Parakeet Pro Lidar

User Manual and Specifications

Revision History

Date	Description
November 10, 2022	Original Revision
December 2, 2022	Updated Layout, Added IAP Sections
February 28, 2023	Updated Messages Added Time Sync Messages
March 27, 2023	Added Block Settings Request Messages
April 5, 2023	Updated Data Packet Checksum
April 24, 2023	Beam Divergence Spec, Data Packet Flags

Table of Contents

Contents

Overview and Specifications

The Parakeet ProE lidar is a high performance, two-dimensional (2D) laser scanner for OEM clients seeking to optimize size, performance, and cost. The Parakeet ProE works via TOF (time-of-flight) pulsed laser technology. Laser pulses are diffusely reflected off the target and the return time is measured. The results are the angle, distance and intensity values of the target for each return.
Two variations of the sensor are available the Parakeet Pro uses RS-232 serial communications and the Parakeet ProE uses an IPv4 connection over Ethernet. Both Parakeet Pros can transmit up to 18,000 data points per second, with an angular resolution of between 0.20° and 0.30° [based on scan rate of 10Hz (600RPM) or 15 Hz (900RPM)], at a maximum range of 40 meters at a 360° radius in a 2D plane.

The Parakeet ProE uses a near-infrared pulsed laser. As such, safety to humans is ensured with FDA Class 1 laser safety standards. The Parakeet ProE by itself does not meet any safety standards for use cases requiring a type 3, safety device to IEC 61496. The Parakeet ProE is designed for long lasting performance with its magnetic inductive coupling transmission of data and power. The sensor can be used in both indoor and outdoor applications up to 80,000 lux.

Application Scenarios

The Parakeet ProE can be used in the following application scenarios:

• General robot navigation and localization
• Environment scanning and 3D modeling
• Service robot or industrial robot
• Home service / cleaning robot navigation and localization
• General simultaneous localization and mapping (SLAM)
• Smart toy localization and obstacle avoidance

Sensor Physical Arrangement

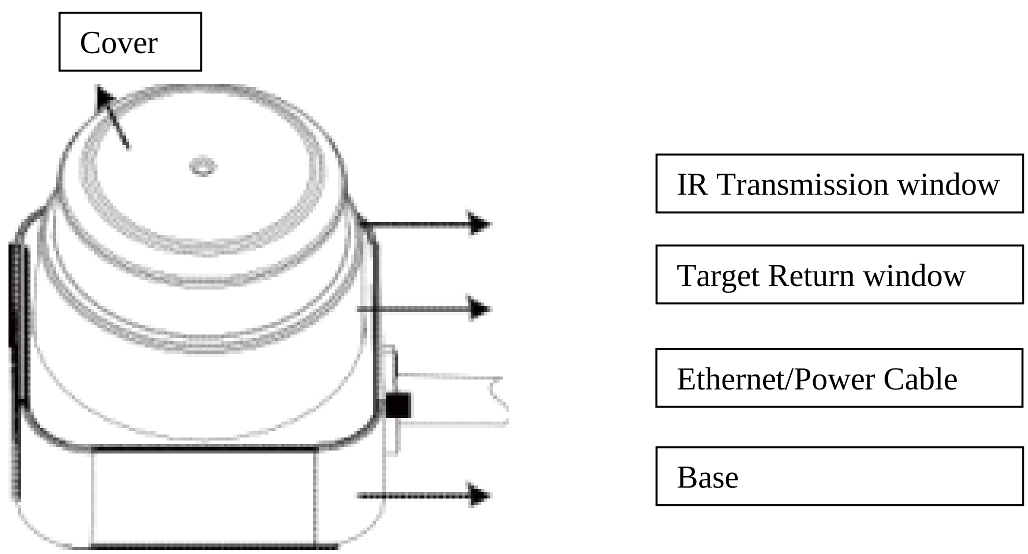


FIG. 1-1 General Diagram of Parakeet ProE system

Sensor Dimensions

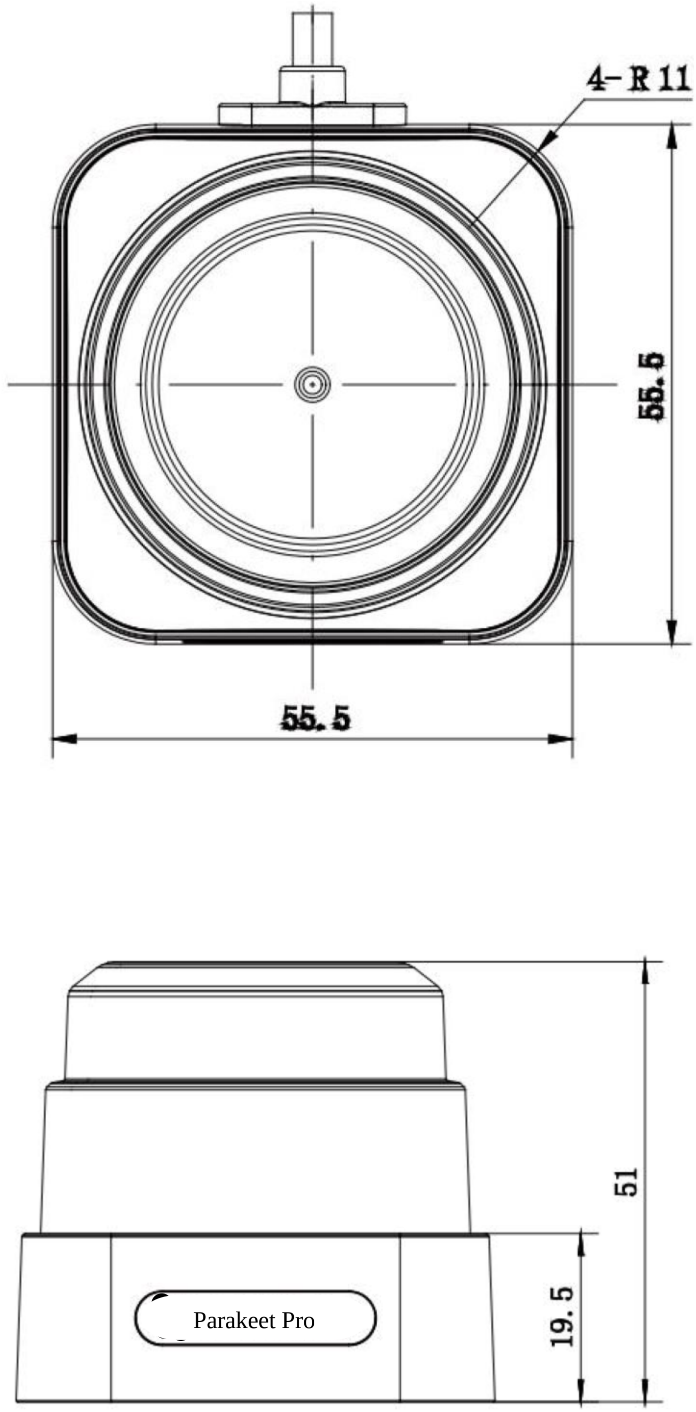


FIG. 1-2 Sensor dimensions of Parakeet ProE. Units (mm)

