
Livox wiki

Release 0.1

livox

Apr 16, 2024

LIVOX INTRODUCTION

1	Product	3
1.1	New product introduction	3
1.2	Other product introduction	5
2	Introduction to Livox Scanning Features	11
2.1	Point Cloud FOV Coverage	11
2.2	Unique Scanning Pattern	12
2.3	Livox Specification	12
3	Livox Point Cloud and Coordinate System	15
3.1	Point Cloud Properties	15
3.2	Coordinate System	21
3.3	Point Cloud Data	22
4	Showcase	25
4.1	Customer Success Story	25
4.2	Livox Technical Applications	27
5	New Product	31
5.1	HAP	31
5.2	Mid-360	31
5.3	Public Information	58
6	Other Product	67
6.1	Single lidar data acquisition	67
6.2	Multi-lidars data collection (using Livox Hub)	69
6.3	Data format and conversion	72
6.4	Lidar Calibration	75
6.5	Livox SDK	75
7	Summary of Official Open Source Materials	77
7.1	Drive	77
7.2	Tools	77
7.3	Open Source Algorithm	78
7.4	Documentation and Manuals	79
7.5	Point Cloud Data	80
7.6	Livox Vehicle Platform	80
8	Summary of Unofficial Open Source Materials	83
8.1	Paper	83

9	Dataset	89
9.1	Livox Simu-dataset v1.0 CN	89
10	FAQ	91



Livox Wiki

Livox Technology Company Limited is an independent company founded in 2016 through DJI's Open Innovation Program. Livox is committed to breaking through the current technological barrier in the LiDAR industry and helping customers incorporate LiDAR sensors into efficient commercialization of their projects. Our products have been sold to 26 countries and are being widely applied in industries such as autonomous driving, robots, 3D mapping, smart city, safety and more.

PRODUCT

1.1 New product introduction

1.1.1 HAP

Developed by Livox for over two years, Livox HAP is our first automotive-grade lidar designed for intelligent driving assistance systems. HAP meets up to 74 items of the automotive-grade reliability tests, and began serial production for OEMs such as Xpeng and FAW Jiefang since 2021 at Livox's own brand-new automotive-grade manufacturing center, helping customers build industry-leading smart driving systems that deliver superior safety and comfort. [HAP](#)



1.1.2 Mid-360

Mid-360 is the latest generation of Livox LiDAR for low speed robotics. This new product inherits the cost-effectiveness of the Livox Mid series and delivers 3D perception in 360 degrees. Compact and lightweight, this LiDAR module is easy to install. The Mid-360 is optimized based on the navigation and obstacle avoidance requirements of mobile robots. This allows you to use the LiDAR with algorithms to deliver a wide range of functions such as SLAM and obstacle avoidance. [Mid-360](#)



1.2 Other product introduction

1.2.1 New Mid-70

The Mid-70 is another component adding to the overall market appeal for Livox with tailored use applications such as low-speed autonomous driving and mobile robotics. By significantly expanding the vertical viewing angle, reducing blind spots and increasing close-range precision, the Mid-70 offers a thorough and comprehensive point cloud data quality to ensure safer autonomy. [Mid-70](#)



1.2.2 New Avia

The Livox Avia combines compact and lightweight design with improved detection range and efficiency, featuring an FOV greater than 70°. Its triple-echo and dual-scanning modes are ideal for use cases such as mapping and low-speed autonomous driving. [Avia](#)



1.2.3 Mid Series

The Livox Mid-40 LiDAR sensor is incredibly cost effective. It detects objects as far as 260 meters away and uses an advanced non-repetitive scanning pattern to deliver highly accurate details in the FOV. A compact body enables users to easily embed units into existing designs for greater flexibility and performance. The Mid-40 has been mass-produced and is ready to ship immediately to facilitate uses in autonomous driving, robotics, mapping, security, and other areas from small batch testing to large-scale applications. For more information [Mid Series](#)



1.2.4 Horizon

The Livox Horizon is a high-performance LiDAR sensor built for Level 3 and Level 4 autonomous driving. It has a long-distance detection range of up to 260 meters, high-density point clouds, and is so small that can be embedded easily into vehicles. It is also incredibly accessible, with a price point that is just 1% of traditional LiDAR sensors. The Horizon is poised to reinvent the LiDAR industry, bringing the mass production of autonomous vehicles into reality with innovative technologies and enormous cost advantages. For more information [Horizon](#)



1.2.5 Tele-15

Made for advanced long-distance detection, the Livox Tele-15 offers a compact size, high-precision, and durability. It allows vehicles to detect and avoid obstacles well in advance when moving at high speeds. For more information [Tele-15](#)



INTRODUCTION TO LIVOX SCANNING FEATURES

2.1 Point Cloud FOV Coverage

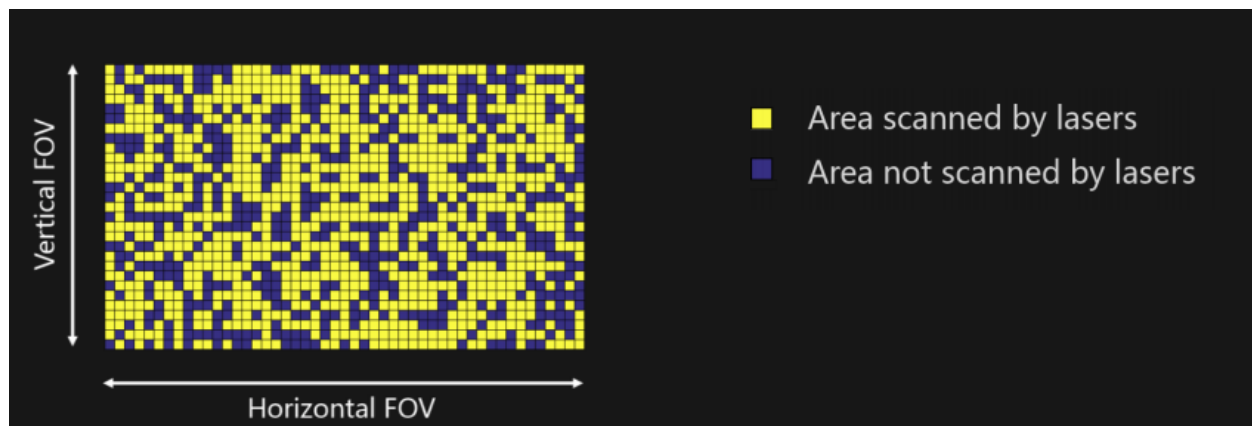
The lidar field of view refers to the area detected by the lidar, and all target objects in this area may be detected. However, not all objects in the field of view can be detected. The reasons include:

- The target object is too far away from the lidar or is not within the detection range, causing the reflected light of the object to be too weak to be recognized.
- During a certain period of time, because the lidar can only illuminate the objects on its scanning path and measure its distance, objects that are not on the lidar scanning path within this period of time cannot be detected.

If the scanning method can be optimized to increase the proportion of the area covered by the laser beam path in the field of view, the probability of the second situation described above can be reduced. Therefore, the area of the laser irradiated area in the field of view is related to the measurement performance of the lidar. This indicator can be quantitatively characterized by the proportion of the area detected by the laser in the field of view of the lidar, which is the field of view coverage. The calculation formula is as below:

$$C = \frac{\text{Total area illuminated by laser beams}}{\text{Total area in FOV}} \times 100\%$$

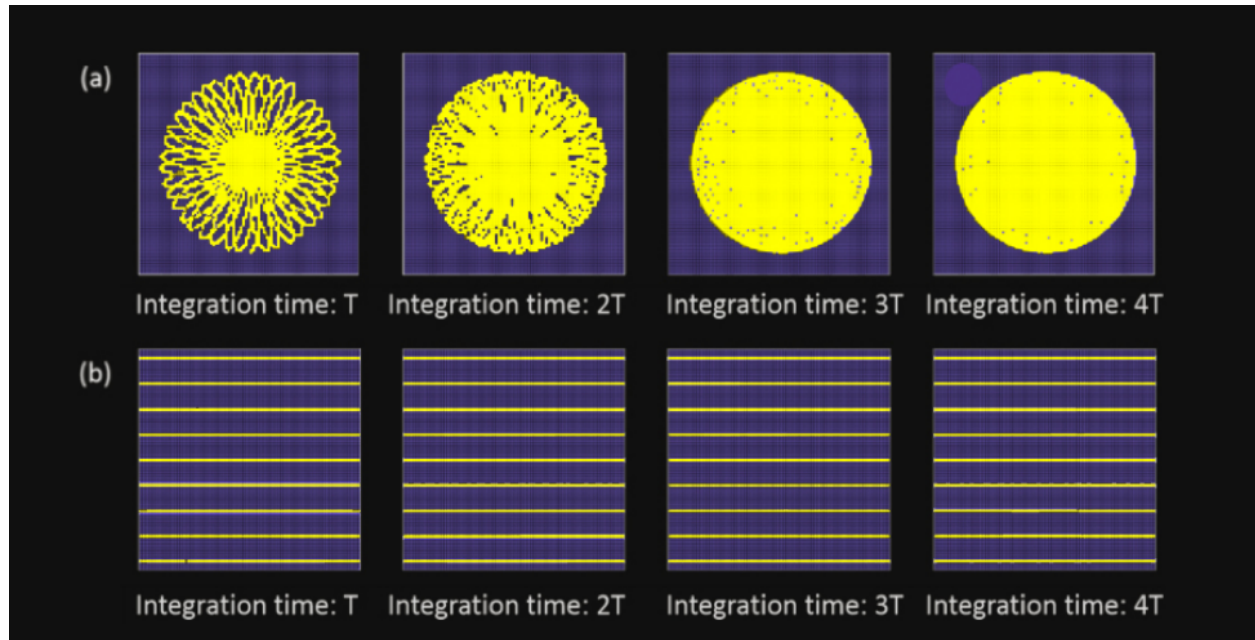
The following figure shows the scanning field of view of Lidar. The yellow squares indicate the scanned area, and the blue squares indicate the unscanned area:



2.2 Unique Scanning Pattern

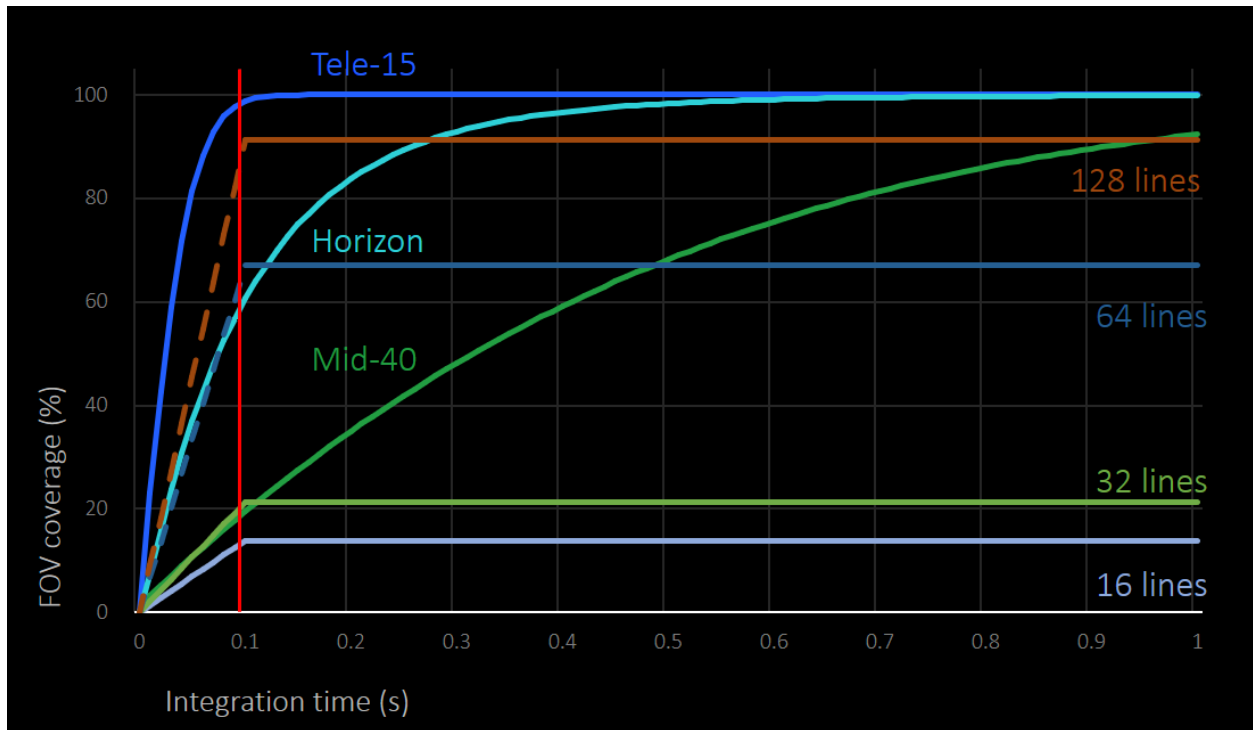
Traditional LiDAR sensors that use mechanical scanning methods often adopt 360-degree linear scanning patterns, with the motor driving the entire laser head to make circling motions. One of the problems with this method is that there are always blind spots, regardless of how long the scanning lasts. This means that some target objects may go undetected.

Livox LiDAR sensors feature non-repetitive scanning patterns which differ significantly from the repetitive linear scanning offered by traditional LiDAR sensors. The areas scanned inside the FOV of a Livox sensor grow the longer the integration time, increasing the likelihood of objects and other details within the FOV being detected. As shown in the images below, a Livox Mid-40 can scan almost the entire area within its FOV as integration time increases.



2.3 Livox Specification

Livox's lidars: Mid-40, Horizon and Tele-15 all use non-repetitive scanning methods, which have a higher field of view coverage, and as the integration time becomes longer, the field of view coverage will also increase significantly, until close to 100%, that is, almost all areas in the field of view will be irradiated by the laser beam, as shown in the following figure:



According to the calculation, the FOV coverage of Livox Mid series sensors within 100 ms is similar to that of traditional 32-line LiDAR sensors.

The Livox Horizon can reach a FOV coverage of 60% within 100 ms, which is the equivalent to most 64-line LiDAR sensors.

The Livox Tele-15 can even reach a FOV coverage of 99.8% within 100 ms.

