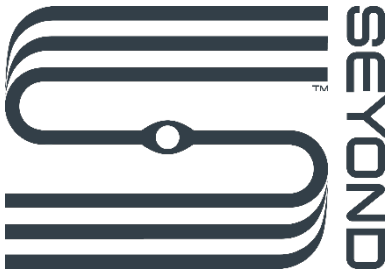


Falcon Data Format Description

Application Note

Table of content

Preface

Product

Falcon LiDAR data format

Manufacturer

SEYOND

Legal information

Overview

The intended users of this manual include project developers (R&D personnel and designers) and service personnel.

Original document

This document is an original document of Seyond.

Manual description

Although this document covers instructions to handle the frequent problems, it is still not guaranteed to get all problems fully solved. If you encounter other problems not covered in the manual, please contact Seyond staff in time. This manual will be updated when new information becomes available.

Tel : (650)963-9573

E-mail: info@ Seyond.com

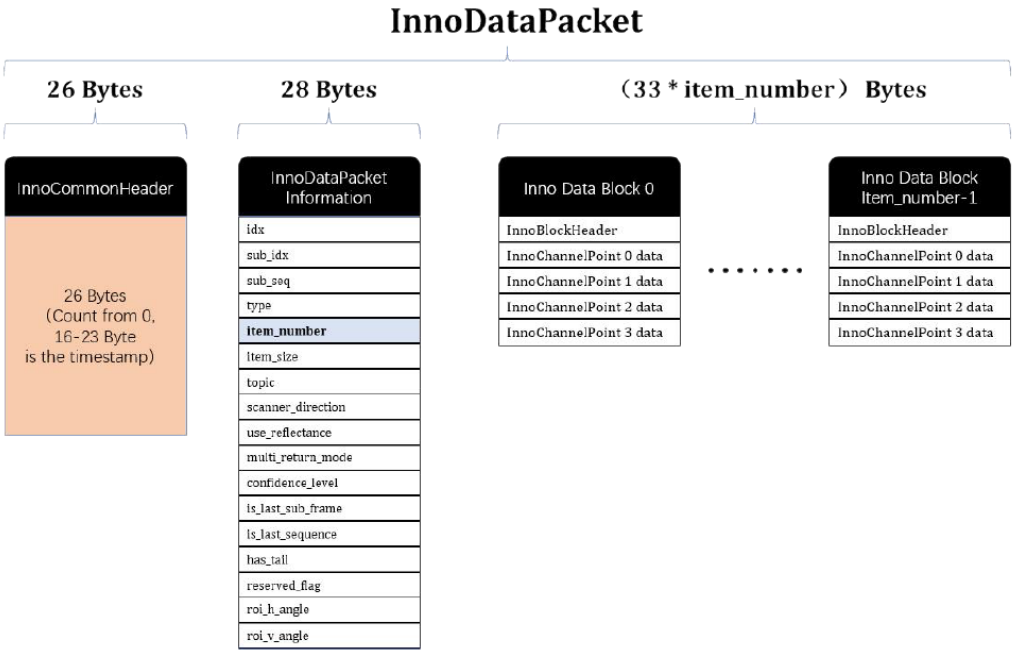
1 Overview

This document describes the structures and field definitions of the InnoDataPacket and InnoStatusPacket packets. Each frame of data is divided into multiple InnoDataPacket packets in accordance with the configured UDP packet size. One InnoStatusPacket packet is sent every 50 ms.

Each InnoDataPacket is composed of an InnoCommonHeader, an InnoDataPacket Information, and several Inno Data Blocks. Each InnoStatusPacket is composed of an InnoCommonHeader and an InnoStatusPacket Information. All data packets are in little-endian mode.

2 InnoDataPacket specification

This chapter mainly describes the InnoDataPacket packet definition. The details of each variable in the packet are explained as follows.



2.1 InnoCommonHeader

InnoCommonHeader: 26 Bytes.