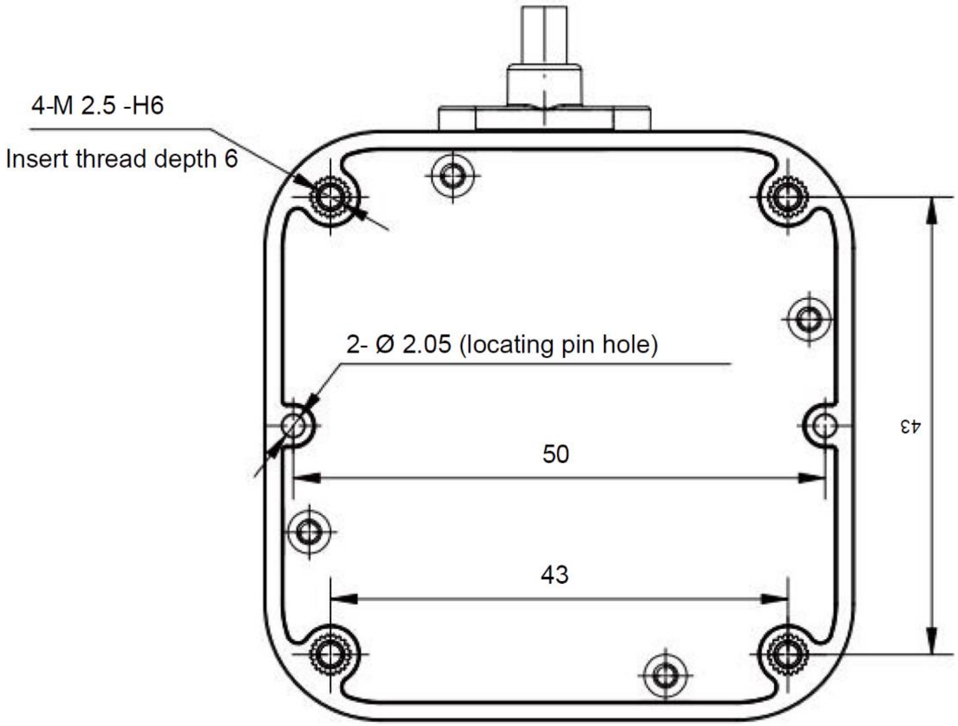


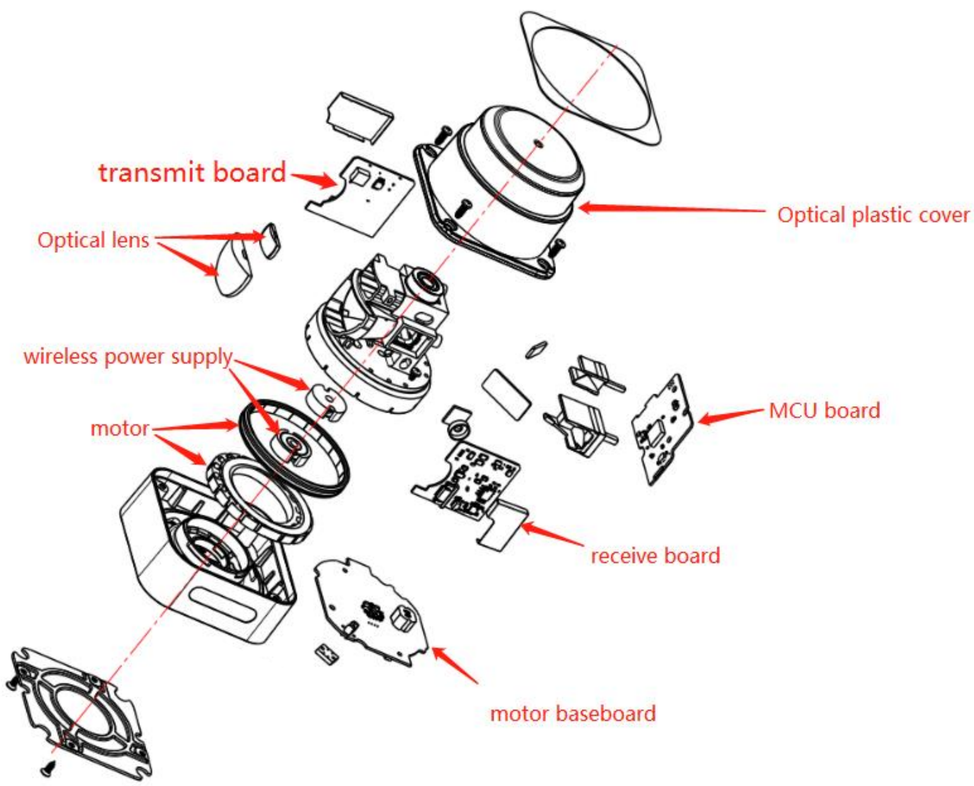
FIG. 1-3 Sensor mounting dimensions of Parakeet ProE. Units (mm)

Working principles

The Parakeet ProE uses pulsed laser Time Of Flight (TOF) ranging technology. With high-precision time measurement and processing circuit and method, it can range up to 18,000 times per second. Each ranging data is sent to the power supply processing module through high-speed optical communication for calculation, and the distance between the irradiated target object and radar as well as the current included Angle information will be output from the communication interface.

Driven by the motor, the ranging module of the Parakeet ProE will rotate clockwise so that the surrounding environment will be fully scanned by 360 degree.

The Parakeet ProE lidar uses a near-infrared pulse laser as a light source whose pulses are emitted only in nanoseconds to reach the safety standard of the FDA Class 1 so that the safety of humans and pets can be ensured. The near-infrared pulse laser combined with the optical filter can effectively avoid light interference, so it can be used in indoor and outdoor environment.



Optical Window

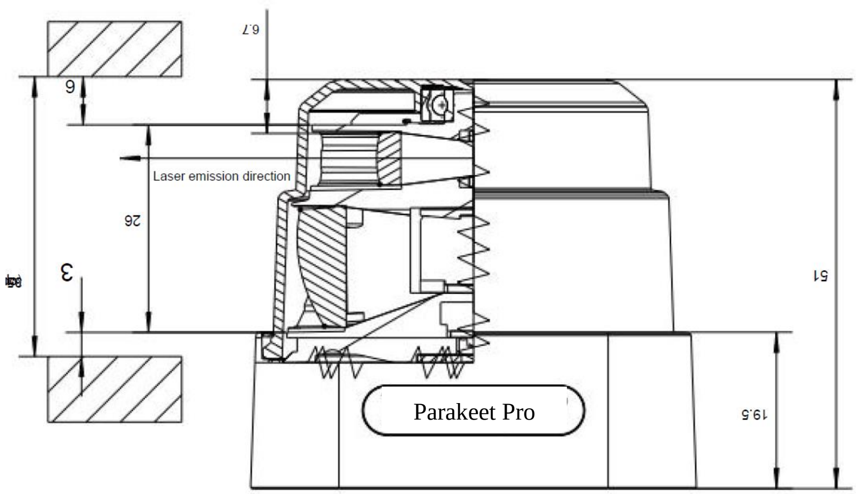
It is important to ensure that an adequate window for the laser transmit and receive, is incorporated when mounting the Parakeet ProE in your product. The ranging performance and precision will be affected by partial occlusion of the optical path.

It is better that this window be an air gap versus a transparent window. A transparent window may impact the performance of the Parakeet ProE by attenuating the laser beam or distorting it by acting as a lens.

If you need to use a transparent cover to protect the sensor, please contact Lidar Engineering to help design a window with minimal impact on the performance of the sensor.

Refer to the mechanical dimensions below for the specific optical window dimensions.

Sensor Optical Window Dimensions



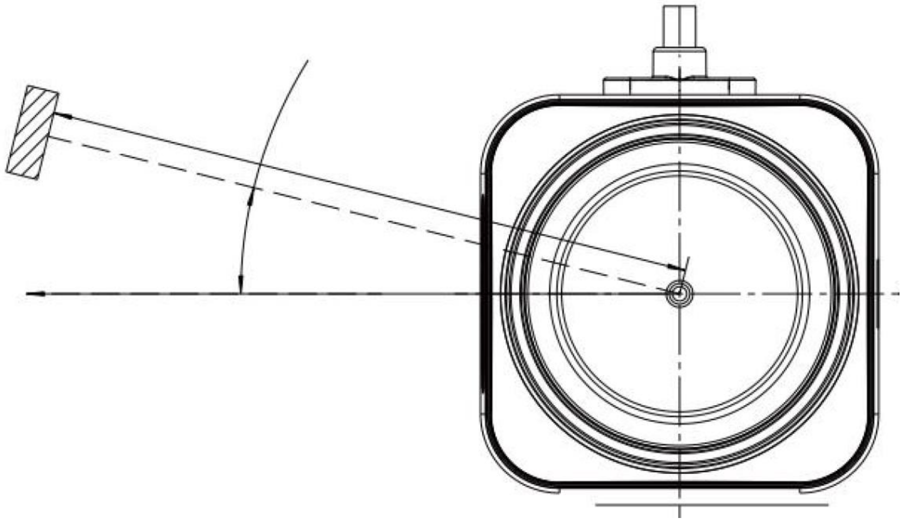


FIG. 1-4 Mechanical Dimensions of Optical Window Units (mm)

Sensor Specifications

Model	LDS-50C-C30E
Laser wavelength	905nm±15nm
Ranging distance	0.1-40m @ 90% reflectivity
	0.1-12m @ 10% reflectivity
Measurement frequency	18kHz
Scan rate	10Hz, 15Hz
Echo strength	0-254
Drive method/motor	Built-in brushless motor
Horizontal field of view	360°
Horizontal angular resolution	0.2°@10Hz
	0.3°@15Hz
Laser Beam Size @ 1000mm	9.0mm Vertical x 14.8mm Horizontal
Beam Divergence	2.5 mRadians Vertical x 0.8 mRadians Horizontal
Ranging accuracy	±30mm
Distance Resolution	mm
Interface	Ethernet
Optical Interference Resistance	>80000LX
Dimensions	55x55x51mm
Reference Weight	110g (not including wires)
Operating Voltage	10VDC-26VDC
Power Consumption	≤2.2W
Storage temperature (℃)	-20℃ ~ 65℃
Operating temperature	-10℃+50℃
degree of protection	IP5X

TBL. 1-1 Specifications of Parakeet ProE system

Power Interface

Power interface: 10V ~ 26V DC 5.5mm barrel connector. Contact Lidar Engineering for custom power interface connector options. The Parakeet Pro can operate on 5V, from a typical USB interface, but the sensor will report low power warnings and the full performance operating specifications are not guaranteed. This this may be useful for software development testing when accurate ranging data is not required.

Communication Interface

Parakeet ProE uses a standard RJ45 using 586B wiring standard. The network interface configured to communicate at 100BT using IPV4 and UPD protocols.
Parakeet Pro uses a serial interface through a USB connection. A USB to serial driver needs to be installed to make the connection appear as a serial comm port within the PC.

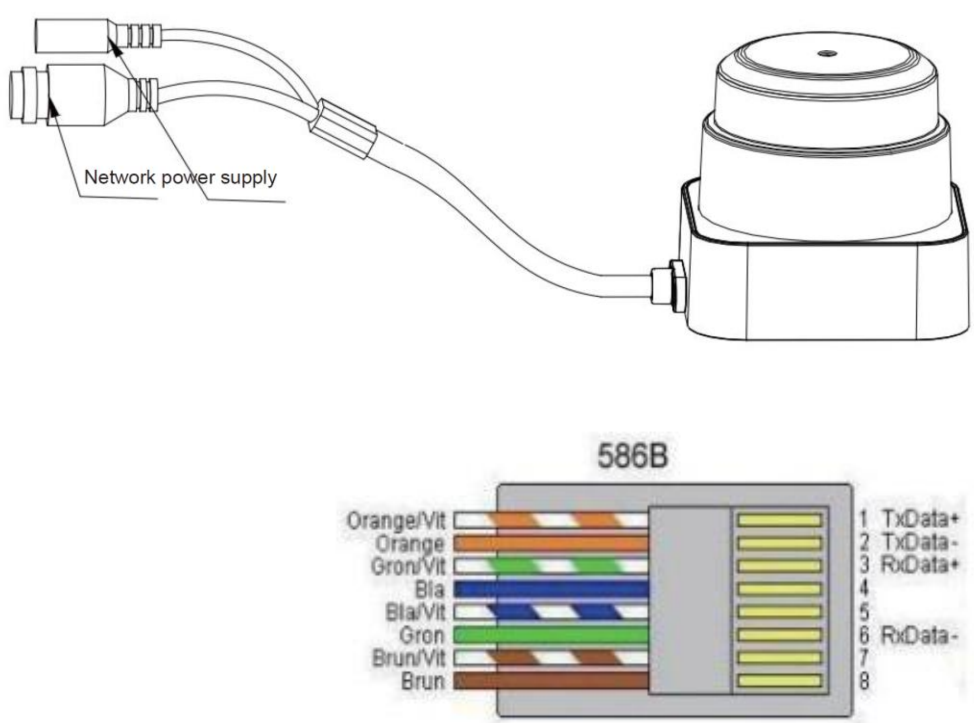


FIG. 1-5 Sensor Cables and Connectors

Development Tools and Support

To facilitate user's evaluation and development with the Parakeet Pro and ProE LiDAR for product development and acceleration of the development cycle, a graphical debugging interface terminal and SDK development kit that can run on platforms such as x86 windows, x86 and ARM Linux are available. If you have any questions, please contact Lidar Engineering.
There is also a command line utility for flashing new firmware into the sensor that may be more convenient than using the Graphical Evaluation tool described below.