LIVOX POINT CLOUD AND COORDINATE SYSTEM

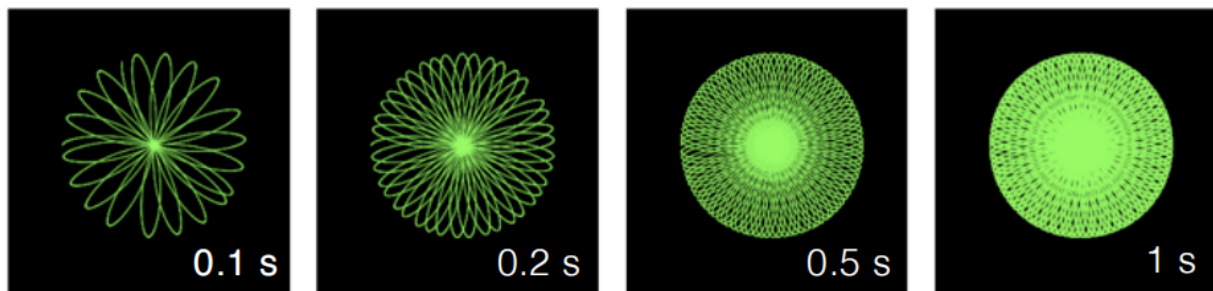
3.1 Point Cloud Properties

Mid Series

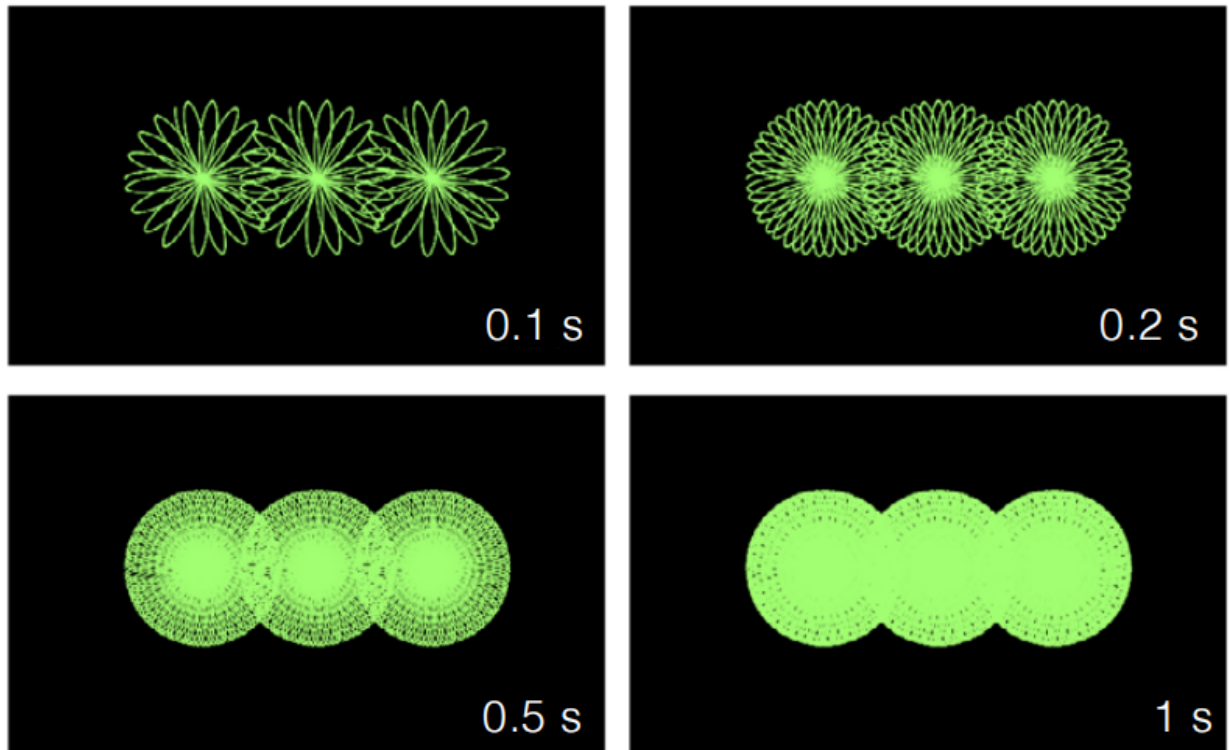
Livox_LiDAR can accurately perceive the complex surrounding environment and provide detailed point cloud data information. It is suitable for advanced driver assistance systems, unmanned vehicles, robots, helicopters, UAV obstacle avoidance and safety inspections. Let's take Mid series and Horizon series as examples to introduce the characteristics and parameters of Livox_LiDAR point cloud:

Livox Mid-40 and Livox Mid-100 have a higher field of view coverage by using non-repetitive scanning technology, and as the integration time increases, the field of view coverage will also increase significantly, capturing more details. The figure below shows the point cloud diagrams of Livox Mid-40 and Mid-100 at different integration times (0.1s, 0.2s, 0.5s and 1s respectively). The point cloud diagrams of each Livox Mid-100 point cloud image can be regarded as three Mid-40.

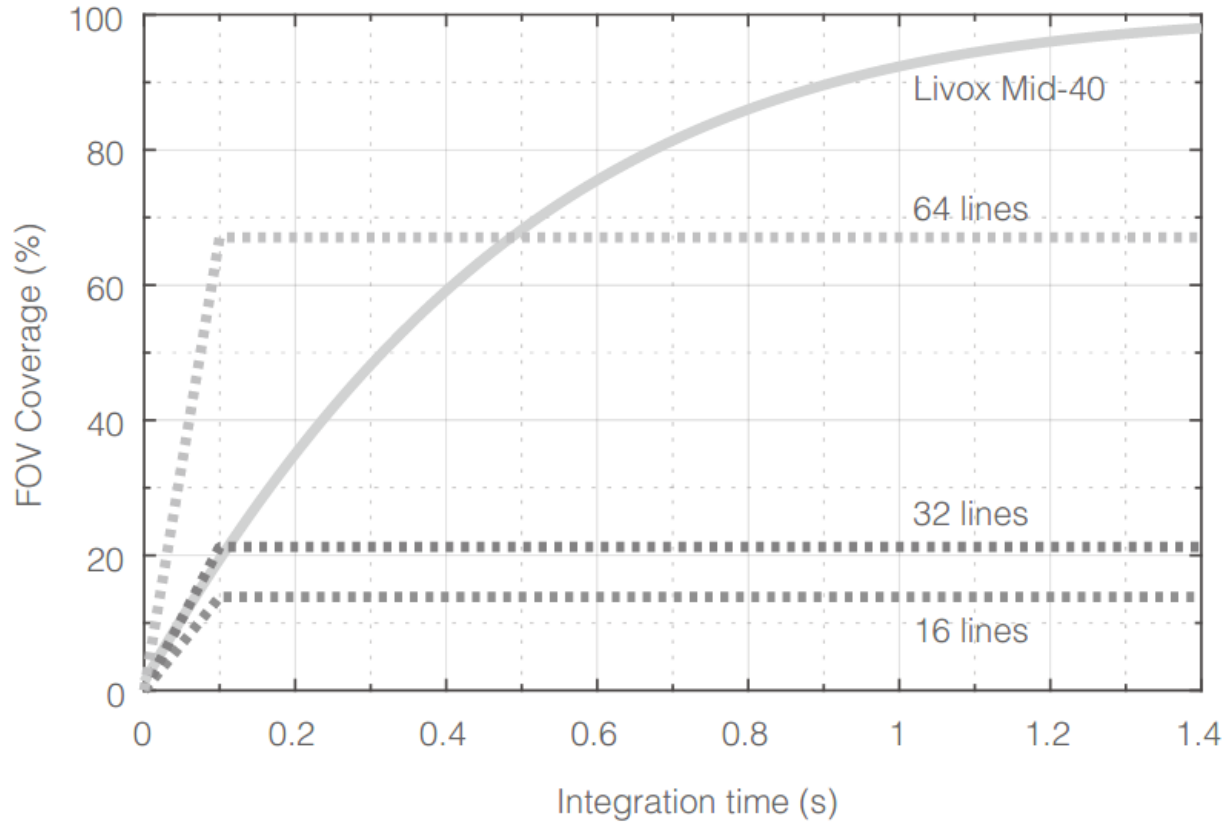
The point cloud effect diagram of Livox Mid-40 in different integration time is as follows:



The point cloud effect diagram of Livox Mid-100 in different integration time is as follows:



The following figure shows the coverage of the field of view of the Livox Mid-40 under different integration times, and the relevant data of several traditional multi-line mechanical lidar on the market. It can be seen from the figure that when the integration time is less than 0.1s, the field of view coverage performance of Livox Mid-40 is equivalent to that of a 32-line product; when the integration time increases, the field of view coverage of Livox Mid-40 significantly improved while the field of view coverage of a 32-line product remained unchanged. When the integration time is 0.5s, the Mid-40 field of view coverage is equivalent to a 64-line product; when the integration time continues to increase to 1.4s, the Mid-40 field of view coverage will be close to 100%, that is, almost all areas in the field of view will be hit by the laser beam.



Mid Series Point Cloud Specifications:

Laser Wavelength	905 nm
Laser Safety	Class 1 (IEC 60825-1:2014) eye safe
Detection Range (@100 klx)	90 m @ 10% reflectivity 130 m @ 20% reflectivity 260 m @ 80% reflectivity
FOV	Livox Mid-40: 38.4° (circular) Livox Mid-100: 98.4° (Horizontal) × 38.4° (Vertical)
Range Precision (1σ @ 20 m)	2 cm
Angular Accuracy	< 0.1°
Beam Divergence	0.28° (Vertical) × 0.03° (Horizontal)
Point Rate	Livox Mid-40: 100,000 points/s Livox Mid-100: 300,000 points/s
False Alarm Ratio (@100 klx)	< 0.01%

HORIZON

The Horizon utilizes Livox's unique non-repetitive scanning technology and multi-laser and multi-APD DL-Pack technology. This ensures a high-density point cloud which is approximately three times denser than the Livox Mid series within the same period. Over time, the coverage inside the FOV increases significantly and reveals more detailed information of the surroundings.

Figure below displays the point cloud allocation of the Horizon inside the FOV within 0.1s. In the center of the FOV, the scanning density is denser with the average spacing of lines being 0.2° (most line spacing is from 0.1° to 0.3°), which is much denser than traditional 64-line LiDAR sensors, in which the spacing of lines is 0.3° to 0.6° . The two circular areas on both sides have a lower scanning density, with an average line spacing of 0.4° (most line spacing is between 0.2° to 0.8°), rivaling traditional 64-lines LiDAR sensors within 0.1s

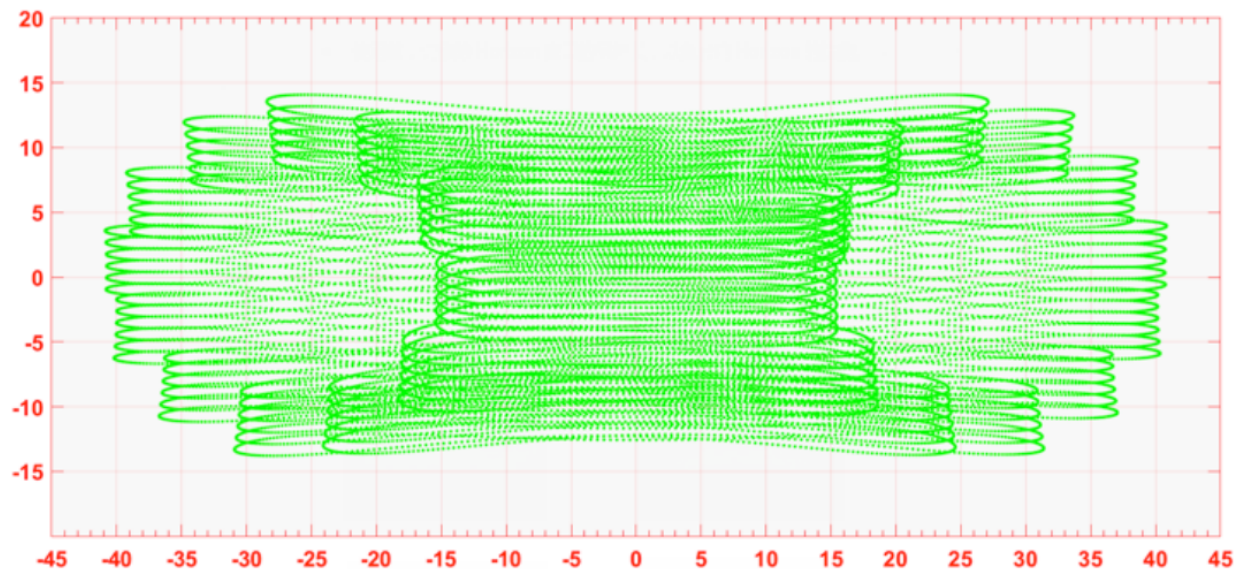


Figure below displays the typical point cloud patterns of the Horizon.

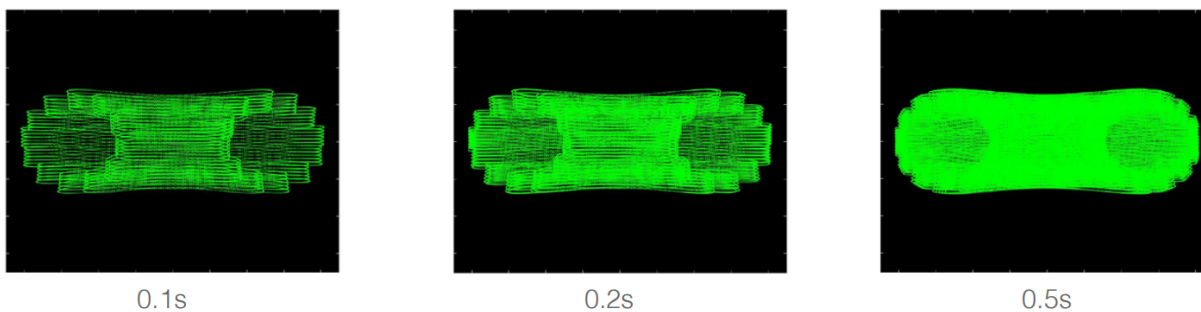
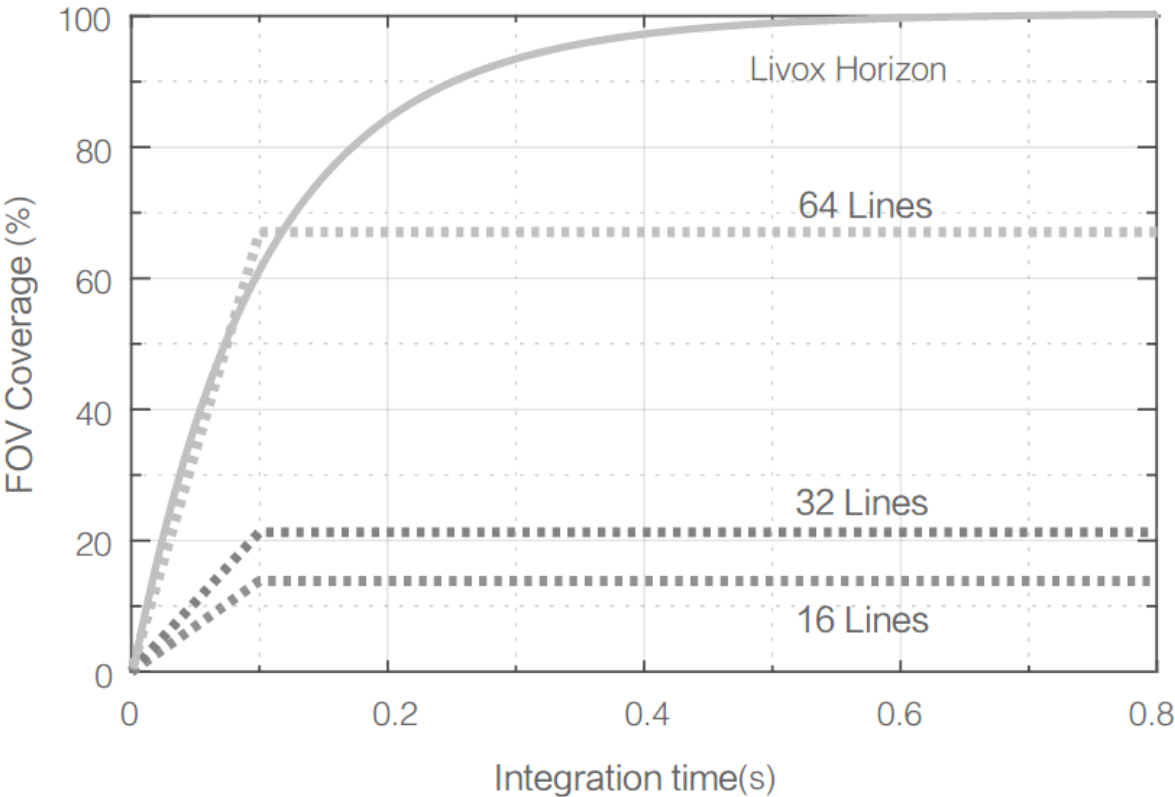


Figure below displays the FOV coverage of the Horizon compared with other non-Livox LiDAR sensors that use common mechanical scanning methods. The diagram shows that when the integration time is less than 0.1 seconds, the FOV coverage of the Horizon approaches 60%, similar to the 64-line LiDAR sensor. As the integration time increases to 0.5 seconds, the FOV coverage approaches 100%, so almost all areas are illuminated by laser beams.