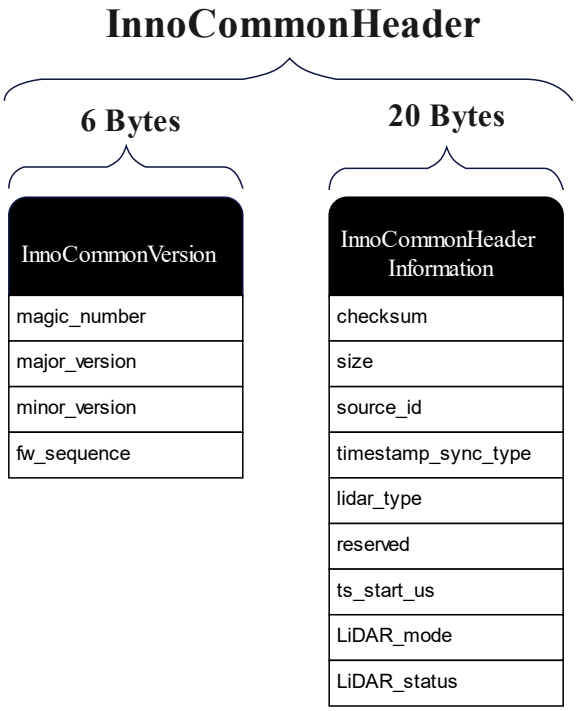


Table 1 InnoCommonHeader Specification

Variable	Data type	Size (Bit)	Description

version	InnoCommonVersion	48	Data packet version information. For details, refer to *2.2 InnoCommonVersion*.
checksum	uint32_t	32	Checksum, used for CRC checking.
size	uint32_t	32	Size of InnoDataPacket.
source_id	uint8_t	4	Currently not used. The default value is 0.
timestamp_sync_type	uint8_t	4	Time synchronization type. A value of timestamp_sync_type=7 means the time has been synced in PTP mode. A value of timestamp_sync_type=11 means the time has been synced in NTP mode.
lidar_type	uint16_t	8	The LiDAR type.
ts_start_us	InnoTimestampUs	64	The trigger time of the laser pulse in the 0th Block of InnoDataPacket.
lidar_mode	uint8_t	8	LiDAR working mode. lidar_mode=1: The SLEEP mode. lidar_mode=2: The STANDBY mode. lidar_mode=3: The NORMAL mode. lidar_mode=4: The SHORT_RANGE mode. (Currently not used) lidar_mode=5: The CALIBRIATION mode. lidar_mode=6: The PROTECTION mode. lidar_mode=7: The WORK_QUIET mode.
lidar_status	uint8_t	8	LiDAR status. lidar_status=0: Initial status lidar_status=1: TRANSITION. The event is in progress. lidar_status=2: NORMAL. The event execution succeeded. lidar_status=3: FAILED. The event execution failed.

2.1.1 InnoCommonVersion

InnoCommonVersion: 6 Bytes.

Table 2 InnoCommonVersion Specification

Variable	Data type	Size (Bit)	Description

magic_number	uint16_t	16	Magic number, used for data verification.
major_version	uint8_t	8	Major version number.
minor_version	uint8_t	8	Minor version number.
fw_sequence	uint16_t	16	Reserved.

2.2 InnoDataPacket Information

InnoDataPacket Information: 28 Bytes

Table 3 InnoDataPacket Information Specification

Variable	Data type	Size (Bit)	Description
idx	uint64_t	64	Frame number, starting from 0.
sub_idx	uint16_t	16	Each frame is divided into multiple InnoDataPackets, and sub_ Idx is the sequence number of InnoDataPacket in a frame.
sub_seq	uint16_t	16	The sequence number of the point cloud packet, starting from 0.
type	uint32_t	8	Data type in InnoDataPacket. type=1: Spherical coordinate point cloud. type=2: Message. type=3: Message log. type=4: Cartesian coordinate point cloud.
item_number	uint32_t	24	The number of Blocks in the current InnoDataPacket.
item_size	uint16_t	16	Size of a single Block. The maximum size is 65535. A value of 0 means the flexible size.
topic	uint32_t	32	Currently not used. The default value is 0.
scanner_direction	uint16_t	1	The vertical direction of the LiDAR scanning. scanner_direction=0: from top to bottom scanner_direction=1: from bottom to top
use_reflectance	uint16_t	1	use_reflectance=0: intensity mode use_reflectance=1: reflectivity mode
multi_return_mode	uint16_t	3	multi_return_mode=1: single return mode multi_return_mode=2: strongest + second strongest return mode multi_return_mode=3: strongest + furthest return

			mode
confidence_level	uint16_t	2	Confidence level of point cloud data. confidence_level=0: lowest confidence level. confidence_level=1: low confidence level confidence_level=2: high confidence level confidence_level=3: highest confidence level
is_last_sub_frame	uint16_t	1	Whether it is the last InnoDataPacket in a frame. is_last_sub_frame=0: No is_last_sub_frame=1: Yes
is_last_sequence	uint16_t	1	Currently not used. The default value is 0.
has_tail	uint16_t	1	Currently not used. The default value is 0.
frame_sync_locked	uint16_t	1	Whether the frame has been synchronized. frame_sync_locked=1: The frame has been synchronized. frame_sync_locked-0: The frame hasn't been synchronized.
is_first_sub_frame	uint16_t	1	is_first_sub_frame =1: The first subframe in a frame.
last_four_channel	uint16_t	1	Currently not used. The default value is 0.
reserved_flag	uint16_t	3	Currently not used. The default value is 0.
roi_h_angle	uint16_t	16	The horizontal angle of the ROI center.
roi_v_angle	uint16_t	16	The vertical angle of the ROI center.

2.3 InnoBlock

Each InnoBlock is composed of an InnoBlockHeader and an InnoChannelPoint.

2.3.1 InnoBlockHeader

InnoBlockHeader: 17 Bytes

Table 4 InnoBlockHeader Specification

Variable	Data type	Size (Bit)	Description
h_angle	uint16_t	16	Horizontal angle of channel 0, ranging from -π to π.
v_angle	uint16_t	16	Vertical angle of channel 0, ranging from -π to π.

