *Figure 9-1 on the facing page* (see the side and top views), which also shows the sensor's frame of reference. See also *VLS-128 Mechanical Drawing on page 128*.

9.2 Calculating X,Y,Z Coordinates from Collected Spherical Data

A computation is necessary to convert the spherical data (radius r, elevation ω, azimuth α) from the sensor to Cartesian coordinates. *Figure 9-1 on the facing page* lists the formulas for converting spherical coordinates (R, ω, α) to Cartesian coordinates (X, Y, Z).

Figure 9-1 Alpha Prime Sensor Coordinate System



9.3 Legacy Packet Format (HDL)

This packet format dates back to the earliest VLP-16 and HDL-32E sensors. It is used by VLP-32C and VLS-128 sensors, as well. It is characterized by a fixed format. It is restricted to 16-bits for raw Distance values.

There are two types of legacy packets generated by the sensor: **Data packets** and **Position packets**. Position packets are sometimes referred to as telemetry packets, or GPS packets.

Data packets contain the 3D data measured by the sensor as well as the calibrated reflectivity of the surface from which the light pulse was returned. Also contained in the data packet is a set of azimuths and a 4-byte timestamp, as well as two factory bytes identifying the model of sensor and the laser return mode. Knowing the model and return mode provides the information your software needs to automatically adjust to different data formats.

Position packets provide a copy of the last NMEA message received (either GPRMC or GPGGA) if you've configured your sensor to synchronize with a GPS time source. See *GPS, Pulse Per Second (PPS) and NMEA Sentence on page 45* for additional information. Position packets also provide a byte identifying the state of the PPS signal for synchronizing with a time source. Newer firmware provides even more information.

Note: In both types of packets, multi-byte fields (e.g. azimuth, distance, and timestamp, but not flag) are transmitted with the least significant byte first (i.e. little-endian).

9.3.1 Definitions

The following sections provide explanations of sensor data packet constructs.

9.3.1.1 Firing Sequence

A firing sequence occurs when all the lasers in a sensor are fired. They are fired in a sequence specific to a given product line or model. Laser recharge time is included. A firing sequence is not allowed to span multiple data packets.

On average, it takes 58.5688 μs to fire all 128 lasers in a Alpha Prime and recharge. Alpha Prime sensors fire their lasers in groups of eight. See *Discreet Point Timing Calculation (HDL) on page 67* for details on firing variability.

9.3.1.2 Laser Channel

A laser channel is a single 905 nm laser emitter and detector pair. Each laser channel is fixed at a particular elevation angle relative to the horizontal plane of the sensor. Each laser channel is given its own Laser ID number. Since the elevation angle of a particular laser channel doesn't change, it doesn't appear in data packets. Its value is inferred by a data point's location in a data packet.

In Alpha Prime sensors, no laser channels are inline (vertically) with the Azimuth in a firing sequence. Instead, each channel is offset by one of eight different azimuthal offsets. See *Chapter 9 • on page 58* for more.

9.3.1.3 Data Point

A data point is a measurement by one laser channel of a reflection of a laser pulse.

A data point is represented in the packet by three bytes - two bytes of distance and one byte of calibrated reflectivity. The distance is an unsigned integer. It has 4 mm granularity. Hence, a reported value of 25,154 represents 100,616 mm or 100.616 m. Calibrated reflectivity is reported on a scale of 0 to 255 as described in *Calibrated Reflectivity on page 36*. The elevation angle (ω) is inferred based on the position of the data point within a data block.

A distance of 0 indicates a non-measurement. The laser is either off or a measurable reflection was not returned in time.

9.3.1.4 Azimuth

A two-byte azimuth value (α) appears after the flag bytes at the beginning of each data block. The azimuth is an unsigned integer. It represents an angle in hundredths of a degree. Therefore, a raw value of 27742 should be interpreted as 277.42°.

Valid values for azimuth range from 0 to 35999. Only one azimuth value is reported per data block.

9.3.1.5 Data Block

The information from one firing sequence of 128 lasers is contained in four consecutive data blocks in Single Return mode, or eight consecutive data blocks in Dual Return mode. Each data packet may contain up to three single-return firing sequences.

Only one Azimuth is returned per data block. One azimuth may be repeated across successive data blocks for a given firing sequence since only one azimuth is determined per firing sequence.

A data block consists of 100 bytes of binary data:

- A two-byte flag (either 0xFFEE, 0xFFDD, 0xFFCC, or 0xFFBB)
- A two-byte Azimuth
- 32 Data Points

[2 + 2 + (32 × 3)] = 100 bytes

9.3.1.6 Frame

A frame of data contains all data points in a single rotation, or a portion thereof beginning at Start FOV and ending at End FOV. It may contain a few extra returns due to inclusion in a firing sequence that straddles a rotation boundary.

9.3.1.7 Time Stamp

The four-byte time stamp is a 32-bit unsigned integer marking a moment within the third firing group of the data packet's first firing sequence, as shown in *Figure 9-8 on page 69*. The time stamp's value is the number of microseconds elapsed since the top of the hour. The number ranges from 0 to 3,599,999,999, the number of microseconds in one hour.

The time stamp is critical because it's used by geo-referencing software to match each laser firing with corresponding data from an inertial navigation system. The inertial navigation system provides a series of time stamped values for pitch, roll, yaw, latitude, longitude, and elevation. By matching the time of the data point to the time-stamped data from the INS, the user's software can mathematically transform the data from the sensor's coordinate frame to an earth-based reference frame. The time stamps are matched to Universal Coordinated Time (UTC) provided by the GPS/INS.

When the sensor powers up it begins counting microseconds using an internal time reference. However, the sensor can synchronize its data with UTC time so you can ascertain the exact firing time of each laser in any particular packet.

UTC synchronization requires a user-supplied GPS/INS receiver generating a synchronizing Pulse Per Second (PPS) signal and an NMEA GPRMC message. The GPRMC message provides minutes and seconds in UTC. Upon synchronization, the sensor reads the minutes and seconds from the GPRMC message and uses the information to set the sensor's time stamp to the number of microseconds past the hour, per UTC.

Note: A full description of electrical and timing requirements can be found in *GPS, Pulse Per Second (PPS) and NMEA Sentence on page 45*. A full description of timing options can be found in *Time Synchronization on page 155*.

9.3.1.8 Factory Bytes

Every data packet includes a pair of bytes called the Factory Bytes. Their values indicate how azimuths and data points are organized in the packet. Their packet locations, values, and meanings are specified in *Table 9-1 below*.

The Return Mode byte indicates how the packet's azimuth and data points are organized. See *Legacy (HDL) Data Packet Structure on the next page* for details.

Every sensor model line has its lasers arrayed vertically at slightly different angles. Use the Product ID byte to identify the correct set of vertical (or elevation) angles. Product IDs are not unique and may be shared by different sensors. For example, per *Table 9-1 below*, the VLP-16 and Puck LITE share the same elevation angles. Hence, the two products share the same Product ID. Conversely, the Puck Hi-Res has a different Product ID since it has a different set of elevation angles.

Table 9-1 Factory Byte Values

Return Mode		Product ID	
Offset in packet: 0x04DE		Offset in packet: 0x04DF	
Mode	**Value**	**Product Model**	**Value**
Strongest	0x37 (55)	HDL-32E	0x21 (33)
Last Return	0x38 (56)	VLP-16 (Puck) Puck LITE	0x22 (34)
Dual Return	0x39 (57)	Puck Hi-Res	0x24 (36)

Return Mode			Product ID	
Offset in packet: 0x04DE			Offset in packet: 0x04DF	
Mode	**Value**		**Product Model**	**Value**
Dual Return with Confidence	0x3B (59)		VLP-32C VLP-32MR	0x28 (40)
--	--		VLS-128	0xA1 (161)

9.3.2 Legacy (HDL) Data Packet Structure

A data packet is 1248 bytes long. It consists of a 42-byte protocol header, twelve contiguous Data Blocks, a four-byte timestamp, and two factory bytes, in that order. It is transmitted via UDP on port 2368 by default.

There are three formats for the data packet:

- Single Return Mode (either Strongest or Last)
- Dual Return Mode
- Dual Return Mode + Confidence

See *Laser Return Modes on page 36* for an illustration of what Strongest, Last, and Dual mean in this context.

See *Dual + Confidence Return Mode on page 64* for additional details on Dual Return Mode.

Figure 9-2 VLS-128 Single Return Mode Data Structure

1248 Bytes			
UDP HEADER	Data Blocks (1200 Bytes)	TimeStamp	Factory Bytes
42 Bytes		4 Bytes	2 Bytes

Data Blocks	1	2	3	4	5	6	7	8	9	10	11	12
	Firing Sequence		ONE		Firing Sequence		TWO		Firing Sequence		THREE	
	0xFFEE	*0xFFDD*	*0xFFCC*	*0xFFBB*	*0xFFEE*	*0xFFDD*	*0xFFCC*	*0xFFBB*	*0xFFEE*	*0xFFDD*	*0xFFCC*	*0xFFBB*
	Azimuth1	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth2*	*Azimuth2*	*Azimuth2*	*Azimuth2*	*Azimuth3*	*Azimuth3*	*Azimuth3*	*Azimuth3*
Channel Number (3 bytes per channel)	0	32	64	96	0	32	64	96	0	32	64	96
	1	33	65	97	1	33	65	97	1	33	65	97
	2	34	66	98	2	34	66	98	2	34	66	98
	3	35	67	99	3	35	67	99	3	35	67	99
	4	36	68	100	4	36	68	100	4	36	68	100
	5	37	69	101	5	37	69	101	5	37	69	101
	6	38	70	102	6	38	70	102	6	38	70	102
	7	39	71	103	7	39	71	103	7	39	71	103
	8	40	72	104	8	40	72	104	8	40	72	104
	9	41	73	105	9	41	73	105	9	41	73	105
	10	42	74	106	10	42	74	106	10	42	74	106
	11	43	75	107	11	43	75	107	11	43	75	107
	12	44	76	108	12	44	76	108	12	44	76	108
	13	45	77	109	13	45	77	109	13	45	77	109
	14	46	78	110	14	46	78	110	14	46	78	110
	15	47	79	111	15	47	79	111	15	47	79	111
	16	48	80	112	16	48	80	112	16	48	80	112
	17	49	81	113	17	49	81	113	17	49	81	113
	18	50	82	114	18	50	82	114	18	50	82	114
	19	51	83	115	19	51	83	115	19	51	83	115
	20	52	84	116	20	52	84	116	20	52	84	116
	21	53	85	117	21	53	85	117	21	53	85	117
	22	54	86	118	22	54	86	118	22	54	86	118
	23	55	87	119	23	55	87	119	23	55	87	119
	24	56	88	120	24	56	88	120	24	56	88	120
	25	57	89	121	25	57	89	121	25	57	89	121
	26	58	90	122	26	58	90	122	26	58	90	122
	27	59	91	123	27	59	91	123	27	59	91	123
	28	60	92	124	28	60	92	124	28	60	92	124
	29	61	93	125	29	61	93	125	29	61	93	125
	30	62	94	126	30	62	94	126	30	62	94	126
	31	63	95	127	31	63	95	127	31	63	95	127

Figure 9-3 VLS-128 Dual Return Mode Data Structure

1248 Bytes

UDP HEADER	1200 Bytes: Data Blocks												TimeStamp	Factory Bytes
	1	2	3	4	5	6	7	8	9	10	11	12		
42 Bytes	Firing Sequence		ONE		Firing Sequence		ONE		Firing Sequence		DUMMY		4 Bytes	2 Bytes
	0xFFEE	*0xFFEE*	*0xFFDD*	*0xFFDD*	*0xFFCC*	*0xFFCC*	*0xFFBB*	*0xFFBB*	*0xFFFF*	*0xFFFF*	*0xFFFF*	*0xFFFF*		
	Azimuth1	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*		
Channel Number (3 bytes per channel)	0	0	32	32	64	64	96	96	0	0	0	0		
	1	1	33	33	65	65	97	97	0	0	0	0		
	2	2	34	34	66	66	98	98	0	0	0	0		
	3	3	35	35	67	67	99	99	0	0	0	0		
	4	4	36	36	68	68	100	100	0	0	0	0		
	5	5	37	37	69	69	101	101	0	0	0	0		
	6	6	38	38	70	70	102	102	0	0	0	0		
	7	7	39	39	71	71	103	103	0	0	0	0		
	8	8	40	40	72	72	104	104	0	0	0	0		
	9	9	41	41	73	73	105	105	0	0	0	0		
	10	10	42	42	74	74	106	106	0	0	0	0		
	11	11	43	43	75	75	107	107	0	0	0	0		
	12	12	44	44	76	76	108	108	0	0	0	0		
	13	13	45	45	77	77	109	109	0	0	0	0		
	14	14	46	46	78	78	110	110	0	0	0	0		
	15	15	47	47	79	79	111	111	0	0	0	0		
	16	16	48	48	80	80	112	112	0	0	0	0		
	17	17	49	49	81	81	113	113	0	0	0	0		
	18	18	50	50	82	82	114	114	0	0	0	0		
	19	19	51	51	83	83	115	115	0	0	0	0		
	20	20	52	52	84	84	116	116	0	0	0	0		
	21	21	53	53	85	85	117	117	0	0	0	0		
	22	22	54	54	86	86	118	118	0	0	0	0		
	23	23	55	55	87	87	119	119	0	0	0	0		
	24	24	56	56	88	88	120	120	0	0	0	0		
	25	25	57	57	89	89	121	121	0	0	0	0		
	26	26	58	58	90	90	122	122	0	0	0	0		
	27	27	59	59	91	91	123	123	0	0	0	0		
	28	28	60	60	92	92	124	124	0	0	0	0		
	29	29	61	61	93	93	125	125	0	0	0	0		
	30	30	62	62	94	94	126	126	0	0	0	0		
	31	31	63	63	95	95	127	127	0	0	0	0		

Figure 9-4 VLS-128 Dual + Confidence Return Mode Data Structure

1248 Bytes

UDP HEADER	1200 Bytes: Data Blocks												TimeStamp	Factory Bytes
	1	2	3	4	5	6	7	8	9	10	11	12		
42 Bytes	Firing Sequence										ONE		4 Bytes	2 Bytes
	0xFFEE	*0xFFEE*	*0xFFEE*	*0xFFDD*	*0xFFDD*	*0xFFDD*	*0xFFCC*	*0xFFCC*	*0xFFCC*	*0xFFBB*	*0xFFBB*	*0xFFBB*		
	Azimuth1	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*	*Azimuth1*		
Channel Number (3 bytes per channel)	0	0	Confidence (3 bytes per laser fire)	32	32	Confidence (3 bytes per laser fire)	64	64	Confidence (3 bytes per laser fire)	96	96	Confidence (3 bytes per laser fire)		
	1	1		33	33		65	65		97	97			
	2	2		34	34		66	66		98	98			
	3	3		35	35		67	67		99	99			
	4	4		36	36		68	68		100	100			
	5	5		37	37		69	69		101	101			
	6	6		38	38		70	70		102	102			
	7	7		39	39		71	71		103	103			
	8	8		40	40		72	72		104	104			
	9	9		41	41		73	73		105	105			
	10	10		42	42		74	74		106	106			
	11	11		43	43		75	75		107	107			
	12	12		44	44		76	76		108	108			
	13	13		45	45		77	77		109	109			
	14	14		46	46		78	78		110	110			
	15	15		47	47		79	79		111	111			
	16	16		48	48		80	80		112	112			
	17	17		49	49		81	81		113	113			
	18	18		50	50		82	82		114	114			
	19	19		51	51		83	83		115	115			
	20	20		52	52		84	84		116	116			
	21	21		53	53		85	85		117	117			
	22	22		54	54		86	86		118	118			
	23	23		55	55		87	87		119	119			
	24	24		56	56		88	88		120	120			
	25	25		57	57		89	89		121	121			
	26	26		58	58		90	90		122	122			
	27	27		59	59		91	91		123	123			
	28	28		60	60		92	92		124	124			
	29	29		61	61		93	93		125	125			
	30	30		62	62		94	94		126	126			
	31	31		63	63		95	95		127	127			

9.3.2.1 Dual + Confidence Return Mode

When Dual + Confidence return mode is in effect, each laser pulse may report up to two returns, as with normal Dual Return mode. In addition, a 3-byte per-laser-fire Return Confidence data structure is also provided with information about both of the returns, as shown in *Figure 9-5 below* and listed in *Figure 9-4 on the previous page* (grey columns).

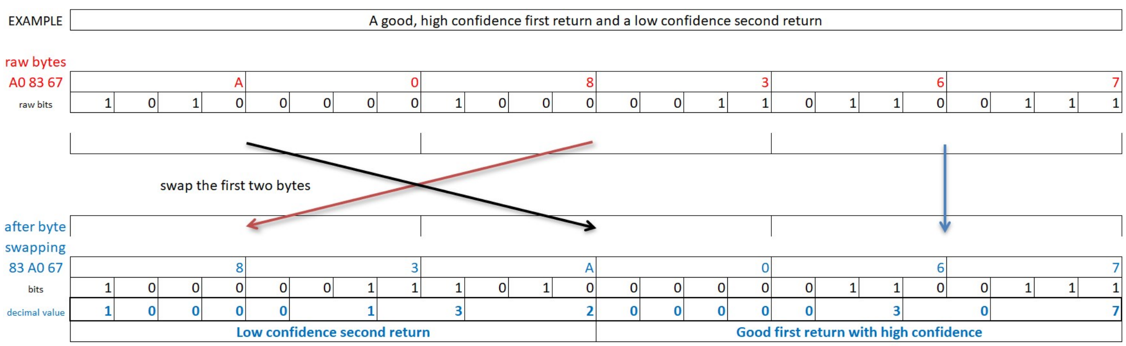
The Dual + Confidence return mode provides Strongest and Last Returns only. **In this mode the sensor does not filter out points, unlike in Dual Return Mode.** Instead, the sensor provides supplementary information in the form of Confidence information about the returns. The onus is then on the consumer to evaluate the returns in light of this Confidence information and decide whether to drop such returns or keep them.

Base confidence is determined by the Signal to Noise Ratio (SNR) of the return and other factors internal to the sensor. Additional bit fields in the data structure give an indication of the presence of sunlight, detection of retro-reflective ghosting or shadowing, probability of interference, a recommendation to drop the return based on confidence, etc.

Information about the Second Return occupies the high 12 bits, while information about the First Return occupies the low 12 bits. The structures of both 12-bit portions are identical. Each field in each structure is an unsigned integer, to be interpreted per current, common computing standards, i.e. with the high order bit of multi-bit fields as their most significant bit, respectively.

Before using any of the bit-fields, *the first two raw bytes must be swapped.* Follow the example in the figure below.

Figure 9-5 VLS-128 Return Confidence Data Structure



3-Byte Per-Fire Return Confidence Data Structure

	Second Return Confidence												First Return Confidence											
Bit	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						msb	lsb	msb	lsb	msb		lsb						msb	lsb	msb	lsb	msb		lsb
	Drop	*reserved*	Retro Shadow	Range Limited	Retro Ghost	Interference		Sun level		Confidence			Drop	*reserved*	Retro Shadow	Range Limited	Retro Ghost	Interference		Sun level		Confidence		

Once the byte-swap is complete, the bit-fields are organized as shown in the Return Confidence data structure in *Figure 9-5 above*. Sample decimal field values -- ready for use -- are indicated in blue numbers at the bottom of the figure's example, which illustrates how to process the raw bytes to produce usable data.

For any laser returns (remember, there can be up to two) that result in a non-measurement (i.e. Distance is 0), the corresponding Return Confidence will not have useful information. Ignore it.

Descriptions of each of the fields are listed in *Figure 9-6 on the facing page*.

Figure 9-6 VLS-128 Return Confidence Bit-field Definitions

Field Name	# bits	Values	Threshold for Drop	Definition	Persistence of Noise
SNR/ Confidence	3	0-3 = Low Confidence 4-5 = Medium Confidence 6-7 = High Confidence	N/A — No direct action taken based on Confidence metric	Confidence determined by SNR and other factors internal to sensor	Applies to all points
Sun Level	2	0 = No Sun Detection 1-3 = Range of Sun Detection	N/A — No direct action taken based on Sun Level	Detection of Sun Saturation	N/A
Interference	2	**0 = High Probability Interferer** **1 = Medium Probability Interferer** **2 = Low Probability Interferer** 3 = High Probability Non-interferer (Authentic Return)	**<= 2 will assert Drop**	VLI Interference Mitigation detection	May impact both persistent and random noise
Retro Ghost	1	0 = Not Detected as False Return **1 = Detected as False Return**	**== 1 will assert Drop**	False Returns associated with Retro firing transitions	Noise is persistent
Range Limited	1	0 = Return Not Limited 1 = Return Limited to 16m	N/A — Informative only	Indicates all returns associated with Firing event are limited to 16m (limiting detection of a Last return)	N/A
Retro Shadow	1	0 = Not in Retro Shadow **1 = In Shadow of Retro**	**== 1 will assert Drop**	False Returns associated with Retro firing transitions in Retro's Shadow	Noise is persistent
Drop	1	0 = No Drop, Keep the Return **1 = Recommend Drop**	***1 is Drop***	**Global recommendation for "Dropping" the return based on certain confidence information**	N/A

Text in Bold is associated with a Drop indication.

If Drop is asserted (i.e. the Drop bit is 1), the sensor *would have* dropped the return (i.e. its Distance would be set to 0) had the sensor been in ordinary Dual Return mode. However, in Dual + Confidence return mode, if Drop is asserted *the Distance reading is retained* and associated Confidence info (12 bits) gives you an idea why the return would otherwise have been dropped. It is then up to you, the consumer, to ignore the return or not.

9.3.3 Legacy Position Packet Structure

The role of the Position Packet (often called the Telemetry Packet) is to provide a copy of the most recent, supported NMEA sentence received from an external GPS/INS/IMU source as well as the Pulse Per Second status, plus a time stamp representing when the position packet was assembled, and possibly other related info. If no GPS/INS/IMU is attached or it is disabled, the NMEA sentence, PPS status, and related fields in the position packet will be empty (i.e. all zeros).

The position packet is a 554 byte UDP packet received on port 8308 (by default). Protocol headers account for the first 42 bytes. Payload length is 512 bytes. The structure of the position packet (minus the 42-byte protocol header) is given in *Table 9-2 below*.

Note: You may notice the timestamps in the position packets are occasionally out of order with respect to the data packets. This is normal as the delivery of data packets is the sensor's highest priority, and a position packet may get deferred momentarily in favor of transmitting a data packet. See *NMEA Message Formats on page 51* for details on the NMEA messages supported.

Table 9-2 Position Packet Structure Field Offsets

Description	Number of Bytes	Address Offset	Data Range
Reserved	187	0x00-0xBA	unused (null bytes)

Description	Number of Bytes	Address Offset	Data Range
Temperature of top board	1	0xBB	-128 to +127°C
Temperature of bottom board	1	0xBC	-128 to +127°C
Reserved	9	0xBD-0xC5	unused (null bytes)
µsec since top of the hour (TOH)	4	0xC6-0xC9	Number of microseconds elapsed since the top of the hour
Pulse Per Second (PPS) status	1	0xCA	0: Absent 1: Synchronizing 2: Locked 3: Error
Thermal status	1	0xCB	0: Ok 1: Thermal shutdown
Last shutdown temperature	1	0xCC	Temperature of unit when thermal shutdown occurred (-128 to +127°C)
Temperature of unit at power up	1	0xCD	Temperature of unit (bottom board) at power up (-128 to +127°C)
NMEA sentence	128	0xCE-0x14D	GPRMC or GPGGA sentence
Reserved	178	0x14E-0x1FF	unused (null bytes)

The GPRMC sentence is terminated with CR/LF and padded to end of payload with null bytes.

Table 9-3 PPS Status Byte Values

Value	Description
0	No PPS detected
1	Synchronizing to PPS
2	PPS Locked
3	Error

A position packet is shown in the figure below. Packet offsets are on the left, raw data bytes in hex are in the center, and the ASCII interpretation is on the right.

Figure 9-7 Wireshark Position Packet Trace

```
0000  ff ff ff ff ff ff 60 76  88 10 2b 9b 08 00 45 00   ......`v ..+...E.
0010  02 1c a2 0a 00 00 40 11  14 56 c0 a8 01 c9 ff ff   ......@. .V......
0020  ff ff 20 74 20 74 02 08  98 50 00 00 00 00 00 00   .. t t.. .P......
0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
00f0  df f5 e8 d5 02 00 00 00  24 47 50 52 4d 43 2c 32   ........ $GPRMC,2
0100  30 35 39 34 38 2c 41 2c  33 37 31 36 2e 36 36 39   05948,A, 3716.669
0110  34 2c 4e 2c 31 32 31 35  33 2e 34 35 35 30 2c 57   4,N,1215 3.4550,W
0120  2c 30 30 30 2e 30 2c 30  37 38 2e 34 2c 32 36 30   ,000.0,0 78.4,260
0130  37 31 35 2c 30 31 33 2e  39 2c 45 2c 44 2a 30 37   715,013. 9,E,D*07
0140  0d 0a 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0170  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0200  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0210  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0220  00 00 00 00 00 00 00 00  00 00                     ........ ..
```

Legend:

- light blue: Ethernet+IP+UDP packet header (42 bytes)
- orange: Timestamp (4 bytes)
- red: PPS status (1 byte)
- green: NMEA GPRMC sentence (length varies)

9.3.4 Discreet Point Timing Calculation (HDL)

For the purpose of geo-referencing lidar data, particularly on a moving platform, it is helpful to calculate the exact moment of a particular laser firing in order to more closely associate it with time stamped data from a GPS, INS, or other source of location information. By using the firing timing information given in *Figure 9-8 on page 69* you will be able to compute the exact firing time for each data point within a data packet.

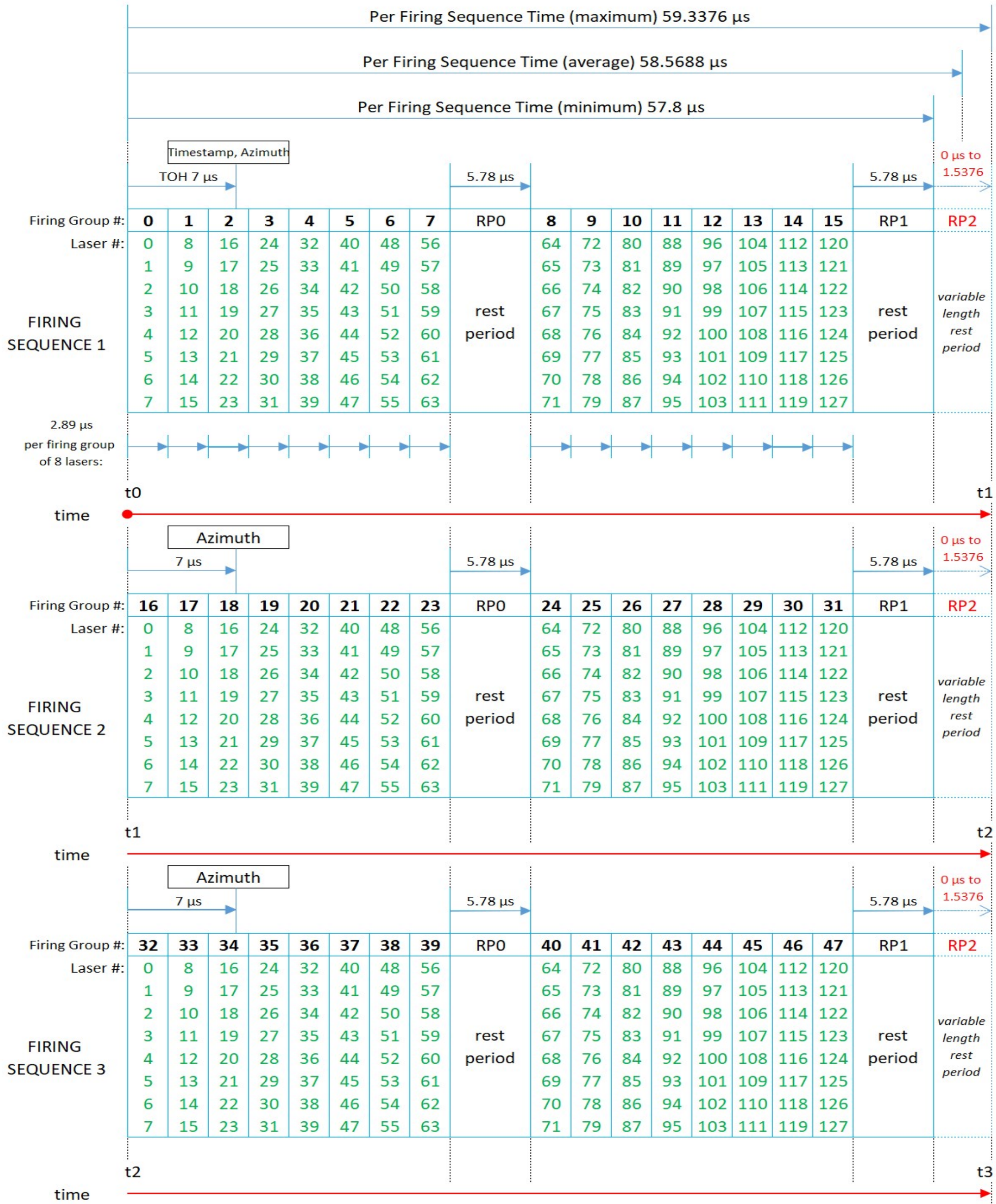
Changes to firing timing have occurred to the Alpha Prime product line as the product evolved. The changes were introduced in specific firmware releases. Here is a brief guide:

- Firmware prior to 5.1.7.1 performed firing sequences every 53.3 µs like clockwork. See Rev 2 of this manual for details on that firing pattern. Rev 2 is the final manual for Alpha Puck sensors.
- Firmware version 5.1.7.1 introduced variable firing sequence timing. The purpose of this variance is to mitigate sensor to sensor interference. It remained the same up to and including firmware 5.2.1. See Rev 3 of this manual for details.
- Firmware version 5.2.3 lengthened the time each firing group takes to fire to accommodate detections up to 300 meters away. (Prior versions were limited to 250 meter readings, so they didn't have to "listen" quite as long.) As shown in the figure below, the bulk of a firing sequence takes, on average, 58.5688 µs to fire all 128 lasers. That includes two fixed-duration rest periods (RP0 and RP1) and one variable-duration rest period (RP2), which may range from 0 µs up to 1.5376 µs. The average RP2 duration is 0.7688 µs. Note that this firing timing is the same in both HDL and APF mode.

7 µs into the first firing sequence in a data packet, the packet's Timestamp is determined. Timing offsets for the firing groups before (and including) that point in time are negative, and positive for those that follow (until the next data packet). TOH is synched to the timestamp.

For each firing sequence, the Azimuth that appears in the sequence's data blocks is fixed at similar points 7 µs into the sequence.

Figure 9-8 Firing Sequence Timing



NOTE:

- Groups of eight lasers fire at a time (laser numbers are in green).
- Everything red may vary in duration.
- Each RP2 rest period is independent of the others.
- Time span from t0 to t3 may range from 173.4 µs minimum to 178.0128 µs maximum.
- Average time span t0 to t3: 175.7064 µs.

Alpha Prime lasers are fired in groups of eight at widely separated angles to minimize IR self-crosstalk. The sequence of firings is specified in *Figure 9-8 on the previous page*.

To calculate the exact time, in microseconds, of each data point in the **first firing sequence** in a data packet, look up the corresponding timing offset for each data point in the appropriate figure below. (Note that RP0 is embedded in the timing offsets listed, but not RP1 or RP2. As shown in *Figure 9-8 on the previous page*, RP1 and RP2 occur *after* the first firing sequence.) Add the timing offset value from the table to the data packet's Timestamp to determine the exact time of firing relative to TOH.

For single return mode only, to calculate a similar time for any given point in the **second firing sequence** (in blue), perform a similar lookup, *then determine when the equivalent TOH occurred for that firing sequence, and add the two*. Similarly for the **third firing sequence** (in red). An advanced method for inferring TOH would be to use the azimuth difference between the firing sequences to help estimate RP2. This may have an error of around a few percent.

