USER'S MANUAL AND PROGRAMMING GUIDE

Firmware Version 4.07

HDL-64E S2 and S2.1

High Definition LiDAR™ Sensor

i SAFETY NOTICES

1 INTRODUCTION

In The Box

2 PRINCIPLES OF OPERATION

3 INSTALLATION OVERVIEW

3 Front/Back Mounting

4 Side Mounting

5 Top Mounting

6 Wiring

6 USAGE

6 Use the Included Point-cloud Viewer

6 Develop Your Own Application-specific Point-cloud Viewer

7 db.xml Calibration Parameters

8 Change Run-Time Parameters

10 Control Spin Rate

— Change Spin Rate in Flash Memory

— Change Spin Rate in RAM Only

10 Limit Horizontal FOV data Collected

11 Define Sensor Memory IP Source and Destination Addresses

11 Upload Calibration Data

11 External GPS Time Synchronization

— GPS Receiver Option 1: Velodyne Supplied GPS Receiver

— GPS Receiver Option 2: Customer Supplied GPS Receiver

13 Packet Format and Status Byte for GPS Time Stamping

13 Time Stamping Accuracy Rules

13 Laser Firing Sequence and Timing

14 FIRMWARE UPDATE

15 APPENDIX A: Mechanical Drawings

16 APPENDIX B: Wiring Diagram

17 APPENDIX C: Digital Sensor Recorder (DSR)

17 Install

17 Calibrate

17 Live Playback

18 Record Data

18 Playback of Recorded Files

19 DSR Key Controls

19 DSR Mouse Controls

20 APPENDIX D: Matlab Sample Code

22 Reading Calibration and Sensor Parameter Data

23 APPENDIX E: Data Packet Format

27 Last Six Bytes Examples

30 APPENDIX F: Dual Two Point Calibration Methodology

34 APPENDIX G: Ethernet Transmit Timing Table

36 APPENDIX H: Laser and Detector Arrangement

37 APPENDIX I: Angular Resolution

38 TROUBLESHOOTING

38 SERVICE AND MAINTENANCE

39 SPECIFICATIONS

IMPORTANT SAFETY INSTRUCTIONS

CAUTION

RISK OF ELECTRIC SHOCK
DO NOT OPEN

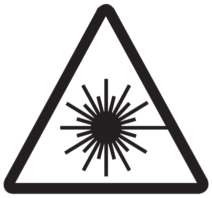
Caution

To reduce the risk of electric shock and to avoid violating the warranty, do not open sensor body. Refer servicing to qualified service personnel.

The lightning flash with arrowhead symbol is intended to alert the user to the presence of uninsulated “dangerous voltage” within the product’s enclosure that may be of sufficient magnitude to constitute a risk of electric shock to persons.

The exclamation point symbol is intended to alert the user to the presence of important operating and maintenance (servicing) instructions in the literature accompanying the product.

1. **Read Instructions —** All safety and operating instructions should be read before the product is operated.
2. **Retain Instructions —** The safety and operating instructions should be retained for future reference.
3. **Heed Warnings —** All warnings on the product and in the operating instructions should be adhered to.
4. **Follow Instructions —** All operating and use instructions should be followed.
5. **Servicing —** The user should not attempt to service the product beyond what is described in the operating instructions. All other servicing should be referred to Velodyne.



MAX Power: 1 mW
Wave Length: 905nm

Complies with 21 CFR 1040.10
and 1040.11 except for deviations
pursuant to Laser Notice No. 50,
dated 7/2001

INVISIBLE LASER RADIATION
DO NOT VIEW DIRECTLY WITH
OPTICAL INSTRUMENTS
CLASS 1M LASER PRODUCT

AVOID EXPOSURE-LASER
RADIATION IS EMITTED
FROM THESE APERTURES

Model No:
Serial No:
Mfg Date:

VELODYNE LiDAR, Inc.
MORGAN HILL, CA (USA)

Congratulations on your purchase of a Velodyne HDL-64E S2 or S2.1 High Definition LiDAR Sensor. These sensors represent a breakthrough in sensing technology by providing more information about the surrounding environment than previously possible. The HDL-64E S2 or S2.1 High Definition LiDAR sensors are referred to as the sensor throughout this manual.

This manual and programming guide covers:

- Installation and wiring
- HDL-64-ADAPT (GPS Adaptor Box)
- The data packet format
- The serial interface
- Software updates
- GPS installation notes
- Viewing the data
- Programming information

This manual applies to the two versions of the HDL-64E sensor, the S2 and S2.1, unless otherwise indicated. The table below compares the laser layout, vertical field of view (VFOV) and primary application of the two versions.

HDL-64E Version	Lower Laser Block	Upper Laser Block	Vertical Field of View (VFOV)	Primary Application
S2	32 lasers separated by ½° vertical spacing	32 lasers separated by 1/3° vertical spacing	+2 to -24.8°	Autonomous navigation
S2.1 (dual lower block)	32 lasers separated by ½° vertical spacing	32 lasers separated by ½° vertical spacing	31.5°	3D mapping

For the latest updates to this manual – check **www.velodynelidar.com**.

In the Box

Each shipment contains:

- Sensor
- HDL-64-ADAPT (GPS Adaptor Box)
- Wiring harness
- CD with user manual, calibration file (db.xml), timing table calculation file (.xls) and DSR viewer

The sensor operates, instead of a single laser firing through a rotating mirror, with 64 lasers fixed mounted on upper and lower laser blocks, each housing 32 lasers. Both laser blocks rotate as a single unit. With this design each of the lasers fires tens of thousands of times per second, providing exponentially more data points/second and a more data-intensive point cloud than a rotating mirror design. The sensor delivers a 360° horizontal Field of View (HFOV) and a 26.8° vertical FOV (31.5° VFOV for the S2.1).

Additionally, state-of-the-art digital signal processing and waveform analysis are employed to provide high accuracy, extended distance sensing and intensity data. The sensor is rated to provide usable returns up to 120 meters. The sensor employs a direct drive motor system with no belts or chains in the drive train.

See the specifications at the end of this manual for more information about sensor operating conditions.

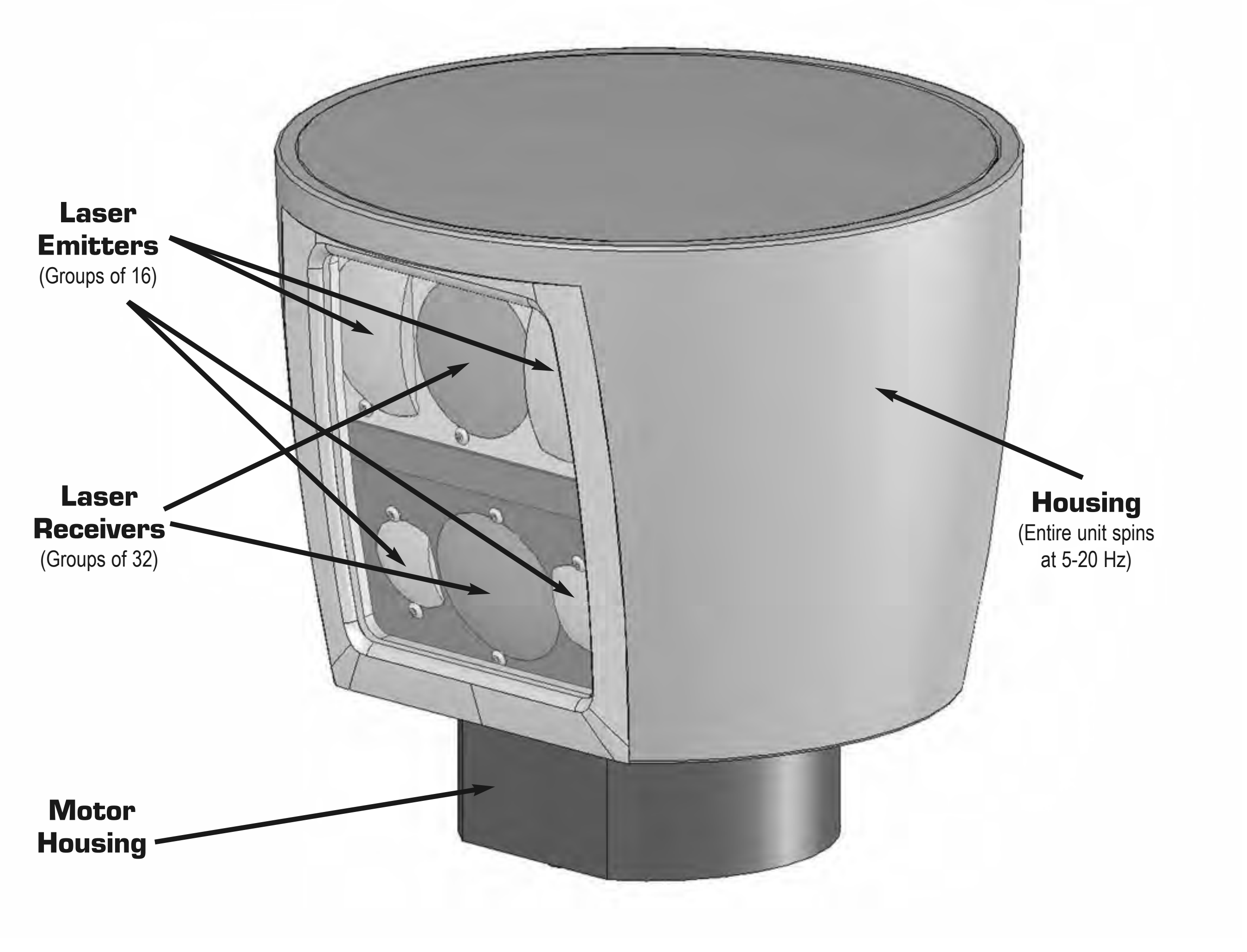


Figure 1. HDL-64E S2 design overview.

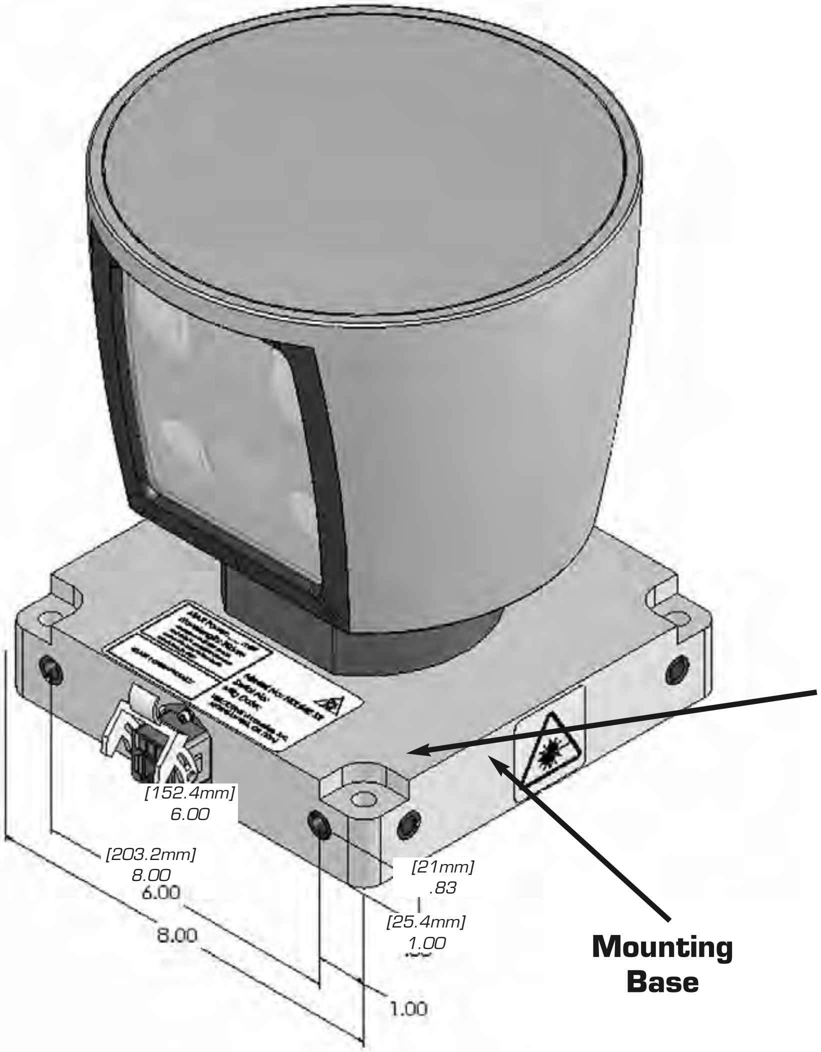
The sensor base provides the following mounting options:

- Front/Back mount (Figure 2)
- Side mount (Figure 3)
- Top mount (Figure 4)

The sensor can be mounted at any angle from 0 to 90° with respect to its base. Refer to Appendix A for complete dimensions. For all mounting options, mount the sensor to withstand vibration and shock without risk of detachment. Although helpful for longer life, the unit doesn't need to be shock proofed as it's designed to withstand standard automotive G-forces.

The sensor is weatherproofed to withstand wind, rain and other adverse weather conditions. The spinning of the sensor helps it shed excess water from the front window that could hamper performance.

Front/Back Mounting



Two M8-1.25mm x 12mm deep mounting points. (Two per side, for a total of 8.)

Mounting Base

Figure 2. Front and back HDL mounting illustration.