ts_10us	uint16_t	16	Time offset of this Block relative to the 0th Block of the same InnoDataPacket in 10 μs.
scan_idx	uint16_t	16	Block ID of the scanning line.
scan_id	uint16_t	9	ID of the scanning line.
h_angle_diff_1	uint64_t	9	Horizontal angle offset of channel 1 relative to channel 0.
h_angle_diff_2	uint64_t	10	Horizontal angle offset of channel 2 relative to channel 0.
h_angle_diff_3	uint64_t	11	Horizontal angle offset of channel 3 relative to channel 0.
v_angle_diff_1	uint64_t	9	Vertical angle offset of channel 1 relative to channel 0. The angle = v_angle + v_angle_diff_1 + kVAngleDiffBase × channel. v_angle_1 = (v_angle + v_angle_diff_1 + 196) × (180 / 32768).
v_angle_diff_2	uint64_t	9	Vertical angle offset of channel 2 relative to channel 0. The angle = v_angle + v_angle_diff_2 + kVAngleDiffBase × channel. v_angle_2 = (v_angle + v_angle_diff_2 + 392) × (180 / 32768).
v_angle_diff_3	uint64_t	9	Vertical angle offset of channel 2 relative to channel 0. The angle = v_angle + v_angle_diff_2 + kVAngleDiffBase × channel. v_angle_2 = (v_angle + v_angle_diff_2 + 392) × (180 / 32768).
in_roi	uint64_t	2	in_roi=0x00: in spare region. in_roi=0x11: in center ROI. in_roi=0x01: in vertical slow region.
facet	uint64_t	3	Polygon facet ID of the current Block (five facets in total, numbered from 0 to 4).
reserved_flags	uint64_t	2	Currently not used. The default value is 0.

2.3.2 InnoChannelPoint

InnoChannelPoint: 4 Bytes

Table 5 InnoChannelPoint Specification

Variable	Data type	Size (Bit)	Description
radius	uint32_t	17	The distance from the point in 1/200 meters, with a range of 0 to 655.35m.
refl	uint32_t	8	The reflectance or intensity, ranging from 1 to 255.
elongation	uint32_t	4	The pulse width of the echoes.
is_2nd_return	unit32_t	1	is_2nd_return=0: single return mode. is_2nd_return=1: dual return mode.
type	uint32_t	2	Reserved.

2.4 InnoMessage

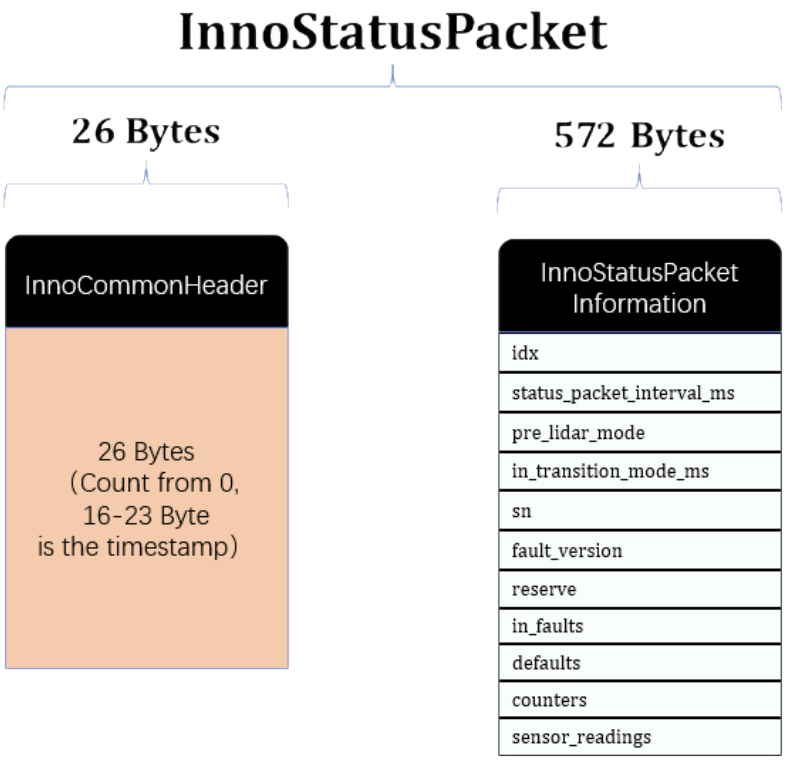
InnoMessage: 40 Bytes

Table 6 InnoMessage Specification

Variable	Data type	Size (Bit)	Description
size	uint32_t	32	The actual memory size of the InnoMessage.
src	unit32_t	32	The source of a message. The return value is the LiDAR handle.
id	uint64_t	64	The message ID.
level	unit32_t	32	The message level. About the detailed definition of the message level, see enum InnoMessageLevel.
code	unit32_t	32	The message code. About the detailed definition of the message code, see enum InnoMessageCode.
reserved	int32_t	128	Reserved.
content	char	0	The detailed content of the message.

3 InnoStatusPacket specification

This chapter mainly describes the InnoStatusPacket packet definition. The details of each field in the packet are explained as follows.



3.1 InnoCommonHeader

InnoCommonHeader: 26 Bytes.