Figure 9-9 Single Return Mode Timing Offsets (in µs)

	Data Blocks											
Order in Packet	1	2	3	4	5	6	7	8	9	10	11	12
1	-7.00000	4.56000	21.90000	33.46000	51.56880	63.12880	80.46880	92.02880	110.13760	121.69760	139.03760	150.59760
2	-7.00000	4.56000	21.90000	33.46000	51.56880	63.12880	80.46880	92.02880	110.13760	121.69760	139.03760	150.59760
3	-7.00000	4.56000	21.90000	33.46000	51.56880	63.12880	80.46880	92.02880	110.13760	121.69760	139.03760	150.59760
4	-7.00000	4.56000	21.90000	33.46000	51.56880	63.12880	80.46880	92.02880	110.13760	121.69760	139.03760	150.59760
5	-7.00000	4.56000	21.90000	33.46000	51.56880	63.12880	80.46880	92.02880	110.13760	121.69760	139.03760	150.59760
6	-7.00000	4.56000	21.90000	33.46000	51.56880	63.12880	80.46880	92.02880	110.13760	121.69760	139.03760	150.59760
7	-7.00000	4.56000	21.90000	33.46000	51.56880	63.12880	80.46880	92.02880	110.13760	121.69760	139.03760	150.59760
8	-7.00000	4.56000	21.90000	33.46000	51.56880	63.12880	80.46880	92.02880	110.13760	121.69760	139.03760	150.59760
9	-4.11000	7.45000	24.79000	36.35000	54.45880	66.01880	83.35880	94.91880	113.02760	124.58760	141.92760	153.48760
10	-4.11000	7.45000	24.79000	36.35000	54.45880	66.01880	83.35880	94.91880	113.02760	124.58760	141.92760	153.48760
11	-4.11000	7.45000	24.79000	36.35000	54.45880	66.01880	83.35880	94.91880	113.02760	124.58760	141.92760	153.48760
12	-4.11000	7.45000	24.79000	36.35000	54.45880	66.01880	83.35880	94.91880	113.02760	124.58760	141.92760	153.48760
13	-4.11000	7.45000	24.79000	36.35000	54.45880	66.01880	83.35880	94.91880	113.02760	124.58760	141.92760	153.48760
14	-4.11000	7.45000	24.79000	36.35000	54.45880	66.01880	83.35880	94.91880	113.02760	124.58760	141.92760	153.48760
15	-4.11000	7.45000	24.79000	36.35000	54.45880	66.01880	83.35880	94.91880	113.02760	124.58760	141.92760	153.48760
16	-4.11000	7.45000	24.79000	36.35000	54.45880	66.01880	83.35880	94.91880	113.02760	124.58760	141.92760	153.48760
17	-1.22000	10.34000	27.68000	39.24000	57.34880	68.90880	86.24880	97.80880	115.91760	127.47760	144.81760	156.37760
18	-1.22000	10.34000	27.68000	39.24000	57.34880	68.90880	86.24880	97.80880	115.91760	127.47760	144.81760	156.37760
19	-1.22000	10.34000	27.68000	39.24000	57.34880	68.90880	86.24880	97.80880	115.91760	127.47760	144.81760	156.37760
20	-1.22000	10.34000	27.68000	39.24000	57.34880	68.90880	86.24880	97.80880	115.91760	127.47760	144.81760	156.37760
21	-1.22000	10.34000	27.68000	39.24000	57.34880	68.90880	86.24880	97.80880	115.91760	127.47760	144.81760	156.37760
22	-1.22000	10.34000	27.68000	39.24000	57.34880	68.90880	86.24880	97.80880	115.91760	127.47760	144.81760	156.37760
23	-1.22000	10.34000	27.68000	39.24000	57.34880	68.90880	86.24880	97.80880	115.91760	127.47760	144.81760	156.37760
24	-1.22000	10.34000	27.68000	39.24000	57.34880	68.90880	86.24880	97.80880	115.91760	127.47760	144.81760	156.37760
25	1.67000	13.23000	30.57000	42.13000	60.23880	71.79880	89.13880	100.69880	118.80760	130.36760	147.70760	159.26760
26	1.67000	13.23000	30.57000	42.13000	60.23880	71.79880	89.13880	100.69880	118.80760	130.36760	147.70760	159.26760
27	1.67000	13.23000	30.57000	42.13000	60.23880	71.79880	89.13880	100.69880	118.80760	130.36760	147.70760	159.26760
28	1.67000	13.23000	30.57000	42.13000	60.23880	71.79880	89.13880	100.69880	118.80760	130.36760	147.70760	159.26760
29	1.67000	13.23000	30.57000	42.13000	60.23880	71.79880	89.13880	100.69880	118.80760	130.36760	147.70760	159.26760
30	1.67000	13.23000	30.57000	42.13000	60.23880	71.79880	89.13880	100.69880	118.80760	130.36760	147.70760	159.26760
31	1.67000	13.23000	30.57000	42.13000	60.23880	71.79880	89.13880	100.69880	118.80760	130.36760	147.70760	159.26760
32	1.67000	13.23000	30.57000	42.13000	60.23880	71.79880	89.13880	100.69880	118.80760	130.36760	147.70760	159.26760

Figure 9-10 Dual Return Mode Timing Offsets (in µs)

	Data Blocks											
Order in Packet	1	2	3	4	5	6	7	8	9	10	11	12
1	-7.00000	-7.00000	4.56000	4.56000	21.90000	21.90000	33.46000	33.46000				
2	-7.00000	-7.00000	4.56000	4.56000	21.90000	21.90000	33.46000	33.46000				
3	-7.00000	-7.00000	4.56000	4.56000	21.90000	21.90000	33.46000	33.46000				
4	-7.00000	-7.00000	4.56000	4.56000	21.90000	21.90000	33.46000	33.46000				
5	-7.00000	-7.00000	4.56000	4.56000	21.90000	21.90000	33.46000	33.46000				
6	-7.00000	-7.00000	4.56000	4.56000	21.90000	21.90000	33.46000	33.46000				
7	-7.00000	-7.00000	4.56000	4.56000	21.90000	21.90000	33.46000	33.46000				
8	-7.00000	-7.00000	4.56000	4.56000	21.90000	21.90000	33.46000	33.46000				
9	-4.11000	-4.11000	7.45000	7.45000	24.79000	24.79000	36.35000	36.35000				
10	-4.11000	-4.11000	7.45000	7.45000	24.79000	24.79000	36.35000	36.35000				
11	-4.11000	-4.11000	7.45000	7.45000	24.79000	24.79000	36.35000	36.35000				
12	-4.11000	-4.11000	7.45000	7.45000	24.79000	24.79000	36.35000	36.35000				
13	-4.11000	-4.11000	7.45000	7.45000	24.79000	24.79000	36.35000	36.35000				
14	-4.11000	-4.11000	7.45000	7.45000	24.79000	24.79000	36.35000	36.35000				
15	-4.11000	-4.11000	7.45000	7.45000	24.79000	24.79000	36.35000	36.35000		BLANK		
16	-4.11000	-4.11000	7.45000	7.45000	24.79000	24.79000	36.35000	36.35000				
17	-1.22000	-1.22000	10.34000	10.34000	27.68000	27.68000	39.24000	39.24000				
18	-1.22000	-1.22000	10.34000	10.34000	27.68000	27.68000	39.24000	39.24000				
19	-1.22000	-1.22000	10.34000	10.34000	27.68000	27.68000	39.24000	39.24000				
20	-1.22000	-1.22000	10.34000	10.34000	27.68000	27.68000	39.24000	39.24000				
21	-1.22000	-1.22000	10.34000	10.34000	27.68000	27.68000	39.24000	39.24000				
22	-1.22000	-1.22000	10.34000	10.34000	27.68000	27.68000	39.24000	39.24000				
23	-1.22000	-1.22000	10.34000	10.34000	27.68000	27.68000	39.24000	39.24000				
24	-1.22000	-1.22000	10.34000	10.34000	27.68000	27.68000	39.24000	39.24000				
25	1.67000	1.67000	13.23000	13.23000	30.57000	30.57000	42.13000	42.13000				
26	1.67000	1.67000	13.23000	13.23000	30.57000	30.57000	42.13000	42.13000				
27	1.67000	1.67000	13.23000	13.23000	30.57000	30.57000	42.13000	42.13000				
28	1.67000	1.67000	13.23000	13.23000	30.57000	30.57000	42.13000	42.13000				
29	1.67000	1.67000	13.23000	13.23000	30.57000	30.57000	42.13000	42.13000				
30	1.67000	1.67000	13.23000	13.23000	30.57000	30.57000	42.13000	42.13000				
31	1.67000	1.67000	13.23000	13.23000	30.57000	30.57000	42.13000	42.13000				
32	1.67000	1.67000	13.23000	13.23000	30.57000	30.57000	42.13000	42.13000				

9.3.5 Precision Azimuth Calculation (HDL)

The azimuth (α) reported by the sensor in each data block represents the center line of the firing pattern (see *Figure F-3 on page 151* and *Figure F-4 on page 152*) during the firing of the third firing group (see *Figure 9-8 on page 69*) in a firing sequence. But, because all of the lasers are offset from the center line by fixed amounts, an azimuthal offset (δ, delta) must be applied to determine the precise azimuth of each data point. The vertical angle (Elevation) and azimuth offset (AziOffset) of each laser are shown in *Figure 9-11 on the next page* and *Figure 9-17 on page 83*.

Figure 9-11 on the next page reflects the laser channel arrangement specified in the file, **Alpha Prime.xml**. Each block of 8 lasers is a Firing Group. They fire simultaneously. They correspond to the firing groups seen in *Figure 9-8 on page 69*.

See *XML File on page 84* for details on XML files.

Figure 9-11 VLS-128 Azimuth Offsets by Elevation - HDL

HDL

Ch	AziOffset	Elevation	Ch	AziOffset	Elevation	Ch	AziOffset	Elevation	Ch	AziOffset	Elevation
0	-6.354	-11.742	32	-6.354	-0.340	64	-6.354	-5.620	96	-6.354	1.420
1	-4.548	-1.990	33	-4.548	5.180	65	-4.548	-0.230	97	-4.548	-10.346
2	-2.732	3.400	34	-2.732	-3.640	66	-2.732	5.430	98	-2.732	-1.880
3	-0.911	-5.290	35	-0.911	1.750	67	-0.911	-3.530	99	-0.911	3.510
4	0.911	-0.780	36	0.911	-25.000	68	0.911	0.980	100	0.911	-6.060
5	2.732	4.610	37	2.732	-2.430	69	2.732	-19.582	101	2.732	-0.670
6	4.548	-4.080	38	4.548	2.960	70	4.548	-2.320	102	4.548	4.720
7	6.354	1.310	39	6.354	-5.730	71	6.354	3.070	103	6.354	-3.970
8	-6.354	-6.500	40	-6.354	0.540	72	-6.354	-4.740	104	-6.354	2.300
9	-4.548	-1.110	41	-4.548	9.700	73	-4.548	0.650	105	-4.548	-6.390
10	-2.732	4.280	42	-2.732	-2.760	74	-2.732	11.750	106	-2.732	-1.000
11	-0.911	-4.410	43	-0.911	2.630	75	-0.911	-2.650	107	-0.911	4.390
12	0.911	0.100	44	0.911	-7.650	76	0.911	1.860	108	0.911	-5.180
13	2.732	6.480	45	2.732	-1.550	77	2.732	-7.150	109	2.732	0.210
14	4.548	-3.200	46	4.548	3.840	78	4.548	-1.440	110	4.548	6.980
15	6.354	2.190	47	6.354	-4.850	79	6.354	3.950	111	6.354	-3.090
16	-6.354	-3.860	48	-6.354	3.180	80	-6.354	-2.100	112	-6.354	4.980
17	-4.548	1.530	49	-4.548	-5.510	81	-4.548	3.290	113	-4.548	-3.750
18	-2.732	-9.244	50	-2.732	-0.120	82	-2.732	-5.400	114	-2.732	1.640
19	-0.911	-1.770	51	-0.911	5.730	83	-0.911	-0.010	115	-0.911	-8.352
20	0.911	2.740	52	0.911	-4.300	84	0.911	4.500	116	0.911	-2.540
21	2.732	-5.950	53	2.732	1.090	85	2.732	-4.190	117	2.732	2.850
22	4.548	-0.560	54	4.548	-16.042	86	4.548	1.200	118	4.548	-5.840
23	6.354	4.830	55	6.354	-2.210	87	6.354	-13.565	119	6.354	-0.450
24	-6.354	-2.980	56	-6.354	4.060	88	-6.354	-1.220	120	-6.354	8.430
25	-4.548	2.410	57	-4.548	-4.630	89	-4.548	4.170	121	-4.548	-2.870
26	-2.732	-6.280	58	-2.732	0.760	90	-2.732	-4.520	122	-2.732	2.520
27	-0.911	-0.890	59	-0.911	15.000	91	-0.911	0.870	123	-0.911	-6.170
28	0.911	3.620	60	0.911	-3.420	92	0.911	6.080	124	0.911	-1.660
29	2.732	-5.070	61	2.732	1.970	93	2.732	-3.310	125	2.732	3.730
30	4.548	0.320	62	4.548	-6.850	94	4.548	2.080	126	4.548	-4.960
31	6.354	7.580	63	6.354	-1.330	95	6.354	-6.650	127	6.354	0.430

To get better precision when geo-referencing, it’s useful to precisely calculate the unique azimuth for each point by accounting for the firing timing.

Consider a single data packet with 12 data blocks. In single return mode this fits 3 firing sequences, each sequence occupying four consecutive data blocks. The azimuth at the top of each data block represents the azimuth reported at the moment the third firing group of 8 lasers in the firing sequence was fired. Assuming the rotational speed is constant for the duration of the firing sequence, you can use the algorithm below to estimate a more precise azimuth angle for each point.

The pseudo code below illustrates the concept.

- K represents an index to a data point in the Nth data block, where its valid range is 0 to 31.
- Do this for each data block.
- Single return mode.
- Note: This represents a starting point for implementing the timing in *Figure 9-8 on page 69*.

```
    // Do this for every packet

    // packet_m represents an index to a packet
    // datablock_n represents an index to a data block, valid range is 0 to 11
    // point_k represents an index to a data point in the nth data block, valid range
is 0 to 31

    // Adjust for an azimuth rollover from 359.99° to 0°
    // Note that a firing sequence spans four data blocks
    If (Azimuth[packet_m+1][datablock_0] < Azimuth[packet_m][datablock_0])
    Then
        Azimuth[packet_m+1][datablock_0] := Azimuth[packet_m+1][datablock_0] + 360;
    Endif

    // Adjust for a timestamp rollover from 3,599,999,999 µs to 0 µs
    If (Timestamp[packet_m+1] < Timestamp[packet_m])
    Then
        Timestamp[packet_m+1] := Timestamp[packet_m+1] + 3600000000;
    Endif

    // Determine the azimuth rate
    Azimuth_Gap := Azimuth[packet_m+1][datablock_0] - Azimuth[packet_m][datablock_0];
    Time_Gap := Timestamp[packet_m+1] - Timestamp[packet_m]
    Azimuth_Rate := Azimuth_Gap / Time_Gap;

    // Loop over data blocks
    For (datablock_n = 0 to 11)

        // Loop over data points
        For (point_k = 0 to 31)
            // Determine laser number
            Laser_Number := (datablock_n MOD 4) * 32 + point_k;

            // Lasers are fired in groups of 8
            Firing_Group := Laser_Number DIV 8;

            // Interpolate
            Precision_Azimuth[point_k] := Azimuth[packet_m][datablock_n] + Azimuth_Rate
* (2.89 µs * Firing_Group - 7 µs);

            // Add rotation during RP0
            If (Firing_Group > 7)
            Then
                Precision_Azimuth[point_k] := Precision_Azimuth[point_k] + Azimuth_Rate
* 5.78 µs;
            Endif

            // Apply the azimuth offset
```

```
            Precision_Azimuth[point_k] := Precision_Azimuth[point_k] + azimuth_offset
[Laser_Number];

            // Adjust for any rollover
            If (Precision_Azimuth[point_k] >= 360)
            Then
                Precision_Azimuth[point_k] := Precision_Azimuth[point_k] - 360;
            Endif
        End For
    End For
```

9.4 Alpha Prime Advanced Packet Format (APF)

The Velodyne Advanced Packet Format was developed to break free of the fixed format nature of the Legacy Packet Format (HDL). It is inherently customizable based on what the sensor is designed to achieve, coupled with the various constraints imposed on it.

Support for APF was introduced for the Alpha Prime in the firmware 5.2.3 release.

There is only one type of APF UDP packet supported by the Alpha Prime sensor: Data packets.

The base format for these packets is specified in the **Velodyne Advanced Packet Format Specification**, a facsimile of which appears in *Advanced Packet Format on page 109*. At a high level, the format is variable-length, compact, binary, and flexible. This chapter you are reading contains the Alpha Prime implementation or instance. Other sensors that implement APF in some form document their instance of the format in their user manuals.

IP fragmentation is not supported, nor are jumbo frames. Each packet type is tuned to fit within the standard Ethernet II MTU.

Data packets are streamed by the sensor over Ethernet without acknowledgment. Each has a sequence number to ensure data stream integrity.

Data packets contain the 3D data measured by the sensor as well as the reflectivity of the surface from which the light pulse was returned.

Note: Multi-byte fields (e.g. azimuth, distance, and timestamp) are typically transmitted in network byte order.

9.4.1 Definitions

The following sections provide explanations of sensor data packet constructs.

9.4.1.1 Firing Sequence

A firing sequence occurs when all the lasers in a sensor are fired once each. They are fired in a sequence specific to a given product line or model. Laser recharge time is included. A firing sequence is not allowed to span multiple data packets.

9.4.1.2 Laser Channel

A laser channel is a single 905 nm laser emitter and detector pair.

Each laser channel has a fixed elevation angle relative to the horizontal plane of the sensor. Laser channel scans do not overlap each other.

Each laser channel is given its own Laser ID number. These numbers start at 0.

9.4.1.3 Model Identification Code (MIC)

The Model Identification Code is a 1-byte field in the packet's Payload Header.

The MIC value is 0x00 for Alpha Prime sensors. Its purpose is to distinguish Alpha Prime data from other sensor models.

9.4.1.4 Data Point

A data point is a measurement by one laser channel of a reflection of a laser pulse. The notion includes both distance and reflectivity measurements in one. It does not include direction or time.

9.4.1.5 Calibrated Reflectivity

This field is described in *Calibrated Reflectivity on page 36*. Each data point comes with a calibrated reflectivity value in a Firing Return.

9.4.1.6 Azimuth

The azimuth (α) is the horizontal component of the firing angle with respect to the sensor's own frame of reference. The value given is in hundredths of a degree. For example, a raw value of 12300 should be interpreted as 123.00°.

Valid values for azimuth range from 0° to the sensor's maximum FOV value. Each laser firing group has its own azimuth value.

9.4.1.7 Time Stamp

The data packet time stamp (TREF) is a 64-bit (8-byte) field in the Payload Header. It is in PTP truncated format. It is the value of the sensor's internal clock when the data packet is formulated. The internal clock is independently kept up to date as long as a PTP Master or Grandmaster Clock is available and active. If unavailable, the oscillator may drift an extremely small amount in a full day of operation.

9.4.1.8 Frame

A frame of data includes all laser measurements generated by the sensor in a given rotation. It begins when Azimuth wraps around to 0.

A common technique for detecting the start of frame is to subtract the previous Azimuth from the current one. When it becomes negative, a new frame has begun. This can happen within a data packet.

9.4.2 APF Data Packet Structure

A data packet is the payload of a UDP packet. Its first byte occurs immediately after the UDP header.

9.4.2.1 Data Packet Payload Format

The Data Packet is composed of a Payload Header followed by a variable size Payload.

The Payload is comprised of a variable number of Firing Groups containing Point Cloud Data.

Each Firing Group is comprised of a Firing Group Header followed by a variable number of Firings.

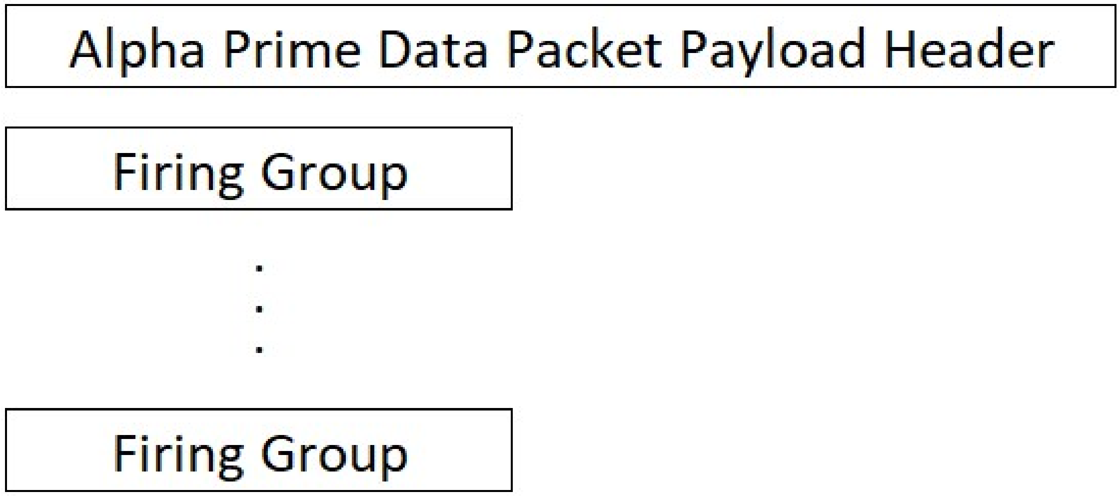
Each Firing is comprised of a Firing Header followed by a variable number of Returns.

Each Return is a packed N-tuple comprised of the 16-bit or 24-bit Distance value followed by the variable number of 8-bit Intensity values.

The Payload Header contains metadata that describes the format and content of Firing Groups, Firings, and Returns in the payload.

The Point Cloud Payload is organized hierarchically as follows:

Figure 9-12 Alpha Prime Data Packet Payload Format



NOTE: A Firing Group may not be split across multiple Payloads.

Extension Headers are not permitted at this time.

Individual scalar fields are unsigned integers.

This Data Packet does not have a Payload Trailer.

9.4.2.2 Alpha Prime Data Packet Payload Header

The Payload Header contains metadata describing the format of the payload as well as stream identification and timing.

The Payload Header extends the APF Nominal Payload Header with additional fields to describe the format of the Point Cloud payload data.

Figure 9-13 Alpha Prime Data Packet Payload Header

Byte Offset	0		1	2		3
00	VER	HLEN	NXHDR	PTYPE	TLEN	MIC
04	PSEQ					
08	TREF					
0C						
10	GLEN	FLEN	DSET	ISET		
...	*Additional data allowed by HLEN (reserved)*					

GLEN

Size (bits): 4 (MSN)

Firing Group Header Length

The number of 32-bit words in a Firing Group Header. The minimum value of this field is 2.

FLEN

Size (bits): 4 (LSN)

Firing Header Length

The number of 32-bit words in a Firing Header. The minimum value of this field is 1.

DSET

Size (bits): 8 (bitmask)

Distance Set

Determines the number and type of distance values in a Return.

DSET is defined as a bit-set as follows:

- Bit 7 - Encoding size. 0: 16-bit distance. 1: 24-bit distance.
- Bit 6 - Format. 0: Mask format, 1: Count format.

In Mask Format the remaining bits are a 'set' defining which returns are included as follows:

- Bit 4:5 - Reserved for additional return types.
- Bit 3 - Last return - Supported.
- Bit 2 - Second strongest return - Supported.
- Bit 1 - Strongest return - Supported.
- Bit 0 - First return - Not Supported.

In Mask format, the order of returns in the Firing is from LSB to MSB of the DSET mask bits [5:0].

In Count format, the order of returns is not specified.

A value of zero indicates that no returns are included after the Firing Header.

ISET

Size (bits): 16

Intensity Set

Determines the number and type of 8-bit intensity (aka reflectivity) values in each Return. ISET is defined as a bit-set as follows:

- Bit 0 - Reflectivity - Supported.
- Bit 1 - Intensity - Reserved.
- Bit 2 - Confidence - Supported.
- All other intensities are reserved / unsupported.

The order of intensities in the Return is from LSB to MSB of the ISET mask.

A value of zero indicates that no intensities are included after the distance value in a Return.

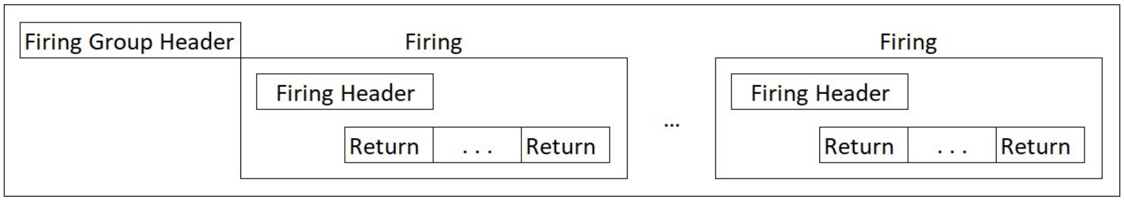
The timestamp (TREF) is essentially set with the first laser firing. It is truncated, 64-bit PTP time. Add the remainder of a full PTP time to determine when the data packet was collected in real world time. This remainder is something you determine during data collection or look up in a packet capture and calculate.

9.4.2.3 Firing Group

Following the Payload Header are a variable number of Firing Groups. The size of each Firing Group is determined by metadata in the Payload Header and Firing Group Header. Each Firing Group is aligned on a 32-bit offset and should be padded as necessary.

A Firing Group consists of a Firing Group Header followed by zero or more Firings.

Figure 9-14 Alpha Prime Firing Group



The number of Firings in a Firing Group is specified by the Firing Group Header FCNT value.

9.4.2.4 Firing Group Header

The Firing Group header contains timing, scan direction, vertical deflection, and azimuth information for the Firing Group:

Figure 9-15 Alpha Prime Firing Group Header

Byte Offset	0	1	2	3
00	TOFFS		FCNT / FSPN	FDLY
04	HDIR / VDIR / VDFL		AZM	

TOFFS

Size (bits): 16

Time Offset

Unsigned time fraction offset from payload timestamp to the firing time of the Firing Group in 64-nanosecond units.

FCNT

Size (bits): 5

Firing Count

(FCNT + 1) is the number of Firings in the Firing Group.

FSPN

Size (bits): 3

Firing Span

(FSPN + 1) is the count (span) of channels fired simultaneously in the Firing Group.

FDLY

Size (bits): 8

Firing Delay

Unsigned time fraction delay between the firings within the Firing Group in 64-nanosecond units.

HDIR

Size (bits): 1

Horizontal Direction

- 0: Clockwise
- 1: Counter-clockwise

VDIR

Size (bits): 1

Vertical Direction

N/A

VDFL

Size (bits): 14

Vertical Deflection

N/A

AZM

Size (bits): 16

Azimuth

Horizontal angle as a scaled unsigned integer encoding 0.01 degree increments in the range of [0..35999]. The value range [0..35999] encodes 0 to 359.99 degrees.

When the unsigned value is in the range [36000..65535] it represents a negative Horizontal Deflection angle as a scaled signed 2's compliment integer encoding 0.01 degree increments in the range [-29536..-1]. The value range [-29536..-1] encodes -295.36 to -0.01 degrees.

Following the Firing Group Header is a sequence of Firings. The number of Firings is (FCNT+1). The length of a Firing is calculated based on DSET and ISET.

If FDLY is zero, all channels in the Firing Group were fired simultaneously and the FSPN value may be ignored as there is no need to calculate a per-channel time offset.

If FDLY is non-zero, the channels were fired in groups separated by FDLY. The span of each group is (FSPN+1). For a firing where each channel is fired separately, FSPN is 0. For a firing where two adjacent channels are fired together the FSPN is 1. For a firing where eight channels are fired together the FSPN is 7.

The azimuth (AZM), in hundredths of a degree, indicates the horizontal component of the direction the laser was fired. This is the angle **α** in *Figure 9-1 on page 59*.

9.4.3 Firing

Following a Firing Group Header are a variable number of Firings. The size of each Firing is determined by the metadata values in the Payload Header DSET and ISET.

The Firing consists of a Firing Header followed by Returns. The number of Returns is determined by the Payload Header DSET metadata value.

9.4.3.1 Firing Header

The Firing header contains the channel number, power level, and channel status for each Firing:

Figure 9-16 Alpha Prime Firing Header

Byte Offset	0	1		2	3
00	LCN	FM	PWR	NF	STAT

LCN

Size (bits): 8

Logical Channel Number (aka Laser ID)

This value uniquely identifies the laser/detector channel position in the sensor array.

FM

Size (bits): 4 (MSN)

Firing Mode

- 0: Passive (laser not fired).
- 1: Normal.
- 2-15: Reserved.

PWR

Size (bits): 4 (LSN)

Power Level

Reserved for factory use.

NF

Size (bits): 8

Noise Factor

Sun Noise Level indicator. 0 indicates no sun.

STAT

Size (bits): 8

Channel Status Flags

Reserved for future use

9.4.3.2 Firing Return

Following each Firing Header are a variable number of Returns. The number and order of Returns is determined from the DSET value.

Each return consists of a distance followed by zero or more Intensity values.

All values of Intensity (aka Reflectivity) are meaningful, even 0, which is just one step above noise floor.

The encoding and size of the Distance value is specified by the Payload Header DSET metadata value.

The number of Intensity values is specified by the Payload Header ISET metadata value. The number of bits set in the ISET field determine the number of intensities present. The first intensity corresponds to the lowest significant bit set in the ISET, the second intensity corresponds to the next least significant bit set in the ISET, etc. Intensities are serialized in the order of least significant ISET bit to most significant ISET bit.

Each Return is byte-packed without any padding and thus alignment assumptions cannot be made when unpacking.

9.4.3.2.1 Return: Distance

The raw distance value (DIST) must be scaled to become meaningful. Multiply the raw value by the value of distLSB_ (see *Sensor Data on page 58*) to compute the measured distance in centimeters. Adjust it as needed to convert it to meters. If you are using Floating Point arithmetic, be mindful of where the decimal point is or you may lose precision. This is **R** in *Figure 9-1 on page 59*.

Note: A raw distance of 0 indicates a non-measurement. The laser was either off or a measurable reflection was not returned during the detection window.

9.4.3.2.2 Return: Confidence

Confidence

Size (bits): 8

Various per-Return Flags and Indicators

- Bit 7: DROP: Drop Flag

  1 indicates return is recommended to be dropped

- Bit 6: RESH: Retro Shadow Flag

  1 indicates return is suspected to be an artifact on trailing edge of retro: Recommended to Drop

- Bit 5: REHI: High Retro Flag

  1 indicates return is suspected to be an artifact of retro in the scene: Recommended to Drop

- Bit 4 - Bit 3: INTF: (2-bits) Interference Confidence Indicator

  3 indicates no interference suspected

  0 through 2 indicates levels of suspected interference, with 0 being most likely to be interference --> Any of these values will recommend to Drop

- Bit 2 - Bit 0: SNR: (3-bits) SNR / Accuracy indicator

  7: Highest SNR

  . . .

  4: Degraded SNR: Distance and Reflectivity Accuracy may be impacted

  . . .

  0: Lowest SNR

9.4.4 Discreet Point Timing Calculation (APF)

Calculating the exact time of a laser measurement is easier in APF mode than in HDL mode.

Laser firing times are with respect to TREF from the data packet's Payload Header.

If FDLY is zero...

Since all lasers fired in a firing group fire at the same time, the time offset (TOFFS) in the firing group header applies to them all. First, multiply TOFFS by 64 to adjust it to nanoseconds. Now, add the result to the data packet's TREF. That gives you the time of firing for the group of lasers since the epoch.

If FDLY is non-zero...

The channels were fired in groups separated by FDLY.

The epoch depends on how the sensor's clock is disciplined.

- If the sensor's clock has been free running since boot-up, the epoch corresponds to the sensor boot-up time.
- If the sensor's clock is disciplined by GPS/PPS, the epoch corresponds to the UTC epoch.

- If the sensor's clock is disciplined by gPTP, the epoch corresponds to PTP time, i.e. whatever the best master clock on the time-sensitive network happens to be.

9.4.5 Precision Azimuth Calculation (APF)

To compute an individual laser's firing azimuth, add the laser's azimuth offset to the firing group's AZM. Azimuth offsets are obtained from Alpha Prime APF.xml. See *XML File on page 84* for details on XML files and how to obtain this particular one.

Figure 9-17 on the facing page reflects the laser channel arrangement specified in **Alpha Prime APF.xml**. Each block of 8 lasers is a Firing Group. They fire simultaneously. These groups correspond to the firing groups seen in *Figure 9-8 on page 69*, which applies to both HDL and APF operation.

The APF version of the xml file, Alpha Prime APF.xml, has most of its laser channels reassigned vs the HDL firing pattern. While laser 0 fires in the same direction, laser 1 now fires in the direction laser 97 fired at, and so on. The firing groups remain the same; meaning that lasers 0 through 7 still fire in the first firing group, then lasers 8 through 15 in the second firing group, and so on. But when each group of lasers fire, the directions the lasers are fired differ from the same firing group in the HDL firing pattern. In the rearrangement, the set of AziOffsets remained the same, but most of the Elevation angles were swapped around. Only 16 lasers [0, 8, 21, 29, 34, 42, 55, 63, 65, 73, 86, 94, 99, 107, 112, 120] were not reassigned.