Side Mounting

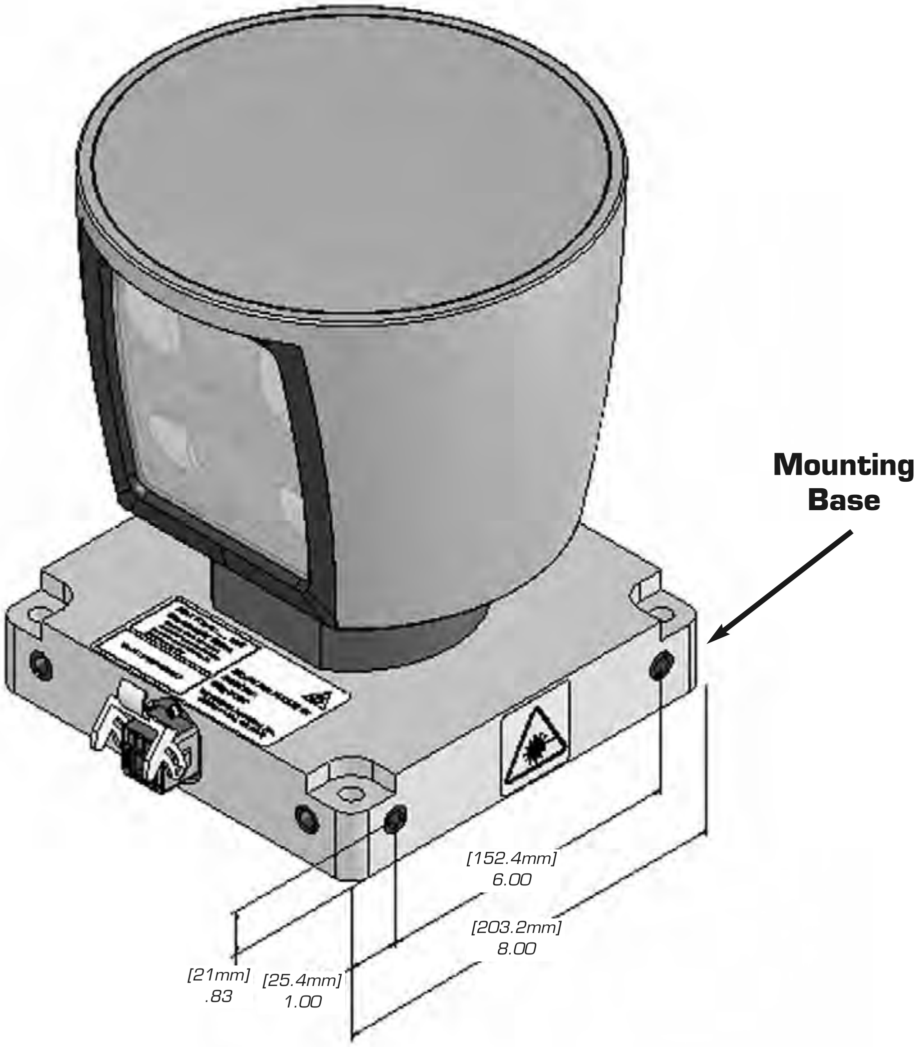

Mounting
Base
[152.4mm]
6.00
[203.2mm]
8.00
[21mm]
.83
[25.4mm]
1.00

Figure 3. Side HDL mounting illustration.

Top Mounting

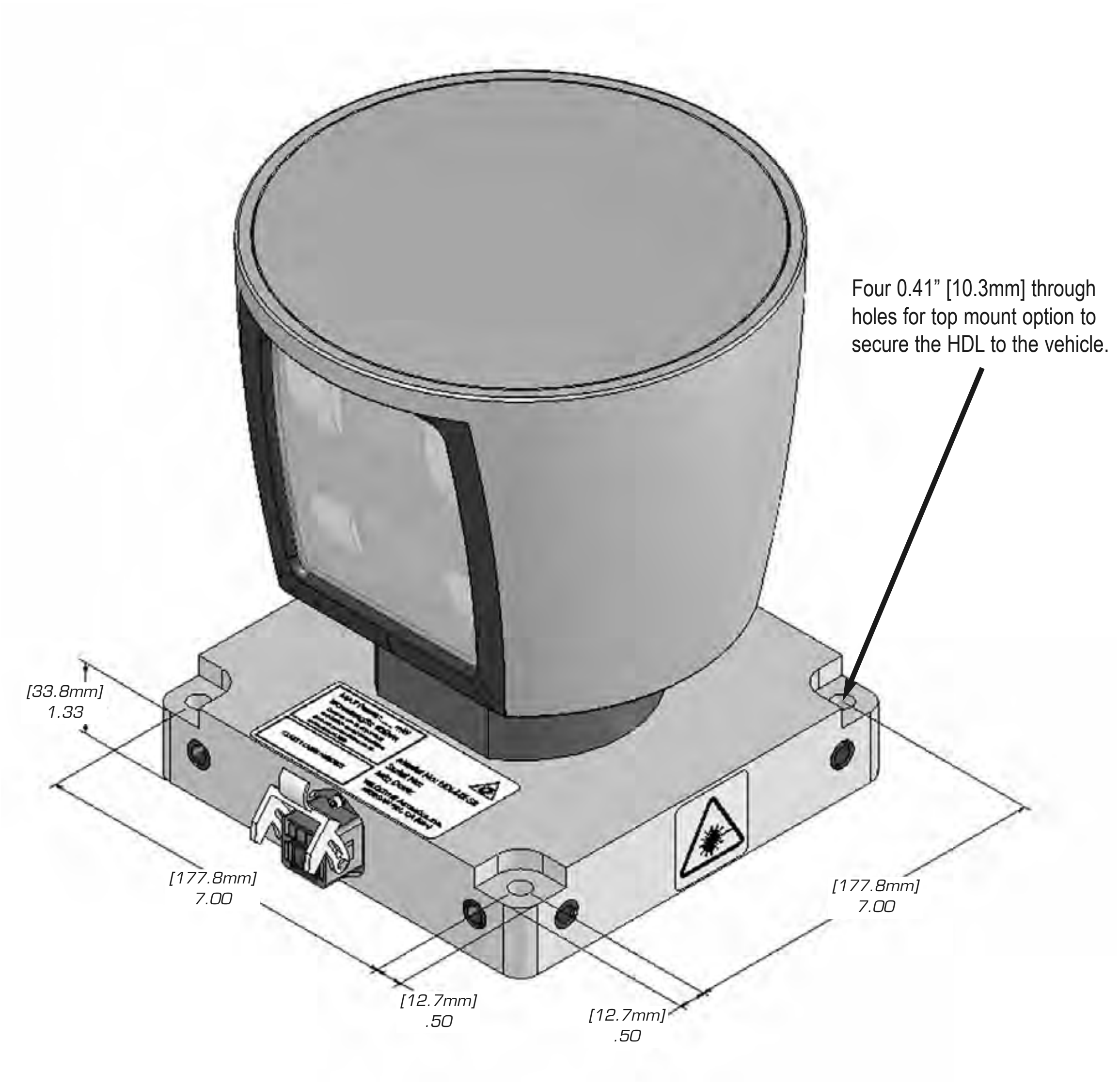

Four 0.41” [10.3mm] through
holes for top mount option to
secure the HDL to the vehicle.
[33.8mm]
1.33
[177.8mm]
7.00
[177.8mm]
7.00
[12.7mm]
.50
[12.7mm]
.50

Figure 4. Top HDL mounting illustration.

Wiring

The sensor comes with a pre-wired connector, wired with power, DB9 serial and standard RJ45 Ethernet connectors. The connector wires are approximately 10' [3 meters] in length.

Power. Connect the red and black wires to vehicle power. Be sure red is positive polarity. THE SENSOR IS RATED ONLY FOR 12 - 16 VOLTS. Any voltage applied over 16 volts could damage the sensor. The sensor draws 4-6 AMPS during normal usage.

The sensor doesn't have a power switch. It spins whenever power is applied.

Lockout Circuit. The sensor has a lockout circuit that prevents its lasers from firing until it achieves approximately 300 RPMs.

Ethernet. This standard Ethernet connector is designed to connect to a standard PC.

The sensor is only compatible with network cards that have either MDI or AUTO MDIX capability.

Serial Interface RS-232 DB9. This standard connector allows for a firmware update to be applied to the sensor. Velodyne may release firmware updates from time to time. It also accepts commands to change the RPM of the unit, control HFOV, change the unit's IP address, and other functions described later in this manual.

Wiring Diagram. If you need to wire your own connector for your installation, refer to the wiring diagram in Appendix B.

USAGE

The sensor needs no configuration, calibration, or other setup to begin producing usable data. Once the unit is mounted and wired, supplying it power causes it to start scanning and producing data packets.

Use the Included Point-cloud Viewer

The quickest way to view the data collected as an image is to use the included Digital Sensor Recorder (DSR). DSR is Velodyne's point-cloud processing data viewer software. DSR reads in the packets from the sensor over Ethernet, performs the necessary calculations to determine point locations and then plots the points in 3D on your PC monitor. You can observe both distance and intensity data through DSR. If you have never used the sensor before, this is the recommended starting point. For more on installing and using DSR, see Appendix C.

Develop Your Own Application-specific Point-cloud Viewer

Many users elect to develop their own application-specific point cloud tracking and plotting and/or storage scheme, which requires these fundamental steps:

1. Establish communication with the sensor.
2. Create a calibration table either from the calibration data included in-stream from the sensor or from the included db.xml data file.
3. Parse the packets for rotation, block, distance and intensity data
4. Apply the calibration factors to the data.
5. Plot or store the data as needed.

The following provides more detail on each of the above steps.

1. Establish communication with the sensor.

The sensor broadcasts UDP packets. By using a network monitoring tool, such as Wireshark, you can capture and observe the packets as they are generated by the sensor. See Appendix E for the UDP packet format. The default source IP address for the sensor is 192.168.3.043, and the destination IP address is 192.168.3.255. To change these IP addresses, see page 11.

2. Create an internal calibration table either from the calibration data included in-stream from the sensor or from the included db.xml data file.

This table must be built and stored internal to the point-cloud processing software. The easiest and most reliable way to build the calibration table is by reading the calibration data directly from the UDP data packets. A MatLab example of reading and building such a table can be found in Appendix D and on the CD included with the sensor named **CALTABLEBUILD.m**.

Alternatively, the calibration data can be found in the included db.xml file found on the CD included with the sensor. A description of the calibration data is shown in the following table.

db.xml Calibration Parameters

Parameter	Unit	Description	Values
rotCorrection	degree	The rotational correction angle for each laser, as viewed from the back of the unit.	Positive factors rotate to the left. Negative values rotate to the right.
vertCorrection	degree	The vertical correction angle for each laser, as viewed from the back of the unit.	Positive values have the laser pointing up. Negative values have the laser pointing down.
distCorrection	cm	Far distance correction of each laser distance due to minor laser parts' variances.	Add directly to the distance value read in the packet.
distCorrectionX	cm	Close distance correction in X of each laser due to minor laser parts variances interpolated with far distance correction then applied to measurement in X.	
distCorrectionY	cm	Close distance correction in Y of each laser due to minor laser parts variances interpolated with far distance correction then applied to measurement in Y.	
vertOffsetCorrection	cm	The height of each laser as measured from the bottom of the base.	One fixed value for all upper block lasers. Another fixed value for all lower block lasers.
horizOffsetCorrection	cm	The horizontal offset of each laser as viewed from the back of the laser.	Fixed positive or negative value for all lasers.
Maximum Intensity			Value from 0 to 255. Usually 255.
Minimum Intensity			Value from 0 to 255. Usually 0.
Focal Distance		Maximum intensity distance.	
Focal Slope		The control intensity amount.	

The calibration table, once assembled, contains 64 instances of the calibration values shown in the table above to interpret the packet data to calculate each point's position in 3D space. Use the first 32 points for the upper block and the second 32 points for the lower block. The rotational info found in the packet header is used to determine the packets position with respect to the 360° horizontal field of view.

3. **Parse the packets for rotation, block, distance and intensity data.** Each sensor's LIFO data packet has a 1206 byte payload consisting of 12 blocks of 100 byte firing data followed by 6 bytes of calibration and other information pertaining to the sensor.

   Each 100 byte record contains a block identifier, then a rotational value followed by 32 3-byte combinations that report on each laser fired for the block. Two bytes report distance to the nearest 0.2 cm, and the remaining byte reports intensity on a scale of 0 -255. 12 100 byte records exist, therefore, 6 records exist for each block in each packet. For more on packet construction, see Appendix E.

4. **Apply the calibration factors to the data.** Each of the sensor's lasers is fixed with respect to vertical angle and offset to the rotational index data provided in each packet. For each data point issued by the sensor, rotational and horizontal correction factors must be applied to determine the point's location in 3D space referred to by the return. Intensity and distance offsets must also be applied. Each sensor comes from Velodyne's factory calibrated using a dual-point calibration methodology, explained further in Appendix F.

The minimum return distance for the sensor is approximately 3 feet (0.9 meters). Ignore returns closer than this.

A file on the CD called "HDL Source Example" shows the calculations using the above correction factors. This DSR uses this code to determine 3D locations of sensor data points.

5. **Plot or store the data as needed.** For DSR, the point-cloud data, once determined, is plotted onscreen. The source to do this can be found on the CD and is entitled "HDL Plotting Example." DSR uses OpenGL to do its plotting.

   You may also want to store the data. If so, it may be useful to timestamp the data so it can be referenced and coordinated with other sensor data later. The sensor has the capability to synchronize its data with GPS precision time. For more in this capability, see page 11.

Change Run-Time Parameters

The sensor has several run-time parameters that can be changed using the RS-232 serial port. For all commands, use the following serial parameters:

- Baud 9600
- Parity: None
- Data bits: 8
- Stop bits: 1

All serial commands, except one version of the spin rate command, store data in the sensor's flash memory. Data stored in flash memory through serial commands is retained during firmware updates or power cycles.

The sensor has no echo back feature, so no serial data is returned from the sensor. Commands can be sent using a terminal program or by using batch files (e.g. .bat). A sample .bat file is shown below.

Sample Batch File (.bat)

```
MODE COM3: 9600,N,8,1 COPY SERCMD.txt COM3 Pause
```

Sample SERCMD.txt file

This command sets the spin rate to 300 RPM and stores the new value in the unit's flash memory.

```
#HDLRPM0300$
```

Available commands

The following run-time commands are available with the sensor:

Command	Description	Parameters
#HDLRPMnnnn$	Set spin rate from 300 to 1200 RPM n flash memory (default is 600 RPM)	nnnn is an integer between 0300 and 1200
#HDLRPNnnnn$	Set spin rate from 300 to 1200 RPM in RAM (default is 600 RPM)	nnnn is an integer between 0300 and 1200
#HDLIPAssssssssssssdddddddddddd$	Change source and/or destination IP address	• ssssssssssss is the source 12-digit IP address • dddddddddddd is the destination 12-digit IP address
#HDLFOVsssnnn$	Change horizontal Field of View (HFOV)	• sss = starting angle in degrees; sss is an integer between 000 and 360 • nnn = ending angle in degrees; nnn is an integer between 000 and 360

You can also upload calibration data from db.xml into flash memory and use GPS time synchronization.

.

Control Spin Rate

Change Spin Rate in Flash Memory

The sensor can spin at rates ranging from 300 RPM (5 Hz) to 1200 RPM (20 Hz). The default is 600 RPM (10 Hz). Changing the spin rate does not change the data rate – the unit sends out the same number of packets (at a rate of ~1.3 million data points per second) regardless of its spin rate. The horizontal image resolution increases or decreases depending on rotation speed.

See the Angular Resolution section found in Appendix I for the angular resolution values for various spin rates.