Graphical Evaluation and Debug Software

This program provides a quick way to setup and evaluate the Parakeet Pro LiDAR sensor. This is an MS Windows program and it has been testing working with Win 10 and 11. It likely works with Win 8. Windows by default installs a firewall to prevent unwanted intrusion into your computer from the network. When first running this program the Windows firewall will pop up a message asking if you want to allow this program to use the network. Click the response to allow the program to access the network.
This program consists of a single executable file. There is no complicated install procedure. Simply move the program file into a folder on your computer. To start the program open Windows Explorer, browse to the program's location and double click on it. You may use Windows to create a shortcut or pin it to your task bar to make it more convenient.

Graphical Evaluation Program Operation

Connect the Parakeet Pro LiDAR to power and Ethernet connection of your computer. Navigate to the Parakeet Test Program and double click the icon to start the program. You will see a window similar to the one shown below.

Main Program Window Description

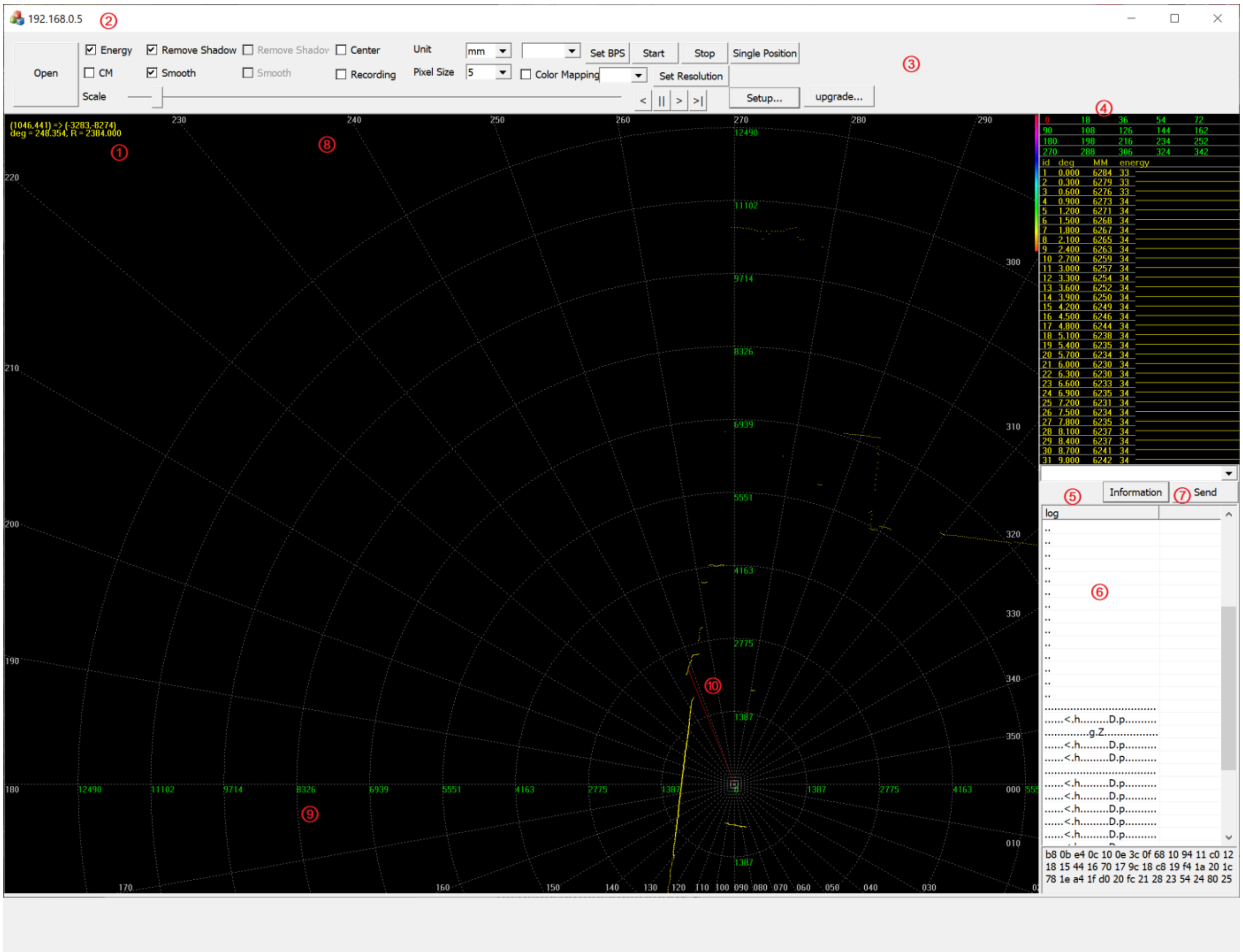


FIG 2.1 Main Program Window Areas

① ***Graphical Display Area*** – Black window that will display the scanner data.

② ***Title Bar*** – Displays the IP address of the currently connected LiDAR.

③ ***Main User Controls*** – This group of buttons and check boxes allow you to change the configuration of the sensor and control the graphical display.

④ ***Numeric Scan Data*** – Displays the raw scan data in numerical format.

⑤ ***Information Button*** – Queries the sensor for its software and hardware version information.

⑥ ***Message Display Window*** – Displays Status and Heartbeat messages from the sensor.

⑦ ***Send Button*** – Future Feature. Send a manually entered command to the sensor.

⑧ ***Angle Labels*** – The white numbers around the exterior of the graphical display window indicate the angle from the origin of the sensor. The zero angle of the sensor is in the center of the side of the sensor opposite the side where the cable exits the sensor.

⑨ ***Range Labels*** – The green numbers along the X and Y axis of the graphical display indicate the range from the center of the sensor. The units of these numbers are selected in the main user controls. See below.

⑩ ***Mouse Radial Line*** – This red line indicates the angle from the center of the sensor to the data point selected by the position of the mouse pointer.

Main User Controls

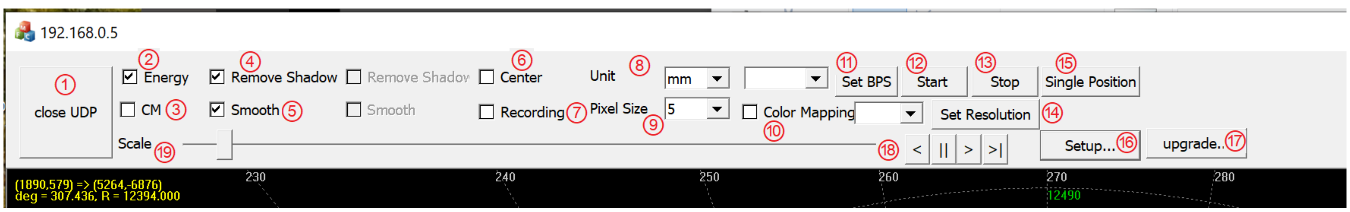
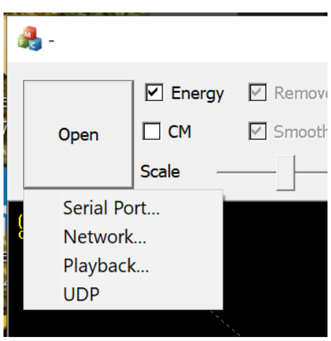


FIG 2.2 Main User Controls Area

① ***Open*** – click this button to choose a connected sensor to display in the Graphical Display Area. Select the type of sensor or playback data file to display. See details of each option in the next section below.



②***Energy*** – Future feature, currently unsupported. Sensor returns response “unsupport”.

③***CM*** – Future feature, currently unsupported. Sensor returns response “unsupport”.

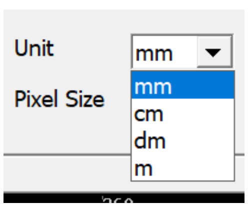
④***Remove Shadow*** – Select this check box to enable or disable shadow removal filter. This filter removes phantom data points where the scanning beam is transitioning between close and far targets. This is caused by the size of the laser beam illuminating both targets at the same time.

⑤***Smooth*** – Select this check box to enable or disable a filter to smooth the variation between adjacent scan points. This can best be seen when scanning a large flat object such as a wall. This smoothing may help an algorithm to better detect a flat target.

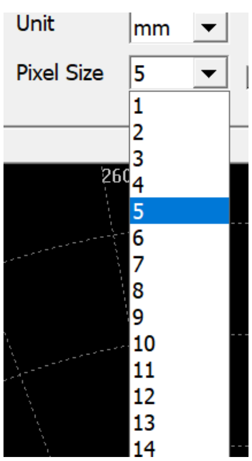
⑥***Center*** – Select this check box to lock the origin of the sensor to the center of the display. Clicking and dragging the mouse will no longer pan the display.

⑦***Recording*** – Select this check box to begin capturing the scan data from the sensor for later playback. The captured data files are stored in the local directory where this program is located. There are two files created, named: 20221018-134441.dump and 20221018-134441.udpp. These files contain the same data and either can be used when you want to play back the data. The filenames are created based on the Date and Time that the data recording was initiated. Press the stop button to end the recording.

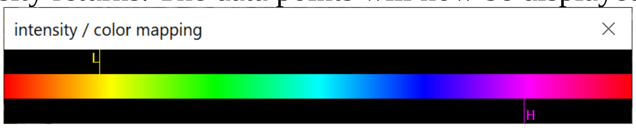
⑧***Unit*** – Use this drop down box to select the units to display the range data. Millimeters (mm) is the default setting.