Figure 9-17 VLS-128 Azimuth Offsets by Elevation - APF

APF

Ch	AziOffset	Elevation	Ch	AziOffset	Elevation	Ch	AziOffset	Elevation	Ch	AziOffset	Elevation
0	-6.354	-11.742	32	-6.354	-3.860	64	-6.354	-0.340	96	-6.354	3.180
1	-4.548	-10.346	33	-4.548	-3.750	65	-4.548	-0.230	97	-4.548	3.290
2	-2.732	-9.244	34	-2.732	-3.640	66	-2.732	-0.120	98	-2.732	3.400
3	-0.911	-8.352	35	-0.911	-3.530	67	-0.911	-0.010	99	-0.911	3.510
4	0.911	-25.000	36	0.911	-4.300	68	0.911	-0.780	100	0.911	2.740
5	2.732	-19.582	37	2.732	-4.190	69	2.732	-0.670	101	2.732	2.850
6	4.548	-16.042	38	4.548	-4.080	70	4.548	-0.560	102	4.548	2.960
7	6.354	-13.565	39	6.354	-3.970	71	6.354	-0.450	103	6.354	3.070
8	-6.354	-6.500	40	-6.354	-2.980	72	-6.354	0.540	104	-6.354	4.060
9	-4.548	-6.390	41	-4.548	-2.870	73	-4.548	0.650	105	-4.548	4.170
10	-2.732	-6.280	42	-2.732	-2.760	74	-2.732	0.760	106	-2.732	4.280
11	-0.911	-6.170	43	-0.911	-2.650	75	-0.911	0.870	107	-0.911	4.390
12	0.911	-7.650	44	0.911	-3.420	76	0.911	0.100	108	0.911	3.620
13	2.732	-7.150	45	2.732	-3.310	77	2.732	0.210	109	2.732	3.730
14	4.548	-6.850	46	4.548	-3.200	78	4.548	0.320	110	4.548	3.840
15	6.354	-6.650	47	6.354	-3.090	79	6.354	0.430	111	6.354	3.950
16	-6.354	-5.620	48	-6.354	-2.100	80	-6.354	1.420	112	-6.354	4.980
17	-4.548	-5.510	49	-4.548	-1.990	81	-4.548	1.530	113	-4.548	5.180
18	-2.732	-5.400	50	-2.732	-1.880	82	-2.732	1.640	114	-2.732	5.430
19	-0.911	-5.290	51	-0.911	-1.770	83	-0.911	1.750	115	-0.911	5.730
20	0.911	-6.060	52	0.911	-2.540	84	0.911	0.980	116	0.911	4.500
21	2.732	-5.950	53	2.732	-2.430	85	2.732	1.090	117	2.732	4.610
22	4.548	-5.840	54	4.548	-2.320	86	4.548	1.200	118	4.548	4.720
23	6.354	-5.730	55	6.354	-2.210	87	6.354	1.310	119	6.354	4.830
24	-6.354	-4.740	56	-6.354	-1.220	88	-6.354	2.300	120	-6.354	8.430
25	-4.548	-4.630	57	-4.548	-1.110	89	-4.548	2.410	121	-4.548	9.700
26	-2.732	-4.520	58	-2.732	-1.000	90	-2.732	2.520	122	-2.732	11.750
27	-0.911	-4.410	59	-0.911	-0.890	91	-0.911	2.630	123	-0.911	15.000
28	0.911	-5.180	60	0.911	-1.660	92	0.911	1.860	124	0.911	6.080
29	2.732	-5.070	61	2.732	-1.550	93	2.732	1.970	125	2.732	6.480
30	4.548	-4.960	62	4.548	-1.440	94	4.548	2.080	126	4.548	6.980
31	6.354	-4.850	63	6.354	-1.330	95	6.354	2.190	127	6.354	7.580

9.5 Converting PCAP Files to Point Cloud Formats

Converting a packet capture (pcap) file of Velodyne Lidar data to a LAS, LAZ, XYZ, PLY, or other point cloud file format can be a non-trivial process.

The data provided by the sensor in the pcap file is measured relative to the sensor's reference frame - the sensor's internal three dimensional coordinate system which moves with the sensor. That's markedly different from a point cloud file where the data points are referenced to a single coordinate system. That coordinate system might be an earth-frame (latitude, longitude, elevation) or another convenient reference frame.

Processing the raw lidar data into a point cloud is called geo-referencing. In geo-referencing, the user takes into account the sensor's position (X/Y/Z) and orientation (pitch/roll/yaw) for each measurement. Knowing these six values enables a user to perform the proper mathematical rotations and translations to reference the lidar data into a single coordinate system.

Two popular techniques Velodyne Lidar customers use for geo-referencing are inertial referencing and Simultaneous Localization and Mapping (SLAM).

With inertial referencing, the location and orientation of the sensor at every moment is recorded with an Inertial Navigation System (INS). An INS combines a Global Positioning System (GPS) receiver with an Inertial Measurement Unit (IMU). Data from both the INS and lidar sensor are time-synchronized to the GPS satellite's reference clock, enabling the user to match each lidar data point with its corresponding position and orientation from the INS. Having matched the lidar data with the INS data, each measurement is mathematically translated into a single coordinate system.

SLAM is a technique used by robots to analyze and navigate their environment. SLAM software automatically identifies stationary objects in the lidar data. The algorithm then uses the location of the stationary objects to mathematically back-out the movement of the lidar and translate the data into a single coordinate system.

Most Velodyne Lidar customers develop their own geo-referencing systems. However, inertial referencing and SLAM solutions are available through third parties, and many can be found on the system integrators page at: http://velodynelidar.com/integrators.php.

9.6 XML File

The purpose of the sensor xml file is to provide programmatic access to fixed parameters associated with the sensor model. These include the range scaling factor (distLSB_), the vertical angles (vertCorrection_) for each channel, and the azimuthal offsets (rotCorrection_) for each channel. (Other corrections in the px block are not used.)

Raw distance values from data packets must be scaled by the range scaling factor to have a meaningful value. The scaling factor unit is centimeters. So, for example, a distLSB_ of 0.4 represents a scaling factor of 0.4 cm, or 4 mm.

vertCorrection_ is found in the points_.item.px structures. The laser channel id (id_) indicates which laser channel (aka LCN) the correction applies to.

rotCorrection_ is also found in the points_.item.px structures. The laser channel id (id_) indicates which laser channel the correction applies to. They are both angles given in degrees. rotCorrection_ only applies to sensors with lasers arranged azimuthally as well as vertically, i.e. the VLP-32C and VLS-128. rotCorrection_ does not apply to directional sensors such as the Velarray or rotational sensors with lasers organized and fired in a single column.

Beyond this sensor model configuration information, the sensor xml file does not contain any calibration data. We no longer refer to them as calibration files as a result. Only the HDL-64E model used it that way.

Official, publicly available xml files are obtained when you install a recent version of VeloView. Look in its *share* folder upon installation (or unzip in the case of linux). Any other xml files, such as for new or special models, may be obtained by submitting a Tech Support Request ticket. Use the **Contact Technical Support** form. A signed, complete NDA on file may be required.

Chapter 10 • Sensor Communication

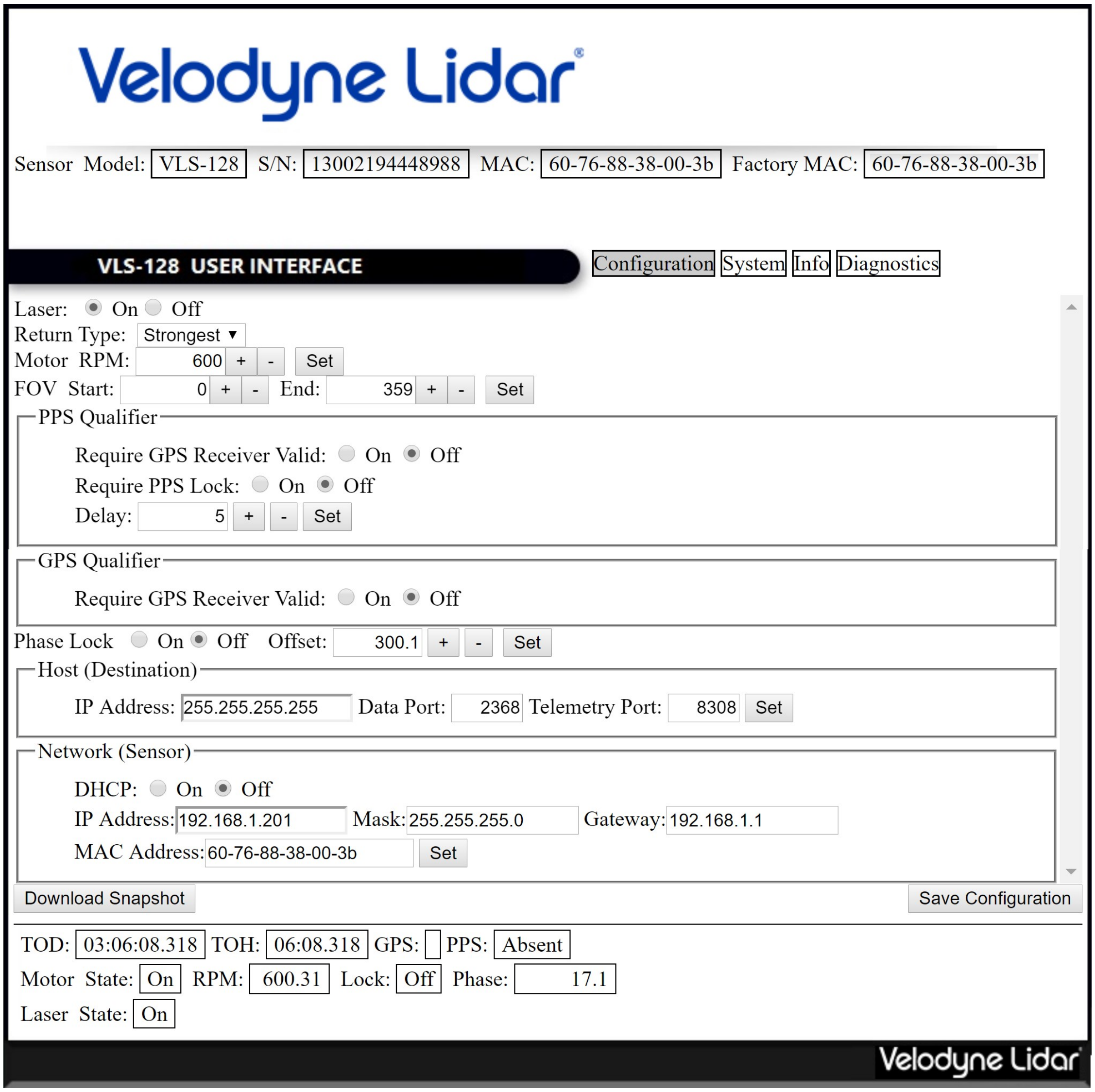
This section provides detailed information about sensor communication methods.

10.1 Web Interface

The easiest way to control a sensor's operation is through its Web Interface. It has four screens (or tabs). This section provides an explanation of the feature sets and functionality on each screen.

10.1.1 Configuration Screen

Figure 10-1 Alpha Prime Configuration Screen



Note: Variable overlaid controls from Point Cloud Stream Format and Time Of Day not shown

Table 10-1 Configuration Screen Functionality and Features

Function	Description	Default
Laser ON/OFF	Turns the sensor's laser ON or OFF.	ON
Return Type	**Strongest** – Sensor provides only the strongest detected return. **Last** – Sensor provides only the last (temporally) detected return.	Strongest

Function	Description	Default
	Dual – Sensor provides both the strongest and last returns. If the last is also the strongest, the sensor provides the second-strongest return. If only one return is detected, the two values will be identical. **Strongest + Confidence (APF Only)** – Sensor provides only the strongest returns. Confidence data is also provided. **Dual + Confidence** – Sensor provides both the strongest and last returns. If the last is also the strongest, the sensor provides the second-strongest return. If only one return is detected, the two values will be identical. Confidence data is also provided. **Custom (APF only)** – Sensor provides the specified returns. Confidence data is added if specified.	
Point Cloud Stream Format	Instantly changes the data stream output format. **HDL** - Changes stream output to HDL Legacy Packet Format. **APF** - Changes stream output to Velodyne Advanced Packet Format.	HDL
Motor RPM	Sets the sensor's internal spin rate. Valid values include the range from 300 to 1200 RPM, and 0. Setting it to 0 during active scanning will auto-disable the sensor's lasers, which re-enable automatically the next time RPM is set in the valid non-zero range. See *Rotation Speed (RPM) on page 55* for more. **Set** - Activates the changes.	600
FOV Start	Sets the start azimuth for a user-defined Field of View. Units: integer degrees. **Set** - Activates the changes.	0
FOV End	Sets the end azimuth for a user-defined Field of View. Units: integer degrees. **Set** - Activates the changes.	359
Time of Day	Sets the sensor clock update policy. **Free Running**: Clock runs without provision for updates. No options to configure. **GPS**: Sensor accepts clock updates via GPS / PPS. See PPS Qualifier and GPS Qualifer settings below. **PTP**: Sensor accepts clock updates via gPTP protocol. See Top-Of-Hour Timestamp and PTP Configuration settings below.	Free Running
(GPS) **PPS Qualifier**	Adjusts parameters that affect how PPS is used when synchronizing time stamps to UTC. **Require GPS Receiver Valid**: ON/OFF. **Require PPS Lock**: ON/OFF. **Delay**: Sets the qualification period (in seconds) for PPS. **Set** - Activates the changes.	OFF ON 5
(GPS) **GPS Qualifier**	Controls whether GPS Receiver Valid is required when synchronizing time stamps to UTC. **Require GPS Receiver Valid**: ON/OFF. **Set** - Activates the changes.	OFF
(PTP) **Top-Of-Hour**	Applies UTC offset (from TAI) to PTP times if enabled. **Apply UTC Offset**: ON / OFF.	OFF

Function	Description	Default
Timestamp		
(PTP) **PTP Configuration**	Sets the PTP propagation delay. Increase it for a lower quality time-sensitive network to allow for higher latencies. **Neighbor Propagation Delay Threshold (ns)**: _____. **Set** - Activates the changes.	10000
Phase Lock	Used to synchronize the firing of two or more sensors. The offset is used to set the azimuth at which the laser fires in conjunction with the PPS signal's rising edge. Requires a valid PPS signal. **Offset**: Enter the phase offset in decimal degrees. For example, if the desired offset is 270º, enter 270. If it's 120.5, enter 120.5. **Set** - Activates the changes.	OFF 0
Host (Destination)	**IP Address** (IPv4) to which the sensor transmits its data. By default it is set to the broadcast address. If using one or more sensors in a network it must be set to a non-broadcast IP address to avoid cross-loading. **Data Port** – Port number for laser data packets. **Telemetry Port** – Port number for position (telemetry) packets. **Set** – Activates changes.	255.255.255.255 2368 8308
Network (Sensor)	**DHCP** – When ON, the sensor obtains its IP address from a DHCP server on its local network. When set to OFF, it uses the IP Address, Mask, and Gateway the user configures. **IPv4 Address** – Sensor IP address as specified by the user. **Mask** – Sensor subnet mask as specified by the user. **Gateway** – Sensor gateway as specified by the user. **MAC Address** - Sensor MAC address override. **Set** – Activates changes. **Note:** Take care when turning DHCP ON. Before doing so, ensure a DHCP server on the network is available to provide the sensor with an IP address. If you turn DHCP ON and lose contact with the sensor, follow the *Turned DHCP On, Lost Contact With Sensor on page 106* procedure to get it back.	OFF 192.168.1.201 255.255.255.0 192.168.1.1
Save Configuration	Saves all current settings to flash memory. Whenever the sensor powers up or is reset, it is restored to these saved settings. Only the most recent save is kept.	
Download Snapshot	Clicking this button generates an ASCII text file (name format: <SerialNumber>.hdl) for downloading. It contains a JSON object capturing the sensor's state at that moment. The file can be sent to Velodyne *Technical Support on page 108* for analysis and to obtain troubleshooting advice.	
Time Status	**TOD** – Displays the Time of Day (i.e. time since midnight) [HH:MM:ss.mmm]. **TOH** – Displays the time since the Top of the Hour [MM:ss.mmm]. **GPS** – Displays the most recent GPS latitude and longitude. **PPS** – Displays the current state of the PPS signal provided from a	

Function	Description	Default
	GPS. Absent, Synchronizing, Locked, and Error are the valid states.	
Motor Status	**Motor State** – Indicates if the sensor's internal motor is spinning: ON/OFF. **RPM** – The sensor's current RPM. 0 means that the motor is not spinning. **Lock** – Indicates the phase lock feature state: ON/OFF. **Phase** – Indicates the current phase offset. Only relevant when the phase lock feature is active. The value presented is in hundredths of a degree.	
Laser Status	**Laser State** – Indicates if the lasers are collectively enabled or disabled: ON / DISABLED.	

See *Time Synchronization on page 155* for additional information on the PPS and GPS Qualifier functions.

10.1.1.1 MAC Address

It may be advantageous to change the sensor's MAC address for asset management in multi-sensor situations where the MAC address is the closest thing to a hard sensor ID, or as an anti-tracking measure. Bear in mind, however, that multiple sensors with the same MAC address running on the same network will likely produce undesirable results.

Note: When changing the sensor's MAC Address, the first octet must not be set to an odd value. `00` is ok. `60` is ok. `ff` is not ok.

As defined by IEEE 802 Standards, the least significant bit of the first octet of a MAC address identifies it as a multicast address. A sensor bearing a multicast MAC address may cause undesirable network issues.

By extension, never set a MAC Address to the broadcast address (aka. all bits on) or any other group address.

10.1.1.2 Correctly reset MAC Address to Factory MAC Address

If you have changed the sensor's MAC Address to something other than the Factory MAC Address and wish to reset it back, follow the procedure below.

1. On the sensor's Configuration Screen, enter `00-00-00-00-00-00` in the MAC Address field. The '-' characters must be present and the six octet values must be expressed as two hexadecimal digits each, all zeroes.
2. Click the **Save Configuration** button.
3. Click **System** to view the System Screen.
4. Click the **Reset System** button.

Once the sensor is done resetting, you may refresh the page and go back to the Configuration Screen. The MAC Address should match the Factory MAC Address.

10.1.2 System Screen

Figure 10-2 Alpha Prime System Screen

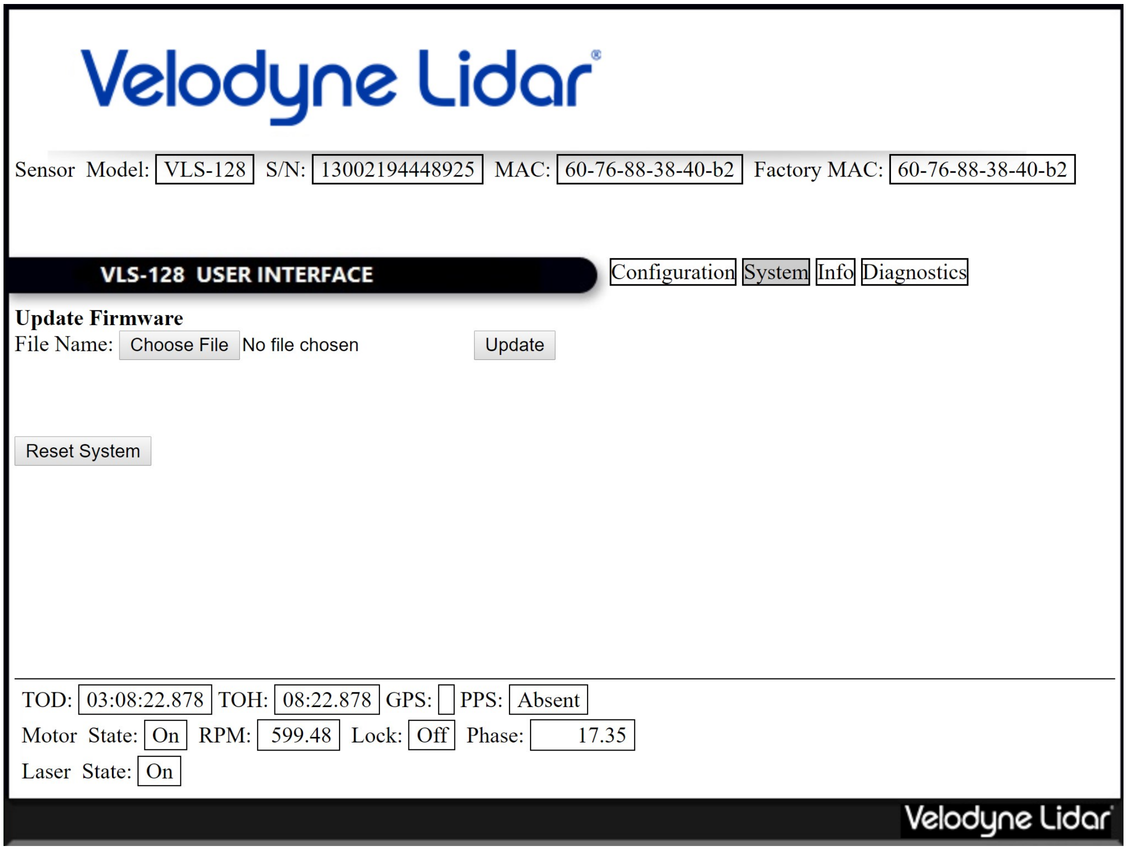


Table 10-2 System Screen Functionality and Features

Function	Description
Update Firmware	**Choose File** – Enables user to select a new firmware image file. **Update** – Initiates the firmware update process.
Reset System	Used to reset the system after a firmware update.
Time Status	See description in *Table 10-1 on page 86*.
Motor Status	See description in *Table 10-1 on page 86*.
Laser Status	See description in *Table 10-1 on page 86*.

10.1.3 Info Screen

Figure 10-3 Alpha Prime Info Screen

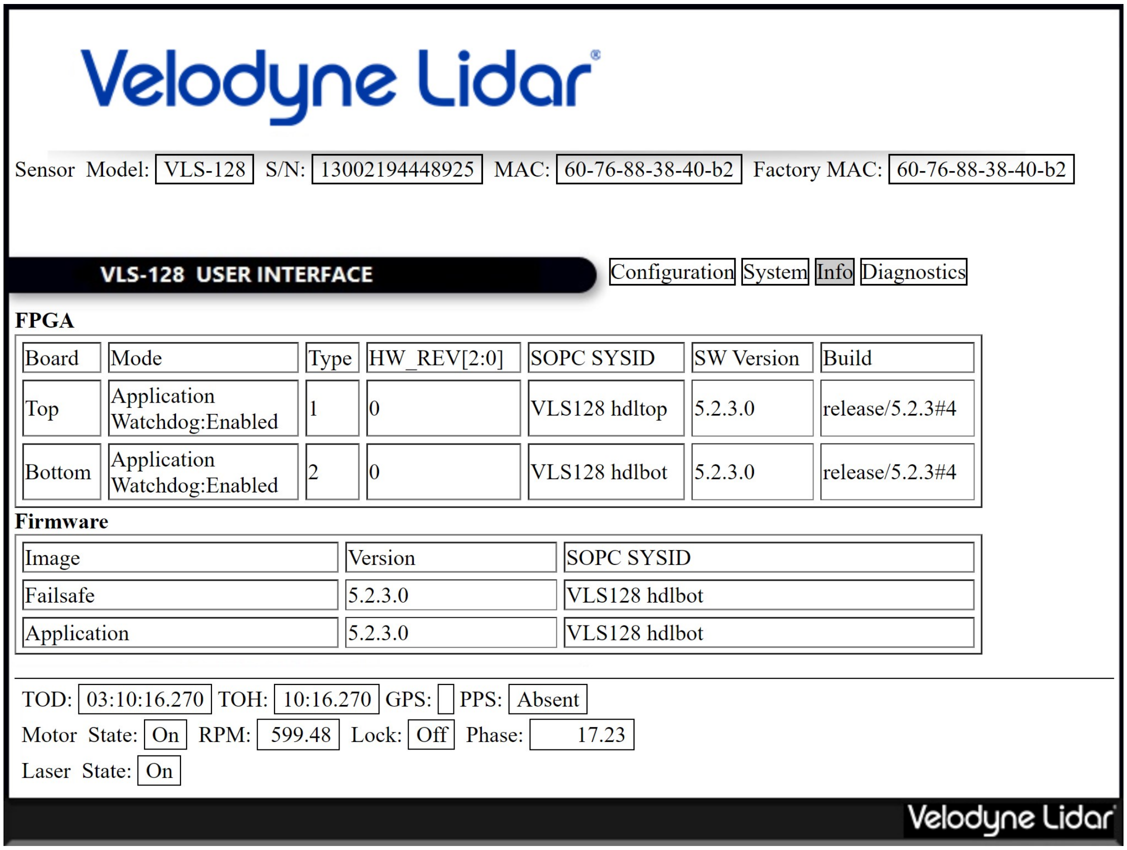


Table 10-3 Info Screen Functionality and Features

Function	Description
FPGA (top board)	**Board** – Top or Bottom board information. **Mode** – State of the Application Watchdog - Velodyne use only. **Type** – Velodyne Use Only. **HW Version** – Velodyne Use Only. **SOPC SYSID** – Velodyne Use Only. **SW Version** – Current operating firmware version. **Build** – Velodyne Use Only.
FPGA (bottom board)	**Board** – Top or Bottom board information. **Mode** - Velodyne use only. **Type** – Velodyne Use Only. **HW Version** – Velodyne Use Only. **SOPC SYSID** – Velodyne Use Only. **SW Version** – Current operating firmware version. **Build** – Velodyne Use Only.
Firmware Image	**Image** – Firmware type (Failsafe and Application).
Firmware Version	**Version** – Version of the stored Failsafe and Application images.
SOPC SYSID	**Failsafe boot(00)** – Velodyne Use Only.

Function	Description
	Application hdlbot(03) – Velodyne Use Only.
Time Status	See description in *Table 10-1 on page 86*.
Motor Status	See description in *Table 10-1 on page 86*.
Laser Status	See description in *Table 10-1 on page 86*.

10.1.4 Diagnostics Screen

Figure 10-4 Alpha Prime Diagnostics Screen

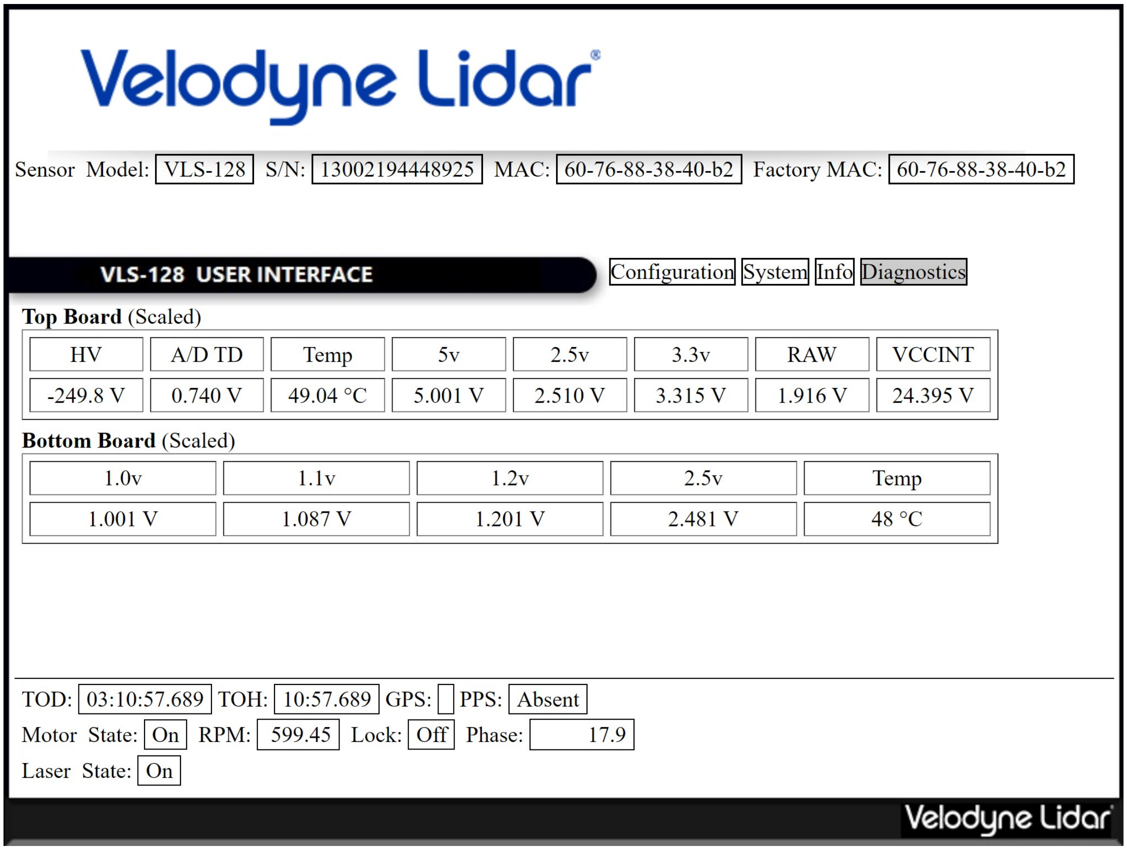


Table 10-4 System Screen Functionality and Features

Function	Description
Top Board (Scaled)	**HV** – Velodyne Use Only. **A/D TD** – Velodyne Use Only. **Temp** – Top board temperature. **5v** – 5 V rail. **2.5v** – 2.5 V rail. **3.3v** – 3.3 V rail. **RAW** – A raw, internal voltage. **VCCINT** – VCC voltage.
Bottom Board (Scaled)	**1.0v** – 1.0 V rail. **1.1v** – 1.1 V rail. **1.2v** – 1.2 V rail. **2.5v** – 2.5 V rail. **Temp** – Bottom board temperature.
Time Status	See description in *Table 10-1 on page 86*.

Function	Description
Motor Status	See description in *Table 10-1 on page 86*.
Laser Status	See description in *Table 10-1 on page 86*.

10.2 Sensor Control with curl

In addition to the Web Interface, Velodyne Lidar sensors provide an HTTP interface for programmatic configuration and control. One of the easiest tools to use for this purpose is curl. curl is a command line program freely available on a wide variety of operating systems. curl specializes in transferring data between devices using Internet protocols. You can learn more about curl and libcurl, the library (written in C) it uses to do all the heavy lifting, at this page: https://ec.haxx.se/curl-does.html. libcurl has bindings for many environments and programming languages.

Note: curl is aliased to Invoke-WebRequest in recent versions of PowerShell (Windows OS). If your curl command does not behave as expected, this may be the reason. Solutions include invoking curl.exe instead of curl, removing the curl alias, and reworking the command the Invoke-WebRequest way. Perhaps the easiest solution is to switch from PowerShell to CMD. Select one that works best for you.

The sections below show how to command and control a sensor from the command line using curl, and from a Python program using pycurl, a Python binding for libcurl. There are many other ways to do it. You aren't limited to these techniques. Underneath them all, you are either issuing http GET requests or http POST requests with optional parameters as data.

10.2.1 Using curl with Velodyne Lidar Sensors

Note: It is recommended to issue at most one curl command every 5 seconds to the sensor. This applies specifically to the classic curl command on the command line. The sensor only has 4 or 5 sessions it may have open at a given time. Classic curl times out after 5 seconds, closing the session. Too many sessions open may block subsequent ones. Whereas reusing a session with a series of HTTP requests in python, C++, etc. will not see such issues. To override classic curl's session close behavior, pass it the **-H "Connection: close"** option.

---

curl takes care of the following items for you. But you should follow them when using libcurl in your software.

- Always check the response from the sensor, even if the function used does not return one. (curl can and will respond with status/error messages.)
- Always close the socket connection to the sensor when finished.

10.2.2 curl Command Parameters

To obtain JSON-formatted sensor status, invoke curl with the following URLs:

- http://192.168.1.201/cgi/status.json - Returns the current operational state and parameters of the sensor.
- http://192.168.1.201/cgi/diag.json - Returns diagnostic information from the sensor.
- http://192.168.1.201/cgi/snapshot.hdl - Returns current sensor configuration and status data.

The following sensor parameters can be changed by sending a urlencoded tagged key and data pair to http://192.168.1.201/cgi/setting:

1. rpm=[integer]
2. returns=[Strongest]|[Last]|[Dual]

Set Field of View by sending a urlencoded key and data pair to http://192.168.1.201/cgi/setting/fov:

1. [start]|[end]=[integer]

Set Sensor Address, Destination Port, or Telemetry Port by sending a urlencoded key and data pair to http://192.168.1.201/cgi/setting/host:

1. addr=255.255.255.255
2. dport=2368
3. tport=8309

Network Settings can be set by providing the urlencoded key and data pair to http://192.168.1.201/cgi/setting/net:

1. addr=192.168.1.200
2. mask=255.255.255.0
3. gateway=192.168.1.2
4. dhcp=[on|off]

Special Commands to Save settings to NVRAM and Reset the sensor:

1. http://192.168.1.201/cgi/save
    a. submit
2. http://192.168.1.201/cgi/reset
    a. reset_system

10.2.3 Command Line curl Examples

10.2.3.1 Get Diagnostic Data

Returns a JSON-formatted string containing values as seen on the Diagnostics page of the sensor's Web Interface.