To control the sensor's spin rate, issue a serial command of the case sensitive format #HDLRPMnnnn$ where nnnn is an integer between 0300 and 1200. The sensor immediately adopts the new spin rate. You don't need to power cycle the unit, and the new RPM is retained with future power cycles.

Change Spin Rate in RAM Only

If repeated and rapid updates to the RPM are needed, such as for synchronizing multiple sensors controlled by a closed loop application, you can adjust the sensors' spin rates without storing the new RPM in flash memory (this preserves flash memory over time).

To control the sensor's spin rate in RAM only, issue a serial command of the case sensitive format #HDLRPNnnnn$ where nnnn is an integer between 0300 and 1200. The sensor immediately adopts the new spin rate. You shouldn't power cycle the sensor as the new RPM is lost with future power cycles, which returns to the last known RPM.

Limit Horizontal FOV Data Collected

The sensor defaults to a 360° surrounding view of its environment. It may be desirable to reduce this horizontal Field of View (HFOV) and, hence, the data created.

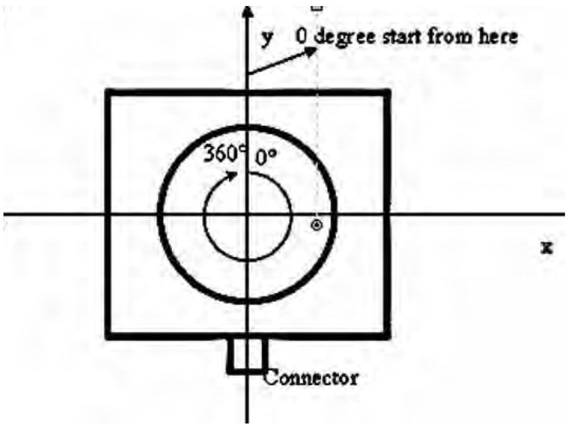
To limit the horizontal FOV, issue a serial command of the case sensitive format #HDLFOVsssnnn$ where:

- sss = starting angle in degrees; sss is an integer between 000 and 360
- nnn = ending angle in degrees; nnn is an integer between 000 and 360

The HDL unit immediately adopts the new HFOV angles without power cycling and will retain the new HFOV settings upon power cycle.

Regardless of the FOV setting, the lasers will always fire at the full 360° HFOV. Limiting the HFOV only limits data transmission to the HFOV of interest.

The following diagram shows the HFOV from the top view of the sensor.

<table>
<tr><th colspan="2">Examples</th></tr>
<tr><td>
</td><td>Case 1: FOV 0° to 360°
FOV command: #HDLFOV000360$
Case 2: FOV 0° to 90°
FOV command: #HDLFOV000090$
Case 3: FOV -90° to 90°
FOV command: #HDLFOV270090$</td></tr>
</table>

Top view of Sensor

Define Sensor Memory IP Source and Destination Addresses

The HDL-64E comes with the following default IP addresses:

- **Source:** 192.168.3.043
- **Destination:** 192.168.3.255

To change either of the above IP addresses, issue a serial command of the case sensitive format #HDLIPAssssssssssssdddddddddddd$ where,

- ssssssssssss is the source 12-digit IP address
- dddddddddddd is the destination 12-digit IP address

Use all 12 digits to set an IP address. Use 0 (zeros) where a digit would be absent. For example, 192168003043 is the correct syntax for IP address 192.168.3.43.

The unit must be power cycled to adopt the new IP addresses.

Upload Calibration Data

Sensors use the db.xml file exclusively for calibration data. The calibration data found in db.xml can be uploaded and saved to the unit's flash memory by following the steps outlined below.

1. Locate the files HDLCAL.bat, loadcal.exe, and db.xml on the CD and copy them to the same directory on your PC connected to the sensor.
2. Edit HDLCAL.bat to ensure the copy command lists the right COM port for RS-232 communication with the sensor.
3. Run HDLCAL.bat and ensure successful completion.
4. The sensor received and saved the calibration data.

To verify successful load of the calibration data, ensure the date and time of the upload have been updated. Refer to Appendix E for where in the data packets this data can be located.

External GPS Time Synchronization

The sensor can synchronize its data with precision GPS-supplied time pulses so you can ascertain the exact firing time of each laser in any particular packet. The firing time of the first laser in a particular packet is reported in the form of microseconds since the top of the hour, and from that time each subsequent laser's firing time can be derived via the table published in Appendix H and included on the CD.

Calculating the exact firing time requires a GPS receiver generating a sync pulse and the $GPRMC NMEA record over a dedicated RS-232 serial port. The output from the GPS receiver is connected to an external GPS adaptor box supplied by Velodyne that conditions the signal and passes it to the sensor. The GPS receiver can either be supplied by Velodyne or the customer can adapt their GPS receiver to provide the required sync pulse and NMEA record.

GPS Receiver Option 1: Velodyne Supplied GPS Receiver

Velodyne provides an optional pre-programmed GPS receiver (HDL-64-GPS) This receiver is pre-wired with an RS-232 connector that plugs into the GPS adapter box. To obtain a pre-programmed GPS receiver, contact Velodyne sales or service.

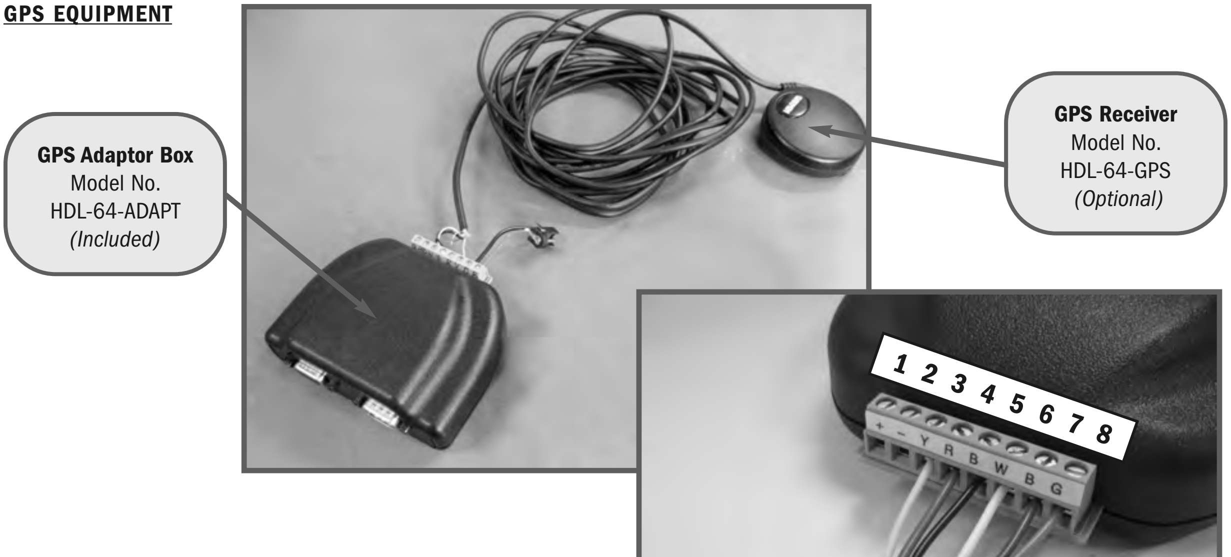
GPS Receiver Option 2: Customer Supplied GPS Receiver

You can supply and configure your own GPS device. If using your own GPS device:

- Issue a once-a-second synchronization pulse, typically output over a dedicated wire.
- Configure an available RS-232 serial port to issue a once-a-second $GPRMC NMEA record. No other output can be accepted from the GPS device.
- Issue the sync pulse and NMEA record sequentially.
- The sync pulse length is not critical (typical lengths are between 20ms and 200ms)
- Start the $GPRMC record between 50ms and 500ms after the end of the sync pulse.
- Configure the $GPRMC record either in the hhmmss or hhmmss.s format.

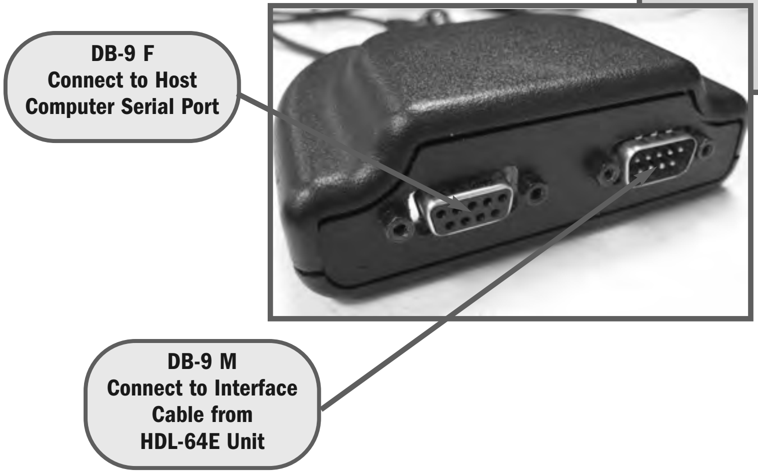
The images below show the GPS adaptor box, included with the HDL-64E, and optional GPS receiver.

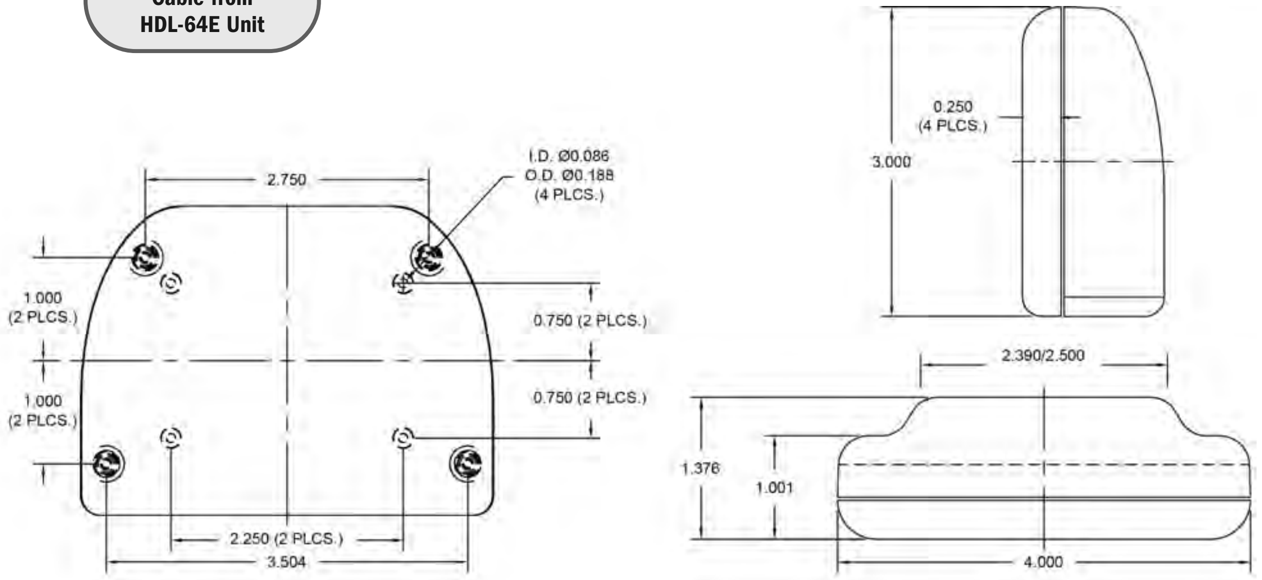
GPS EQUIPMENT


GPS Adaptor Box
Model No.
HDL-64-ADAPT
(Included)
GPS Receiver
Model No.
HDL-64-GPS
(Optional)
1 2 3 4 5 6 7 8
+ – Y R B W B G

GPS Adaptor Box Front & Back View

#	COLOR	SIGNAL NAME
1	Red	+12V DC Power
2	Black	Power Ground
3	Yellow	1 PPS (positive edge only)
4	Red	Vin (+5V)
5	Black	Ground
6	White	Transmit Data
7	Brown	Ground (Drain Wire)
8	Green	Receive Data


DB-9 F
Connect to Host Computer Serial Port
DB-9 M
Connect to Interface Cable from HDL-64E Unit


2.750
I.D. Ø0.086
O.D. Ø0.188
(4 PLCS.)
1.000 (2 PLCS.)
1.000 (2 PLCS.)
0.750 (2 PLCS.)
0.750 (2 PLCS.)
2.250 (2 PLCS.)
3.504
0.250 (4 PLCS.)
3.000
2.390/2.500
1.376
1.001
4.000

Packet Format and Status Byte for GPS Time Stamping

The 6 bytes at the end of the data packet report GPS timing and synchronization data. For every packet, the last 6 bytes are formatted as follows:

Timestamp Bytes in Reverse Order in Microseconds		
Bytes	**Description**	**Notes**
4	GPS timestamp	32 bit unsigned integer timestamp. This value represents microseconds from the top of the hour to the first laser firing in the packet.
1	Status Type	8 bit ASCII status character as described in Appendix E. The status byte rotates through many kinds of sensor information.
1	Status Value	8 bit data as described in Appendix E.

Within the GPS status byte, there are 4 GPS status indicators:

- **0**: no GPS connection.
- **A**: both PPS and GPS command have signal.
- **V**: only GPS command signal, no PPS.
- **P**: only PPS signal, no GPS time command.

Time Stamping Accuracy Rules

The following rules and subsequent accuracy apply for GPS timestamps:

GPS Connection	Timestamp Info	Accuracy	Notes
GPS isn't connected (GPS Status 0)	The sensor starts running on its own clock starting at midnight Jan 1 2000. This date and time data is reflected in the H, M, S, D, N, and Y data values.	Expect a drift of about 5 seconds/day	The sensor clock does not correct for leap years. See Appendix E for more information.
GPS is connected	The H, M, S, D, N, and Y data values are obtained from the $GPRMC NMEA record.	GPS time synching runs in one of two modes: • The GPS has an internal clock that runs for several weeks that is used first. The accuracy is that of the GPS device employed. • When the GPS achieves lock, the sensor clock is then within +/-50µs of the correct time at all times.	
GPS is disconnected after being connected	The sensor continues to run on its own clock.	Expect drift of about 5 seconds/day	

Laser Firing Sequence and Timing

If the GPS timestamp feature is used, it can be useful to determine the exact firing time for each laser so as to properly time-align the sensor point cloud with other data sources.

The upper block and lower block collect distance points simultaneously, with each block issuing one laser pulse at a time. That is, each upper block laser fires in sequence and in unison with a laser from the lower block.

