VLP-32C User Manual

63-9325 Rev. D

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™, and Velarray™ 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

Revision History

Sensor	Firmware	Release Date	Release Notes
VLP-32C	4.4.0.0	2019-01-18	• IMPROVED: Retro Distortion Mitigation fine tunes distance distortion seen on some retro and white surfaces. • IMPROVED: FOV Power Save feature to reduce laser power output when outside the field of view. • ADDED: New option to enable UDP Checksum for data packet stream. • Improved characterization of GPS and PPS signals (see Chapter 7).

Table of Contents

Chapter 1 • About This Manual

Chapter 2 • VLP-32C Overview

Chapter 3 • Safety Precautions

Chapter 4 • Unboxing & Verification

Chapter 5 • Installation & Integration

Chapter 8 • Sensor Operation

Chapter 9 • Sensor Data

Chapter 10 • Sensor Communication

Appendix C • Mechanical Diagrams

Appendix D • Wiring Diagrams

Appendix E • VeloView

Appendix F • Laser Pulse

Appendix G • Time Synchronization

Appendix H • Phase Lock

Appendix I • Sensor Care

Appendix J • Network Configuration

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 VLP-32 "Ultra Puck" family of Velodyne Lidar sensors.

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

CRC - Cyclic Redundancy Check. CRC values are used to 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.

TOH - Top of the hour.

Chapter 2 • VLP-32C Overview

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

2.1 Product Overview

The VLP-32C sensor uses 32 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 approximately 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

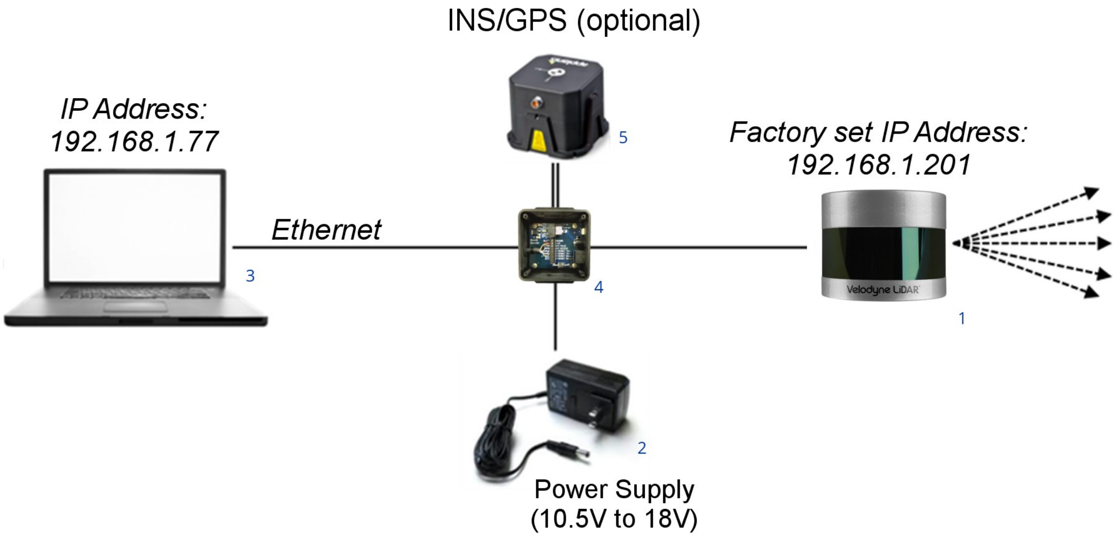


Table 2-1 3D Sensing System Components

Item	Description
1	VLP-32C Sensor

Item	Description
2	AC/DC Power Adapter or DC Power Supply for the sensor
3	Desktop/Laptop/Embedded Computer
4	Velodyne Interface Box
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 Support on page 98* if you are experiencing this problem.

2.2 Product Models

For ordering information, contact Sales at http://www.velodynelidar.com/contacts.php. Data sheets for available models can be downloaded from http://velodynelidar.com/downloads.html#datasheets.

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 73*.

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 VLP-32C 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 97* for details.

3.2.2 Mechanical Safety

	⚠CAUTION
	The VLP-32C sensor contains a rapidly spinning assembly. Do not operate the VLP-32C 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

LASER
1
IEC 60825-1: 2014 / 2007

Note: The VLP-32C 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.

Never look directly at the transmitting laser through a magnifying device.

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 VLP-32C sensor comes packaged in its own cardboard box, surrounded by foam padding.

Ensure all the components are present:

- VLP-32C sensor in a protective shroud, with a fixed 3.0 m data/power cable terminated inside its Interface Box
- AC/DC power adapter and 1.8 m AC power cord (once assembled, this is the power cord)
- 1.0 m Ethernet cable

4.1.1 Variants

Variants of the sensor exist, particularly with other connector types and/or cable lengths, and even without an Interface Box. 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.

A video of the VLP-16 installation process is on YouTube at the following location: https://www.y-outube.com/watch?v=Pa-q5elS_nE.

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 the facing page*.
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 27*.

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.

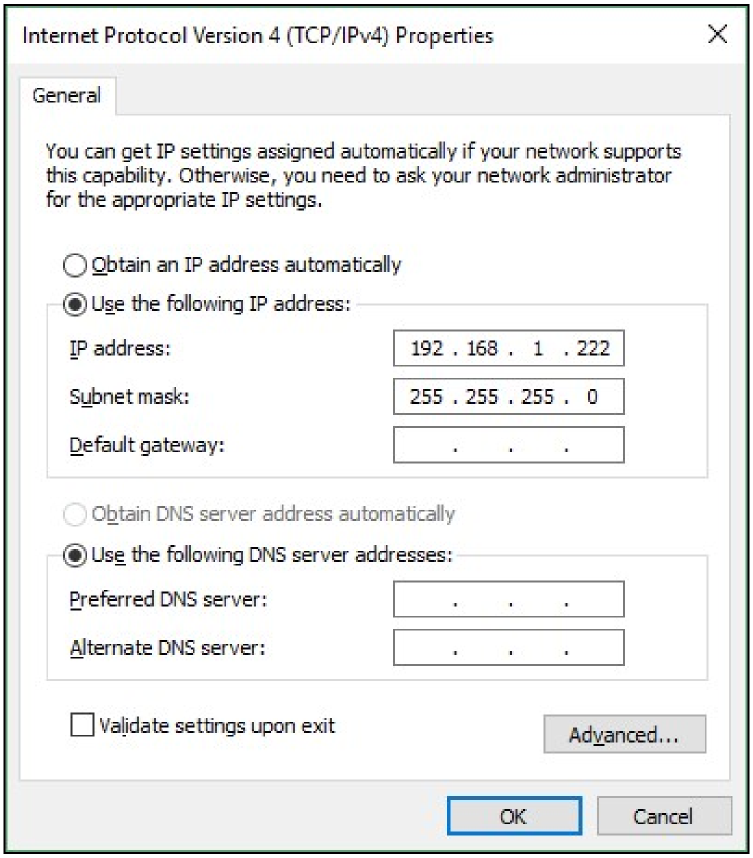
Note: If using the computer's main Ethernet port, disconnect it from whatever network it's on. If using a secondary Ethernet interface, the primary network cannot be a 192.168.1 network. If it is, use the primary Ethernet interface instead.

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 on the facing page***).**
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.

Figure 4-1 Sensor Network Settings



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 1 . 222

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

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

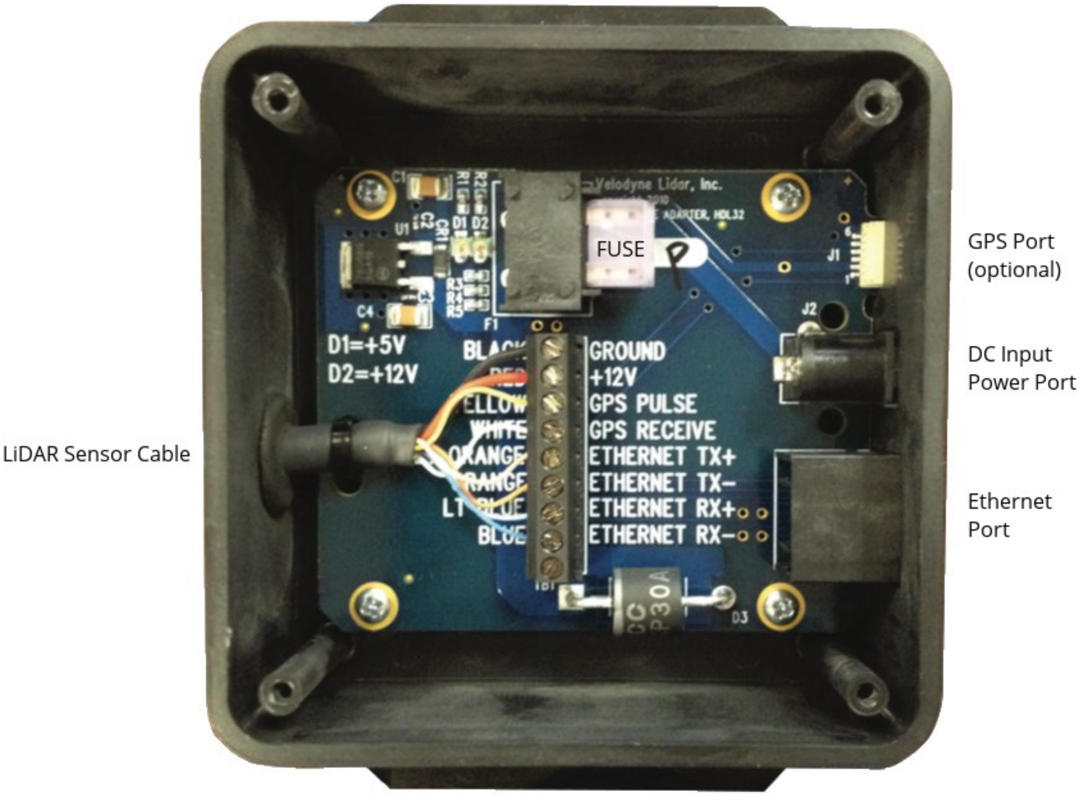
In some cases it may be necessary to disable the computer's firewall or configure it to allow UDP I/O on that Ethernet interface. How to do this is not covered here as the process varies widely.

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 on the next page* shows the Interface Box, its external ports, internal sensor terminal, and fuse.

Figure 4-2 Interface Box (power and data connections)



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

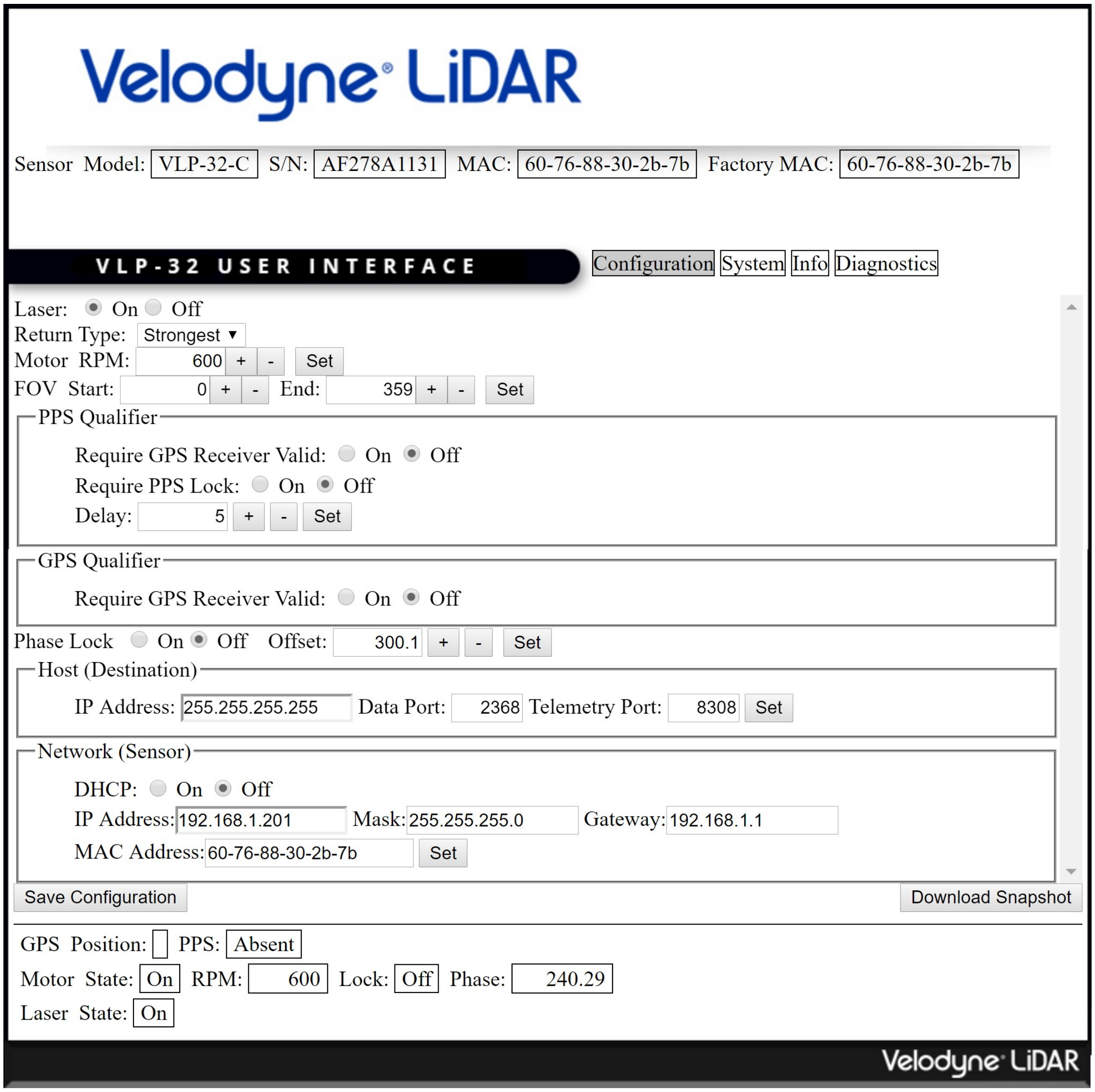
   When power is applied, two green LEDs in the Interface Box light up. The sensor begins scanning its environment and transmitting data approximately 30 seconds after power up.

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

4. The sensor's Web Interface should appear (*Figure 4-3 on the facing page*).

   The Web Interface provides access to many of the sensor's control settings. See *Web Interface on page 75* for details.

Figure 4-3 Sample Web Interface Main Configuration Screen



Sensor Model: VLP-32-C S/N: AF278A1131 MAC: 60-76-88-30-2b-7b Factory MAC: 60-76-88-30-2b-7b

Laser: ● On ○ Off
Return Type: Strongest ▾
Motor RPM: 600 + - Set
FOV Start: 0 + - End: 359 + - Set

PPS Qualifier
Require GPS Receiver Valid: ○ On ● Off
Require PPS Lock: ○ On ● Off
Delay: 5 + - Set

GPS Qualifier
Require GPS Receiver Valid: ○ On ● Off

Phase Lock ○ On ● Off Offset: 300.1 + - Set

Host (Destination)
IP Address: 255.255.255.255 Data Port: 2368 Telemetry Port: 8308 Set

Network (Sensor)
DHCP: ○ On ● Off
IP Address: 192.168.1.201 Mask: 255.255.255.0 Gateway: 192.168.1.1
MAC Address: 60-76-88-30-2b-7b Set

GPS Position: PPS: Absent
Motor State: On RPM: 600 Lock: Off Phase: 240.29
Laser State: On

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.

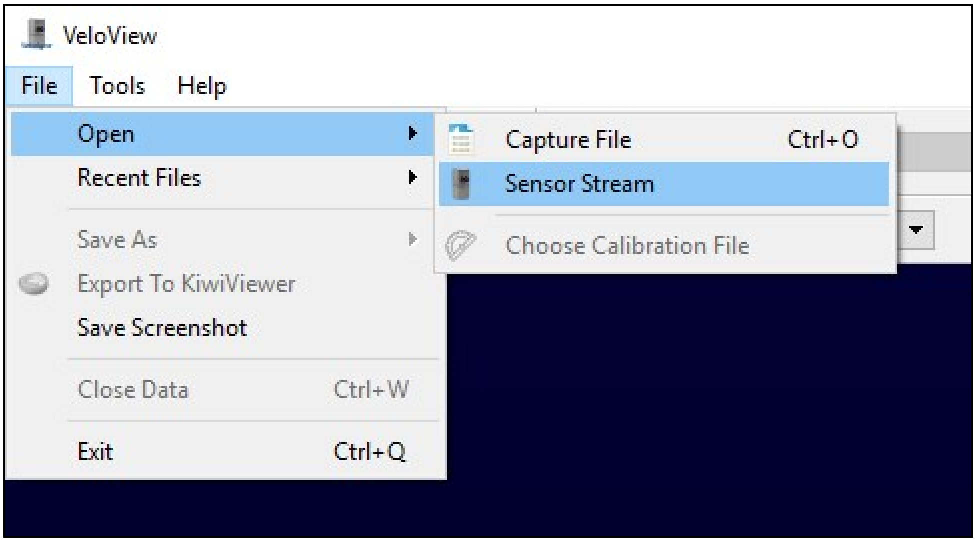
Note: VeloView is an open source visualization and 3D data recording application tailored for Velodyne Lidar sensors. Other visualization software (e.g. ROS, DSR and PCL) and middleware (e.g. Apollo, Autoware) can perform similar functions and may be used instead.

VeloView is documented in more detail in *VeloView on page 114*. If the application isn't already on the computer, perform the procedure detailed in *Install VeloView on page 115*. Older versions should be updated to at least the version installed by following the procedure.

4.2.3.1 VeloView Operation

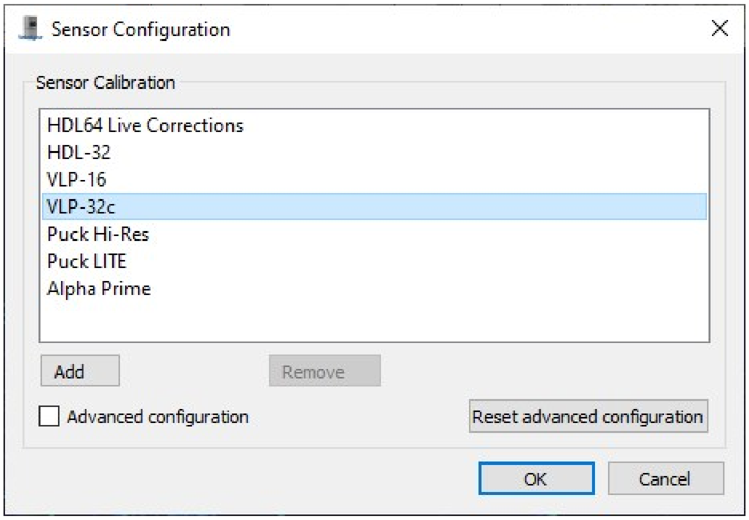
1. Power-up the sensor.
2. Start the VeloView application.
3. Click on **File**->**Open** and select **Sensor Stream** (*Figure 4-4 below*).

Figure 4-4 VeloView Open Sensor Stream



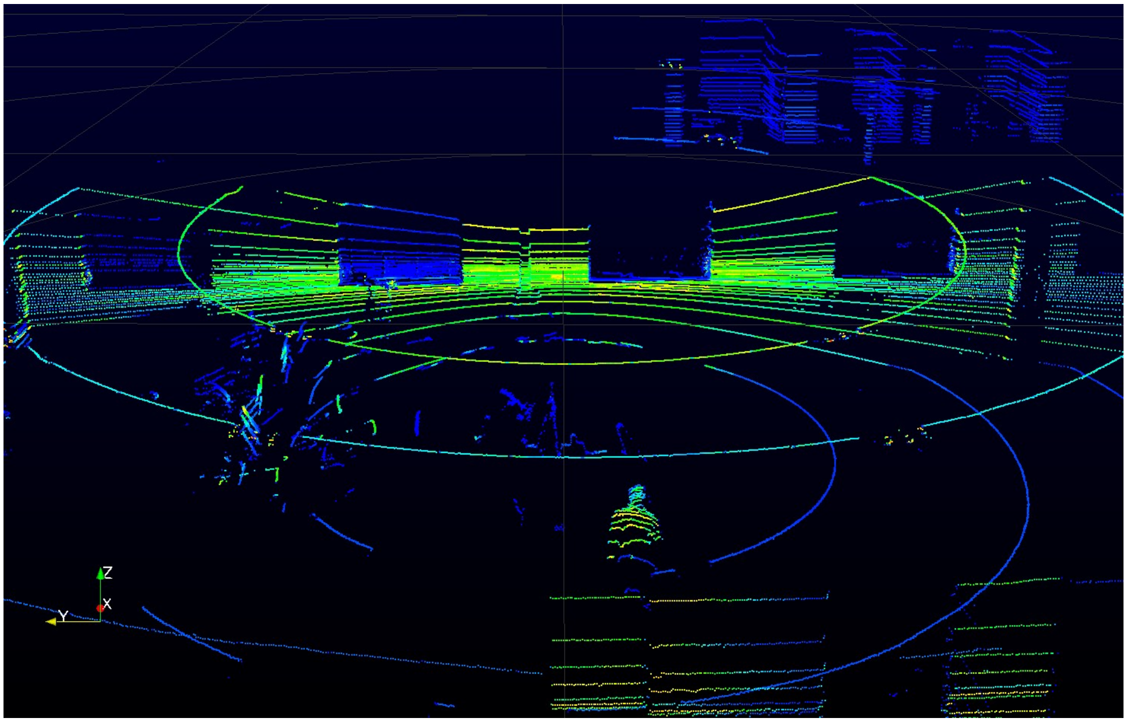
4. The Sensor Configuration dialog will appear (*Figure 4-5 below*). Select the correct sensor type then click **OK**.

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 VLP-32C sensor into your application environment.

5.1 Overview

Ensure the sensor is functional first before beginning sensor integration. See *Verification Procedure on page 23*.

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 140*
- Optionally, connecting a GPS receiver or INS (Inertial Navigation System) - see *GPS, Pulse Per Second (PPS) and NMEA Sentence on page 43*

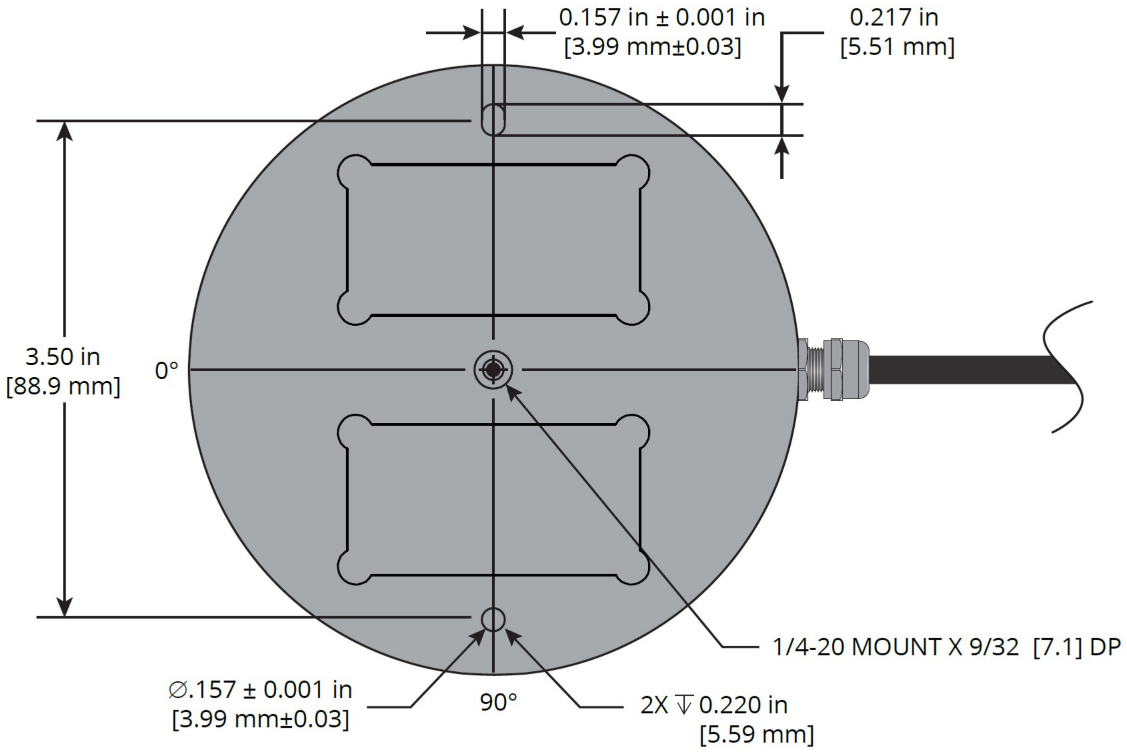
The typical sensor setup uses a standard computer or laptop connected to the sensor. However, it is recommended to use at least a 100 Mbps Ethernet adapter to accommodate the sensor data rate.

5.2 Mounting

The sensor base provides one ¼”-20-threaded, 9/32"-deep mounting hole, and two precision locating holes for locator pins (*Figure 5-1 below*). 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.

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.

Figure 5-1 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 VLP-32C 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 75* 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 141* before connecting one or more Velodyne Lidar sensors physically to your network. See *Ethernet and Network Setup on page 140* for instructions on how to configure the sensor's Ethernet connection.

Note: Do not twist or otherwise adjust the strain relief gland nut where the interface cable enters the sensor. Adjusting the position of the gland nut may damage the VLP-32C sensor internally.

5.4.1 Interface Box and Cable

The standard VLP-32C sensor comes with an integrated cable that is terminated inside the Interface Box. The cable is approximately 3.0 m in length and is permanently attached to the sensor.

Note: If you need to cut the integrated cable for any reason, please contact Velodyne Sales to get a waiver. Cutting this cable could have warranty and RMA cost implications. Also, there is a minimum length requirement.

The Interface Box provides convenient connections for power, Ethernet, 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 42*.

5.4.2 Operation Without an Interface Box

The VLP-32C's Interface Box is a convenience, but it also adds bulk. The sensor may be ordered and used without the Interface Box to accommodate the needs of the user. Contact Velodyne Sales for details (i.e. SKUs).

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.

The schematic for the circuitry in the Interface Box may be found in *Wiring Diagrams on page 111*.

5.4.3 Power

The VLP-32C sensor does not have a physical power switch. The power source must be capable of providing as much as 3.0 A to accommodate the current surge that occurs during rotor spin-up when power is supplied to the sensor.

In normal operation the sensor draws between 9 and 18 W of power.

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.

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, get *Technical Support on page 98*.

Chapter 6 • Key Features

6.1 Calibrated Reflectivity

The VLP-32C 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 VLP-32C 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 100 for reflectivities from 0% to 100%.
- Retroreflectors report values from 101 to 255, where 255 represents an ideal reflection.

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

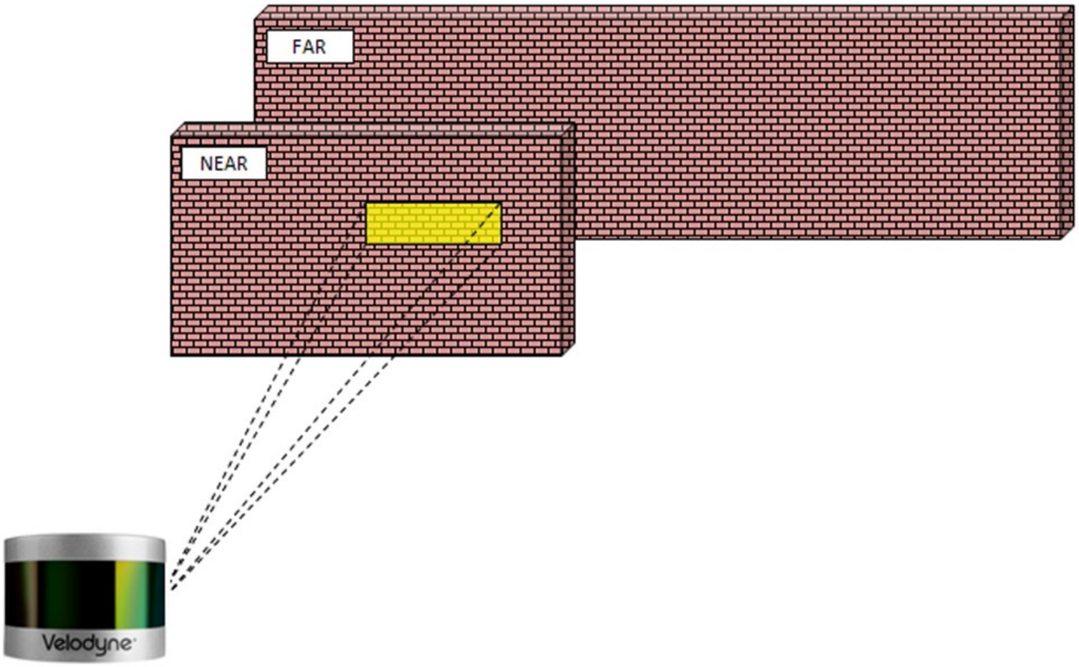
The VLP-32C supports three laser return modes: **Strongest**, **Last**, and **Dual**. A sensor can be configured to handle laser returns in one of these ways interactively via the sensor's web interface (where the setting is called Return Type) or programmatically via curl command. (See *Configuration Screen on page 76* or *Sensor Control with curl on page 83* 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 VLP-32C.

6.2.1 Single Return Modes: Strongest, Last

As shown in *Figure 6-1 on the facing page*, 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 125*.)

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.