Command:

```
curl http://192.168.1.201/cgi/diag.json
```

Example Response:


```
{
    "volt_temp":{
        "top":{
            "hv":2917,
            "ad_temp":606,
            "lm20_temp":1105,
            "pwr_5v":2075,
            "pwr_2_5v":2047,
            "pwr_3_3v":2693,
            "pwr_raw":1428,
            "pwr_vccint":631
        },
        "bot":{
            "pwr_1_0v":1640,
            "pwr_1_1v":1792,
            "pwr_1_2v":1964,
            "pwr_2_5v":4064,
```

```
                "lm20_temp":1231,
                "valid":true
            }
        },
    }
```


10.2.3.2 Conversion Formulas

Equation 10-1 Standard Voltage Conversion

$$volts\ (V) = \frac{raw\ value \text{ x } 5.0}{4096}$$

Equation 10-2 Standard Current Conversion

$$amps\ (A) = 10.0 \text{ x} \left(\frac{raw\ value \text{ x } 5.0}{4096} - 2.5 \right)$$

Equation 10-3 Standard Temperature Conversion

$$temp\ (^{\circ}\ C) = \text{-}1481.96 + \sqrt{2.1962 \times 10^{6} + \frac{\left(1.8639 - \left(\frac{raw\ value \text{ x } 5.0}{4096}\right)\right)}{3.88 \times 10^{-6}}}$$

10.2.3.3 Interpret Diagnostic Data

Use the formulas above to convert raw diagnostic data fields below to usable values. Some fields are scaled by additional factors.

10.2.3.3.1 top:hv

The HV field comes from the sensor's top board. It represents the high voltage bias to the APD (Avalanche photodiode).

To convert the raw value, use *Equation 10-1 above* then scale the result by subtracting 5.0 and multiplying the result by 101.0.

Example:

hv = **2917** *(raw value)*

(2917 * 5.0) / 4096 = 3.561

101.0 * (3.561 - 5.0) = **-145.34 V**

10.2.3.3.2 top:ad_temp

Despite having 'temp' in its field name, this field indicates an internal voltage on the top board.

To convert the raw value, use *Equation 10-1 on the previous page*.

Example:

ad_temp = **606** *(raw value)*

(606 * 5.0) / 4096 = **0.740 V**

10.2.3.3.3 top:lm20_temp

This field indicates the temperature of the top board.

To convert the raw value, use *Equation 10-3 on the previous page*.

Example:

lm20_temp = **1105** *(raw value)*

(1105 * 5.0) / 4096 = 1.3489

-1481.96 + SQRT(2.1962E6 + ((1.8639 - 1.3489) / 3.88E-6)) = **44.126° C**

10.2.3.3.4 top:pwr_5v

This is the top board's 5.0 V rail.

To convert the raw value, use *Equation 10-1 on the previous page* then scale the result by multiplying it by 2.0.

Example:

pwr_5v = **2075** *(raw value)*

(2075 * 5.0) / 4096 = 2.533

2.533 * 2.0 = **5.066 V**

10.2.3.3.5 top:pwr_2_5v

This is the top board's 2.5 V rail.

To convert the raw value, use *Equation 10-1 on the previous page*.

Example:

pwr_2_5v = **2047** *(raw value)*

(2047 * 5.0) / 4096 = **2.499 V**

10.2.3.3.6 top:pwr_3_3v

This is the top board's 3.3 V rail.

To convert the raw value, use *Equation 10-1 on page 96*.

Example:

pwr_3_3v = **2693** *(raw value)*

(2693 * 5.0) / 4096 = **3.287 V**

10.2.3.3.7 top:pwr_raw

This is the top board's unregulated power rail.

To convert the raw value, use *Equation 10-1 on page 96*.

Example:

pwr_raw = **1428** *(raw value)*

(1428 * 5.0) / 4096 = **1.743 V**

10.2.3.3.8 top:pwr_vccint

This is the top board's VCCINT core voltage rail.

To convert the raw value, use *Equation 10-1 on page 96* then scale the result by multiplying it by 32.915.

Example:

pwr_vccint = **631** *(raw value)*

(631 * 5.0) / 4095 = 0.770451 V

0.770 * 32.915 = **25.359 V**

10.2.3.3.9 bot:lm20_temp

This field indicates the temperature of the bottom board.

To convert the raw value, use *Equation 10-3 on page 96*.

Example:

lm20_temp = **1231** *(raw value)*

(1231 * 5.0) / 4096 = 1.503

-1481.96 + SQRT(2.1962E6 + ((1.8639 - 1.503) / 3.88E-6)) = **31.082° C**

10.2.3.3.10 bot:pwr_1_0v

This is the bottom board's 1.0 V rail.

To convert the raw value, use *Equation 10-1 on page 96* then scale the result by dividing by 2.0.

Example:

pwr_1_0v = **1640** *(raw value)*

(1640 * 5.0) / 4096 = 2.002

2.002 / 2.0 = **1.001 V**

10.2.3.3.11 bot:pwr_1_1v

This is the bottom board's 1.1 V rail.

To convert the raw value, use *Equation 10-1 on page 96* then scale the result by dividing by 2.0.

Example:

pwr_1_1v = **1792** *(raw value)*

(1792 * 5.0) / 4096 = 2.1875

2.1875 / 2.0 = **1.094 V**

10.2.3.3.12 bot:pwr_1_2v

This is the bottom board's 1.2 V rail.

To convert the raw value, use *Equation 10-1 on page 96* then scale the result by dividing by 2.0.

Example:

pwr_1_2v = **1964** *(raw value)*

(1964 * 5.0) / 4096 = 2.397

2.397 / 2.0 = **1.199 V**

10.2.3.3.13 bot:pwr_2_5v

This is the bottom board's 2.5 V rail.

To convert the raw value, use *Equation 10-1 on page 96* then scale the result by dividing by 2.0.

Example:

pwr_2_5v = **4064** *(raw value)*

(4064 * 5.0) / 4096 = 4.9609

4.9609 / 2.0 = **2.480 V**

10.2.3.4 Get Snapshot

Downloads sensor snapshot data to an HDL file. This has essentially the same effect as clicking on the **Download Snapshot** button under the Configuration tab.

Command:

```
curl http://192.168.1.201/cgi/snapshot.hdl > snapshot.hdl
```

10.2.3.5 Get Sensor Status

Returns sensor state data, such as: GPS PPS is present, GPS position, motor status, phase, and laser states.

Command:

curl http://192.168.1.201/cgi/status.json

Example Response:


```
{
    "gps":{
            "pps_state":"Absent",
            "position":""
    },
    "motor":{
            "state":"On",
            "rpm":600,
            "lock":"Off",
            "phase":0
    },
    "laser":{
        "state":"On"
    }
}
```


10.2.3.6 Get Sensor Info

Returns sensor information, such as: Sensor Model, Serial Number, and firmware Versions.

Command:

curl http://192.168.1.201/cgi/info.json

10.2.3.7 Set Motor RPM

Sets the RPM of the motor. Valid integer values range from 300 to 1200, in increments of 60. (**If the RPM setting is not evenly divisible by 60, neither motor speed control nor phase lock functions will function properly.**) For values 1 through 299, the sensor defaults back to 300 RPM. If a value of 0 or less is entered, the sensor motor powers down and the lasers are turned off, as leaving them on with the motor stopped would be an unsafe eye state. This has the same effect as setting the value for the **Motor RPM** in the Web Interface.

Command:

curl --data “rpm=[integer]” http://192.168.1.201/cgi/setting

Example:

curl --data “rpm=600” http://192.168.1.201/cgi/setting

10.2.3.8 Set Field of View

Sets the field of view (0° to 359°). Numbers outside this range are quietly ignored. This has the same effect as setting the **FOV Start** and **FOV End** values on the Web Interface.

Command:

curl --data “[start]|[end]=[integer]” http://192.168.1.201/cgi/setting/fov

Examples:

curl --data "start=10" http://192.168.1.201/cgi/setting/fov

curl --data "end=270" http://192.168.1.201/cgi/setting/fov

10.2.3.9 Set Return Type (Strongest, Last, Dual)

This command sets the return type (or mode) of the sensor. Choose one: Strongest, Last, and Dual. This has the same effect as selecting the Web Interface **Return Type**.

Command:

```
curl --data “returns=[Strongest]|[Last]|[Dual]” http://192.168.1.201/cgi/setting
```

Examples:

```
curl --data “returns=Strongest” http://192.168.1.201/cgi/setting
curl --data “returns=Last” http://192.168.1.201/cgi/setting
curl --data "returns=Dual" http://192.168.1.201/cgi/setting
```

10.2.3.10 Save Configuration

Saves the configuration so that the settings are persistent across power cycles. This is equivalent to clicking on the **Save Configuration** button under the Configuration tab in the Web Interface.

Command:

```
curl --data “submit” http://192.168.1.201/cgi/save
```

10.2.3.11 Reset System

Resets the sensor. This command performs the same operation as pressing the **Reset System** button under the System tab in the Web Interface, or if you cycled power to the sensor.

Command:

```
curl --data "reset_system" http://192.168.1.201/cgi/reset
```

Example Response:

The system resets.

10.2.3.12 Set Phase Lock Offset

This command performs the same operation as pressing the Phase Lock **SET** button under Configuration tab in the Web Interface.

Command:

```
curl --data “enabled={on|off}&offset=27025&offsetInput=270.25” http://192.168.1.201/cgi/settings/phaselock
```

The enabled parameter value must be either on or off. Note the relationship between the offset and offsetInput parameter values. The latter is what a user would have entered in the Offset field. The offset value must be the integer equivalent of offsetInput multiplied by 100. The ampersand characters are parameter delimiters.

Once set, the effect takes place immediately. However, if the web interface is open, it won't be updated until the page is refreshed.

10.2.3.13 Get Sensor Settings

Returns current sensor settings including laser state, laser return type, RPM, FOV, time-of-day, PTP, GPS/PPS control, phase lock, and network configuration.

Command:

```
curl http://192.168.1.201/cgi/settings.json
```

10.2.3.14 Set Host (Destination) IP Address

This command sets the destination IP address where firing data and Position/Telemetry packets are sent. This has the same effect as entering a **Host IP** address on the sensor's Web Interface.

Command:

```
curl --data "addr=255.255.255.255" http://192.168.1.201/cgi/setting/host
```

10.2.3.15 Set Data Port

This command sets the sensor's data port (default: 2368). It has the same effect as setting the Web Interface's **Data Port** field to the given number.

Command:

```
curl --data "dport=2368" http://192.168.1.201/cgi/setting/host
```

10.2.3.16 Set Telemetry Port

This command sets the sensor's telemetry port (default: 8308). It has the same effect as setting the port number in the Web Interface's **Telemetry Port** field.

Command:

```
curl --data "tport=8309" http://192.168.1.201/cgi/setting/host
```

10.2.3.17 Set Network (Sensor) IP Address

This sets the IP address of the device. The default is 192.168.1.201. This has the same effect as entering an IP address in the Web Interface **Network IP** field.

Command:

```
curl --data "addr=192.168.1.200" http://192.168.1.201/cgi/setting/net
```

10.2.3.18 Set Netmask

This sets the network mask of the sensor. The default is 255.255.255.0. This has the same effect as entering a net mask in the Web Interface **Mask** field.

Command:

```
curl --data "mask=255.255.0.0" http://192.168.1.201/cgi/setting/net
```

10.2.3.19 Set Gateway

This sets the gateway address of the sensor. The default is 192.168.1.1. This has the same effect as entering a gateway address in the Web Interface **Gateway** field.

Command:

```
curl --data "gateway=192.168.1.2" http://192.168.1.201/cgi/setting/net
```

10.2.3.20 Set DHCP

This determines if the sensor is to rely on a DHCP server for its IP address.

This has the same effect as selecting the corresponding radio button for **DHCP** in the Web Interface.

Take care when turning DHCP ON. Before doing so, ensure a DHCP server on the network is available to provide the sensor with an IP address. If you turn DHCP ON and lose contact with the sensor, follow the *Turned DHCP On, Lost Contact With Sensor on page 106* procedure to get it back.

Command:

```
curl --data "dhcp=[on|off]" http://192.168.1.201/cgi/setting/net
```

10.2.4 curl Example using Python

The script below demonstrates how to do basic operations with a sensor using PycURL, a Python package built on libcurl. It works in Python 2 and 3. For more information on PycURL go to: http://pycurl.io/docs/.

The script expects the sensor to have its default IP address. If it's different, change Base_URL accordingly.

The sensor object contains the interface to the actual sensor. A buffer is required because PycURL does not provide storage for the network response. After each sensor request, the HTTP response code is checked for success.

The first operation is to reset the sensor. After a delay, the script changes a couple of sensor parameters. Waiting another 10 seconds (for the motor to spin up), sensor status is obtained and printed to standard out.

```
    import pycurl
    try:
        from io import BytesIO
    except ImportError:
        from StringIO import StringIO as BytesIO
    try:
        from urllib.parse import urlencode
    except ImportError:
        from urllib import urlencode
    import urllib2
    import json
    import time

    def sensor_do(s, url, pf, buf):
        s.setopt(s.URL, url)
        s.setopt(s.POSTFIELDS, pf)
        s.setopt(s.WRITEDATA, buf)
        s.perform()
        rcode = s.getinfo(s.RESPONSE_CODE)
        success = rcode in range(200, 207)
        print('%s %s: %d (%s)' % (url, pf, rcode, 'OK' if success else 'ERROR'))
        return success

    Base_URL = 'http://192.168.1.201/cgi/'

    sensor = pycurl.Curl()
    buffer = BytesIO()

    rc = sensor_do(sensor, Base_URL+'reset', urlencode({'data':'reset_system'}), buf-
fer)
    if rc:
        time.sleep(10)
        rc = sensor_do(sensor, Base_URL+'setting', urlencode({'rpm':'300'}), buffer)
    if rc:
        time.sleep(1)
        rc = sensor_do(sensor, Base_URL+'setting', urlencode({'laser':'on'}), buffer)
    if rc:
        time.sleep(10)

    response = urllib2.urlopen(Base_URL+'status.json')
    if response:
        status = json.loads(response.read())
        print 'Sensor laser is %s, motor rpm is %s' % \
              (status['laser']['state'], status['motor']['rpm'])
```

```
    sensor.close()
```

Typical output looks like the following:

```
http://192.168.1.201/cgi/reset data=reset_system: 204 (OK)
http://192.168.1.201/cgi/setting rpm=300: 204 (OK)
http://192.168.1.201/cgi/setting laser=on: 204 (OK)
Sensor laser is On, motor rpm is 301
```

10.2.5 Python Example using Requests HTTP library

The script below demonstrates how to query a sensor for its model name, serial number, MAC address, and firmware version(s).

You should check response.status_code for success (if it's within the 200..229 range) before extracting json content from the response.

Firmware versions are decimal integers packed like IP addresses; thus, the use of the ipaddress library to unpack them, though there are other similar methods that work.

```
import requests
import ipaddress

response = requests.get("http://192.168.1.201/cgi/info.json")
data = response.json()

model = data['model']
sn = data['serial']
mac = data['mac_addr']
top_firmware_version = str(ipaddress.ip_address(data['image']['top']['version']))
bottom_firmware_version = str(ipaddress.ip_address(data['image']['application']
['version']))
```

Chapter 11 • Troubleshooting

This section provides detail on how to troubleshoot your sensor, how to request technical assistance, how to have the sensor repaired, and how to replace the fuse in the Interface Box.

11.1 Troubleshooting Process

Table 11-1 Common Problems and Resolutions

Problem	Resolution
Interface Box LEDs do not light	Verify: • Power connection and polarity. • Fuse in the Interface Box is okay. See *Fuse Replacement on page 107* if it's blown.
Rotor doesn't spin	Verify: • Power connection and polarity. • Fuse in the Interface Box is okay. See *Fuse Replacement on page 107* if it's blown.
Unit spins but no data	Verify: • Ethernet wiring is functional. • Packet output using another application (e.g. VeloView/Wireshark). • Receiving computer's network settings. • Correct static IP address in your computer's network settings. • No security software is installed which may block Ethernet broadcasts. • Laser **ON** radio button is selected in Web Interface *Configuration Screen on page 86*. • Input voltage and current draw are in proper ranges.
Can see data in Wireshark but not VeloView	Wireshark sniffs packets promiscuously but VeloView does not. VeloView needs permission to receive packets from the system firewall, if active. Check: • No firewall is active on receiving computer.

Problem	Resolution
	• Sensor's destination address. • Port numbers are set to 2368 (data) and 8308 (position/telemetry).
Cannot see sensor's Web Interface	Verify: • Ethernet wiring is functional. • Packet output using another application (e.g. VeloView/Wireshark). • Receiving computer's network settings. • Correct static IP address in your computer's network settings. • No security software is installed which may block Ethernet broadcasts. Action: • Reboot computer - sometimes network settings do not take effect until the computer is rebooted.
Data dropouts	This is nearly always an issue with the network and/or user computer. Check the following: • Is the sensor's horizontal field of view set to less than 360°? • Is there excessive traffic and/or collisions on network? • Is a network device throttling back traffic? Devices such as wireless access points often throttle broadcast data. • Are excessive broadcast packets from another service being seen by the sensor? This can slow the sensor down. • Is the computer fast enough to keep up with the packet flow coming from the sensor? • Remove all network devices and test with a computer directly connected to the sensor.

GPS Not Synchronizing	Check the following:
	• Baud rate set to 9600 and serial port set to 8N1 (8 bits, no parity, 1 stop bit).
	• Electrical continuity of PPS and serial wiring.
	• Incorrect construction of NMEA sentence or incorrect NMEA sentence.
	• Incorrect signal levels (see *Electrical Requirements on page 45* for details):
	  ◦ "High" voltage must be greater than 3.0 V and less than 5.0 V.
	  ◦ "Low" voltage must be less than 1.2 V.
	• Serial line not sourcing enough current:
	  ◦ The GPS/INS unit must be able to source at least 2 mA in the "high" voltage state.
	• The polarity of the serial and PPS signals.
	• The GPS signal may be obscured by being indoors, near buildings, tree cover, etc. Bring the receiver outside, exposed to as much open sky as possible for best signal reception.
	• If GPS is needed indoors, such as when testing in a garage, considering using a GPS Repeater.

11.1.1 Turned DHCP On, Lost Contact With Sensor

When the sensor's DHCP is set to ON, the configuration saved, and the sensor is reset or power cycled, the sensor will no longer have an IP address as it expects a DHCP server to assign it one.

Because no DHCP server is present, the sensor performs Automatic Private IP Addressing (APIPA) and self-assigns a Class B IP address that looks like 169.254.x.x. Your computer's IP address will likely be very different (not on the same logical network) resulting in a failure to communicate with the sensor.

Perform the following procedure to regain communication with the sensor.

1. Isolate the sensor and your computer on the same network segment.
2. Use any of the following techniques to discover the sensor’s current IP address.
    a. Launch Wireshark with the sensor powered on and connected. Capture UDP packets on the Ethernet interface. You should see packets coming from the sensor with an IP address that looks like 169.254.x.x.
    b. Find the IP address of the sensor using the arp command on your operating system.
        - On Windows, if Cygwin is installed, use the ‘`arp -a`’ command. Cygwin can be obtained from https://www.cygwin.com/.
        - On Mac OS and various linuxes, use the ‘`arp -a`’ command.

        Velodyne Lidar sensors respond to ARP. The arp command displays the current IP-to-physical address translation table. If desired, you can narrow it to a specific interface with the ‘`-N if_addr`’ argument. Look for an address that matches the pattern 169.254.x.x. This will be your sensor's self-assigned address.
    c. Many third-party software tools capable of performing IP scans and ARP sweeps for various operating systems exist. Any of them should be able to discover the sensor’s IP address. However, this may be the least efficient, most time-consuming approach as 169.254.x.x is a large list to scan.
3. Once you have the address of the sensor, change the IP address of the corresponding Ethernet interface on your computer to something on the same logical network segment. For example, if the sensor is found at 169.254.134.79, you can change the computer’s IP address to 169.254.134.222.
4. Now point your browser at the sensor's IP address and you should see its Web Interface. Set DHCP to OFF and check that the sensor's network settings (specifically, IP Address, Mask, and Gateway) are configured properly.
5. Press the **Save Configuration** button.
6. Either power cycle the sensor or reset it (press the **Reset System** button on the System page).
7. Set your computer's IP address and subnet mask to something compatible with the sensor’s address (step 4) and communication will be restored.

11.2 Service and Maintenance

Apart from a small, replaceable automotive fuse in the Interface Box, and keeping the sensor clean (see *Cleaning the Sensor on page 163*), there are no field-serviceable components, maintenance requirements or procedures for the sensor.

For service or maintenance, see *Technical Support on the next page*.

Note: Opening the sensor or compromising the integrity of the sensor housing by modifying it (e.g. drilling holes) will void the warranty.

11.2.1 Fuse Replacement

If the LEDs in the sensor’s Interface Box do not light up when power is provided, check the horizontally-mounted fuse inside. It is designed to blow if a power surge of 7.5 A or greater occurs.

Fuse Specification: **7.5 A Mini ATM Blade Fuse (32 V Max, Color: brown (or red from some makers))**

To replace the fuse:

1. Unplug/disconnect power from the Interface Box.
2. Remove the Interface Box cover with a small Phillips screwdriver, retaining all four screws.
3. Carefully remove the fuse.
4. Replace it with an identical, fresh fuse.
5. Replace the Interface Box's cover and restore power.

11.3 Technical Support

11.3.1 Purchased through a Distributor

IMPORTANT: If you purchased the sensor through a distributor, you should pursue technical assistance through the distributor first.

11.3.2 Factory Technical Support

Follow the steps below to obtain technical assistance from the factory.

1. In your browser, go to the Contact Us page.
2. Click **CONTACT TECHNICAL SUPPORT**.
3. Fill out the form. Required fields are marked with a red asterisk. Optional fields may be left empty.
4. IMPORTANT: In the **Description** field, please describe the problem concisely, including any non-default sensor configuration, cabling, and setup details as necessary.
5. Attachments to the form are limited to a maximum individual file size of 10 GB. No limit is placed on the number of attachments, nor on the total size of all attachments. For very large files, please host them somewhere suitable in the cloud, and include the link to it in the Description field. Bundling files in zip files is fine. (Note that attachment file size limits are more restrictive when using email clients later on in the support conversation; upper limits anywhere from 20 to 30 MB are likely.)
6. Click the **Submit** button.

11.4 Return Merchandise Authorization (RMA)

If your product needs repair for whatever reason, a Return Merchandise Authorization (RMA) Number (starts with 'R') must be issued to you from Velodyne Lidar Customer Service *prior* to shipping the product to the factory for repair.

The Return Merchandise Authorization (RMA) form is online only. To issue an RMA Request, click this link: Request RMA, fill out the form as completely as you can, then click Submit. (The link is also available on our Contact Us page.) After the request is received, a member of the Customer Service team will reach out to you. Once the requirements are satisfied, they will give you the RMA Number (starts with 'R') and provide shipping instructions. Follow the instructions to send the unit in. Once the factory receives the unit, you may later request status of the repair by emailing lidarservice@velodyne.com. Be sure to specify the RMA Number in your email.

One RMA Request per sensor, please. The form has room for just one Serial Number. This allows each unit to be tracked separately.

Chapter 12 • Advanced Packet Format

12.1 Motivation

As Velodyne Lidar sensors evolved, so have applications that use lidar data. Environment awareness applications are capable of processing larger volumes of more richly detailed laser data.

12.2 Audience

This document is intended for application developers tasked with integrating Velodyne Lidar sensor Advanced Packet Format streams into their real-time or batch processing systems.

12.3 Status

This chapter is subject to **REVIEW** by Engineering. It is a derivative of the **Advanced Packet Format Specification** Rev 12, which is under their control.

12.4 Overview

In contrast to our Legacy Packet Format, APF is designed to be flexible. While the lead structure is mandatory and fixed-format, other structures that follow are optional and may vary somewhat.

12.4.1 Design Goal

The pattern set here is designed to be extended by sensors that implement APF in some form or another. Differences are expected to be identified and described by the particular sensor's user manual.

The goal here is to define a pattern that allows for a variety of sensor model-specific format variants that allow each to leverage its full power of data expression.

For example, a sensor that is designed to be short-range only need not pay the penalty of supporting 3-byte distance values.

12.4.2 XML File

It should be noted that base-angles for laser channels are outside the scope of APF. They are intended to be looked up in the sensor's XML file.

12.5 Packet Format

This section specifies the packet format pattern followed by sensors that implement APF datagrams of one type or another. In OOD terms, this is the Class to other sub-classes from which various other packet types inherit. APF-compliant packets either generated by or sent to Velodyne Lidar sensors descend from this description.

12.5.1 Field Names

AZM: Azimuth

DSET: Distance Set

FCNT: Firing Count

FDLY: Firing Delay

FLEN: Firing header Length

FM: Firing Mode

FSPN: Firing Span

GLEN: Group Length

HDIR: Horizontal Direction [0, 1]

HLEN: Header Length

ISET: Intensity Set

LCN: Logical Channel Number

MIC: Model Identification Code

NF: Noise Factor

NXHDR: Next Header

PLEN: Payload Length

PSEQ: Packet Sequence number

PTYPE: Payload Type

PWR: Power

STAT: Status

TLEN: Trailer Length

TOFFS: Time Offset

TREF: Time Reference

VDFL: Vertical Deflection; adjustment applied to base-angle of laser channel found in sensor's XML file

VDIR: Vertical Direction [0, 1]

VER: Protocol Version

12.5.2 Other Terms

GPS
: Global Positioning System

LSN
: Least Significant Nybble

MSN
: Most Significant Nybble

NBO
: Network Byte Order

PTP
: Precision Time Protocol

12.5.3 Byte Order

All multi-byte fields are sent in network byte order. Any sub-byte field will have ordering specified in that field's description.

12.5.4 Advanced Packet Format Composition

The APF-compliant packet is composed of the following elements, in sequence:

- A Mandatory Payload Header -OR- Nominal Payload Header -OR- Sensor-Specific Payload Header (defined in its manual), each of which encapsulates a Mandatory Payload Header at the start
- [Optional] Extension Header(s) and their data
- [Optional] Payload
- [Optional] Payload Trailer

Figure 12-1 Advanced Packet Format (top-level)

Payload Header	Mandatory Payload Header
	Additional PTYPE-specific fields
Extention Header(s)	
Payload	
Payload Trailer	

Each of the elements are described in sections below.

12.5.5 Mandatory Payload Header

The Mandatory Payload Header consists of a minimum set of fields in a specific format which ensure that any following elements can be parsed from the packet, no matter how it varies. It is always 4-bytes in size (i.e. one 32-bit word), and must occur first in an APF packet of any type.

Figure 12-2 APF Mandatory Payload Header

Byte Offset	0		1	2		3
00	VER	HLEN	NXHDR	PTYPE	TLEN	MIC

VER

Size (bits): 4 (MSN)

Protocol Version

- 0, 2-15: Reserved for Velodyne Use
- 1: The current version defined by the specification.

HLEN

Size (bits): 4 (LSN)

Header Length

The number of 32-bit words that comprise the header. The minimum value of this field is 1, corresponding to the size of the Mandatory Payload Header, which is always one 32-bit word in length.

NXHDR

Size (bits): 8

Next Header

The type of the “next header." This value specifies the type of any Extension Header that immediately follows the Payload Header.

- 0: The Payload follows this Header, i.e. no extension header(s) follow.
- 1-65535: Reserved for allocation to extension header types.

PTYPE

Size (bits): 4 (MSN)

Payload Type

This value specifies the type of payload that immediately follows the Payload Header and any Extension Header(s).

- 0: Base Advanced Point Cloud Format (APF) Point Cloud Data.
- 1: Reduced Point Cloud Format (RPF) Point Cloud Data.
- 2-15: Reserved for allocation to future payload types.

TLEN

Size (bits): 4 (LSN)

Payload Trailer Length

The number of 32-bit words of trailing data present at the end of the packet.

For the purposes of this packet format the content of this area is opaque and uninterpreted. The logical payload length is thus: `PacketSize – (PayloadOffset + (4*TLEN))`, where PayloadOffset is the offset into the packet where the Payload starts after the Payload Header and any optional Extension Headers.

If TLEN > 0 the length of the packet must be a multiple of 4 as the trailer starts and ends on a 32-bit word boundary.

MIC

Size (bits): 8

Model Identification Code

This byte identifies the sensor model. The geometry of the LiDAR array for spatial interpretation of the sensor data may be determined by using the MIC.

- 0: Reserved for development.
- 1-255: Reserved for allocation to various sensor types. Consult the Sensor Data section of their sensor user manuals.

12.5.6 APF Nominal Payload Header

The Nominal Payload Header adds Payload Sequence Numbering and Time Reference fields to the Mandatory Payload Header in a standardized location.

Figure 12-3 APF Nominal Payload Header

<table>
<tr><th>Byte Offset</th><th colspan="2">0</th><th>1</th><th colspan="2">2</th><th>3</th></tr>
<tr><td>00</td><td>VER</td><td>HLEN</td><td>NXHDR</td><td>PTYPE</td><td>TLEN</td><td>MIC</td></tr>
<tr><td>04</td><td colspan="6">PSEQ</td></tr>
<tr><td>08</td><td colspan="6" rowspan="2">TREF</td></tr>
<tr><td>0C</td></tr>
<tr><td>...</td><td colspan="6">Additional PTYPE-specific fields</td></tr>
</table>

PSEQ

Size (bits): 32

Payload Sequence Number

Monotonically Increasing sequence number starting at zero when the sensor is started or reset and incre-menting with each payload in the flow.

This field is encoded in Network Byte Order (Big Endian).

TREF

Size (bits): 64

Precision Time Reference

Payload reference time in PTP truncated (TAI, 64-bit) format. It provides a reference time for firing offsets. The semantic relevance of this time is payload type specific.

This field is encoded in Network Byte Order (Big Endian).

12.5.7 APF Extension Headers

The Extension Header consists of a minimum mandatory set of fields which ensure that any following elements can be parsed from the packet. Zero or more of these headers may follow the Payload Header. The final one must have NXHDR set to 0 (zero) to terminate the chain.

Figure 12-4 APF Extension Headers

Byte Offset	0	1	2	3
00	HLEN	NXHDR	DATA + PADDING	
...	...			

HLEN

Size (bits): 8

Header Length

The number of 32-bit words that comprise the header. The minimum value of this field is 1 describing the length of the Mandatory Header Fields.

NXHDR

Size (bits): 8

Next Header

The type of the “next header." This value specifies the type of any Extension Header that immediately follows the Payload Header.

- 0: The Payload follows this Header, i.e. no extension headers follow.
- 1-65535: Reserved for allocation to extension header types.

DATA / PADDING

Size (bits): variable

Extension Data

The data field must end on a 32-bit boundary. The format of the data is extension specific and determined by the NXHDR value of the previous header.

12.5.8 Payload

The Payload starts after the Header containing the NXHDR field set to 0 (zero). The Payload content format is identified by the Payload Type (PTYPE) value in the Payload Header.

The length of the Payload is determined from the overall length of the packet after accounting for the starting offset of the Payload and any optional Trailer length.

12.5.9 Payload Trailer

The Trailer starting offset is referenced to the end of the packet. The start of the Trailer is 4*TLEN octets before the end of the packet. A packet with a Trailer (TLEN > 0) must have an overall length which is a multiple of 4 octets.

The contents of the Trailer are opaque and application specific.

Possible uses of the Trailer are:

- Message Integrity Assertion(s)
- OEM specific data

The APF specification specifically defers to the application for the specification of this content with the only constraints as noted prior on the overall packet length and data alignment.

12.5.10 Packet Flow Encapsulation in UDP

APF packets are typically encapsulated in UDP/IP4 with 802.3 Ethernet framing.

The UDP source port of a packet transmitted by the sensor identifies the Flow Source for the purposes of packet sequence numbering.

Appendix A • Sensor Specifications

A.1 Sensor Specifications

Sensor specifications (including environmental and regulatory specifications) can be found on the Velodyne Lidar web site under this address: https://velodynelidar.com/products/. Visit the product of interest's page, click the *Resources* link, then access the product data sheet.

For additional information, contact Velodyne Lidar Sales: https://velodynelidar.com/contact-us/.

Appendix B • Firmware Update

To update your sensor's firmware, follow the procedure below. If a problem occurs, follow the steps in *If An Error Occurs on page 125*.

B.1 Firmware Update Procedure

Before starting the procedure, make sure your computer can communicate with the sensor over Ethernet.

The first part of the procedure checks if the sensor needs an update. The second part (beginning with step 6) performs the update in two stages.