Lasers are numbered sequentially starting with 0 for the first lower block laser to 31 for the last lower block laser; and 32 for the first upper block laser to 63 for the last upper block laser. For example, laser 32 fires simultaneously with laser 0, laser 33 fires with laser 1, and so on.

The sensor has an equal number of upper and lower block returns. Hence, when interpreting the delay table, each sequential pair of data blocks represents the upper and lower block respectively. Each upper and lower block data pair in the Ethernet packet has the same delay value.

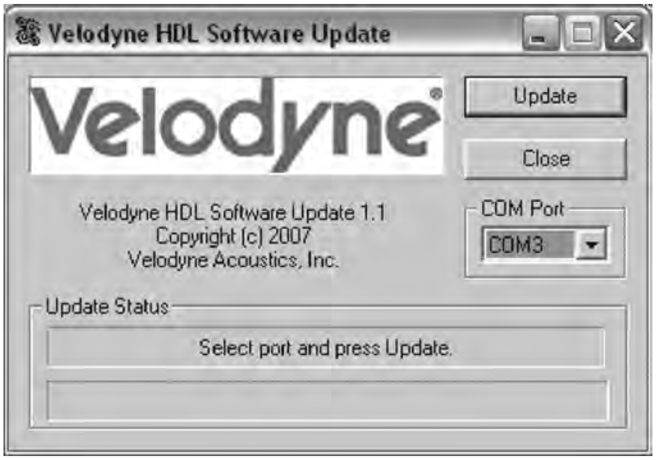
The first firing of a laser pair occurs 419.3 μs after the issuance of the fire command. Six firings of each block takes 139 μs and then the collected data is transmitted. It takes 100 μs to transmit the entire 1248 byte Ethernet packet. This is equal to 12.48 Bytes/μs and 0.080128 μs/Byte. See Appendix E for more information.

A timing table, shown in Appendix G, shows how much time elapses between the actual capturing of a distance point and when that point is output from the device. By registering the event of the Ethernet data capture, you can calculate back in time the exact time at which any particular distance point was captured.

FIRMWARE UPDATE

From time to time Velodyne issues firmware updates. To update the sensor's firmware:

1. Obtain the update file from Velodyne.
2. Connect the wiring harness RS-232 cable to a standard Windows compatible PC or laptop serial port.
3. Power up the sensor.
4. Execute the update file; the screen below appears.

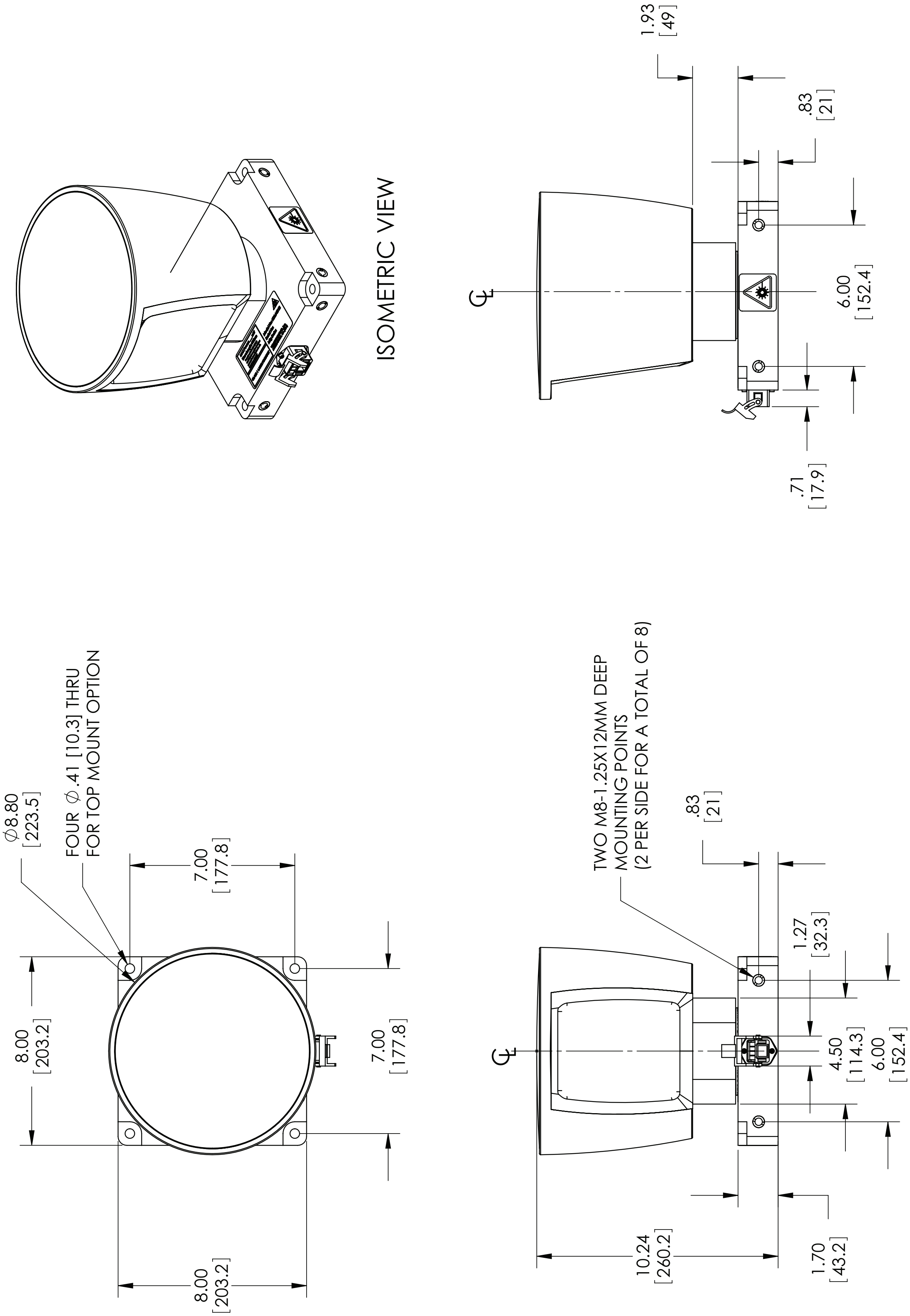


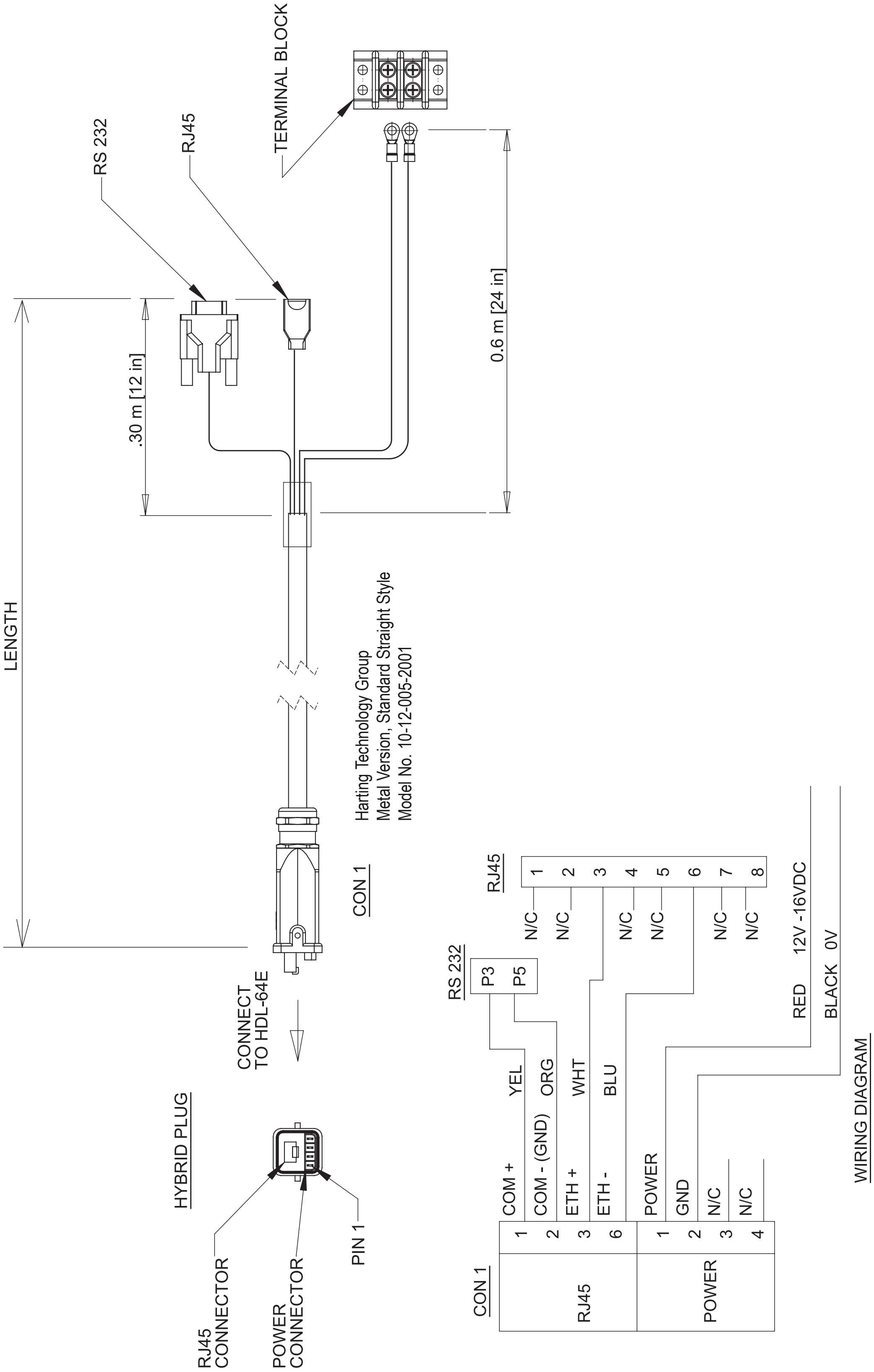
Velodyne HDL Software Update 1.1
Copyright (c) 2007
Velodyne Acoustics, Inc.

Select port and press Update.

Figure 5. HDL software update screen capture.

5. Select the appropriate **COM port**.
6. **Click Update.**
7. The firmware is uploaded and check summed before it is applied to the flash memory inside the sensor. If the checksum is corrupted, no update occurs. This protects the sensor in the event of power or data loss during the update.
   - If the update is successful, the sensor begins to spin down for a few seconds and then powers back up with the new firmware running.
   - If the update is not successful, try the update several times before seeking assistance from Velodyne.


ISOMETRIC VIEW
1.93 [49]
.83 [21]
6.00 [152.4]
.71 [17.9]
℄
∅8.80 [223.5]
FOUR ∅.41 [10.3] THRU FOR TOP MOUNT OPTION
7.00 [177.8]
8.00 [203.2]
7.00 [177.8]
8.00 [203.2]
TWO M8-1.25X12MM DEEP MOUNTING POINTS (2 PER SIDE FOR A TOTAL OF 8)
.83 [21]
1.27 [32.3]
4.50 [114.3]
6.00 [152.4]
℄
10.24 [260.2]
1.70 [43.2]


TERMINAL BLOCK
RS 232
RJ45
LENGTH
.30 m [12 in]
0.6 m [24 in]
Harting Technology Group
Metal Version, Standard Straight Style
Model No. 10-12-005-2001
CON 1
CONNECT TO HDL-64E
HYBRID PLUG
RJ45 CONNECTOR
POWER CONNECTOR
PIN 1
CON 1
RJ45
1 COM +
2 COM - (GND)
3 ETH +
6 ETH -
POWER
1 POWER
2 GND
3 N/C
4 N/C
YEL
ORG
WHT
BLU
RS 232
P3
P5
RJ45
1 N/C
2 N/C
3
4 N/C
5 N/C
6
7 N/C
8 N/C
RED 12V -16VDC
BLACK 0V

WIRING DIAGRAM

Digital Sensor Recorder (DSR)

DSR is a 3D point-cloud visualization software program designed for use with the sensor. This software is an “out of the box” tool for the rendering and recording of point cloud data from the HDL unit.

You can develop visualization software using the DSR as a reference platform. A code snippet is provided on the CD to aid in understanding the methods at which DSR parses the data points generated by the HDL sensor. See page 20 for more information.

Install

To install the DSR on your computer:

1. Locate the DSR executable program on the provided CD.
2. Double-click this DSR executable file to begin the installation onto a computer connected to the sensor. We recommend that you use the default settings during the installation.
3. Copy the db.xml file supplied with the sensor into the same directory as the DSR executable (defaults to c:\program files\ Digital Sensor Recorder). You may want to rename the existing default db.xml that comes with the DSR install.

Failure to use the calibration db.xml file supplied with your sensor will result in an inaccurate point cloud rendering in DSR.

Calibrate

The db.xml file provided with the sensor contains correction factors for the proper alignment of the point cloud information gathered for each laser. When implemented properly, the image viewable from the DSR is calibrated to provide an accurate visual representation of the environment in which the sensor is being used. Also use these calibration factors and equations in any program using the data generated by the unit.

Live Playback

For live playback:

1. Secure and power up the sensor so that it is spinning.
2. Connect the RJ45 Ethernet connector to your host computer’s network connection. You may wish to use auto DNS settings for your computers network configuration.
3. Open DSR from your desktop icon created during the installation.

DSR desktop icon

4. Select Options from the menu.
5. Select the proper input device.
6. Go to Options again.
7. Deselect the Show Ground Plane option. (Leave this feature off for the time being or until the ground plane has been properly adjusted).
8. (Optional) Go to Options > Properties to change the individual settings for each LASER channel.
9. Provided that your computer is now receiving data packets, click the Refresh button to start live viewing of a point cloud. The initial image is of a directly overhead perspective. See page 19 for mouse and key commands used to manipulate the 3D image within the viewer.

REFRESH button =

The image can be manipulated in all directions and become disorienting. If you lose perspective, simply press F1 to return to the original view.

Record Data

1. Confirm the input of streaming data through the live playback feature.
2. Click the **Record button**.

RECORD button = ●

5. Enter the name and location for the pcap file to be created.
6. Recording begins immediately once the file information has been entered.
7. Click Record again to discontinue the capture.
8. String multiple recordings together on the same file by performing the Record function repeatedly. A new file name isn't requested until after the session has been aborted.

An Ethernet capture utility, such as Wireshark, can also be used as a pcap capture utility.