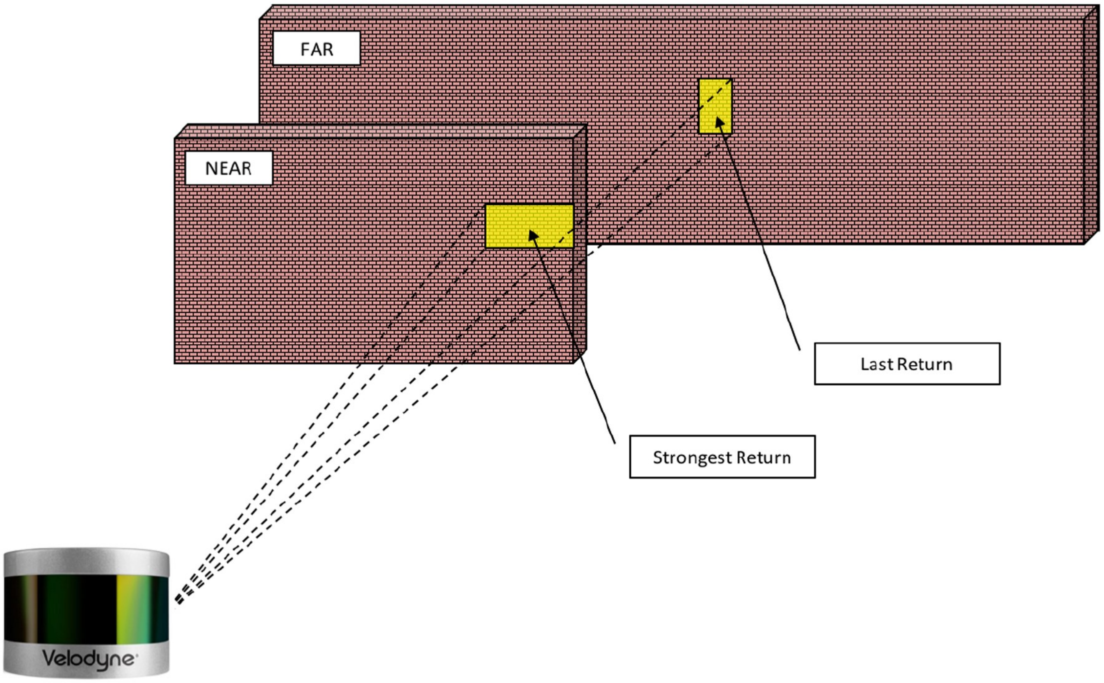
The VLP-32C analyzes multiple returns and reports either the strongest return, the last return, or both returns, depending on the laser return mode setting. If the setting is Strongest and a pulse produces multiple returns, only the strongest reflection results in a measurement. Likewise, if the setting 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 meter or more.

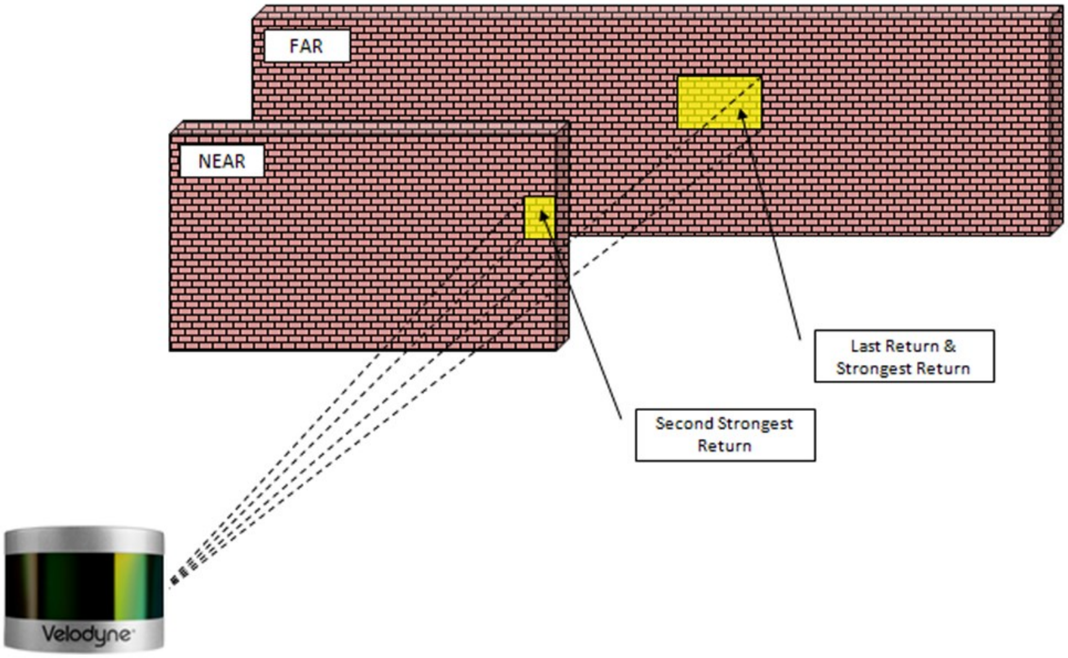
Figure 6-2 Dual Return with Last and Strongest Returns



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.

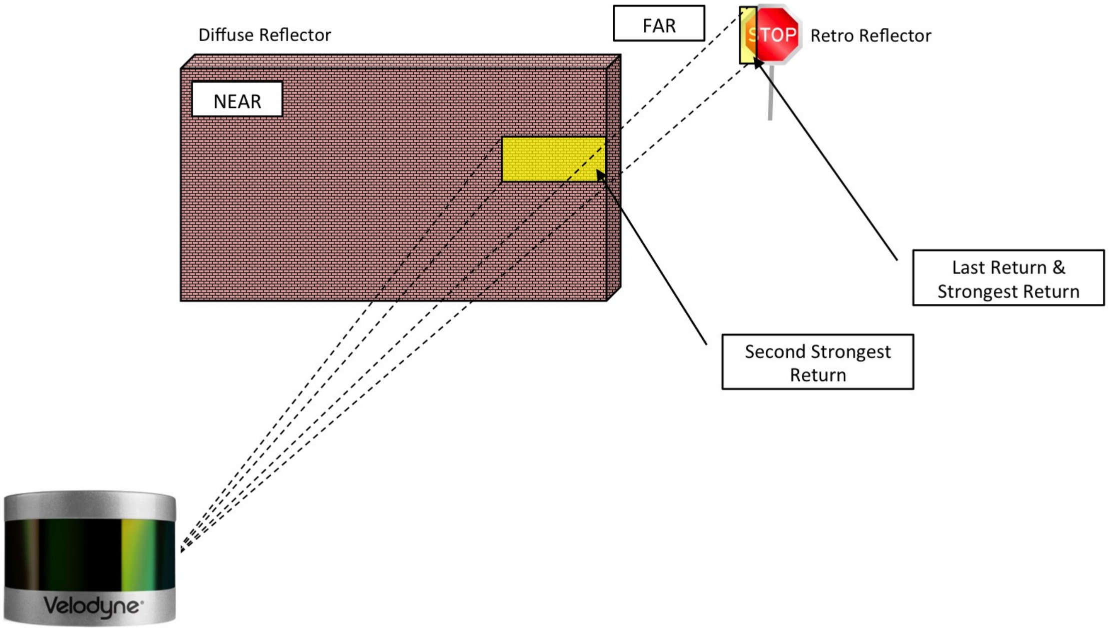
It's entirely possible, however, that the far wall might be far enough away that despite reflecting the majority of the beam, the return from the near wall would be the strongest return.

Figure 6-3 Dual Return with Second Strongest Return



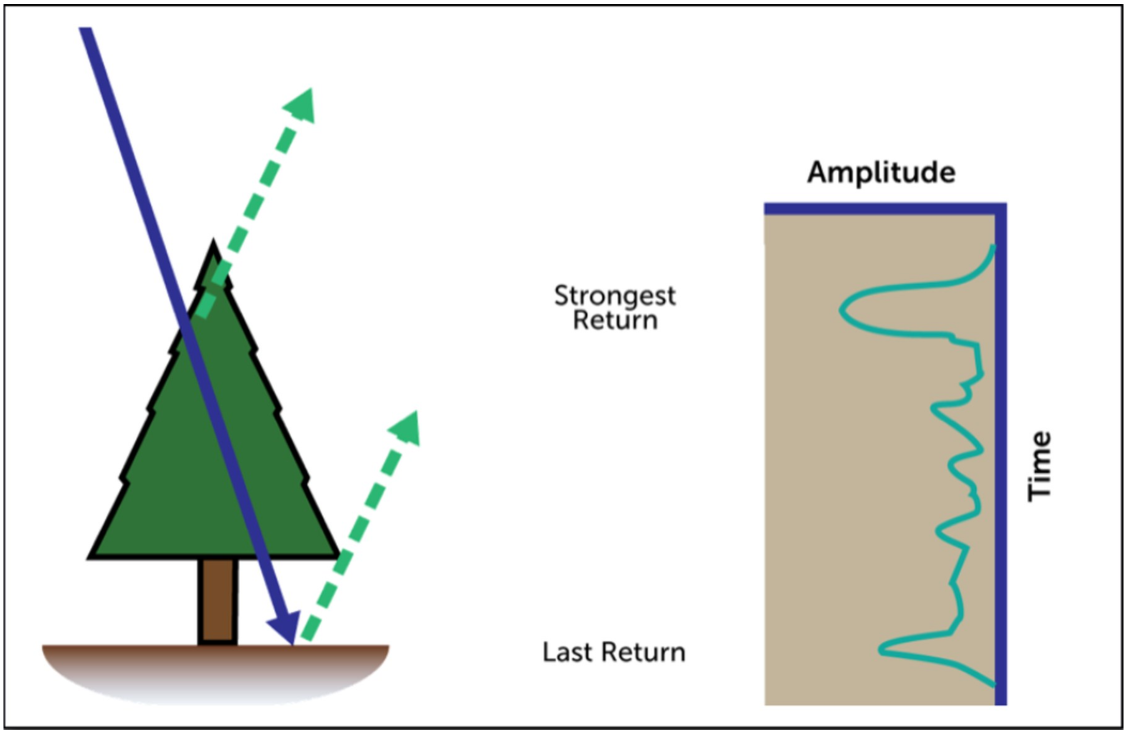
It's also possible that a small portion of the beam clips a retro-reflector and returns more energy than the majority of the beam, as in *Figure 6-4 on the next page*.

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.

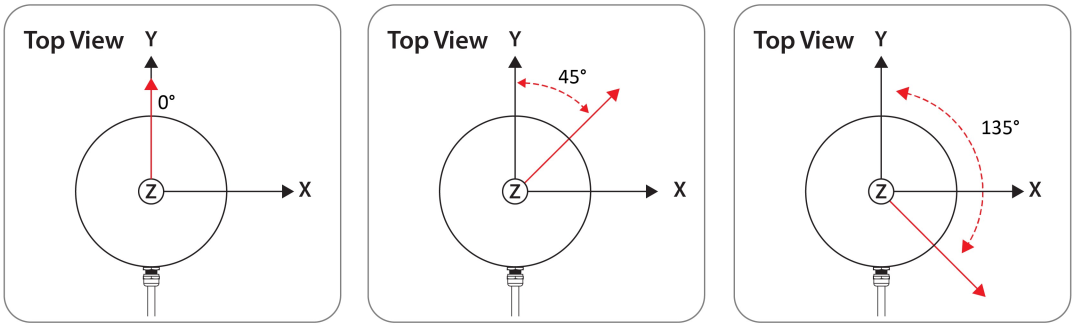
6.3 Phase Locking Multiple Sensors

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).

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 133*.

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 33*.

7.2 Interface Box Signals

The Interface Box is described in *Interface Box and Cable on page 33*.

Figure 7-1 Interface Box

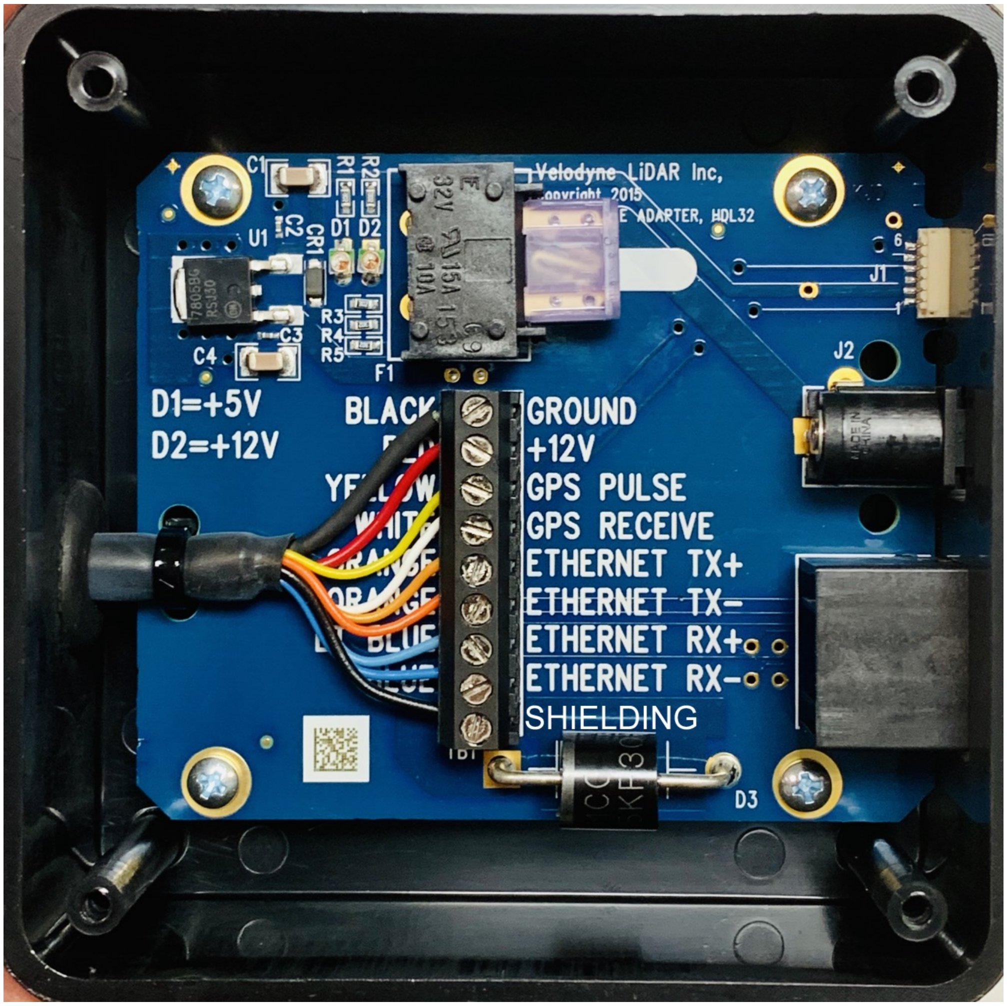


Table 7-1 Interface Box Signals (Terminal Block)

Wire Color	Signal	Input/Output
Black	Ground	Input
Red	Power	Input
Yellow	GPS Sync Pulse	Input
White	GPS Serial Receive	Input

Wire Color	Signal	Input/Output
Light Orange	Ethernet TX+	Output
Orange	Ethernet TX-	Output
Light Blue	Ethernet RX+	Input
Blue	Ethernet RX-	Input
Black	Shielding	tied to Ground

See *Wiring Diagrams on page 111* 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 RJ45 Ethernet connector on the Interface Box connects to any standard 100 Mbps Ethernet NIC or switch with MDI or AUTO MDIX capability.

See *Ethernet and Network Setup on page 140* 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 129* 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 RECEIVE.”

The PPS output from the GPS/INS is connected to the sensor’s Interface Box via the screw terminal labeled: “GPS PULSE.”

The ground signal from the GPS/INS is connected to the sensor’s Interface Box via the screw terminal labeled: “GROUND.” This ground is the reference signal the GPS RECEIVE and GPS PULSE 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 47* 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 7-2 Synchronizing PPS with NMEA Message

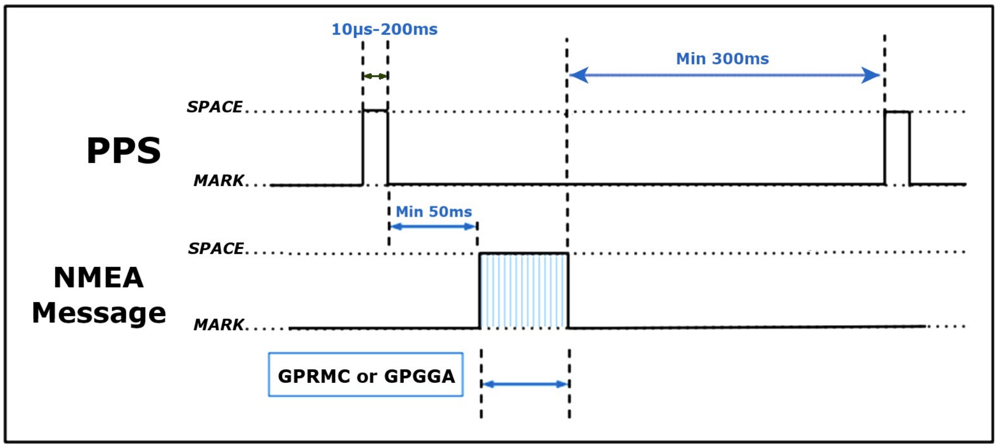


Figure 7-3 on the facing page is an example of a PPS sync pulse followed too closely by an NMEA message. The minimum 50 ms gap requirement between trailing edge of PPS and leading edge of NMEA message is not met.

Figure 7-3 PPS Signal Followed Too Closely by NMEA Message

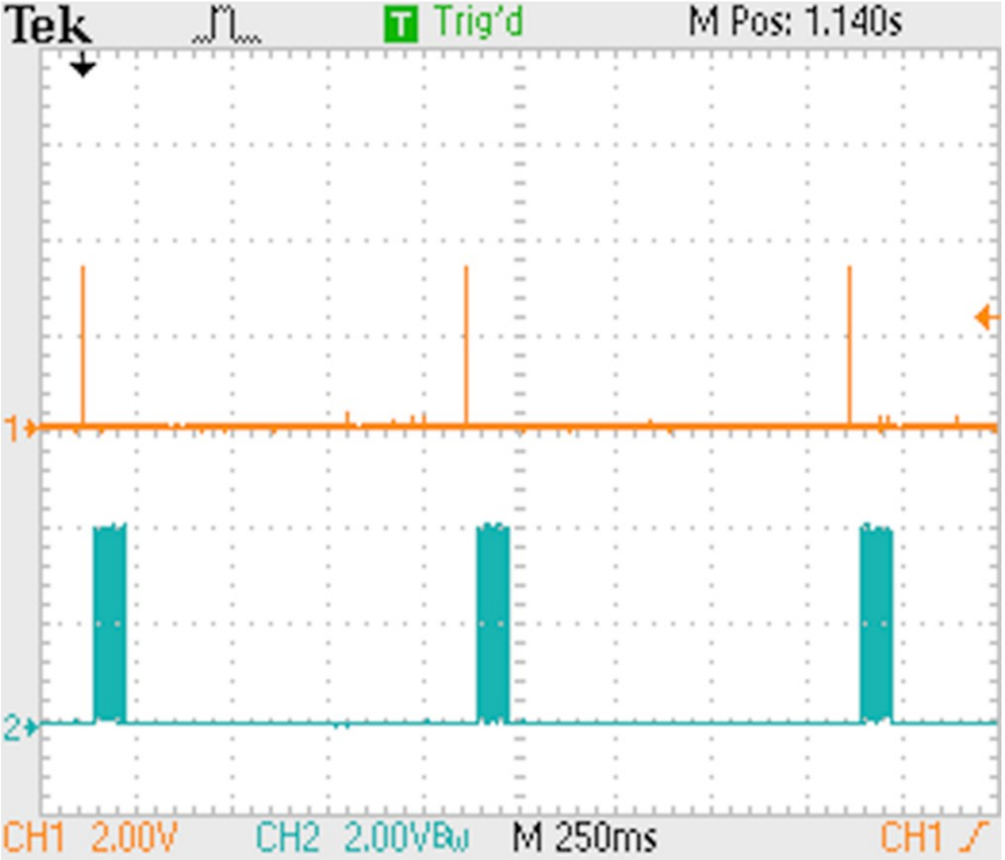
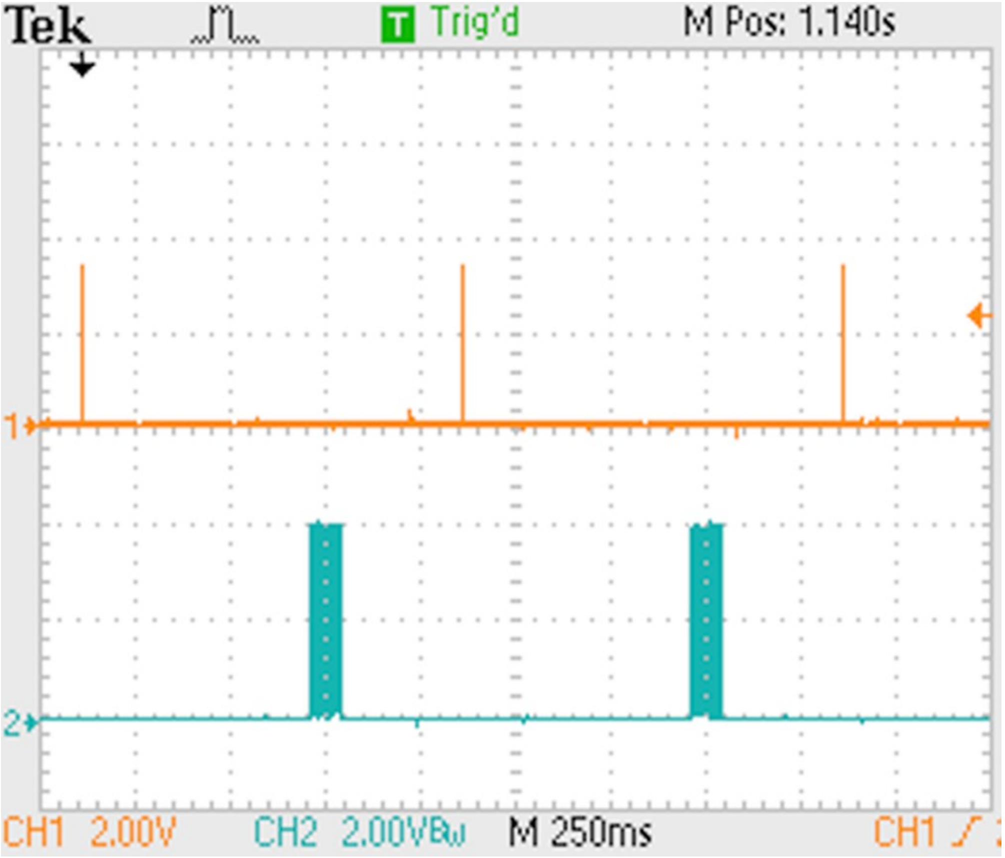


Figure 7-4 on the next page is an example of an NMEA message falling in the sweet spot between two PPS signals. The minimum 50 ms gap requirement between trailing edge of PPS and leading edge of NMEA message is met, and the minimum 300 ms gap requirement between trailing edge of the NMEA message and leading edge of PPS is met.

Figure 7-4 NMEA Message Leading PPS Signal by 320 ms



7.4.4 Sample Serial Transmission

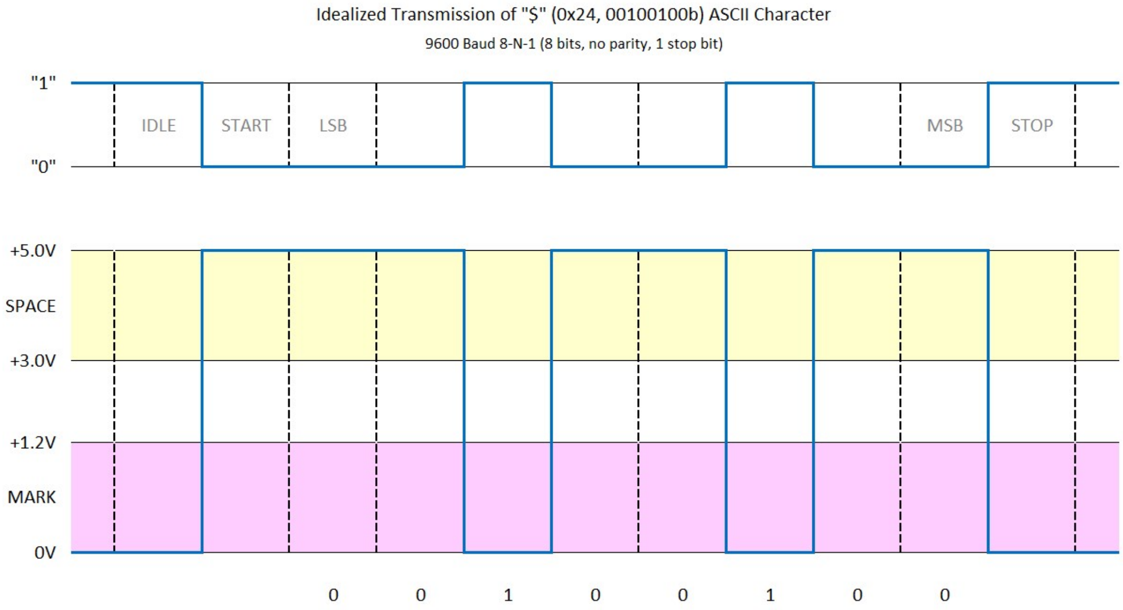
The serial connection for the NMEA message is TTL serial, or more commonly termed TTL RS-232. It is not True RS-232.

- 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 7-5 Sample Serial Transmission of $ Character



Idealized Transmission of "$" (0x24, 00100100b) ASCII Character
9600 Baud 8-N-1 (8 bits, no parity, 1 stop bit)
"1"
"0"
IDLE
START
LSB
MSB
STOP
+5.0V
SPACE
+3.0V
+1.2V
MARK
0V
0
0
1
0
0
1
0
0

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 7-2 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

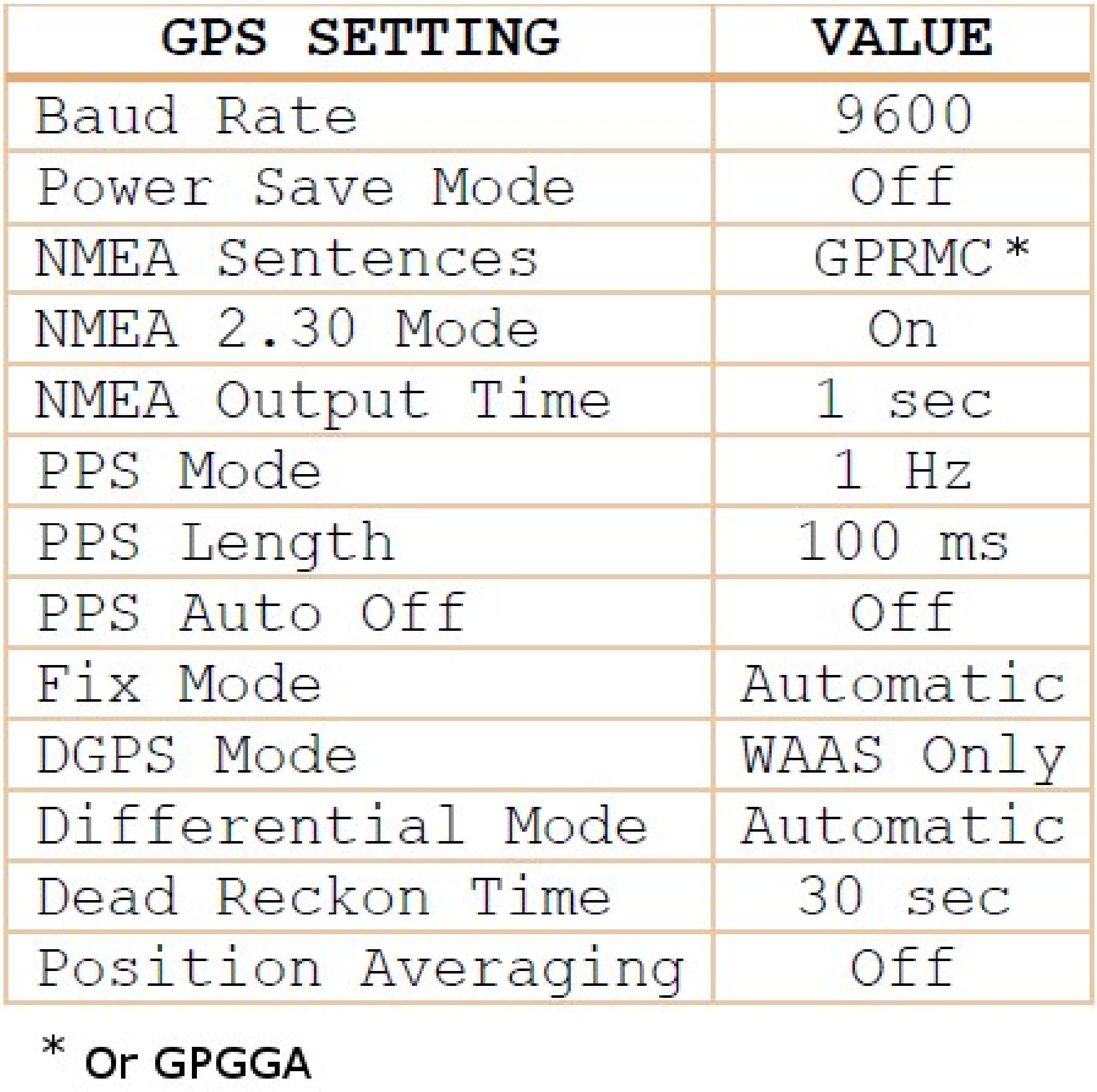
Depending on the user’s application, the source of the NMEA message (and PPS) can be a GPS/INS receiver or a substitute device such as a laptop or microcomputer.

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 7-6 below*) for optimized operation with your sensor. GPRMC is configured by default but GPGGA is also supported.

Figure 7-6 Garmin 18x LVC Configuration

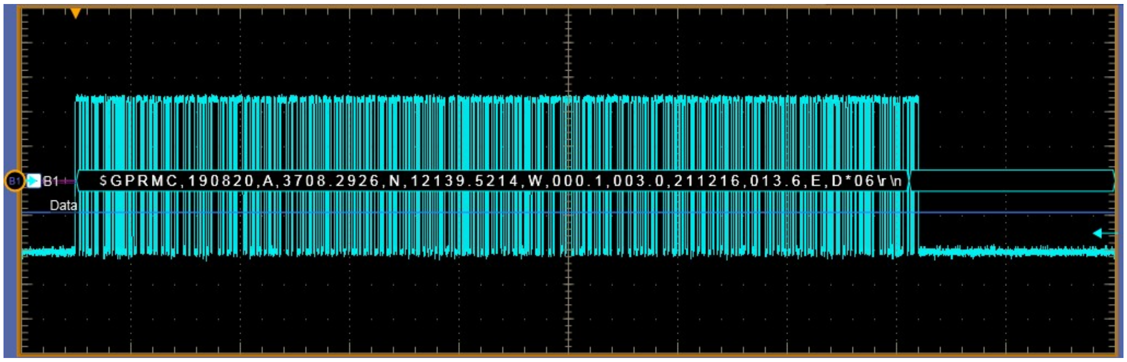


GPS SETTING	VALUE
Baud Rate	9600
Power Save Mode	Off
NMEA Sentences	GPRMC*
NMEA 2.30 Mode	On
NMEA Output Time	1 sec
PPS Mode	1 Hz
PPS Length	100 ms
PPS Auto Off	Off
Fix Mode	Automatic
DGPS Mode	WAAS Only
Differential Mode	Automatic
Dead Reckon Time	30 sec
Position Averaging	Off

* 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 41* where the signal is normally low and zeros are represented by the high voltage.