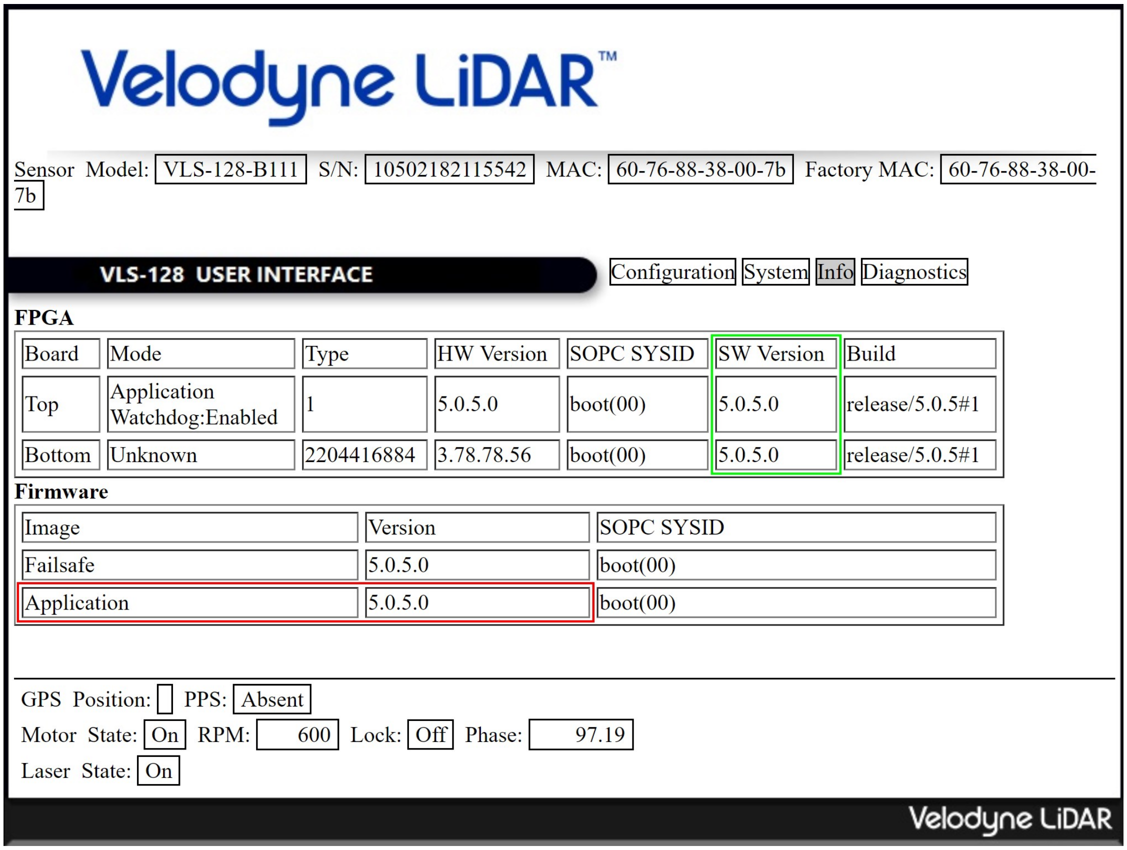
This procedure works best with the Chrome browser.

1. Point your browser to the Velodyne Downloads page at https://velodynelidar.com/downloads/.

Figure B-1 Velodyne Downloads Page

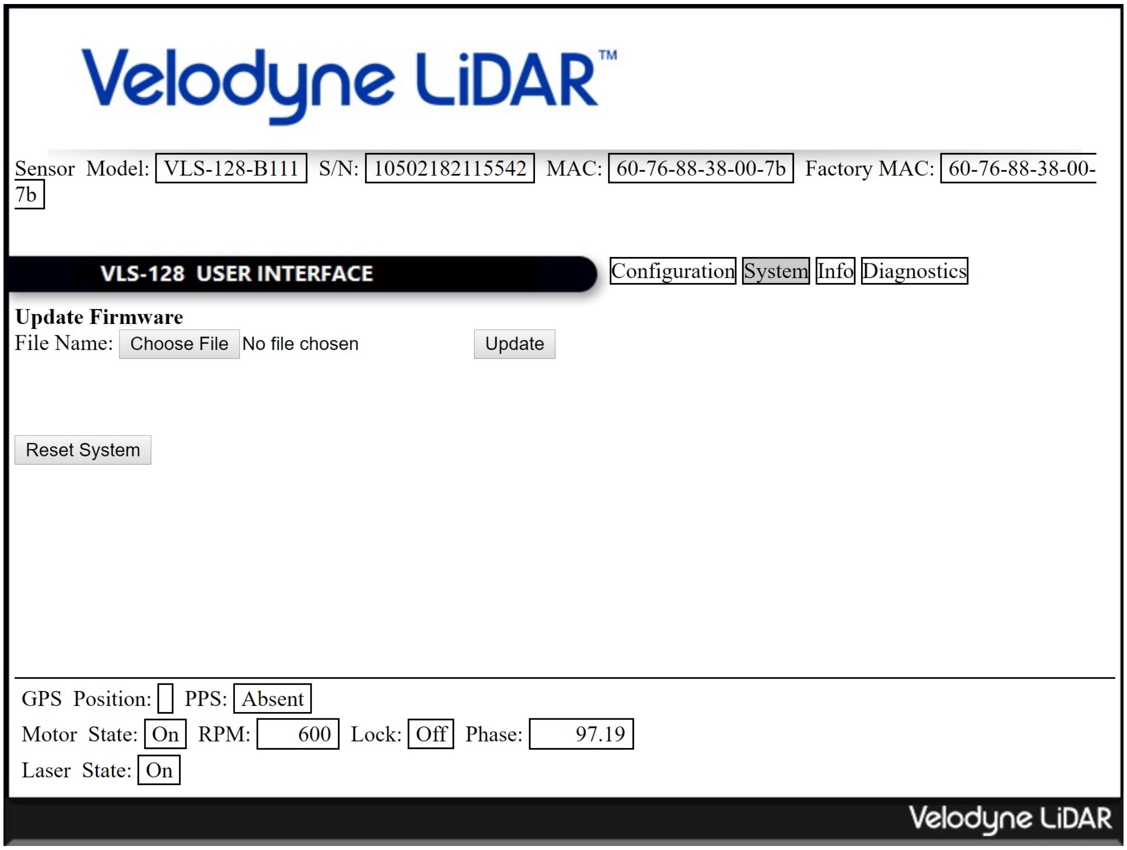
2. Click the FIRMWARE icon (on the left) to advance to the Firmware section. Note the latest version available.
3. Open the sensor's Web Interface in your browser and click the **Info** button.
4. Check the Firmware Application Version on the **Info** page (red box in the figure below). This is the active firmware on the sensor.
5. Compare the versions. If they differ, download the newer firmware version to your computer from the Downloads page, and continue the procedure. If not, stop here.

Figure B-2 Compare Firmware Versions



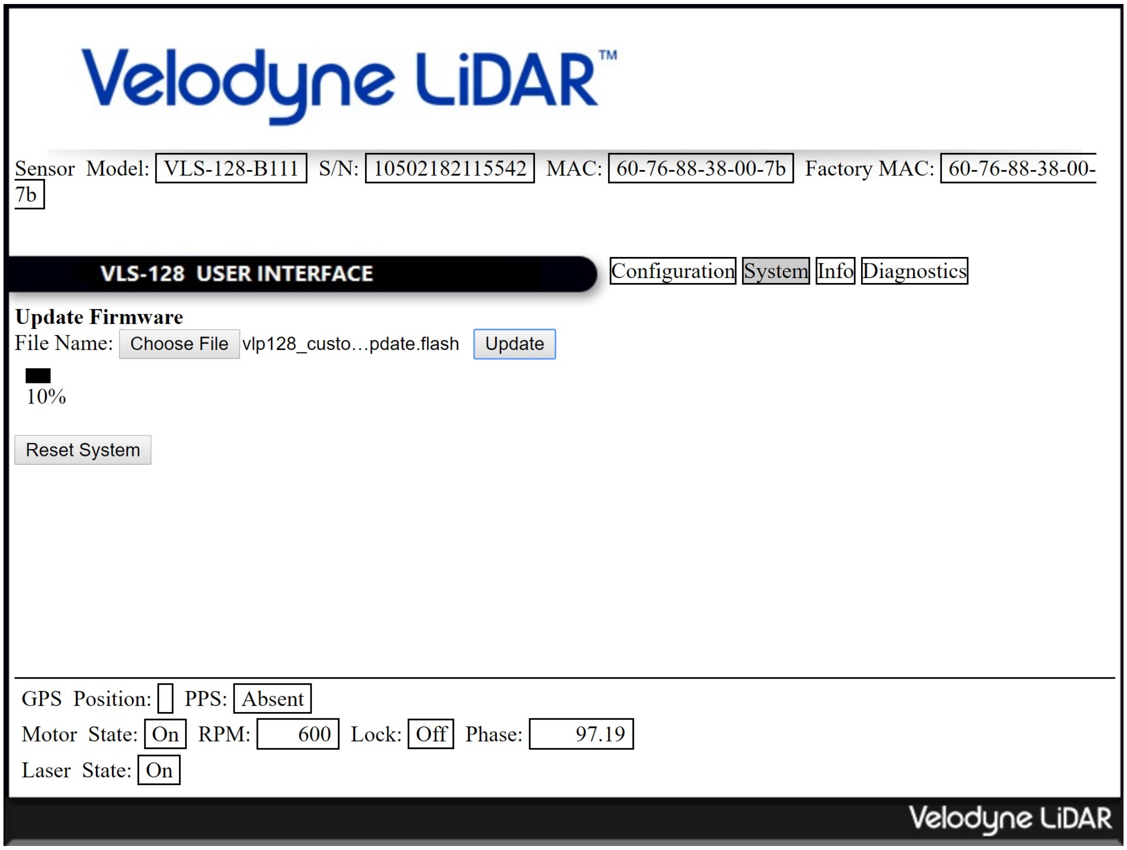
Note: The firmware image labeled “Failsafe” is the original image programmed into the sensor when it was manufactured. It cannot be updated.

Figure B-3 Select New Firmware Image



6. Click the **System** button.
7. Click the **Choose File** button under **Update Firmware**.
8. Using the dialog, locate the new firmware file on your computer and select it.
9. Verify that the correct file for your sensor model has been selected. It must have the '.flash' file extension.
10. Click **Update Firmware**'s **Update** button to initiate the update process.

Figure B-4 Upload New Firmware Image



11. As shown in the figure above, a progress bar indicating the percentage of update completion is displayed.
12. When this first stage of the update is complete, the **Firmware Update Complete** screen (below) is displayed. *But wait, there's more.* Internally, you've updated the bottom board but not the top, yet.

Figure B-5 Firmware Update Complete Page

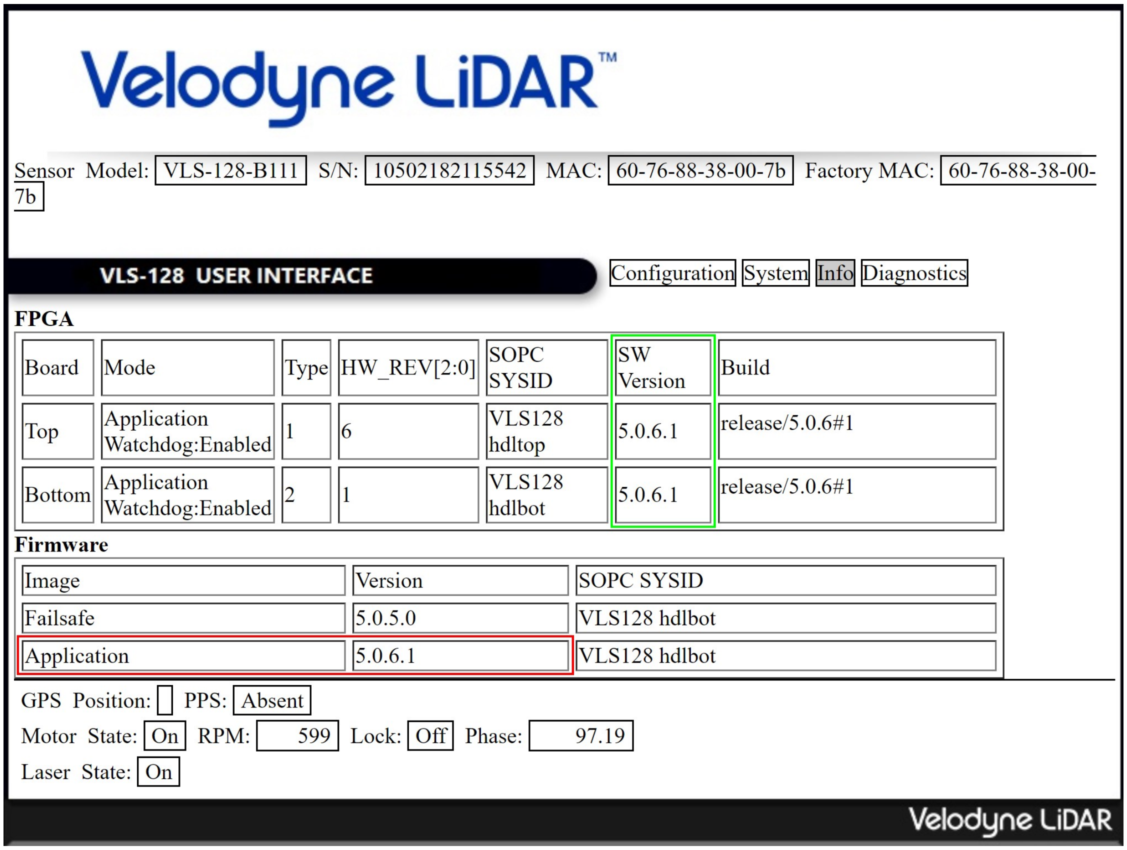
13. Now click the **Process Firmware Update** button to initiate the second stage of the procedure. The firmware image that you just uploaded is being prepared for use by the sensor.
14. A second progress bar is shown.

Note: This step should take up to a few minutes. If it completes in less than 10 seconds, an error occurred. Instead of continuing, perform the *If An Error Occurs on page 125* procedure.

Figure B-6 Finalize Firmware Update

15. When processing completes, press the **Reset System** button to finalize the update. The sensor will perform a warm boot.
16. The **Configuration** screen will be shown when the sensor finishes rebooting.
17. Return to the **Info** page.

Figure B-7 Verify Firmware Versions



18. Verify that the **Firmware Application Version** (red box) matches the version of the firmware downloaded and installed. Also verify that the same version appears in the **SW Version** column for both the Top and Bottom boards (green box). If everything matches, the procedure completed successfully.

Any mismatches indicate an error occurred in the last stage of the update. (Occasionally, the screen may indicate that "Firmware Update Processing is not available in Failsafe mode." In Failsafe mode, the lasers and motor are disabled.) Perform the If An Error Occurs on the facing page *procedure below to resolve the issue.*

B.1.1 Special Procedure to Update Firmware

The firmware update process requires both the top and bottom boards be flashed. A minor tweak is made to the steps normally used to update both boards. When this tweak is made, the normal procedure becomes the "Special Procedure to Update Firmware."

1. First, go to the **System** tab.
2. Click on **Choose File** and specify the flash file name.
3. Press **Update** to update the bottom board.
4. Wait a bit.
5. You will eventually see another screen entitled, "Firmware Upload Complete."

6. Do not press the **Process Firmware Update** button to flash the top board. **This is where the process deviates from the normal procedure.**
7. Instead, reset the sensor, either by pressing the **Reset System** button or cycling power to the sensor.
8. The sensor boots up. At this point, you do not have to run the update process for the bottom board again. Go to the **System** tab and click on the **Update** button associated with 'Update Firmware'. Do not specify a flash file name.
9. After several seconds, a page for updating the top board's firmware appears.
10. Press the **Process Firmware Update** button to update the top board's firmware.

Note: Sometimes the update process for the top board times out and eventually fails. When this happens, reset the sensor or cycle power to it. You will see explicit instructions to cycle power to the sensor if the case warrants it.

---

11. Wait a bit.
12. Reset sensor for the new firmware version to take effect.
13. Return to the **Info** page and verify that the **Firmware Application Version** and **SW Versions** are as expected (*Figure B-7 on the previous page*).

B.1.2 If An Error Occurs

If an error occurs during the Firmware Update Procedure, it did not finish. Perform the steps below to complete the process.

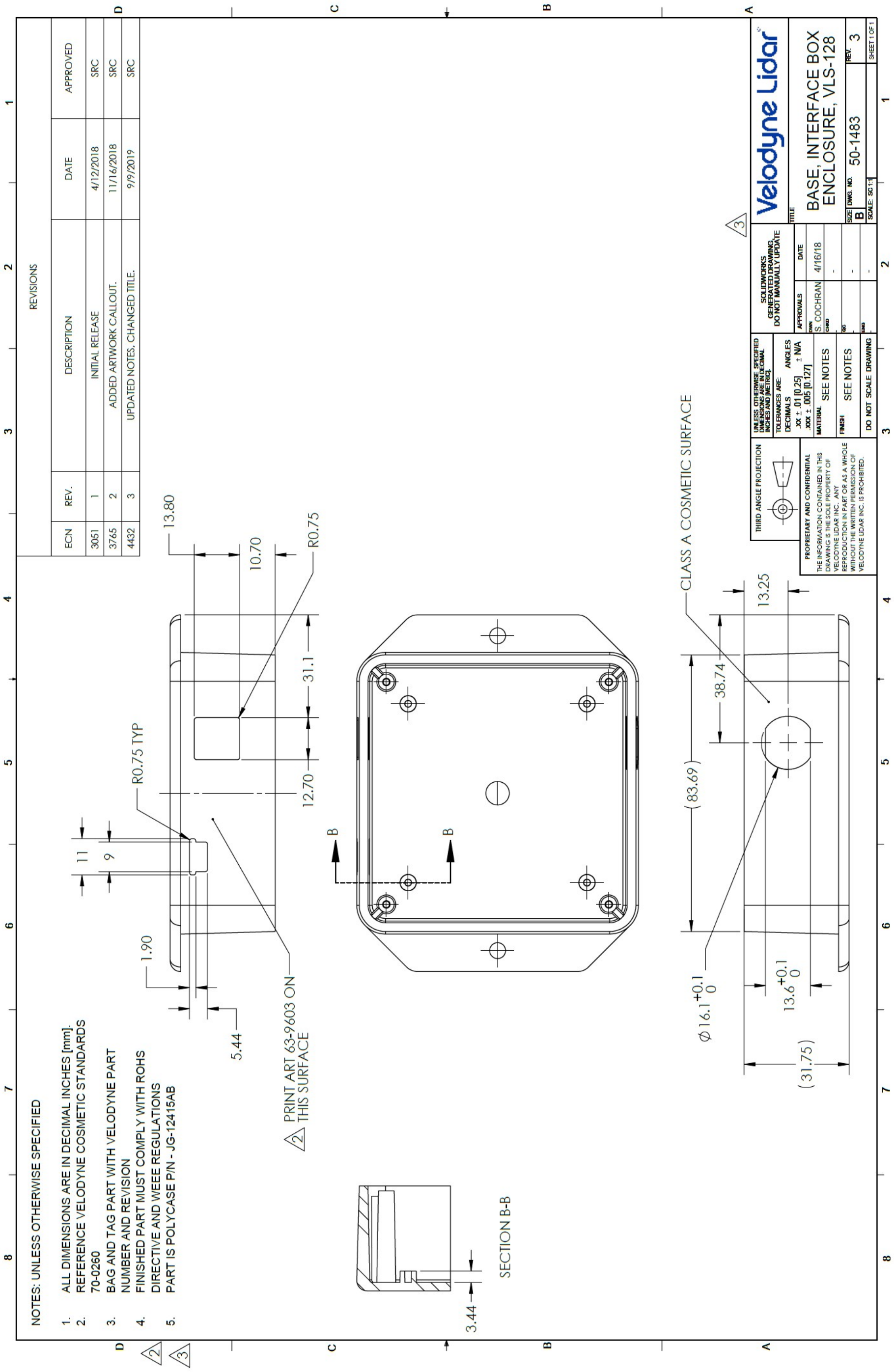
1. Reset the sensor. Either power cycle it (remove power, wait at least 10 seconds, apply power, wait 30 seconds), or click the **Reset System** button if you're on the sensor's **Firmware Update Complete** screen or can get to the sensor's **System** page. The **Configuration** screen will be shown when the sensor finishes rebooting.
2. Go back to the sensor's **System** page (*Figure B-3 on page 120*).
3. In the **Update Firmware** section, just click the **Update** button (*Figure B-4 on page 121*). A progress bar is briefly displayed.
4. When complete, the **Firmware Update Complete** screen is displayed (*Figure B-5 on page 122*).
5. Click the **Process Firmware Update** button. A second progress bar is displayed. This step should take anywhere from 70 seconds up to a few minutes (*Figure B-6 on page 123*).
6. When processing completes, press the **Reset System** button to finalize the update. The sensor will perform a warm boot. The **Configuration** screen will be shown when the sensor finishes rebooting.
7. Return to the sensor's **Info** page.
8. Verify that the **Firmware Application Version** (red box in *Figure B-2 on page 119*) matches the version of the firmware downloaded and installed. Also verify that the same version appears in the **SW Version** column for both the Top and Bottom boards (green box in *Figure B-2 on page 119*). At this point they should all match.

Appendix C • Mechanical Diagrams

This appendix provides technical drawings and diagrams of mechanical assemblies. High resolution versions may be accessed on the Velodyne Lidar web site.

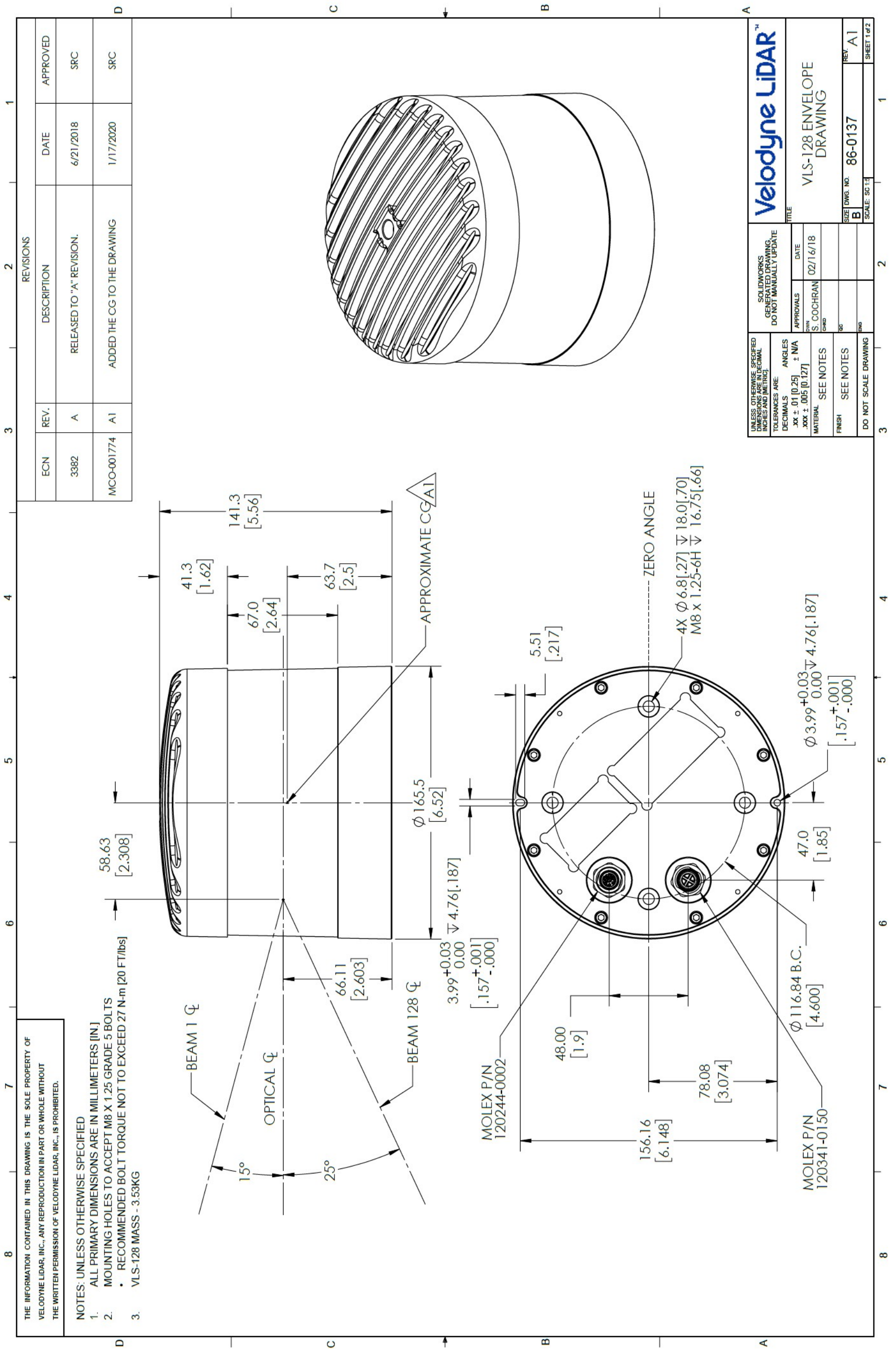
C.1 Interface Box Mechanical Drawing

Figure C-1 Interface Box Mechanical Drawing 50-1483 Rev 3



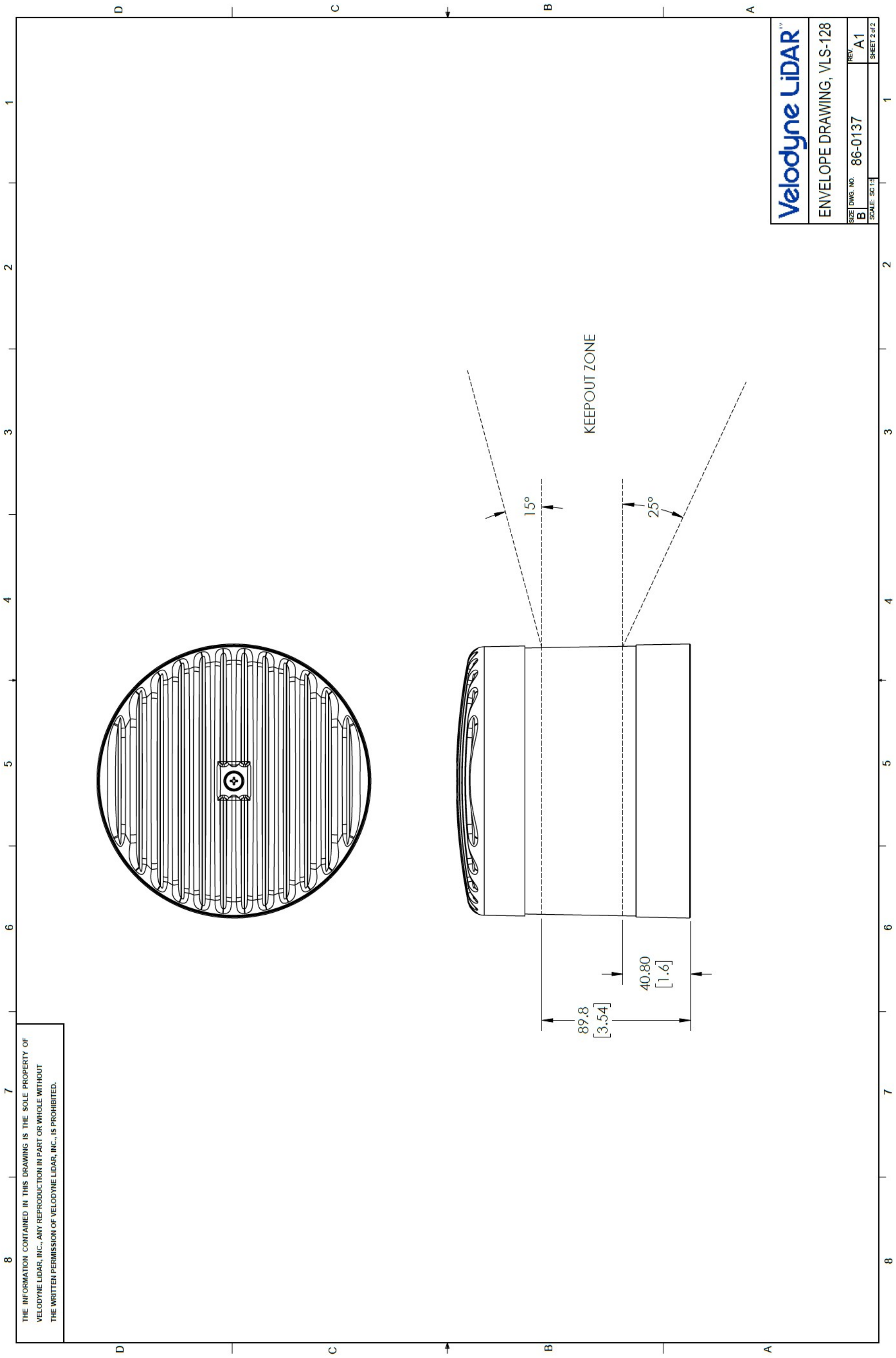
C.2 VLS-128 Mechanical Drawing

Figure C-2 VLS-128 Mechanical Drawing 86-0137 REV A1



C.3 VLS-128 Optical Keep Out Zone

Figure C-3 VLS-128 Optical Keep Out Zone 86-0137 REV A1



Appendix D • Wiring Diagrams

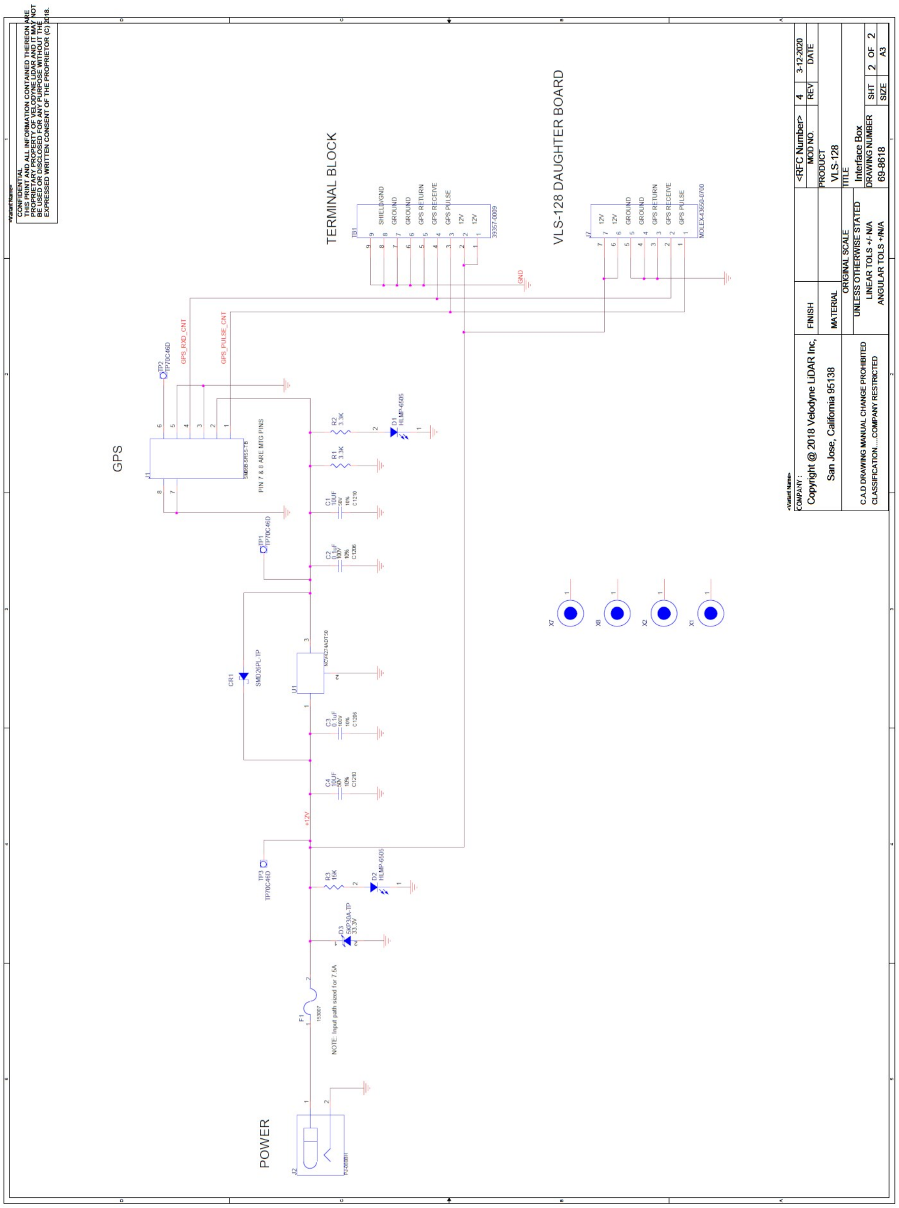
This appendix provides technical wiring and schematic drawings and diagrams. High resolution versions may be accessed on the Velodyne Lidar web site.

D.1 Interface Box Wiring Diagram

A wiring diagram specifically for the Alpha Prime does not exist. Instead, the relevant information can be found in *Interface Box Signals on page 44*.