Playback of Recorded Files

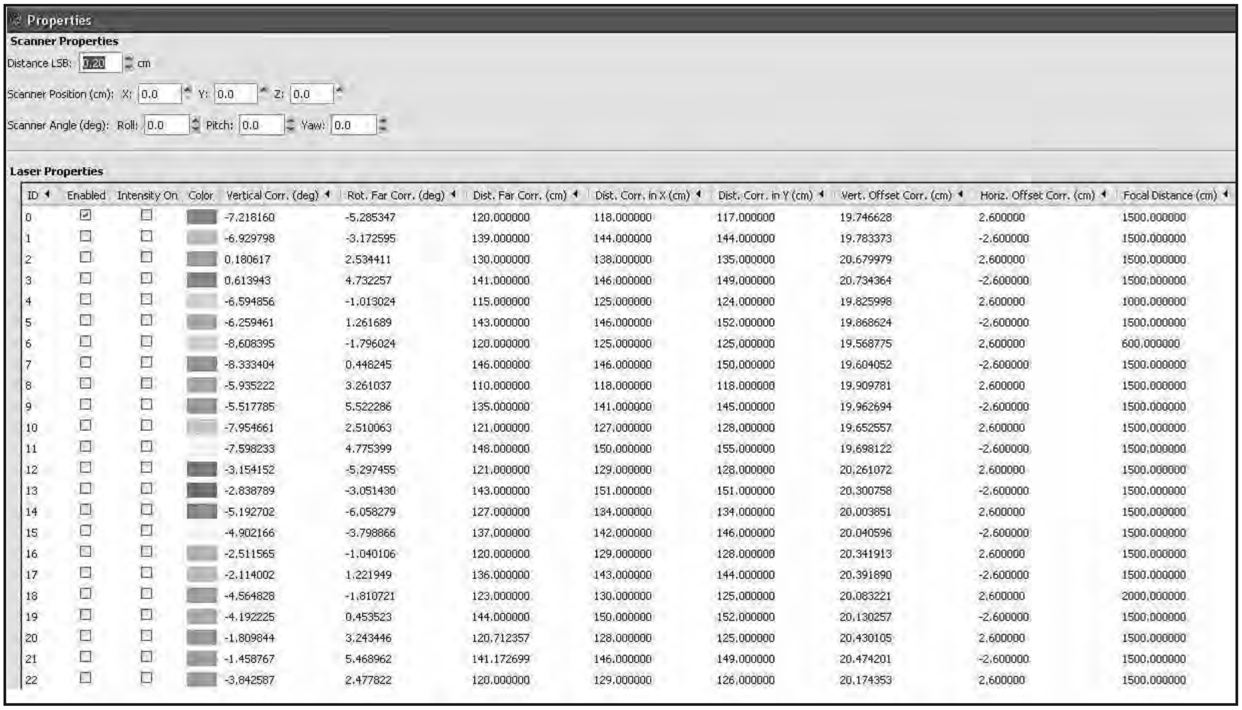
1. Use the **File > Open** command to open a previously captured pcap file for playback. The DSR playback controls are similar to any DVD/VCR control features.
2. Press the **Play** button to render the file. The **Play** button toggles to a **Pause** button when in playback mode.

PLAY button = ▷ *PAUSE button =* ⏸

Use the Forward and Reverse buttons to change the direction of playback..

FORWARD button = ⏩ *REVERSE button =* ⏪

The X, Y, Z and distance figures at the bottom of the image represent the distance of the x, y, z crosshairs with respect to the origin point indicated by the small white circle. The concentric gray circles and grid lines represent 10 meter increments from the sensor. Following is an example image of the calibration values as seen in DSR > File > Properties screen. Values are different than those on your CD..



Scanner Properties

Distance LSB: 0.20 cm

Scanner Position (cm): X: 0.0 Y: 0.0 Z: 0.0

Scanner Angle (deg): Roll: 0.0 Pitch: 0.0 Yaw: 0.0

Laser Properties

ID	Enabled	Intensity On	Color	Vertical Corr. (deg)	Rot. Far Corr. (deg)	Dist. Far Corr. (cm)	Dist. Corr. in X (cm)	Dist. Corr. in Y (cm)	Vert. Offset Corr. (cm)	Horiz. Offset Corr. (cm)	Focal Distance (cm)
0	☑	☐	▨	-7.218160	-5.285347	120.000000	118.000000	117.000000	19.746628	2.600000	1500.000000
1	☐	☐	▨	-6.929798	-3.172595	139.000000	144.000000	144.000000	19.783373	-2.600000	1500.000000
2	☐	☐	▨	0.180617	2.534411	130.000000	138.000000	135.000000	20.679979	2.600000	1500.000000
3	☐	☐	▨	0.613943	4.732257	141.000000	146.000000	149.000000	20.734364	-2.600000	1500.000000
4	☐	☐	▨	-6.594856	-1.013024	115.000000	125.000000	124.000000	19.825998	2.600000	1000.000000
5	☐	☐	▨	-6.259461	1.261689	143.000000	146.000000	152.000000	19.868624	-2.600000	1500.000000
6	☐	☐	▨	-8.608395	-1.796024	120.000000	125.000000	125.000000	19.568775	2.600000	600.000000
7	☐	☐	▨	-8.333404	0.448245	146.000000	146.000000	150.000000	19.604052	-2.600000	1500.000000
8	☐	☐	▨	-5.935222	3.261037	110.000000	118.000000	118.000000	19.909781	2.600000	1500.000000
9	☐	☐	▨	-5.517785	5.522286	135.000000	141.000000	145.000000	19.962694	-2.600000	1500.000000
10	☐	☐	▨	-7.954661	2.510063	121.000000	127.000000	128.000000	19.652557	2.600000	1500.000000
11	☐	☐		-7.598233	4.775399	148.000000	150.000000	155.000000	19.698122	-2.600000	1500.000000
12	☐	☐	▨	-3.154152	-5.297455	121.000000	129.000000	128.000000	20.261072	2.600000	1500.000000
13	☐	☐	▨	-2.838789	-3.051430	143.000000	151.000000	151.000000	20.300758	-2.600000	1500.000000
14	☐	☐	▨	-5.192702	-6.058279	127.000000	134.000000	134.000000	20.003851	2.600000	1500.000000
15	☐	☐		-4.902166	-3.798866	137.000000	142.000000	146.000000	20.040596	-2.600000	1500.000000
16	☐	☐	▨	-2.511565	-1.040106	120.000000	129.000000	128.000000	20.341913	2.600000	1500.000000
17	☐	☐	▨	-2.114002	1.221949	136.000000	143.000000	144.000000	20.391890	-2.600000	1500.000000
18	☐	☐	▨	-4.564828	-1.810721	123.000000	130.000000	125.000000	20.083221	2.600000	2000.000000
19	☐	☐	▨	-4.192225	0.453523	144.000000	150.000000	152.000000	20.130257	-2.600000	1500.000000
20	☐	☐	▨	-1.809844	3.243446	120.712357	128.000000	125.000000	20.430105	2.600000	1500.000000
21	☐	☐	▨	-1.458767	5.468962	141.172699	146.000000	149.000000	20.474201	-2.600000	1500.000000
22	☐	☐	▨	-3.842587	2.477822	120.000000	129.000000	126.000000	20.174353	2.600000	1500.000000

Figure 6. Calibration values as seen in DSR/File/Properties

DSR Key Controls

Zoom:
Z = Zoom in
Shift, Z = Zoom out

Z Axis Rotation:
Y = Rotate CW
Shift, Y = Rotate CCW

X Axis Rotation:
P = Rotate CW
Shift, P = Rotate CCW

Y Axis Rotation:
R = Rotate CW
Shift, R = Rotate CCW

Z Shift:
F = Forward
B = Back

X Shift:
L = Left
H = Right

Y Shift:
U = Up
D = Down

Aux. Functions:
Ctrl, (Z,Y,P,R,F,B,L,H,U,D) Direction = Fine Movement

Alt, (Z,Y,P,R,F,B,L,H,U,D) Direction = Very Fine Movement

DSR Mouse Controls

Rotational:
Left Button/Move

Slide:
Right Button/Move

Zoom:
Scroll forward = Zoom In
Scroll backward = Zoom Out

Matlab sample code to read calibration data from HDL-64E output.

```
fileFilter = '*.pcap';
      [File_name,Directory]=uigetfile(fileFilter,'Open a .pcap file') ;

         Filename=[Directory File_name];

         tic;
         fid=fopen(Filename);
         ttc=fread(fid,40);
         ttc=fread(fid,42);
         ttc=fread(fid,inf,'1206*uint8=>uint8',58);
    %ttch=dec2hex(ttc);
% Determine how many data packets.
         Packet=size(ttc)/1206;
         % Convert data to single precision.
         S1=single(ttc(2,:))*256+single(ttc(1,:));
         S2=single(ttc(102,:))*256+single(ttc(101,:));
         S3=single(ttc(202,:))*256+single(ttc(201,:));
         S4=single(ttc(302,:))*256+single(ttc(301,:));
             for i=0:10000  % Packets loop

             status(i+1)=(ttc(1205+i*1206));
             value(i+1)=(ttc(1206+i*1206));
             end
             a=[85 78 73 84 35]
         fclose(fid);
         toc;

         Ind=strfind(value,a);
% Loop through 64 lasers.
         for i=1:64
             temp=single(value(Ind(1)+64*(i-1)+16:Ind(1)+64*(i-1)+16+7));
             temp1=single(value(Ind(1)+64*(i-1)+32:Ind(1)+64*(i-1)+32+7));
             temp2=single(value(Ind(1)+64*(i-1)+48:Ind(1)+64*(i-1)+48+7));
             temp3=single(value(Ind(1)+64*(i-1)+64:Ind(1)+64*(i-1)+64+7));
             LaserId(i)=temp(1);
% Add high and low bytes of Vertical Correction Factor together and check if
positive or negative correction factor.
             VerticalCorr(i)=temp(3)*256+temp(2);
             if VerticalCorr(i)>32768
                 VerticalCorr(i)=VerticalCorr(i)-65536;
             End
% Scale Vertical Correction Factor by Diving by 100.
             VerticalCorr(i)=VerticalCorr(i)/100;
```

```
% Add high and low bytes of Rotational Correction Factor together and check if
positive or negative correction factor.
             RotationalCorr(i)=temp(5)*256+temp(4);
             if RotationalCorr(i)>32768
                  RotationalCorr(i)=RotationalCorr(i)-65536;
             End
% Scale Rotational Correction Factor by Diving by 100.
             RotationalCorr(i)=RotationalCorr(i)/100;
% Add high and low bytes of remaining 2 Byte Correction Factors together and
check if positive or negative correction factor, if necessary. Scale dimensions
in mm to cm by Diving by 10. Scale Focal Slope by Dividing by 10.
             DistanceCorr(i)=(temp(7)*256+temp(6))/10;
             DistanceCorrX(i)=(temp1(2)*256+temp1(1))/10;
             DistanceCorrY(i)=(temp1(4)*256+temp1(3))/10;
             VerticalOffset(i)=(temp1(6)*256+temp1(5))/10;
             HorizonOffset(i)=(temp2(1)*256+temp1(7));
             if HorizonOffset(i)>32768
                  HorizonOffset(i)=HorizonOffset(i)-65536;
             end
             HorizonOffset(i)=HorizonOffset(i)/10;
             FocalDist(i)=temp2(3)*256+temp2(2);
             if FocalDist(i)>32768
                  FocalDist(i)=FocalDist(i)-65536;
             end
             FocalDist(i)=FocalDist(i)/10;
             FocalSlope(i)=temp2(5)*256+temp2(4);
             if FocalSlope(i)>32768
                  FocalSlope(i)=FocalSlope(i)-65536;
             end
             FocalSlope(i)=FocalSlope(i)/10;
% Maximum and Minimum Intensity only 1 Byte each.
             MinIntensity(i)=temp2(6);
             MaxIntensity(i)=temp2(7);
         End
% Done with correction factors.

% Get Unit Parameter Data
       s=Ind(1)
         char(status(s-80:s+6))
         value(s-80:s+6)
```

```
        Version=dec2hex(value(s-1))
        Temperature=value(s-2)
        GPS=value(s-3)
        speed=single(value(s-48))+single(value(s-47))*256
        Fov_start=single(value(s-46))+single(value(s-45))*256
        Fov_end=single(value(s-44))+single(value(s-43))*256
        warning=value(s-13)
        power=value(s-12)
        Humidity=value(s-58)
% Done with Unit Parameters.
```

Reading Calibration and Sensor Parameter Data

Laser ID # is a 1 byte integer. Most of the output calibration data are 2 byte signed integers, except minimum and maximum intensity, which use 1 byte each. See Appendix E for more information.

Status Type	ASCII Value Interpretation and Scaling
Vertical correction	Divide by 100 for mm
Rotational angle correction	Divide by 100 for mm
Distance far correction	mm
Distance correction X	mm
Distance correction Y	mm
Vertical offset correction	mm
Horizontal offset correction	mm
Focal distance	mm
Focal slope	Divide by 10 to scale
Minimum intensity	No scaling
Maximum intensity	No scaling

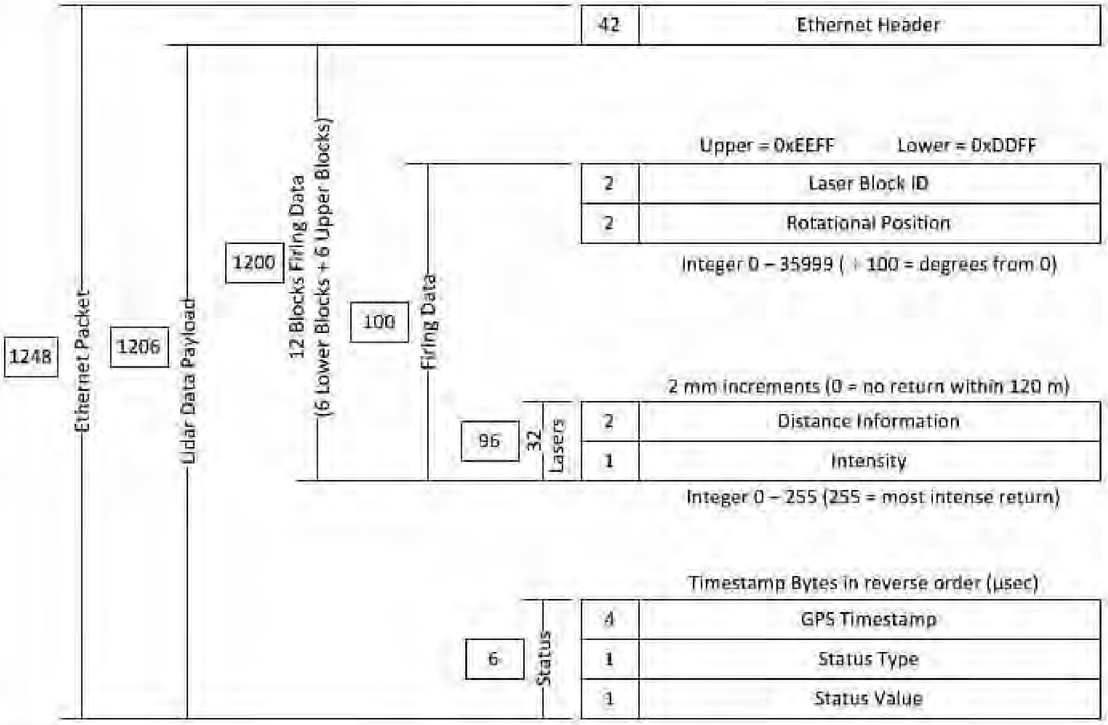
Data Packet Format

The sensor outputs UDP Ethernet packets. Each packet contains a header, a data payload of firing data and status data. Data packets are assembled with the collection of all firing data for six upper block sequences and six lower block sequences. The upper block laser distance and intensity data is collected first followed by the lower block laser data. The data packet is then combined with status and header data in a UDP packet transmitted over Ethernet. The data packet is transmitted starting with the last byte acquired in a last in, first out (LIFO) data order.