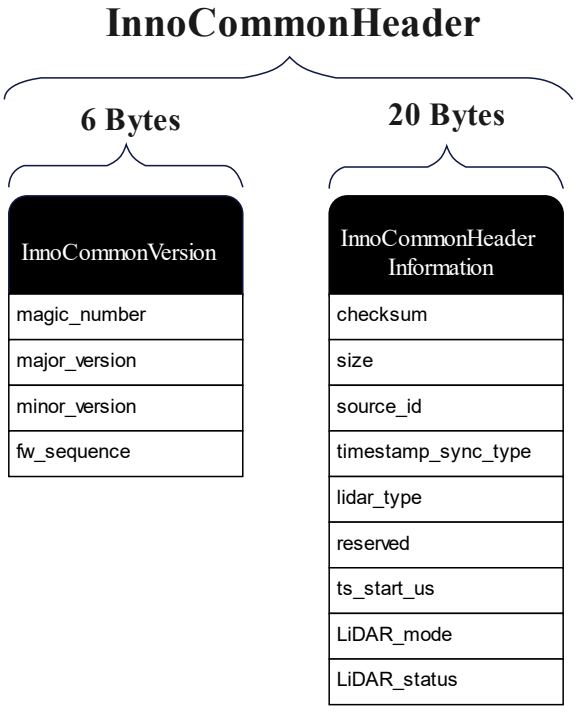


Table 7 InnoCommonHeader Specification

Variable	Data type	Size (Bit)	Description
version	InnoCommonVersion	48	Data packet version information. For details, refer to *2.2 InnoCommonVersion*.
checksum	uint32_t	32	Checksum, used for CRC checking.
size	uint32_t	32	Size of InnoDataPacket.
source_id	uint8_t	4	Currently not used. The default value is 0.
timestamp_sync_type	uint8_t	4	Time synchronization type. A value of timestamp_sync_type=7 means the time has been synced in PTP mode. A value of timestamp_sync_type=11 means the time has been synced in NTP mode.
lidar_type	uint16_t	8	The LiDAR type.
ts_start_us	InnoTimestampUs	64	The trigger time of the laser pulse in the 0th Block of InnoDataPacket.
lidar_mode	uint8_t	8	LiDAR working mode. lidar_mode=1: The SLEEP mode. lidar_mode=2: The STANDBY mode. lidar_mode=3: The NORMAL mode. lidar_mode=4: The SHORT_RANGE mode. (Currently not used) lidar_mode=5: The CALIBRIATION mode. lidar_mode=6: The PROTECTION mode. lidar_mode=7: The WORK_QUIET mode.
lidar_status	uint8_t	8	LiDAR status. lidar_status=0: Initial status lidar_status=1: TRANSITION. The event is in progress. lidar_status=2: NORMAL. The event execution succeeded. lidar_status=3: FAILED. The event execution failed.

3.1.1 InnoCommonVersion

InnoCommonVersion: 6 Bytes.

Table 8 InnoCommonVersion Specification

Variable	Data type	Size (Bit)	Description
magic_number	uint16_t	16	Magic number, used for data verification.
major_version	uint8_t	8	Major version number.
minor_version	uint8_t	8	Minor version number.
fw_sequence	uint16_t	16	Reserved.

3.2 InnoStatusPacket Information

InnoStatusPacket Information: 572 Bytes

Table 9 InnoStatusPacket Information specification

Variable	Data type	Size (Bit)	Description
idx	uint64_t	64	Status packet number, starting from 0.
status_packet_interval_ms	uint8_t	8	Time interval for status packet sending. The unit is ms.
pre_lidar_mode	uint8_t	8	Previous mode of LiDAR.
in_transition_mode_ms	uint16_t	16	Time required for LiDAR mode switching. The unit is ms.
sn	char	128	LiDAR serial number.
fault_version	uint16_t	16	The number of fault status changed. The initial value is 1.
ref_count_enough_ts_ms	uint16_t	16	The statistical time required for the reference light point number to reach a certain percentage of the trigger number.
ref_intensity	uint16_t	64	Reference light intensity.
hw_num	uint8_t	24	Reserved.
reserved	uint8_t	8	Reserved. The default value is 0.
in_faults	InnoStatusInFaults	96	Inner fault code, ranging from 0 to 95. About the detailed inner fault codes, see InnoStatusInFaults.
ex_faults	InnoStatusExFaults	32	Currently not used. The default value is 0.
counters	InnoStatusCou	2560	Data statistics. For details, refer to 3.4

	nters		InnoStatusCounters.
sensor_readings	InnoStatusSensorReadings	1536	Sensor information. For details, refer to 3.5 InnoStatusSensorReadings.

3.3 InnoStatusInFaults

InnoStatusInFaults: 12 Bytes

Table 10 InnoStatusInFaults specification

Variable	Data type	Size (Bit)	Description
faults	uint64_t	64	The fault status code.
Extended_faults	uint32_t	32	Currently not used. The default value is 0.

3.4 InnoStatusExFaults

InnoStatusExFaults: 4 Bytes

Table 11 InnoStatusExFaults specification

Variable	Data type	Size (Bit)	Description
Ex_faults	uint32_t	32	Currently not used. The default value is 0.