D.2 Interface Box Schematic

Figure D-1 Interface Box Schematic 69-8618 Rev 4



Appendix E • VeloView

This appendix gets you started with VeloView. It shows you how to acquire and install the program, and visualize, save, and replay sensor data.

You can examine sensor data with other free tools, such as Wireshark (available from wireshark.org) or tcpdump. But to visualize the 3D data, we recommend that you use VeloView. It is free and relatively easy to use.

Other software packages that include Velodyne sensor visualizers are: ROS, PCL, Apollo, and Autoware. Hundreds of opensource Velodyne-related repos can be found on GitHub.

E.1 Features

VeloView provides real-time visualization of 3D lidar data from Velodyne Lidar sensors. VeloView can also playback pre-recorded data stored in “pcap” (Packet Capture) files.

Note: VeloView does not support .pcapng files. If you have such a recording, open it with Wireshark then use the **File** > **Save As** > **Save PCAP** feature to save a copy in the pcap format.

VeloView displays distance measurements from a Velodyne Lidar sensor as point data. It supports custom-colored display of variables such as intensity-of-return (i.e. calibrated reflectivity), time, distance, azimuth, and laser ID. The data can be exported in CSV format.

Functionality and features include:

- Visualize live streaming sensor data over Ethernet
- Record live sensor data in pcap files
- Visualize sensor data from a recording (pcap file)
- Interpret point data such as distance, reflectivity, timestamp, azimuth, laser ID, etc.
- Tabular laser measurement data inspector
- Export columnar data to CSV format
- Record and export GPS data (in position packets)
- Ruler tool
- Display multiple frames of data simultaneously (Trailing Frames)
- Display or hide subsets of lasers
- Crop views
- Built-in Python 3.9.5 console
- Multi-sensor support

Note: VeloView is not intended to generate point cloud files in XYZ or certain other formats (see *Converting PCAP Files to Point Cloud Formats on page 83*). See the list of Velodyne system integrators at https://velodynelidar.com/automated-with-velodyne/ for vendors who can provide you with more advanced imaging software or a complete mapping system.

E.1.1 Learning VeloView Basics

VeloView comes with built-in documentation. Hit the F1 key to open the *VeloView Users Guide*. The guide dives more deeply into certain aspects than this appendix, so, reading it is recommended, particularly for advanced users.

Hover your cursor over a user interface tool to see its tool-tip. Right-clicking the toolbar opens a context menu.

E.2 Install VeloView

Installers for VeloView for Windows (64-bit only) and Macintosh computers may be found on the USB stick included with your sensor if one was provided. If not, you can follow the steps below to install from the web site (which includes an x86 linux version).

1. Point your browser to https://www.paraview.org/VeloView/.
2. Scroll down to the **User Instructions** section.
3. Select the correct binary installer or tarball for your operating system; click the link.
4. Save the executable installer to disk.

On Windows, launch the installer and follow the on-screen instructions.

On Mac, launch the installer and follow the on-screen instructions. You may have to use Ctrl+Right Click to open it.

On linux, unpack the tarball. Move the tree as desired. You'll find the VeloView executable in the *bin* subdirectory and the VeloView User Guide pdf in the *doc* subdirectory. You may add the bin path to your PATH environment variable if you like.

Multiple VeloView versions may co-exist on a given computer.

E.2.1 Association with a Different GPU

On underpowered computers with a discreet GPU installed, it may improve performance to manually associate VeloView (or whichever visualizer you choose) with a specific graphics processor. Or, you may have a laptop which defaults programs to run on the embedded graphics processor, leaving the high-performance GPU relatively idle. This assumes you have a machine with a choice of GPUs. Exactly how you create the association depends on your computer's configuration (software and hardware).

Here's one example: Windows 10 laptops with embedded Intel GPU and Nvidia GPU graphics card. VeloView often installs with the embedded GPU selected. To associate it with the Nvidia graphics card, first, run VeloView then exit. Launch Nvidia Control Panel. Select 'Manage 3D settings' then select Program Settings. Click the Add button. Select the VeloView in the list that you launched, then click the 'Add Selected Program' button. Next, select 'High-performance Nvidia processor' in the #2 pull-down. Click the Apply button. Rebooting at this point is recommended but not necessary. Launch VeloView. It should be assigned to the Nvidia graphics card for rendering acceleration, offloading the heavy lifting from your CPU.

E.3 Visualize Streaming Sensor Data

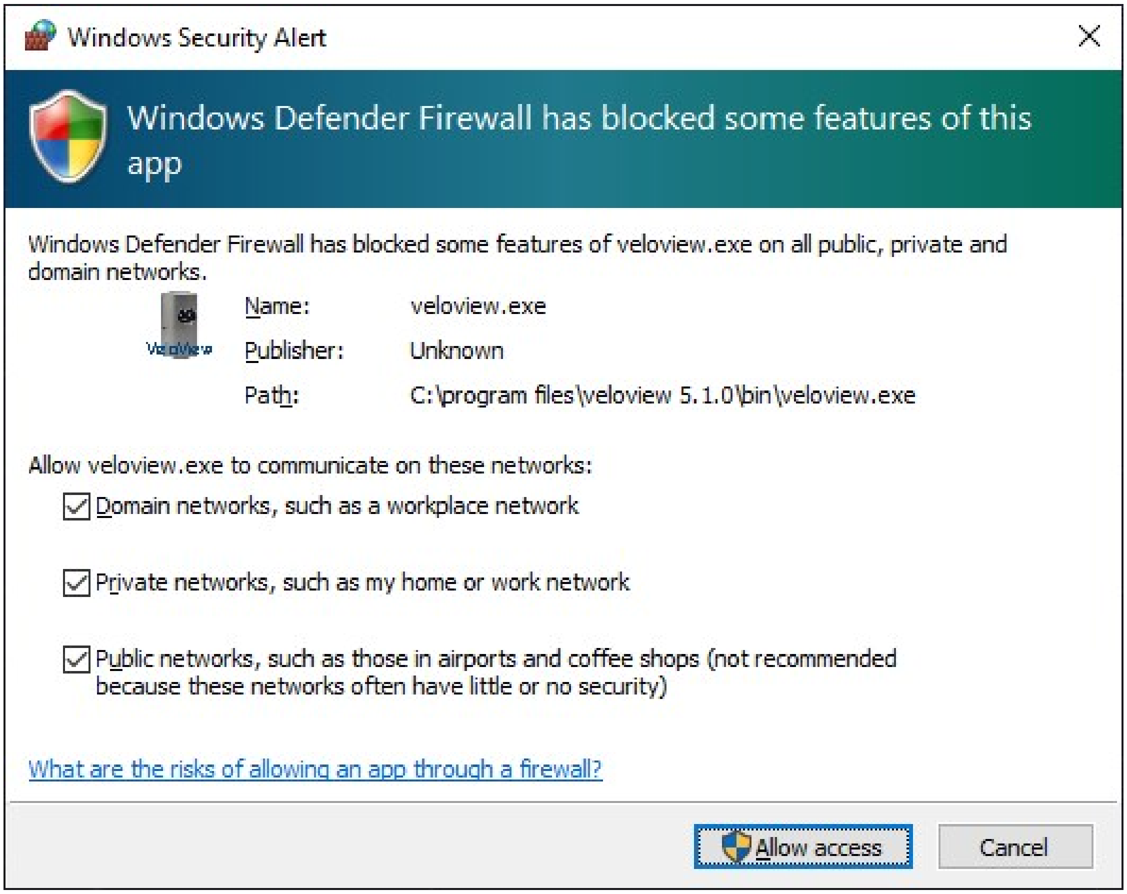
1. Setup access to your sensor over Ethernet as described in *Unboxing & Verification on page 26* or *Installation & Integration on page 33* and power it up.
2. Start the VeloView application.
3. Click on **File** > **Open** and select **Sensor Stream**.

E.3.1 Windows-Only: Getting Past the Firewall

At this point, Windows Firewall may interrupt the proceedings by displaying a Windows Security Alert dialog box with two or three checkboxes. This happens only if the firewall is enabled -- and it only asks *once*. The first time that particular VeloView accesses a network port, the firewall asks for permission to allow communication on [ ] public networks, [ ] private networks, and perhaps [ ] domain networks (such as a workplace network), via that interface.

IMPORTANT: Checkmark all checkboxes then click the Allow Access button.

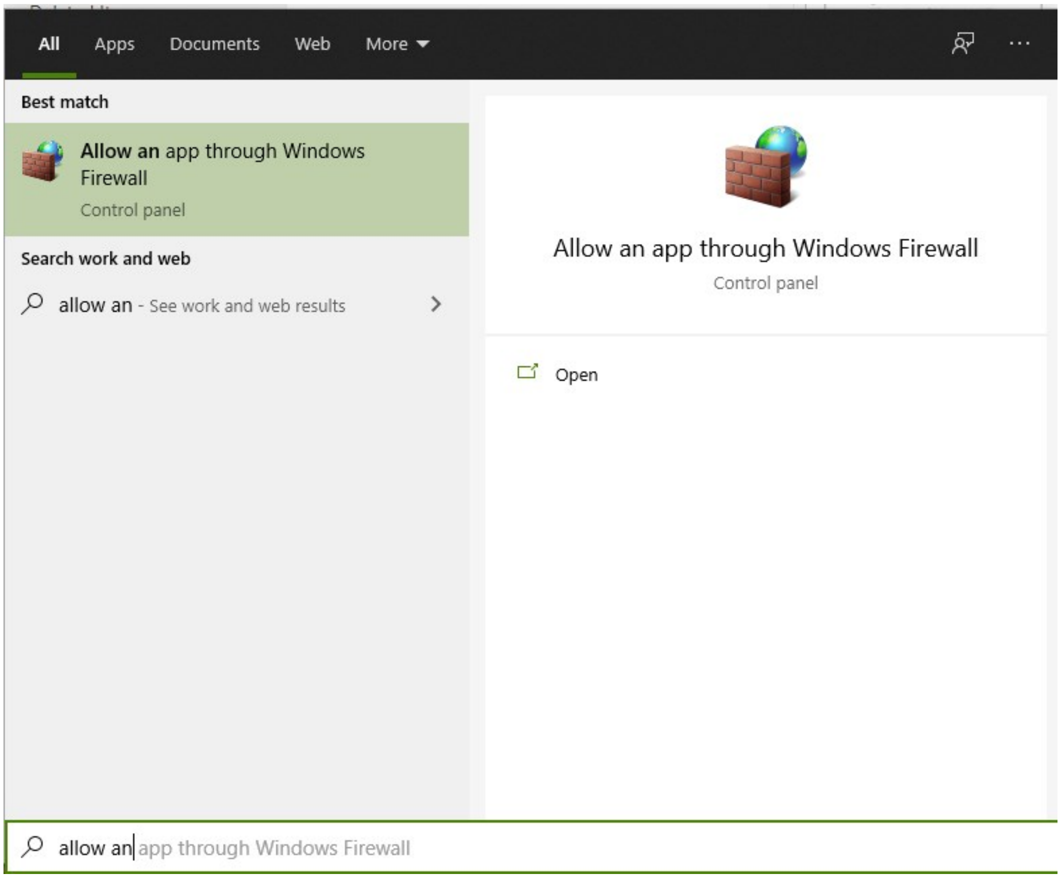
Figure E-1 Allow VeloView to Communicate on These Networks



If you do not checkmark all boxes, the firewall will block communications with the sensor. You will get no data from the sensor.

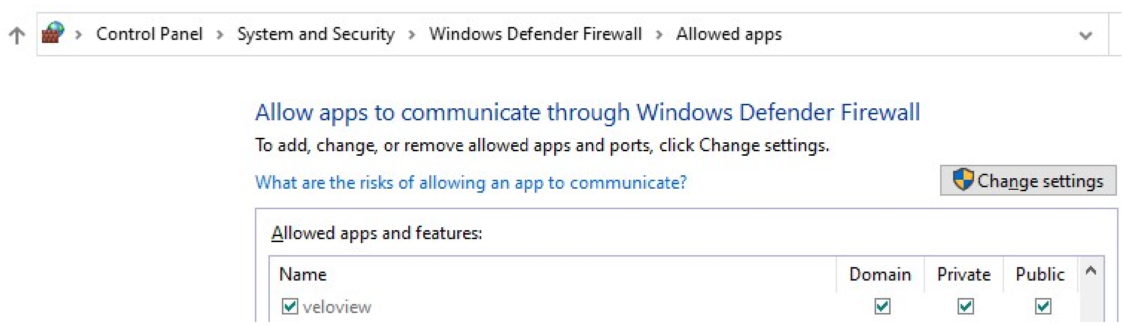
To resolve that particular situation, select 'Allow an app through Windows Firewall' in the Control Panel (see *Figure E-2 on the next page* below).

Figure E-2 Allow an App Through Windows Firewall



The 'Allowed apps' dialog will be displayed. Click the 'Change settings' button to enable changes. Next, scroll down to 'veloview' and click on the entry. Make sure it has a checkmark in front of it so its rules are enabled. Now put checkmarks in each column to the right of veloview, then click OK to make the changes. It should look like *Figure E-3 below*. Once done, you should now be able to see live sensor data in VeloView.

Figure E-3 Enable VeloView to Communicate Through Windows Firewall



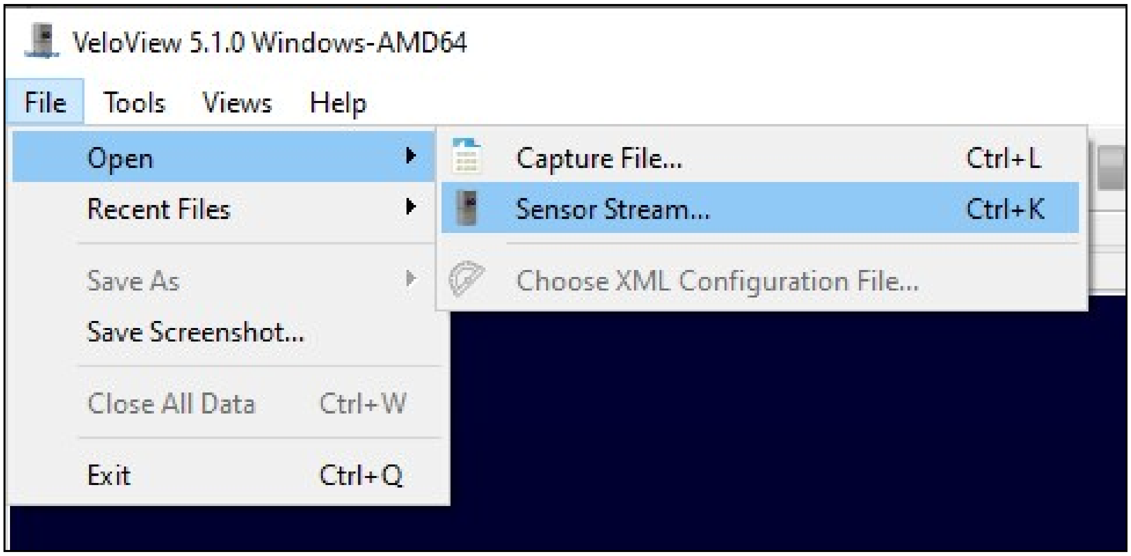
An alternate method would be to uninstall VeloVew then reinstall it, but this is more drastic. Once done, the next time you open a stream to a sensor, when the Windows Security Alert dialog appears, be sure to put checkmarks in all the boxes.

Click 'Allow access' to have the firewall create the appropriate rules. You should now be able to see live sensor data in VeloView.

Or, you could disable the firewall completely, but that is not recommended.

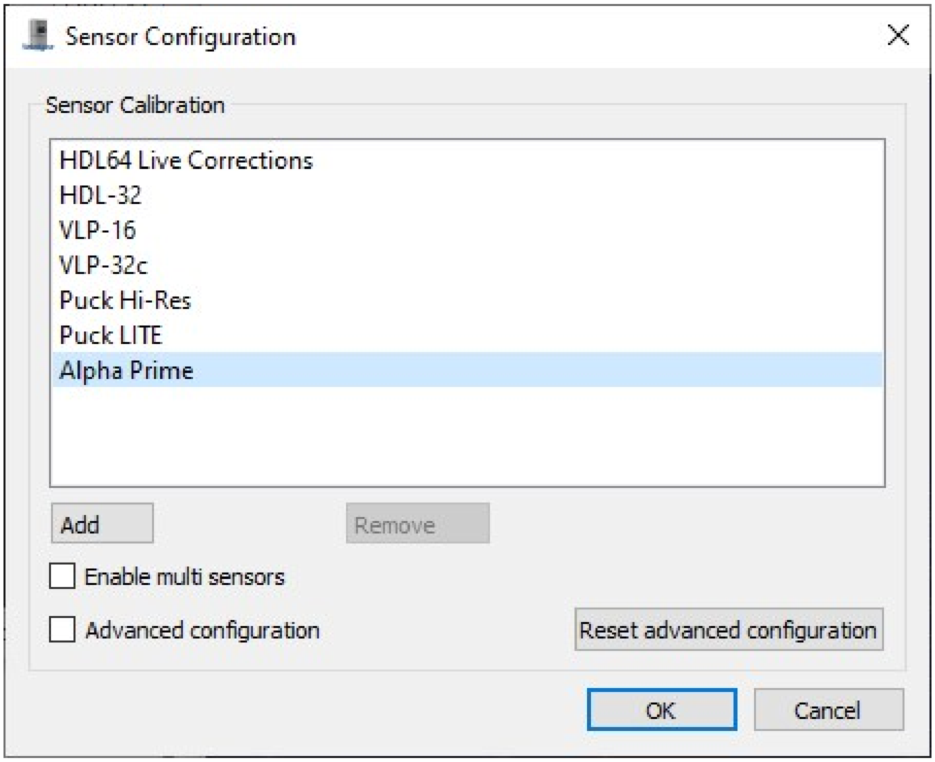
E.3.2 Select the XML File

Figure E-4 Open Sensor Stream



4. The Sensor Configuration dialog will appear. Select your sensor type then click **OK**.

Figure E-5 Select Sensor Calibration (XML) File

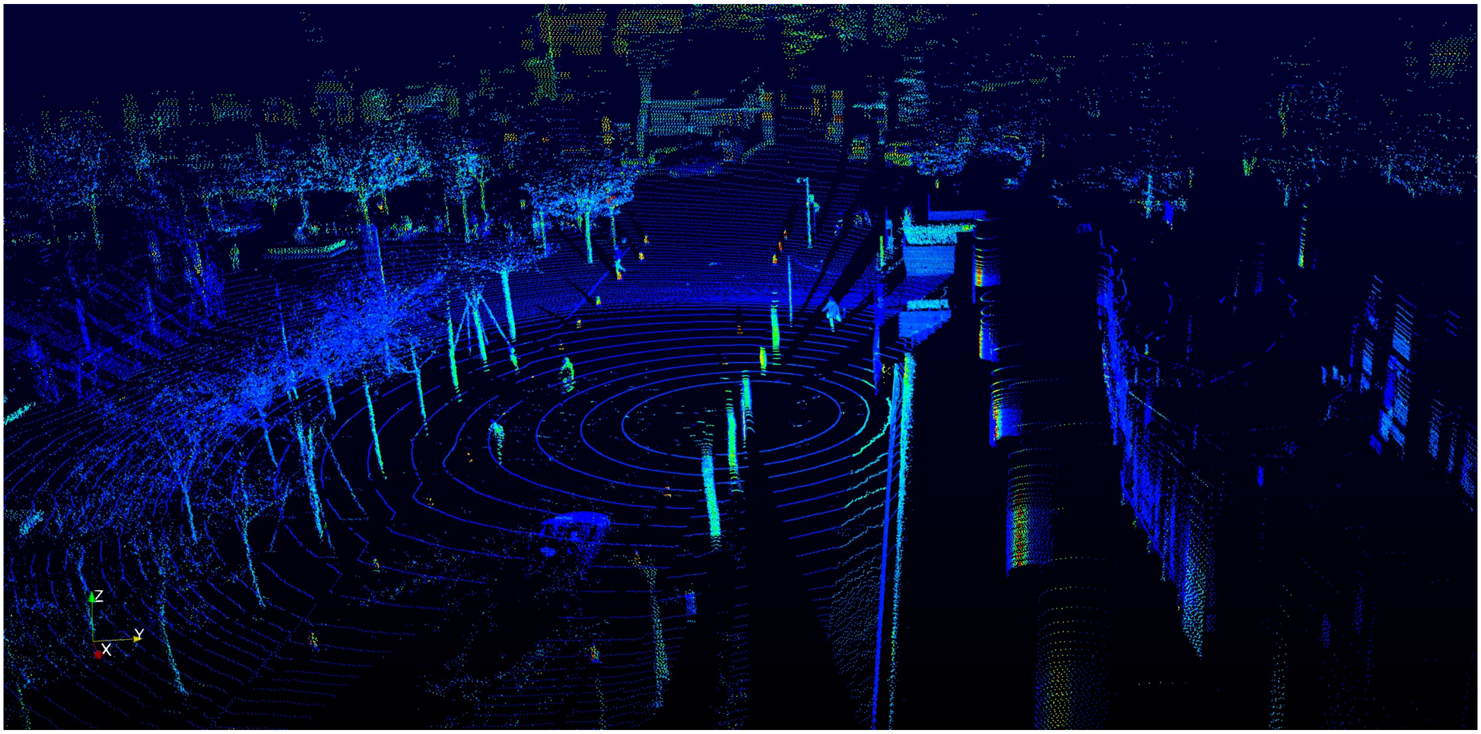


For APF operation, a separate xml is required. See *Precision Azimuth Calculation (HDL) on page 71* and *XML File on page 84* for details.

E.3.3 Live Stream

5. VeloView begins displaying the sensor data stream.

Figure E-6 VeloView Sensor Stream Display



The live stream can be paused by pressing the *Play/Pause button on page 141*. Press it again to resume streaming from that moment on.

E.3.4 Navigating Within a Frame

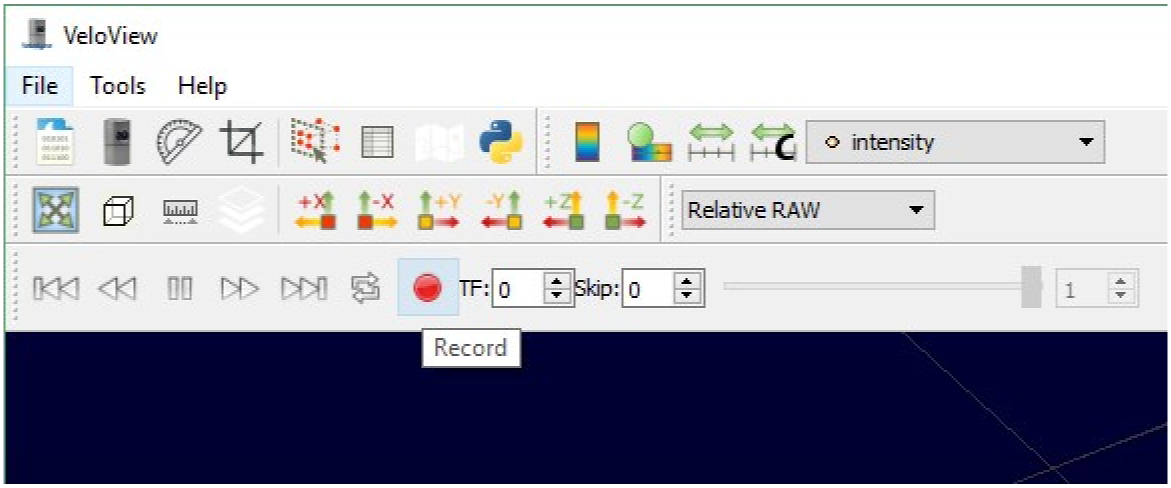
You can move the virtual camera around with either mouse button, the mouse wheel, or with touch controls on a track-pad. You can shift the entire image around using the mouse wheel as a button. Hold it down on the display to move the image. You can easily untilt an image by holding down the Shift key while manipulating the image with the left button. More tips can be found in the VeloView user guide built into the program.

The live stream can be paused by pressing the *Play/Pause button on page 141*. Press it again to resume streaming from that moment on.

E.4 Capture Streaming Sensor Data to PCAP File

1. Click the **Record** button.

Figure E-7 VeloView Record Button



A **Choose Output File** dialog will pop up with the file name field pre-loaded with something like 2022-08-07-08-30-50_ Velodyne-Alpha Prime-Data.pcap. You may alter it if you like.

2. Navigate to where you want the file to be saved and click the **Saved** button.

VeloView begins writing packets to your pcap file.

3. Recording will continue until the **Record** button is clicked again, which stops the recording and closes the pcap file.

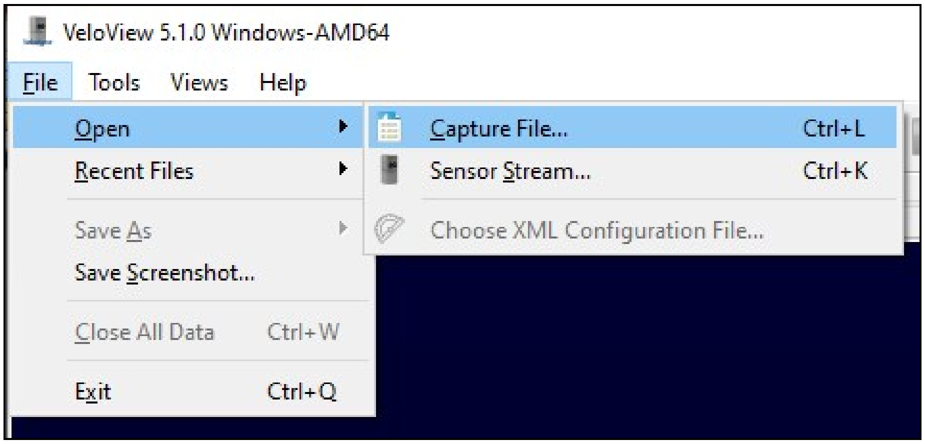
Note: Velodyne Lidar sensors generate a lot of data. (See *Throughput Calculations on page 54* for data rates.) The pcap file can become quite large if the recording duration is lengthy. Also, it is best to record to a fast, local HDD or SSD, not to a slow subsystem such as a USB storage device or network drive.

E.5 Replay Captured Sensor Data from PCAP File

To replay (or examine) a pcap file, open it with VeloView. You can press Play to let it run, or scrub through the data frames with the Scrub slider. Select a set of 3D rendered data points with your mouse and examine the numbers with a Spreadsheet sidebar.

1. Click on **File** > **Open** and select **Capture File**.

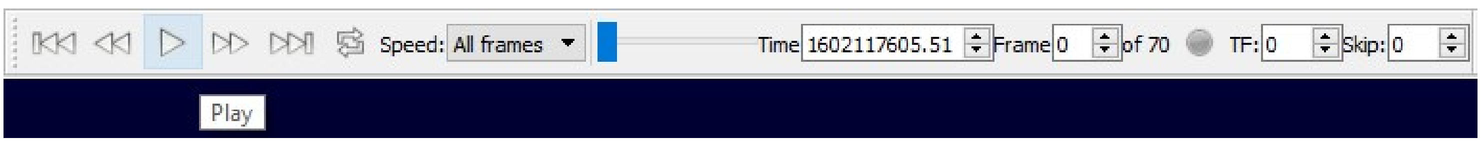
Figure E-8 VeloView Open Capture File



2. An **Open File** dialog will pop up. Navigate to a pcap file, select it, and click the **Open** button. The Sensor Configuration dialog will pop-up.
3. Select your sensor type and click **OK**.

VeloView should display frame 0.

Figure E-9 Play/Pause button

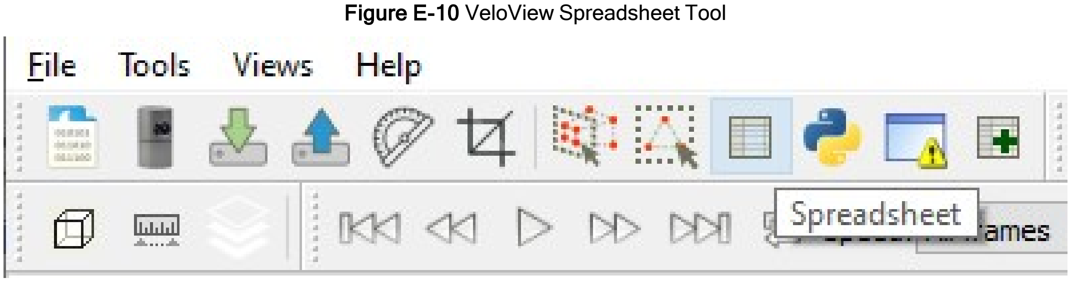


4. Press **Play** to replay/pause the data stream. Use the **Scrub** slider tool (it looks like an old-fashioned volume slider) to move back and forth through the data frames. Both controls are in the same toolbar as the Record button.

E.6 Inspecting a Frame of Data

5. To take a closer look at some data, scrub to an interesting frame and click the **Spreadsheet** button.

Figure E-10 VeloView Spreadsheet Tool

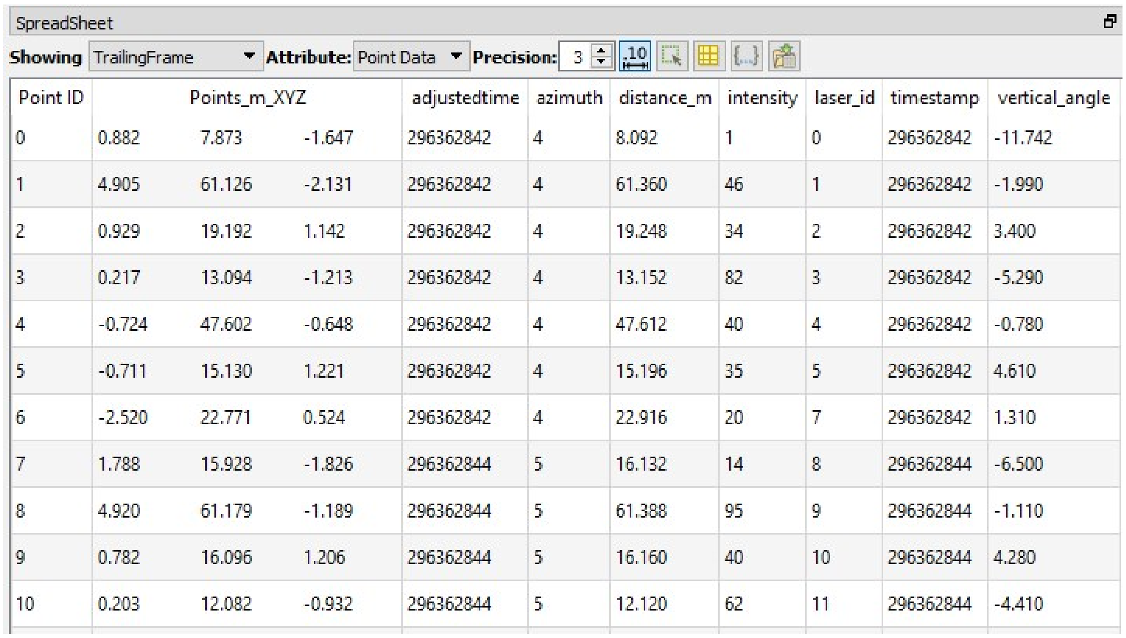


A sidebar of tabular data is displayed to the right of the rendered frame, containing all data points in the frame.

6. Left-click the 'Toggle column visibility' control to enable/disable various columns. Adjust the columns to get a better view of the numbers. If you've adjusted columns in Excel, some of this will be familiar. You can change

column widths by dragging the column header divider left or right, and by double-clicking them. Drag column headers left or right to reorder them. Sort the table by clicking column headers. And you can make the table itself wider by dragging the table's sides left or right. Make the Points_m_XYZ (aka Points) column wider to expose the component X|Y|Z values (expressed in meters).