Figure 7-7 Sample NMEA Message

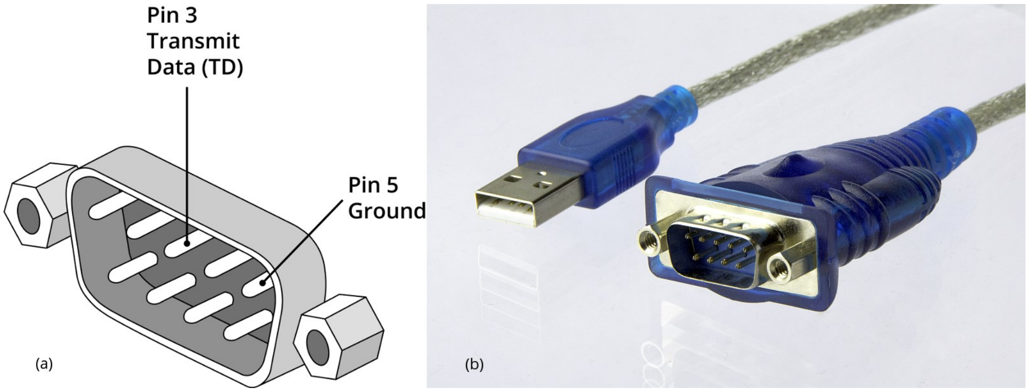


```
$GPRMC,190820,A,3708.2926,N,12139.5214,W,000.1,003.0,211216,013.6,E,D*06\r\n
```

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 7-8 DB9 Pin-outs (DTE) and USB-to-Serial Adapter



An example of a standard DB9 serial port is shown in *Figure 7-8 above* (a). Because modern computers use USB ports, a USB to Serial Adapter, as shown in *Figure 7-8 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 RECEIVE
- DB9 pin 5 connects to GROUND

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 RECEIVE and GROUND connections on the terminal strip.

Figure 7-9 on the next page shows a signal directly from a Raspberry Pi UART output and *Figure 7-10 on the next page* shows the same output inverted into a signal compatible with your Velodyne sensor.

Figure 7-9 Signal Directly from UART (incorrect polarity)

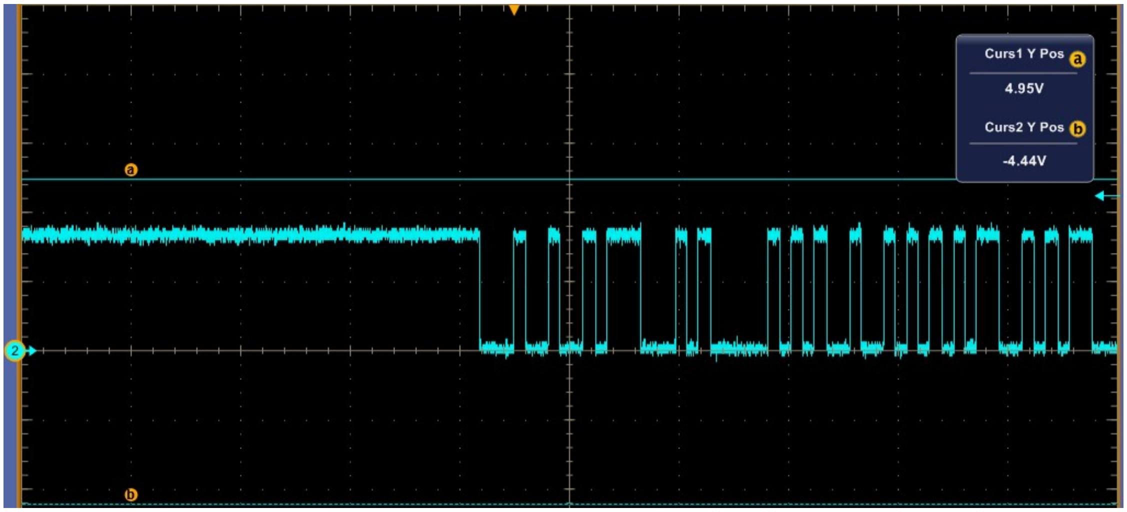
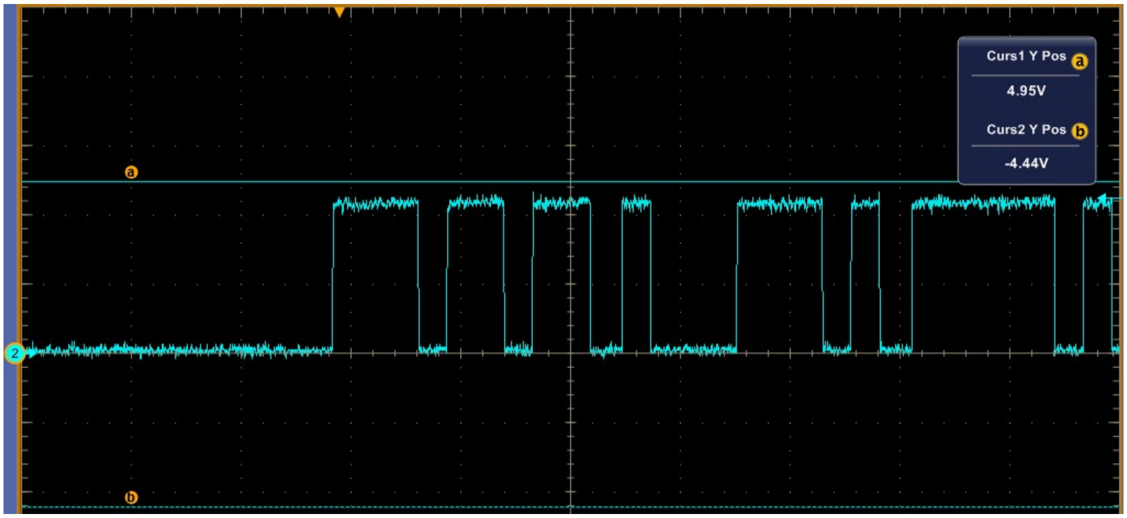


Figure 7-10 Inverted Signal from UART (correct polarity)



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 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 7-3 on the facing page provides a description of the pre-NMEA 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 7-3 Pre-NMEA 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 Version 2.3 Message Format

Table 7-4 below provides a description of the post-NMEA 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 7-4 Post-NMEA 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

Value	Description
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 129*.

7.4.8 Accepting NMEA Messages Via Ethernet

Your sensor can accept NMEA 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
- http://arundale.com/docs/ais/nmearouter.html

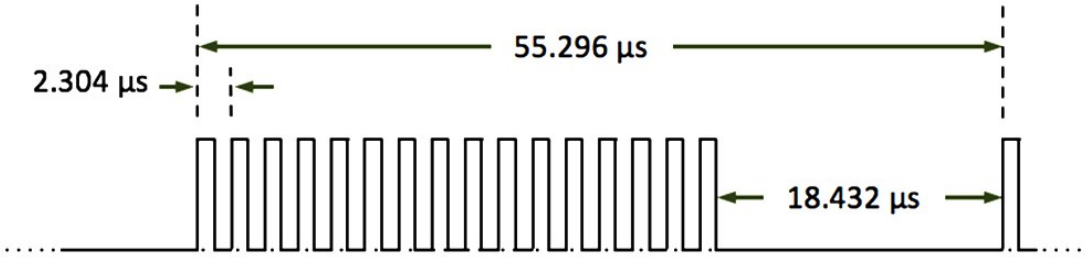
Chapter 8 • Sensor Operation

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

All 32 lasers are fired and recharged every 55.296 µs. Lasers are fired in pairs. The time between firings is 2.304 µs. There are 16 firings (16 × 2.304 µs) followed by an idle period of 18.43 µs. Therefore, the timing cycle to fire and recharge all 32 lasers is given by (16 × 2.304 µs) + 18.432 µs = 55.296 µs.

Figure 8-1 Firing Sequence Timing



8.2 Throughput Calculations

Some terms used here are defined in *Packet Types and Definitions on page 58*. The *Data Packet Structure on page 61* and *Position Packet Structure on page 63* are defined in Chapter 9.

8.2.1 Data Packet Rate

There are 12 firing cycles in a data packet. 12 x 55.296 µs = 0.663552 ms is the accumulation delay per packet.

1 packet/0.663552 ms ≈ 1507 packets/second

1248 bytes/packet * 1507 packets/second ≈ 1880736 bytes/second

8.2.2 Position Packet Rate

Position packets arrive at approximately 1/14th the rate of data packets. (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.)

1507 data packets/second / 14 ≈ 107.65 position packets/second

554 bytes/position packet * (1507 packets/second / 14) = 59634 bytes/second

8.2.3 Total Packet Rate

Total packet rate is approximately 1507 data packets/second + 107.65 position packets/second = 1614.65 packets/second.

Summing yields 1940370 bytes/second for single return mode.

Dual return mode doubles the data packet rate but not the position packet rate: (2 * 1880736 bytes/second) + 59634 bytes/second = 3821106 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.

32 data points/data block * 12 data blocks/packet = 384 data points/packet

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

384 data points/packet * 1507 packets/second = 578688 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.

32 data points/data block * 12 data blocks/packet = 384 data points/packet

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

384 data points/packet * 3014 packets/second = 1157376 laser measurements per second

8.3 Rotation Speed (RPM)

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

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 76* or *Sensor Control with curl on page 83* 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 depends 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 cart-based mapping solution might use a lower RPM to increase the detail in each frame.

8.3.1 Horizontal Angular (Azimuth) Resolution

The firing timing of the sensor is fixed at 55.296 µs per firing sequence. Only changing the speed of rotation changes the angular, azimuthal 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

$$\boxed{\text{Azimuth}_{\text{Resolution}} = 600\,\frac{rev}{min} \times \frac{1}{60}\,\frac{min}{s} \times 360\,\frac{^\circ}{rev} \times \frac{55.296\times10^{-6}\,s}{firing\ cycle}}$$

$$\text{Azimuth}_{\text{Resolution}} = 0.199^\circ / firing\ cycle$$

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

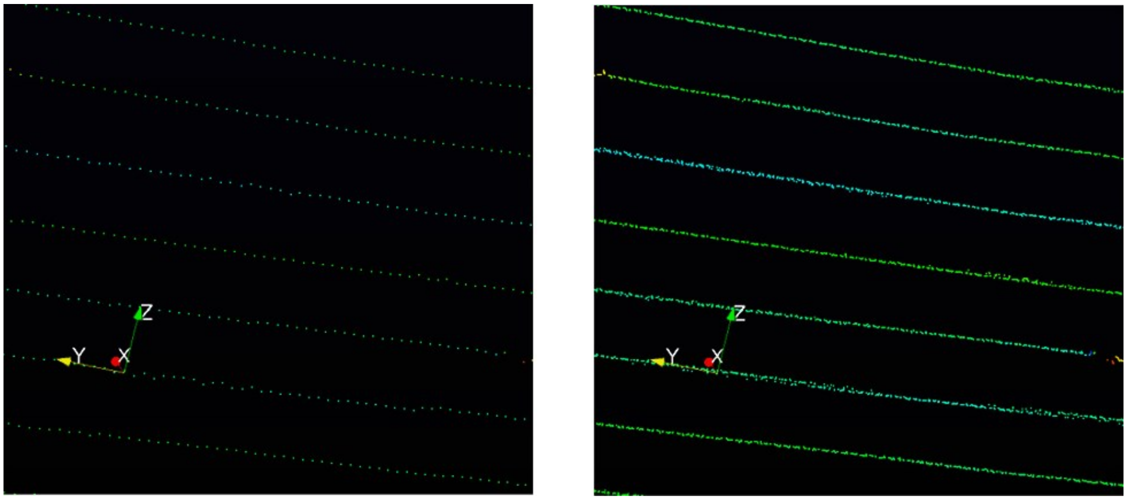
RPM	Resolution
300	0.1°
600	0.2°
900	0.3°
1200	0.4°

8.3.2 Rotation Speed Fluctuation and Point Density

The sensor uses a feedback control function to maintain its rotational speed within ±3 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 8-2 on the next 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 8-2 Point Density Example



Chapter 9 • Sensor Data

This chapter provides detailed information about sensor data characteristics.

9.1 Sensor Origin and Frame of Reference

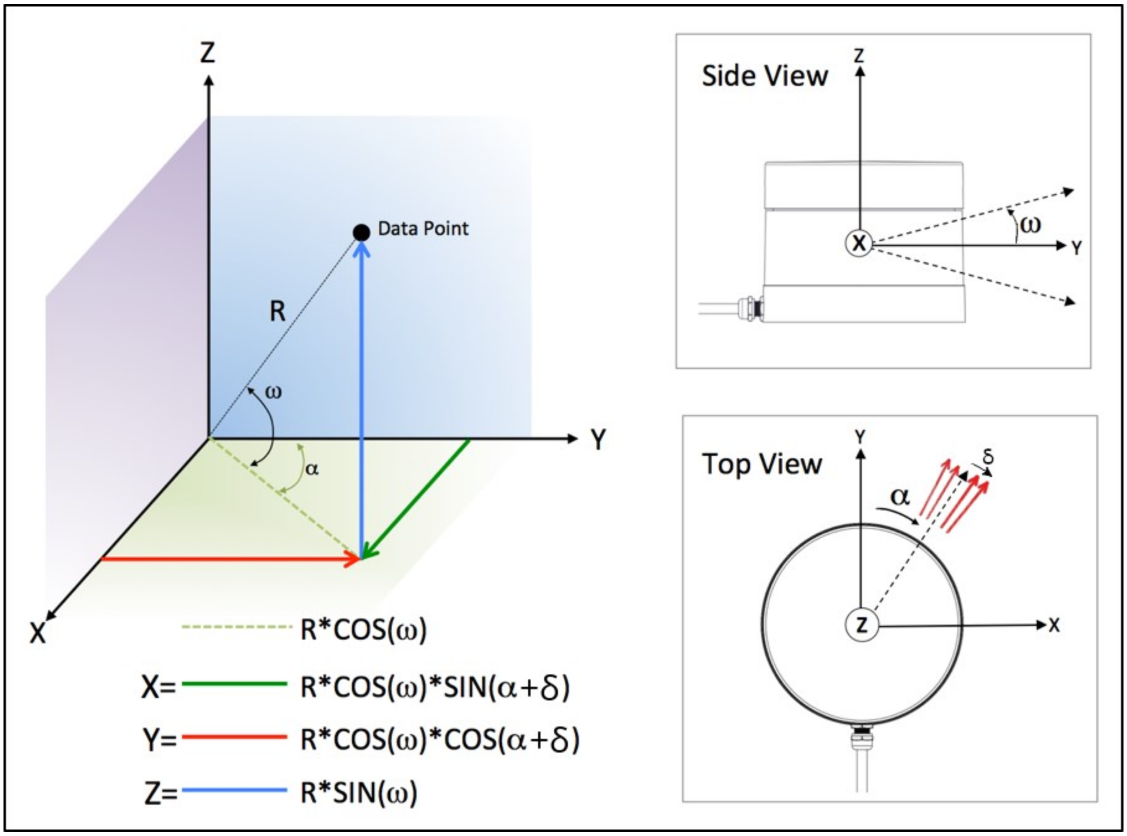
The sensor reports distances relative to itself in spherical coordinates (radius r, elevation ω, azimuth α).

Sensor data origin (0,0,0) is 37.34 mm above the sensor base, on the center axis, as shown in *Figure 9-1 on the next page* (see the side and top views), which also shows the sensor's frame of reference. See also *VLP-32C Mechanical Drawing on page 110*.

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 next page* lists the formulas for converting spherical coordinates (R, ω, α) to Cartesian coordinates (X, Y, Z).

Figure 9-1 VLP-32C Sensor Coordinate System



9.3 Packet Types and Definitions

There are two types of 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 43* 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 sometimes referred to as a firing group. A firing sequence is not allowed to span multiple data packets.

It takes 55.296 µs to fire all 32 lasers in a VLP-32C and recharge. VLP-32C sensors fire their lasers in pairs.

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 VLP-32C sensors, no laser channels are inline (vertically) with the Azimuth in a firing sequence. Instead, each channel is offset by one of four different azimuthal offsets.

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 34*. 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 32 lasers is contained in each data block. Each data packet has 12 data blocks.

Only one Azimuth is returned per data block.

A data block consists of 100 bytes of binary data:

- A two-byte flag (0xFFEE)
- 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 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 the moment of the first data point in the first firing sequence of the first data block. 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 43*. A full description of timing options can be found in *Time Synchronization on page 129*.

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 *Data Packet Structure on the facing 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)
--	--	VLP-32C VLP-32MR	0x28 (40)
--	--	Velarray	0x31 (49)
--	--	VLS-128	0xA1 (161)

9.3.2 Data Packet Structure

A data packet is 1248 bytes long and sent via a UDP packet on port 2368. The data packet is comprised of 42 bytes of protocol header, twelve Data Blocks, a four-byte timestamp, and two factory bytes.

There are two formats for the data packet:

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

See *Laser Return Modes on page 34* for an illustration of what Strongest, Last, and Dual mean in this context.

The packet data structure for Single Return Mode is shown in *Figure 9-2 below*. The packet data structure for Dual Return Mode is shown in *Figure 9-3 on the next page*.

There are several key differences between the packet structures.

First, in Dual Return Mode the sensor sends a pair of data blocks for each azimuth angle firing. The odd numbered blocks (1, 3, ..., 9, 11) contain either the strongest or second-strongest return and the even numbered blocks (0, 2, ..., 8, 10) contain the last return.

If the strongest return is also the last return, then the second-strongest return is provided. If only one return was detected, the data will be identical in the even|odd block pairs (0|1, 2|3, 4|5, 6|7, 8|9, 10|11).

Figure 9-2 VLP-32 Single Return Mode Data Structure

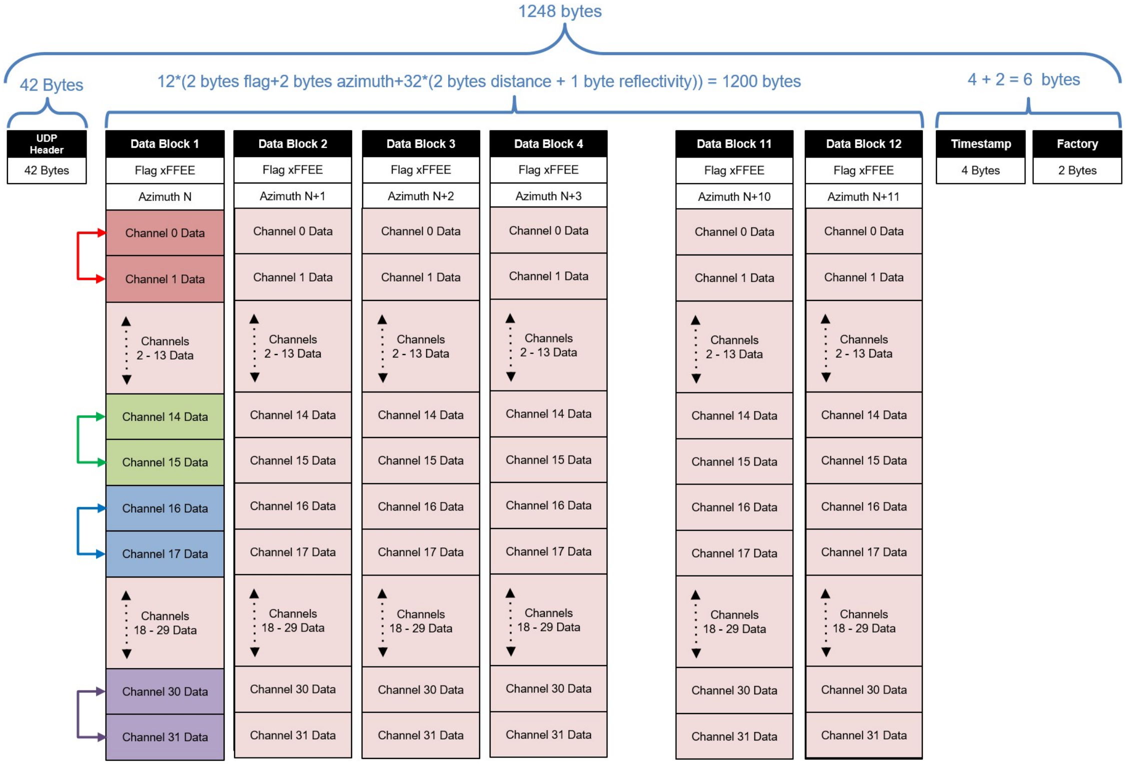
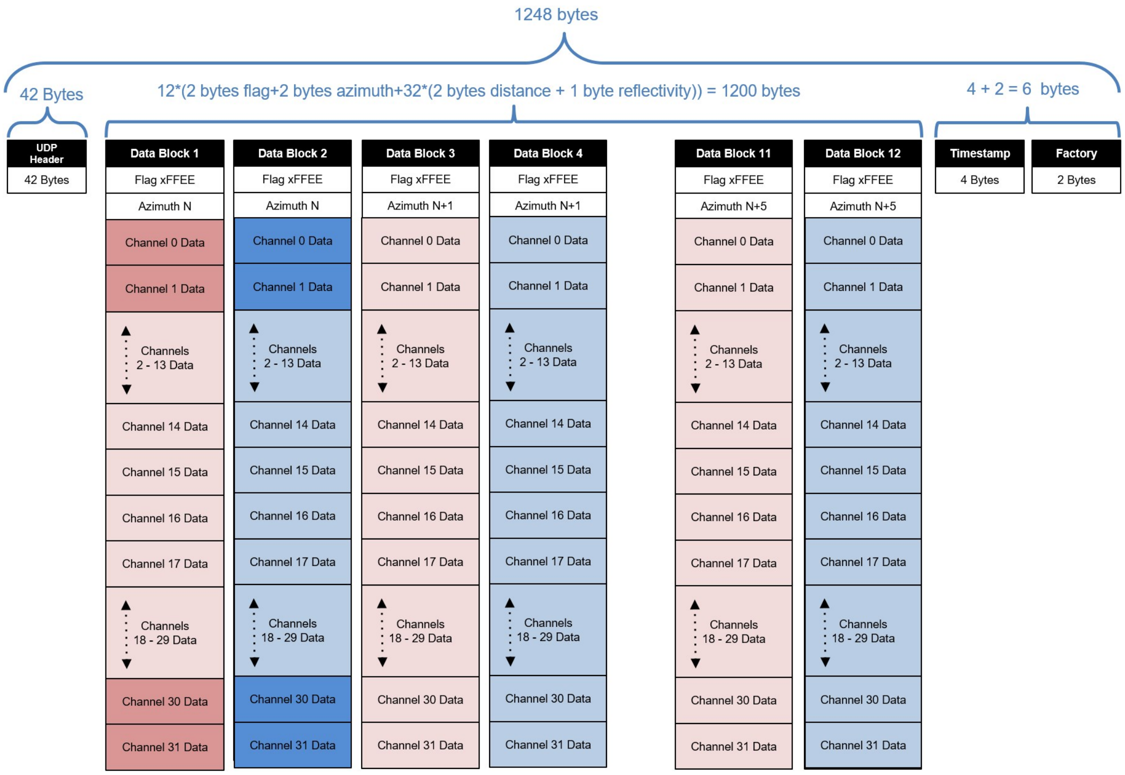


Figure 9-3 VLP-32 Dual Return Mode Data Structure



Lasers within each data block are ordered as shown in *Table 9-2 below*, which also shows the aim of each laser relative to the 0 plane (elevation angle ω) and the data block's Azimuth (azimuth offset α). See *Figure 9-1 on page 58*'s "Top View" inset for a visual representation of azimuth offset (δ).

Table 9-2 VLP-32C Data Order in Data Block

Laser ID	Elevation Angle (°)	Azimuth Offset (°)
0	-25	-1.4
1	-1	4.2
2	-1.667	-1.4
3	-15.639	1.4
4	-11.31	-1.4
5	0	1.4
6	-0.667	-4.2
7	-8.843	1.4
8	-7.254	-1.4
9	0.333	4.2
10	-0.333	-1.4

Laser ID	Elevation Angle (°)	Azimuth Offset (°)
11	-6.148	1.4
12	-5.333	-4.2
13	1.333	1.4
14	0.667	-4.2
15	-4	1.4
16	-4.667	-1.4
17	1.667	4.2
18	1	-1.4
19	-3.667	4.2
20	-3.333	-4.2
21	3.333	1.4
22	2.333	-1.4
23	-2.667	1.4
24	-3	-1.4
25	7	1.4
26	4.667	-1.4
27	-2.333	4.2
28	-2	-4.2
29	15	1.4
30	10.333	-1.4
31	-1.333	1.4

9.3.3 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 9-3 on the next page*.

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 50* for details on the NMEA messages supported.

Table 9-3 Position Packet Structure Field Offsets

Description	Number of Bytes	Address Offset	Data Range
Reserved	187	0x00-0xBA	unused (null bytes)
Temperature of top board	1	0xBB	-128 to +127°C
Temperature of bottom board	1	0xBC	-128 to +127°C
Temperature when ADC calibration last ran	1	0xBD	0 to 150°C
Change in temperature since last ADC calibration	2	0xBE-0xBF	-150 to 150°C
Elapsed seconds since last ADC calibration	4	0xC0-0xC3	0 to 2^{32}-1
Reason for the last ADC calibration	1	0xC4	0: No calibration 1: Power-on calibration performed 2: Manual calibration performed 3: Delta temperature calibration performed 4: Periodic calibration performed
Bitmask indicating current status of ADC calibration	1	0xC5	These bit fields are not mutually exclusive. The condition is True if the bit field value is 1. b0: Calibration in progress b1: Delta temperature limit has been met b2: Periodic time elapsed limit has been met Each status bit reflects the current sensor condition but does not imply any calibration occurred.
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
Reserved	178	0x14E-0x1FF	unused (null bytes)

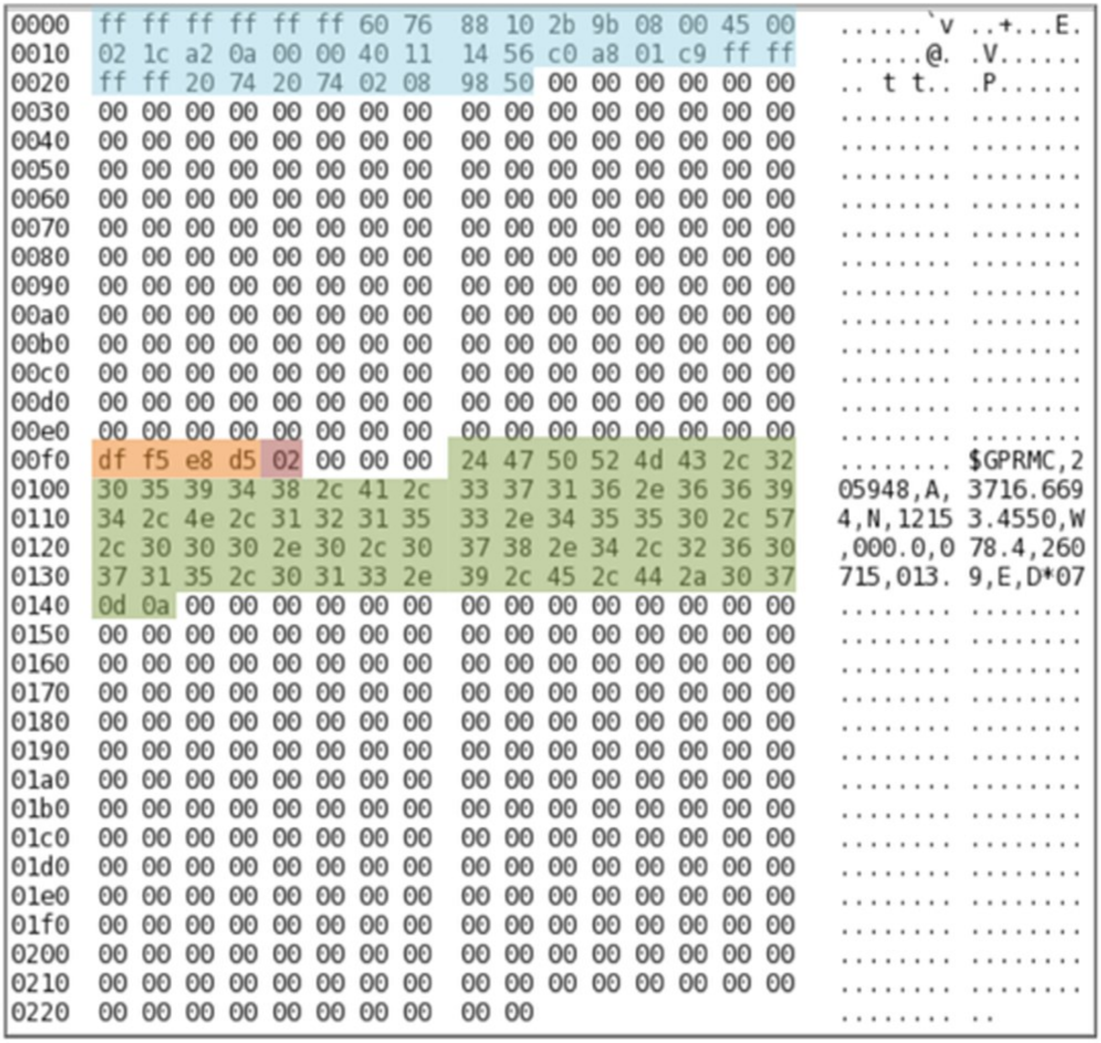
The GPRMC sentence is terminated with CR/LF and padded to end of payload with null bytes.

Table 9-4 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 9-4 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:

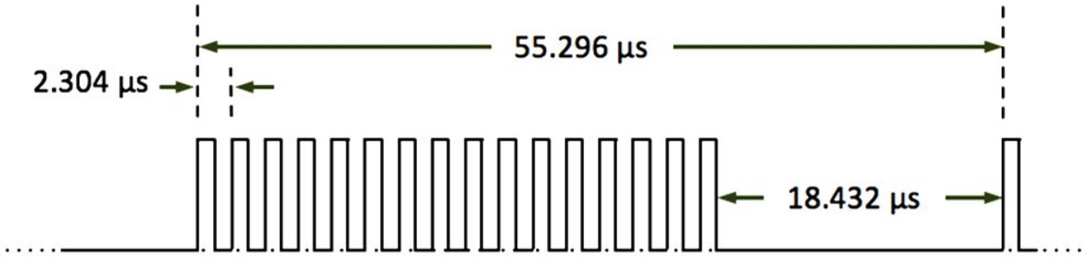
- light blue: Ethernet+IP+UDP packet header (42 bytes)
- orange: Timestamp (4 bytes)
- red: PPS status (1 byte)
- green: NMEA GPRMC sentence (length varies)

9.4 Discreet Point Timing Calculation

For the purpose of geo-referencing lidar data, 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 an INS or other source. By using the firing timing information given in *Figure 9-5 on the facing page* you are able to compute the exact firing time for each data point within a firing sequence.