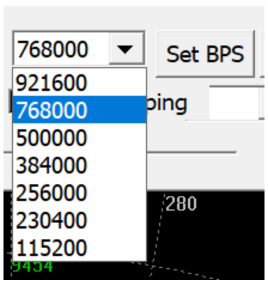
⑨***Pixel Size*** – Use this drop down box to select the size of the range data point on the graphical display.



⑩ ***Color Mapping*** – Select this check box to display the intensity / color mapping tool. Clicking above the color bar will set the color for low intensity returns. Clicking above the bar will set the color for high intensity returns. The data points will now be displayed with the colors in the range defined.



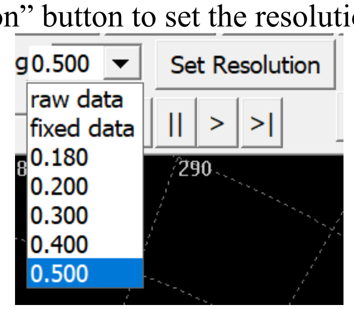
⑪ ***Set BPS*** – Use the drop down box to select the baud rate for a serial sensor. Select the button to send the command to the sensor. Parakeet ProE ethernet sensors will respond “unsupport[ed]”



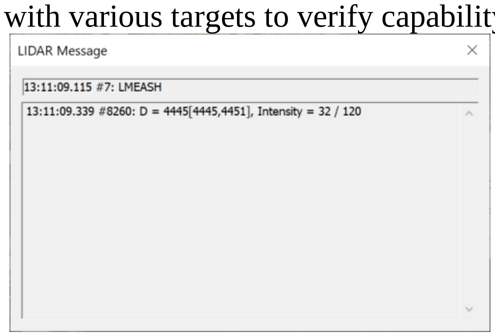
⑫ ***Start*** – Select this button to cause the sensor to start scanning and sending data

⑬ ***Stop*** – Select this button to cause the sensor to stop scanning.

⑭ ***Set Resolution*** – Use the drop-down box to select the desired resolution, then select the “Set Resolution” button to set the resolution in the sensor.

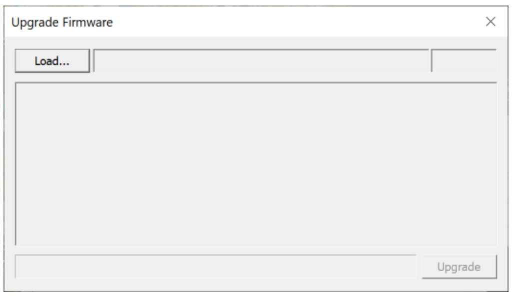


⑮ ***Single Position*** – Select this button to stop the motor within the sensor, but continue to make range measurements and send the data to the program. The scanner will stop at a random position and the distance and intensity data will be displayed in a pop up window. This can be useful for verifying operation with various targets to verify capability of the sensor.



⑯ ***Setup…*** – Select this button to open a dialog box to adjust the various settings of the sensor. See the detailed description in the Setup section below.

⑰ ***upgrade***… – Select this button to download updated firmware to the sensor. The box shown below opens, then click the "Load" button. This brings up a standard Windows dialog box to select the Firmware upgrade file for loading into the sensor. Click OK, then at this box click, Upgrade to start the update process. The software uses the IAP interface to send the firmware upgrade file to the connected sensor.
A standalone command line utility is available to perform the upgrade task also. See the IAP section of this manual for more information.

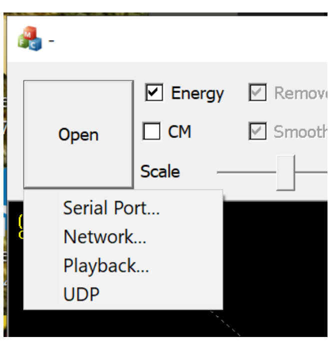


⑱ ***Playback Controls*** – Use these buttons to step back, pause/run, step forward and step to end of the playback file. The playback data file is selected using the Open-Playback selection.

⑲ ***Scale*** – Use this slider to set the zoom scale of the graphic display window. You can also use the scroll wheel on the mouse while in the graphic window to change the scale.

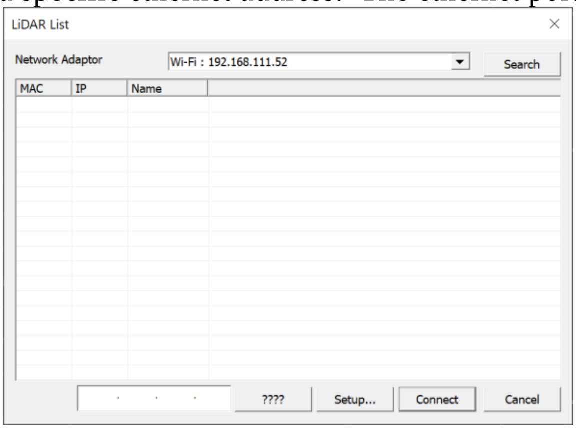
Open Button Options

For most testing and development tasks with the Parakeet ProE you will start by connecting to a sensor. Clicking the Open button will bring up four options described below.



①***Serial Port…*** - Use this option to connect to a sensor that uses a serial port (Parakeet Pro). The dialog box will display the available serial ports. Using the drop down boxes, select the port and the baud rate currently configured in the sensor. Select the OK button to connect the serial port sensor.

②***Network***… - Opens a dialog box that allows you to create a TCP connection to a Parakeet ProE sensor on a specific ethernet address. The ethernet port used is 6668.



③***Playback…*** - Opens standard windows dialog box to select a playback data file. Select the file and click the Open button to load the file and display its contents on the graphical data display. The files are created using the Recording button described earlier. The files are named: 20221018-134441.dump and 20221018-134441.udpp, named using the appropriate Date and Time information from when the recording was started. Both files contain the same data and either can be selected for playback. A set of buttons provide the ability to step forward backward, pause and start the playback of the data.

④***UDP…*** - Opens a dialog box to select a network adapter and sensor. After selecting the network adapter available sensors will be displayed. This is the most common method to connect to Parakeet ProE sensors for configuration and testing.

UDP Connection Details

The most common method for connecting to a Parakeet ProE sensor will be with the UDP networking protocol. Click the Open button and Select the UPD… option. The following dialog box will be displayed.

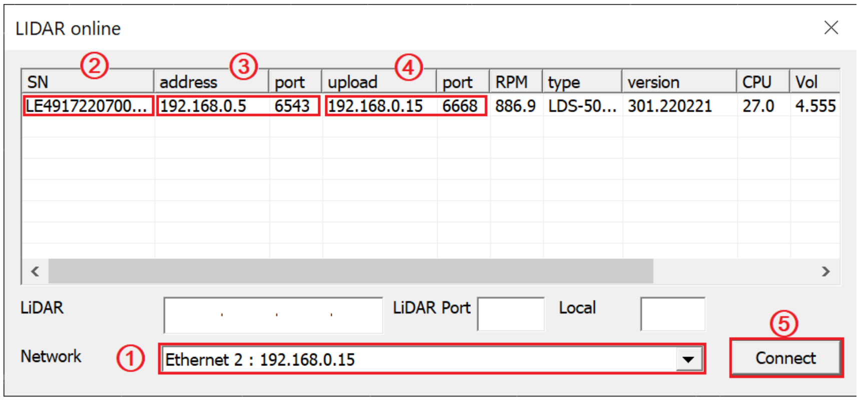


FIG 2.3 UDP Network Connection Dialog

① ***Network*** - Use this drop down box select the network adapter that is connected to the LiDAR sensor you want to use. Available sensors and their network settings will be displayed. Select the sensor you want to use, then click the Connect button⑤.

② ***SN*** – This column displays the serial numbers of the sensors that are connected to the current interface.

③ ***Address & Port*** – This column displays the IPV4 address of the sensor and the port it uses to send the UDP data.

④ ***Upload & Port*** – This column displays the IPV4 address and port for where the sensor will send its data when it is connected, powered up, and started.

You can have multiple sensors connected in your network, each with its own IP address, but you can only work with one at a time with this visualization and test software.

Sensor Setup Dialog Box

If you need to change any network settings or options of the Parakeet, click the ***"Setup..."*** button in the Main User Controls area. This will bring up the LiDAR Setup dialog box shown here.