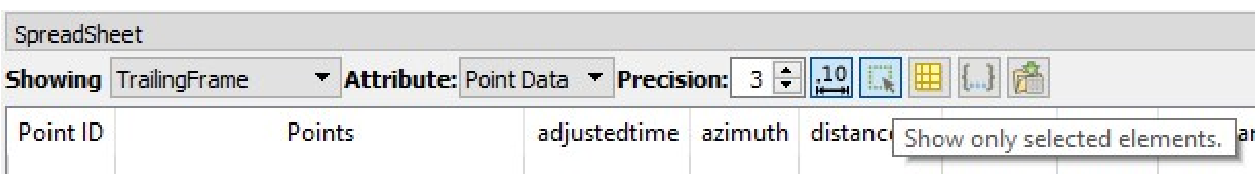
Figure E-11 VeloView Data Point Table



Point ID	Points_m_XYZ			adjustedtime	azimuth	distance_m	intensity	laser_id	timestamp	vertical_angle
0	0.882	7.873	-1.647	296362842	4	8.092	1	0	296362842	-11.742
1	4.905	61.126	-2.131	296362842	4	61.360	46	1	296362842	-1.990
2	0.929	19.192	1.142	296362842	4	19.248	34	2	296362842	3.400
3	0.217	13.094	-1.213	296362842	4	13.152	82	3	296362842	-5.290
4	-0.724	47.602	-0.648	296362842	4	47.612	40	4	296362842	-0.780
5	-0.711	15.130	1.221	296362842	4	15.196	35	5	296362842	4.610
6	-2.520	22.771	0.524	296362842	4	22.916	20	7	296362842	1.310
7	1.788	15.928	-1.826	296362844	5	16.132	14	8	296362844	-6.500
8	4.920	61.179	-1.189	296362844	5	61.388	95	9	296362844	-1.110
9	0.782	16.096	1.206	296362844	5	16.160	40	10	296362844	4.280
10	0.203	12.082	-0.932	296362844	5	12.120	62	11	296362844	-4.410

7. Click **Show only selected elements**.

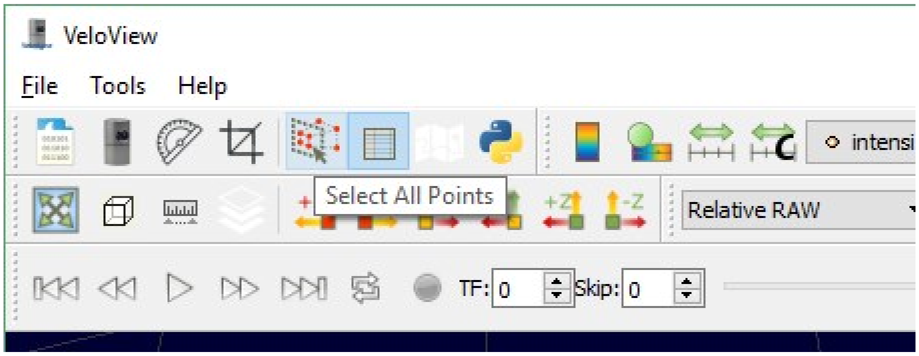
Figure E-12 VeloView Show Only Selected Elements



Since no points are selected yet, the table will be empty.

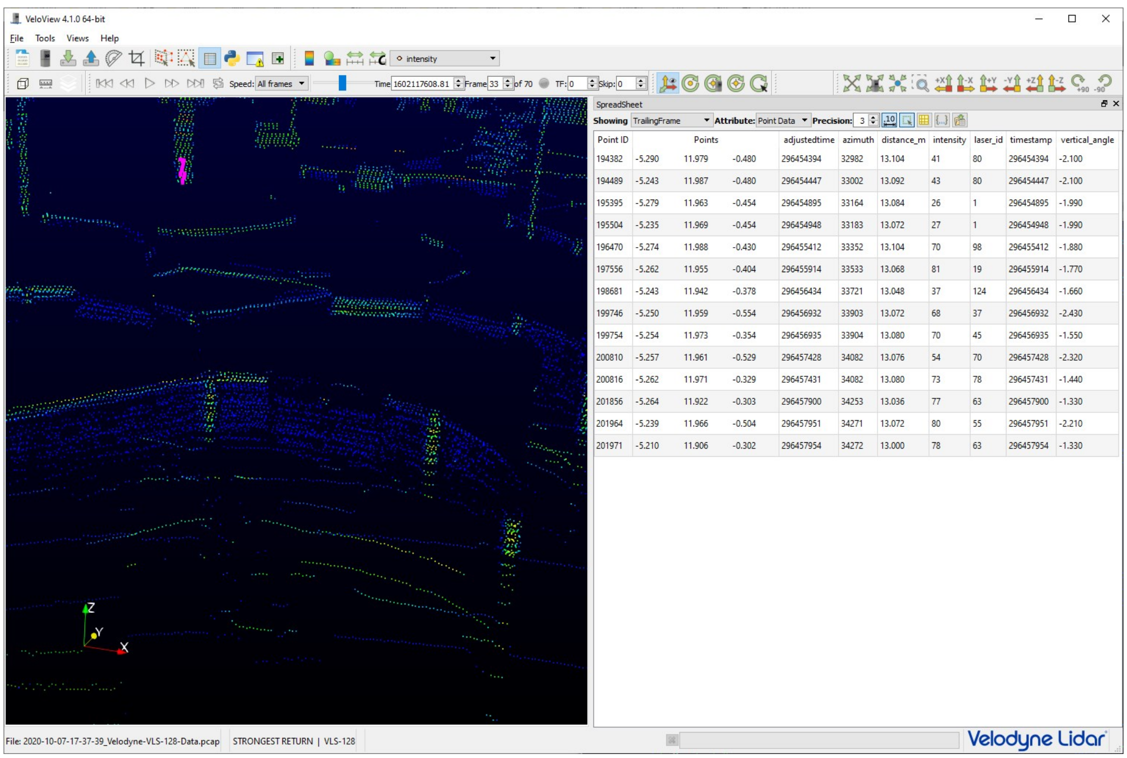
8. Click the **Select All Points** tool. This turns your mouse into a point selection tool.

Figure E-13 VeloView Select All Points



9. In the 3D rendered data pane, use your mouse to draw a rectangle around a small number of points. They will immediately populate the data table.

Figure E-14 VeloView List Selected Points



In *Figure E-14 above*, 14 data points were selected with a small selection rectangle. The points appear in pink. Their attributes are listed on the right. In this example, their intensity values are below 110, indicating a diffuse, non-retroreflective surface (see *Calibrated Reflectivity on page 36* for more).

Azimuth is in hundredths of a degree. Distances are with respect to the sensor's origin.

While timestamp is reset to 0 at the top of the hour (TOH), adjustedtime continues running. It rolls over to 0 once it reaches 2^{32}-1.

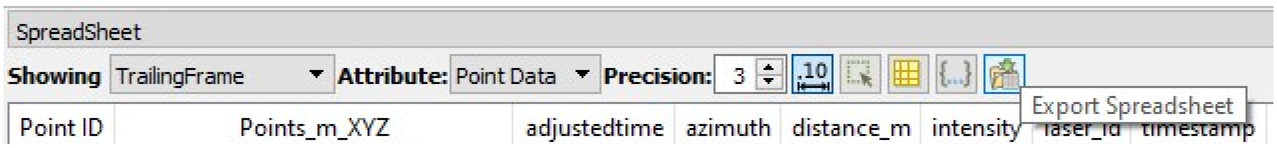
E.7 Exporting Data to PCAP or CSV Files

At any point you can save a subset of data frames by performing **File** > **Save As** > **Save PCAP** then enter the start and end frame numbers. Press OK to open a file-save dialog. Navigate to where you want to save the file. Click Save to save the frame to the named pcap file.

You can do a similar process to save data frames to CSV by performing **File** > **Save As** > **Save CSV** then enter the start and end frame numbers. Press OK to open a file-save dialog. Navigate to where you want to save the file. Click Save to save the frame to the named zip file containing csv files, one per frame.

You can export selected points to a CSV file in Spreadsheet view. Select some points then click the Export Spreadsheet button. Columns made visible by the 'toggle column visibility' control are saved while hidden ones are not. (A few vtk columns may be included to help with integrity checking. You are free to ignore them, of course.)

Figure E-15 Export Spreadsheet (CSV)



E.8 Multi-Sensor Support - NEW!

VeloView 5.1 introduces multi-sensor support. It is documented in VeloView's built-in user guide. You can access the guide from the Help menu or by hitting the F1 key.

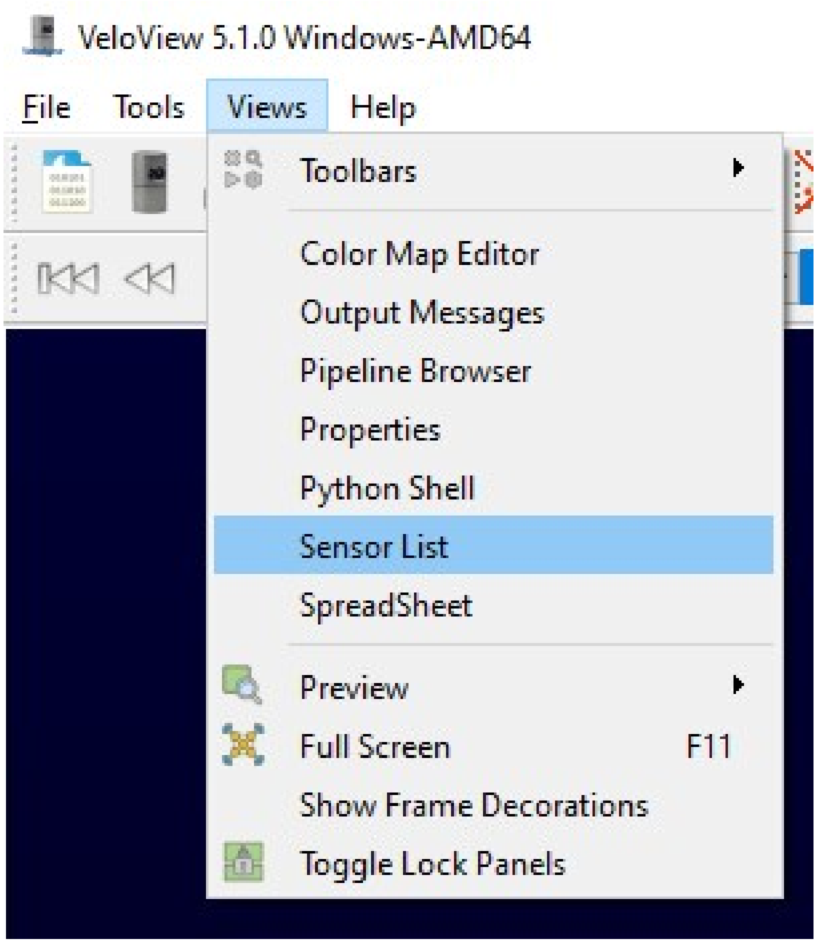
Multi-sensor support allows VeloView to render concurrent live streams from multiple Velodyne sensors, or view multiple Velodyne sensor captures (pcap files) at once.

Currently, it is not possible to select data points from multiple sensors at once in the render pane.

E.8.1 Sensor List

To allow you to manage multiple sensors, a new view called Sensor List was added. Access it from the Views menu.

Figure E-16 Sensor List View



The Sensor List panel has many tools for making adjustments to individual sensor data relative to the other sensors in its coordinate space. Checkmark **Enable Live Data Transform** to expose the *extrinsic calibration* tool set for the sensor.

Figure E-17 Sensor List

The **Configure** button summons the Sensor Configuration dialog specific to that sensor.

Next to the Configure button is the **Save Lidar state** button. This tool allows you to save many parameters associated with the LidarReader processing the data, the packet interpreter (including Crop Region), and, in particular, any Position or Rotation transforms. You select which ones are saved by checkmarking the boxes of interest. The settings are saved in a JSON file with a filename and location of your choosing.

Next to that button is the **Load Lidar state** button. This tool allows you to load a previously saved lidar state json file (or one formatted as such). Only the parameters saved are listed. None are selected by default. You select which parameters you are interested in loading at the time. Checkmark the ones you want, then click OK. The values loaded immediately appear in Sensor Configuration and Live Data Transform fields.

Changes made in a Sensor Configuration dialog (accessed via the Configure button, not the control on the top-level toolbar) are reflected in the Sensor List entry (and vice versa) and in the Save Lidar state dialog. If at any point you want to reset all parameters back to their defaults, click the **Reset advanced configuration** button in the Sensor Configuration dialog (accessed via the Configure button).

Checkmarking **Enable Live Data Transform** uncovers sliders and value spinners for easier adjust of position and rotation parameters for a given sensor in the Sensor List. Uncheck it to hide them again.

Adjustments made by slider are rendered live, as you move the slider bar. The value in the spinner is also adjusted live. To see a change made to a value spinner, hit the Tab key.

The Translation slider/value spinner pairs independently adjust, from left to right, the X, Y, and Z axes. The value range is indicated in the associated Min/Max value spinners, which you may modify.

The Rotation slider/value spinner pairs independently adjust, from left to right, Roll, Pitch, and Yaw. The value range is indicated in the associated Min/Max value spinners, which you may modify.

Adjustments to Roll rotate the data about the X axis.

Adjustments to Pitch rotate the data about the Y axis.

Adjustments to Yaw rotate the data about the Z axis.

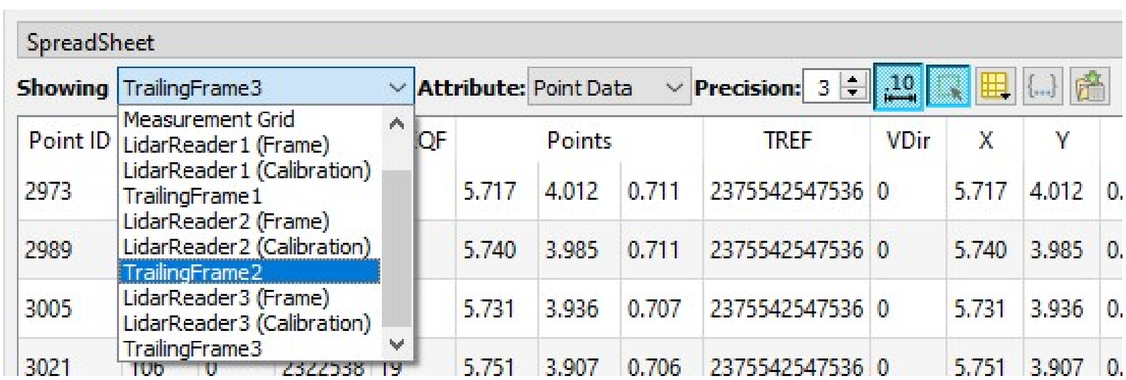
When rotating a sensor's data, it is useful to observe its affects relative to the X/Y/Z unit axes in the lower left of the rendering panel. It is equally useful to have the Grid enabled. (Use Ctrl-g to show/hide the grid as needed.)

E.8.2 Concurrent Playback

For best results during playback of multiple pcaps, checkmark both **Enable multi sensors** and **Use relative start time** in the Sensor Configuration dialog. The latter option, when enabled, effectively allows concurrent playback of multiple pcaps that had been recorded at different time bases by considering their initial timestamps to be zero.

E.8.3 Spreadsheet

Figure E-18 Select a Sensor



To examine data from a specific sensor in the list, select its *TrailingFrame#* in the **Showing** pull-down list in the Spreadsheet panel.

E.8.4 Pipeline Browser

Select Pipeline Browser in the Views menu to open the Pipeline Browser panel. The panel provides access to various data pipelines.

You can enable/disable rendering of individual sensor's data by clicking the EYE icon in the panel's left edge.

Use this panel in tandem with the Properties panel to open up additional settings.

Use this panel in tandem with the Spreadsheet panel to enable/disable rendering individual sensors so that you may select points from a given sensor.

E.8.5 Properties

Select Properties in the Views menu to open the Properties panel. The panel allows you to modify properties of the item selected in the Pipeline Browser.

Pro Tip: When a TrailingFrame is selected, the *Number of Trailing Frames* property allows you to enter a larger number than is allowed in the **TF** field on the toolbar.

E.8.6 VeloView User Guide

Additional documentation can be found in VeloView's built-in user guide. Access the guide from the VeloView's Help menu or by hitting the **F1 key**.

Appendix F • Laser Pulse

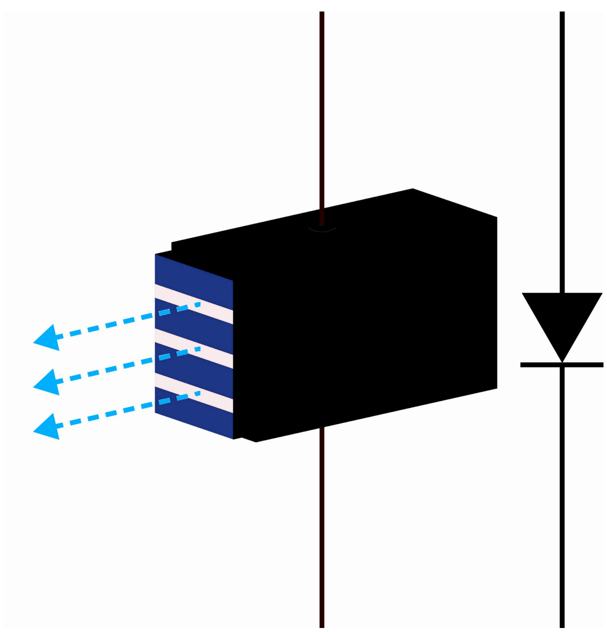
This section provides details on your sensor's laser diodes, the laser pulse and scan patterns, and certain beam characteristics of the laser pulses.

F.1 The Semiconductor Laser Diode

The source of each laser pulse in the sensor is a semiconductor laser diode. The laser diode is a series of stacked p-n junctions similar, at least in concept, to that shown in *Figure F-1 below*. When a current is applied across the junction, photons are produced and routed out one end to form a tightly focused laser beam.

Note: Write-ups on the physics of laser diodes can be found elsewhere on the Internet. Start by searching for *p-n junction*, *laser diode*, and *principles of semiconductor laser diodes*.

Figure F-1 Laser Diode Concept



F.2 Laser Patterns

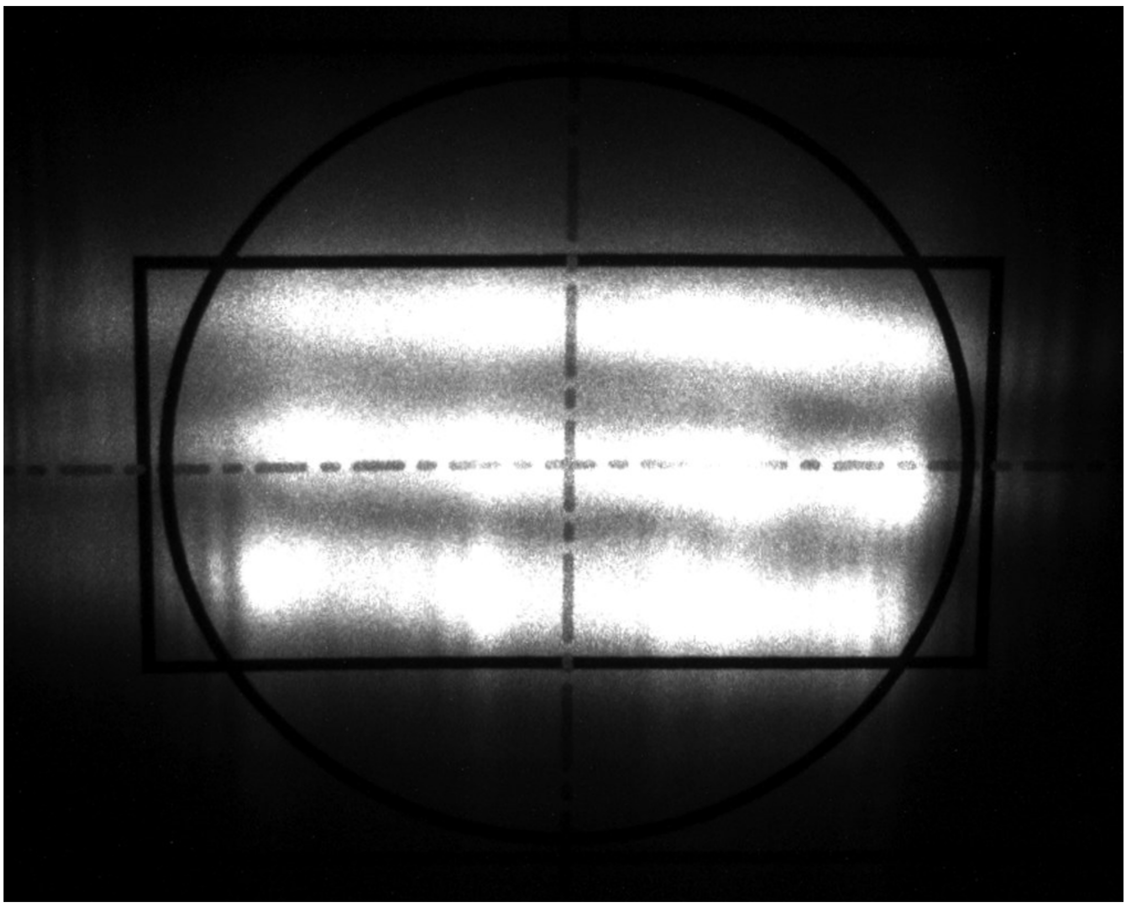
Laser firings produced by the sensor can be viewed with an infrared camera or smartphone camera sans IR filter. Photos in this section were taken with an infrared camera.

F.2.1 Laser Spot Pattern

While the terms laser “spot” and “dot” are often used when describing a laser pulse hitting a target, in reality the sensor’s laser “spot” is a small rectangular area comprised of three smaller bars or bands of light as shown in *Figure F-2 below*. The long axis of the rectangle coincides with the path of the laser scan.

The size of the laser spot upon exiting the sensor is roughly that of a thumbprint - but it doesn’t remain that size as it speeds away. Read more about that in *Beam Divergence on page 153*.

Figure F-2 Laser Spot Shape



F.2.2 Laser Scan Pattern

Inside each Alpha Prime sensor is essentially a two-dimensional array of lasers. The pattern scanned is the result of the sensor's spinning assembly moving the laser firing pattern depicted in *Figure F-3 on the facing page* for HDL operation, or *Figure F-4 on page 152* for APF, across (azimuthally) the sensor's environment in a clockwise motion as seen from above. A single firing's Azimuth (e.g. Azimuth1 for HDL, or AZM for APF) corresponds to the vertical centerline between columns 4 and 5 (from the left) in the figures.

Figure F-3 below reflects the laser channel firing arrangement found in **Alpha Prime.xml**. See *Precision Azimuth Calculation (HDL) on page 71* for details.

Figure F-3 VLS-128 Laser Pattern - HDL

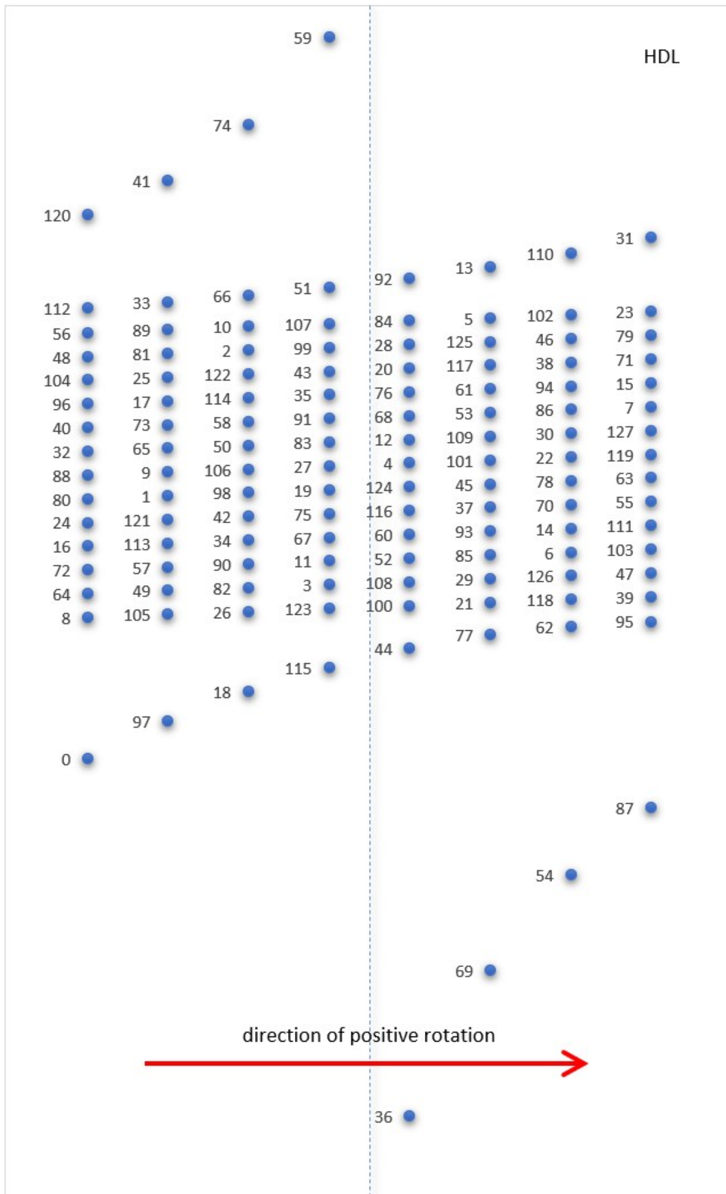
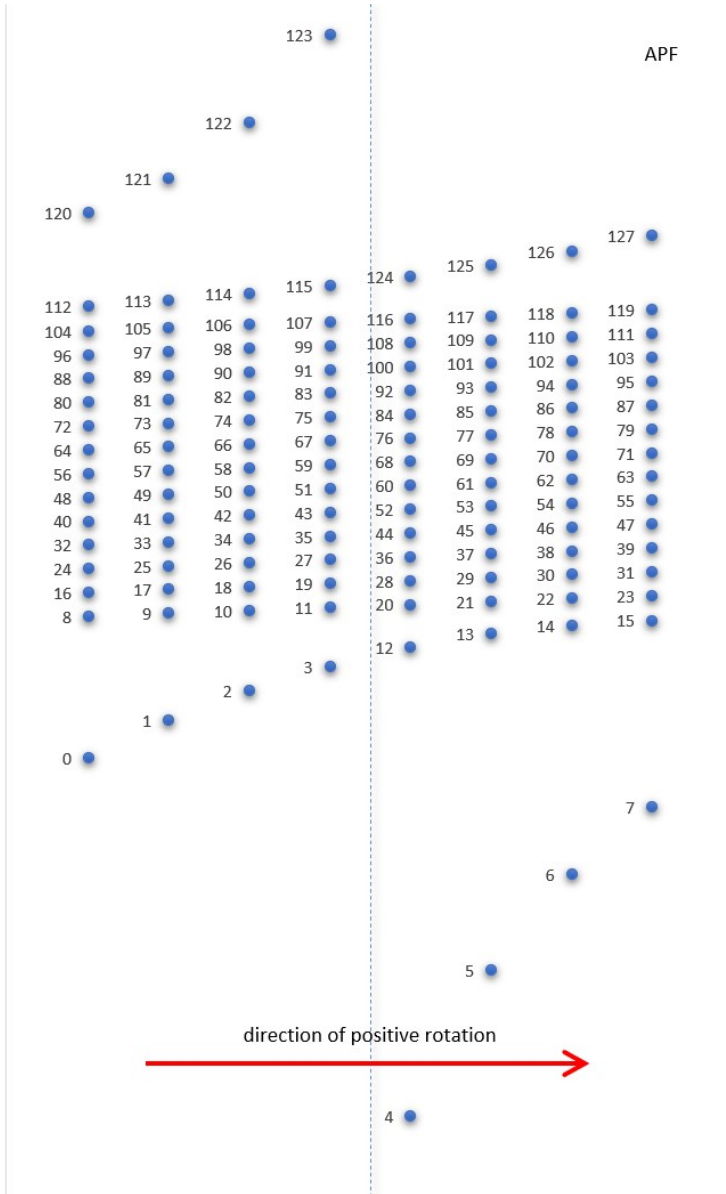


Figure F-4 on the next page reflects the laser channel firing arrangement found in **Alpha Prime APF.xml**. See *Precision Azimuth Calculation (HDL) on page 71* for details regarding the rearrangement of laser channels.

Figure F-4 VLS-128 Laser Pattern - APF



These laser patterns are depicted from the sensor's point of view, where the direction of positive rotation is to the right.

The important thing to grasp here is that, for any firing group of 8 lasers fired simultaneously, the lasers in the group are aimed as far apart from each other as possible. A wider distribution minimizes crosstalk between laser pulses. See *Figure F-3 on the previous page* and *Figure F-4 above* for the numbers.

Any gap between scan lines can be calculated with the following equation:

Equation F-1 Gap Between Scan Lines

$$\text{Gap} = \text{distance to target} \times \tan\left(\text{vertical angle between scan lines}\right)$$

F.2.3 Beam Divergence

As a laser pulse propagates outward from the sensor, the cross-section of the laser beam describing the path of the pulse gradually, steadily grows larger. The angular measure of this increase in beam path diameter is called Beam Divergence.

Alpha Prime beam divergence on the horizontal axis (i.e. along the direction of the laser scan) differs from beam divergence on the vertical axis (transverse to the scan) by roughly a factor of two.

Table F-1 Alpha Prime Beam Divergence

Horizontal Beam Divergence	Vertical Beam Divergence
2.09 mrad (0.1197482°)	1.05 mrad (0.06016057°)

Appendix G • Time Synchronization

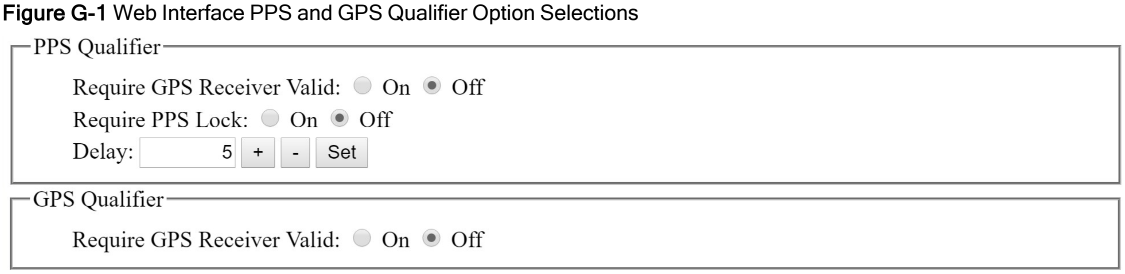
This section provides a detailed discussion of the **GPS Qualifier** and **PPS Qualifier** functions.

G.1 Introduction

These two options control how the sensor utilizes the GPS information supplied to the sensor. The first control option determines how the sensor utilizes the PPS signal (PPS Qualifier). The second control option determines how the sensor utilizes the timestamp provided in a National Marine Electronics Association (NMEA) sentence (GPS Qualifier).

The options, shown below in their default positions, can be found on the Configuration tab in the sensor's Web Interface.

Figure G-1 Web Interface PPS and GPS Qualifier Option Selections

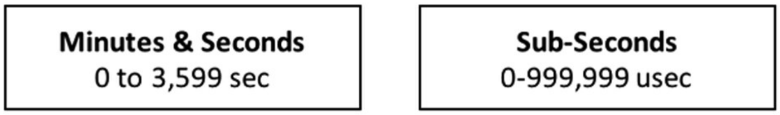


G.2 Background

Your sensor maintains a counter representing the number of microseconds since the Top Of Hour (TOH). The TOH count is incremented based on an internal oscillator. When the sensor is presented with a valid PPS signal the TOH count is adjusted on the rising edge of each PPS to align the TOH with UTC Time. The TOH is sent as a four-byte time stamp in both the data and position/telemetry packets.

The TOH is comprised of two separate counters. One counter maintains the number of seconds since the top of the hour, and the other counter maintains the sub-second count in microseconds (*Figure G-2 on the next page*).

Figure G-2 Top of Hour Counters



The combined value represents the number of microseconds since the top of the hour. It ranges from 0 to 3,599,999,999 µs -- there are 3.6×10^9 µs in one hour. The sub-second counter range alone spans 0 to 999,999 µs.

The sensor continuously monitors the PPS input assessing the timing characteristics of any pulses presented. This process is indicated both on the Web Interface (see *Configuration Screen on page 86*) and in the position packet. When the sensor detects a valid, stable PPS signal, the Web Interface indicates “PPS: Locked” and the PPS status field in the position packet at offset 0xF4 is set to 0x02.

As noted, the sensor counts microseconds using an internal oscillator. This internal oscillator always drives the sub-second counter; however, the sensor may be configured to use a valid PPS signal to adjust the value of the sub-second counter every second, thereby reflecting the "top of second" moment indicated by the rising edge of the PPS.

The seconds counter may be adjusted to the time value provided in the NMEA sentence. If no NMEA sentence is provided, the seconds counter is incremented every second.

Using the PPS signal to adjust the sub-second counter allows the sensor to remain synchronized to a PPS source even if that source is drifting slightly. This might occur when a GPS receiver indicates an invalid fix and begins using its own internal time to drive the PPS reference.

Control options (PPS Qualifier and GPS Qualifier, described below) have been added to allow the user to finely adjust the manner in which the sensor reacts when a valid PPS is confirmed and locked onto.

G.3 PPS Qualifier

The three settings described below (and in *Logic Tables on the facing page*) control how the sensor adjusts the TOH counter based on GPS and PPS signal status.

Depending on the GPS receiver's valid/invalid status, the three settings determine if the sensor's Top-Of-The-Hour (TOH) counter should enter free-running mode or synchronize the sub-second counter with the rising edge of the PPS signal.

G.3.1 Require GPS Receiver Valid

This setting determines the manner in which the sensor utilizes the current status of the GPS receiver provided in the NMEA sentence.