The status data always contains a GPS 4 byte timestamp representing microseconds from the top of the hour. In addition, the status data contains one type of data. The other status data rotates through a sequence of different pieces of information. See datagram on the next page.

User Datagram Protocol (UDP) Ethernet Packet Format: HDL-64E

Firmware version 4.07 (sheet 1 of 3)


42
Ethernet Header
Upper = 0xEEFF
Lower = 0xDDFF
2
Laser Block ID
2
Rotational Position
Integer 0 – 35999 (÷100 = degrees from 0)
1248
Ethernet Packet
1206
Lidar Data Payload
1200
12 Blocks Firing Data
(6 Lower Blocks + 6 Upper Blocks)
100
Firing Data
96
32 Lasers
2 mm increments (0 = no return within 120 m)
2
Distance Information
1
Intensity
Integer 0 – 255 (255 = most intense return)
Timestamp Bytes in reverse order (μsec)
6
Status
4
GPS Timestamp
1
Status Type
1
Status Value

Status Type Rotates Through				Status Value Example		
Hex	ASCII	String	Description	Hex	ASCII	
48	72	H	Hours	15	21	21 Hours (9 pm)
4D	77	M	Minutes	3B	19	19 Minutes
53	83	S	Seconds	37	55	55 Seconds
44	68	D	Date	0F	15	15th day of the Month
4E	78	N	Month	0C	12	12th Month (December)
59	89	Y	Year	0A	10	2010
47	71	G	GPS	50	86	P = GPS has sync signal
54	84	T	Temperature °C	1B	27	Unit at 27°C
56	86	V	Firmware Version	47	N/A	Version 4.07
31	49	1	Calibration & Unit Parameters (refer to sheet 2 of Datagram diagram)			
32	50	2				
33	51	3				
34	52	4				
35	53	5				
36	54	6				
37	55	7				

GPS Status Values		
Hex	ASCII	String and Definition
41	65	A = GPS receiving both sync signal and NMEA time command record
56	86	V = GPS receiving NMEA time command record only
50	80	P = GPS receiving sync signal only
00	00	0 = GPS not connected

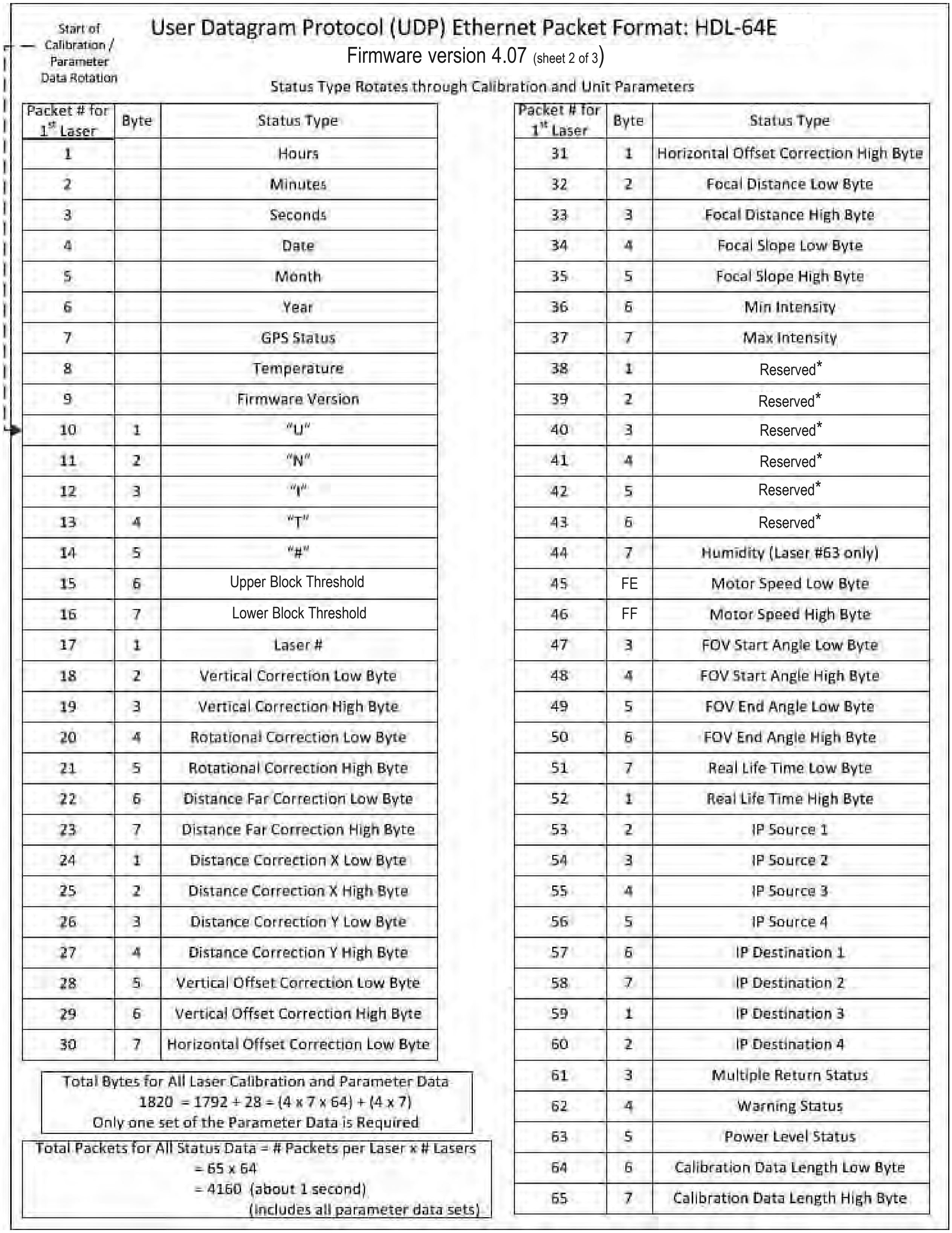
LEGEND

Bytes

32 bit unsigned integer
GPS Timestamp Example
92 18 52 D6

Reverse Order
D6 52 18 92

Convert to Decimal
3595704466 μsec
or
3595.704466 sec



Start of Calibration / Parameter Data Rotation

User Datagram Protocol (UDP) Ethernet Packet Format: HDL-64E

Firmware version 4.07 (sheet 2 of 3)

Status Type Rotates through Calibration and Unit Parameters

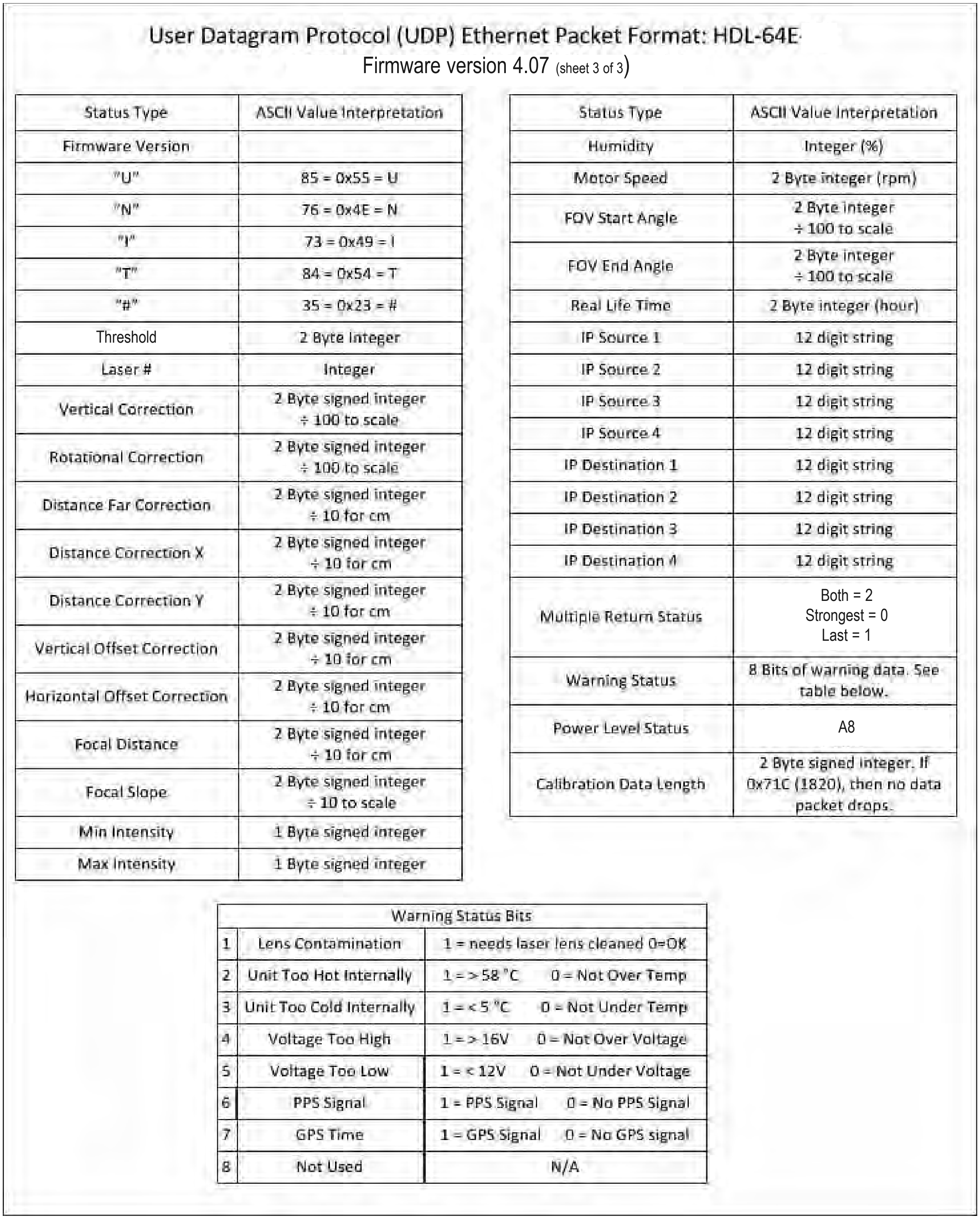
Packet # for 1st Laser	Byte	Status Type
1		Hours
2		Minutes
3		Seconds
4		Date
5		Month
6		Year
7		GPS Status
8		Temperature
9		Firmware Version
10	1	"U"
11	2	"N"
12	3	"I"
13	4	"T"
14	5	"#"
15	6	Upper Block Threshold
16	7	Lower Block Threshold
17	1	Laser #
18	2	Vertical Correction Low Byte
19	3	Vertical Correction High Byte
20	4	Rotational Correction Low Byte
21	5	Rotational Correction High Byte
22	6	Distance Far Correction Low Byte
23	7	Distance Far Correction High Byte
24	1	Distance Correction X Low Byte
25	2	Distance Correction X High Byte
26	3	Distance Correction Y Low Byte
27	4	Distance Correction Y High Byte
28	5	Vertical Offset Correction Low Byte
29	6	Vertical Offset Correction High Byte
30	7	Horizontal Offset Correction Low Byte
31	1	Horizontal Offset Correction High Byte
32	2	Focal Distance Low Byte
33	3	Focal Distance High Byte
34	4	Focal Slope Low Byte
35	5	Focal Slope High Byte
36	6	Min Intensity
37	7	Max Intensity
38	1	Reserved*
39	2	Reserved*
40	3	Reserved*
41	4	Reserved*
42	5	Reserved*
43	6	Reserved*
44	7	Humidity (Laser #63 only)
45	FE	Motor Speed Low Byte
46	FF	Motor Speed High Byte
47	3	FOV Start Angle Low Byte
48	4	FOV Start Angle High Byte
49	5	FOV End Angle Low Byte
50	6	FOV End Angle High Byte
51	7	Real Life Time Low Byte
52	1	Real Life Time High Byte
53	2	IP Source 1
54	3	IP Source 2
55	4	IP Source 3
56	5	IP Source 4
57	6	IP Destination 1
58	7	IP Destination 2
59	1	IP Destination 3
60	2	IP Destination 4
61	3	Multiple Return Status
62	4	Warning Status
63	5	Power Level Status
64	6	Calibration Data Length Low Byte
65	7	Calibration Data Length High Byte

Total Bytes for All Laser Calibration and Parameter Data
1820 = 1792 + 28 = (4 x 7 x 64) + (4 x 7)
Only one set of the Parameter Data is Required

Total Packets for All Status Data = # Packets per Laser x # Lasers
= 65 x 64
= 4160 (about 1 second)
(includes all parameter data sets)

*For Laser 63, these bytes will contain the time stamp representing when the calibration data was uploaded in the following sequence:

- Year
- Month
- Day
- Hour
- Min
- Sec.