3.5 InnoStatusCounters

InnoStatusCounters: 320 Bytes

Table 12 InnoStatusCounters specification

Variable	Data type	Size (Bit)	Description
point_data_packet_sent	uint64_t	64	The number of InnoDataPackets sent.
point_sent	uint64_t	64	The number of InnoBlocks sent.
message_packet_sent	uint64_t	64	The number of messages sent.
raw_data_read	uint64_t	64	The size of raw_data read.
total_frame	uint64_t	64	The number of frames sent.
total_polygon_rotation	uint64_t	64	The number of polygon rotations.
total_polygon_facet	uint64_t	64	The number of polygon facet rotations.
power_up_time_in_second	uint32_t	32	Power up time. The unit is second.

process_up_time_in_second	uint32_t	32	Execution time. The unit is second.
lose_ptp_sync	uint32_t	32	The number of PTP time synchronization loss faults
bad_data	uint32_t	128	Bad data count. Currently not used. The default value is 0.
data_drop	uint32_t	256	Data loss count.
in_signals	uint32_t	256	Data count.
latency_10us_average	uint16_t	96	Average delay. The unit is 10 μs.
latency_10us_variation	uint16_t	96	Fluctuating latency. The unit is 10 μs.
latency_10us_max	uint16_t	96	Maximum latency. The unit is 10 μs.
big_latency_frame	uint32_t	32	The number of long-latency frames. Currently not used. The default value is 0.
bad_frame	uint32_t	32	The number of bad frames. Currently not used. The default value is 0.
big_gap_frame	uint32_t	32	The number of long-interval frames. Currently not used. The default value is 0.
small_gap_frame	uint32_t	32	The number of short-interval frames. Currently not used. The default value is 0.
cpu_percentage	uint16_t	16	CPU utilization.
mem_percentage	uint16_t	16	Memory utilization.
motor	uint16_t	80	Polygon encode statistics.
galvo	uint16_t	80	Galvo encode statistics.
netstat_rx_speed_kBps	uint16_t	16	Packet receiving rate.
netstat_tx_speed_kBps	uint16_t	16	Packet sending rate.
netstat_rx_drop	uint16_t	16	Packet loss count of received packets.
netstat_tx_drop	uint16_t	16	Packet loss count of sent packets.
netstat_rx_err	uint16_t	16	Error count of received packets.
netstat_tx_err	uint16_t	16	Error count of sent packets
sys_cpu_percentage	uint16_t	64	Total CPU utilization.
lifelong_uptime	uint32_t	32	The LiDAR startup time in second.

reserved	uint32_t	576	Reserved. The default value is 0.

3.6 InnoStatusSensorReadings

InnoStatusSensorReadings:192 Bytes

Table 13 InnoStatusSensorReadings specification

Variable	Data type	Size (Bit)	Description
temperature_fpga_10th_c	uint16_t	16	FPGA temperature. The unit is 0.1°C. For example, if the FPGA temperature is 30°C, then temperature_fpga_10th_c = 300.
temperature_laser_10th_c	uint16_t	16	Laser temperature. The unit is 0.1°C. For example, if the laser temperature is 30°C, then temperature_laser_10th_c = 300.
temperature_adc_10th_c	uint16_t	16	ADC module temperature. The unit is 0.1°C. For example, if the ADC module temperature is 30°C, then temperature_adc_10th_c = 300.
temperature_board_10th_c	int16_t	16	Mainboard temperature. The unit is 0.1°C. For example, if the mainboard temperature is 30°C, then temperature_board_10th_c = 300.
temperature_det_10th_c	int16_t	64	Detection board temperature. The unit is 0.1°C. It is used for fault detection. For example, if the detection board temperature is 30°C, then temperature_det_10th_c = 300.
temperature_other_10th_c	int16_t	48	Reserved.
heater_current_ma	uint16_t	16	Thermal current. The unit is mA.
motor_rpm_1000th	uint32_t	32	Rotation speed of the polygon, unit: Revolutions Per Minute (RPM) / 1000. For example, if the rotation speed of the polygon is 4200 RPM, motor_rpm_1000th = 4200000.
galvo_fpm_1000th	uint32_t	32	Scan speed of the Galvo, unit: RPM/1000. For example, if the rotation speed of the Galvo is 10 RPM, then galvo_fpm_1000th = 10000.
motor_rotation_total	uint64_t	64	Total number of polygon rotations.

galvo_round_total	uint64_t	64	Total number of Galvo scanning iterations.
moisture_index	uint16_t	32	Currently not used. The default value is 0.
window_blockage_index	uint16_t	32	Currently not used. The default value is 0.
motor	uint16_t	96	Polygon current. The unit is mA.
galvo	uint16_t	96	Galvo current. The unit is mA.
laser	uint16_t	96	Laser current. The unit is mA.
galvo_status_client	uint16_t	16	The customized parameters.
galvo_offset_angle_client	uint16_t	16	Currently not used. The default value is 0.
motor_dc_bus_voltage	uint32_t	32	The motor voltage in 1mv.
motor_speed_control_err	uint16_t	16	The polygon motor rotation error between 0 to 2 RPM.
galvo_position_control_err	uint16_t	16	Currently not used. The default value is 0.
unit_current	uint16_t	16	The LiDAR current in 1mA.
apd_bias_feedback	uint16_t	64	The feedback value of APD channel bias circuit, ranging from 0 to 75 V.
accel_x	uint16_t	16	Only for Falcon K1 /Falcon K1 lite. The X-axis value in the triaxial accelerometer. This value is read directly from the device registers with a range of -32768 to 32767. The actual gravitational acceleration is equal to the parameter value divided by 16384, with a range of -2g to 2g.
accel_y	uint16_t	16	Only for Falcon K1 /Falcon K1 lite. The Y-axis value in the triaxial accelerometer. This value is read directly from the device registers with a range of -32768 to 32767. The actual gravitational acceleration is equal to the parameter value divided by 16384, with a range of -2g to 2g.
accel_z	uint16_t	16	Only for Falcon K1 /Falcon K1 lite. The Z-axis value in the triaxial accelerometer. This value is read directly from the device registers with a range of -32768 to 32767. The actual gravitational acceleration is equal to the parameter value divided by 16384, with a range of -2g to 2g.