- If the **Require GPS Receiver Valid** is set to **On**, the sensor requires the GPS receiver to indicate a valid satellite condition before using the rising edge of a PPS signal to adjust the internal sub-second counter.

Note: The sensor must also receive a valid NMEA message indicating the GPS receiver is providing a valid time stamp.

- If **Require GPS Receiver Valid** is set to **Off**, the sensor synchronizes its sub-second counter to the rising edge of the PPS signal regardless of GPS receiver satellite status.

G.3.2 Require PPS Lock

This setting determines the manner in which the sensor validates the PPS signal prior to adjusting the internal sub-second counter to the rising edge of that PPS signal.

- If **Require PPS Lock** is set to **On** the sensor utilizes the value in the Delay field to determine the validity of a PPS signal prior to synchronizing its internal sub-second counter to the rising edge of a PPS signal.
- If **Require PPS Lock** is set to **Off** the sensor ignores the value in the Delay field and the sensor uses a rolling window of 2 cycles before the PPS signal may be considered valid and then used as a time reference by the sensor. Turning this option off effectively sets the Delay value to 2.

G.3.3 Delay

This parameter allows the user to extend the time the sensor requires to validate the PPS. The units are in integer seconds. Acceptable values range from 0 to 65535. The default is 5 seconds.

The sensor constantly qualifies the PPS signal over a rolling N-second window defined by the Delay parameter. At the first instance the PPS signal is deemed unstable, the sensor enters free-running mode where the sub-second counter is driven by the internal oscillator. In free-running mode the sub-second counter is no longer adjusted with the rising edge of the PPS signal. Additionally, the minutes and seconds component of the TOH begins incrementing based on the rollover of the free-running sub-second counter.

G.4 GPS Qualifier

This setting determines if the minutes and seconds component of the TOH counter are adjusted to the timestamp provided by the GPS receiver or driven by the rollover of the sub-second counter.

If the **Require GPS Receiver Valid** is **On**, the minutes and seconds field in the NMEA sentence timestamp is used to adjust the minutes and seconds component of the TOH counter only when GPS receiver indicates a valid status.

If the **Require GPS Receiver Valid** is **Off**, then the sensor ignores the GPS receiver status and always adjusts the minutes and seconds component of the TOH counter with the timestamp provided in the NMEA message.

G.5 Application

Most GPS receivers have low-drift internal clocks and may be configured to use this clock to maintain the PPS signal in the event the GPS fix becomes invalid. Alternatively, the GPS might be configured to discontinue the PPS in the event the fix becomes invalid.

In the default settings (and previous versions of firmware), the sensor uses PPS to synchronize to the time contained in the GPRMC message regardless of the state of valid flag in the GPRMC message. In this configuration, all the elements in a larger system (Lidar, IMU, RGB cameras, etc.) will be clocked off the same time source (the internal clock in the GPS receiver) in the event the GPS fix becomes invalid. This allows for proper reconciliation of the data during post- or real-time processing.

G.6 Logic Tables

Figure G-3 Sub-Second Counter Behavior

Setting: Require PPS Lock	Setting: Require GPS Receiver Valid	Sub-Second Counter Behavior
Off	Off	2-second qualifying delay in effect; Always zeroed by rising edge of PPS when PPS is qualified
Off	On	2-second qualifying delay in effect; Always zeroed by rising edge of PPS only when satellites are locked and when PPS is qualified
On	Off	User defined qualifying delay in effect; Always zeroed by rising edge of PPS when PPS is qualified
On	On	User defined qualifying delay in effect; Always zeroed by rising edge of PPS only when satellites are locked and when PPS is qualified

Figure G-4 Minutes and Seconds Counter Behavior

Setting: Require GPS Receiver Valid	Minutes and Seconds Counter Behavior
Off	Always updated with provided NMEA values when PPS is qualified
On	Only updated with provided NMEA values when satellites are locked and when PPS is qualified

---

Note: If no NMEA sentence is provided, the Seconds counter is driven by the rollover of the sub-seconds counter.

Appendix H • Phase Lock

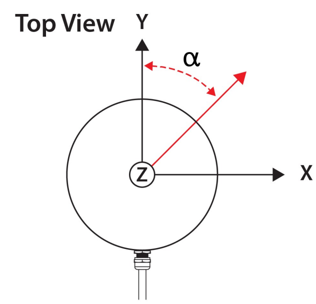
When using multiple sensors close to one another (e.g. mounted on top of a vehicle), occasional interference patterns may appear in the sensor data. Velodyne provides firing controls to minimize this interference by controlling where data is gathered. The sensors can then be configured to ignore the data containing the interference.

H.1 Phase Lock

With the introduction of firmware 5.2.3, when the sensor's Time-of-Day setting is GPS, the Phase Lock feature requires that a PPS signal be present and locked. The sensor uses the rising edge of the PPS as the zero-degree reference moment for all its firing references. The sensor then adjusts its timing such that it initiates a firing sequence at the phase lock offset specified by the user. However, if Time-of-Day is set to PTP, the sensor uses the top of the second as a synthetic PPS.

For example, assume the user enters an angle of 35° (α) as the phase offset. The red arrow in *Figure H-1 below* indicates the laser firing direction precisely as the sensor receives the rising edge of the PPS signal, or when the top of the second occurs if PTP time synchronization is selected.

Figure H-1 Direction of Laser Firing



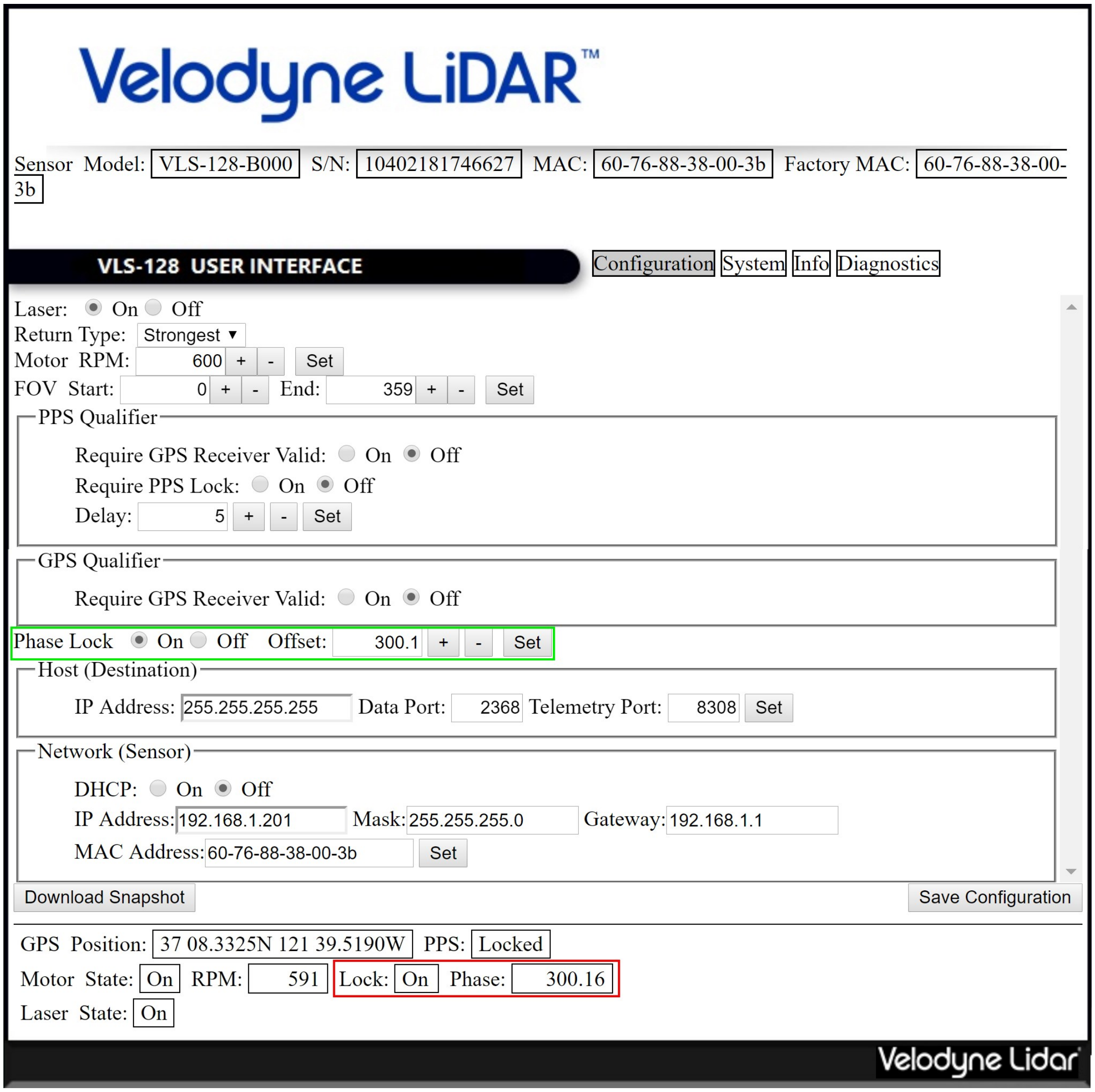
H.1.1 Setting the Phase Lock

To enable Phase Lock, enter the desired phase offset in the Phase Lock **Offset** field shown in *Figure H-2 on the next page*. Enter the offset in decimal degrees. For example, if the desired offset is 270°, enter **270** in the Offset field. Click the Phase Lock **On** radio button (as needed) and then the Set button to the right.

Note: To retain these settings over the next power cycle or reset, click the **Save Configuration** button.

The current phase lock status (On/Off) and phase lock offset can be viewed on the Web Interface (red box). The current phase lock offset is presented in degrees. The accuracy of the offset is ±5° (subject to change).

Figure H-2 Configuration Screen - Phase Lock

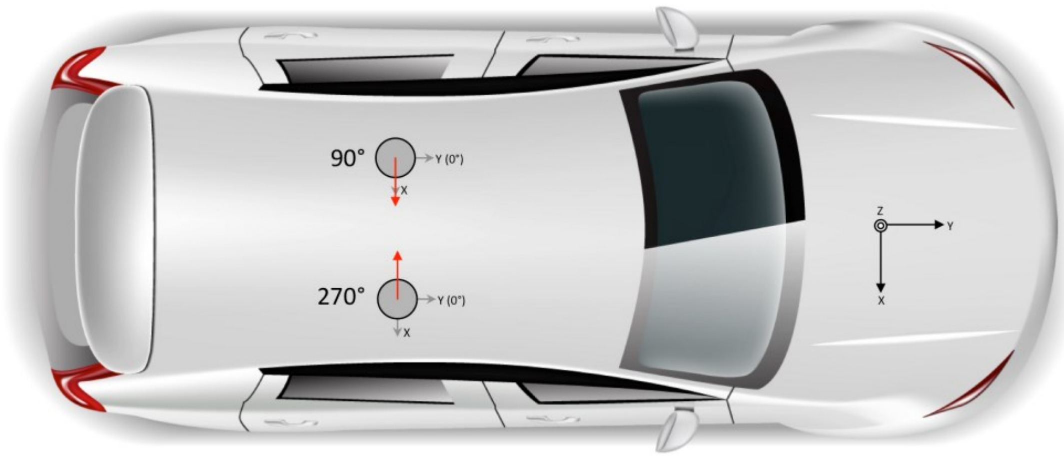


H.1.2 Application Scenarios

When setting the phase lock offset for two or more sensors, Velodyne recommends the sensors be configured to fire at each other. This is the optimal configuration for minimizing interference because the location of the interference is under user control.

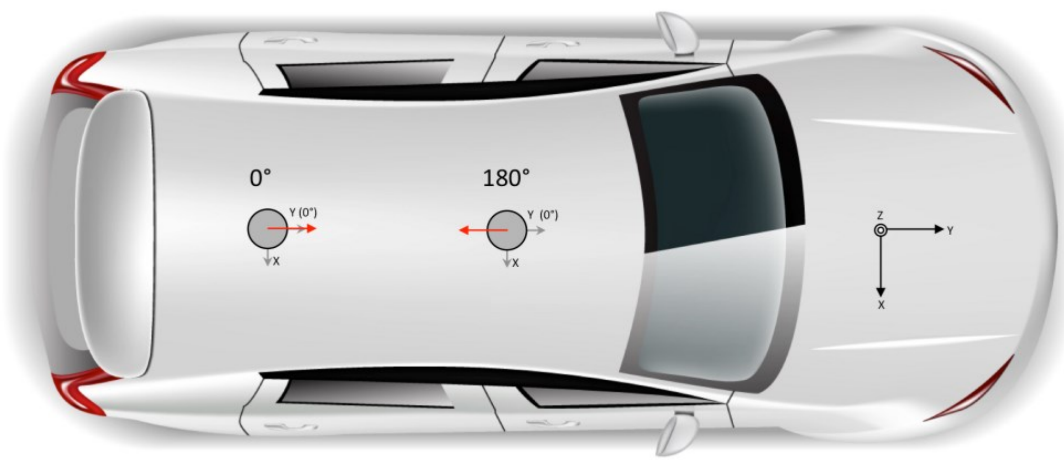
Figure H-3 on the facing page shows two sensors mounted on a vehicle. The sensor mounted on the car's left side has its phase lock offset set to 90°, and the phase lock offset of the sensor mounted on the right side of the vehicle is set to 270°, as shown by the red arrows.

Figure H-3 Right and Left Sensor Phase Offset



When sensors are placed on the roof in the fore and aft positions, the phase offsets are set to 180° and 0° as shown in *Figure H-4 below*.

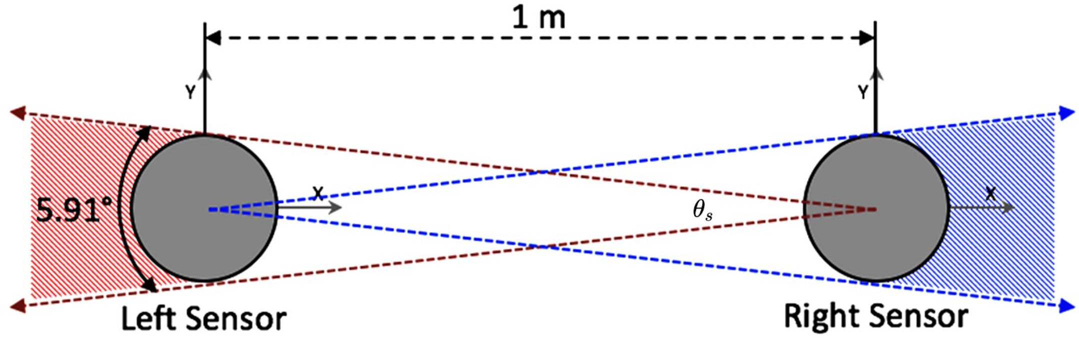
Figure H-4 Fore and Aft Sensor Phase Offset



In both scenarios the two sensors create data shadows behind each other.

To avoid any spurious data due to blockage or reflections from the opposing sensor, the user should ignore any data in the shadowed azimuth ranges as shown in *Figure H-5 on the next page*. To do that, you need to know the diameter of the sensors (see *Sensor Specifications on page 117*) and distance between the sensor centers.

Figure H-5 Sensor Data Shadows



The angle subtended by the shadow of a sensor in range but at some distance from another sensor is given by the following formula:

Equation H-1 Arc of Shadow

$$\theta_s = 2 \times \tan^{-1}\left(\frac{0.5 \times D_{\text{Sensor}}}{d_{\text{Sensors}}}\right)$$

$$\theta_s = \text{Subtended Angle}$$

$$D_{\text{Sensor}} = \text{Diameter of the far sensor}$$

$$d_{\text{Sensors}} = \text{Distance between sensor centers}$$

Data reported at the azimuths included within the subtended angle should be ignored.

H.2 Field of View

Alternatively, each sensor's Field of View control may be used to remove the subtended azimuths from the data stream.

Use *Figure H-5 above* and *Equation H-1 above* to determine for each sensor the azimuthal angles at which the shadows begin and end.

Then, using each sensor's Web Interface or curl commands (or equivalent programmatic commands), configure the sensor's horizontal FOV start and end angles. See *Configuration Screen on page 86* and *Set Field of View on page 100* for more.

Appendix I • Sensor Care

This section lists various approved cleaning methods, but it is important to use the correct method. Start with *Determine Method of Cleaning the Optical Window below*.

I.1 Cleaning the Sensor

⚠CAUTION

READ THROUGH THIS ENTIRE SECTION BEFORE CLEANING YOUR Alpha Prime SENSOR
Improper handling can permanently damage it.

I.1.1 Required Materials

1. Clean microfiber cloths
2. Mild, liquid dish-washing soap
3. Spray bottle with warm, clean water
4. Spray bottle with warm, mildly soapy water
5. 70% isopropyl alcohol (Method 2 ONLY)
6. NACL Precision Optics Cleaner (Method 3 ONLY, Optional)

I.1.2 Determine Method of Cleaning the Optical Window

The *Ring Lens* is the optical window through which the laser pulses are fired and reflections return. In addition to its base optical material, it has a number of coatings. For optimal performance, it is important to keep the lens clean while avoiding damage to it.

If the first two characters of the sensor's serial number are letters then the first five characters of the serial number (e.g. AE031) represent the year (AE = 2015) and day of manufacture (032 = February 1st) of the device.

Use this list to determine which cleaning method to employ:

- If the first five characters of the serial number are between AE001 and AE229, clean the sensor using Method 1 below.
- If the first five characters of the serial number are AE230 or greater, or if the sensor was serviced by Velodyne Lidar after August 17th, 2015, use either Method 2 or 3 below.
- If the serial number is composed of all digits, no letters, use either Method 2 or 3 below.

Note: The easiest way to get a sensor's serial number is from the label on the device's underside. Sometimes, however, it is mounted somewhere inaccessible and that is not possible. The next easiest way is to view the sensor's *Web Interface on page 85*. Look for S/N at the top. Other ways include getting a snapshot from the command line (see *Get Snapshot on page 99*) or requesting the snapshot programmatically (see *curl Example using Python on page 103*). In both cases, the serial number can be found in the `snapshot['info']['serial']` field.

I.1.3 Cleaning Tips

Avoid using hard water when cleaning the sensor. Salt deposits may degrade performance. Removing salt deposits may abrade the surface.

Avoid using cleaning products containing ammonia (e.g. Windex) or bleach (e.g. Clorox). They contain chemicals potentially harmful to the sensor's exterior materials and coatings, and may etch or produce a cloudy looking surface.

Avoid using paper towels or anything abrasive such as cotton automotive rags.

Do not attempt to scrape, buff, or polish scratches in optical components. Doing so may add more scratches or remove coatings.

Never immerse the sensor in any liquid. Avoid cleaning with a high-pressure jet. Using either technique may result in liquid getting past one or more of the O-rings, requiring factory repair.

I.1.4 Method 1

This sensor's ring lens is made of acrylic. If the sensor is caked with mud and bugs, use a spray bottle with clean, warm water to loosen any debris from it. Do not wipe dirt directly off the sensor without loosening it sufficiently. Doing so may abrade the surface. Try to spray it off with warm water first. Next, use warm, mildly soapy water and a clean microfiber cloth to gently wipe the sensor, again taking care not to abrade the surface. When cleaning the ring lens, wipe gently along the curve of the sensor, not top-to-bottom. To finish, spray the sensor with clean water to rinse off any remaining soap, then dry with another clean microfiber cloth.

I.1.5 Method 2

This sensor’s ring lens is made of polycarbonate and may be cleaned with **isopropyl alcohol**.

But first, if the sensor is caked with mud or bugs, use a spray bottle with clean, warm water to loosen any debris from it. Do not wipe dirt directly off the sensor. Doing so may abrade the surface. Try to spray it off with warm water first.

Then, if necessary, use isopropyl alcohol and a clean microfiber cloth to clean any remaining dirt from the sensor. Next, use warm, mildly-soapy water and gently wipe the sensor with a clean microfiber cloth. Wipe the ring lens gently along the curve of the sensor, not top-to-bottom. To finish, spray the sensor with clean water to rinse off any remaining soap, then dry with another clean microfiber cloth.

I.1.6 Method 3

North American Coating Laboratories (NACL) has formulated a cleaning solution for Velodyne Lidar optical devices. It can be ordered directly from them.

- NACL part number: 98-0020
- NACL description: NACL Precision Optics Cleaner 6 oz

North American Coating Laboratories
9450 Pineneedle Drive, Mentor, OH 44060
Toll-Free: +1 (866) 216-6225
Tel: +1 (440) 357-7000
Fax: +1 (440) 357-7001
Email: info@nacl.com
URL: http://www.nacl.com/

1. Spray NACL Precision Optics Cleaner solution onto a clean, dry microfiber cloth.
2. Gently wipe the Alpha Prime's ring lens along the curve of the sensor, not top-to-bottom.

I.2 Cleaning Non-Optical Sensor Surfaces

Clean non-optical sensor surfaces with soapy water and a clean microfiber cloth. Follow the Cleaning Tips above. Avoid wetting electronic components, such as connectors and the PCB in the Interface Box, if present.

Appendix J • Network Configuration

Your sensor generates lots of data which it transmits via Ethernet. This section covers various aspects of this interface and how to connect and configure it. It also touches on the situation where you have more than one sensor on the same network.

J.1 Ethernet and Network Setup

The RJ45 Ethernet connector on the Interface Box connects to any standard 1 Gbps Ethernet NIC or switch with MDI or AUTO MDIX capability.

Note: The Alpha Prime is not capable of accepting power over Ethernet (POE). Its Ethernet hardware is incapable of handling current in excess of 2 A. You may still employ POE, but only if a POE splitter is used to provide separate Ethernet and power to the sensor. See *Power Considerations on page 35* for additional details.

J.1.1 Defaults

Each sensor's IP address is set at the factory to 192.168.1.201.

By default, the sensor sends UDP data packets to broadcast address 255.255.255.255.

Note: Each sensor has a unique MAC Address and Serial Number set at the factory by Velodyne Lidar that cannot be changed.

J.1.2 Establishing Communication via Ethernet

The instructions below are for Windows computers. For linux or Macintosh computers, perform equivalent steps (not specified here).

1. Connect the computer to the Interface Box with an Ethernet cable.

   Velodyne recommends disabling WiFi on your computer to avoid possible network conflicts.

2. Provide power to the sensor.
3. Open your computer's Network/Ethernet settings page.

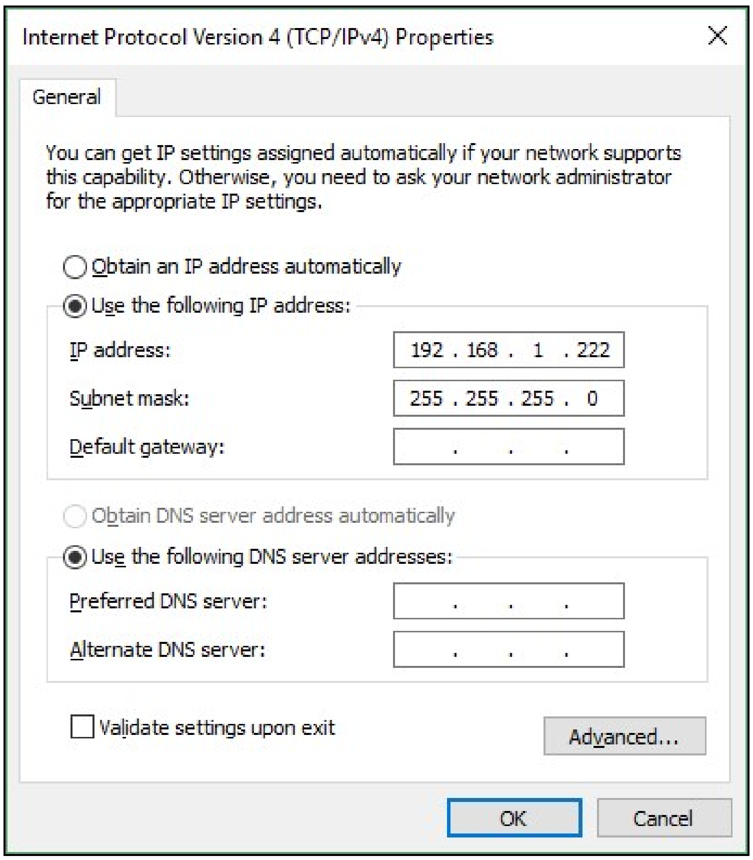
4. Select **Internet Protocol Version 4 (TCP/IPv4)**.
5. Select the **Use the following IP address:** function.

   The sensor requires a static IP address.
6. Enter the following IP address: 192.168.1.XXX. See *Figure J-1 below*.

   "XXX" can be any number from 2 to 254 except 201.
7. Enter the subnet mask: 255.255.255.0

   When using a Windows OS based computer, you can press the TAB key and the subnet mask automatically populates with the 255.255.255.0 value.
8. Click **OK**.
9. We recommend disabling any firewall software the computer may have running.
10. Point your browser to 192.168.1.201 to access the sensor's Web Interface to confirm communication.

Figure J-1 Sensor Network Settings



J.2 Network Considerations

Your application network topology may be simple, with a single sensor transmitting data on a basic network. Or, it could be complicated, with multiple sensors. This section presents certain topics to consider.

J.2.1 Throughput Requirements

When actively sensing its environment, your sensor produces a lot of data which it transmits via Ethernet. The volume of data depends partly on which Return Type (or mode) it's in. Details on return modes can be found in *Laser Return Modes on page 36*.

You should assess network loading to see if your network's topology can accommodate the sensor data rate you select.

J.2.2 Single Sensor Transmitting to a Broadcast Address

Below is a single sensor on a network transmitting to a broadcast address works without any contention.

Figure J-2 Single Sensor Broadcasting on a Simple Network

This is the simplest network, with sensor data flow free from competition or interference. Here it is acceptable for the sensor to broadcast its data on the network.

J.2.3 Multiple Sensors in the Same Network

Note: Each sensor must have its own, unique IP address in a given network.

When using multiple Velodyne Lidar sensors in a network it is imperative that you set each sensor's destination IP address to a specific, non-broadcast IP address. However, two or more sensors may share the same destination address.

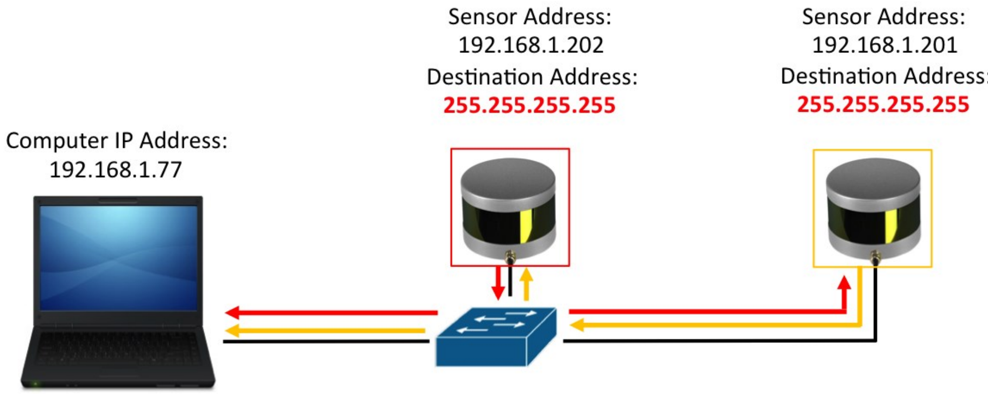
The scenarios below illustrate the wrong way and then a better way to configure them. (An alternative would be to direct-connect sensors to separate NICs.)

J.2.3.1 Multiple Sensors Transmitting to a Broadcast Address

If multiple sensors on a network transmit data to a broadcast address, each sensor will see the broadcast data of the other sensors. The additional overhead steals cycles from the sensor (and possibly other networked devices) and may lead to a degradation in performance. (See *Phase Lock on page 159* and *Phase Locking Multiple Sensors on page 41* for additional information on using multiple sensors within sensing range of each other.)

An improper setup is shown in *Figure J-3 on the facing page*. Note that each sensor's destination address is set to 255.255.255.255.

Figure J-3 Multiple Sensors - Improper Network Setup

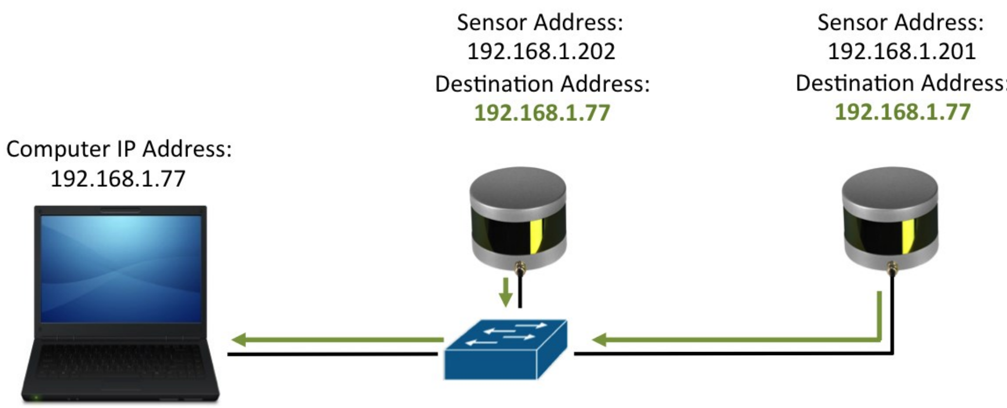


J.2.3.2 Multiple Sensors Transmitting to a Specific Address

The solution is to configure each sensor on such a network to transmit their data to a non-broadcast address.

If every sensor in the same network transmits packets to a specific, non-broadcast, destination address (doesn't have to be the same one), as illustrated in *Figure J-4 below*, the other sensors will not suffer unnecessary network overhead.

Figure J-4 Multiple Sensors - Proper Network Setup



Appendix K • Time Synchronization via PTP

K.1 GPS/PPS Support

Time synchronization by GPS/PPS is not supported.

K.2 gPTP Support

The Alpha Prime uses IEEE 802.1AS-2011 (gPTP) for time synchronization when a PTP master or grandmaster clock (hereafter referred to as GM) is available. If not, its built-in clock free-runs.

The sensor's internal clock updates once a PTP-compliant clock becomes available and the PTP protocol conversation takes place.

The sensor has hardware (Ethernet PHY) and firmware support for gPTP. This allows you to integrate it into your time-aware application framework.

PTP is supported in firmware 5.2.3 and later. (In 5.2.1.1 firmware, the UTC offset published by the GM is ignored. The flag is not honored.)

The default Neighbor (Peer Link) Propagation Delay Threshold is 10000 ns (10 µs). This is the delay threshold for which a peer will be considered unable to run generalized Precision Time Protocol (gPTP) if the threshold is exceeded. It is configurable on the sensor's Web Interface under Time Of Day > PTP > PTP Configuration.

K.3 Clocks

The rate quality of your PTP clock matters. It can affect the accuracy of the sensor's timestamp.

If a computer is being used as a PTP clock, the quality of the clock will depend on the quality of the oscillator and the feature set of the NIC used. Your best results will be seen when the NIC supports Hardware Timestamping. On linux, invoke `ethtool -T` *`eth_interface_name`* to see the Ethernet interface's PTP feature set.

It is possible for the Alpha Prime to act as a PTP Clock for other devices on a TSN. If it does become the best available clock, according to the PTP protocol, it does so on domain 0. Normally, however, you should have other, better PTP clocks on your TSN, which will provide a proper, best available clock. The sensor, then, operates in client mode, and not as a clock.

K.4 Sensor Startup

The sensor does not store a previous clock value, nor does it keep its clock updated while powered off (no battery onboard), so TOH (top of hour) starts at 0.

With gPTP, once the sensor receives the PTP time from the GM, it updates its clock almost immediately. No slew time should be observed. At best, it takes one SYNC message to update (from the moment the sensor starts sending data packets).

K.5 External Confirmation

Confirmation that the sensor is accepting PTP time from a GM can be inferred by detecting significant jumps in the timestamp (TREF) in data and diagnostic packets before and after a PTP clock becomes available to the sensor.

The Alpha Prime displays TREF as "TOH" in the Status portion of its web interface. Confirmation can also come by watching it jump to catch up to current time once an out of date sensor obtains a new time from a PTP clock.

K.6 Sensitivity to Other PTP Network Components

Any non-802.1AS-compliant device introduced between the sensor and the GM may interfere with or block the PTP conversation.

Care should be taken when using dongles to pass Ethernet (1000BASE-T1) from RJ45 on the sensor side (via a Media Converter), to a different type of (presumably automotive grade) connector. Ensure that it doesn't add unwelcome jitter or time accuracy may degrade.

Switches and other network devices the PTP conversation may traverse must be IEEE 802.1AS compliant.

Corporate Headquarters:

Velodyne Lidar, Inc.
5521 Hellyer Ave
San Jose, CA 95138
U.S.A.

www.velodynelidar.com

Phone +1 408-465-2800
Fax +1 408-779-9227

Email lidar@velodyne.com

Alpha Prime User Manual
63-9483 Rev. 4
Updated 2023-02-06