All 32 lasers are fired in pairs and recharged every 55.296 µs. The cycle time between firings is 2.304 µs. There are 16 firings (16 × 2.304 µs) followed by a recharge period of 18.43 µs. Therefore, the timing cycle to fire and recharge all 32 lasers is given by (16 × 2.304 µs) + 18.43 µs = 55.296 µs.

Figure 9-5 Firing Sequence Timing



VLP-32C lasers are fired in pairs at widely separated angles to minimize IR crosstalk. The sequence of firings is specified in *Table 9-5 below*. The first pair to fire is lasers 0 and 1. The second are 2 and 3. And so on.

Table 9-5 VLP-32C's Lasers Fire in Pairs

Firing Order	Laser ID	Elevation Angle (°)	Elevation Angle (°)	Laser ID
0	0	-25	-1	1
1	2	-1.667	-15.639	3
2	4	-11.31	0	5
3	6	-0.667	-8.843	7
4	8	-7.254	0.333	9
5	10	-0.333	-6.148	11
6	12	-5.333	1.333	13
7	14	0.667	-4	15
8	16	-4.667	1.667	17
9	18	1	-3.667	19
10	20	-3.333	3.333	21
11	22	2.333	-2.667	23
12	24	-3	7	25
13	26	4.667	-2.333	27
14	28	-2	15	29
15	30	10.333	-1.333	31

To calculate the exact time, in microseconds, of each data point in a packet, first number the points in the firing sequence from 0 to 31. This becomes the data point index (aka Laser ID) for your calculations. Next, number the firing sequences 0 to 11. This becomes your sequence index.

The timestamp in the packet indicates the time of the first data point in the packet. You'll need to calculate a time offset for each data point and then add that offset to the timestamp. See *Figure 9-7 on page 70* for specific Data Point Timing Offsets (Single Return Mode) and *Figure 9-8 on page 71* for specific Data Point Timing Offsets (Dual Return Mode).

The offset equation is given by:

```
TimeOffset = (55.296 µs * SequenceIndex) + (2.304 µs * DataPointIndex)
```

To calculate the exact point time, add the TimeOffset to the timestamp.

```
ExactPointTime = Timestamp + TimeOffset
```

Example: Calculate the timing offsets of every point in a packet, in single or dual return mode. The following is working Python code:

```
def make_table(dual_mode):
    timing_offsets = [[0.0 for x in range(12)] for y in range(32)] # Init matrix
    # constants
    full_firing_cycle = 55.296 # µs
    single_firing = 2.304 # µs
    # compute timing offsets
    for x in range(12):
        for y in range(32):
            dataBlockIndex = x / 2 if dual_mode else x
            dataPointIndex = y / 2
            timing_offsets[y][x] = \
                (full_firing_cycle * dataBlockIndex) + (single_firing * dataPointIndex)
    return timing_offsets
```

To verify the values are correct, you can print them out and compare them with values in *Figure 9-7 on page 70* and *Figure 9-8 on page 71*.

```
timing_offsets = make_table(False)  # False : single return mode
print('\n'.join([', '.join(['{:8.3f}'.format(value) for value in row])
    for row in timing_offsets]))
```

Example: Calculate the exact firing time of the last firing in a packet if the timestamp value is 45,231,878 µs.

```
Timestamp = 45,231,878 µs
TimeOffset = timing_offsets[31][11]
ExactPointTime = Timestamp + TimeOffset
ExactPointTime = 45,231,878 + 642.816 µs
ExactPointTime = 45,232,520.816 µs
```

Figure 9-6 on the facing page shows the timing calculation for a few data points.

Figure 9-6 Example Data Point Timing Calculation

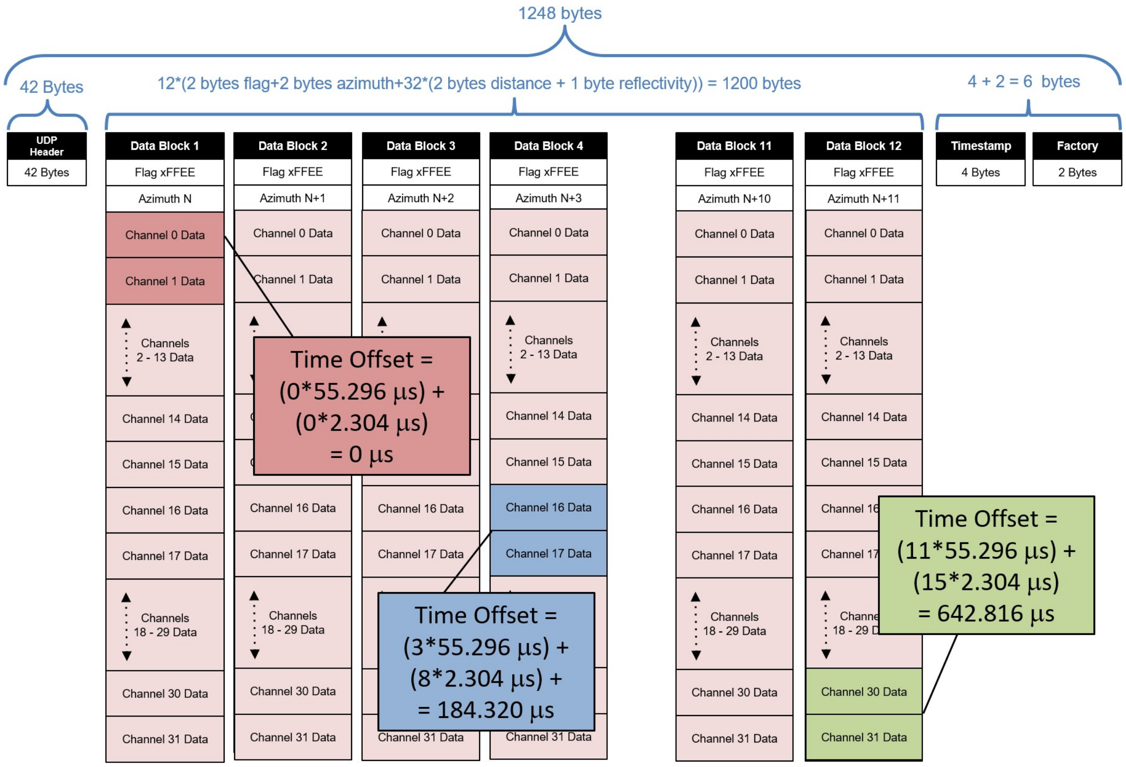
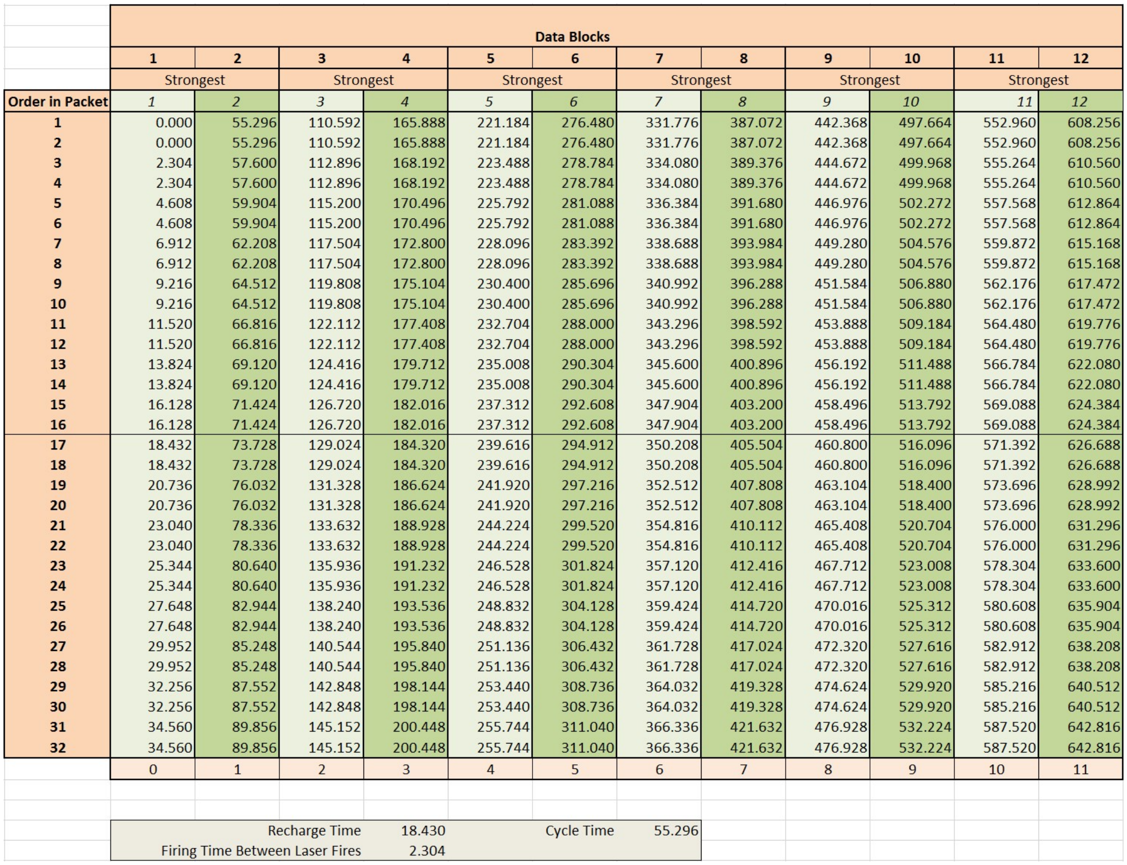


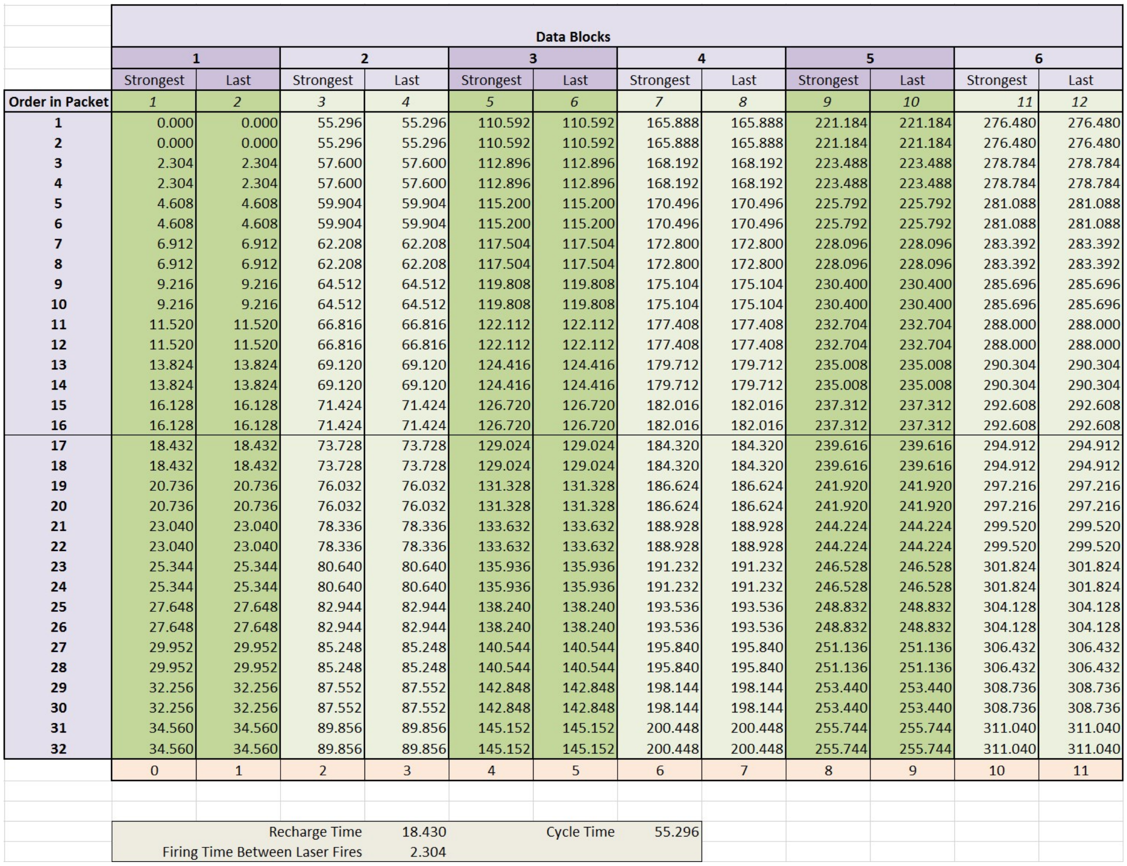
Figure 9-7 Single Return Mode Timing Offsets (in µs)



	Data Blocks											
	1	2	3	4	5	6	7	8	9	10	11	12
	Strongest		Strongest		Strongest		Strongest		Strongest		Strongest	
Order in Packet	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
1	0.000	55.296	110.592	165.888	221.184	276.480	331.776	387.072	442.368	497.664	552.960	608.256
2	0.000	55.296	110.592	165.888	221.184	276.480	331.776	387.072	442.368	497.664	552.960	608.256
3	2.304	57.600	112.896	168.192	223.488	278.784	334.080	389.376	444.672	499.968	555.264	610.560
4	2.304	57.600	112.896	168.192	223.488	278.784	334.080	389.376	444.672	499.968	555.264	610.560
5	4.608	59.904	115.200	170.496	225.792	281.088	336.384	391.680	446.976	502.272	557.568	612.864
6	4.608	59.904	115.200	170.496	225.792	281.088	336.384	391.680	446.976	502.272	557.568	612.864
7	6.912	62.208	117.504	172.800	228.096	283.392	338.688	393.984	449.280	504.576	559.872	615.168
8	6.912	62.208	117.504	172.800	228.096	283.392	338.688	393.984	449.280	504.576	559.872	615.168
9	9.216	64.512	119.808	175.104	230.400	285.696	340.992	396.288	451.584	506.880	562.176	617.472
10	9.216	64.512	119.808	175.104	230.400	285.696	340.992	396.288	451.584	506.880	562.176	617.472
11	11.520	66.816	122.112	177.408	232.704	288.000	343.296	398.592	453.888	509.184	564.480	619.776
12	11.520	66.816	122.112	177.408	232.704	288.000	343.296	398.592	453.888	509.184	564.480	619.776
13	13.824	69.120	124.416	179.712	235.008	290.304	345.600	400.896	456.192	511.488	566.784	622.080
14	13.824	69.120	124.416	179.712	235.008	290.304	345.600	400.896	456.192	511.488	566.784	622.080
15	16.128	71.424	126.720	182.016	237.312	292.608	347.904	403.200	458.496	513.792	569.088	624.384
16	16.128	71.424	126.720	182.016	237.312	292.608	347.904	403.200	458.496	513.792	569.088	624.384
17	18.432	73.728	129.024	184.320	239.616	294.912	350.208	405.504	460.800	516.096	571.392	626.688
18	18.432	73.728	129.024	184.320	239.616	294.912	350.208	405.504	460.800	516.096	571.392	626.688
19	20.736	76.032	131.328	186.624	241.920	297.216	352.512	407.808	463.104	518.400	573.696	628.992
20	20.736	76.032	131.328	186.624	241.920	297.216	352.512	407.808	463.104	518.400	573.696	628.992
21	23.040	78.336	133.632	188.928	244.224	299.520	354.816	410.112	465.408	520.704	576.000	631.296
22	23.040	78.336	133.632	188.928	244.224	299.520	354.816	410.112	465.408	520.704	576.000	631.296
23	25.344	80.640	135.936	191.232	246.528	301.824	357.120	412.416	467.712	523.008	578.304	633.600
24	25.344	80.640	135.936	191.232	246.528	301.824	357.120	412.416	467.712	523.008	578.304	633.600
25	27.648	82.944	138.240	193.536	248.832	304.128	359.424	414.720	470.016	525.312	580.608	635.904
26	27.648	82.944	138.240	193.536	248.832	304.128	359.424	414.720	470.016	525.312	580.608	635.904
27	29.952	85.248	140.544	195.840	251.136	306.432	361.728	417.024	472.320	527.616	582.912	638.208
28	29.952	85.248	140.544	195.840	251.136	306.432	361.728	417.024	472.320	527.616	582.912	638.208
29	32.256	87.552	142.848	198.144	253.440	308.736	364.032	419.328	474.624	529.920	585.216	640.512
30	32.256	87.552	142.848	198.144	253.440	308.736	364.032	419.328	474.624	529.920	585.216	640.512
31	34.560	89.856	145.152	200.448	255.744	311.040	366.336	421.632	476.928	532.224	587.520	642.816
32	34.560	89.856	145.152	200.448	255.744	311.040	366.336	421.632	476.928	532.224	587.520	642.816
	0	1	2	3	4	5	6	7	8	9	10	11

Recharge Time	18.430	Cycle Time	55.296
Firing Time Between Laser Fires	2.304		

Figure 9-8 Dual Return Mode Timing Offsets (in µs)



	Data Blocks											
	1		2		3		4		5		6	
	Strongest	Last	Strongest	Last	Strongest	Last	Strongest	Last	Strongest	Last	Strongest	Last
Order in Packet	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*
1	0.000	0.000	55.296	55.296	110.592	110.592	165.888	165.888	221.184	221.184	276.480	276.480
2	0.000	0.000	55.296	55.296	110.592	110.592	165.888	165.888	221.184	221.184	276.480	276.480
3	2.304	2.304	57.600	57.600	112.896	112.896	168.192	168.192	223.488	223.488	278.784	278.784
4	2.304	2.304	57.600	57.600	112.896	112.896	168.192	168.192	223.488	223.488	278.784	278.784
5	4.608	4.608	59.904	59.904	115.200	115.200	170.496	170.496	225.792	225.792	281.088	281.088
6	4.608	4.608	59.904	59.904	115.200	115.200	170.496	170.496	225.792	225.792	281.088	281.088
7	6.912	6.912	62.208	62.208	117.504	117.504	172.800	172.800	228.096	228.096	283.392	283.392
8	6.912	6.912	62.208	62.208	117.504	117.504	172.800	172.800	228.096	228.096	283.392	283.392
9	9.216	9.216	64.512	64.512	119.808	119.808	175.104	175.104	230.400	230.400	285.696	285.696
10	9.216	9.216	64.512	64.512	119.808	119.808	175.104	175.104	230.400	230.400	285.696	285.696
11	11.520	11.520	66.816	66.816	122.112	122.112	177.408	177.408	232.704	232.704	288.000	288.000
12	11.520	11.520	66.816	66.816	122.112	122.112	177.408	177.408	232.704	232.704	288.000	288.000
13	13.824	13.824	69.120	69.120	124.416	124.416	179.712	179.712	235.008	235.008	290.304	290.304
14	13.824	13.824	69.120	69.120	124.416	124.416	179.712	179.712	235.008	235.008	290.304	290.304
15	16.128	16.128	71.424	71.424	126.720	126.720	182.016	182.016	237.312	237.312	292.608	292.608
16	16.128	16.128	71.424	71.424	126.720	126.720	182.016	182.016	237.312	237.312	292.608	292.608
17	18.432	18.432	73.728	73.728	129.024	129.024	184.320	184.320	239.616	239.616	294.912	294.912
18	18.432	18.432	73.728	73.728	129.024	129.024	184.320	184.320	239.616	239.616	294.912	294.912
19	20.736	20.736	76.032	76.032	131.328	131.328	186.624	186.624	241.920	241.920	297.216	297.216
20	20.736	20.736	76.032	76.032	131.328	131.328	186.624	186.624	241.920	241.920	297.216	297.216
21	23.040	23.040	78.336	78.336	133.632	133.632	188.928	188.928	244.224	244.224	299.520	299.520
22	23.040	23.040	78.336	78.336	133.632	133.632	188.928	188.928	244.224	244.224	299.520	299.520
23	25.344	25.344	80.640	80.640	135.936	135.936	191.232	191.232	246.528	246.528	301.824	301.824
24	25.344	25.344	80.640	80.640	135.936	135.936	191.232	191.232	246.528	246.528	301.824	301.824
25	27.648	27.648	82.944	82.944	138.240	138.240	193.536	193.536	248.832	248.832	304.128	304.128
26	27.648	27.648	82.944	82.944	138.240	138.240	193.536	193.536	248.832	248.832	304.128	304.128
27	29.952	29.952	85.248	85.248	140.544	140.544	195.840	195.840	251.136	251.136	306.432	306.432
28	29.952	29.952	85.248	85.248	140.544	140.544	195.840	195.840	251.136	251.136	306.432	306.432
29	32.256	32.256	87.552	87.552	142.848	142.848	198.144	198.144	253.440	253.440	308.736	308.736
30	32.256	32.256	87.552	87.552	142.848	142.848	198.144	198.144	253.440	253.440	308.736	308.736
31	34.560	34.560	89.856	89.856	145.152	145.152	200.448	200.448	255.744	255.744	311.040	311.040
32	34.560	34.560	89.856	89.856	145.152	145.152	200.448	200.448	255.744	255.744	311.040	311.040
	0	1	2	3	4	5	6	7	8	9	10	11

Recharge Time	18.430	Cycle Time	55.296
Firing Time Between Laser Fires	2.304		

9.5 Precision Azimuth Calculation

The azimuth (α) reported by the sensor in each data block represents the center line of the firing pattern at the moment the first pair of lasers in a firing sequence fires. But, because the lasers are offset from the center line by fixed amounts, an azimuthal offset (δ) must be applied to determine the precise azimuth of each data point. The vertical angle and azimuth offset of each laser are given by its location in the packet as shown in *Table 9-2 on page 62*. Both *Table 9-6 below* and *Figure F-3 on page 127* illustrate where a given laser is aimed when it fires. Laser ids are listed in four columns according to where they are situated in the VLP-32C laser firing map.

Table 9-6 VLP-32C Azimuth Offsets by Elevation

	Azimuth Offsets (δ)			
Elevation Angle (°)	Col 1 +4.2°	Col 2 +1.4°	Col 3 -1.4°	Col 4 -4.2°
15.000		29		
10.333			30	
7.000		25		

	Azimuth Offsets (δ)			
Elevation Angle (°)	Col 1 +4.2°	Col 2 +1.4°	Col 3 -1.4°	Col 4 -4.2°
4.667			26	
3.333		21		
2.333			22	
1.667	17			
1.333		13		
1.000			18	
0.667				14
0.333	9			
0.000		5		
-0.333			10	
-0.667				6
-1.000	1			
-1.333		31		
-1.667			2	
-2.000				28
-2.333	27			
-2.667		23		
-3.000			24	
-3.333				20
-3.667	19			
-4.000		15		
-4.667			16	
-5.333				12
-6.148		11		
-7.254			8	
-8.843		7		
-11.310			4	
-15.639		3		
-25.000			0	

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.

These precision azimuths may be calculated/interpolated using the timing offsets shown in *Figure 9-7 on page 70* and *Figure 9-8 on the previous page*.

Consider a single data packet with 12 data blocks. The azimuth given at the beginning of each data block represents the azimuth reported at the moment the first pair of lasers in the block fire. If you assume the rotational speed is constant over the 55.296 µs firing sequence for each data block, 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.
- azimuth_offset[K] represents an array of azimuth offsets derived from *Table 9-2 on page 62*.

```
    // First, adjust for an Azimuth rollover from 359.99° to 0°
    If (Azimuth[datablock_n+1] < Azimuth[datablock_n])
    Then
        Azimuth[datablock_n+1] := Azimuth[datablock_n+1] + 360;
    Endif;

    // Determine the Azimuth Gap between data blocks
    AzimuthGap = Azimuth[datablock_n+1] - Azimuth[datablock_n];

    // Perform the interpolation using the timing firing
    // Note that since pairs of lasers fire at once, each pair shares the same baseline azimuth
    // See Table 9-5
    K = 0;
    While (K < 31)
        // Interpolate
        Precision_Azimuth[K] := Azimuth[datablock_n] + (AzimuthGap * 2.304 µs * K) / 55.296 µs);
        Precision_Azimuth[K+1] := Precision_Azimuth[K]

        // Apply the azimuth offsets
        Precision_Azimuth[K] := Precision_Azimuth[K] + azimuth_offset[K];
        Precision_Azimuth[K+1] := Precision_Azimuth[K+1] + azimuth_offset[K+1];

        // Adjust for any rollover
        If Precision_Azimuth[K] >= 360
        Then
            Precision_Azimuth[K] := Precision_Azimuth[K] – 360;
        Endif
        If Precision_Azimuth[K+1] >= 360
        Then
            Precision_Azimuth[K+1] := Precision_Azimuth[K+1] – 360;
        Endif

        K = K + 2;
    End While
```

9.6 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.

Chapter 10 • Sensor Communication

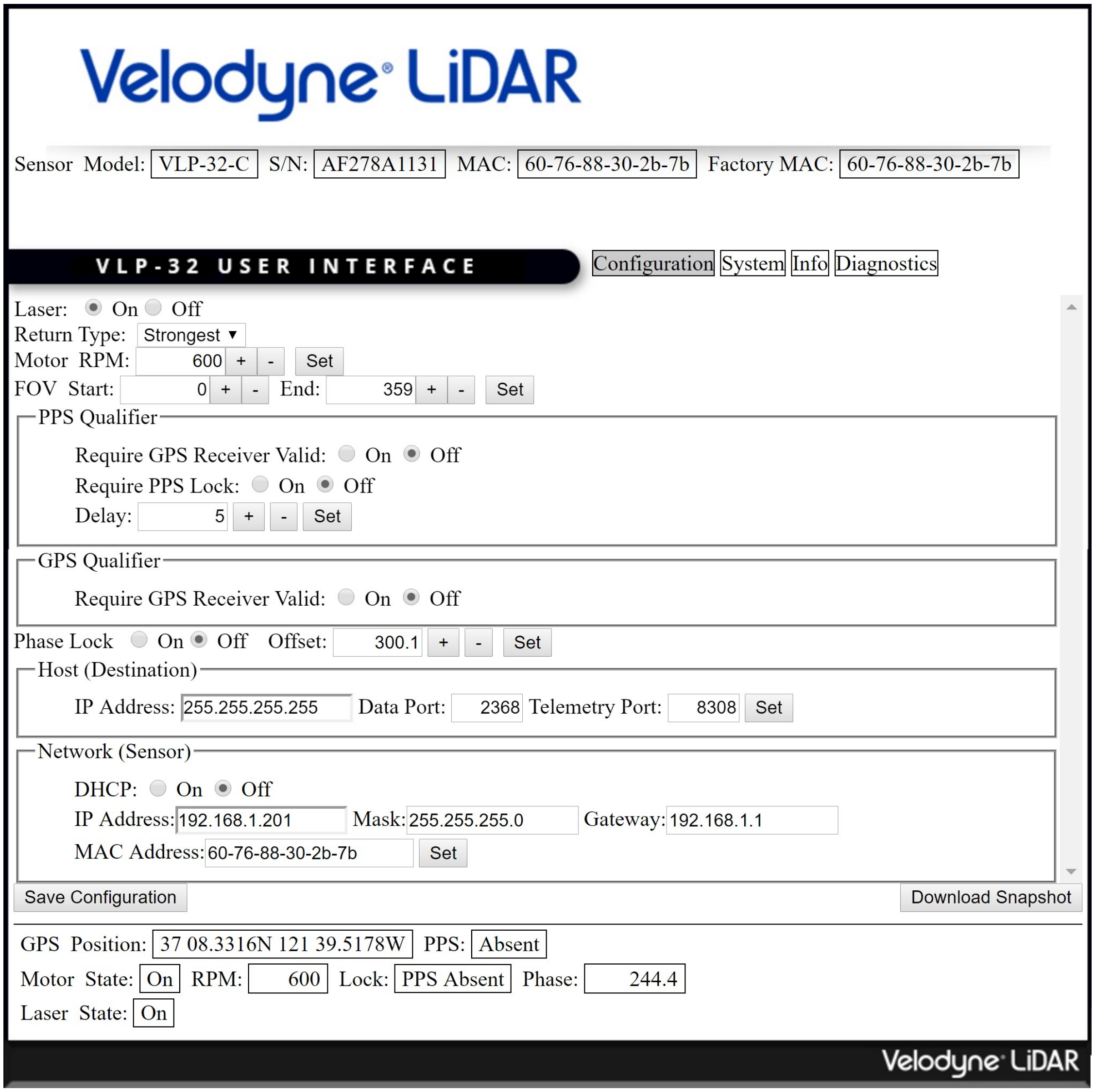
This section provides detailed information about sensor communication methods.

10.1 Web Interface

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

10.1.1 Configuration Screen

Figure 10-1 VLP-32C Configuration Screen



Sensor Model: VLP-32-C S/N: AF278A1131 MAC: 60-76-88-30-2b-7b Factory MAC: 60-76-88-30-2b-7b

Laser: ● On ○ Off
Return Type: Strongest ▾
Motor RPM: 600 + - Set
FOV Start: 0 + - End: 359 + - Set

PPS Qualifier
Require GPS Receiver Valid: ○ On ● Off
Require PPS Lock: ○ On ● Off
Delay: 5 + - Set

GPS Qualifier
Require GPS Receiver Valid: ○ On ● Off

Phase Lock ○ On ● Off Offset: 300.1 + - Set

Host (Destination)
IP Address: 255.255.255.255 Data Port: 2368 Telemetry Port: 8308 Set

Network (Sensor)
DHCP: ○ On ● Off
IP Address: 192.168.1.201 Mask: 255.255.255.0 Gateway: 192.168.1.1
MAC Address: 60-76-88-30-2b-7b Set

GPS Position: 37 08.3316N 121 39.5178W PPS: Absent
Motor State: On RPM: 600 Lock: PPS Absent Phase: 244.4
Laser State: On

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. **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

Function	Description	Default
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 54* 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
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 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
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.	OFF 192.168.1.201 255.255.255.0 192.168.1.1

Function	Description	Default
	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 96* procedure to get it back.	
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 98* for analysis and to obtain troubleshooting advice.	
GPS Position	Displays the last received GPS latitude and longitude.	
PPS	Displays the current state of the PPS signal if provided from a GPS. Absent, Synchronizing, Locked, and Error are the valid states.	
Motor State	Indicates if the sensor's internal motor is spinning: ON/OFF. **RPM** – The sensor's current RPM. **Lock** – Indicates the phase lock feature state: ON/OFF. **Phase** – Indicates the current phase offset. Only relevant when the phase lock feature is active. This value is presented in hundredths of a degree.	
Laser State	Indicates the current laser array firing status: ON/DISABLED.	

See *Time Synchronization on page 129* for additional information on the PPS and GPS Qualifier functions.

10.1.1.1 MAC Address

The ability to change the sensor's MAC Address was introduced with VLP-32C firmware release 4.1.2.0.