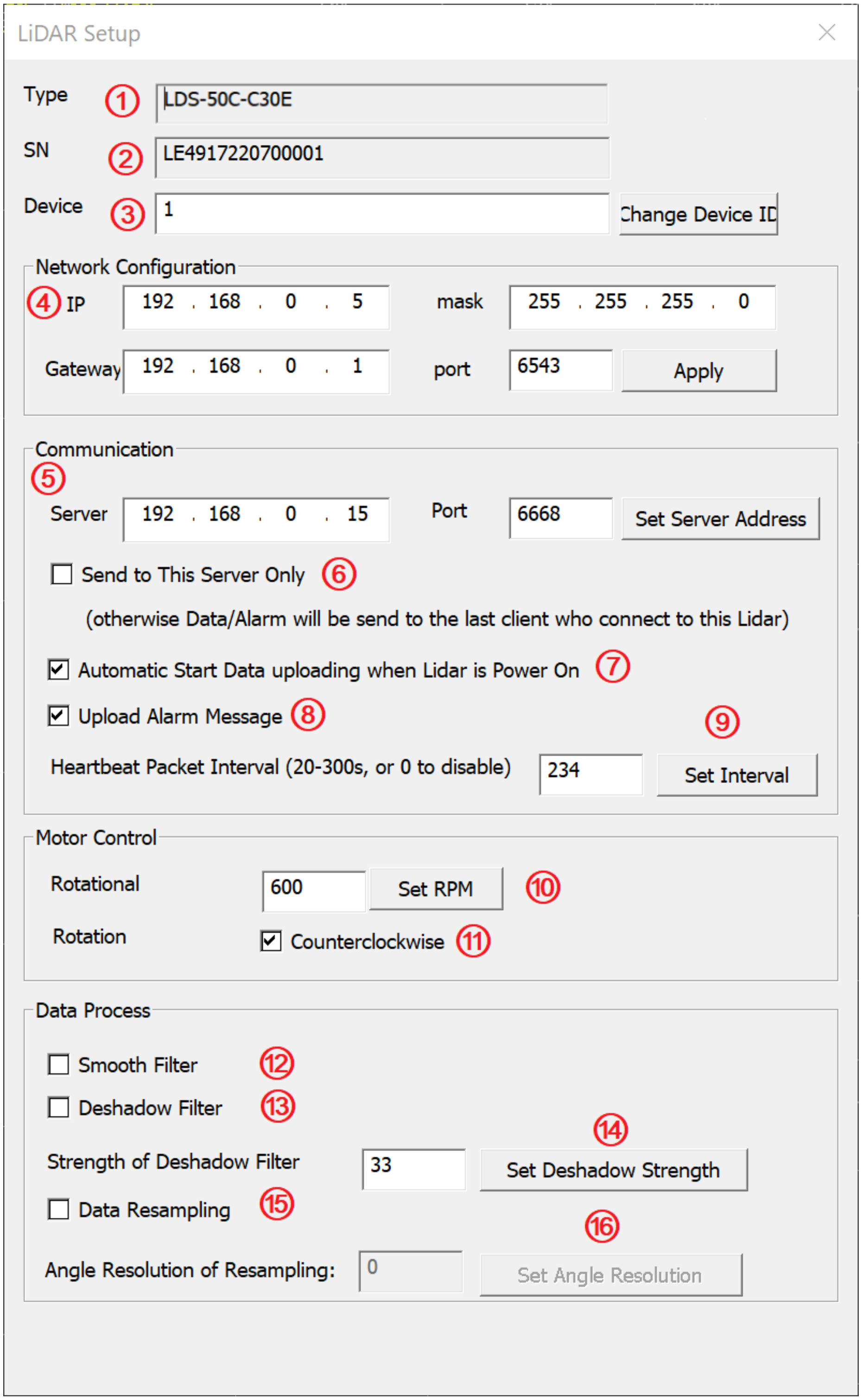


FIG 2.4 Sensor Settings Dialog Box

① ***Type:*** is the Parakeet model number.

② ***SN:*** is the factory serial number.

③ ***Device ID:*** customers can set the number according to their own needs, can set the range of 0~2147483647(D) (7FFFFFFF(H))

④ ***Network Configuration:*** is the IP address, subnet mask, gateway, and source Port used by the lidar.

⑤ ***Server Address:*** is the IP address and Port of the data upload server target. You can set the IP address and port number according to your needs. The device IP and the target IP should be in the same subnet.

⑥ ***Send to This Server Only:*** If enabled, the sensor will send the data to the Server IP address. If it is disabled, the software will send data packets to the IP broadcast address, and will change to sending data to the address from which it receives a command.

⑦ ***Automatic Start:*** If this is enabled, (checked) the sensor will startup and data will be uploaded continuously after the lidar is powered on. If disabled, the sensor will remain idle when it is powered on. A start command will need to be send to the lidar when data is required.

⑧ ***Upload Alarm Message:*** Enable or Disable sending of alarm packets from the sensor.

⑨ ***Heartbeat Packet Interval:*** Value can be 20~300 seconds or 0 to disable. When any message from the host computer is sent to the Lidar the Lidar will restart its watch dog timeout interval with this value. If no messages are received by the Lidar from the host computer in this time interval, the Lidar will restart it self in an attempt to clear any internal communication errors.

⑩ ***Rotational Speed:*** Value can be 600 or 900 RPM.

⑪ ***Rotation Counter Clockwise:*** Future enhancement, currently unavailable.

⑫ ***Smooth Filter:*** When the filtering function is enabled, the linear fluctuation of lidar data will be reduced and the linearity will be smoother.

⑬ ***Deshadow Filter:*** When enabled, this will cause the sensor to filter the data to modify readings at sudden transitions of distance to remove intermediate ‘drag’ readings.

⑭ ***Deshadow Filter Strength:*** Filters the sending of alarm packets from the sensor. Value 1~200

⑮ ***Data Resampling:*** Enable or Disable Resampling

⑯ ***Angle Resolution of Resampling:*** Value 0 = Disable; 1 = Enable; 43 = 0.043 ~ 999 = 0.999

Sensor Communication

The Parakeet ProE uses a set of messages that are placed in a packet format and then sent via UDP to and from the sensor. The visualization and testing software, described above, implements many of these messages. The visualization software also displays the content of each message and sensor response in a pop-up dialog whenever an option is changed in the program.

For building a custom software control program to work with the Parakeet ProE the following sections provide the details of the various messages and packages. Using the visualization testing software to explore examples of how the messages are assembled should be useful in helping to understand the tables below.

A packet sniffing tool such as "WireShark" will be useful to build understanding of how the messages are built into packets, and how the packets are sent via UDP over ethernet.

While UDP does not require IP level acknowledgement like the TCP protocol, the Parakeet ProE implements a simple response to the command and configuration messages. Implementing your control software to verify the responses from the sensor can provide additional robustness for your application.

Control Word Messages

The Parakeet ProE has a set of control messages that can be sent via the UPD protocol on the ethernet connection. These commands are simple actions for the sensor to perform immediately such as start scanning; stop scanning; or restart. There are also Set Word messages to adjust configuration settings of the sensor, these are described in the next section.

All messages between the controlling computer and the sensor are formatted similarly and can be sent directly over serial communications or packaged in a UDP packet for transmission to the sensor. These Control Word Messages use 0x0043 as the message identifier

Control Word UDP Packet Format

All messages between the controlling computer and the sensor are formatted similarly and packaged in a UDP packet for transmission to the sensor. This packet format is described here. There are separate packet definitions for messages to the sensor and the sensor's response packets. (Serial communication Parakeet sensors use the same messages, but do not use the UDP packet format)

Control Word Message Format			
A message which is sent to the sensor in order to configure the device. This command packet is embedded in a UDP packet for transmission. The sensor responds with a response message in a UDP packet.			
Field #	**Name**	**Length**	**Interpretation**
1	signature	2	Fixed as "0x484C"
2	identifier	2	Tells the sensor which category of command this packet describes. Control Word Messages use: 0x0043
3	sequenceNumber	2	A unique (random) identifier for this packet. The response packet which is returned for this message, will use the same sequence number
4	length	2	The number of characters in field 5 message Typically 6.
5	message	varies	A Control Word message or a Setting Word message. The Command Set section has information describing this
6	crc32	4	A CRC32 checksum of fields 1-5. Initial value: 0xFFFFFFFF Polynomial value: 0x04C11DB7 Code example maybe found here

Control Word Response Message Format			
The response message packet is sent to the host in response to the sensor receiving a control word message. This packet is embedded in a UDP frame and sent via ethernet.			
Field #	**Name**	**Length**	**Interpretation**
1	signature	2	Fixed as “0x484C”
2	identifier	2	Fixed as “0xFFBC”
3	sequenceNumber	2	Contains the sequence number matching the control packet the device is responding to
4	length	2	The number of characters in field 5
5	message	varies	The message response from the sensor. More information on this can be found in each control word section.
6	crc32	4	A CRC32 checksum of fields 1-5. Initial value: 0xFFFFFFFF Polynomial value: 0x04C11DB7 Code example maybe found here

Each of the following tables define the specific details of each control word message. Placing the contents of each message within the Parakeet ProE packet and sending to the sensor using UDP will cause that command to be executed. Most commands cause the sensor to generate a response message indicating that the message was received and understood.

LSTARH

Start Sensor Spinning			
Tell the sensor to start spinning. This starts the transmission of lidar data packets.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LSTARH”
Device response value:	“OK”		

LSTOPH

Stop Sensor Spinning			
Tell the sensor to stop spinning. This stops the transmission of lidar data packets but does not stop heartbeat packet transmission.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LSTOPH”
Device response value:	“OK”		

LMEASH

Measure Single Point			
Stop the rotation of the sensor and take multiple measurements at one point			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LMEASH”
Device response value:	“OK”		

LRESTH

Reset and Reboot Sensor			
Tell the sensor to reboot and reset any non-persistent setting changes			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LRESTH”
Device response value:	No Response sensor restarts immediately		

LVERSH

Get Sensor Version			
Sensor replies with the version number of the sensor			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LVERSH”
Device response value:	“MCU VERSION:V#.#” “MOTOR VERSION:V#.#”		

LUUIDH

Get Sensor Serial Number			
Sensor replies with the version number of the sensor			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LUUIDH”
Device response value:	“LE4917YYMM#####””		

LTYPEH

Get Sensor Model Number			
Sensor replies with the Model number of the sensor			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LTYPEH”
Device response value:	“LDS-50C-C30E”		

LBUILH

Get Sensor Firmware Build Date			
Sensor replies with the firmware build date and time string.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LBUILH”
Device response value:	“build Nov 2 2022 08:54:19”		

LMDMMH

Set Sensor Resolution mm – Future			
Sets Sensor Measurement resolution to millimeters. Future feature, not currently implemented			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LMDMMH”
Device response value:			“unsupport:LMDMMH””

LMDCMH

Set Sensor Resolution cm – Future			
Sets Sensor Measurement resolution to centimeters. Future feature, not currently implemented			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LMDCMH”
Device response value:			“unsupport:LMDCMH””

LNCONH

Set Sensor Energy Measurement Mode – Future			
Sets Sensor Energy measurement feature Future feature, not currently implemented			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LNCONH”
Device response value:			“unsupport:LNCONH””

LSSS#H

Enable/Disable Smooth Filter Command – Deprecated			
Tell the sensor to enable or disable the Smoothing Filter. This setting is not persistent after a sensor restart. The updated version of this command is now preferred			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	4	Fixed as “LSSS”
2	Enable	1	Boolean determining enable status
3	End Identifier	1	Fixed as “H”
Example Usage		ON	“LSSS1H”
		OFF	“LSSS0H”
Device response value:			“OK”

LFFF#H

Enable/Disable Deshadow Filter - Deprecated			
Enable or Disable the Deshadow filter option. This filter removes phantom points between a close target and a far target. As the laser spot over gaps between the two targets, intermediate phantom points can be created. This setting is not persistent after a sensor restart. The updated version of this command is now preferred.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	4	Fixed as “LFFF”
2	Enable	1	Boolean determining enable status
3	End Identifier	1	Fixed as “H”
Example Usage		ON	“LFFF1H”
		OFF	“LFFF0H”
Device response value:		“OK”	

LSBPS:

Modify Sensor Serial Port Baud Rate			
Sets the baud rate for serial communications (unsupported in ProE which uses Ethernet , included here for completeness)			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LSBPS:”
2	Baud Rate	6	Baud Rate of Serial Comm 768000
4	End Identifier	1	Fixed as “H”
Example Usage:		“LSBPS:768000H”	
Device response value:		“unsupport:LSBPS:”	
Extra: Supported Rates		921600, 768000, 500000, 384000, 256000, 230400, 115200	

Set Word Messages

The Parakeet ProE has a set of configuration messages that can be sent to it via the UPD protocol on the ethernet connection. These configuration settings control how the sensor operates, enabling or disabling various filters, or setting values like speed or resolution.