Horizon Point Cloud Specifications:

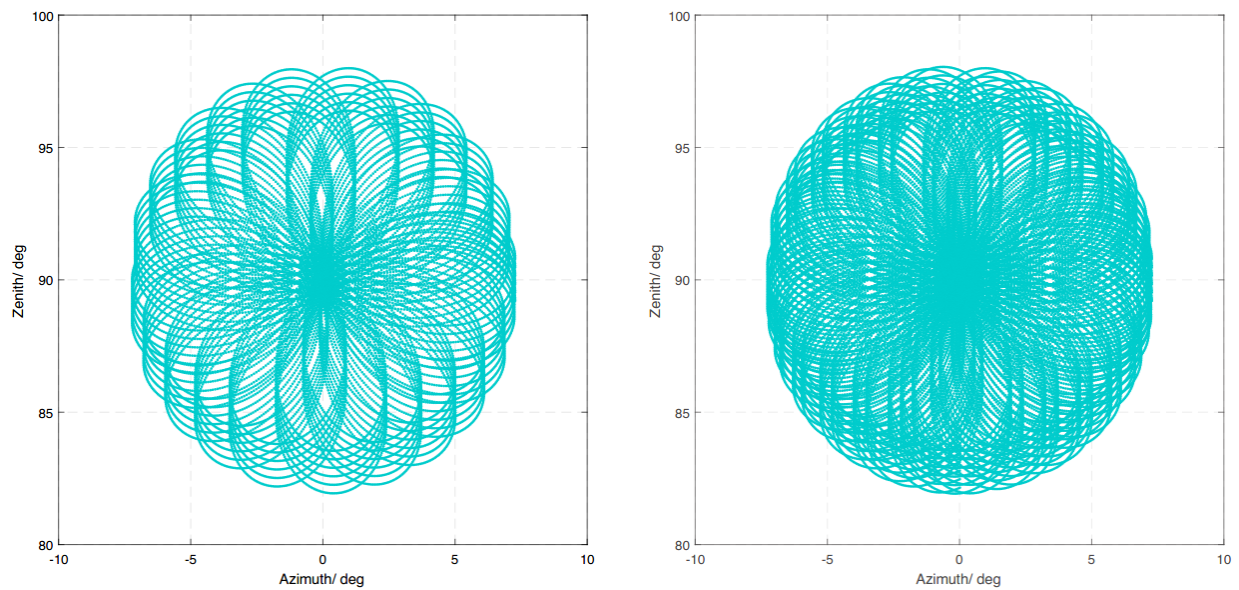
Laser Wavelength	905 nm
Laser Safety	Class 1 (IEC 60825-1:2014) (Safe for eyes)
Detection Range (@100 klx)	90 m @ 10% reflectivity, 130 m @ 20% reflectivity 260 m @ 80% reflectivity
FOV	81.7° (Horizontal) × 25.1° (Vertical)
Distance Random Error	(1σ @ 20 m) < 2 cm
Angular Random Error	1σ < 0.05 °
Beam Divergence	0.28° (Horizontal) × 0.03° (Vertical)
Point Rate	240,000 points/s (first or strongest return) 480,000 points/s (dual return)
False Alarm Ratio (@100 klx)	< 0.01%

TELE-15

The Tele-15 utilizes Livox’s unique non-repetitive scanning technology and multi-laser and multi-APD DL-Pack technology. This ensures a high-density point cloud which is approximately five times denser than the Livox Mid-40 within the same period.

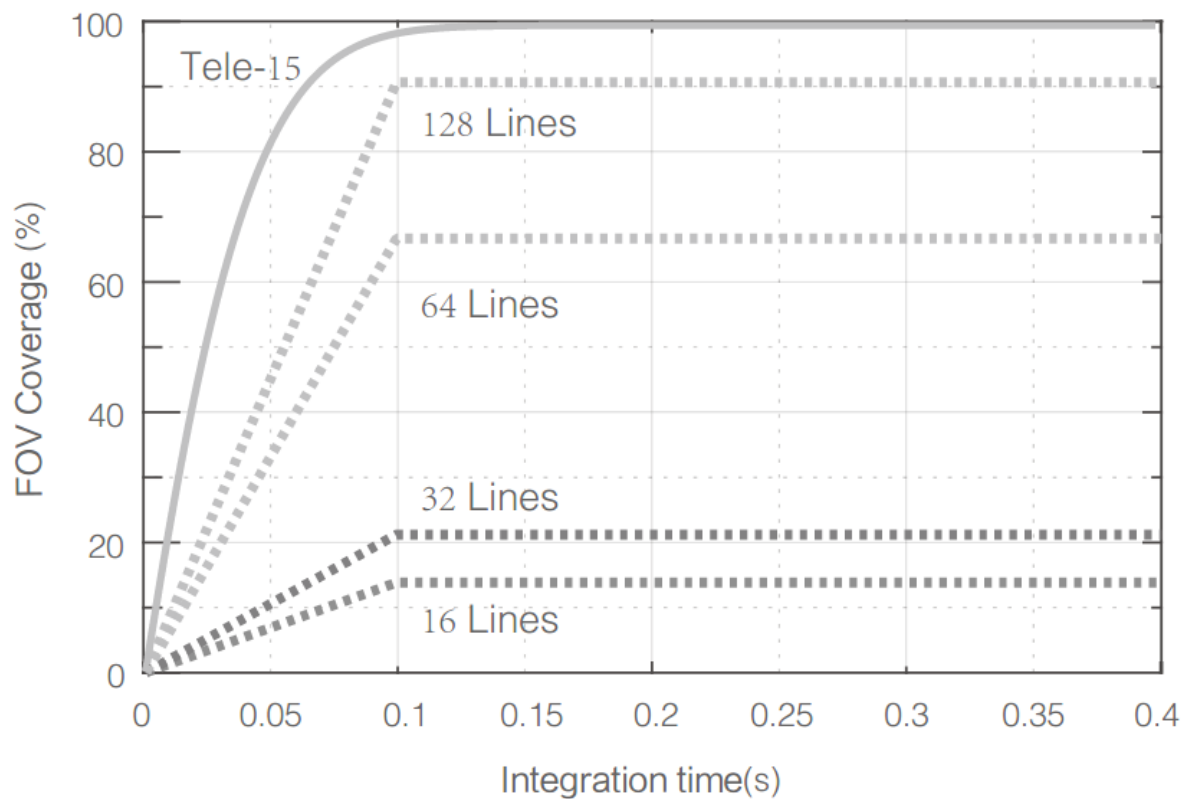
Figure below displays the point cloud data of the Tele-15 inside the FOV within 0.1s. In the center of the FOV, the scanning density is as dense as traditional 300-line LiDAR sensors. In other areas of the FOV, the scanning density is significantly denser than 100-line LiDAR sensors. Within Tele-15’s FOV, the overall scanning density rivals traditional

128-lines LiDAR sensors within 0.1s.



According to the definition of Livox coordinate system, the azimuth range is $0^{\circ}\sim 360^{\circ}$, where $-10^{\circ}\sim 0^{\circ}$ is equivalent to $350^{\circ}\sim 360^{\circ}$.

Figure below displays the FOV coverage of the Tele-15 compared with other non-Livox LiDAR sensors that use common mechanical scanning methods. The diagram shows that when the integration time is 0.1 seconds, the FOV coverage of the Tele-15 is approximately 99%, higher than the 128-line LiDAR sensor.



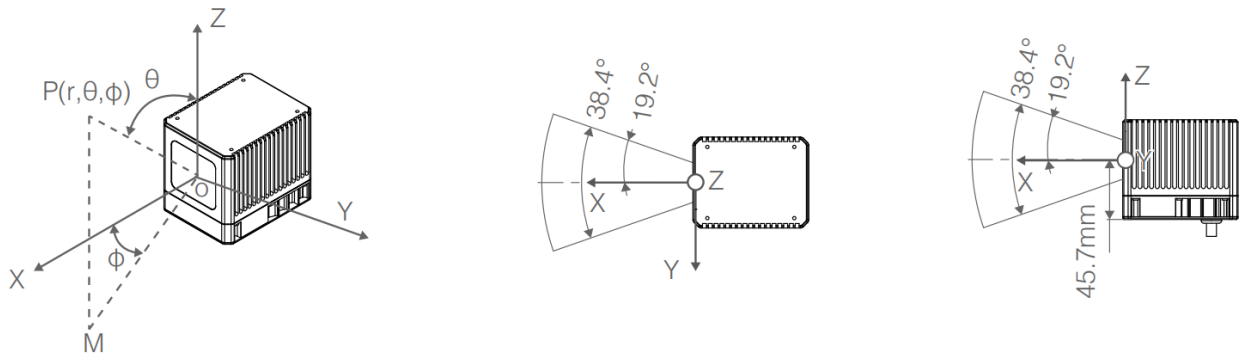
TELE-15 Specifications:

Laser Wavelength	905 nm
Laser Safety	Class 1 (IEC 60825-1:2014) (Safe for eyes)
Detection Range (@100 klx)	200 m @ 10% reflectivity, 500 m @ 80% reflectivity
FOV	14.5° (Horizontal) × 16.2° (Vertical)
Distance Random Error (1σ)	< 2 cm (5 to 70 m), < 4 cm (70 to 120 m) @ 10% reflectivity < 1 cm (5 to 150 m), < 2 cm (> 150 m) @ 80% reflectivity
Angular Random Error (1σ)	< 0.03 °
Beam Divergence	0.02° (Horizontal) × 0.12° (Vertical)
Point Rate	240,000 points/s (first or strongest return) 480,000 points/s (dual return)
False Alarm Ratio (@100 klx)	< 0.01%

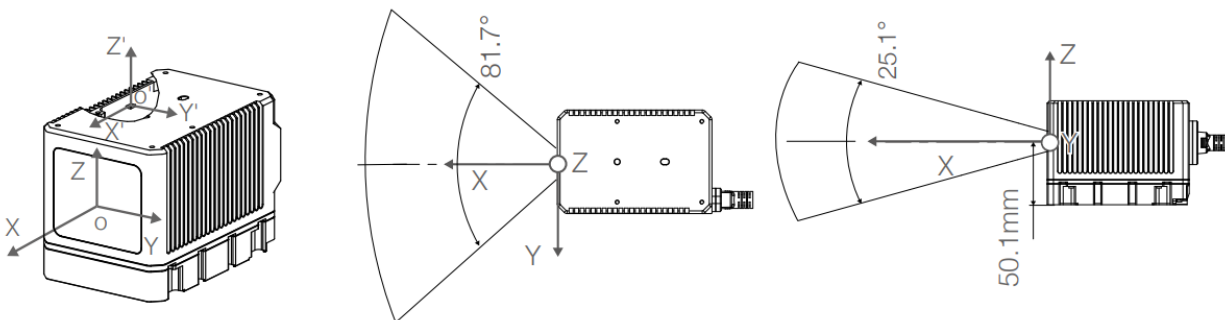
3.2 Coordinate System

In multi-sensor intelligent body navigation, multiple sensors are usually installed for data fusion and calculation. It is necessary to understand that the Cartesian coordinates of Livox lidars are defined as follows:

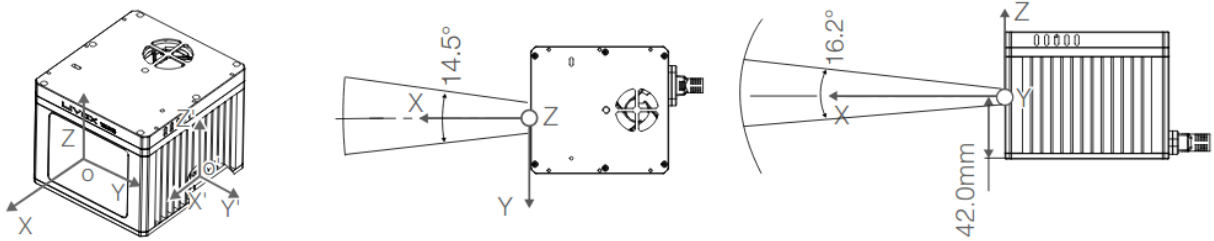
- Livox Mid-40 coordinate definition:



- The IMU is built-in in Livox Horizon, the point cloud coordinate system O-XYZ and the IMU coordinate system O-X'Y'Z' are defined as shown in the figure below:



- The IMU is built-in in Livox Tele-15, the point cloud coordinate system O-XYZ and the IMU coordinate system O-X'Y'Z' are defined as shown in the figure below:



Among them, the coordinates of the IMU coordinate origin O' on the point cloud coordinate system O-XYZ are (-84.7, -42.5, -35.3) (unit: mm)

3.3 Point Cloud Data

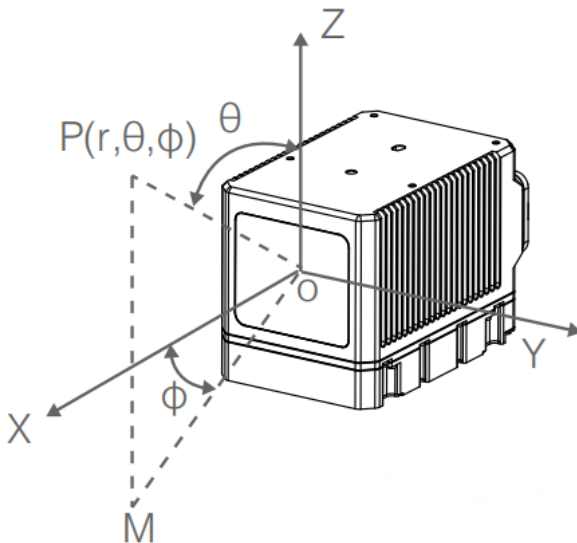
The point cloud data is the sum of all the point clouds detected on the surface of the measured object in the field of view of lidar. Each point cloud contains the following information.

- Target reflectivity:

Expressed from 0 to 255. Among them, 0 to 150 correspond to diffuse scattering objects with reflectivity between 0 and 100%; and 151 to 255 correspond to total reflection objects.

- Coordinate information:

The coordinate information of Livox Horizon can be expressed as rectangular coordinates (x, y, z) or spherical coordinates (r, θ , ϕ). The corresponding relationship between rectangular coordinates and spherical coordinates is shown in the figure below. If there is no detected object ahead or the detected object exceeds the range (for example, 600m), the point cloud output is (0, 0, 0) in the Cartesian coordinate system; the point cloud output is (0, ,) in the spherical coordinate system. The relationship between spherical coordinates and rectangular coordinates in point cloud data is as follows:



$$x = r \times \sin(\theta) \times \cos(\phi)$$

$$y = r \times \sin(\theta) \times \sin(\phi)$$

$$z = r \times \cos(\theta)$$

- Tag: Indicates the return type of the laser and if the point detected is a noise. The format of the tag is as shown below

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Reserved		Return number: 00:return 0 01:return 1 10:return 2 11:return 3		Point property based on intensity: 00:Normal 01:High confidence level of the noise 10:Moderate confidence level of the noise 11:Reserved		Point property based on spatial position: 00:Normal 01:High confidence level of the noise 10:Moderate confidence level of the noise 11:Low confidence level of the noise	

Each tag is composed of one byte. In this byte, bit7 and bit6 are Group 1, bit5 and bit4 are Group 2, bit3 and bit2 are Group 3 while bit1 and bit0 are Group 4. Group 2 indicates the return sequence of the sampling point. Featuring a coaxial optical path, the Horizon itself will generate a laser return even if there is no detectable object around. This return is recorded as return 0. After that, if there is any object within the detectable range, the first laser that returns to the Horizon is recorded as return 1, and then return 2, and so on. If the object is too close to the Horizon such as 1.5 m away, the first effective return will be merged into return 0, and be recorded as return 0. Group 3 indicates if the sampling point is a noise based on the intensity of the return. Normally, the intensity of the returns of the noises generated due to the interference of atmosphere particles such as dust, rain, fog, and snow is quite low. Therefore, the noises are divided into two categories based on the intensity of the return received: “01” stands for low intensity of the return, indicating that the samples have a high possibility of being noises such as dust; “10” stands for moderate intensity of the return, indicating that the samples have a moderate possibility of being noises such as rain and fog. The lower the confidence level of the sample is, the lower the possibility that it being a noise is. Group 4 indicates if the sampling point is a noise based on its spatial position. Normally, when the Horizon LiDAR sensors detect two close objects, there will be some thread-like noises between the two objects. The noises are divided into three categories. The lower the confidence level of the noise is, the lower the possibility that it being a noise is.