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

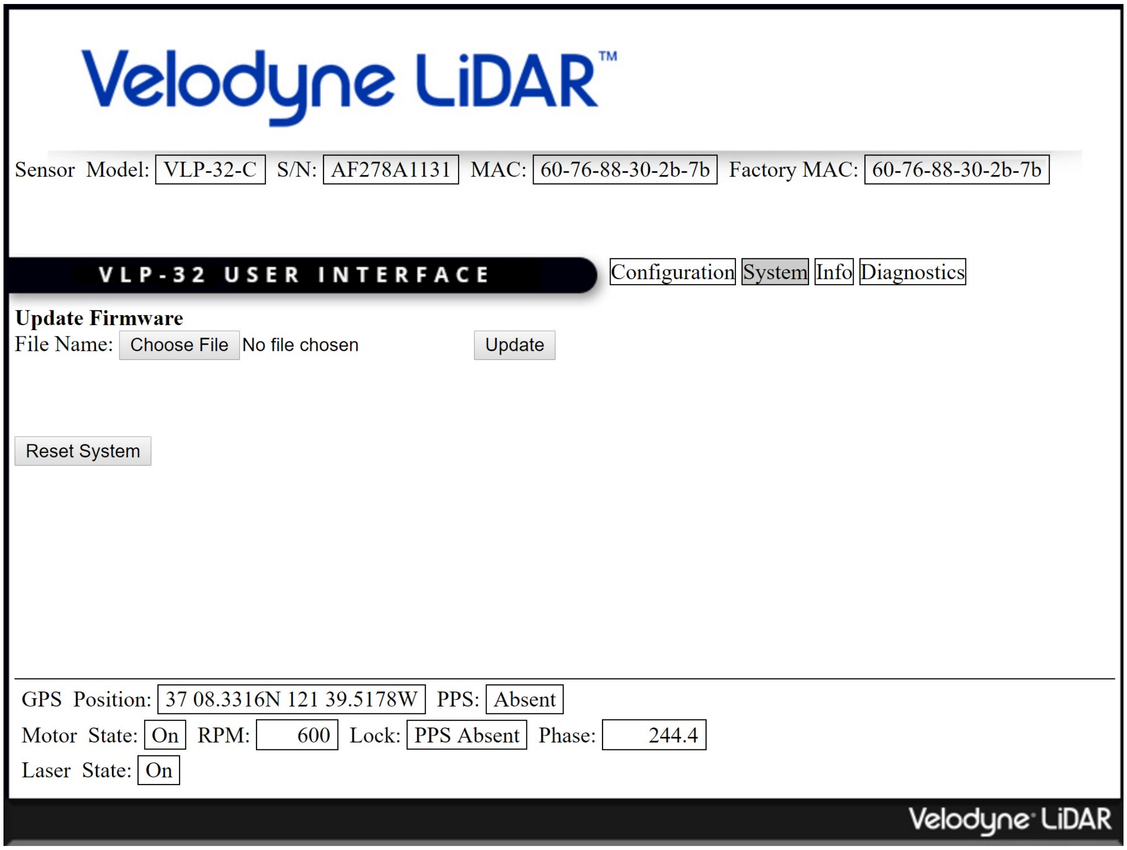
If you have changed the sensor's MAC Address to something other than the Factory MAC Address and wish to reset it back, follow the procedure below.

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 VLP-32C System Screen



Sensor Model: VLP-32-C S/N: AF278A1131 MAC: 60-76-88-30-2b-7b Factory MAC: 60-76-88-30-2b-7b

GPS Position: 37 08.3316N 121 39.5178W PPS: Absent

Motor State: On RPM: 600 Lock: PPS Absent Phase: 244.4

Laser State: On

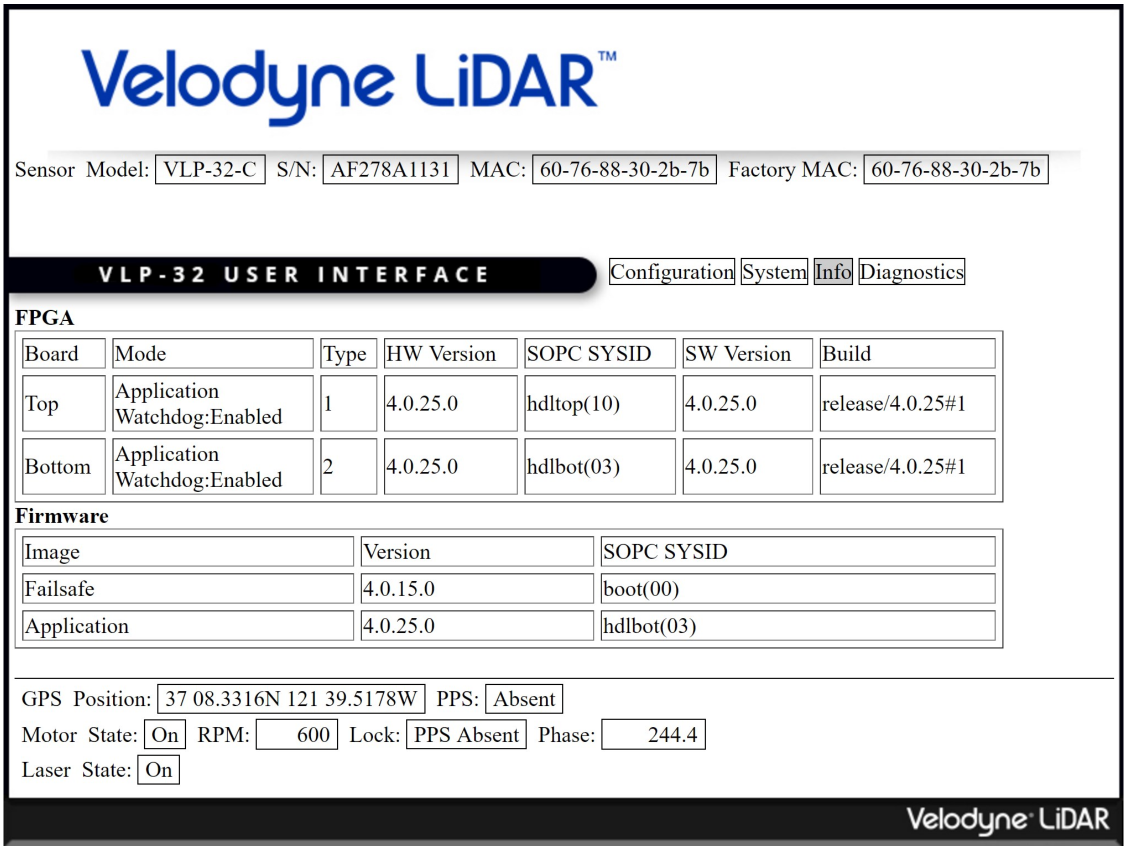
Table 10-2 System Screen Functionality and Features

Function	Description
Update Firmware	**Choose File** – Enables user to select a new firmware image file. **Update** – Initiates the firmware update process.

Function	Description
Reset System	Used to reset the system after a firmware update.
GPS Position	See description in *Table 10-1 on page 76*.
PPS	See description in *Table 10-1 on page 76*.
Motor State	See description in *Table 10-1 on page 76*.
Laser State	See description in *Table 10-1 on page 76*.

10.1.3 Info Screen

Figure 10-3 VLP-32C Info Screen



Sensor Model: VLP-32-C S/N: AF278A1131 MAC: 60-76-88-30-2b-7b Factory MAC: 60-76-88-30-2b-7b

FPGA

Board	Mode	Type	HW Version	SOPC SYSID	SW Version	Build
Top	Application Watchdog:Enabled	1	4.0.25.0	hdltop(10)	4.0.25.0	release/4.0.25#1
Bottom	Application Watchdog:Enabled	2	4.0.25.0	hdlbot(03)	4.0.25.0	release/4.0.25#1

Firmware

Image	Version	SOPC SYSID
Failsafe	4.0.15.0	boot(00)
Application	4.0.25.0	hdlbot(03)

GPS Position: 37 08.3316N 121 39.5178W PPS: Absent
Motor State: On RPM: 600 Lock: PPS Absent Phase: 244.4
Laser State: On

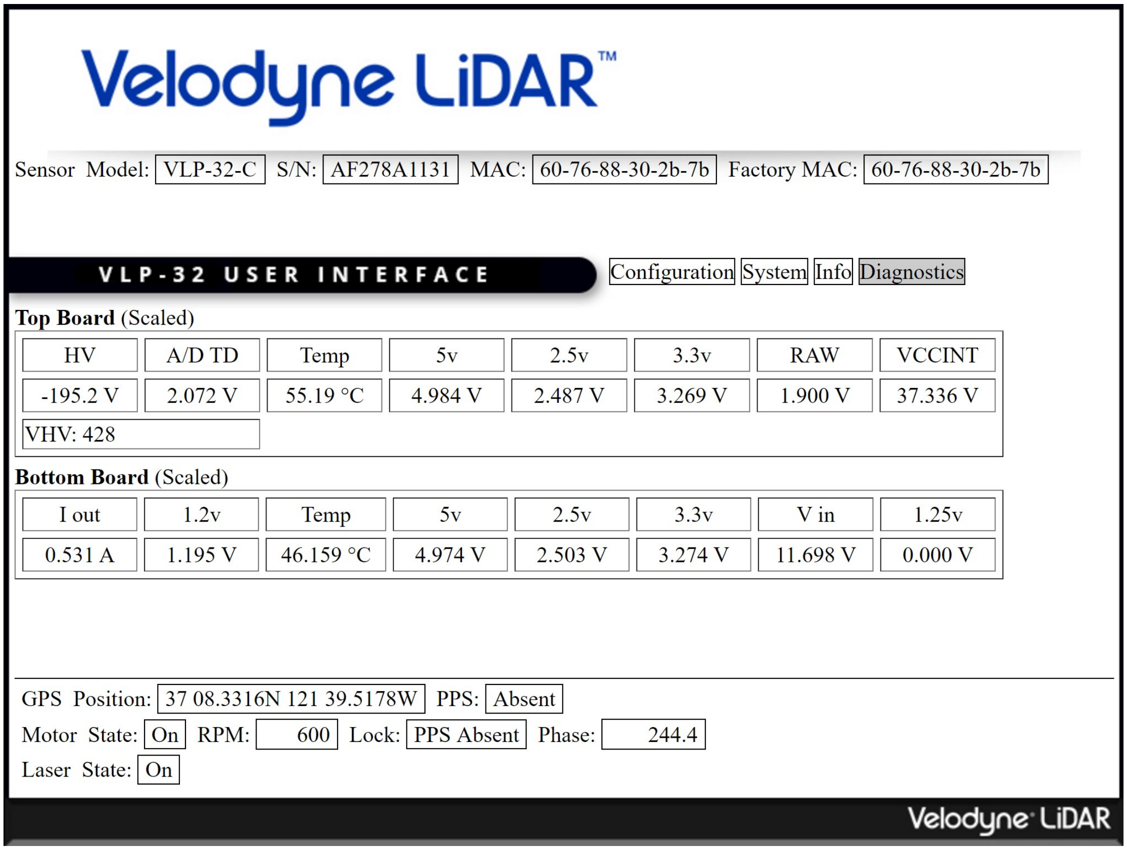
Table 10-3 Info Screen Functionality and Features

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.

Function	Description
	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. **Application hdlbot(03)** – Velodyne Use Only.
GPS Position	See description in *Table 10-1 on page 76*.
PPS	See description in *Table 10-1 on page 76*.
Motor State	See description in *Table 10-1 on page 76*.
Laser State	See description in *Table 10-1 on page 76*.

10.1.4 Diagnostics Screen

Figure 10-4 VLP-32C Diagnostics Screen



Sensor Model: VLP-32-C S/N: AF278A1131 MAC: 60-76-88-30-2b-7b Factory MAC: 60-76-88-30-2b-7b

Top Board (Scaled)

HV	A/D TD	Temp	5v	2.5v	3.3v	RAW	VCCINT
-195.2 V	2.072 V	55.19 °C	4.984 V	2.487 V	3.269 V	1.900 V	37.336 V
VHV: 428							

Bottom Board (Scaled)

I out	1.2v	Temp	5v	2.5v	3.3v	V in	1.25v
0.531 A	1.195 V	46.159 °C	4.974 V	2.503 V	3.274 V	11.698 V	0.000 V

GPS Position: 37 08.3316N 121 39.5178W PPS: Absent

Motor State: On RPM: 600 Lock: PPS Absent Phase: 244.4

Laser State: On

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. **VHV** – Velodyne Use Only.
Bottom Board (Scaled)	**I out** – Sensor input current. **1.2v** – Internal voltage status. **Temp** – Circuit board temperature. **5v** – Internal voltage status. **2.5v** – Internal voltage status. **3.3v** – Internal voltage status. **V in** – Sensor input voltage.

Function	Description
	1.25v – N/A.
GPS Position	See description in *Table 10-1 on page 76*.
PPS	See description in *Table 10-1 on page 76*.
Motor State	See description in *Table 10-1 on page 76*.
Laser State	See description in *Table 10-1 on page 76*.

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 per second to the sensor.