All messages between the controlling computer and the sensor are formatted similarly and can be sent directly over serial communications or packaged in a UDP packet for transmission to the sensor. These configuration messages use 0x0053 as the message identifier.

Each of the following tables define the specific details of each configuration message. Placing the contents of each message within the Parakeet ProE data frame and sending to the sensor using UDP will cause that configuration setting to be changed. All configuration messages cause the sensor to generate a response message indicating that the message was received and understood.

Set Word UDP Packet Format

All messages between the controlling computer and the sensor are formatted similarly and packaged in a UDP packet for transmission to the sensor. This packet format is described here. There are separate packet definitions for messages to the sensor and the sensor's response packets. (Serial communication Parakeet sensors use the same messages, but do not use the UDP packet format)

Set Word Message Format			
A message which is sent to the sensor in order to configure the device. This command packet is embedded in a UDP packet for transmission. The sensor responds with a response message in another UDP packet.			
Field #	**Name**	**Length**	**Interpretation**
1	signature	2	Fixed as "0x484C"
2	identifier	2	Tells the sensor which category of command this packet describes. Set Word Messages use: 0x0053
3	sequenceNumber	2	A unique identifier for this packet. The response packet which is returned from this packet, will have the same sequence number
4	length	2	The number of characters in field 5 message.
5	message	varies	A Control Word message or a Setting Word message. The Command Set section has information describing this
6	crc32	4	A CRC32 checksum of fields 1-5. Initial value: 0xFFFFFFFF Polynomial value: 0x04C11DB7 Code example maybe found here

Set Word Response Message Format			
The response message packet is sent to the host in response to the sensor receiving a control word message. This packet is embedded in a UDP frame and sent via ethernet.			
Field #	**Name**	**Length**	**Interpretation**
1	signature	2	Fixed as "0x484C"
2	identifier	2	Fixed as "0xFFAC"
3	sequenceNumber	2	Contains the sequence number matching the control packet the device is responding to
4	length	2	The number of characters in field 5
5	message	varies	The message response from the sensor. More information on this can be found in each control word section.
6	crc32	4	A CRC32 checksum of fields 1-5. Initial value: 0xFFFFFFFF Polynomial value: 0x04C11DB7 Code example maybe found here

LSSMT:

Enable/Disable Smooth Filter Command – Preferred			
Sets the state of the data smoothing filter. This setting is persistent across restart.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as "LSSMT:"
2	Enable	1	Boolean determining enable status
3	End Identifier	1	Fixed as "H"
Example Usage		ON	"LSSMT:1H"
		OFF	"LSSMT:0H"
Device response value:			"OK"

LSATS:

Enable/Disable Auto Scan on Powerup			
The sensor will spin up and start sending data automatically when power is connected.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	4	Fixed as "LSATS:"
2	Enable	1	Boolean determining enable status
3	End Identifier	1	Fixed as "H"
Example Usage		ON	"LSATS:1H"
		OFF	"LSATS:0H"
Device response value:			"OK"

LSTFX:

Enable/Disable Fixed Server Only			
If Enabled, then send data to the IP address identified as the Server, else send data to 0.0.0.0 until a message is received, then send data to the return address received.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	4	Fixed as "LSTFX:"
2	Enable	1	Boolean determining enable status
3	End Identifier	1	Fixed as "H"
Example Usage		ON	"LSTFX:1H"
		OFF	"LSTFX:0H"
Device response value:		"OK"	

LSDID:

Set Device ID			
Set the Device ID of the sensor.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	4	Fixed as "LSDID:"
2	Value	1~10	ID Value 0~2147483647
3	End Identifier	1	Fixed as "H"
Example Usage		1	"LSDID:1H"
		12345	"LSDID:12345H
		2,147,483,647	"LSDID:2147483647H"
Device response value:		"OK"	

LSPST:

Enable/Disable Alarm Message Upload			
Sensor will send an alarm message periodically when any alarm is detected.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	4	Fixed as "LSPST:"
2	Enable	1	Value determining enable status
3	End Identifier	1	Fixed as "H"
Example Usage		ON	"LSPST:3H"
		OFF	"LSPST:1H"
Device response value:		"OK"	

LSDOG:

Watchdog Timing Interval			
Set the interval within which messages must be received to avoid sensor restart. Values from 20 Sec to 300 Sec or 0 to disable. Any received message will reset watchdog timeout, but TimeSync is most common sent by the host computer.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	4	Fixed as “LSDOG:”
2	Value	1~3	Value setting Packet Interval (20~300sec, 0=Disable)
3	End Identifier	1	Fixed as “H”
Example Usage		20 Sec	“LSDOG:20H”
		123 Sec	“LSDOG:123H”
		0 Disable	“LSDOG:0H”
Device response value:		“OK”	

LSRPM:

Set Rotation RPM			
Set the RPM of the sensor scan motor. Values of 600 and 900 valid only			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	4	Fixed as “LSRPM:”
2	Enable	3	Value defining scan RPM
3	End Identifier	1	Fixed as “H”
Example Usage		10Hz	“LSRPM:600H”
		15Hz	“LSRPM:900H”
Device response value:		“OK”	

LSCCW:

Set Rotation Direction CW or CCW			
Set the sensor rotational direction to CW or CCW. This is a future feature that is not available in current devices.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LSCCW:”
2	Enable	1	Boolean 1 = CCW; 0 = CW
3	End Identifier	1	Fixed as “H”
Example Usage		CCW	“LSCCW:1H”
		CW	“LSCCW:0H”
Device response value:		“NG” (No Good - until implemented)	

LSDSW:

Enable/Disable Deshadow Filter Command			
Enable or Disable the Deshadow filter option. This filter removes phantom points between a close target and a far target. As the laser spot over laps between the two targets, intermediate phantom points can be created. This newer style of command matches the large number of new configuration messages that were added. This style is persistent over restarts.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as "LSDSW:"
2	Enable	1	Boolean determining enable status
3	End Identifier	1	Fixed as "H"
Example Usage		ON	"LSDSW:1H"
		OFF	"LSDSW:0H"
Device response value:		"OK"	

LSAF:

Set Deshadow Filter Strength			
Set control value for Deshadow Filter. Value range of 1 to 200 allowed			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	4	Fixed as "LSAF:"
2	Value	1~3	Value determining filter strength. 1~200
3	End Identifier	1	Fixed as "H"
Example Usage		20	"LSAF:20H"
		199	"LSAF:199H"
Device response value:		"OK"	

LSRES:

Set Angle Resolution of Resampling			
Set the resampling Angle Resolution. Valid values are 043 : 999 when RPM is set to 600 Valid values are 065 : 999 when RPM is set to 900 A leading decimal point is assumed. This value is ignored if resampling is disabled. A value of 000 will disable the resampling. A value of 001 will enable resampling.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	4	Fixed as "LSRES:"
2	Value	3	Value determining angle resolution
3	End Identifier	1	Fixed as "H"
Example Usage		0.000	"LSRES:000H" Disable
		0.001	"LSRES:001H" Enable
		0.043	"LSRES:043H"
		0.999	"LSRES:999H"
Device response value:		"OK"	

IP Communication Set Word Messages

The Parakeet ProE has a set of communication configuration messages that can be sent to it via the UPD protocol on the ethernet connection. These communication settings control how the sensor sends messages to the controlling computer.

All messages between the controlling computer and the sensor are formatted similarly and packaged in a UDP packet for transmission to the sensor. This message format is described above for all Set Word messages.

Each of the following tables define the specific details of each configuration message. Placing the contents of each message within the Parakeet ProE data frame and sending to the sensor using UDP will cause that configuration setting to be changed. All configuration messages cause the sensor to generate a response message indicating that the message was received and understood.