SHOWCASE

4.1 Customer Success Story

AutoX: Democratizing Autonomy with Latest Livox LiDAR Technology

“We’ve been testing and using a great number of lidar sensors available on the market and we are excited to find one that offers a good combination of performance, reliability and cost,” said Jianxiong Xiao also known as Professor X, the CEO and Founder of AutoX. “We are excited to see Livox is redefining the lidar market with their innovative technology and I am glad to announce that AutoX will be using Horizon and Tele-15 in our new robotaxi fleets in 2020.” [Learn More](#)



2.Livox X CiDi: Reshape Heavy Truck Logistics Industry

“At present, CiDi’s L4 smart heavy trucks are already capable of autonomous driving at 100 kilometers per hour on highways. We are very confident in Livox products and are satisfied with the technical support and services they provide. With Livox LiDAR, we are very much looking forward to the rapid implementation of commercial heavy truck autonomous driving solutions,” said Matan, CEO of CiDi. [Learn More](#)



3.Livox X Gaussian: Improving The Quality of Life

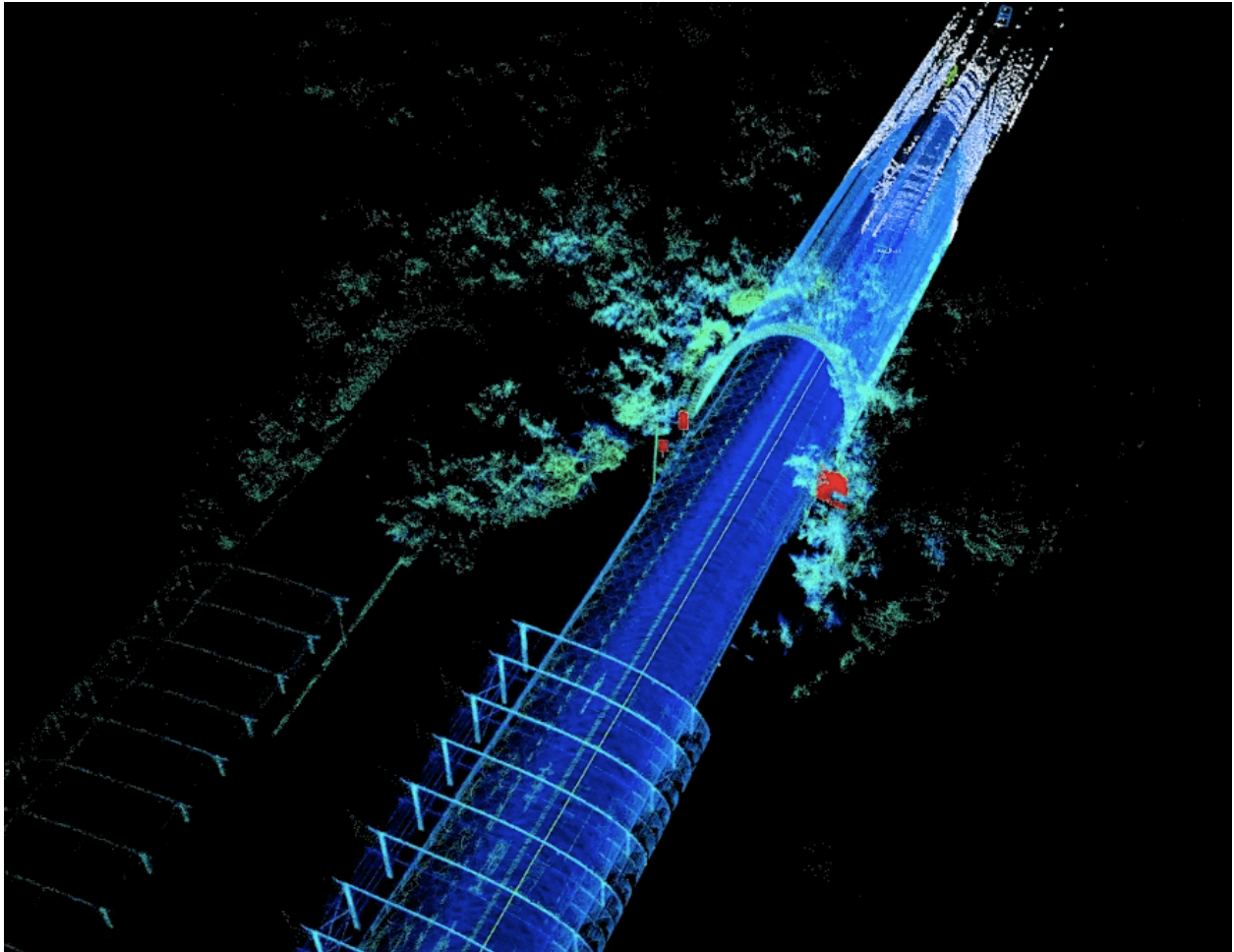
Through cooperation with Livox, we have more confidence in launching more cost-effective service robots.” Song Le, Director of Algorithm, Gaussian Robotics. [Learn More](#)



4.2 Livox Technical Applications

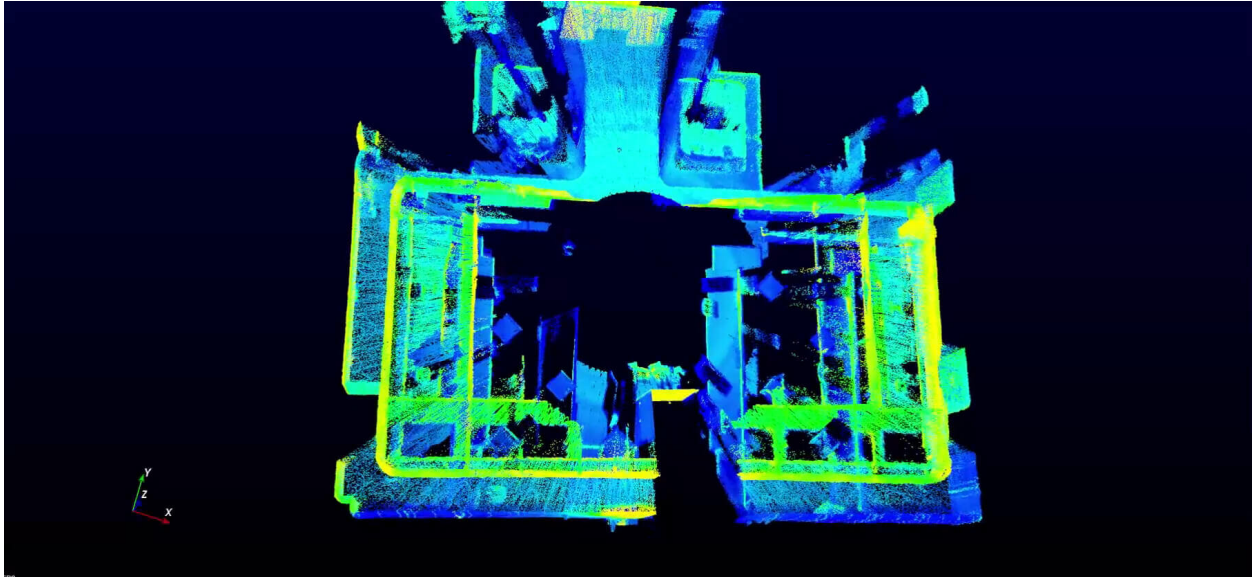
1.LOAM-Livox Makes Easier Lidar by Shortening the Algorithm Development Cycle

The demonstration of actual effect of Livox Horizon proves the feasibility of applying its non-repetitive scanning of point cloud data to existing high-performance algorithms. The cost reduction far surpasses the inevitable cost of algorithm swapping. [Learn More](#)



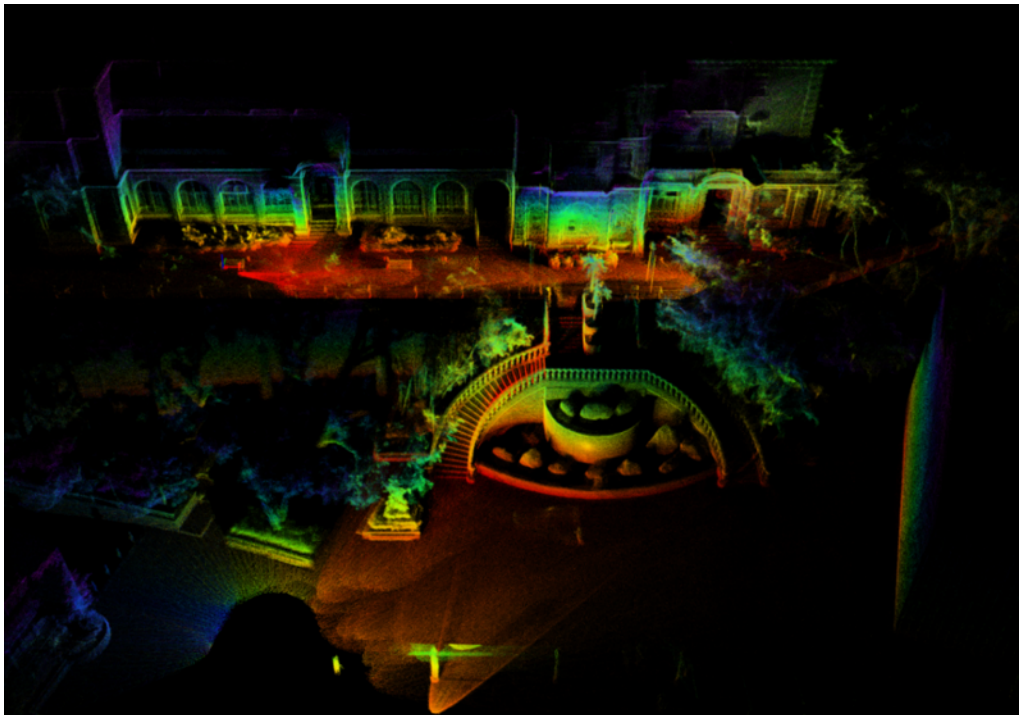
2.Stationary Environment 3D Scanning - Efficient and Portable Mid-40 solution on a Tripod

In this case, we use one Mid-40 and the motor mounted on top of a tripod to rotate around the z axis to scan different parts of the room, and integrate the point cloud data to give an overall 3D image. [Learn More](#)



3.Mobile 3D Mapping - LiDAR Odometry and Mapping Algorithms Package from HKU

To demonstrate the enormous potential of Livox LiDAR products in mobile 3D mapping, the Mechatronic and Robotic Systems (MaRS) laboratory (in the University of Hong Kong (HKU)) has developed a LiDAR odometry and mapping (LOAM) algorithm specifically for Livox LiDARs – LOAM-Livox. The LOAM algorithms are open source and are available for free for the benefit developers and customers [Learn More](#)



4.UAV 3D Mapping

To demonstrate the great potential of Livox LiDARs in 3D mapping, this showcase how they can be integrated into a drone to accomplish mapping tasks. As a test, the 3km-long Huizhou Bay Bridge, located in Huizhou, Guangdong, China was mapped. [Learn More](#)



NEW PRODUCT

5.1 HAP

5.1.1 Communication Protocol

5.2 Mid-360

Note: 1. When connecting the power cable and the function cable, pay attention to their respective voltage ranges and polarities. DO NOT connect any PoE device to the RJ-45 network connector. Incorrect connection can lead to irreversible damage to the LiDAR.

2. Overlapping with other Lidar FOVs should be avoided. If the laser beams are pointed right at each other, irreversible damage may be caused to the Mid-360.

5.2.1 Communication Protocol

Livox LiDAR Communication Protocol—Mid360

Release History		
Date	Version	Description
20220519	v1.0	1. Initialized version
20220530	v1.1	1. Merge status and parameter <code>key_value</code> information fields2. The information push command supports pushing LiDAR parameter information and status information
20220602	v1.2	1. Add log related commands
20220622	v1.3	1. Port 56000 only supports discovery command by broadcasting. Discovery command by broadcasting <code>ack</code> adds the control command port field of the current LiDAR
20220628	v1.4	1. Update working status description
20220629	v1.4.1	1. Add <code>firmware_type</code> to <code>key_value</code> list
20220815	v1.4.2	1. Add <code>detect_mode</code> & <code>func_io_cfg</code> to <code>key_value</code> list
20220825	v1.4.3	1. Add <code>tag_type</code> to the point cloud frame header to describe the type of point cloud tag2. Added description of tag type
20220915	v1.4.4	1. Add point cloud format description in spherical coordinate system2. Modified description of ROI configuration
20221028	v1.4.5	1. Modified description of FOV configuration2. Modified description of GPS time synchronization
20221114	v1.4.6	1. Add description of communication process
20221213	v1.4.7	1. Modified description of <code>pattern_mode</code> configuration
20230904	v1.4.8	1. Add description of log related protocols
20240102	v1.4.9	1. Add upgrade error related abnormal code description2. Add environment temp related high abnormal code
20240117	v1.4.1	1. Update description of GPS sync high-level time of the second pulse
20240222	v1.4.1	1. Add description related to time synchronization slope2. Add descriptions related to shock absorption and heat dissipation

Overview

This communication protocol is only used for livox Mid360 LiDAR.

Protocol Types

There are two types of data protocols between the user and the LiDAR, see as below:

Point cloud data protocol:

- Point cloud data;
- IMU data.

For details, see *2.3 Data Types*.

Control command protocol:

- Configure and query LiDAR parameters;
- LiDAR reset;
- Push and query LiDAR status;

- Upgrade LiDAR;

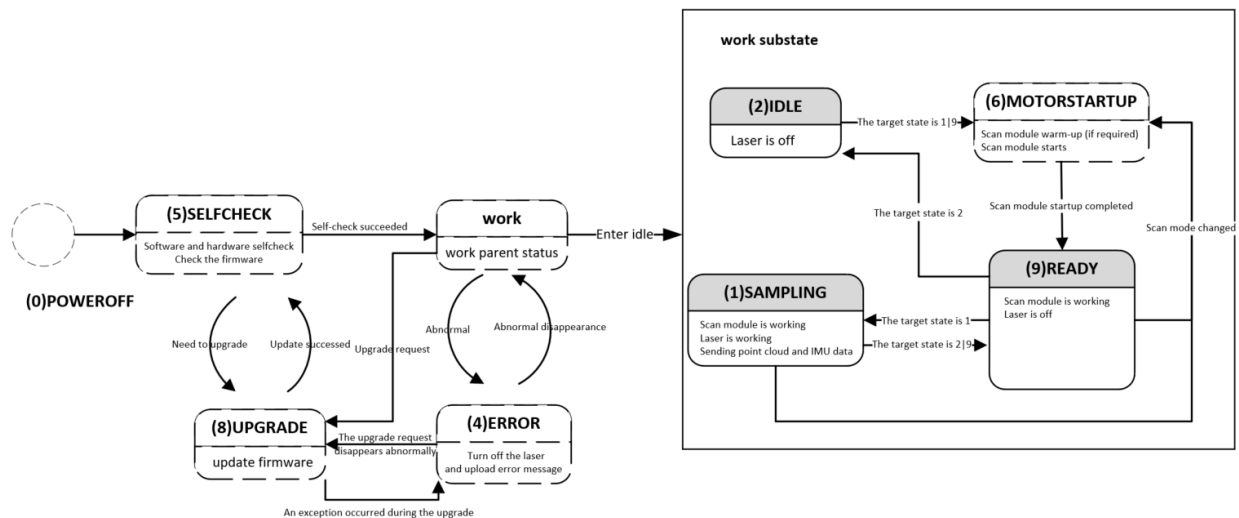
For details, see *4 CMD Details*

Both protocols are encapsulated using UDP data segments, and the protocol data is **Little-endian**.

LiDAR Working Status

The state machine of the current LiDAR is shown in the figure, The state enumeration values currently used by the Mid360 are as follows:

LiDAR Status	Enumeration Value	Whether to Support User Settings
SAMPLING	0x01	Support(By Parameter Configuration)
IDLE	0x02	Support(By Parameter Configuration)
ERROR	0x04	Not Support
SELFCECK	0x05	Not Support
MOTORSTARUP	0x06	Not Support
UPGRADE	0x08	Support(By Entering Upgrade)
READY	0x09	Support(By Parameter Configuration)



Notes:

1. The target working mode that the user can set has (1.IDLE, 2.SAMPLING, 3.READY)
2. Other states are intermediate states, which cannot be set by the user

Port Description

The source port and destination port are explained as follows according to the different data types:

Data Type	Transmission Direction	LiDAR Port	Host Port	Computer	Transmission Type	Transmission Protocol
Device Query	LiDAR <—> PC	56000	Any		Broadcast	UDP

56000 is used as the fixed listening port of Livox LiDAR, mainly used for the host computer to query the device through broadcast:

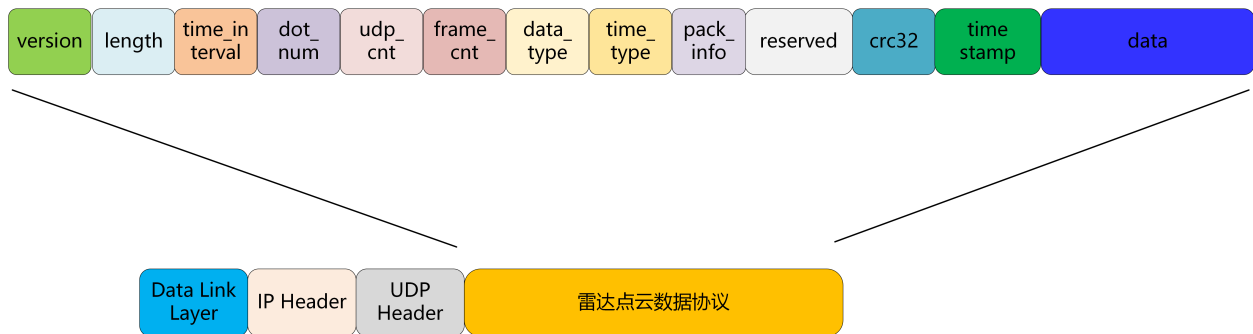
1. Only the device type query command (cmd_id: 0x0000) is supported. The host computer obtains the specific device type of the LiDAR through this port. This command response is replied by broadcast, so that when the host computer and the LiDAR IP are not in the same network segment, the host computer can still identify the LiDAR.