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,
            "lm20_temp":1105,
            "pwr_5v":2075,
            "pwr_2_5v":2047,
            "pwr_3_3v":2693,
            "pwr_raw":1428,
            "pwr_vccint":977
        },
        "bot":{
            "i_out":2086,
            "pwr_1_2v":992,
            "lm20_temp":1231,
            "pwr_5v":2046,
            "pwr_2_5v":2046,
            "pwr_3_3v":2692,
            "pwr_v_in":910,
            "pwr_1_25v":1
        }
    },
}
```

10.2.3.2 Conversion Formulas

Equation 10-1 Standard Voltage Conversion

$$\boxed{\mathit{volts}\ (V) = \frac{\mathit{raw\ value}\ \mathrm{x}\ 5.0}{4096}}$$

Equation 10-2 Standard Current Conversion

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

Equation 10-3 Standard Temperature Conversion

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

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.

Note: Operating range values in this section are subject to change.

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 85*.

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 85*.

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 85*.

Example:

pwr_vccint = **977** *(raw value)*

(977 * 5.0) / 4096 = **1.193 V**

10.2.3.3.9 bot:i_out

This is the sensor's input power line.

Its operating range is 0.3 A to 1.0 A. (It may spike when the motor spins up.)

To convert the raw value, use *Equation 10-2 on page 85*.

Example:

i_out = **2086** *(raw value)*

(2086 * 5.0) / 4096 = 2.546

10.0 * (2.546 - 2.5) = **0.464 A**

10.2.3.3.10 bot:lm20_temp

This field indicates the temperature of the bottom board.

To convert the raw value, use *Equation 10-3 on page 85*.

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**

10.2.3.3.11 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 85*.

Example:

pwr_1_2v = **992** *(raw value)*

(992 * 5.0) / 4096 = **1.211 V**

10.2.3.3.12 bot:pwr_1_25v

This is the bottom board's 1.25 V rail.

To convert the raw value, use *Equation 10-1 on page 85*.

Example:

pwr_1_25v = **1** *(raw value)*

(1 * 5.0) / 4096 = **0.001 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 85*.

Example:

pwr_2_5v = **2046** *(raw value)*

(2046 * 5.0) / 4096 = **2.498 V**

10.2.3.3.14 bot:pwr_3_3v

This is the bottom board's 3.3 V rail.

To convert the raw value, use *Equation 10-1 on page 85*.

Example:

pwr_3_3v = **2692** *(raw value)*

(2692 * 5.0) / 4096 = **3.286 V**

10.2.3.3.15 bot:pwr_5v

This is the bottom board's 5.0 V rail.

To convert the raw value, use *Equation 10-1 on page 85* then scale the result by multiplying it by 2.0.

Example:

pwr_5v = **2046** *(raw value)*

(2046 * 5.0) / 4096 = 2.498

2.498 * 2.0 = **4.995 V**

10.2.3.3.16 bot:pwr_v_in

This is the voltage of the input power.

To convert the raw value, use *Equation 10-1 on page 85* then scale the result by multiplying it by 11.0.

Example:

pwr_v_in = **910** *(raw value)*

(910 * 5.0) / 4096 = 1.111

1.111 * 11.0 = **12.219 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 Network Configuration

Use the curl commands below to alter the network configuration of the sensor.

Once changed, the settings must be saved and the sensor reset for the new configuration to take effect.

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 96* 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'}), buffer)
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 97* 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 97* 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 76*. - 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. - Sensor's destination address. - Port numbers are set to 2368 (data) and 8308 (position/telemetry).

Problem	Resolution
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 44* 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 137*), 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 3.0 A or greater occurs.

Fuse Specification: **3.0 A Mini ATM Blade Fuse (32 V Max, Color: violet)**

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 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. Select 'for Technical Support' if it isn't already selected.
4. Fill out the rest of the form. Required fields are marked with a red asterisk. Optional fields may be left empty.
5. IMPORTANT: In the **Description** field, please describe the problem concisely, including any non-default sensor configuration and setup details as necessary.
6. 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 and include a link 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; anywhere from 20 to 30 MB is likely.)
7. 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 us for repair.

The Return Merchandise Authorization (RMA) form is now online. To issue an RMA Request, click this link: Request RMA, fill out the form as completely as you can, then click Submit. 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.

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

Appendix A • Sensor Specifications

A.1 Sensor Specifications

Sensor specifications (including environmental and regulatory specifications) can be found on the Velodyne Lidar web site at this address: https://www.velodynelidar.com/downloads/.

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 107*.

B.1 Firmware Update Procedure

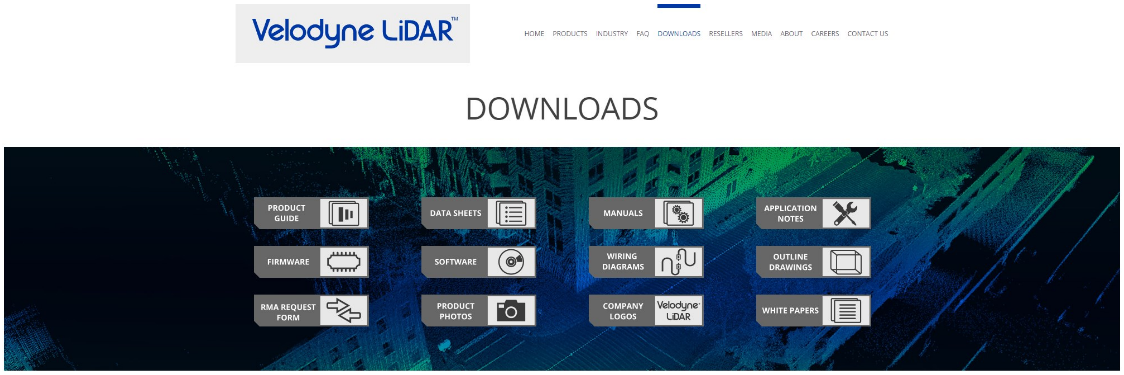
Before starting the procedure, make sure your computer can communicate with the sensor over Ethernet.

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.

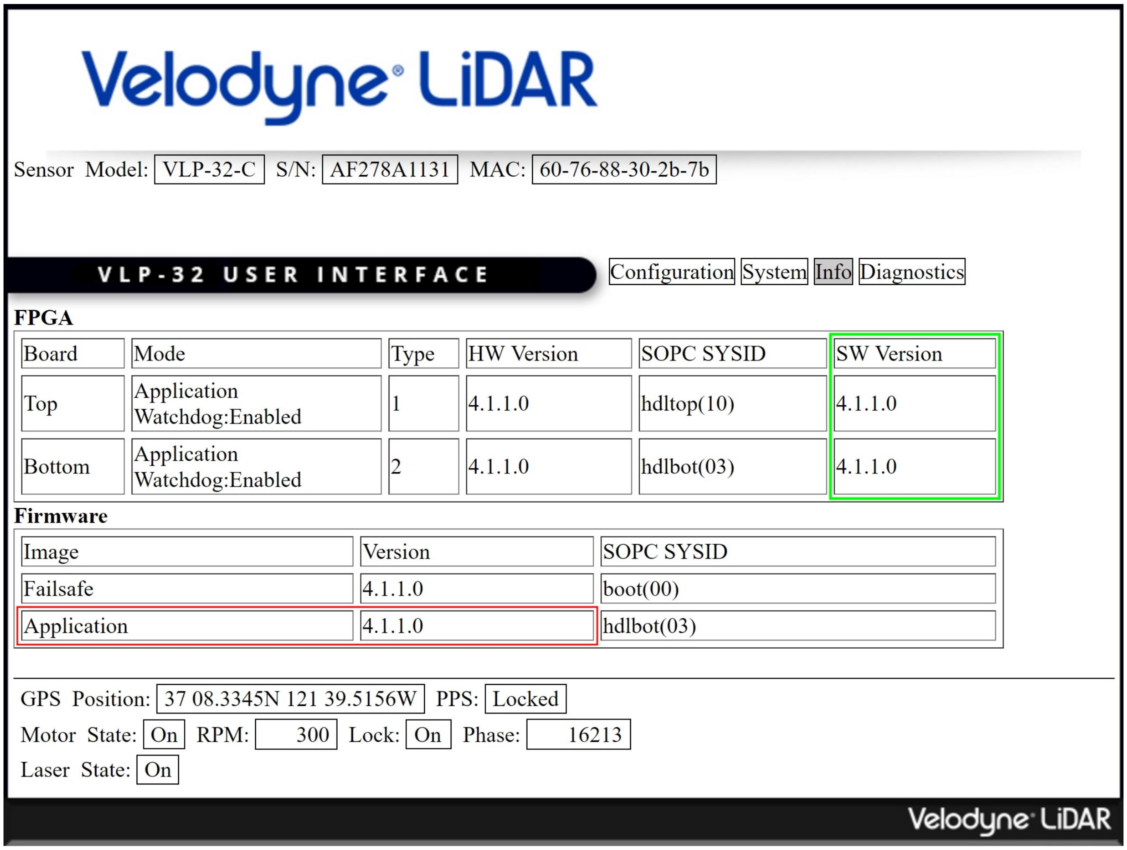
1. Point your browser to the Velodyne Downloads page at https://velodynelidar.com/downloads/.

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



Sensor Model: VLP-32-C S/N: AF278A1131 MAC: 60-76-88-30-2b-7b

FPGA

Board	Mode	Type	HW Version	SOPC SYSID	SW Version
Top	Application Watchdog:Enabled	1	4.1.1.0	hdltop(10)	4.1.1.0
Bottom	Application Watchdog:Enabled	2	4.1.1.0	hdlbot(03)	4.1.1.0

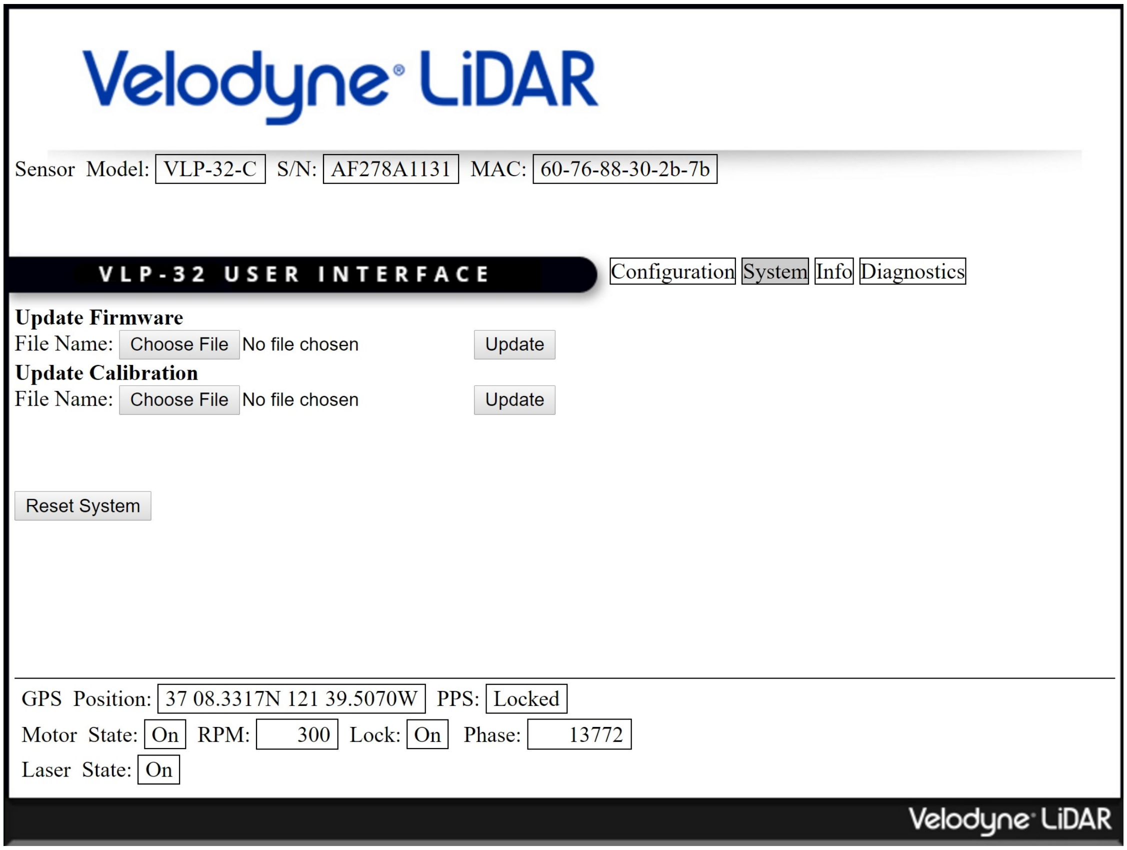
Firmware

Image	Version	SOPC SYSID
Failsafe	4.1.1.0	boot(00)
Application	4.1.1.0	hdlbot(03)

GPS Position: 37 08.3345N 121 39.5156W PPS: Locked
Motor State: On RPM: 300 Lock: On Phase: 16213
Laser State: On

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



Sensor Model: VLP-32-C S/N: AF278A1131 MAC: 60-76-88-30-2b-7b

Update Firmware

File Name: No file chosen

Update Calibration

File Name: No file chosen

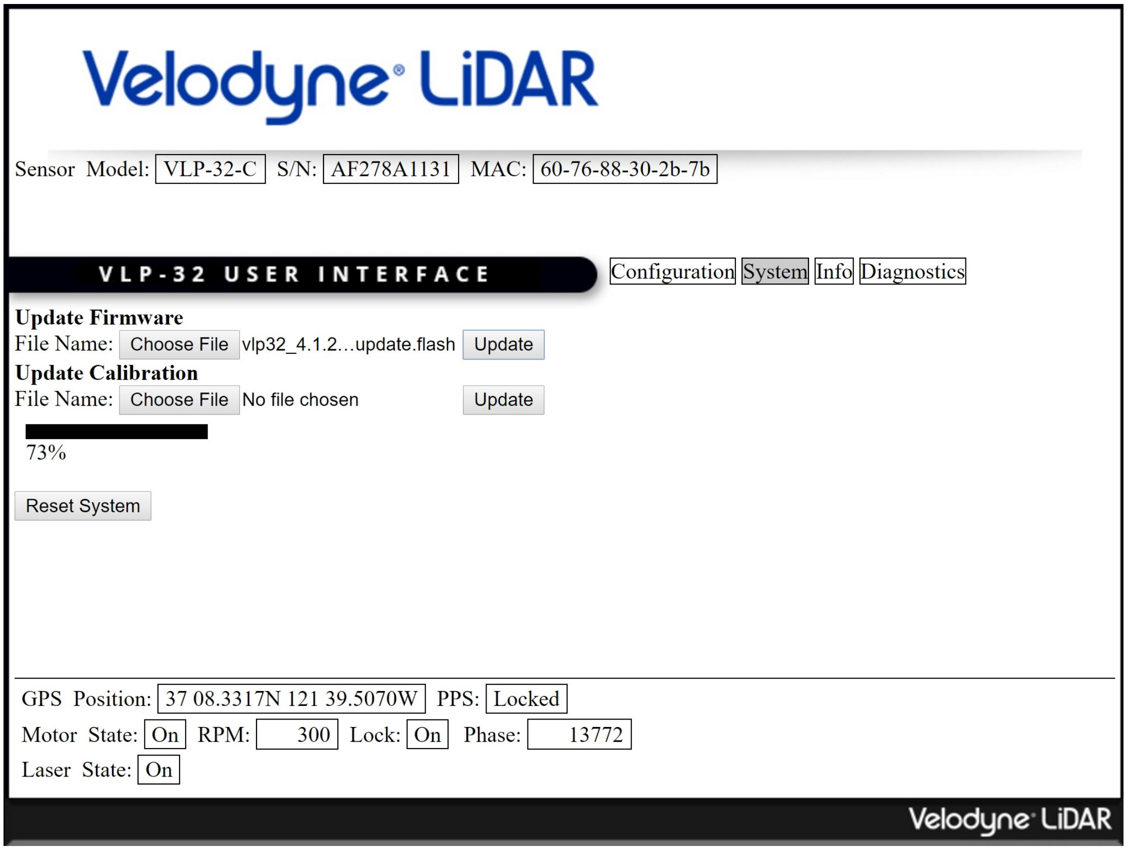
GPS Position: 37 08.3317N 121 39.5070W PPS: Locked

Motor State: On RPM: 300 Lock: On Phase: 13772

Laser State: On

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



Sensor Model: VLP-32-C S/N: AF278A1131 MAC: 60-76-88-30-2b-7b

Update Firmware

File Name: vlp32_4.1.2...update.flash

Update Calibration

File Name: No file chosen

73%

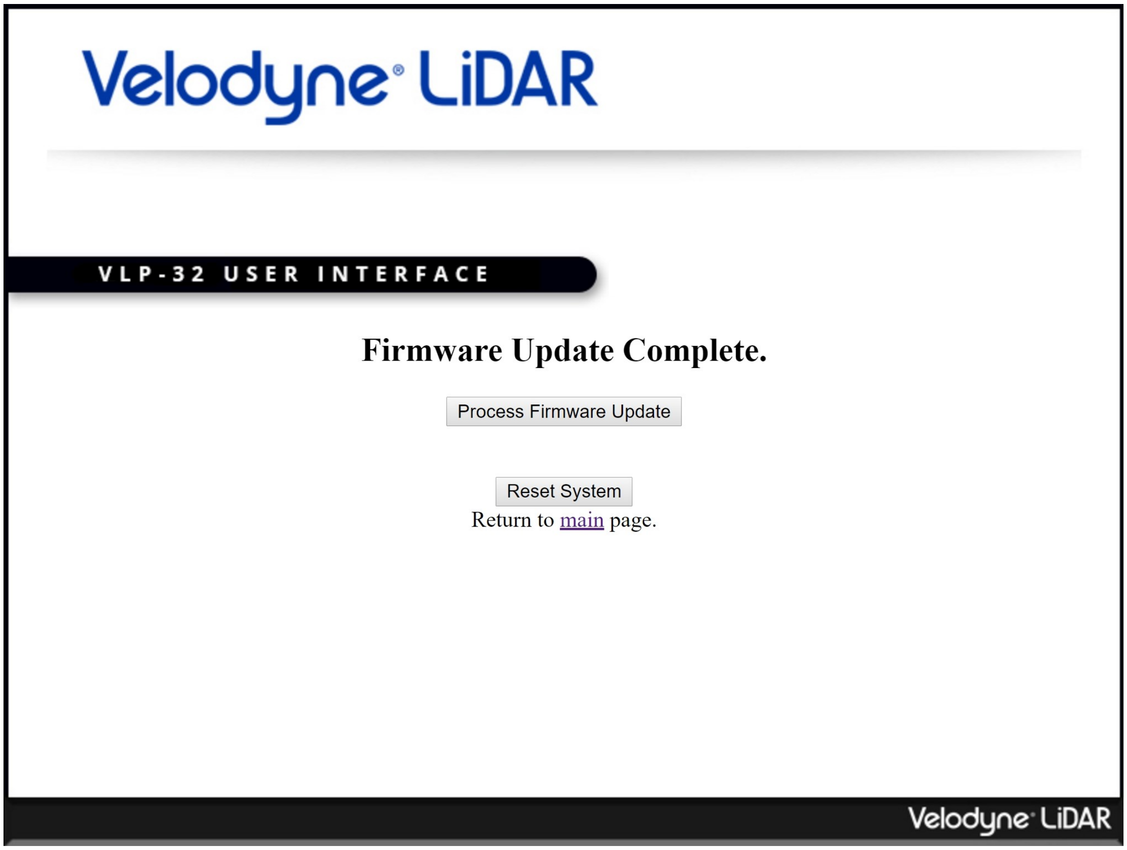
GPS Position: 37 08.3317N 121 39.5070W PPS: Locked

Motor State: On RPM: 300 Lock: On Phase: 13772

Laser State: On

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



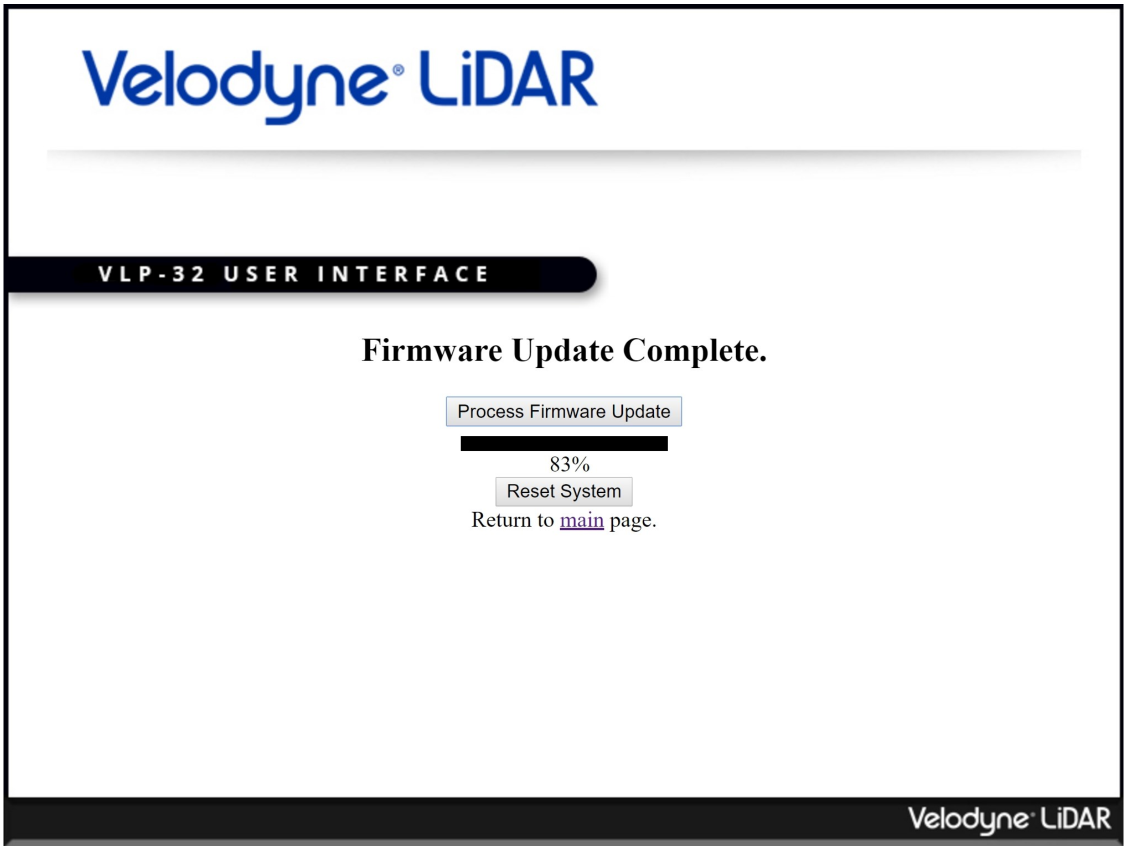
Firmware Update Complete.

Return to main 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 107* procedure.

Figure B-6 Finalize Firmware Update



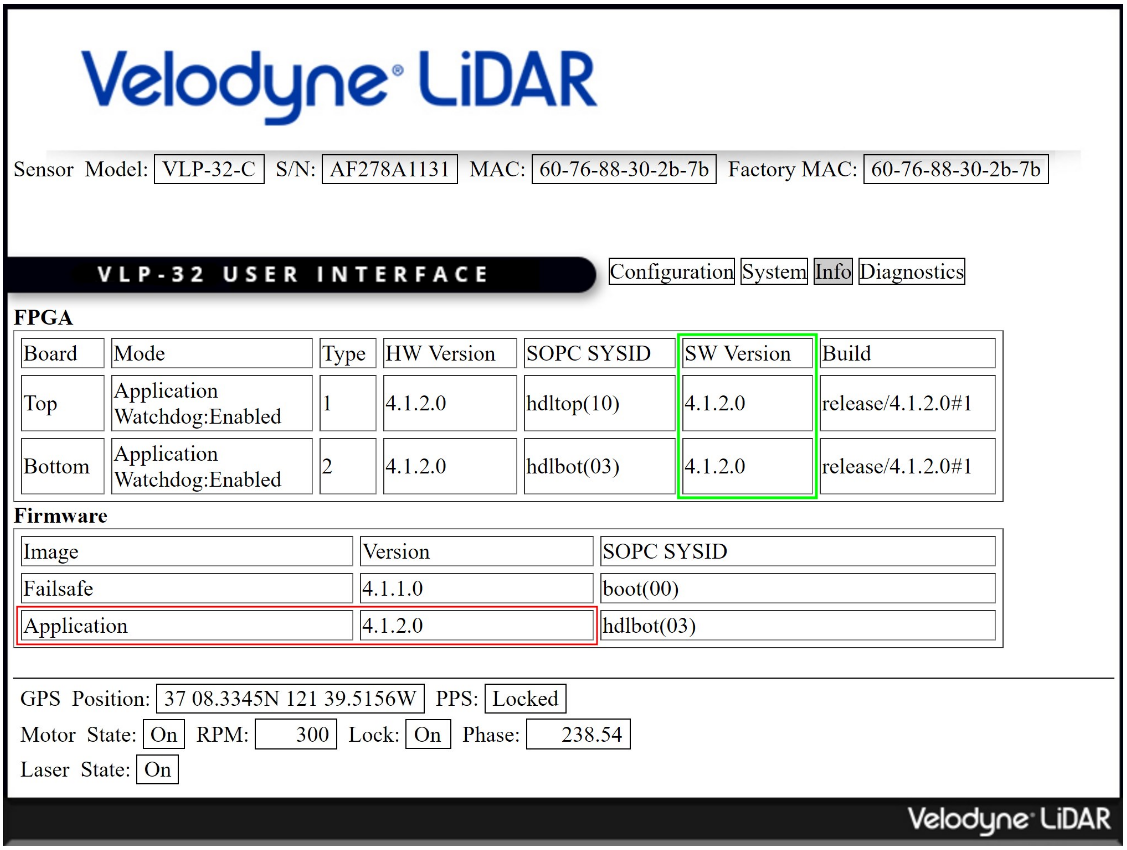
Firmware Update Complete.

83%

Return to main page.

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



Velodyne LiDAR

Sensor Model: VLP-32-C S/N: AF278A1131 MAC: 60-76-88-30-2b-7b Factory MAC: 60-76-88-30-2b-7b

FPGA

Board	Mode	Type	HW Version	SOPC SYSID	SW Version	Build
Top	Application Watchdog:Enabled	1	4.1.2.0	hdltop(10)	4.1.2.0	release/4.1.2.0#1
Bottom	Application Watchdog:Enabled	2	4.1.2.0	hdlbot(03)	4.1.2.0	release/4.1.2.0#1

Firmware

Image	Version	SOPC SYSID
Failsafe	4.1.1.0	boot(00)
Application	4.1.2.0	hdlbot(03)

GPS Position: 37 08.3345N 121 39.5156W PPS: Locked

Motor State: On RPM: 300 Lock: On Phase: 238.54

Laser State: On

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.

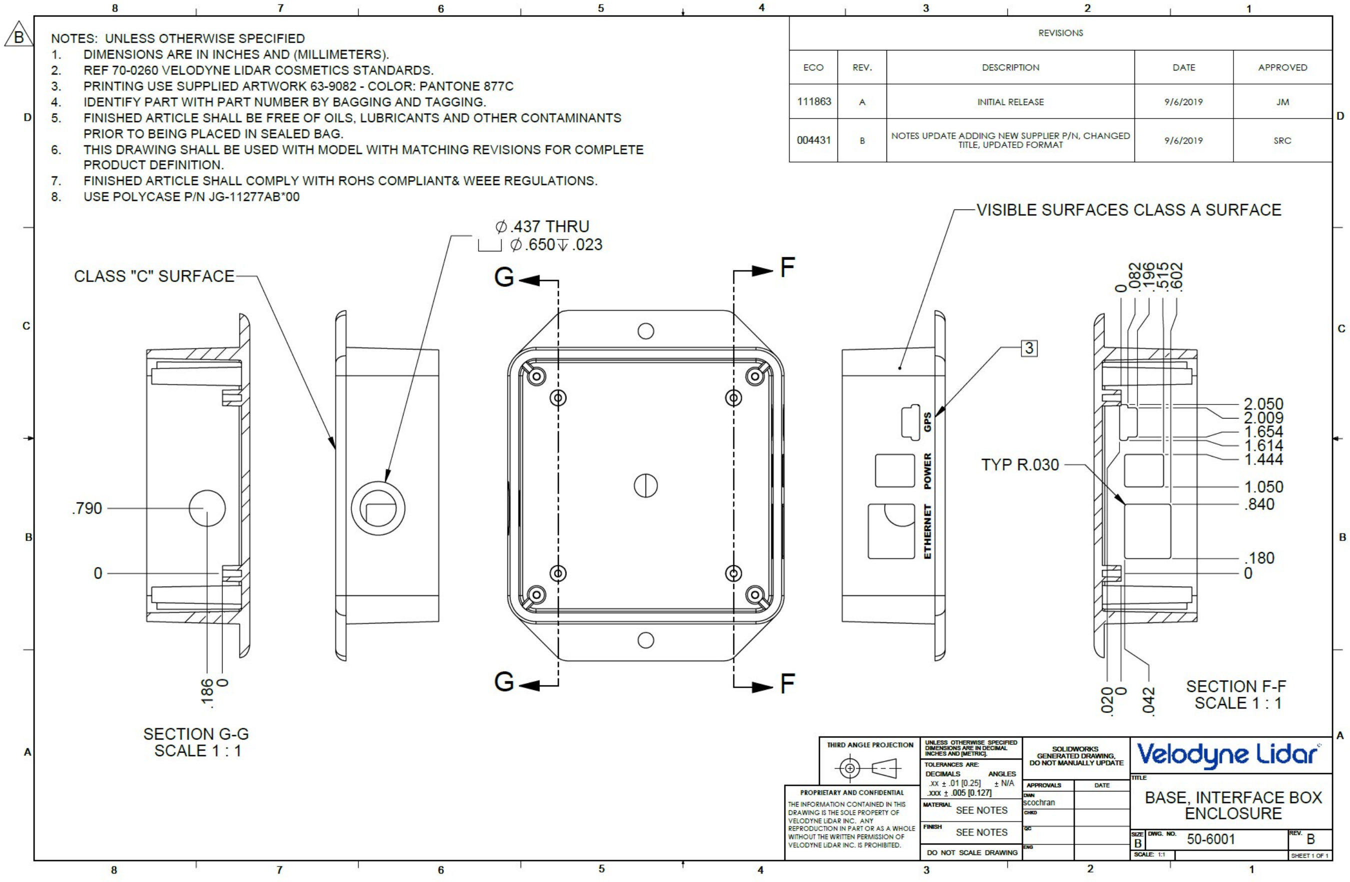
1. Reset the sensor. Either power cycle it (remove power, wait at least 10 seconds, apply power, wait 30 seconds), or click the **Reset System** button if you're on the sensor's **Firmware Update Complete** screen or can get to the sensor's **System** page. The **Configuration** screen will be shown when the sensor finishes rebooting.
2. Go back to the sensor's **System** page (*Figure B-3 on page 102*).
3. In the **Update Firmware** section, just click the **Update** button (*Figure B-4 on page 103*). A progress bar is briefly displayed.
4. When complete, the **Firmware Update Complete** screen is displayed (*Figure B-5 on page 104*).
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 105*).
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 101*) 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 101*). 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 50-6001 Rev B



NOTES: UNLESS OTHERWISE SPECIFIED

1. DIMENSIONS ARE IN INCHES AND (MILLIMETERS).
2. REF 70-0260 VELODYNE LIDAR COSMETICS STANDARDS.
3. PRINTING USE SUPPLIED ARTWORK 63-9082 - COLOR: PANTONE 877C
4. IDENTIFY PART WITH PART NUMBER BY BAGGING AND TAGGING.
5. FINISHED ARTICLE SHALL BE FREE OF OILS, LUBRICANTS AND OTHER CONTAMINANTS PRIOR TO BEING PLACED IN SEALED BAG.