gyro_x	uint16_t	16	Only for Falcon K1 /Falcon K1 lite. The X-axis value in the three-axis gyroscope. This value is read directly from the device registers with a range of -32768 to 32767. The actual angular velocity is equal to the parameter value divided by 131, with a range of -250 to 250 dps(degree per second).
gyro_y	uint16_t	16	Only for Falcon K1 /Falcon K1 lite. The Y-axis value in the three-axis gyroscope. This value is read directly from the device registers with a range of -32768 to 32767. The actual angular velocity is equal to the parameter value divided by 131, with a range of -250 to 250 dps(degree per second).
gyro_z	uint16_t	16	Only for Falcon K1 /Falcon K1 lite. The Z-axis value in the three-axis gyroscope. This value is read directly from the device registers with a range of -32768 to 32767. The actual angular velocity is equal to the parameter value divided by 131, with a range of -250 to 250 dps(degree per second).
accel_unit_x	uint32_t	32	Only for Falcon K1 /Falcon K1 lite. The X-axis value in the triaxial accelerometer. This value is converted with a range of -200000 to 200000. The actual gravitational acceleration is equal to the parameter value divided by 100000, with a range of -2g to 2g.
accel_unit_y	uint32_t	32	Only for Falcon K1 /Falcon K1 lite. The Y-axis value in the triaxial accelerometer. This value is converted with a range of -200000 to 200000. The actual gravitational acceleration is equal to the parameter value divided by 100000, with a range of -2g to 2g.
accel_unit_z	uint32_t	32	Only for Falcon K1 /Falcon K1 lite. The Z-axis value in the triaxial accelerometer. This value is converted with a range of -200000 to 200000. The actual gravitational acceleration is equal to the parameter value divided by 100000, with a range of -2g to 2g.
gyro_unit_x	uint32_t	32	Only for Falcon K1 /Falcon K1 lite. The X-axis value in the three-axis gyroscope. This value is converted with a range of -25000000 to 25000000. The actual angular velocity is equal to the parameter value divided by 131, with a range of -250 to 250

			dps(degree per second).
gyro_unit_y	uint32_t	32	Only for Falcon K1 /Falcon K1 lite. The Y-axis value in the three-axis gyroscope. This value is converted with a range of -25000000 to 25000000. The actual angular velocity is equal to the parameter value divided by 131, with a range of -250 to 250 dps(degree per second).
gyro_unit_z	uint32_t	32	Only for Falcon K1 /Falcon K1 lite. The Y-axis value in the three-axis gyroscope. This value is converted with a range of -25000000 to 25000000. The actual angular velocity is equal to the parameter value divided by 131, with a range of -250 to 250 dps(degree per second).
gyro_temp	uint16_t	16	Only for Falcon K1 /Falcon K1 lite. The imu sensor temperature data. The actual temperature is equal to the parameter value divided by 10, with a range of -40 to 105 °C.
reserved	uint16_t	320	Currently not used. The default value is 0.

4 Packet capture and parsing

Innovusion.lua is a plug-in for checking the UDP data format in LiDAR. The function of innovussion.lua plug-in is as follows.

- Identify the UDP packets of port 8010 as LiDAR packets.
- Divide the content of a UDP packet into innovusion:status and innovusion:data, and parse the specific values in the data packet.

This section is mainly about using the Wireshark tool and innovusion.lua to capture and parse LiDAR data packets.

1. Download and install Wireshark.

Caution

➢ Wireshark download address: https://www.wireshark.org/

- Windows system: Directly install Wireshark.
- Ubuntu system: Execute the following command to install Wireshark.

```
sudo apt-get install wireshark
```

2. Download and install the innovusion.lua.
   - Windows system: Copy the innovusion.lua to the <Wireshark installation directory>/plugins.
   - Ubuntu system: Execute the following commands to copy the innovusion.lua to the following directory.

```
cd /<innovusion.lua directory>  //Execute the cd command to enter the directory where innovusion.lua is located.
sudo cp innovusion.lua /usr/lib/x86_64-linux-gnu/wireshark/plugins  //Copy the innovusion.lua to the target directory.
```

Caution

Innovusion.lua download address: https://innovusioncn.atlassian.net/wiki/spaces/IPD/pages/1585611983/03-Tools+Management