LSUDP:

Modify Sensor Source IPv4 Settings			
Sets the IP Address, subnet mask, gateway, and service port number the device will be listed as on the network.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LSUDP:”
2	IP Address	15	IP Address the sensor should be listed under on the network By default: 192.168.158.098
3	Subnet Mask	15	Subnet Mask the sensor should be listed under on the network By default: 255.255.255.000
4	Gateway	15	Gateway the sensor should be listed under on the network By default: 192.168.158.001
5	Service Port	5	Port the sensor will be listening on By default: 06543
6	End Identifier	1	Fixed as “H”
Example Usage:		“LSUDP:192.168.000.005 255.255.255.000 192.168.000.001 06543H”	
Device response value:		“OK”	
NOTE:		Spaces are used to delineate between fields 2-3, 3-4 and 4-5	

LSDST:

Modify Sensor Destination IPv4 Settings			
Sets the IP Address, port number the sensor will send packets to.			
Field #	**Name**	**Length**	**Interpretation**
1	Start Identifier	6	Fixed as “LSDST:”
2	Destination IP Address	15	IP Address the destination device is listed at on the network By default: 192.168.158.015
3	Destination Port	5	Port the destination device will be listening on By default: 06668
4	End Identifier	1	Fixed as “H”
Example Usage:		“LSDST:192.168.000.015 06668H”	
Device response value:		“OK”	
NOTE:		A space is used to delineate between fields 2-3	

Bulk Settings Request Message Definition

Bulk Settings Request Message Definition			
Sending this message to the sensor will cause it to respond with a list of many of the current settings of the sensor. This message is used when the “Setup…” dialog is opened in the test and configuration software.			
Field #	**Name**	**Type**	**Interpretation**
1	messageHeader	uint8[4]	Fixed as “LHSG”
2	sequenceNumber	uint16	Sequence Number - Random Number
3	length	uint16	Command Length 0x0000
4	crc32	uint32	CRC 32

Bulk Settings Response Packet Definition

Bulk Settings Response Packet Defintion			
Packet containing the current settings of the Parakeet. Most of these settings are available in other messages, but this allows the collection of most settings with a single command / response message pair.			
Field #	**Name**	**Type**	**Interpretation**
1	signature	uint8[2]	Fixed as 0x484C "LH"
2	identifier	uint16	Identifier 0xB8AC
3	sequenceNumber	uint16	Reply of same random number sent by Host
4	length	uint16	Number of bytes of data, currently 152. May change with future FW changes.
5 4	label	char[4]	Info Label "EPRM"
6 6	pp_ver	uint16	Protocol 0x0101
7 8	size	uint16	Currently 0x98 or 152 Does not include CRC32
8 28	serialNumber	uint8[20]	Device SN "LE4917220700001"0x00000000
9 44	modelNumber	uint8[16]	Device Model Number "LDS-50C-C30E" 0x00000000
10 48	dev_id	uint32	Number from 1 to 2147483647
11 52	sourceIPAddress	uint8[4]	Device Source IP address on the network
12 56	sourceSubnetMask	uint8[4]	Subnet mask of the device
13 60	sourceGateway	uint8[4]	Gateway address the device uses
14 64	destinationIPAddress	uint8[4]	The IP Address of the destination host the sensor is currently communicating to
15 66	destinationPort	uint16	IP Destination Port
16 68	sourcePort	uint16	IP Source Port
17 70	RPM	uint16	Sensor RPM Setting
18 72	RPM_pulse	uint16	RPM per Pulse Motor Startup Parameter
19 73	fir_filter	uint8	Setting for related Lidar Sensor type : Unused
20 74	cir	uint8	Setting for related Lidar Sensor type : Unused
21 76	resample	uint16	Resampling Resolution
22 77	auto_start	uint8	Auto Start Sampling on Power up
23 78	target_fixed	uint8	Upload to defined destination
24 79	with_smooth	uint8	Data point smoothing enable
25 80	with_filter	uint8	Drag point smoothing enable
26 88	ranger_bias	uint8[8]	Measurement correction bias
27 92	heartbeatInterval	uint32	Heart Beat Packet Interval
28 96	pnp_flags	uint32	IO Port polarity configuration
29 100	not_used	uint32	Spare
30 116	fun_map	uint8[16]	IO Port mapping configuration
21 152	reserved	uint8[36]	Reserved
32	crc32	uint32	A CRC32 checksum of fields 5-21. Code example may be found here

Alarm Message

The sensor will send alarm messages at the rate of about 1.536 seconds between messages. These messages can be disabled by sending the settings message "LSPST:1H". See the LSPST section above for more information.

Alarm Packet Definition

Alarm Packet Definition			
Packet from sensor which describes an ongoing alarm			
Field #	**Name**	**Type**	**Interpretation**
1	alarmHeader	uint8[4]	Fixed as "LMSG"
2	protocolVersion	uint32	Protocol version currently 0x0101
3	deviceSerialNumber	uint8[20]	Serial number of the device
4	deviceID	uint32	Device number
5	timestamp	uint32	Timestamp of when this message was captured
6	alarmType	uint32	0 = No Error 1 = Error occurred Other values = Reserved
7	alarmMessage	uint32	Bit 1 = Low Power Bit 2 = Motor Locked Rotation Bit 3 = Overheated Bit 4 = Network Error Bit 5 = No output from ranging module Bit 6-32 = Reserved
8	extra	uint16 [42]	Reserved

Heart Beat Message

The Heart Beat Message is sent on a regular basis of 1.00 seconds between messages.

Heartbeat Packet Definition

Network Heartbeat Protocol Packet			
Packet containing the current state of the lidar device. Transmissions of these packets begins on sensor startup. This message is always broadcast to 255.255.255.255. A heartbeat packet is transmitted from the device every second.			
Field #	**Name**	**Type**	**Interpretation**
1	heartbeatHeader	uint8[4]	Fixed as “LiDA”
2	protocolVersion	uint32	Protocol version
3	timestamp	uint64	Timestamp
4	deviceSerialNumber	uint8[20]	Serial number of the device
5	deviceType	uint8[16]	Type of device
6	deviceVersion	uint32	Bottom control panel version
7	deviceID	uint32	Device number
8	sourceIPAddress	uint8[4]	Device address on the network
9	sourceSubnetMask	uint8[4]	Subnet mask of the device is listed on
10	sourceGateway	uint8[4]	Gateway the device is listed on
11	destinationIPAddress	uint8[4]	The IP Address of the destination device the sensor is currently communicating to
12	destinationPort	uint16	The port which the sensor is sending data to
13	sourcePort	uint16	The port which the sensor is listening on
14	reserved	uint16	NA
15	rpm	uint16	# of revolutions per minute. Multiplied by 10
16	frequency_Hz	uint16	# of revolutions per second. Multiplied by 100
17	rangeVersion	uint8[2]	Top ranging head program version
18	cpuTemp	uint16	Temperature of device CPU. measured in Celsius. Multiplied by 10.
19	inputVoltage	uint16	Voltage through the device. Multiplied by 1000
20	deviceState	uint8[16]	Device state information, each byte is a boolean value matching the **Device State table below**.
21	crc32	uint32	A CRC32 checksum of fields 1-20. Code example maybe found here

Device State Definitions

Device State As used in Network Heartbeat Packet			
Index	**Name**	**Interpretation**	**Available for ProE**
0	LV_1	Viewing area	✗
1	LV_2	Warning area	✗
2	LV_3	Alarm area	✗
3	COVER	Shelter	✗
4	NO_DATA	No data from ranging head	✓
5	ZONE_ACTIVE	No defense zone setting	✗
6	SYS_ERR	System internal error	✓
7	RUN	Abnormal system operation	✓
8	NET_LINK	Network connection error	✓
9	UPDATING	Device update in progress	✓
10	ZERO_POS	0 output	✗
11-13	Reserved	reserved	✗
14	USB_LINK	USB Connect	✗
15	ZERO_DEFINED	Zero position could not be detected	✓
16-32	Reserved	reserved	✗

Time Synchronization Messages

Time Sync messages between the controlling (host) computer and the sensor are formatted as shown below and packaged in a UDP packet for transmission to the sensor. There are similar formats for the host message to the sensor and the sensor's response message.

If not synced the time stamp will roll over every 25 seconds. A typical repeat rate for this message, to keep the sensor synced up would be between 2 and 5 seconds. The visualization and test software repeats the time stamp every 2 seconds.

The host time stamp is in milliseconds. The sensor arrival time is in frequency ticks as reported in the second field by the sensor. The host calculates the delta time for two host packets ~2000 msec, and the delta time between arrival times divided by the frequency times 1000 to get ticks per msec. This result will also be ~2000.

Keeping the time stamp of the sensor data packets synchronized with the control computer's clock allows the control computer to sync the LiDAR data with other sensor data that is being used for SLAM navigation. The time span of the sync only needs to be long enough so that previous saved data is not confused with new data.

Since this is typically the only message that is sent by the host to the sensor on a regular basis, this message usually resets the Lidar's internal watch dog. If this message is not sent within the timeout interval the sensor will restart in an attempt to clear any communication faults.

Time Synchronization Packet from Host

Time Synchronization Packet Format			
A packet which is sent to the sensor in order to set the time of the sensor. The sensor responds with a response packet defined below.			
Field #	**Name**	**Length**	**Interpretation**
1	signature	2	Fixed as "0x484C" "LH"
2	typeID	2	Tells the sensor which category of command this packet describes. Time Sync command uses: "0x4B41" "AK"
3	sequenceNumber	2	A unique identifier for this packet. The response packet which is returned from this packet, will have the same sequence number
4	dataLength	2	The number of characters in data fields fixed as "0x20" for time sync
5	timeStamp	4	This is the time of the control computer in milliseconds.
6	Frequency	4	0x00000000 Lidar Internal Use.
7	arrivalTime	4	"0x00000000 Lidar Internal Use
8	delayTime	4	Lidar Internal Use
9	reserved	16	Fill space with 16x "0x00".
10	crc32	4	A CRC32 checksum of fields 1-7. Initial value: 0xFFFFFFFF Polynomial value: 0x04C11DB7 Code example maybe found here

Time Synchronization Response from Sensor

Time Synchronization Response Packet Format			
A packet which is sent to the host in response to the sensor receiving a command packet.			
Field #	**Name**	**Length**	**Interpretation**
1	signature	2	Fixed as “0x484C” “LH”
2	identifier	2	Fixed as “0xB4BE”
3	sequenceNumber	2	Contains the sequence number matching the time sync packet the device is responding to
4	length	2	The number of characters in data fields fixed as “0x20” for time sync
5	timeStamp	4	SyncTime value as received in Time sync message
6	Frequency	4	“0x0A037A00” internal frequency of sensor Used to convert timestamp to Milliseconds.
7	arrivalTime	4	32 bit time stamp internal to sensor
8	delayTime	4	Delay time from sensor
9	reserved	16	Space filled with 16x“0x00”
10	crc32	4	A CRC32 checksum of fields 1-9. Initial value: 0xFFFFFFFF Polynomial value: 0x04C11DB7 Code example maybe found here