6. THIS DRAWING SHALL BE USED WITH MODEL WITH MATCHING REVISIONS FOR COMPLETE PRODUCT DEFINITION.
7. FINISHED ARTICLE SHALL COMPLY WITH ROHS COMPLIANT& WEEE REGULATIONS.
8. USE POLYCASE P/N JG-11277AB*00

REVISIONS

ECO	REV.	DESCRIPTION	DATE	APPROVED
111863	A	INITIAL RELEASE	9/6/2019	JM
004431	B	NOTES UPDATE ADDING NEW SUPPLIER P/N, CHANGED TITLE, UPDATED FORMAT	9/6/2019	SRC

THIRD ANGLE PROJECTION

PROPRIETARY AND CONFIDENTIAL

THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF VELODYNE LiDAR INC. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF VELODYNE LiDAR INC. IS PROHIBITED.

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN DECIMAL INCHES AND [METRIC].

TOLERANCES ARE:

DECIMALS	ANGLES
.XX ± .01 [0.25]	± N/A
.XXX ± .005 [0.127]	

MATERIAL SEE NOTES

FINISH SEE NOTES

DO NOT SCALE DRAWING

SOLIDWORKS GENERATED DRAWING, DO NOT MANUALLY UPDATE

APPROVALS	DATE
DWN scochran	
CHKD	
QC	
ENG	

Velodyne Lidar

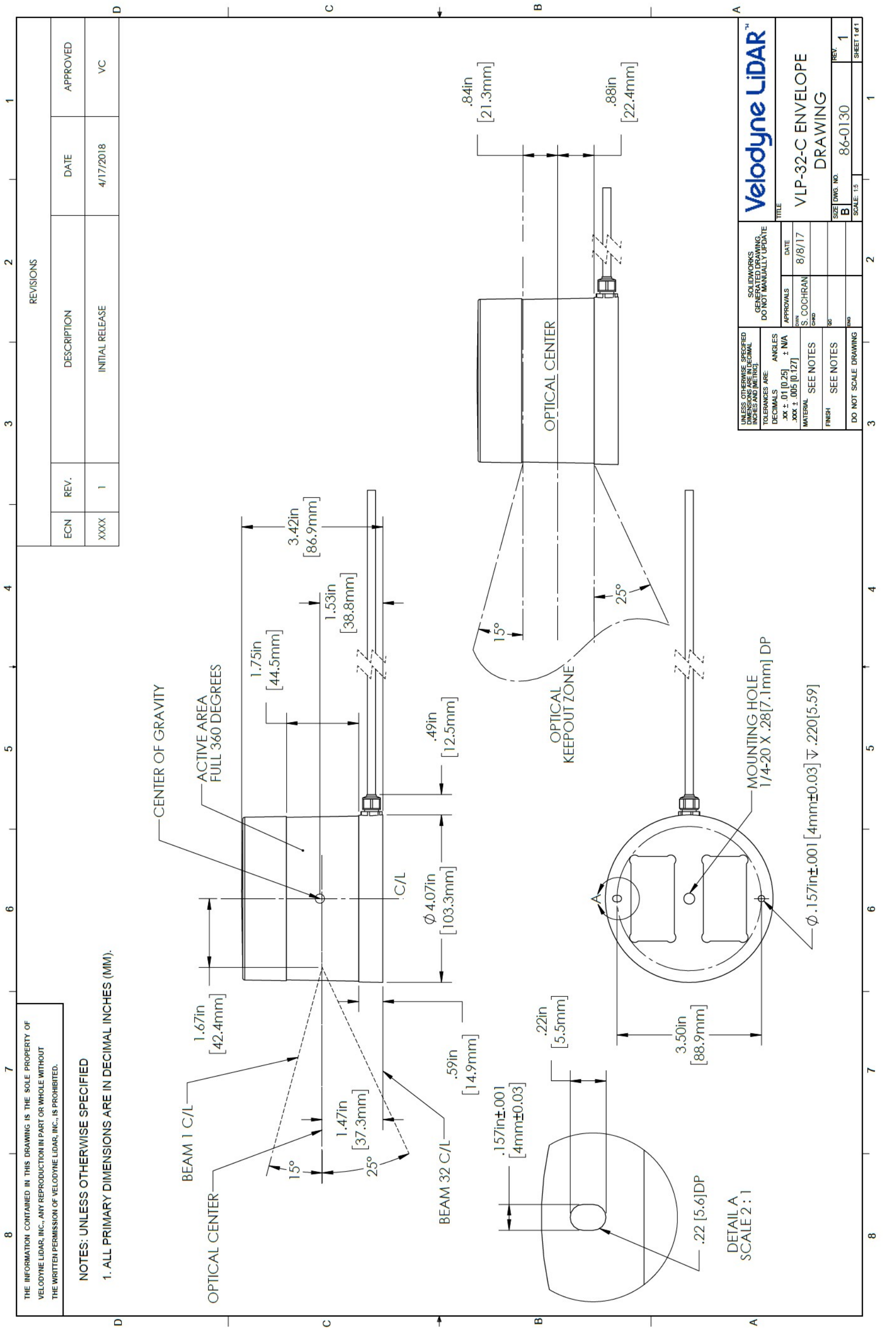
TITLE BASE, INTERFACE BOX ENCLOSURE

SIZE B DWG. NO. 50-6001 REV. B

SCALE: 1:1 SHEET 1 OF 1

C.2 VLP-32C Mechanical Drawing

Figure C-2 VLP-32C Mechanical Drawing 86-0130 Rev 1



Appendix D • Wiring Diagrams

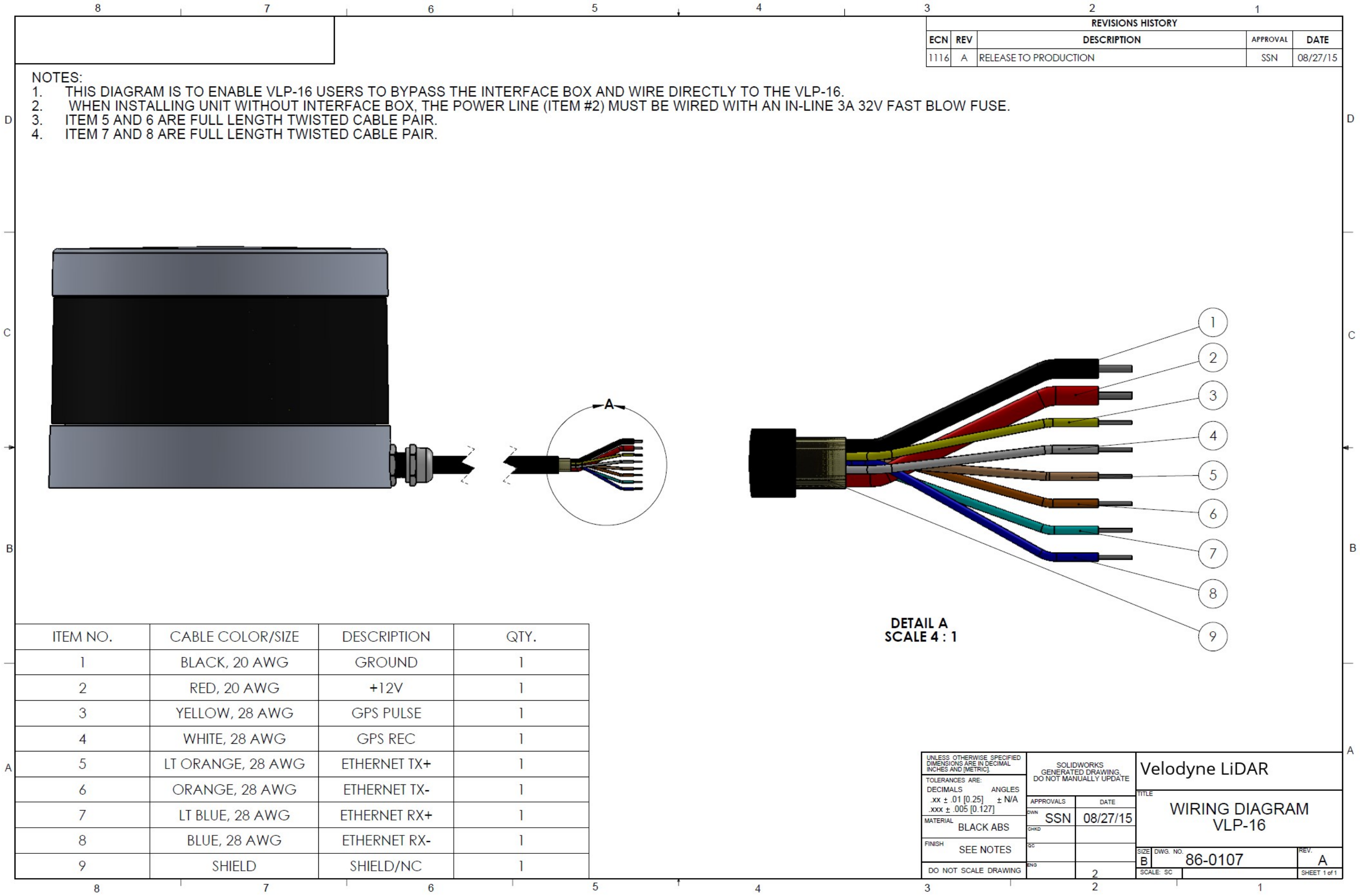
This appendix provides technical wiring and schematic drawings and diagrams. High resolution versions may be accessed on the Velodyne Lidar web site.

Note: The VLP-16 wiring diagram is specified below because the wiring leading to the Interface Box for the VLP-32C is essentially identical to that of the VLP-16.

D.1 Interface Box Wiring Diagram

Figure D-1 Interface Box Wiring Diagram 86-0107A

This diagram also applies to the VLP-32C sensor.



REVISIONS HISTORY				
ECN	REV	DESCRIPTION	APPROVAL	DATE
1116	A	RELEASE TO PRODUCTION	SSN	08/27/15

NOTES:

1. THIS DIAGRAM IS TO ENABLE VLP-16 USERS TO BYPASS THE INTERFACE BOX AND WIRE DIRECTLY TO THE VLP-16.
2. WHEN INSTALLING UNIT WITHOUT INTERFACE BOX, THE POWER LINE (ITEM #2) MUST BE WIRED WITH AN IN-LINE 3A 32V FAST BLOW FUSE.
3. ITEM 5 AND 6 ARE FULL LENGTH TWISTED CABLE PAIR.
4. ITEM 7 AND 8 ARE FULL LENGTH TWISTED CABLE PAIR.

ITEM NO.	CABLE COLOR/SIZE	DESCRIPTION	QTY.
1	BLACK, 20 AWG	GROUND	1
2	RED, 20 AWG	+12V	1
3	YELLOW, 28 AWG	GPS PULSE	1
4	WHITE, 28 AWG	GPS REC	1
5	LT ORANGE, 28 AWG	ETHERNET TX+	1
6	ORANGE, 28 AWG	ETHERNET TX-	1
7	LT BLUE, 28 AWG	ETHERNET RX+	1
8	BLUE, 28 AWG	ETHERNET RX-	1
9	SHIELD	SHIELD/NC	1

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN DECIMAL INCHES AND [METRIC].

TOLERANCES ARE:
DECIMALS .xx ± .01 [0.25] .xxx ± .005 [0.127]
ANGLES ± N/A

MATERIAL BLACK ABS

FINISH SEE NOTES

DO NOT SCALE DRAWING

SOLIDWORKS GENERATED DRAWING, DO NOT MANUALLY UPDATE

APPROVALS		DATE
DWN	SSN	08/27/15
CHKD		
QC		
ENG		2

Velodyne LiDAR

TITLE WIRING DIAGRAM VLP-16

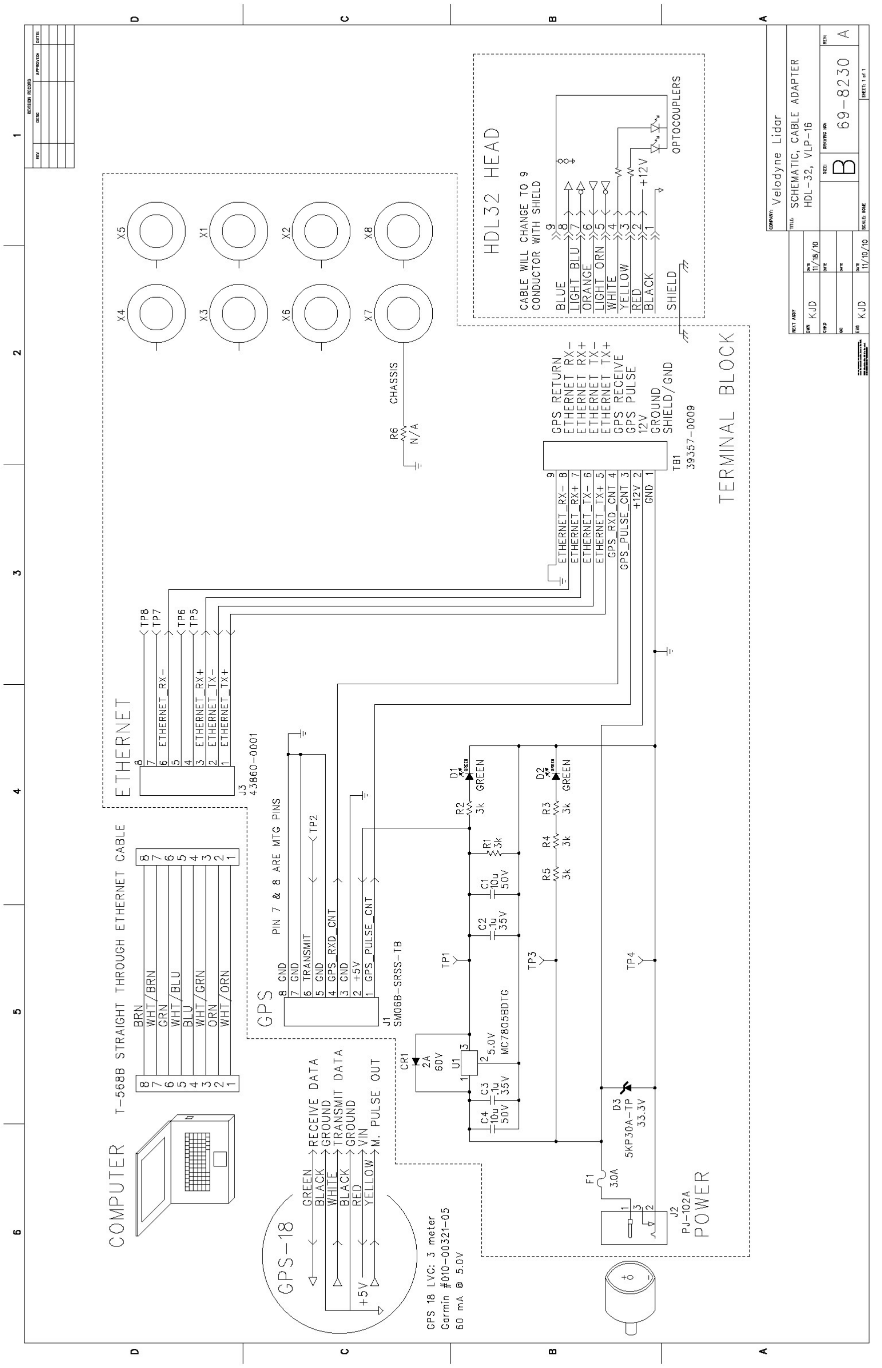
SIZE B DWG. NO. 86-0107 REV. A

SCALE: SC SHEET 1 of 1

D.2 Interface Box Schematic

Figure D-2 Interface Box Schematic 69-8230 Rev A

Though it does not explicitly state so, this diagram also applies to the VLP-32C sensor.



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.

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 as XYZ data 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, timestamp, azimuth, laser ID, etc.
- Tabular laser measurement data inspector

- Export 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 console

Note: VeloView is not intended to generate point cloud files in LAS, XYZ, or PLY formats (see *Converting PCAP Files to Point Cloud Formats on page 73*). See the list of Velodyne system integrators at http://velodynelidar.com/integrators.php 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.

You can 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 http://velodynelidar.com/downloads.html#software.
2. Select VeloView to go the official VeloView download page.
3. Scroll down to the **User Instructions** section.
4. Select the correct binary installer for your operating system; click the link.
5. 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.

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. Add the bin path to your PATH environment variable is as desired.

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

E.2.1 Association with a Different GPU

On underpowered computers, it may improve performance to manually associate VeloView (or whichever visualizer you choose) with a 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. 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 put it on the 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. Launch VeloView. It should be using the graphics card for rendering.

E.3 Visualize Streaming Sensor Data

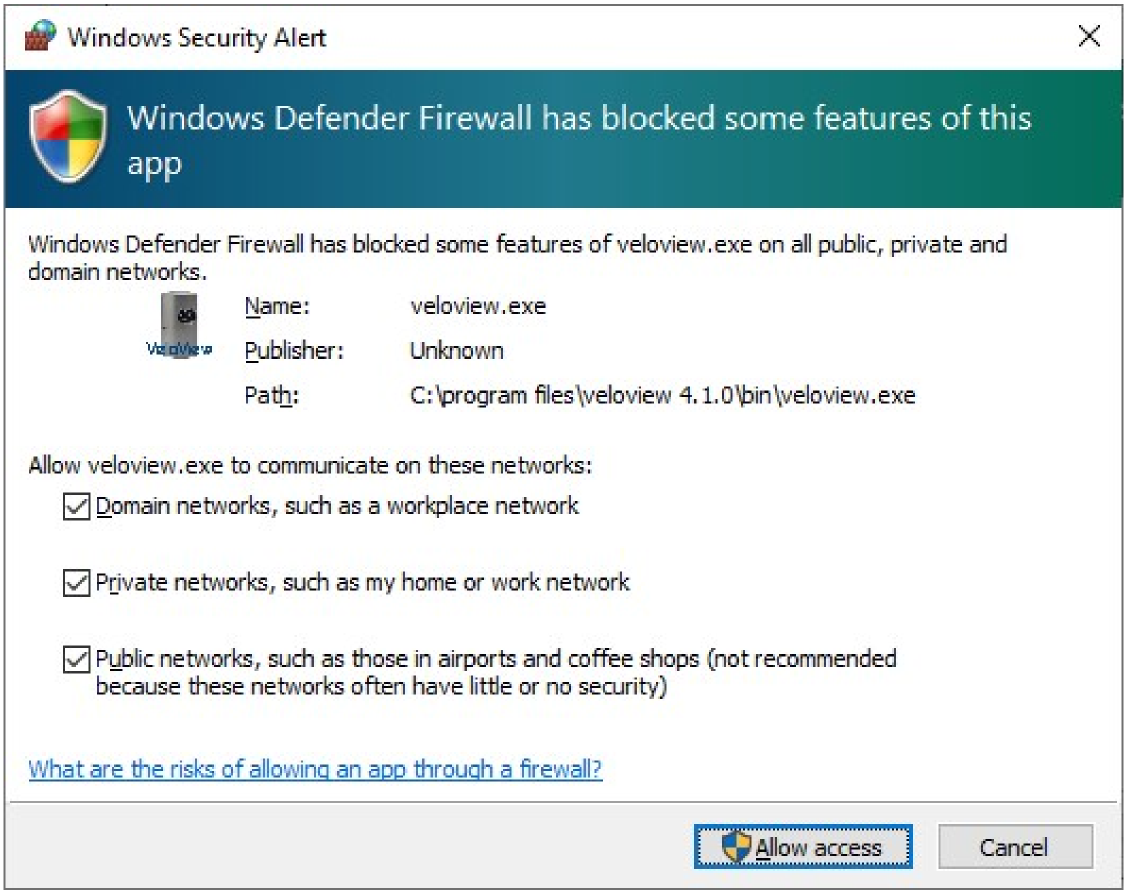
1. Setup access to your sensor over Ethernet as described in *Unboxing & Verification on page 23* or *Installation & Integration on page 31* and power it up.
2. Start the VeloView application.
3. Click on **File** > **Open** and select **Sensor Stream**.

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



Windows Security Alert

Windows Defender Firewall has blocked some features of this app

Windows Defender Firewall has blocked some features of veloview.exe on all public, private and domain networks.

Name: veloview.exe

Publisher: Unknown

Path: C:\program files\veloview 4.1.0\bin\veloview.exe

Allow veloview.exe to communicate on these networks:

☑ Domain networks, such as a workplace network

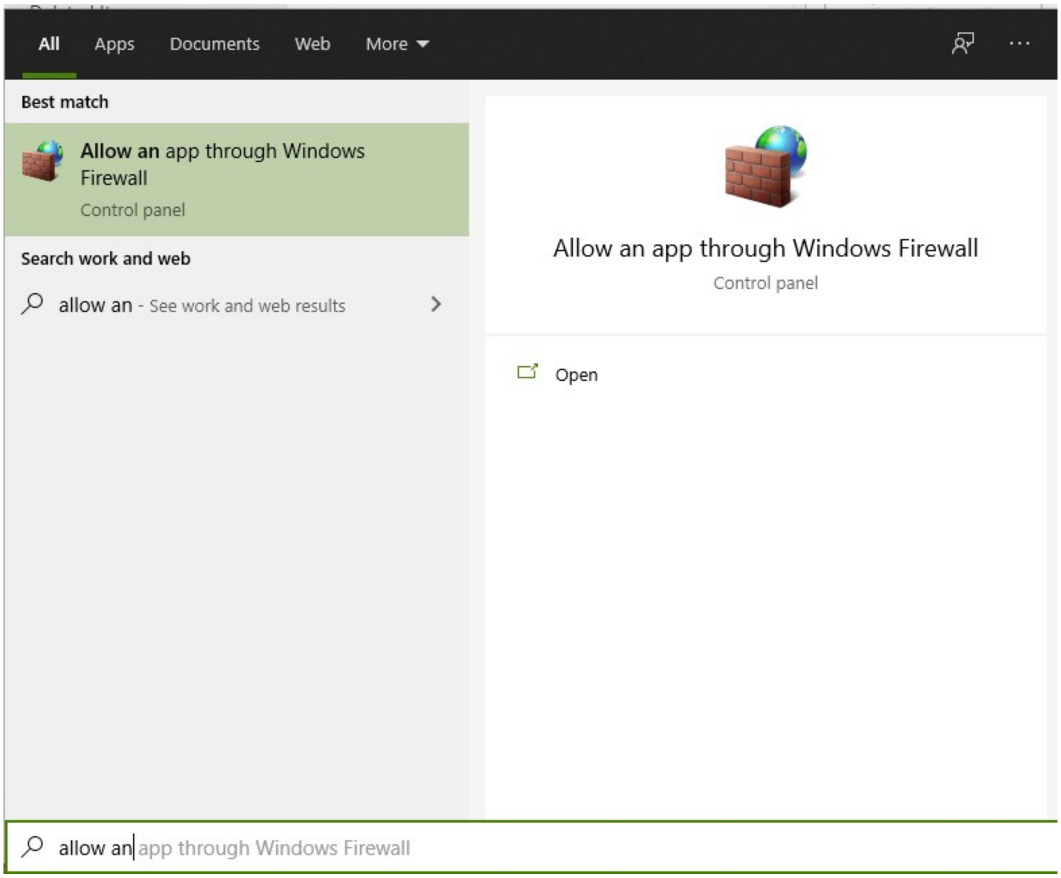
☑ Private networks, such as my home or work network

☑ Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

If you don't do that, the firewall will block communications with the sensor. You'll get no data from the sensor.

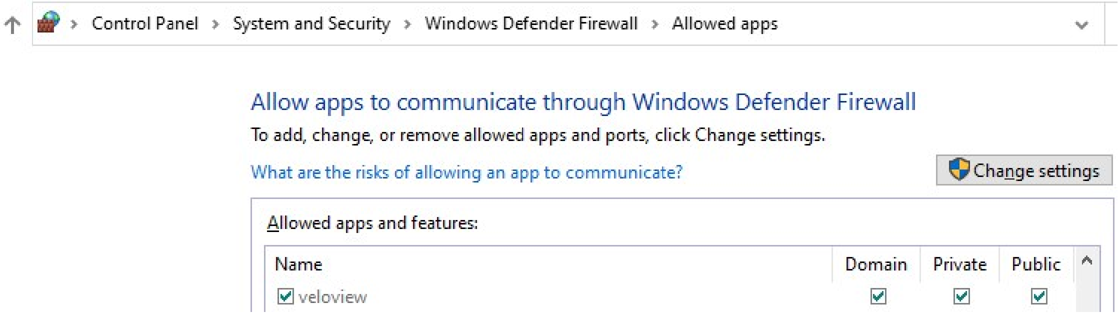
To resolve that situation, select 'Allow an app through Windows Firewall' in the Control Panel (see *Figure E-2 below* below). The 'Allowed apps' dialog will be displayed. Click 'Change settings' to enable changes. Next, scroll down to veloview and click on the entry. Make sure it has a checkmark in 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*.

Figure E-2 Allow an App Through Windows Firewall


All
Apps
Documents
Web
More
Best match
Allow an app through Windows Firewall
Control panel
Search work and web
allow an - See work and web results
Allow an app through Windows Firewall
Control panel
Open

allow an app through Windows Firewall

Figure E-3 Enable VeloView to Communicate Through Windows Firewall


Control Panel › System and Security › Windows Defender Firewall › Allowed apps
Change settings
Allowed apps and features:

Allow apps to communicate through Windows Defender Firewall

To add, change, or remove allowed apps and ports, click Change settings.

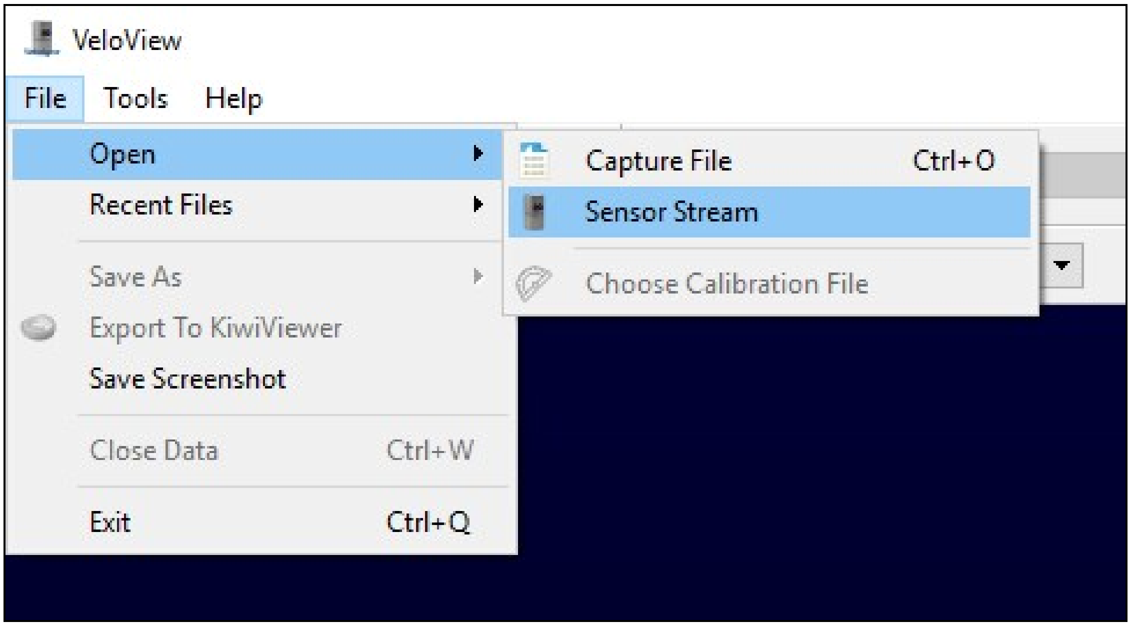
What are the risks of allowing an app to communicate?

Name	Domain	Private	Public
☑ veloview	☑	☑	☑

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

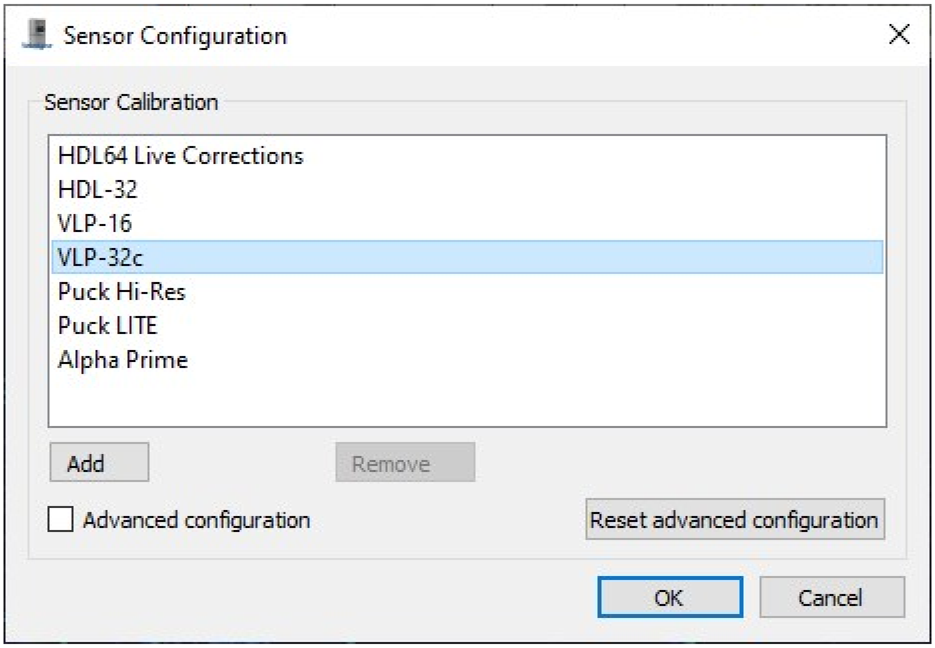
E.3.2 Select the XML File

Figure E-4 Open Sensor Stream



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

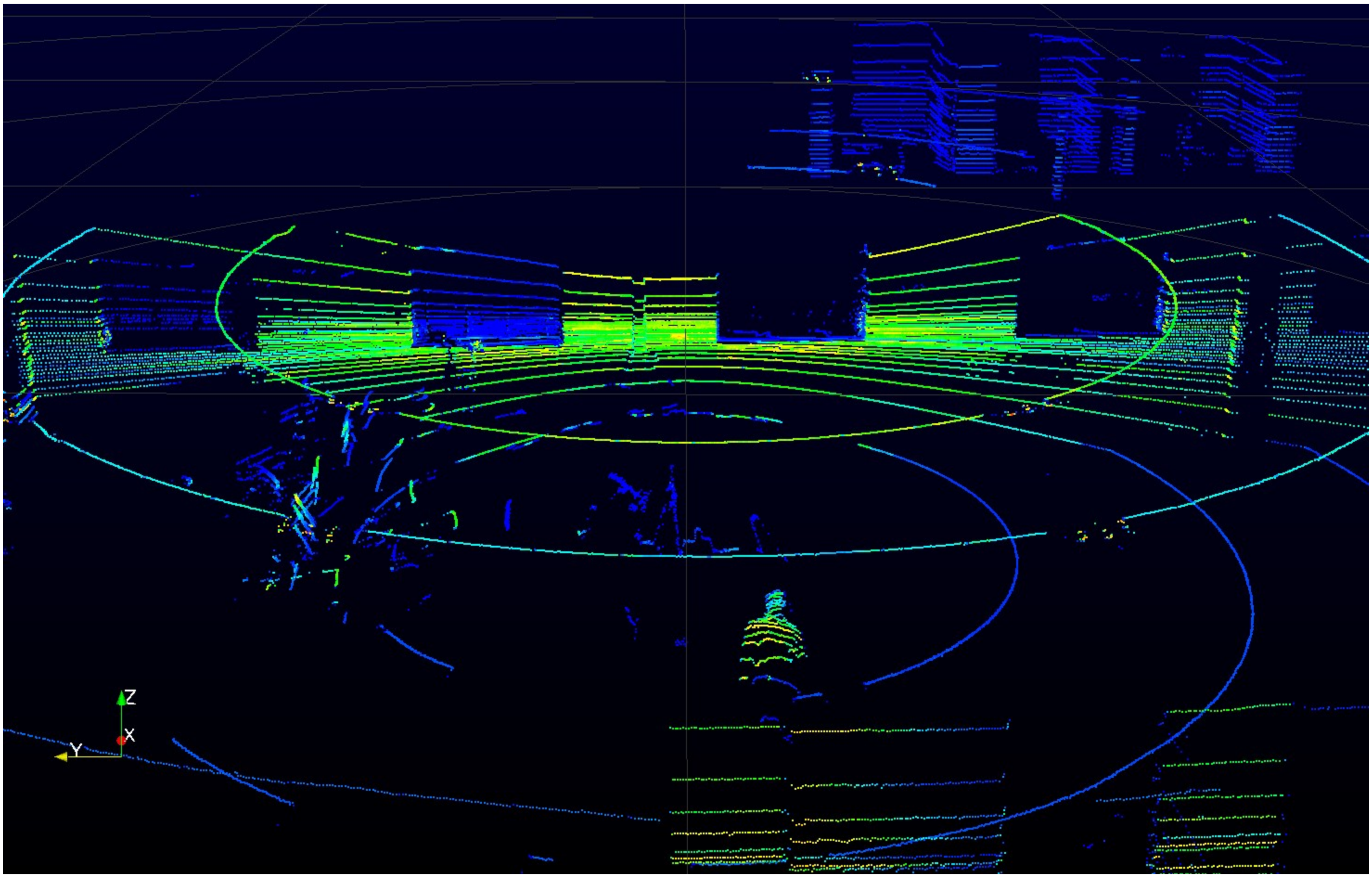
Figure E-5 Select Sensor Calibration (XML) File



E.3.3 Live Stream

5. VeloView begins displaying the sensor data stream.

Figure E-6 VeloView Sensor Stream Display



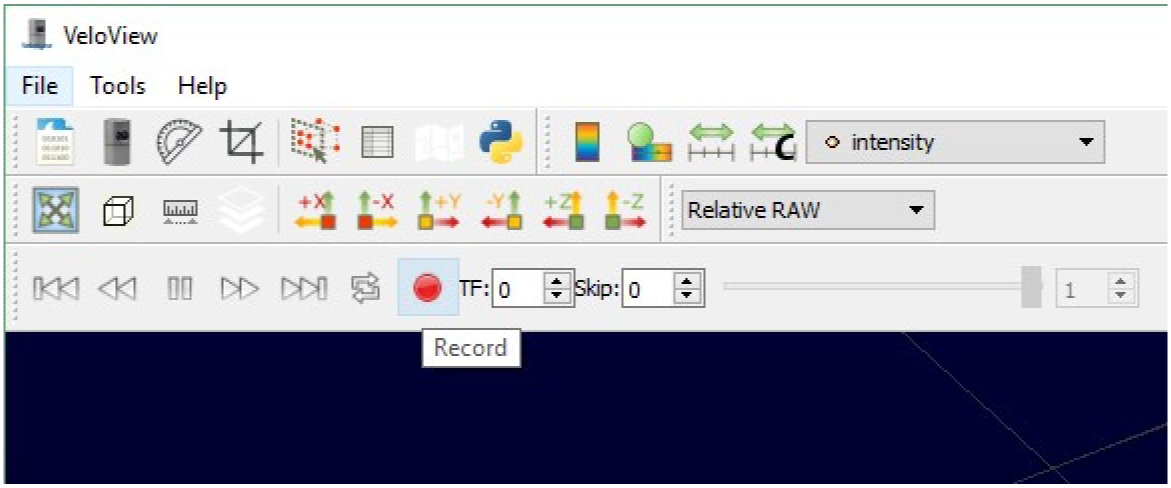
You can move the virtual camera around with either mouse button, the mouse wheel, or with touch controls on a trackpad.

The stream can be paused by pressing the *Play/Pause button on page 121*. Press it again to resume streaming.

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 2019-08-07-08-30-50_ Velodyne-VLP-32C-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.

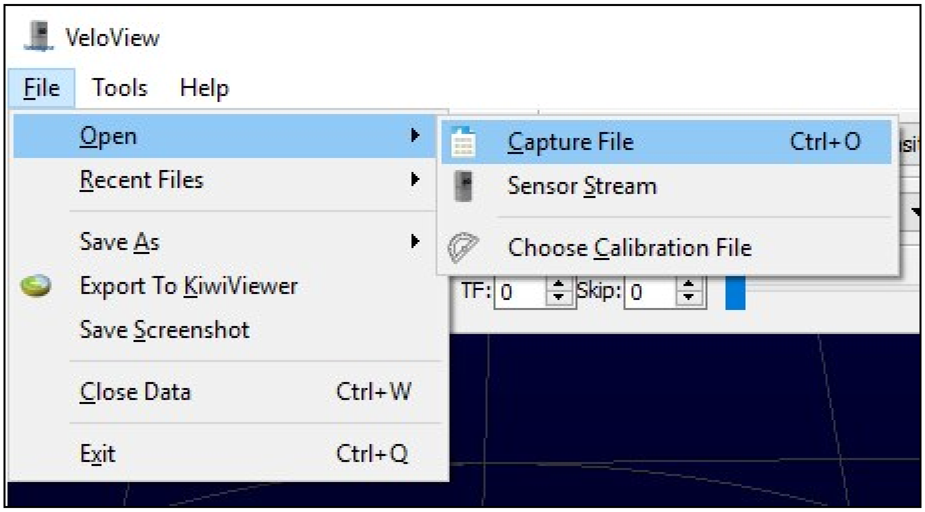
Note: Velodyne Lidar sensors generate a lot of data. (See *Throughput Calculations on page 53* for data rates.) The pcap file can become quite large if the recording duration is lengthy. Also, it is best to record to a fast, local HDD or SSD, not to a slow subsystem such as a USB storage device or network 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**.

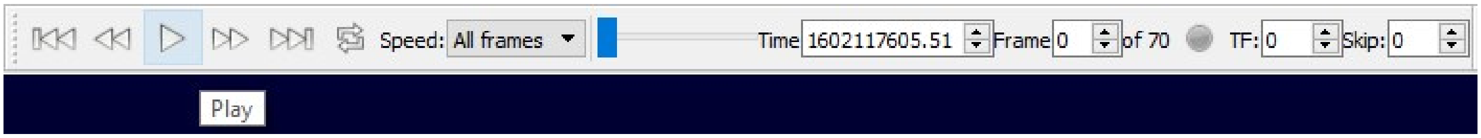
Figure E-8 VeloView Open Capture File



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

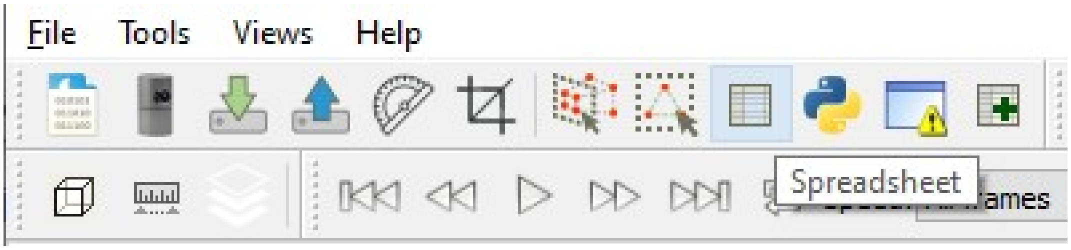


4. Press **Play** to replay/pause the data stream. Use the **Scrub** slider tool (it looks like an old-fashioned volume slider) to move back and forth through the data frames. Both controls are in the same toolbar as the Record button.

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.

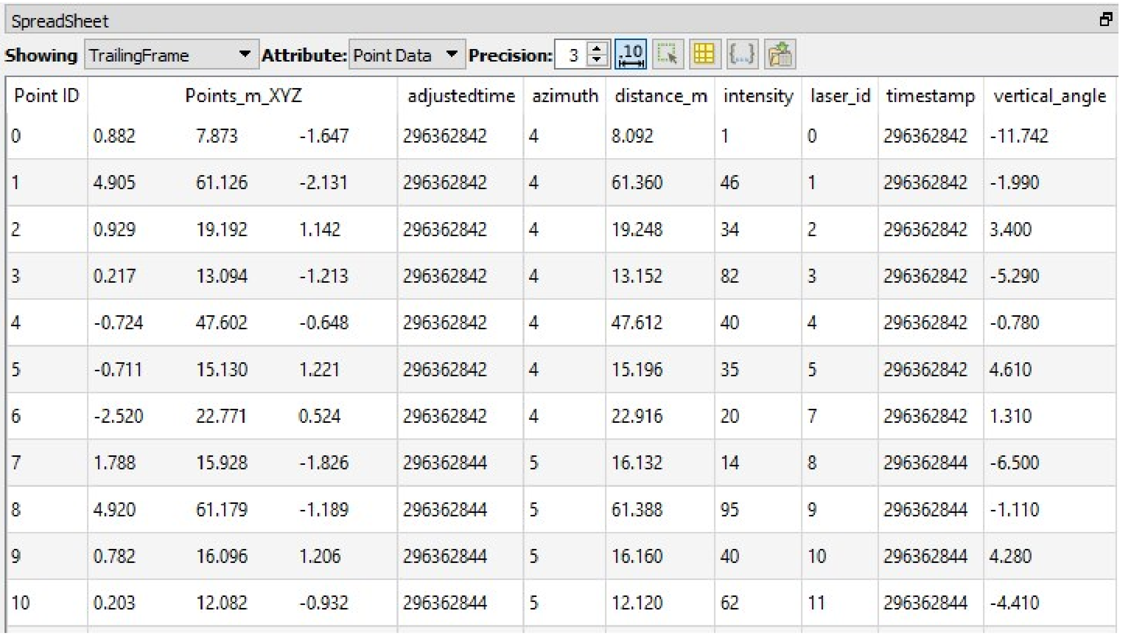
Figure E-10 VeloView Spreadsheet Tool



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



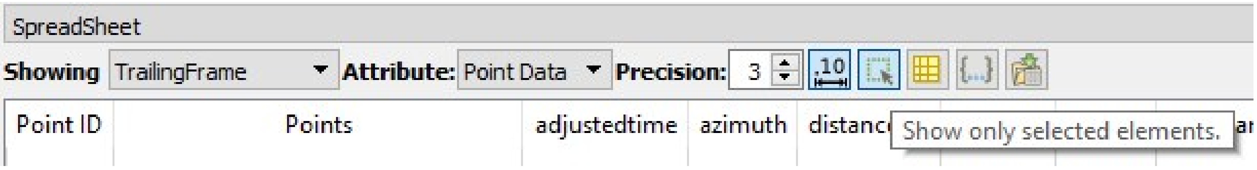
SpreadSheet

Showing TrailingFrame **Attribute:** Point Data **Precision:** 3

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



SpreadSheet

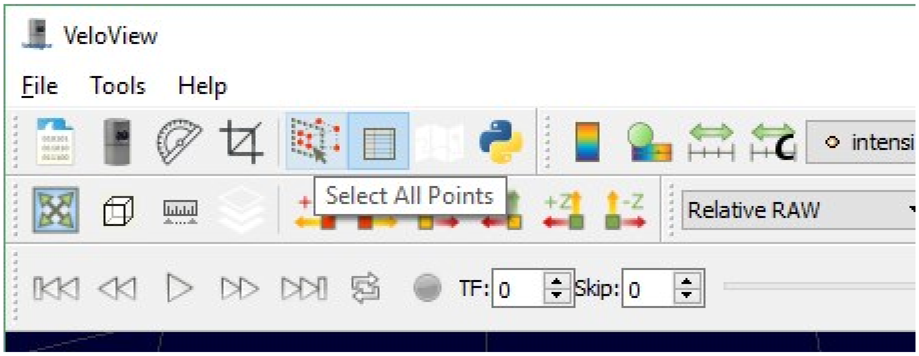
Showing TrailingFrame **Attribute:** Point Data **Precision:** 3

Point ID	Points	adjustedtime	azimuth	distanc

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.

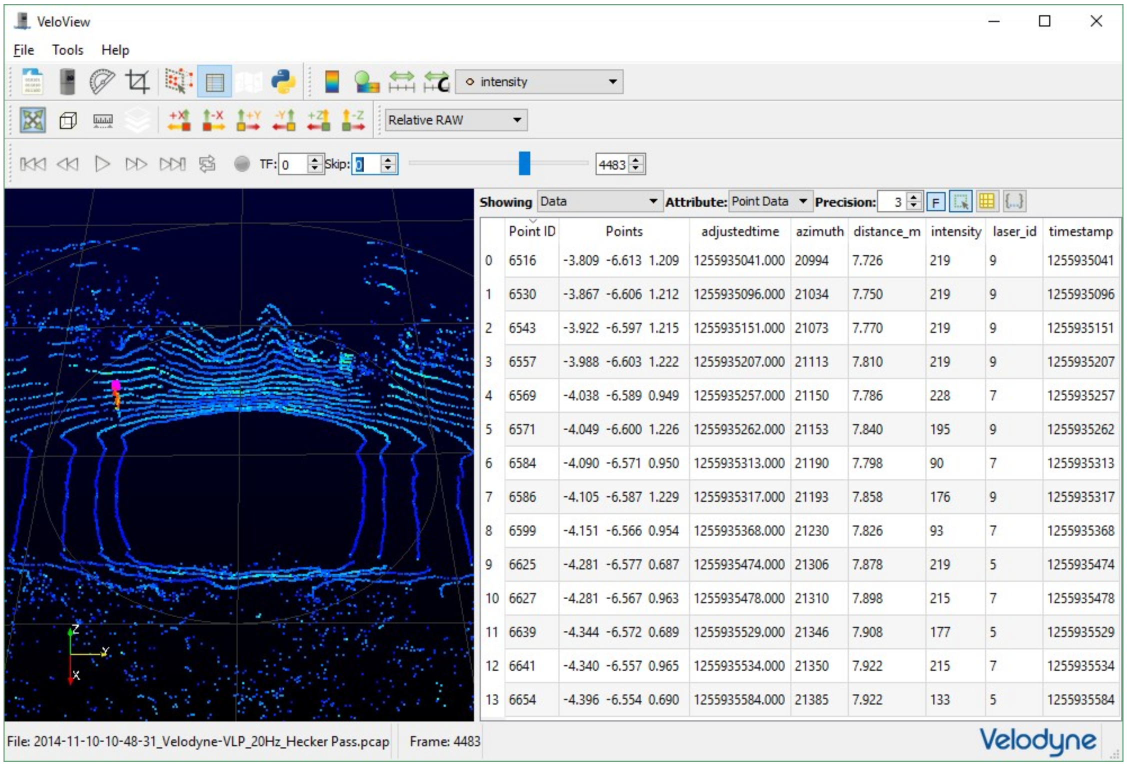
Figure E-13 VeloView Select All Points