Mid-360 Communication Port:

Data Type	Transmission Direction	Direction	LiDAR Port	Host Port	Computer	Transmission Type	Transmission Protocol
Control Command	LiDAR <—> Computer	Host	56100	Any (56101 recommended)		Unicast	UDP
Push Command	LiDAR —> Computer	Host	56200	Configurable (default 56201)		Default Unicast	UDP
Point Cloud Data	LiDAR —> Computer	Host	56300	Configurable (default 56301)		Default Unicast(Support Multicast)	UDP
IMU Data	LiDAR —> Computer	Host	56400	Configurable (default 56401)		Default Unicast(Support Multicast)	UDP
LOG Data	LiDAR <—> Computer	Host	56500	Any (56501 recommended)		Unicast	UDP

Point Cloud & IMU Data Protocol

Protocol Format



Point cloud data format:

Fields	Offset (bytes)	Size (bytes)	Description
version	0	1	Package protocol version: currently 0.
length	1	2	The length of the entire UDP data segment starting from version .
time_inte	3	2	Intra-frame point cloud sampling time (Unit: 0.1us);In this frame of point cloud data, the time of the last point minus the time of the first point.
dot_num	5	2	The current UDP packet data field contains the number of points.
udp_cnt	7	2	Point cloud UDP packet count, each UDP packet is incremented by 1 in turn, and cleared to 0 at the beginning of the point cloud frame.
frame_cr	9	1	Point cloud frame count, plus 1 for each frame of point cloud (10Hz/15Hz, etc.);For non-repeating scans, this field is invalid.
data_type	10	1	Data type. For details, see 2.3 <i>Data Types</i> .
time_type	11	1	Timestamp type. For details, see 2.2 <i>Timestamp</i> .
re-served	12	12	Reserved.
crc32	24	4	Timestamp + data segment check code, using CRC-32 algorithm. For details, see 6 <i>CRC Algorithm</i> .
times-tamp	28	8	Point cloud timestamp. For details, see 2.2 <i>Timestamp</i> .
data	36	–	Data information. For details, see 2.3 <i>Data Types</i> .

Timestamp

The LiDAR system supports GPS time synchronization and PTP network time synchronization.

GPS time synchronization can be input through UDP packets, or through serial port GPRMC packets, see the time synchronization documentation for more details;

PTP time synchronization supports IEEE1588v2.0 UDP/IP synchronization, and gPTP L2 synchronization;

The timestamp in each packet represents the time of the first point cloud, and there are N point clouds in each packet. The time of these N point clouds is equally spaced, and the total time interval is the value of **time_interval**.

Timestamp type:

Timestamp Type	Sync Source Type	Data Format	Description
0	No synchronization source, the timestamp is the time when the LiDAR is turned on	uint64_t	Unit: ns
1	gPTP/PTP synchronization, the time of master clock source as a timestamp	uint64_t	Unit: ns
2	GPS time synchronization	uint64_t	Unit: ns

Notice:

- The time limit for GPS time synchronization is from 1/1/2000 to 31/12/2037
- PTP time synchronization does not support IEEE1588v2.1
- Not recommended for use in scenarios where IEEE1588v2.0 and gPTP coexist;

Data Types

There are N samples in each packet. The size of N depends on the data type.

There are four data types from 0 to 3, the default data type is 1:

Data Type	Sampling Type	Echo Mode	Coordinate Mode	N
0	IMU data			
1	Point cloud data 1	Single echo mode	Cartesian coordinates(32bit)	96
2	Point cloud data 2	Single echo mode	Cartesian coordinates(16bit)	96
3	Point cloud data 3	Single echo mode	Spherical coordinates	96

Data Type 0

IMU Data:

Field	Offset (bytes)	Data Type	Description
gyro_x	0	float	Unit: rad/s
gyro_y	4	float	Unit: rad/s
gyro_z	8	float	Unit: rad/s
acc_x	12	float	Unit: g
acc_y	16	float	Unit: g
acc_z	20	float	Unit: g

Data Type 1

Single echo Cartesian coordinate data format:

Field	Offset (bytes)	Data Type	Description
x	0	int32_t	X axis, Unit: mm
y	4	int32_t	Y axis, Unit: mm
z	8	int32_t	Z axis, Unit: mm
Re-flec-tivity	12	uint8_t	Reflectivity
Tag	13	uint8_t	According to the point cloud frame header <code>pack_info.tag_type</code> field, match the specific tag type. For details, see 2.4 Tag Information .

Data Type 2

Single echo Cartesian coordinate data format (16bit):

Field	Offset (bytes)	Data Type	Description (10mm is the resolution in 16bit format)
x	0	int16_t	X axis, Unit: 10mm
y	2	int16_t	Y axis, Unit: 10mm
z	4	int16_t	Z axis, Unit: 10mm
Re-flec-tivity	6	uint8_t	Reflectivity
Tag	7	uint8_t	According to the point cloud frame header <code>pack_info.tag_type</code> field, match the specific tag type. For details, see 2.4 <i>Tag Information</i> .

Data Type 3

Single return spherical coordinate data format:

Field	Offset (bytes)	Data Type	Description (10mm is the resolution in 16bit format)
depth	0	uint32_t	Depth, Unit: mm
theta	4	uint16_t	Zenith angle[0, 18000], Unit: 0.01 degree
phi	6	uint16_t	Azimuth[0, 36000], Unit: 0.01 degree
Re-flec-tivity	8	uint8_t	Reflectivity
Tag	9	uint8_t	According to the point cloud frame header <code>pack_info.tag_type</code> field, match the specific tag type. For details, see 2.4 <i>Tag Information</i> .

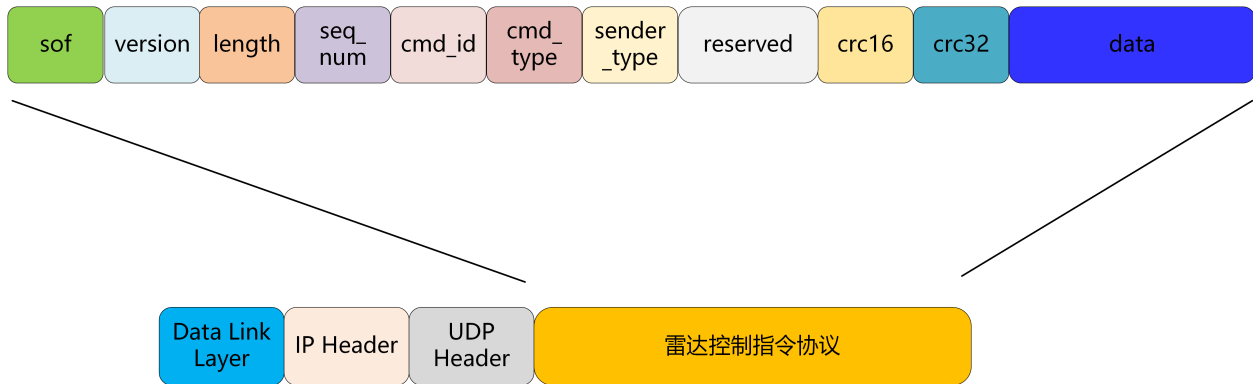
Tag Information

Tag mainly indicates other additional information of the detection point. The point cloud is marked as an 8bit unsigned integer, which is divided into several areas according to the bit, and each area represents an attribute of the detection point, including rain, fog, dust, adhesion point clouds between similar objects, etc.; Among them, the confidence level indicates the reliability of the detection point, and the normal point is generally 0 (High confidence); Low confidence indicates that the detection point is greatly affected by the corresponding attributes, and the reliability of the detection result is poor; if necessary, the point cloud can be filtered based on this information. Details are as follows:

Bit	Description
bit 6~7	Reserved
bit 4~5	Properties of Detection Points: other0: High confidence (normal point)1: Medium confidence2: Low confidence3: Reserved
bit 2~3	Properties of Detection Points: Rain, fog, dust and other tiny particles0: High confidence (normal point)1: Medium confidence2: Low confidence3: Reserved
bit 0~1	Properties of Detection Points: Glue point cloud between adjacent objects0: High confidence (normal point)1: Medium confidence2: Low confidence3: Reserved

Control Command

Frame Format



Format of control command is as follows:

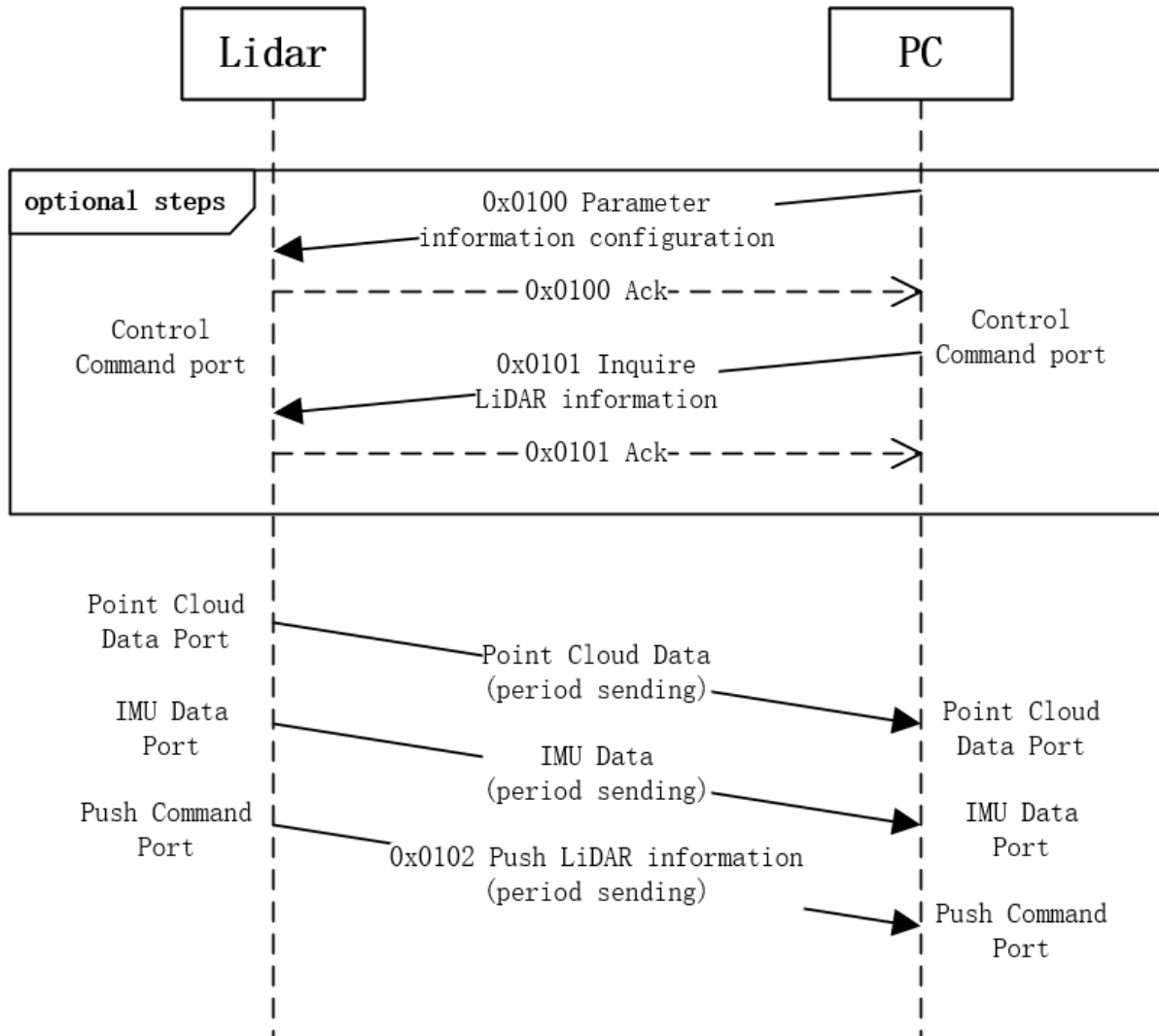
Field	Off-set(byte)	Size (byte)	Description
sof	0	1	Starting byte, fixed to be 0xAA.
version	1	1	Protocol version, 0 for current version.
length	2	2	Length of frame; The number of bytes from beginning of sof to end of entire data segment. Max value: 1400.
seq_num	4	4	This field is incremented by 1 for each new REQ request message; This field of ACK message is the same as REQ and can be used for message matching.
cmd_id	8	2	Different types of messages are distinguished by this field, For details, see <i>5 Return Code Description</i> .
cmd_type	10	1	Command Type: 0x00: REQ, actively send data request; 0x01: ACK, response to REQ data.
sender_id	11	1	The sender's device type: 0x00: Host computer, 0x01: LiDAR.
resv	12	6	Reserved.
crc16	18	2	Frame header checksum, check data starts from sof to crc16 (not included), 18 bytes in total. For details, see <i>6 CRC Algorithm</i> .
crc32	20	4	Frame data checksum. For details, see <i>6 CRC Algorithm</i> ; When length of data field is 0, CRC32 needs to be padded with 0.
data	24	n	<i>4 CMD details.</i>

Command ID

Command id list:

Function Type	CMD ID	Function
Device Type Query	0x0000	Discovery by broadcasting
LiDAR Information	0x0100	Parameter information configuration
	0x0101	Inquire LiDAR information
	0x0102	Push LiDAR information
Control CMD	0x0200	Request reboot device
	0x0201	Restore factory settings
	0x0202	Set GPS timestamp
log CMD	0x0300	Log file push
	0x0301	Log collection configuration
	0x0302	Log system time synchronization
	0x0303	Debug raw data collection configuration
General Upgrade CMD	0x0400	Request to start upgrade
	0x0401	Firmware data transfer
	0x0402	Firmware transfer complete
	0x0403	Get firmware upgrade status

Communication Process



CMD Detials

Device Type Query

0x0000 Discovery By Broadcasting

Request

CMD	Name	Offset(byte)	Data Type	Description
data	NULL			

ACK

ACK	Name	Off-set(byte)	Data Type	Description
data	ret_code	0	uint8_t	Return code (For details, see <i>5 Return Code Description</i>)
	dev_type	1	uint8_t	LiDAR type
	se- rial_numbe	2	uint8_t[1	LiDAR SN
	lidar_ip	18	uint8_t[4	LiDAR IP address E.g: AA.BB.CC.DD lidar_ip[0] = AA lidar_ip[1] = BB lidar_ip[2] = CC lidar_ip[3] = DD
	cmd_port	22	uint16_t	LiDAR command port

LiDAR Information

0x0100 Parameter Configuration

Request

CME	Name	Off-set(byte)	Data Type	Description
data	key_num	0	uint16_t	Number of key in key_value_list
	rsvd	2	uint16_t	Reserved
	key_value_l	4	key_value_l	Key list For convenience of expansion, multiple pieces of information are placed in a variable-length key_value_list

Format of each parameter in key_value_list is as follows:

Data Field	Offset(byte)	Data Type	Description
key	0	uint16_t	Key number, see table below
length	2	uint16_t	Length of value to the key
value	4	–	Value content to the key

ACK

ACK	Name	Off-set(byte)	Data Type	Description
data	ret_code	0	uint8_t	Return code (For details, see <i>5 Return Code Description</i>)
	er- ror_key	1	uint16_t	If the ret_code is not LVX_RET_SUCCESS, the error_key shows which key is error