Lidar Data Packet Definition

Lidar Data packet			
Each scan revolution is made up of 10 sectors, and each sector is made up of subsectors. The number of subsectors varies depending on the number of points within the sector. A Lidar data packet maps to one subsector. Transmission of these packets starts when the sensor starts spinning. All subsector packets of the same sector are transmitted at the same time.			
Field #	**Name**	**Type**	**Interpretation**
1	lidarDataHeader	uint16	Fixed as 0xFAC7
2	numPoints	uint16	Number of points within this packet
3	totalPointsInSector	uint16	Total number of points within this sector of the lidar scan. E.g: a scan of 100 points may be broken up into two packets, where numPoints is 90 for one, and 10 for another. In both packets, this value will be set as 100
4	offset	uint16	Index of the first point of this packet within the sector.
5	startAngle	uint32	Starting angle of this sector, with 0.001° as the resolution. E.g: 180° will be stored as 180000
6	endAngle	uint32	Ending angle of this sector, with 0.001° as the resolution. E.g: 180° will be stored as 180000
7	flags	uint32	Bit 1 = Distance scale, 0 = mm, 1 = cm Bit 2 = Intensity data included, 0 = not included, 1 = included Bit 3 = Drag point removal filter, 0 = Off, 1 = On Bit 4 = Data smoothing filter 18° sector, 0 = Off, 1 = On Bit 5 = Data smoothing filter 9° sector, 0 = Off, 1 = On Bit 8 = Resample filter, 0 = Off, 1 = On Bit 9 = Rotation Direction, 0 = CW, 1 = CCW
8	timestamp	uint32	0 - 25565, in milliseconds. Indicates the laser emission time of the first point in the sector, relative to when the sensor is turned on. The sensor timestamp defaults to 0 on startup
9	deviceNumber	uint32	Equipment number assigned to this device
10	distanceArray	uint16[numPoints]	Distance between the lidar device and where the lidar laser beam had come into contact. Stored as millimeters or centimeters, dependent on the value in flags
11	relativeAngleArray	uint16[numPoints]	Angle relative to startAngle, with 0.001° as the unit. E.g: 180° will be stored as 180000
12	intensityArray	uint8[numPoints] OR uint8[0]	Return strength of lidar laser beam. If the intensity is enabled in flags, intensityArray has size of numPoints, otherwise it has size of 0
13	checksum	uint16	Sum of fields 2 -> 12. More information in the Lidar Data Packet Checksum Calculation section

Block View of Data Packet

Byte	0	1	2	3	4	5	6	7
	headerCode		numPoints		totalPointsInSector		offset	
	startAngle				endAngle			
	flags				timestamp			
	deviceNumber				distanceArray			
	relativeAngleArray				intensityArray			
	checksum							

Example View of Data Packet

Example Lidar data packet (big-endian form)				
headerCode	numPoints	totalPointsInSector	offset	startAngle
C7FA	0004	00BC	00B8	0003D860
endAngle		flags	timestamp	
00046500		0000000E	DDE60861	
deviceNumber		distanceArray		
00000001		0000 1708 0000 1704		
relativeAngleArray		intensityArray		
C689 848A 428B 058C		00 11 00 0F		
checksum				
F27E				

Data Packet Checksum Calculation

Each data packet the sensor sends out begins with a header and ends with a checksum. The value of this checksum can be calculated from summing all values in the packet excluding the header and checksum. Data types smaller than uint16 get expanded to uint16, ie: 0xF -> 0x0F. Data types larger than uint16 get separated into uint16, ie: 0xDDFF -> 0xDD 0xFF. Bytes will also need to be converted from little-endian to big-endian.
This is only used for Lidar Data Packets and is not the same as the CRC32 value used by UDP packets in general.

Checksum Example

This is an example of how to calculate a checksum for a Lidar data packet. Note that field 1 of a Lidar data packet is the header, and headers are not included in the checksum calculation. Also note that since field 13 is the checksum, it will also not be included in the calculation. For simplicity, mod 0x10000 will be done as the final step in this calculation to get it to fit into 16 bits. A code example may be located here.

Field #	Little Endian	Big Endian	Sum of components	Cumulative sum
2	0400	0004	0004	0004
3	BC00	00BC	00BC	00000C0
4	B800	00B8	00B8	0178
5	60D80300	0003D860	0003 + D860 = D863	D9DB
6	00650400	00046500	0004 + 6500 = 6504	13EDF
7	0E000000	0000000E	0000 + 000E = 000E	13EED
8	6108E6DD	DDE60861	DDE6 + 0861 = E647	22534
9	01000000	00000001	0000 + 0001 = 0001	22535
10	0000 0817 0000 0417	0000 1708 0000 1704	0000 + 1708 + 0000 + 1704 = 2E0C	25341
11	C689 848A 428B 058C	89C6 8A84 8B42 8C05	89C6 + 8A84 + 8B42 + 8C0 = 22B91	47ED2
12	00 11 00 0F	00 11 00 0F	0000 + 0011 + 0000 + 000F = 0020	47EF2 % 0x10000 = 7EF2

In Application Programming Interface

The Parakeet LiDAR sensor has a communication interface for reprogramming the Program FLASH memory. This IAP interface provides several commands that allow a binary image file to be down loaded to the sensor and programmed into the program memory. The commands have a common format shown below. For commands the Opcode is stored in the Offset field and the CRC is set to its initial value 0xFFFFFFFF. The command opcodes are large values that are out of range for normal offsets for programming data.

IAP Message Formats

In Application Programming - Message Format			
Message to program the flash memory			
Field #	**Name**	**Length**	**Interpretation**
1	Offset	4	32 bit value of where to write the data
2	CRC32	4	32 bit CRC of the data
3	Data	512	Up to 512 bytes of data
Device response value:			

In Application Programming - Response Format			
Message to program the flash memory			
Field #	**Name**	**Length**	**Interpretation**
1	Offset	4	32 bit value of where to write the data
2	Result	4	32 bit CRC of the data
3	Msg	128	Up to128 bytes of data (usually a few ascii char)

First Operation – Erase FLASH Memory

The first step in upgrading the program code in the sensor is to erase the FLASH memory where the program will be stored. This command is inserted into a UDP packet and sent to the sensor. The opcode for FLASH_ERASE is 0xFE00EEEE, with a 32 bit CRC. The data of the message is a 32 bit "int" of the length, in bytes, of memory to be erased/programmed. The length must be a multiple of 512 bytes.
The sensor will respond with a message using the same opcode, the size erased and the message text "erase flash ok".

In Application Programming - Erase Flash Message			
Message to program the flash memory			
Field #	**Name**	**Value**	**Interpretation**
1	Offset	0xFE00EEEE	Erase Op Code
2	CRC32	0xFFFFFFFF	Dummy CRC
3	Data	Length	Data bytes in the Upgrade File/ bytes to erase
Device response value:		Result is 0 Msg is "erase flash ok"	

Second Operation(s) – Send Firmware in 512 byte packets

The second step in upgrading the program code in the sensor is to send all of the new code to the sensor in a series of 512 byte packets. This data is held in RAM until all of the firmware is loaded and is then programmed into the flash memory during the reboot process. This command is inserted into a UDP packet and sent to the sensor. The Offset value is a multiple of 512 and is less than 0x00FFFFFF. The CRC value is calculated across the 512 bytes of data not the full message.

In Application Programming - Program Flash Message			
Message to program the flash memory			
Field #	**Name**	**Value**	**Interpretation**
1	Offset	0x00######	Offset location to write this data block
2	CRC32	0x########	Calculated CRC of 512 data bytes
3	Data	Firmware	Data bytes to be programmed
Device response value:		Result = 0 Msg is "write flash ok"	

Third Operation – Send IAP Confirmation Value

The third step in upgrading the program code in the sensor is to send an IAP compete message with a data value of the total data file length.

In Application Programming - Program IAP Length			
Message to program the flash memory			
Field #	**Name**	**Value**	**Interpretation**
1	Offset	0xFE00AAAA	Program IAP Opcode
2	CRC32	0xFFFFFFFF	Dummy CRC
3	Data	32uint Length	Total number of data bytes that were programmed
Device response value:		Result = 0 Msg is "write IAP ok"	

Final Operation – Send Restart Command

The fourth step in upgrading the program code in the sensor is to send a Reset command. The Offset value is the Reset Opcode 0xFE00BBBB, the CRC is 0xFFFFFFFF, and the data is 0xabcd1234. This final data signature is used as a double check that it is really time to reset the sensor. The programming of the flash memory in the sensor is accomplished in this step

In Application Programming – Reset Firmware			
Message to program the flash memory			
Field #	**Name**	**Value**	**Interpretation**
1	Offset	0xFE00BBBB	Reset Firmware Opcode
2	CRC32	0xFFFFFFFF	Dummy CRC
3	Data	0xabcd1234	Additional verification signature
Device response value:		No Response – Firmware is restarting	

IAP UDP Packet Format

All messages between the controlling computer and the sensor are formatted similarly and packaged in a UDP packet for transmission to the sensor. This packet format is described here. There are separate packet definitions for messages to the sensor and the sensor's response packets. (Serial communication sensors use the same messages, but do not use the UDP packet format)

Command Packet Format			
A packet which is sent to the sensor in order to configure the device. The sensor responds with a response packet.			
Field #	**Name**	**Length**	**Interpretation**
1	signature	2	Fixed as "0x484C"
2	identifier	2	Tells the sensor which category of command this packet describes. IAP operations use: 0x0046
3	sequenceNumber	2	A unique identifier for this packet. The response packet which is returned from this packet, will have the same sequence number
4	length	2	The number of characters in field 5
5	message	1024	An IAP message. The IAP Operation section has information describing this
6	crc32	4	A CRC32 checksum of fields 1-5. Initial value: 0xFFFFFFFF Polynomial value: 0x04C11DB7 Code example maybe found here

IAP Upgrade Utility

Source code that implements the IAP protocol, described above, is available from Lidar Engineering as "upgrade.c". This C program can be compiled to run under Linux command line with arguments to specify the sensor and upgrade firmware file. Contact Lidar Engineering LLC for this software.

Example Command Line - Linux "upgrade -d 192.168.0.5 -p 6543 -f Parakeet.lhl"

Arguments

-d	Lidar IP Address	ie. 192.168.0.5
-p	Destination IP Port	ie. 6543
-f	Firmware File Name	ie. Parakeet.lhl

This protocol is also included in the Graphical Evaluation Program described earlier in this document.

Lidar Engineering 2023
www.LidarEngineering.com