TF: 0 Skip: 0

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



TF: 0 Skip: 0 4483

Showing Data Attribute: Point Data Precision: 3

	Point ID	Points	adjustedtime	azimuth	distance_m	intensity	laser_id	timestamp
0	6516	-3.809 -6.613 1.209	1255935041.000	20994	7.726	219	9	1255935041
1	6530	-3.867 -6.606 1.212	1255935096.000	21034	7.750	219	9	1255935096
2	6543	-3.922 -6.597 1.215	1255935151.000	21073	7.770	219	9	1255935151
3	6557	-3.988 -6.603 1.222	1255935207.000	21113	7.810	219	9	1255935207
4	6569	-4.038 -6.589 0.949	1255935257.000	21150	7.786	228	7	1255935257
5	6571	-4.049 -6.600 1.226	1255935262.000	21153	7.840	195	9	1255935262
6	6584	-4.090 -6.571 0.950	1255935313.000	21190	7.798	90	7	1255935313
7	6586	-4.105 -6.587 1.229	1255935317.000	21193	7.858	176	9	1255935317
8	6599	-4.151 -6.566 0.954	1255935368.000	21230	7.826	93	7	1255935368
9	6625	-4.281 -6.577 0.687	1255935474.000	21306	7.878	219	5	1255935474
10	6627	-4.281 -6.567 0.963	1255935478.000	21310	7.898	215	7	1255935478
11	6639	-4.344 -6.572 0.689	1255935529.000	21346	7.908	177	5	1255935529
12	6641	-4.340 -6.557 0.965	1255935534.000	21350	7.922	215	7	1255935534
13	6654	-4.396 -6.554 0.690	1255935584.000	21385	7.922	133	5	1255935584

File: 2014-11-10-10-48-31_Velodyne-VLP_20Hz_Hecker Pass.pcap Frame: 4483

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, most of their intensity values exceed 100, indicating something retro-reflective is there (see *Calibrated Reflectivity on page 34* for details).

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.

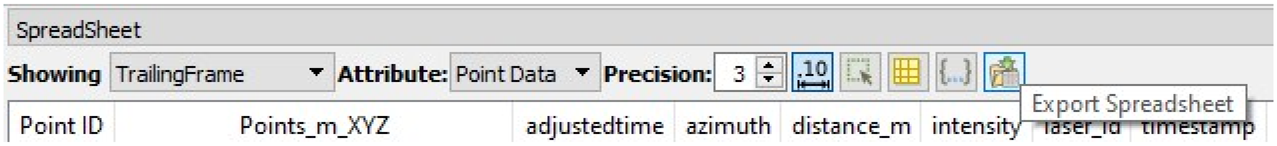
E.7 Exporting Data to PCAP or CSV Files

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)



Appendix F • Laser Pulse

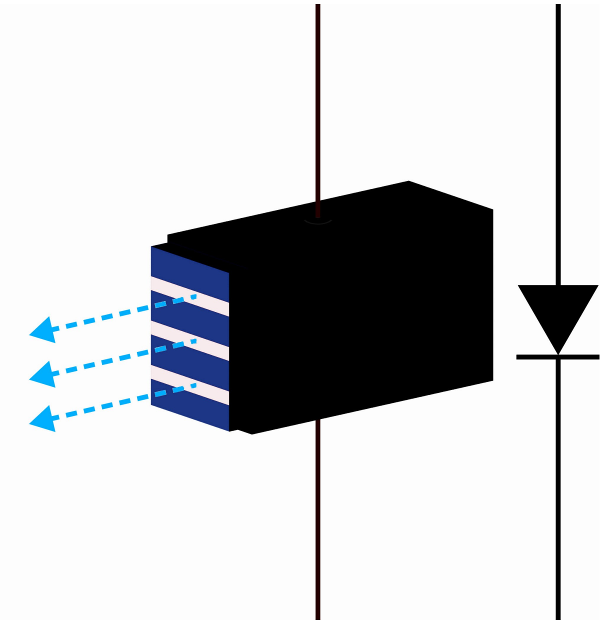
This section provides details on your sensor's laser diodes, the laser pulse and scan patterns, and certain beam characteristics of the laser pulses.

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

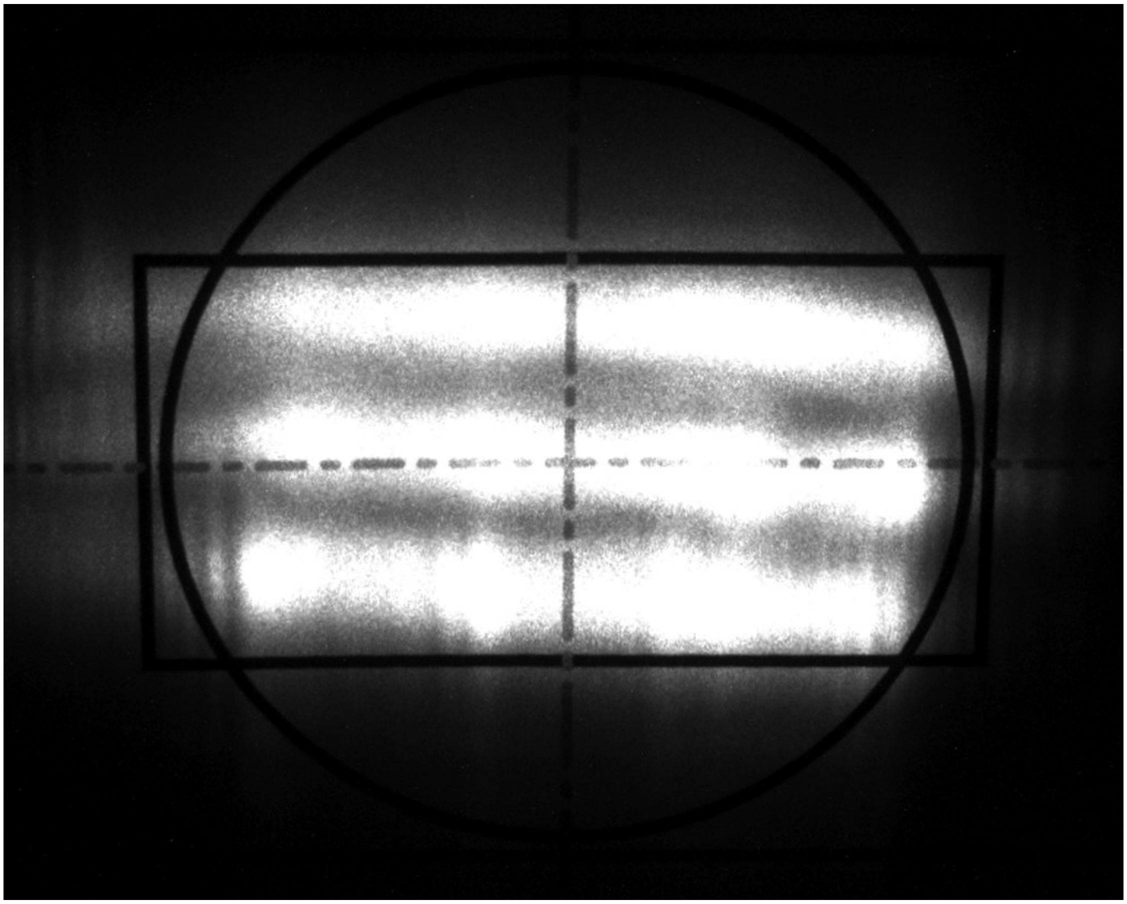
Laser firings produced by the sensor can be viewed with an infrared camera or smartphone camera sans IR filter. Photos in this section were taken with an infrared camera.

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 fingernail - but it doesn’t remain that size as it speeds away. Read more about that in *Beam Divergence on page 128*.

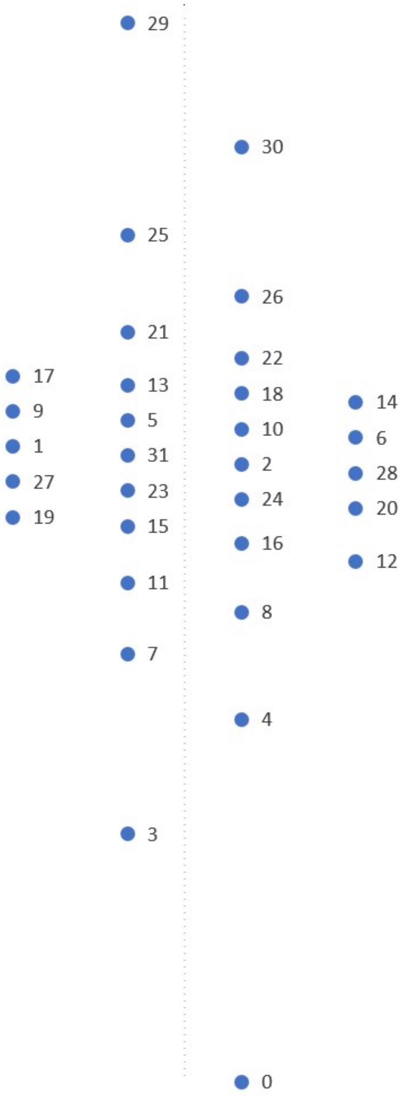
Figure F-2 Laser Spot Shape



F.2.2 Laser Scan Pattern

Inside each VLP-32C sensor is 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* across (azimuthally) the environment in a clockwise motion as seen from above. The thin dotted line down the center represents a single firing's Azimuth in degrees. See *Precision Azimuth Calculation on page 71* for details.

Figure F-3 VLP-32C Laser Pattern



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

Equation F-1 Gap Between Scan Lines

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

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.

VLP-32C 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 VLP-32C Beam Divergence

Horizontal Beam Divergence	Vertical Beam Divergence
3.43 mrad (0.1965245 deg)	1.72 mrad (0.09854874 deg)

Appendix G • Time Synchronization

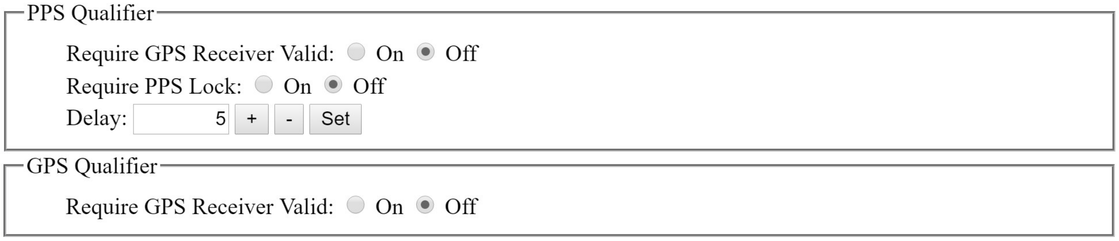
This section provides a detailed discussion of the **GPS Qualifier** and **PPS Qualifier** functions.

G.1 Introduction

These two new 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 new 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



PPS Qualifier

Require GPS Receiver Valid: ○ On ● Off

Require PPS Lock: ○ On ● Off

Delay: 5

GPS Qualifier

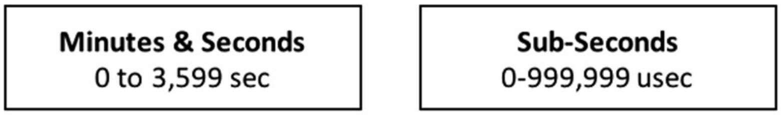
Require GPS Receiver Valid: ○ On ● Off

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 minutes and seconds since the top of the hour, and the other counter maintains the sub-second count (*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 76*) 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 minutes and seconds counter may be adjusted to the time values provided in the NMEA sentence. If no NMEA sentence is provided, the minutes and 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.

New 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.

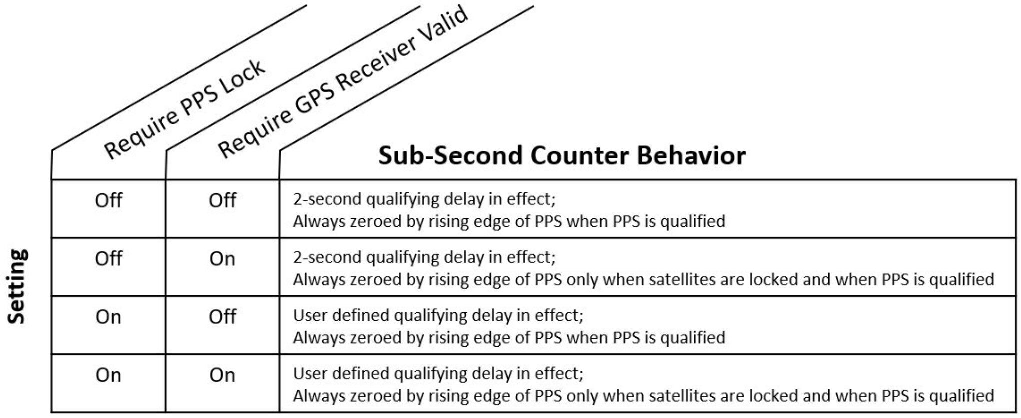
G.5 Application

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.

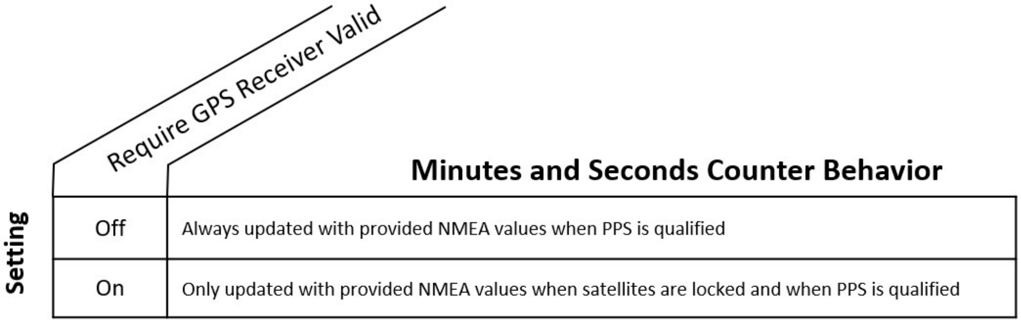
G.6 Logic Tables

Figure G-3 Sub-Second Counter Behavior



Setting	Require PPS Lock	Require GPS Receiver Valid	Sub-Second Counter Behavior
Setting	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
Setting	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 Minutes and 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.

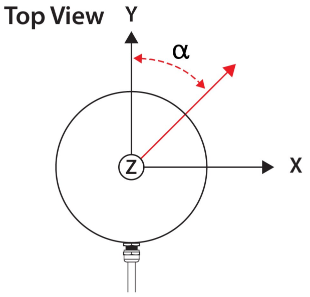
H.1 Phase Lock

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.

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.

Note: For Phase Lock to work correctly, the sensor's RPM must be set to a multiple of 60 RPM between 300 RPM and 1200 RPM (inclusive).

Figure H-1 Direction of Laser Firing



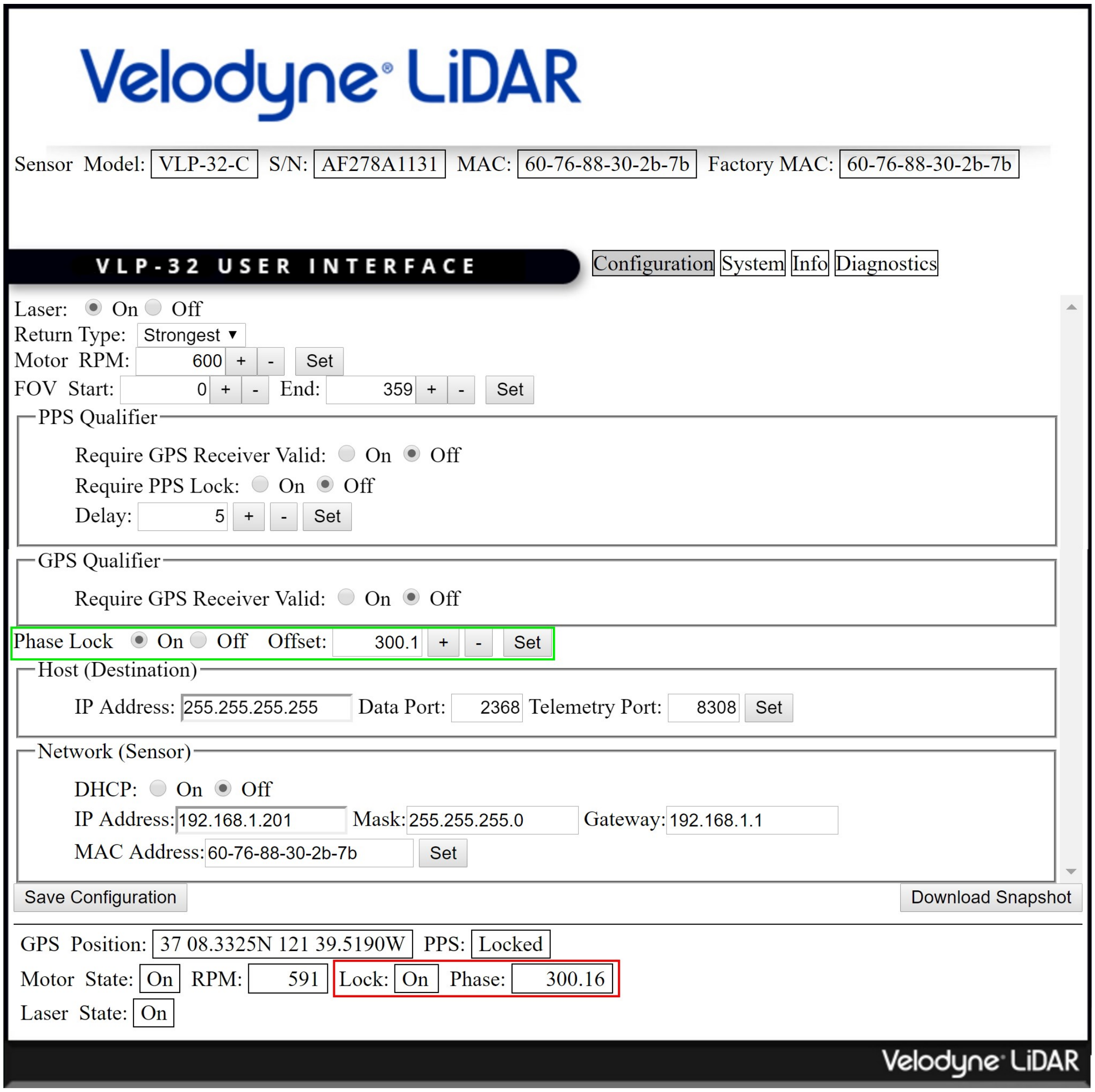
H.1.1 Setting the Phase Lock

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



Sensor Model: VLP-32-C S/N: AF278A1131 MAC: 60-76-88-30-2b-7b Factory MAC: 60-76-88-30-2b-7b

VLP-32 USER INTERFACE

Configuration System Info Diagnostics

Laser: ● On ○ Off

Return Type: Strongest ▾

Motor RPM: 600 + - Set

FOV Start: 0 + - End: 359 + - Set

PPS Qualifier

Require GPS Receiver Valid: ○ On ● Off

Require PPS Lock: ○ On ● Off

Delay: 5 + - Set

GPS Qualifier

Require GPS Receiver Valid: ○ On ● Off

Phase Lock ● On ○ Off Offset: 300.1 + - Set

Host (Destination)

IP Address: 255.255.255.255 Data Port: 2368 Telemetry Port: 8308 Set

Network (Sensor)

DHCP: ○ On ● Off

IP Address: 192.168.1.201 Mask: 255.255.255.0 Gateway: 192.168.1.1

MAC Address: 60-76-88-30-2b-7b Set

Save Configuration Download Snapshot

GPS Position: 37 08.3325N 121 39.5190W PPS: Locked

Motor State: On RPM: 591 Lock: On Phase: 300.16

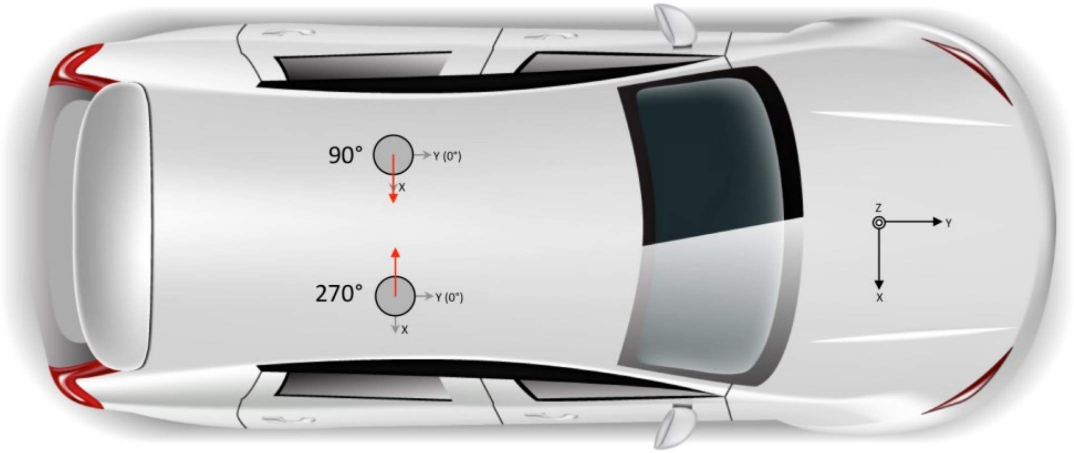
Laser State: On

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.

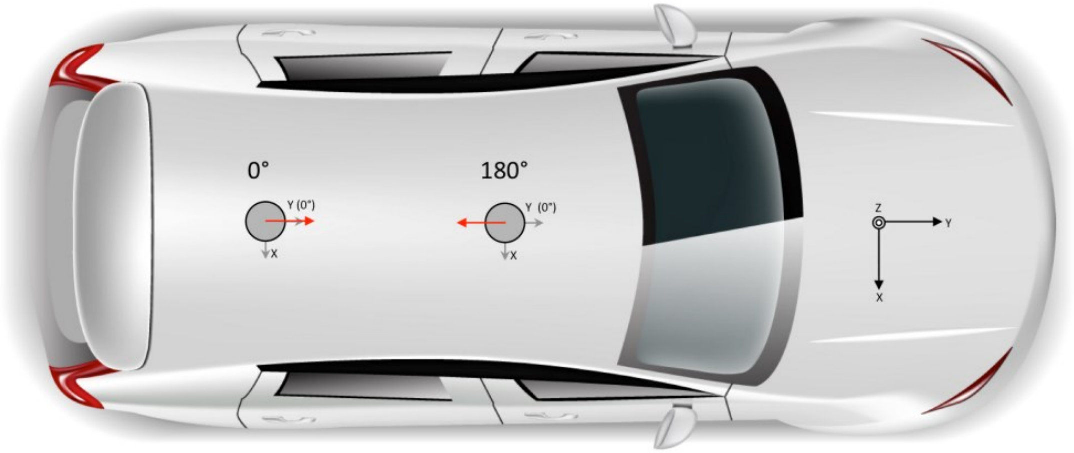
Figure H-3 on the facing page shows two sensors mounted on a vehicle. The sensor mounted on the car's left side has its phase lock offset set to 90°, and the phase lock offset of the sensor mounted on the right side of the vehicle is set to 270°, as shown by the red arrows.

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*.

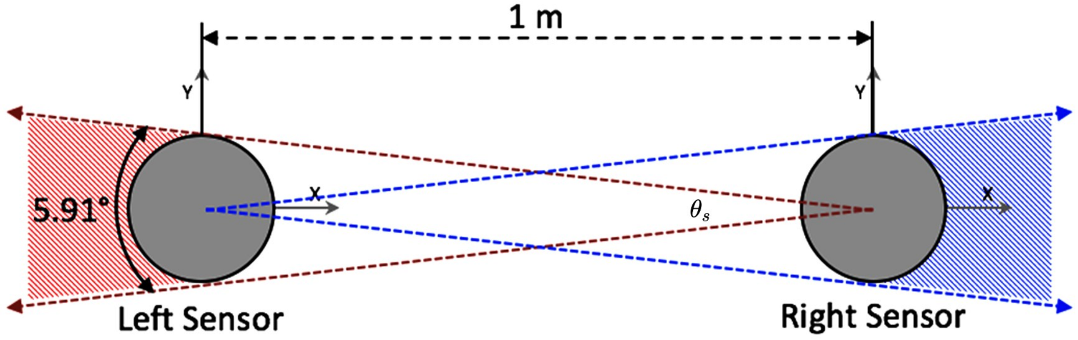
Figure H-4 Fore and Aft Sensor Phase Offset



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 99*) 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

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

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

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

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

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 76* and *Set Field of View on page 90* for more.

Appendix I • Sensor Care

To accurately sense its environment, the VLP-32C sensor should be kept clean, especially its ring lens.

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 VLP-32C 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 75*. Look for S/N at the top. Other ways include getting a snapshot from the command line (see *Get Snapshot on page 89*) or requesting the snapshot programmatically (see *curl Example using Python on page 92*). 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 VLP-32C'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 100 Mbps Ethernet NIC or switch with MDI or AUTO MDIX capability.

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 on the facing page*.

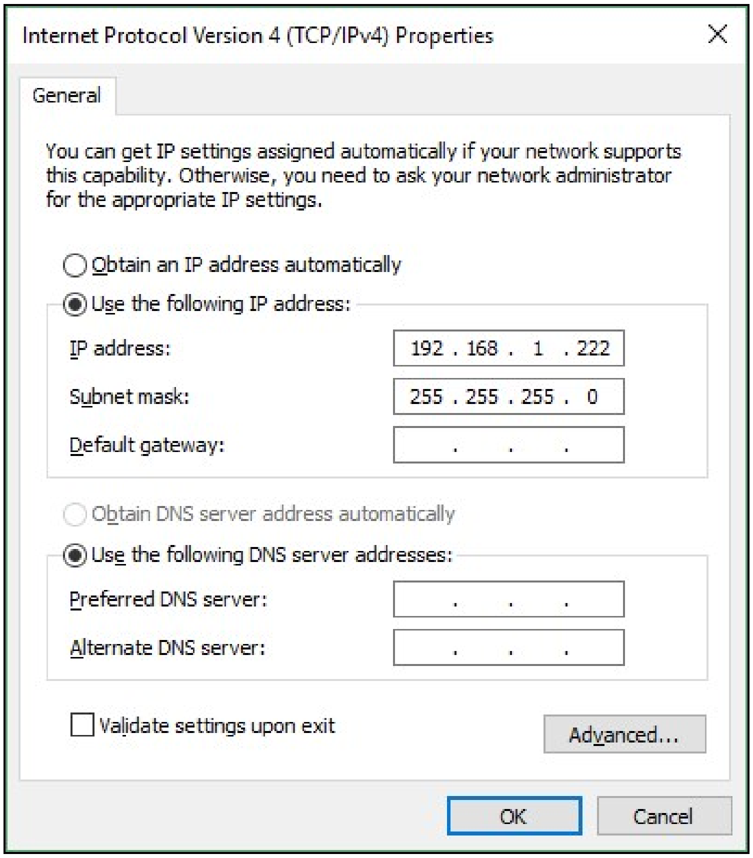
   "XXX" can be any number from 2 to 254 except 201.

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



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address:	192 . 168 . 1 . 222
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	. . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server:	. . .
Alternate DNS server:	. . .

□ Validate settings upon exit

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

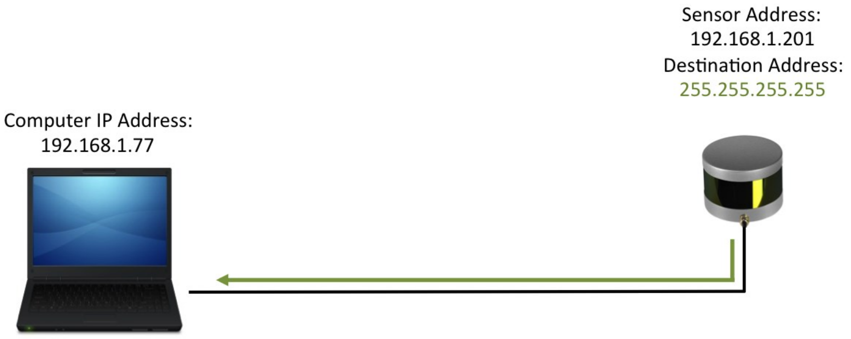
When actively sensing its environment, your sensor produces a lot of data which it transmits via Ethernet. The volume of data depends partly on which Return Type (or mode) it's in. Details on return modes can be found in *Laser Return Modes on page 34*.

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.

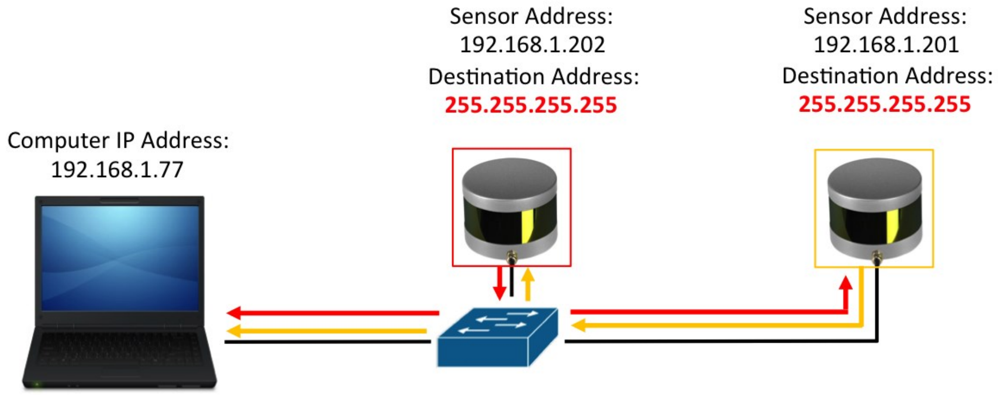
The scenarios below illustrate the wrong way and then a better way to configure them. (An alternative would be to direct-connect sensors to separate NICs.)

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 133* and *Phase Locking Multiple Sensors on page 39* 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

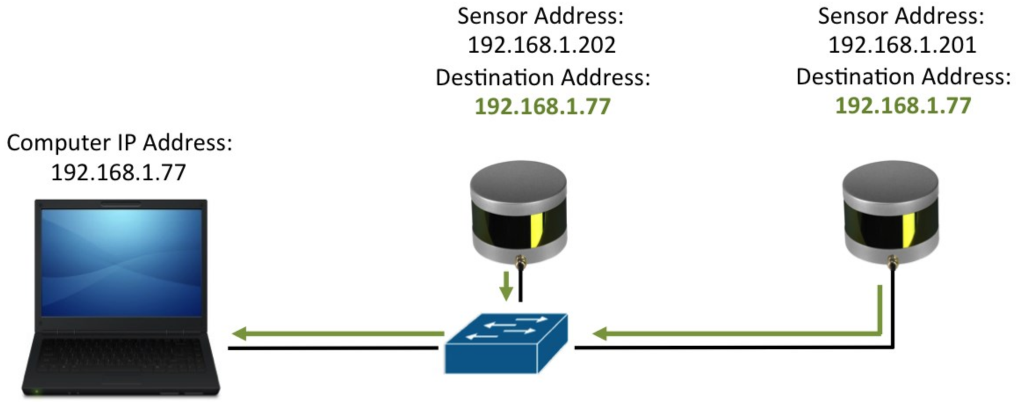


J.2.3.2 Multiple Sensors Transmitting to a Specific Address

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



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

VLP-32C User Manual
63-9325 Rev. D
Updated 2021-02-03