0x0101 Parameter Inquire

Request

CMD	Name	Offset(byte)	Data Type	Description
data	key_num	0	uint16_t	Number of key in key_list
	rsvd	2	uint16_t	
	key_list	4	–	Key list

The key type in the key_list:

Data	Offset(byte)	Data Type	Description
key	0	uint16_t	key number

ACK

ACK	Name	Offset(byte)	Data Type	Description
data	ret_code	0	uint8_t	Return Code(For details, see <i>5 Return Code Description</i>)
	key_num	1	uint16_t	Number of key in key_value_list
	key_value_list	3	–	Value of key

0x0102 LiDAR Information Push

LiDAR push Actively

CMD	Name	Offset(byte)	Data Type	Description
data	key_num	0	uint16_t	Number of key in key_value_list
	rsvd	2	uint16_t	
	key_value_list	4	–	Value of key

Format of the key_value_list :

Data Field	Offset(byte)	Data Type	Description
key	0	uint16_t	key number, see table below
length	2	uint16_t	Length of value to the key
value	4	–	Value content to the key

Specific meanings to key_value_list are as follows:

Number(key)	Name	Length	Type	Content
0x0000	pcl_data_type	1	uint8_t	Type of point cloud data, For details, see[Data Types](2.3 Data Types)
0x0001	pattern_mode	1	uint8_t	Pattern mode 0x00: Non-repetitive scan0x01: Repetitive scan(Not supported)
0x0004	lidar_ipcfg	12	uint8_t[12]	LiDAR IP address configurationLiDAR IP address: AA.BB.CC.DD
0x0005	state_info_host_ipcfg	8	uint8_t[8]	IP address configuration for pushing LiDAR status informationdata[0-3]: LiDAR status
0x0006	pointcloud_host_ipcfg	8	uint8_t[8]	IP address configuration for pushing LiDAR point cloud datadata[0-3]: LiDAR point cloud data
0x0007	imu_host_ipcfg	8	uint8_t[8]	IP address configuration for pushing LiDAR IMU datadata[0-3]: LiDAR IMU data
0x0012	install_attitude	24		Installation location of LiDAR on user device.Assigned by user, the user can set it to 0
0x0015	fov_cfg0	20		FOV configuration, if you do not need to use all the FOV, you can set it to 0
0x0016	fov_cfg1	20		Same as fov_cfg0, this is the configuration of the second working mode
0x0017	fov_cfg_en	1	uint8_t	Dividing the enable of different regions by bitBit 0: 1 means fov_cfg0 is enabled
0x0018	detect_mode	1	uint8_t	Detect mode0x00: Normal0x01: SensitiveNote: Sensitive detection mode is not supported
0x0019	func_io_cfg	4	uint8_t[4]	Function IO configuration. Bytes 0-3 represent functions of IN0, IN1, IN2, IN3
0x001A	work_tgt_mode	1	uint8_t	LiDAR target working mode, For details, see1.2 LiDAR Working Modes
0x001C	imu_data_en	1	uint8_t	0x00: Disable IMU data push 0x01: Enable IMU data push
0x8000	sn	16	uint8_t[16]	String type(Use '\0' padding for less than 16 bits)LiDAR SN(Use '\0' padding for less than 16 bits)
0x8001	product_info	64	char[64]	String type(Use '\0' padding for less than 32 bits)LiDAR type + LiDAR model
0x8002	version_app	4	uint8_t[4]	App firmware version number: aa.bb.cc.ddversion_info[0]: aaversi
0x8003	version_loader	4	uint8_t[4]	Loader firmware version number
0x8004	version_hardware	4	uint8_t[4]	Hardware version number
0x8005	mac	6	uint8_t[6]	LiDAR MAC address: aa:bb:cc:dd:ee:ffdata[0]: aadata[1]: bbdata[2]: ccdata[3]: dd
0x8006	cur_work_state	1	uint8_t	Current working state of LiDAR
0x8007	core_temp	4	int32_t	Core temperature(Unit: 0.01°C)
0x8008	powerup_cnt	4	uint32_t	Count of LiDAR powerup
0x8009	local_time_now	8	uint64_t	Local time of device
0x800A	last_sync_time	8	uint64_t	Master time of last synchronization: If PTP synchronized: this time is PTP time
0x800B	time_offset	8	int64_t	Time offset of current local time from synchronization source, unit: 0.01s
0x800C	time_sync_type	1	uint8_t	Type of time synchronization0: No time synchronization1: PTP time synchronization
0x800E	lidar_diag_status	2	uint16_t	LiDAR diag status code Level of module status abnormality: 0: no abnormality
0x8010	FW_TYPE	1	uint8_t	Firmware type: 0: loader1: application_image
0x8011	hms_code	1	uint32_t[8]	Diagnostic Trouble Code, Each non-0 value represents a piece of code

Control Command

0x0200 Reboot Device

Request

CMD	Name	Offset (byte)	Data Type	Description
data	timeout	0	uint16_t	Reboot device delay time: ms

ACK

ACK	Name	Offset (byte)	Data Type	Description
data	ret_code	0	uint8_t	Return code: (For details, see 5 Return Code Description)

0x0201 Restore Factory Settings

After this command is issued, the LiDAR will be restored to the factory settings. After the reply is successful, the LiDAR will reboot immediately.

Request

CMD	Name	Offset (byte)	Data Type	Description
data	SN	0	uint8_t[16]	Reserved

ACK

ACK	Name	Offset (byte)	Data Type	Description
data	ret_code	0	uint8_t	Return code: (For details, see <i>5 Return Code Description</i>)

0x0202 Set GPS Timestamp

Request

CMD	Name	Offset (byte)	Data Type	Description
data	type	0	uint8_t	Timestamp format 0-1: Reserved 2: GPS time synchronization
	time_stamp	1	uint64_t	If GPS synchronized, this field should be the time of the last rising edge of the PPS signal, the type is uint64_t, unit: ns

ACK

ACK	Name	Offset (byte)	Data Type	Description
data	ret_code	0	uint8_t	Return code: (For details, see <i>5 Return Code Description</i>)

Log Command

0x0300 Log file push

Request

CMC	Name	Offset (byte)	Data Type	Description
data	log_type	0	uint8_t	Log type: 0 : real time log 3 : flash log
	file_index	1	uint8_t	File index
	file_num	2	uint8_t	This field is valid only log_type is 3it means the number of flash log
	flag	3	uint8_t	bit 0: 1 means need ackbit 1: 1 means create filebit 2: 1 means finish file
	timestamp	4	uint32_t	unix time stampthis count starts at the Unix Epoch on January 1st, 1970 at UTC log file naming rules: log_<log_type_name>_<file_index>_<system_time>.dat log_fully_log_1_20220613203140.dat
	rsvd	8	uint16_t	
	trans_index	10	uint32_t	send index
	log_data	14	uint16_t	log content length
	log_data	16	–	log content

ACK

ACK	Name	Offset (byte)	Data Type	Description
data	ret_code	0	uint8_t	Return code: (For details, see <i>5 Return Code Description</i>)
	log_type	1	uint8_t	
	file_index	2	uint8_t	
	trans_index	3	uint32_t	

0x0301 Log collection configuration

Request

CMD	Name	Offset (byte)	Data Type	Description
data	log_type	0	uint8_t	Log type: 0 : real time log 3 : flash log
	enable	1	uint8_t	0 : disable1 : enable

ACK

ACK	Name	Offset (byte)	Data Type	Description
data	ret_code	0	uint8_t	Return code: (For details, see <i>5 Return Code Description</i>)

0x0302 Log system time synchronization

Request

CMD	Name	Offset (byte)	Data Type	Description
data	times-tamp	0	uint32_t	unix time stampthis count starts at the Unix Epoch on January 1st, 1970 at UTC

ACK

ACK	Name	Offset (byte)	Data Type	Description
data	ret_code	0	uint8_t	Return code: (For details, see <i>5 Return Code Description</i>)

0x0303 Debug raw data collection configuration

Request

CMD	Name	Offset (byte)	Data Type	Description
data	enable	0	uint8_t	0:disable1 : enable
	host_ip	1	uint8_t[4]	Debug raw data destination IP addressAA.BB.CC.DD data[0]: AA data[1]: BB data[2]: CC data[3]: DD
	host_port	5	uint16_t	Debug raw data destination port
	reserved	7	uint16_t	reserved

ACK

ACK	Name	Offset (byte)	Data Type	Description
data	ret_code	0	uint8_t	Return code: (For details, see <i>5 Return Code Description</i>)

General Upgrade**0x0400 Request to Start Upgrade**

Request

CMD	Name	Offset (byte)	Data Type	Description
data	firmware_type	0	uint8_t	Firmware type: 0: Package firmware1: LiDAR firmware2: Other firmware
	encrypt_type	1	uint8_t	Encrypt type: 0: None encrypt1: AES1282: AES2563: DES4: 3DES
	firmware_length	2	uint32_t	Firmware length
	dev_type	6	uint8_t	Device type9: 9Mid360

ACK

ACK	Name	Offset (byte)	Data Type	Description
data	ret_code	0	uint8_t	Return code: (For details, see <i>5 Return Code Description</i>)

0x0401 Firmware Data Transmit

Request

CMD	Name	Offset (byte)	Data Type	Description
data	firmware_offset	0	uint32_t	
	current_length	4	uint32_t	
	encrypt_type	8	uint8_t	
	rsvd	9	uint8_t[3]	
	data	12	uint8_t[n]	firmware data

ACK

ACK	Name	Offset (byte)	Data Type	Description
data	ret_code	0	uint8_t	Return code: (For details, see <i>5 Return Code Description</i>)
	current_offset	1	uint32_t	
	re-ceived_length	5	uint32_t	

0x0402 Firmware Transmit Complete

Request

CMD	Name	Offset (byte)	Data Type	Description
data	checksum_type	0	uint8_t	
	checksum_length	1	uint8_t	
	checksum_data	2	uint8_t[n]	

ACK

ACK	Name	Offset (byte)	Data Type	Description
data	ret_code	0	uint8_t	Return code: (For details, see <i>5 Return Code Description</i>)

0x0403 Get Firmware Upgrade State

Request

CMD	Name	Offset (byte)	Data Type	Description
data	NULL			

ACK

ACK	Name	Offset (byte)	Data Type	Description
data	ret_code	0	uint8_t	Return code: (For details, see <i>5 Return Code Description</i>)
	up-grade_progress	1	uint8_t	Upgrade progress: 0 - 100

Return Code Description

Name	Return Code	Description
LVX_RET_SUCCESS	0x00	Execution succeed
LVX_RET_FAILURE	0x01	Execution failed
LVX_RET_NOT_PERMIT_NOW	0x02	Current state does not support
LVX_RET_OUT_OF_RANGE	0x03	Setting value out of range
LVX_RET_PARAM_NOTSUPPORT	0x20	The parameter is not supported
LVX_RET_PARAM_REBOOT_EFF	0x21	Parameters need to reboot to take effect
LVX_RET_PARAM_RD_ONLY	0x22	The parameter is read-only and cannot be written
LVX_RET_PARAM_INVALID_LEN	0x23	The request parameter length is wrong, or the ack packet exceeds the maximum length
LVX_RET_PARAM_KEY_NUM_EI	0x24	Parameter <code>key_ num</code> and <code>key_ list</code> mismatch
LVX_RET_UPGRADE_PUB_KEY_	0x30	Public key signature verification error
LVX_RET_UPGRADE_DIGEST_EI	0x31	Digest check error
LVX_RET_UPGRADE_FW_TYPE_	0x32	Firmware type mismatch
LVX_RET_UPGRADE_FW_OUT_C	0x33	Firmware length out of range
LVX_RET_UPGRADE_FW_ERASI	0x34	Firmware erasing

CRC Algorithm

CRC Name	Polynomial Formula	Width	Poly-nomial	Initial Value	Result XOR value	Input Reverse	Output Reverse
CRC-16/CCITT-FALSE	$x^{16} + x^{12} + x^5 + 1$	16	0x1021	0xFFFF	0x0000	false	false
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	32	0x04C11DB7	0xFFFF	0xFFFFFFFF	true	true

5.2.2 Supported Time Synchronization Methods

Livox Mid-360 supports 3 time synchronization methods

PTPIEEE 1588v2.0 UDP/IP Network protocol synchronization, see details: [PTP Time Synchronization](#)

gPTP: Automotive ethernet time synchronization protocol(2 layer), see details: [gPTP Time Synchronization](#);

GPS: Second pulse + GPRMC time data to form GPS time synchronization mode, see details: [GPS Time Synchronization](#)

Notice:

- The time limit for GPS time synchronization is from 1/1/2000 to 31/12/2037

- PTP time synchronization does not support IEEE1588v2.1
- Not recommended for use in scenarios where IEEE1588v2.0 and gPTP coexist

5.2.3 HMS Diagnostic Code Introduction

The HMS(health management system) codes in the communication protocols are the diagnostic codes of Mid360, which push periodically by the lidar. They are used to notify some special situations or help users check equipment failures.

Composition format

A complete HMS code is a 4 bytes(uint32) code. It consists of an abnormal ID and abnormal level, which is shown below:

Byte[3:2]	Byte[1]	Byte[0]
Abnormal ID	RSVD	Abnormal level

In addition, an abnormal level is a value that is not equal to 0, the value is bigger, and the problem is more serious. As is shown below:

value	level	description
0x01	Info	Notification during normal work
0x02	Warning	There might be problems, which will influence the normal work
0x03	Error	Device work in abnormal situation, please check the problem in time
0x04	fatal	Serious abnormal happened, it might not be recovered, lidar will stop working now

Parsing example

For example, an HMS code: 0x01020002, refer to [Composition format](#), the abnormal ID is 0x0102, and the abnormal level is 0x02(warning)

Refer to below HMS code table, the code represents “environment temperature is slightly high”, the abnormal level is warning, suggested action is “Please check the environment temperature and losing heat measures”