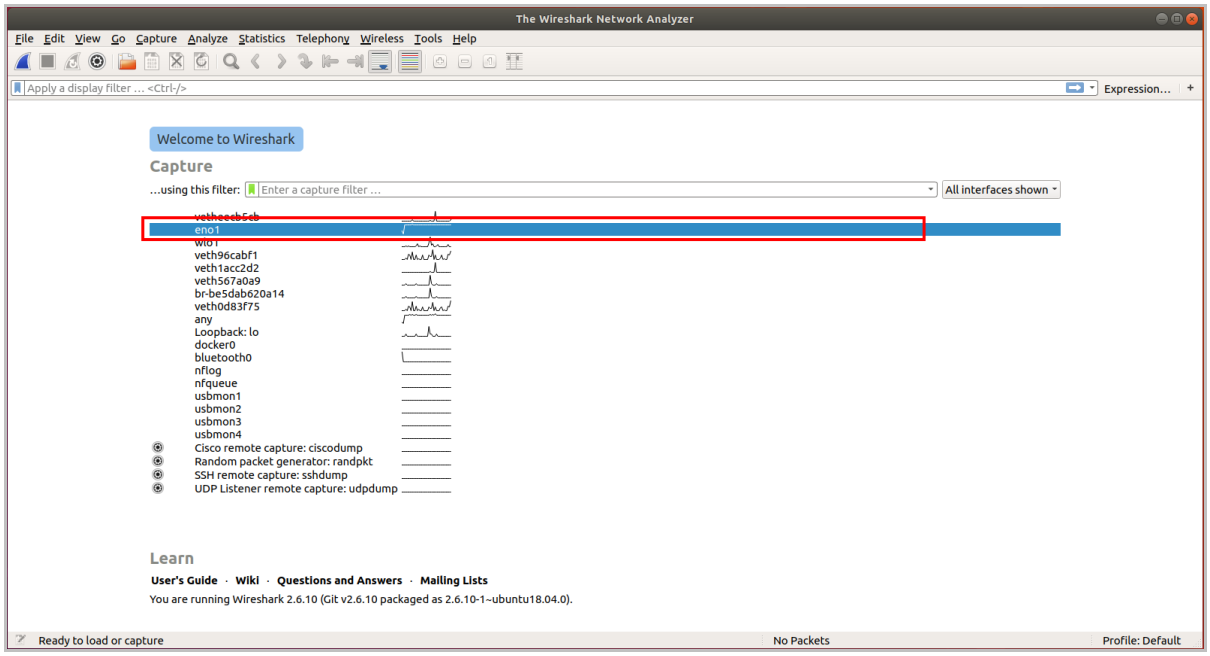
3. Turn on the LiDAR. The LiDAR has no power switch and can work after being powered on.
4. Connect the computer to LiDAR.
5. Change the computer IP address so that the computer IP address and the LiDAR IP address are in the same network segment.

Caution

➢The default IP address of Falcon K/P is 172.168.1.10. The default IP address of Falcon AI is 192.168.1.220.

➢ It is recommended that you ping the IP address of the LiDAR to ensure that the LiDAR is connected properly.

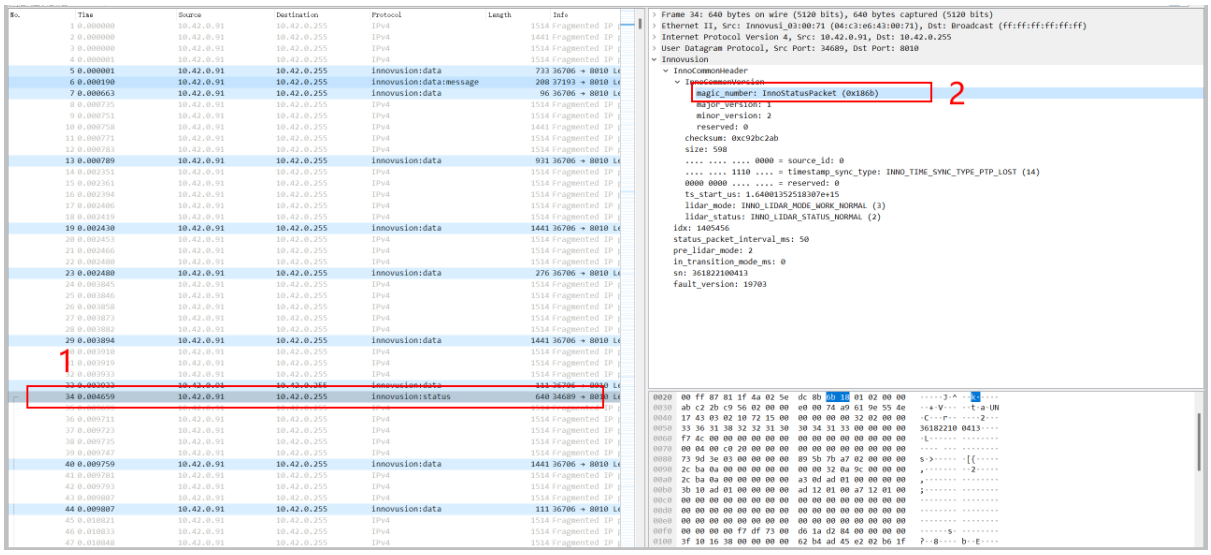
6. Select **Analyze > Reload Lua Plugins** to reload the innovusion.lua.
7. Double-click the network interface connecting to the LiDAR on the home page.