User Datagram Protocol (UDP) Ethernet Packet Format: HDL-64E

Firmware version 4.07 (sheet 3 of 3)

Status Type	ASCII Value Interpretation
Firmware Version	
"U"	85 = 0x55 = U
"N"	76 = 0x4E = N
"I"	73 = 0x49 = I
"T"	84 = 0x54 = T
"#"	35 = 0x23 = #
Threshold	2 Byte Integer
Laser #	Integer
Vertical Correction	2 Byte signed integer ÷ 100 to scale
Rotational Correction	2 Byte signed integer ÷ 100 to scale
Distance Far Correction	2 Byte signed integer ÷ 10 for cm
Distance Correction X	2 Byte signed integer ÷ 10 for cm
Distance Correction Y	2 Byte signed integer ÷ 10 for cm
Vertical Offset Correction	2 Byte signed integer ÷ 10 for cm
Horizontal Offset Correction	2 Byte signed integer ÷ 10 for cm
Focal Distance	2 Byte signed integer ÷ 10 for cm
Focal Slope	2 Byte signed integer ÷ 10 to scale
Min Intensity	1 Byte signed integer
Max Intensity	1 Byte signed integer

Status Type	ASCII Value Interpretation
Humidity	Integer (%)
Motor Speed	2 Byte integer (rpm)
FOV Start Angle	2 Byte integer ÷ 100 to scale
FOV End Angle	2 Byte integer ÷ 100 to scale
Real Life Time	2 Byte integer (hour)
IP Source 1	12 digit string
IP Source 2	12 digit string
IP Source 3	12 digit string
IP Source 4	12 digit string
IP Destination 1	12 digit string
IP Destination 2	12 digit string
IP Destination 3	12 digit string
IP Destination 4	12 digit string
Multiple Return Status	Both = 2 Strongest = 0 Last = 1
Warning Status	8 Bits of warning data. See table below.
Power Level Status	A8
Calibration Data Length	2 Byte signed integer. If 0x71C (1820), then no data packet drops.

Warning Status Bits		
1	Lens Contamination	1 = needs laser lens cleaned 0=OK
2	Unit Too Hot Internally	1 = > 58 °C 0 = Not Over Temp
3	Unit Too Cold Internally	1 = < 5 °C 0 = Not Under Temp
4	Voltage Too High	1 = > 16V 0 = Not Over Voltage
5	Voltage Too Low	1 = < 12V 0 = Not Under Voltage
6	PPS Signal	1 = PPS Signal 0 = No PPS Signal
7	GPS Time	1 = GPS Signal 0 = No GPS signal
8	Not Used	N/A

Last Six Bytes Examples

Examples of the last row of 11 consecutive packets follows. In all cases, the "seconds" figure represents the origin of the packet expressed in seconds since the top of the hour.

PACKET #7648:

```
04d0  7a 6f 04 47 29 04 1e 59  04 70 92 18 52 d6 48 15    zo.G)..Y .p..R.H.
```

d6 52 18 92 Hex = 3595704466 (10E-6 s) = 3595.704466 seconds	48 ASCII = "H"	15 Hex= 21 21 hours

PACKET #7649:

```
04d0  80 7f 04 59 1e 04 1e 55  04 61 b2 19 52 d6 4d 3b    ...Y...U .a..R.M;
```

d6 52 19 b2 Hex = 3595704754 (10E-6 s) = 3595.704754 seconds	4d ASCII = "M"	3b Hex= 59 59 minutes

PACKET #7650:

```
04d0  85 76 04 55 17 04 1e 50  04 62 d2 1a 52 d6 53 37   .v.U...P .b..R.S7
```

d6 52 1a d2 Hex = 3595705042 (10E-6 s) = 3595.705042 seconds	53 ASCII= "S"	37 Hex= 55 55 seconds

PACKET #7651:

```
04d0  86 69 04 4f 3a 04 1e 41  04 6e f2 1b 52 d6 44 01   .i.O:..A .n..R.D.
```

d6 52 1b f2 Hex = 3595705330 (10E-6 s) = 3595.705330 seconds	44 ASCII = "D"	01 Hex= 01 01 date

PACKET #7652:

```
04d0  73 63 04 60 c3 04 1e 38  04 6c 12 1d 52 d6 4e 0c   sc.`...8 .I..R.N.
```

d6 52 1d 12 Hex = 3595705618 (10E-6 s) = 3595.705618seconds	4e ASCII = "N"	0c Hex= 12 12 month (December)

PACKET #7653:

```
04d0  44 60 04 59 46 04 1e 39  04 61 32 1e 52 d6 59 08   D`.YF..9.a2.R.Y.
```

d6 52 1e 32 Hex = 840848086 (10E-6 s) = 8408.48086 seconds	59 ASCII = "Y"	08 Hex= 08 year 2008

PACKET #7654:

```
04d0  42 63 04 5b fb 05 1e 9d  04 36 52 1f 52 d6 47 41    Bc.[.... .6R.R.GA
```

d6 52 1f 52 Hex = 3595706194 (10E-6 s) = 3595.706194seconds	47 ASCII = "G"	41 ASCII= "A" (GPS has signal)

PACKET #7655:

```
04d0  3f c2 04 2f 70 05 1e 1d  05 37 72 20 52 d6 54 1b    ?../p... .7r R.T.
```

d6 52 20 72 Hex = 3595706482 (10E-6 s) = 3595.706482 seconds	54 ASCII = "T"	1b Hex= 27^{o}C

PACKET #7656:

```
04d0  3f 47 05 33 f0 07 1e c0  05 3a 92 21 52 d6 56 47    ?G.3.... .:.!R.V7
```

d6 52 21 92 Hex = 3595706770 (10E-6 s) = 3595.706770 seconds	56 ASCII = "V"	40 = Ver 4.07

PACKET #7657:

```
04d0  3d db 05 30 51 09 1e 70  06 37 b2 22 52 d6 31 00   =..0Q..p .7."R.1.
```

d6 52 22 b2 Hex = 3595707058 (10E-6 s) = 3595.707058 seconds	31 ASCII = "1"	Not used (Spare)

PACKET #7658:

```
04d0  31 8d 06 2d 28 0b 1e 51  07 37 d2 23 52 d6 32 00   1..-(..Q .7.#R.2.
```

d6 52 23 d2 Hex = 3595707346 (10E-6 s) = 3595.707346 seconds	32 ASCII = "2"	Not used (Spare)

Dual Two Point Calibration Methodology and Code Samples

Velodyne uses a dual point calibration methodology to calculate the values in the db.xml file. This section describes this calibration methodology. The steps for the calibration are as follows:

1: Perform far point calibration at 25.04m
2: Perform near point X calibration at 2.4m
3: Perform near point Y calibration at 1.93m
4: Perform linear interpolation to get distance correction for X and Y (Nearer than 25.00m only)

The formula for the calibration value is as follows:

$$D_y = D_{1y} + (D_2 - D_{1y}) \frac{(x - 0)}{(x_2 - 0)}$$

$$D_x = D_{1x} + (D_2 - D_{1x}) \frac{(x - x_1)}{(x_2 - x_1)}$$

Where:

x_1 = 2.4 m
x_2 = 25.04 m
D_{1x} = corrected X distance for near point
D_{1y} = corrected Y distance for near point
D_{2x} = corrected X distance for far point
D_{2y} = corrected X distance for far point

Coordinate Calculation Algorithm Sample Code

```
firingData::computeCoords(guint16 laserNum, boost::shared_ptr<CalibrationDB> db,
GLpos_t &pos)
{
        guint16 idx = laserNum % VLS_LASER_PER_FIRING;
        boost::shared_ptr<CalibrationPoint> cal = db->getCalibration(laserNum);

        if (data->points[idx].distance == 0) {
                coords[idx].setX(0.0);
                coords[idx].setY(0.0);
                coords[idx].setZ(0.0);
                return;
        }
// Get measured distance, distance1
        float distance1 = db->getDistLSB() * (float)data->points[idx].distance;

// Corrected distance by distance calibration at 25.04m
        float distance = distance1+ cal->getDistCorrection();

        float cosVertAngle = cal->getCosVertCorrection();
        float sinVertAngle = cal->getSinVertCorrection();
```

```
       float cosRotCorrection = cal->getCosRotCorrection();
       float sinRotCorrection = cal->getSinRotCorrection();

       float cosRotAngle = rotCosTable[data->position]*cosRotCorrection +
rotSinTable[data->position]*sinRotCorrection;
       float sinRotAngle = rotSinTable[data->position]*cosRotCorrection -
rotCosTable[data->position]*sinRotCorrection;

       float hOffsetCorr = cal->getHorizOffsetCorrection()/VLS_DIM_SCALE;
       float vOffsetCorr = cal->getVertOffsetCorrection()/VLS_DIM_SCALE;
;  //Convert distance to X-Y plane, formular is: xyDistance = distance * cosVertAngle
       float xyDistance = distance * cosVertAngle
  // Calculate temporal X, use absolute value.
       float xx = xyDistance * sinRotAngle - hOffsetCorr * cosRotAngle + pos.getX();
  // Calculate temporal Y, use absolute value
       float yy = xyDistance * cosRotAngle + hOffsetCorr * sinRotAngle + pos.getY();
       if (xx<0) xx=-xx;
       if (yy<0) yy=-yy;
       //Get 2points calibration values,Linear interpolation to get distance
correction for X and Y, that means distance correction use different value at
different distance
       float distanceCorrX = (cal->getDistCorrection()-cal->getDistCorrectionX())*(xx-
240)/(2504-240)+cal->getDistCorrectionX();
       float distanceCorrY = (cal->getDistCorrection()-cal->getDistCorrectionY())*(yy-
193)/(2504-193)+cal->getDistCorrectionY();  //fix in V2.0
       // Unit convert: cm converts to meter
       distance1 /= VLS_DIM_SCALE;
       distanceCorrX /= VLS_DIM_SCALE;
       distanceCorrY /= VLS_DIM_SCALE;
       // Measured distance add distance correction in X.
       distance = distance1+distanceCorrX;
       xyDistance = distance * cosVertAngle; // Convert to X-Y plane
// Calculate X coordinate
       coords[idx].setX(xyDistance * sinRotAngle - hOffsetCorr * cosRotAngle +
pos.getX()/VLS_DIM_SCALE);
// Measured distance add distance correction in Y.
       distance = distance1+distanceCorrY;
       xyDistance = distance * cosVertAngle;  //Convert to X-Y plane
// Calculate Y coordinate
       coords[idx].setY(xyDistance * cosRotAngle + hOffsetCorr * sinRotAngle +
pos.getY()/VLS_DIM_SCALE);
       //Calculate Z coordinate, formula is : setZ(distance * sinVertAngle +
vOffsetCorr
       coords[idx].setZ(distance * sinVertAngle + vOffsetCorr +
pos.getZ()/VLS_DIM_SCALE);

}
```

Intensity Compensation vs Distance

Intensity compensation is done in the software for different channels by changing a parameter in the calibration window until the measurement gets to a uniform intensity for a reference target.

$$focalOffset = 256 \left(\frac{1 - focalDistance}{13100} \right)^2$$

$$intensityVal = intensityVal + K \left[focalOffset - 256 \left(\frac{1 - distance}{65535} \right)^2 \right]$$

Here K is the slope from the calibration data. Intensity gets to its maximum at the focal distance for different channels and from its calibration data.

Calibration Window

The following new intensity parameters have been added in the db.xml calibration file

- **focal distance:** At this distance, the intensity goes to max. The focal distance is different from laser to laser. On the upper block, it averages 1500cm. On the lower block, it averages 800cm.
- **focal slope:** This parameter controls intensity compensation. Min and Max Intensity are used to scale and offset intensity.

Intensity Value Corrected by Distance Code

```
for (guint i=0; i< VLS_LASER_PER_FIRING; i++) {
   guit laser = i + base;
   if (!db->getEnabled(laser))
   continue;
   bool intensity =db->getIntensity(laser);
   if (!intensity: {
   glColor3fv(db->getDisplayColor(laser).rgb);
   } else {
       guchar minIntensity = 0, maxIntensity = 0;
       float intensityScale = 0;
       minIntensity = db->getMinIntensity(laser);
       maxIntensity = db->getMaxIntensity(laser);
       //Get intensity scale
       intensityScale = (float)(maxIntensity - minIntensity);
       // Get firing “i”  intensity
       guchar intensityVal = it->getPoint(i)->intensity;
       // Get firing “i” distance, here unit is 2mm
       float  distance = it->getPoint(i)->distance;
       // Calculate offset according calibration
       float  focaloffset= 256*(1-db->getFocalDistance(laser)/13100)*(1-db-
       >getFocalDistance(laser)/13100);
       // get slope from calibration
       float   focalslope = db->getFocalSlope(laser);
       // Calculate corrected intensity vs distance
       float intensityVal1 = intensityVal + focalslope*(abs(focaloffset-256*(1-
       distance/65535)*(1-distance/65535)));
       if (intensityVal1 < minIntensity) intensityVal1=minIntensity;
       if (intensityVal1 > maxIntensity) intensityVal1=maxIntensity;
       // Scale to new intensity scale
       float intensityColor = (float)(intensityVal1 - minIntensity) / intensityScale;
       // Convert to jet color
       int   rgb=(int)(intensityColor*63);
       glColor3f(rcolor[rgb], gcolor[rgb], bcolor[rgb]);
       }
       G1Vertex3fv(it->getCoord(i).xyz;
}
it->operator++();
```

HDL-64E Ethernet Timing Table Overview

The Ethernet Timing Table shows how much time elapses between the actual capturing of a point's data event and when that point is an event output from the sensor. By registering the event of the Ethernet data capture, you can calculate back in time the exact time at which any particular distance point was captured. The formula is as follows:

Actual Event Timestamp = (Data Packet Event Output Timestamp) – (Timing Table Event Timestamp)

The upper block and lower block collect distance points simultaneously with each block issuing single laser pulses at a time. That is, each upper block laser fires in sequence and in unison to a corresponding laser from the lower block.

For example, laser 32 fires simultaneously with laser 0, laser 33 fires with laser 1, and so on.

The sensor has an equal number of upper and lower block returns. This is why when interpreting the delay table each sequential pair of data blocks represents the upper and lower block respectively, and each upper and lower block pair of data blocks in the Ethernet packet has the same delay value.

Ethernet packets are assembled until the entire 1200 bytes have been collected, representing six upper block sequences and six lower block sequences. The packet is then transmitted via a UDP packet over Ethernet, starting from the last byte acquired. See a sample of the packet format in Appendix E.

How to use this table

The table represents a sensor data packet. The laser returns come out in the order listed. Simply subtract from the timestamp of the output event of the packet each data value to arrive at the actual time the distance point was captured inside the sensor.