Diagnostic codes table

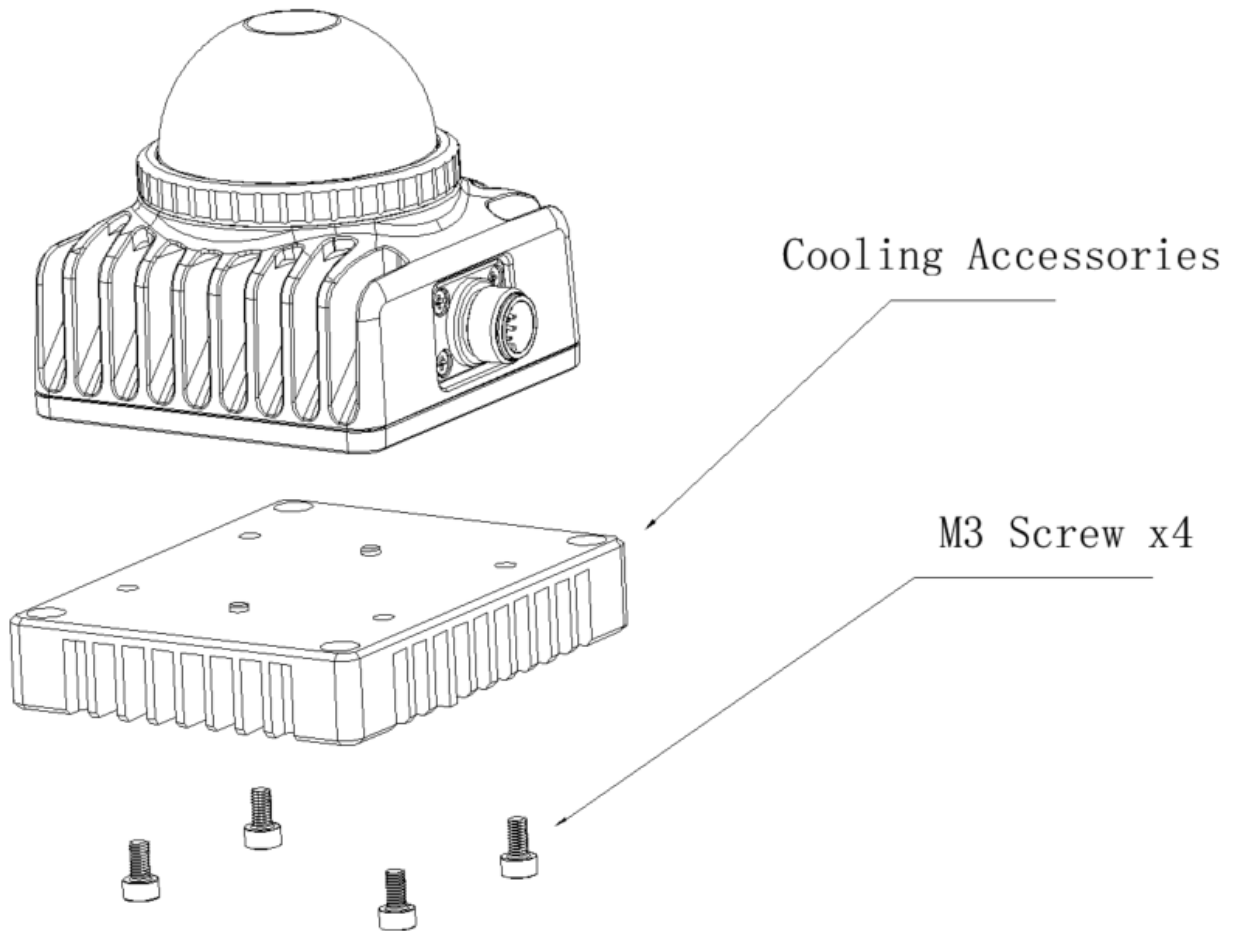
abnor- mal ID	abnor- mal level	abnormal description	suggested solution
0x0102	warning	Environment temperature is slightly high	Please check the environment temperature and losing heat measures
0x0103	warning	Environment temperature is relatively high	Please check the environment temperature and losing heat measures
0x0104	warning	The window is dirty, which will influence the reliability of the point cloud	Please clean the window
0x0105	warning	An error occurred during device upgrade process	Please restart the upgrade process
0x0111	fatal	Abnormal temperature of internal components of the device	Please check the environment temperature and losing heat measures
0x0112	fatal	Abnormal temperature of internal components of the device	Please check the environment temperature and losing heat measures
0x0113	fatal	IMU stopped working	Please try to restart the device to restore
0x0114	error	Environment temperature is high	Please check the environment temperature and losing heat measures
0x0115	fatal	Environment temperature beyond the limit, the device has stopped working	Please check the environment temperature and losing heat measures
0x0116	fatal	Abnormal external voltage	Please check the external voltage
0x0117	fatal	Abnormal lidar parameters	Please try to restart the device to restore
0x0201	warning	Scan module is heating	Please wait for the scan module heating
0x0210- 0x0219	error	Scan module is abnormal, the system is trying to recover	Please wait, if it lasts too long, please try restarting the device to restore
0x0210- 0x0219	fatal	Scan module is abnormal	Please try to restart the device to restore
0x0401	warning	Communication link was linked down, now it is recovered	Please check the communication link
0x0402	warning	PTP time synchronization stop or time gap is too big	Please check the PTP time source
0x0403	warning	The version of PTP is 1588-v2.1, device don't support this version	Please replace 1588-v2.1 version with 1588.2.0 version
0x0404	warning	PPS time synchronization abnormal	Please check the PPS and GPS signal
0x0405	warning	There was an exception in time synchronization	Please check the exception reason
0x0406	warning	Time synchronization accuracy is low	Please check the time source
0x0407	warning	PPS time synchronization fails because of loss of GPS signal	Please check the GPS signal
0x0408	warning	PPS time synchronization fails because of loss of PPS signal	Please check the PPS signal
0x0409	warning	GPS signal is abnormal	Please check the GPS time source
0x040A	warning	The PTP and GPTP signals exist at the same time	Please check the network topology, use PTP or GPTP alone to synchronize

5.2.4 Introduction to shock absorption and cooling accessories

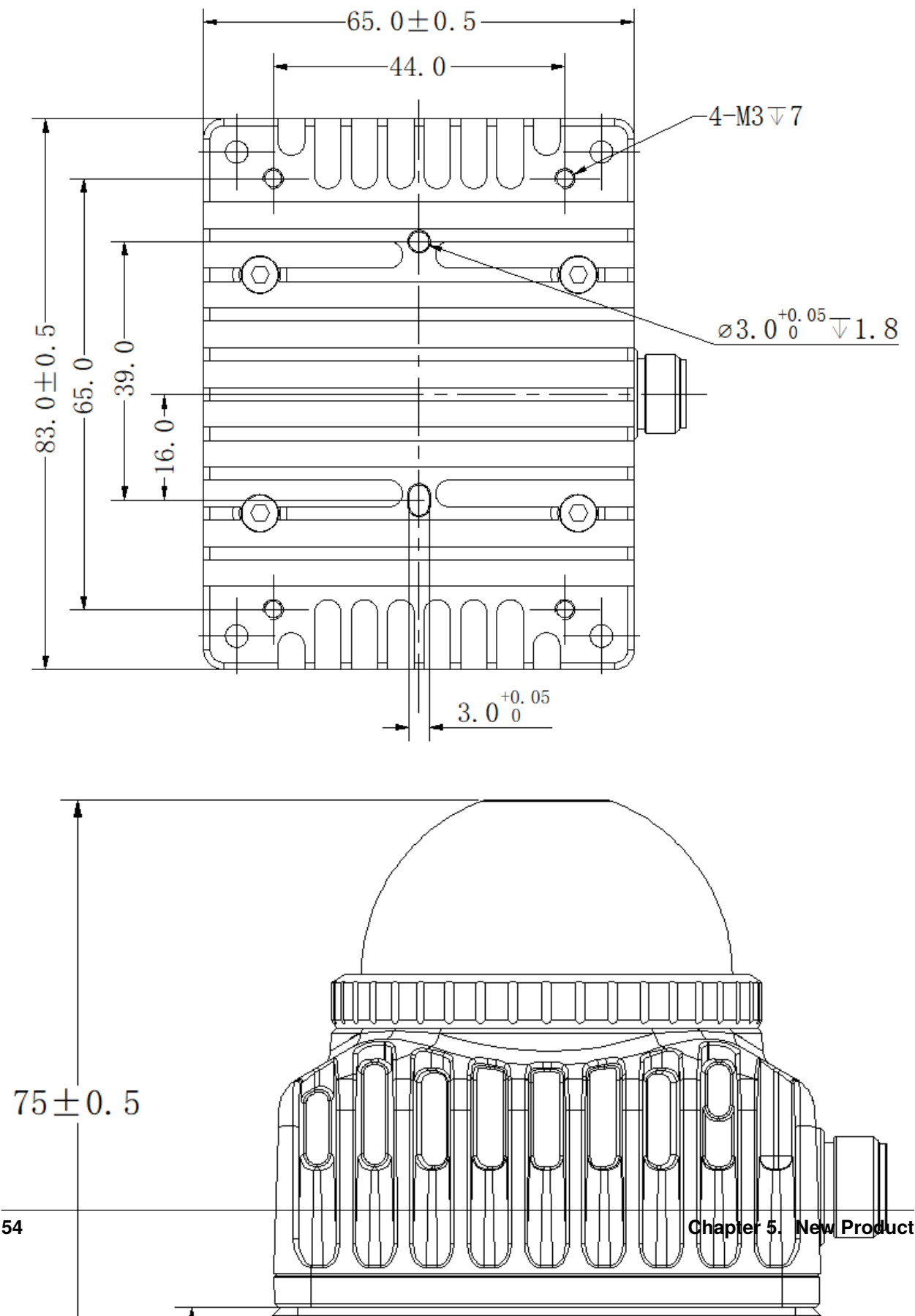
Cooling accessories

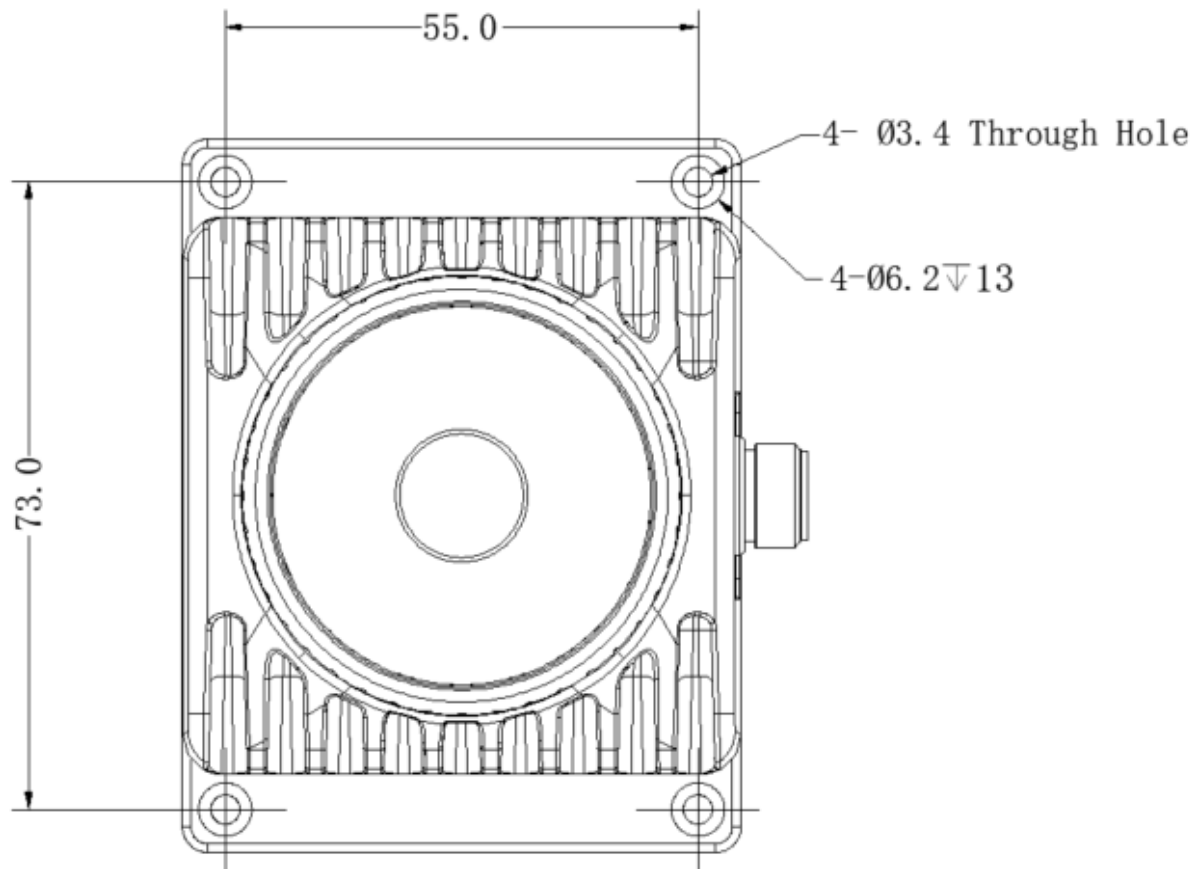
Users can purchase cooling accessories to meet the needs of working in a 60°C environment.

Cooling accessories installation method



Cooling accessories installation dimensions

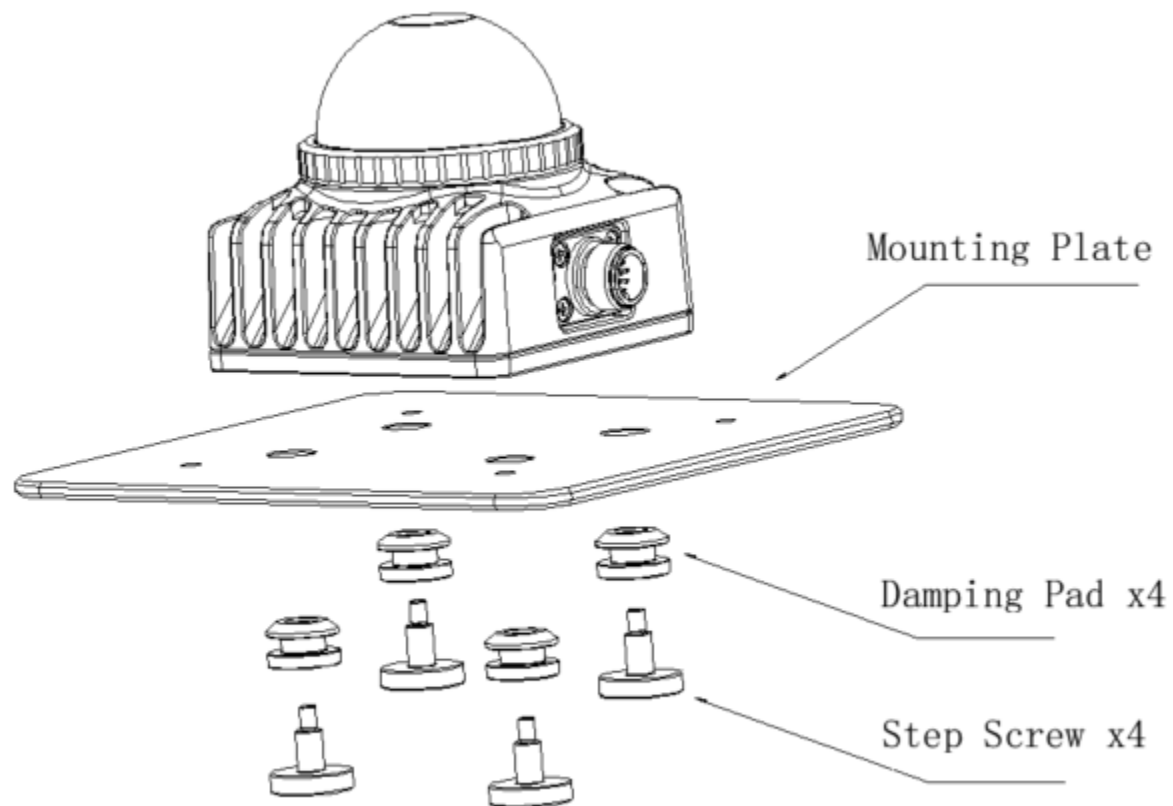




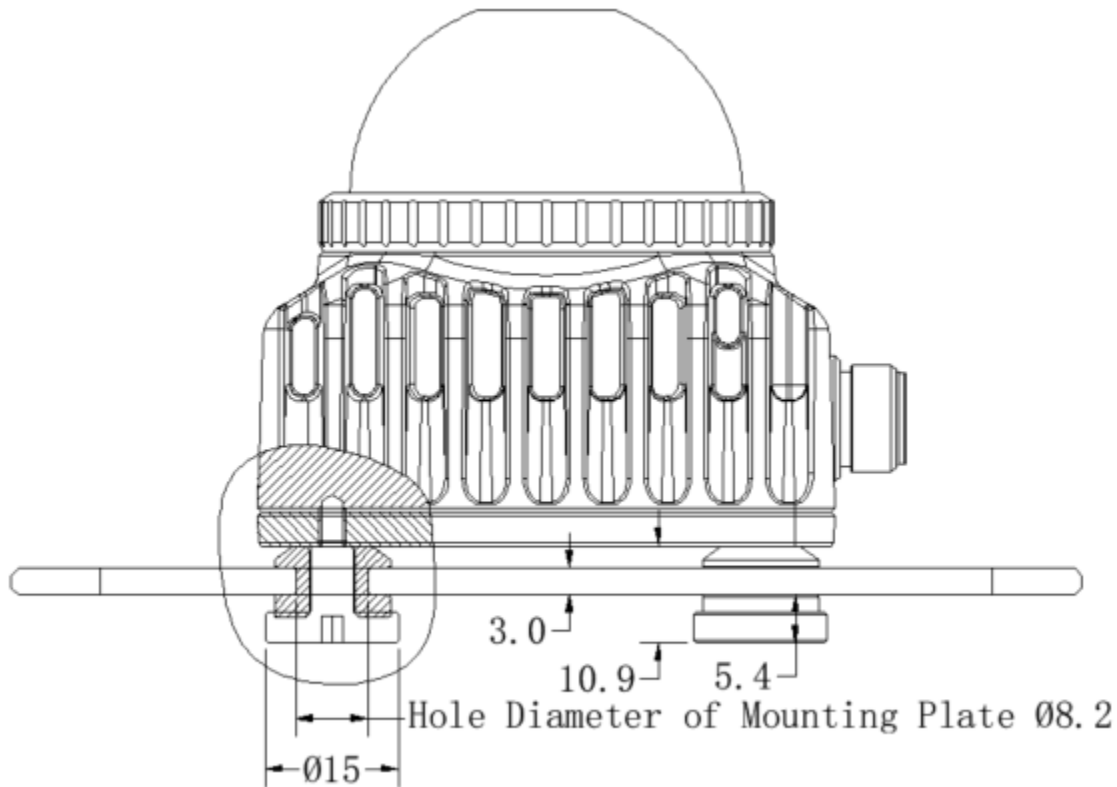
Vibration reduction accessories

If it is found that there is a large vibration and noise problem when installed on the user's base, which is more obvious than the vibration and noise of the radar itself, it may be that the resonance amplification effect of the radar and the installation structure has occurred. In addition to adjusting the installation structure mode to avoid resonance amplification, you can also make vibration reduction accessories to optimize vibration and noise problems. The vibration reduction structure and installation diagram are shown in the figure below.