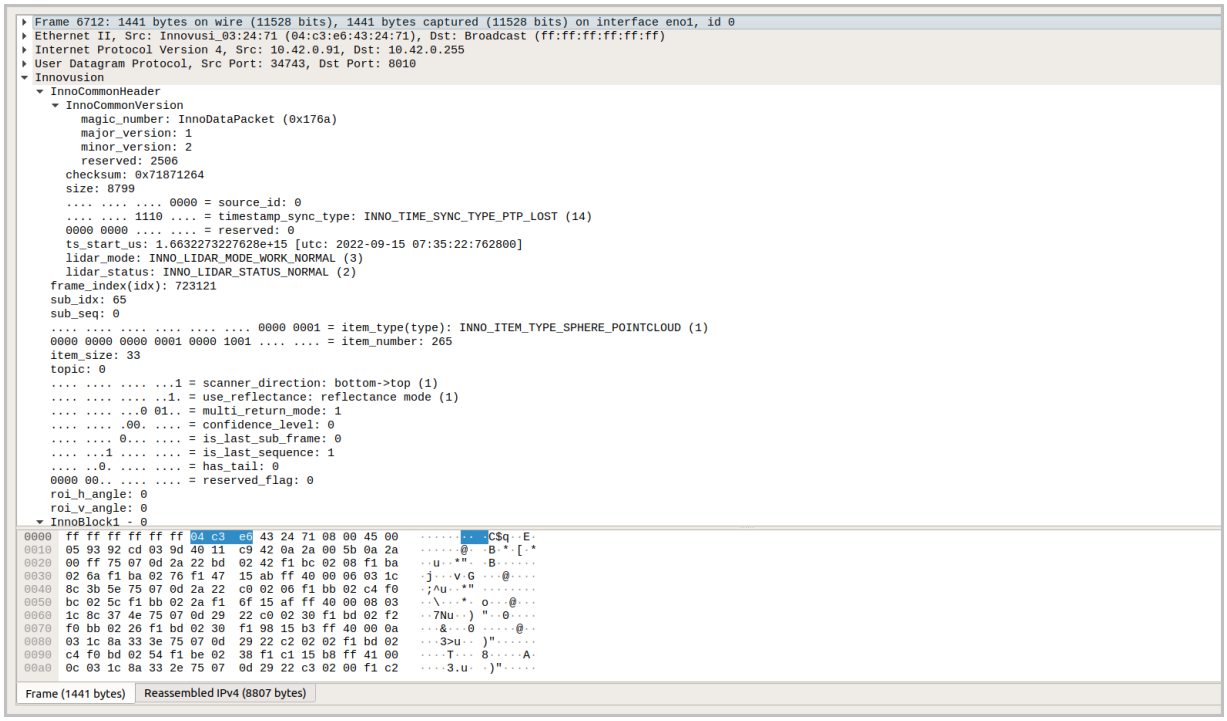


Caution

You can use the **ifconfig** command to check the computer network interface on Ubuntu.

8. Double-click the data packet to check the packet capture result.





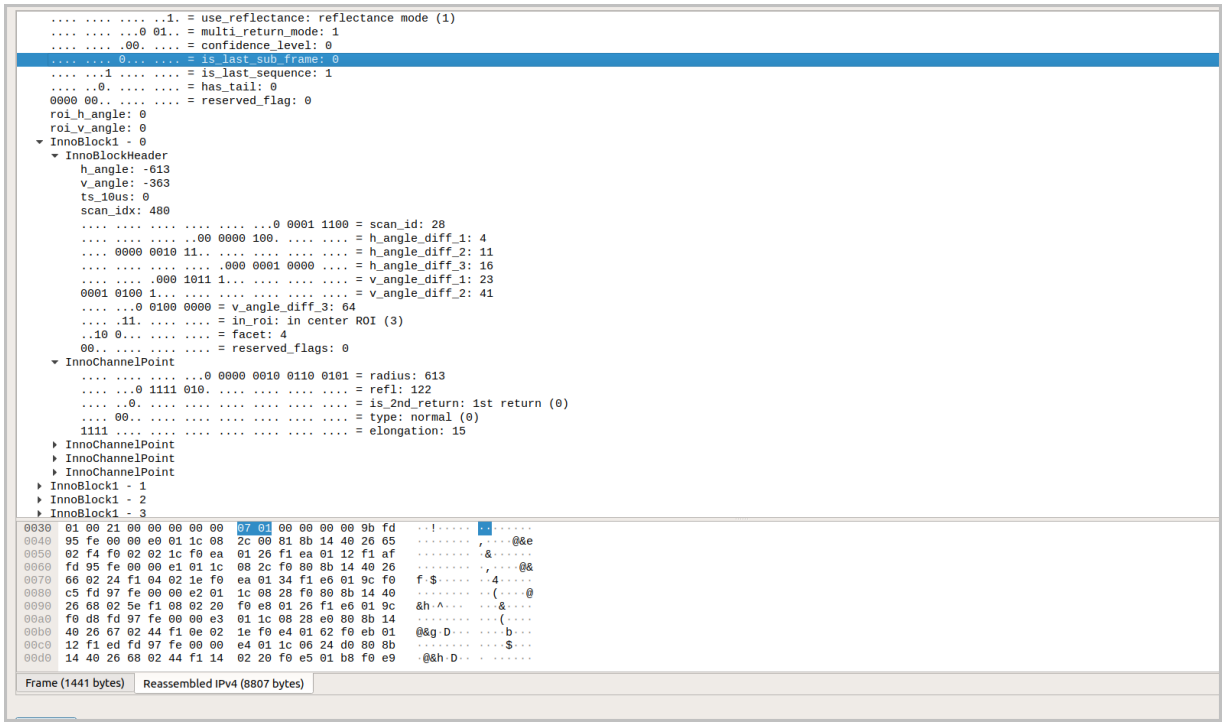


Table 14 Revision History

Version	Date	Remarks
V4.0	20240820	Fix bug
V3.0.7	20240318	Change logo
V1.0	20221110	Draft