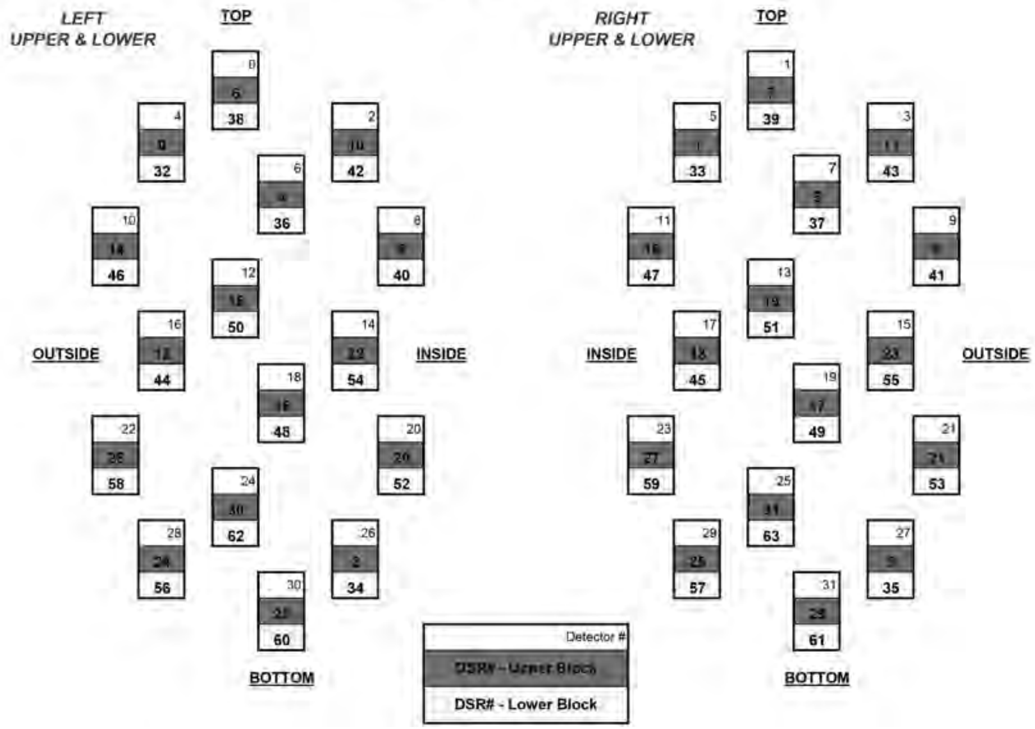
Data	Laser	Laser Numbers 0-7 & 32-39 (Lower,Upper)							
Block	**Block**	**0,32**	**1,33**	**2,34**	**3,35**	**4,36**	**5,37**	**6,38**	**7,39**
1	**Upper**	419.30	418.57	417.84	417.12	416.39	415.66	414.93	414.20
2	**Lower**	419.30	418.57	417.84	417.12	416.39	415.66	414.93	414.20
3	**Upper**	396.00	395.28	394.55	393.82	393.09	392.36	391.64	390.91
4	**Lower**	396.00	395.28	394.55	393.82	393.09	392.36	391.64	390.91
5	**Upper**	372.71	371.98	371.25	370.52	369.80	369.07	368.34	367.61
6	**Lower**	372.71	371.98	371.25	370.52	369.80	369.07	368.34	367.61
7	**Upper**	349.41	348.68	347.96	347.23	346.50	345.77	345.04	344.32
8	**Lower**	349.41	348.68	347.96	347.23	346.50	345.77	345.04	344.32
9	**Upper**	326.12	325.39	324.66	323.93	323.20	322.48	321.75	321.02
10	**Lower**	326.12	325.39	324.66	323.93	323.20	322.48	321.75	321.02
11	**Upper**	302.82	302.09	301.36	300.64	299.91	299.18	298.45	297.72
12	**Lower**	302.82	302.09	301.36	300.64	299.91	299.18	298.45	297.72

Data	Laser	Laser Numbers 8-15 & 40-47 (Lower,Upper)							
Block	**Block**	**8,40**	**9,41**	**10,42**	**11,43**	**12,44**	**13,45**	**14,46**	**15,47**
1	**Upper**	413.48	412.75	412.02	411.29	410.56	409.84	409.11	408.38
2	**Lower**	413.48	412.75	412.02	411.29	410.56	409.84	409.11	408.38
3	**Upper**	390.18	389.45	388.72	388.00	387.27	386.54	385.81	385.08
4	**Lower**	390.18	389.45	388.72	388.00	387.27	386.54	385.81	385.08
5	**Upper**	366.88	366.16	365.43	364.70	363.97	363.24	362.52	361.79
6	**Lower**	366.88	366.16	365.43	364.70	363.97	363.24	362.52	361.79
7	**Upper**	343.59	342.86	342.13	341.40	340.68	339.95	339.22	338.49
8	**Lower**	343.59	342.86	342.13	341.40	340.68	339.95	339.22	338.49
9	**Upper**	320.29	319.56	318.84	318.11	317.38	316.65	315.92	315.20
10	**Lower**	320.29	319.56	318.84	318.11	317.38	316.65	315.92	315.20
11	**Upper**	297.00	296.27	295.54	294.81	294.08	293.36	292.63	291.90
12	**Lower**	297.00	296.27	295.54	294.81	294.08	293.36	292.63	291.90

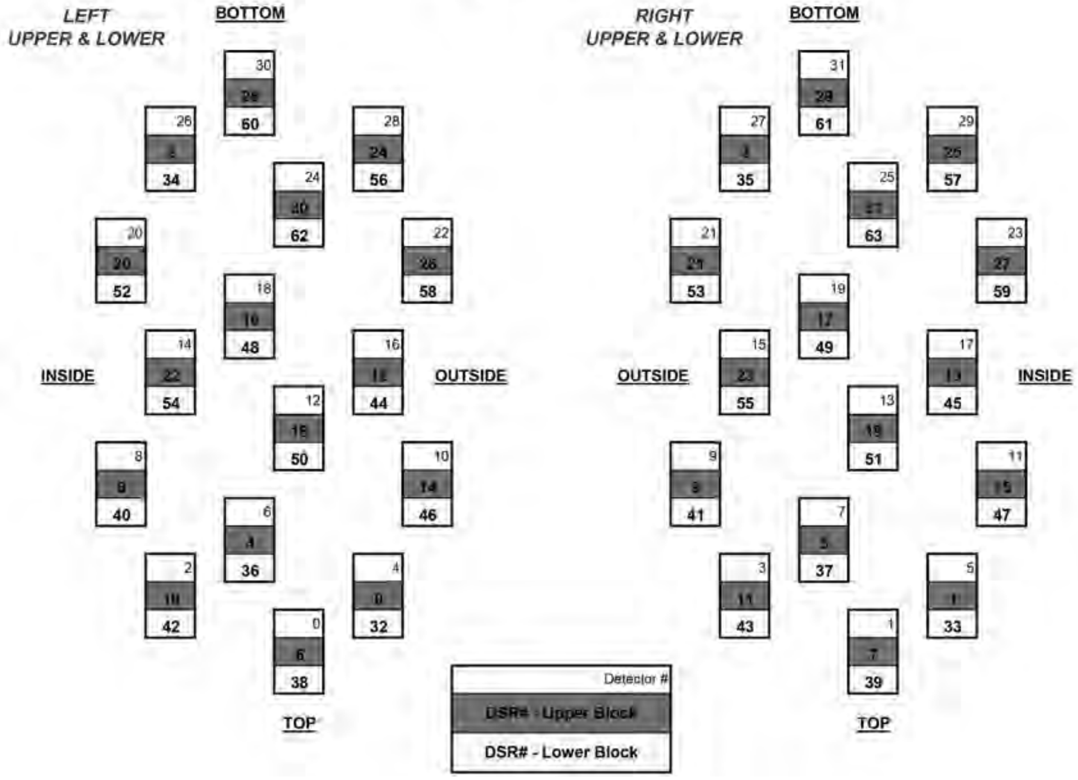
Data	Laser	Laser Numbers 16-23 & 48-55 (Lower,Upper)							
Block	**Block**	**16,48**	**17,49**	**18,50**	**19,51**	**20,52**	**21,53**	**22,54**	**23,55**
1	**Upper**	407.65	406.92	406.20	405.47	404.74	404.01	403.28	402.56
2	**Lower**	407.65	406.92	406.20	405.47	404.74	404.01	403.28	402.56
3	**Upper**	384.36	383.63	382.90	382.17	381.44	380.72	379.99	379.26
4	**Lower**	384.36	383.63	382.90	382.17	381.44	380.72	379.99	379.26
5	**Upper**	361.06	360.33	359.60	358.88	358.15	357.42	356.69	355.96
6	**Lower**	361.06	360.33	359.60	358.88	358.15	357.42	356.69	355.96
7	**Upper**	337.76	337.04	336.31	335.58	334.85	334.12	333.40	332.67
8	**Lower**	337.76	337.04	336.31	335.58	334.85	334.12	333.40	332.67
9	**Upper**	314.47	313.74	313.01	312.28	311.56	310.83	310.10	309.37
10	**Lower**	314.47	313.74	313.01	312.28	311.56	310.83	310.10	309.37
11	**Upper**	291.17	290.44	289.72	288.99	288.26	287.53	286.80	286.08
12	**Lower**	291.17	290.44	289.72	288.99	288.26	287.53	286.80	286.08

Data	Laser	Laser Numbers 24-31 & 56-63 (Lower,Upper)							
Block	**Block**	**24,56**	**25,57**	**26,58**	**27,59**	**28,60**	**29,61**	**30,62**	**31,63**
1	**Upper**	401.83	401.10	400.37	399.64	398.92	398.19	397.46	396.73
2	**Lower**	401.83	401.10	400.37	399.64	398.92	398.19	397.46	396.73
3	**Upper**	378.53	377.80	377.08	376.35	375.62	374.89	374.16	373.44
4	**Lower**	378.53	377.80	377.08	376.35	375.62	374.89	374.16	373.44
5	**Upper**	355.24	354.51	353.78	353.05	352.32	351.60	350.87	350.14
6	**Lower**	355.24	354.51	353.78	353.05	352.32	351.60	350.87	350.14
7	**Upper**	331.94	331.21	330.48	329.76	329.03	328.30	327.57	326.84
8	**Lower**	331.94	331.21	330.48	329.76	329.03	328.30	327.57	326.84
9	**Upper**	308.64	307.92	307.19	306.46	305.73	305.00	304.28	303.55
10	**Lower**	308.64	307.92	307.19	306.46	305.73	305.00	304.28	303.55
11	**Upper**	285.35	284.62	283.89	283.16	282.44	281.71	280.98	280.25
12	**Lower**	285.35	284.62	283.89	283.16	282.44	281.71	280.98	280.25

**SENSOR AS SEEN
FROM THE BACK OF THE UNIT**


LEFT
UPPER & LOWER
TOP
OUTSIDE
INSIDE
BOTTOM
RIGHT
UPPER & LOWER
TOP
INSIDE
OUTSIDE
BOTTOM
Detector #
DSR# - Upper Block
DSR# - Lower Block

**SENSOR BEAM ON THE WALL
AS SEEN ON CAMERA IN
NIGHT VISION MODE**


LEFT
UPPER & LOWER
BOTTOM
INSIDE
OUTSIDE
TOP
RIGHT
UPPER & LOWER
BOTTOM
OUTSIDE
INSIDE
TOP
Detector #
DSR# - Upper Block
DSR# - Lower Block

RPM	RPS (Hz)	Total Laser Points per Revolution	Points Per Laser per Revolution	Angular Resolution (degrees)
300	5	266,627	4167	.0864
600	10	133,333	2083	.1728
900	15	88,889	1389	.2592
1200	20	66,667	1042	.3456

Notes:

These values apply equally to the upper and lower block.

Use this chart to troubleshoot common problems with the sensor. Use this chart to troubleshoot common problems with the sensor.

Problem	**Resolution**
Unit doesn't spin	Verify power connection and polarity. Verify proper voltage – should be 16 volts drawing about 3-4 amps. Remove bottom cover and check inline 10 amp fuse. Replace if necessary. Model No. ATM-10.
Unit spins but no data	Verify Ethernet wiring. Verify packet output with another tool (e.g. Ethereal/Wireshark).
No serial communication	Verify RS-232 cable connection. Unit must be active and spinning for RS-232 update. It may take several tries for the update to be effective.

SERVICE AND MAINTENANCE

No service or maintenance requirements or procedures exist for the sensors. However, Velodyne does offer a preventative maintenance service for a fee. For service maintenance, please contact Velodyne at (408) 465-2800, of log on to our website at www.velodynelidar.com.

Sensor:	• 64 lasers/detectors • 360 degree field of view (azimuth) • 0.09 degree angular resolution (azimuth) • Vertical Field of View: S2: +2 — -8.33 @ 1/3 degree spacing -8.83 — -24.33 @ 1/2 degree spacing S2.1: +2 — -29.5 @ 1/2 degree spacing • <2cm distance accuracy (one sigma) • 5-20 Hz field of view update (user selectable) • 50 meter range for pavement (~0.10 reflectivity) • 120 meter range for cars and foliage (~0.80 reflectivity) • ~ @ 1.333 m points per second • Operating temperature -10° to 50° C • Storage temperature -10° to 80° C • Vibration: MIL-STD-810G, Method 514.6 Annex E Category 24, General Minimum Integrity Exposure (7.7Grms; 20-200Hz (0.040 g^2/Hz 20-1000Hz, -6dB/octave decline to 0.010 g^2/Hz 1000-2000Hz)
Laser:	• Class 1 - eye safe • 4 x 16 laser block assemblies • 905 nm wavelength • 5 nanosecond pulse • Adaptive power system for minimizing saturation, laser life and enhanced eye safety
Mechanical:	• 12V input (16V max) @ 4 amps • <29 lbs. [13.15 Kg] • 10" [254mm] tall cylinder of 8" [203.2mm] OD diameter • 300 RPM - 1200 RPM spin rate (user selectable) • Environmental Protection rated to IP67
Output:	• 100 MBPS UDP Ethernet packets
Dimensions (H/W/D):	• Unit: 10.13” x 8.80” x 9.10” [257.3mm x 223.5mm x 231.1mm] • Crate: 19.5” x 18” x 18” [495.3mm x 457.2mm x 457.2mm]
Shipping Weight: (approx.)	• 68 lbs [30.8 Kg]

Velodyne LiDAR, Inc.
345 Digital Drive
Morgan Hill, CA 95037

408.465.2800 voice
408.779.9227 fax
408.779.9208 service fax

www.velodynelidar.com

Service Email: lidarservice@velodyne.com
Product Email: help@velodyne.com
Technical Email: lidarhelp@velodyne.com
Sales Email: lidar@velodyne.com

All Velodyne products are made in the U.S.A.

Specifications subject to change without notice
Other trademarks or registered trademarks are property of their respective owner.