Vibration reduction accessories installation method



Vibration reduction accessories installation dimensions



Related design drawings

- *Vibration reduction design drawing 1*
- *Vibration reduction design drawing 2*

This document only provides design reference. Specific component groups need to be produced by users according to actual scene requirements.

Precautions for vibration reduction accessories

1. The installation base is only an installation diagram and is not included in the accessories.
2. Vibration-absorbing pads and step screws can be used in conjunction with the above-mentioned heat dissipation accessories.
3. The upper end of the vibration damping pad is chamfered and can be directly deformed and pressed in from the bottom of the mounting plate. It is recommended that the diameter of the contact surface between the mounting base and the vibration damping pad be greater than 15mm.
4. The mode of the installation structure should avoid resonance with the vibration reduction structure, otherwise the vibration reduction effect will not be achieved. It is recommended that customers test and confirm the vibration reduction effect before officially introducing it for use.
5. After using the vibration-absorbing installation structure, the external parameter fluctuations of the radar will be larger than those directly connected. This is only recommended for customers who are not sensitive to external parameter fluctuations.

5.3 Public Information

5.3.1 Livox SDK2

5.3.2 Livox Ros Driver2

5.3.3 Time Synchronization Instructions

The synchronization methods supported by different products are different, and the synchronization methods supported by each product are detailed in the description of each product page.

Application Scenarios of Various Time Synchronization Methods

(Recommend)PTP/gPTP:

PTP v2 and gPTP can be used for time synchronization between Livox LiDAR and other devices without GPS and PPS hardware signals. This method requires the least external hardware support, and only needs to have a master clock device in the entire network.

GPS:

In the case that other sensors are directly synchronized through GPS signals, the synchronization can be completed by directly connecting the hardware to the PPS signal and GPS signal, without additional software deployment;

PTP Time Synchronization

Principle of PTP Time Synchronization

The synchronization process uses the Delay request-response mechanism (two steps) of IEEE 1588v2.0 PTP. The Livox device acts as a slave and performs ptp time synchronization with the master clock device.

The master and slave clocks interact with the Sync, Follow_Up, Delay_Req, and Delay_Resp data packets to obtain the time t_1 , t_2 , t_3 , and t_4 . The transmission delay and the offset of the two clocks are obtained through the following calculations:

Transmission delay:

$$Delay = [(t_4 - t_1) - (t_3 - t_2)] / 2$$

Time offset:

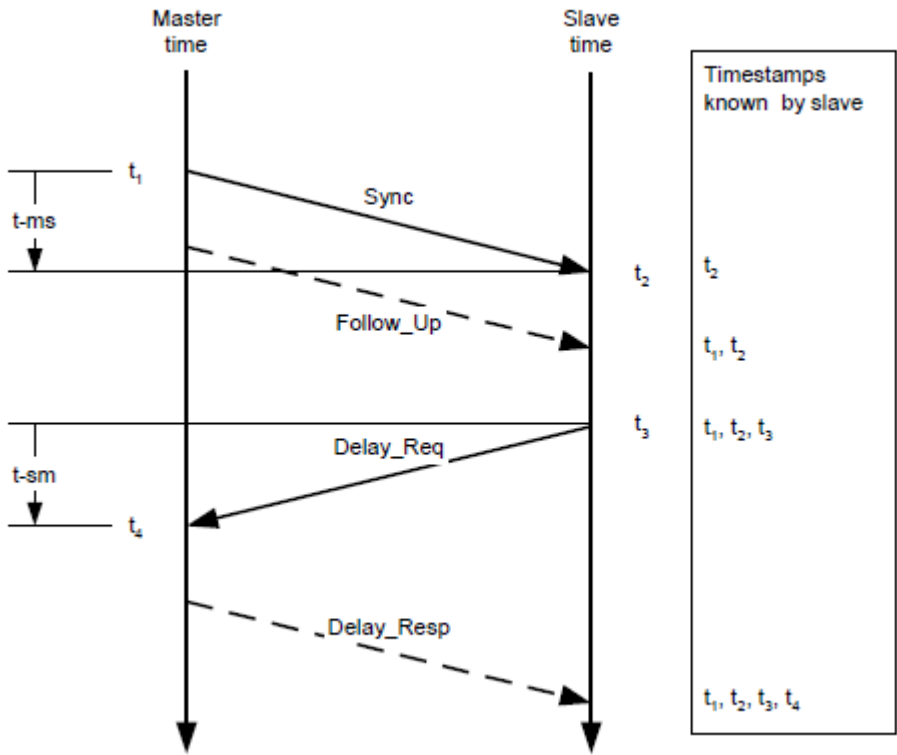
$$Offset = (t_2 - t_1) - Delay = [(t_2 - t_1) + (t_3 - t_4)] / 2$$

PTP Time Synchronization Method

When Livox LiDAR is connected to a network with a PTP master clock, the device will automatically synchronize its own time to the master clock.

****Step 1:** Check whether the master clock in the network is normal: **

For the method of configuring the main clock on the PC, please refer to the appendix [Appendix: How to configure the PTP master clock](#)



ptp						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.250	224.0.1.129	PTPv2	106	Announce Message
2	0.175128	192.168.1.250	224.0.1.129	PTPv2	86	Sync Message
3	0.175130	192.168.1.250	224.0.1.129	PTPv2	86	Follow_Up Message
4	0.999903	192.168.1.250	224.0.1.129	PTPv2	106	Announce Message
5	1.175057	192.168.1.250	224.0.1.129	PTPv2	86	Sync Message
6	1.175058	192.168.1.250	224.0.1.129	PTPv2	86	Follow_Up Message

If there are two data of Sync and Follow_Up in the network, it means that the PTP master clock source exists and works normally:

Step 2: Connect Livox device

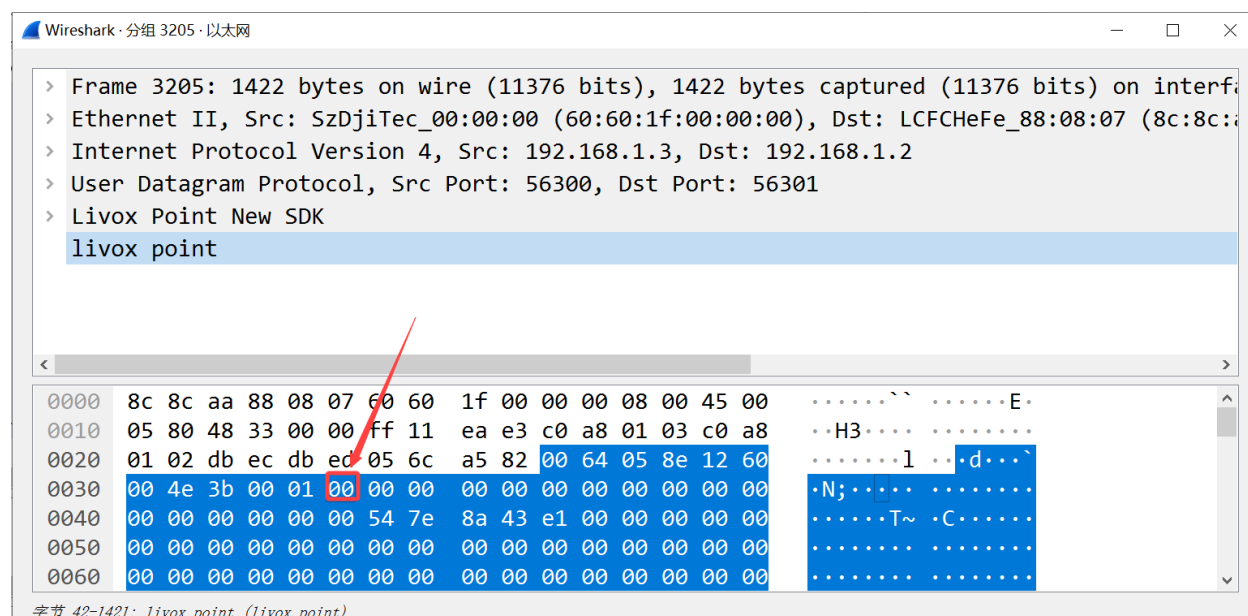
You can connect Livox LiDAR normally through a network cable, no additional wiring is required.

Step 3: How to view the device is using PTP synchronization

1. By viewing the timestamp_type data in the point cloud data packet header (see the product communication protocol for details)

If timestamp_type is 1, it means that the device is performing PTP time synchronization

timestamp_type	data_type	time type
1	uint64_t	ns



2. Or view the Sync Type in the “Settings” option in the Livox Viewer host computer

Note:

1. PTP synchronization has the highest priority. When PTP or GPS are available at the same time, PTP network synchronization will be used first;
2. The network environment with multiple master clocks is not supported;

gPTP Time Synchronization

Principle of gPTP Time Synchronization

In the “Specification of Time Synchronization over Ethernet” standard of AUTOSAR, corresponding special requirements are made for the application of IEEE 802.1AS in the vehicle field, forming the AUTOSAR Time Sync over Ethernet standard.

Such as:

- Cancel the Announce message in the network;
- BMCA algorithm is not supported;
- The master side does not send delay_req message;
- ...

and other requirements.

The Livox LiDAR time synchronization protocol and functions also comply with the requirements of the above standards, and the synchronization process follows the IEEE 802.1AS standard, where the domain controller acts as the master and the Livox device acts as the slave.

In the above way, the transmission delay is calculated: $\text{link_delay} = [(t_4 - t_1) - (t_3 - t_2)] / 2$

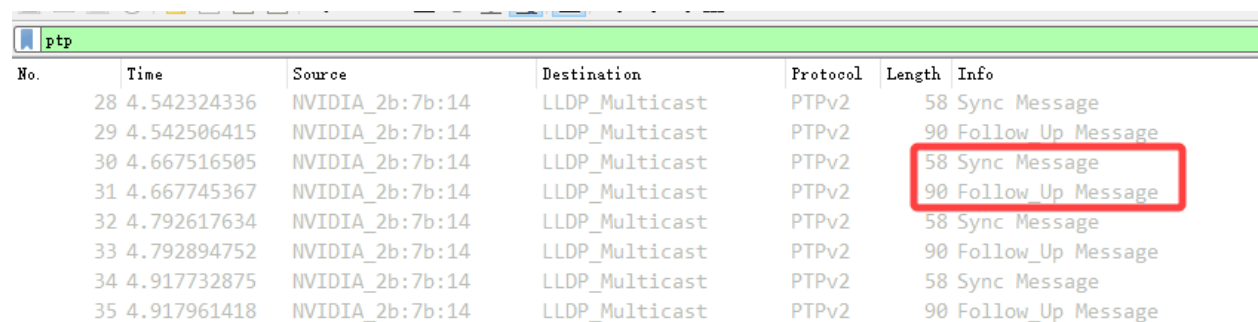
Calculate time offset by Sync, Followup: $\text{Offset} = (t_2' - t_1') - \text{link_delay}$

gPTP Time Synchronization Method

Step 1: Check whether the master clock in the network is normal:

For the method of configuring the main clock on the PC, please refer to the appendix *Appendix: How to configure the PTP master clock*;

If there are Sync and Follow_Up data in the network, it means that the gPTP master clock source exists and works normally:



No.	Time	Source	Destination	Protocol	Length	Info
28	4.542324336	NVIDIA_2b:7b:14	LLDP_Multicast	PTPv2	58	Sync Message
29	4.542506415	NVIDIA_2b:7b:14	LLDP_Multicast	PTPv2	90	Follow Up Message
30	4.667516505	NVIDIA_2b:7b:14	LLDP_Multicast	PTPv2	58	Sync Message
31	4.667745367	NVIDIA_2b:7b:14	LLDP_Multicast	PTPv2	90	Follow Up Message
32	4.792617634	NVIDIA_2b:7b:14	LLDP_Multicast	PTPv2	58	Sync Message
33	4.792894752	NVIDIA_2b:7b:14	LLDP_Multicast	PTPv2	90	Follow Up Message
34	4.917732875	NVIDIA_2b:7b:14	LLDP_Multicast	PTPv2	58	Sync Message
35	4.917961418	NVIDIA_2b:7b:14	LLDP_Multicast	PTPv2	90	Follow Up Message

Step 2: Connect Livox device

You can connect Livox LiDAR normally through a network cable, no additional wiring is required.

Step 3: How to check whether the device is using gPTP time synchronization

Same as PTP, there is no distinction between PTP synchronization or gPTP time synchronization in the device status.

Appendix: How to configure the PTP master clock

refer to: [The Linux PTP Project](#)

Download:

This method has been tested and is recommended to use [linuxptp v3.1.1](#)

You can also use git clone (not recommended)

```
git clone http://git.code.sf.net/p/linuxptp/code linuxptp
```

Compile and install:

```
cd linuxptp
make
sudo make install
```

If you want ptp4l to work normally, you need the system's Linux kernel version 3.0 or higher. Use the following command to check whether the network card supports software/hardware timestamp function (for example: query the status of network card 'eth0').

```
ethtool -T eth0
```

The network card capabilities option includes the following parameters, indicating that the network card supports software timestamps:

```
SOF_TIMESTAMPING_SOFTWARE
SOF_TIMESTAMPING_TX_SOFTWARE
SOF_TIMESTAMPING_RX_SOFTWARE
```

The network card capabilities option includes the following parameters, indicating that the network card supports hardware timestamps:

```
SOF_TIMESTAMPING_RAW_HARDWARE
SOF_TIMESTAMPING_TX_HARDWARE
SOF_TIMESTAMPING_RX_HARDWARE
```

The example in the figure below shows that the eth0 network card supports hardware timestamps.

In the example below, the wlan0 network card does not support hardware and software timestamps, and linuxptp/ptp4l cannot work normally.

Run the following command to start the ptp4l master clock function in PTP mode:

```
sudo ptp4l -i eth0 -l 6 -m
```

Run the following command to start the ptp4l master clock function using the gPTP mode software timestamp:

```
sudo ptp4l -i eth0 -S -ml 6 -f automotive-master.cfg
```

Run the following command to start the ptp4l master clock function using the gPTP mode hardware timestamp:

```
sudo ptp4l -i eth0 -H -ml 6 -f automotive-master.cfg
```

If Sync Message data appears in the network, it means that the master clock function of the ptp4l program is running normally:

```

Time stamping parameters for eth0:
Capabilities:
    hardware-transmit      (SOF_TIMESTAMPING_TX_HARDWARE)
    software-transmit      (SOF_TIMESTAMPING_TX_SOFTWARE)
    hardware-receive       (SOF_TIMESTAMPING_RX_HARDWARE)
    software-receive       (SOF_TIMESTAMPING_RX_SOFTWARE)
    software-system-clock  (SOF_TIMESTAMPING_SOFTWARE)
    hardware-raw-clock     (SOF_TIMESTAMPING_RAW_HARDWARE)
PTP Hardware Clock: 0
Hardware Transmit Timestamp Modes:
    off                    (HWTSTAMP_TX_OFF)
    on                     (HWTSTAMP_TX_ON)
Hardware Receive Filter Modes:
    none                   (HWTSTAMP_FILTER_NONE)
    ptpv1-l4-sync          (HWTSTAMP_FILTER_PTP_V1_L4_SYNC)
    ptpv1-l4-delay-req     (HWTSTAMP_FILTER_PTP_V1_L4_DELAY_REQ)
    ptpv2-l4-sync          (HWTSTAMP_FILTER_PTP_V2_L4_SYNC)
    ptpv2-l4-delay-req     (HWTSTAMP_FILTER_PTP_V2_L4_DELAY_REQ)
    ptpv2-l2-sync          (HWTSTAMP_FILTER_PTP_V2_L2_SYNC)
    ptpv2-l2-delay-req     (HWTSTAMP_FILTER_PTP_V2_L2_DELAY_REQ)
    ptpv2-event            (HWTSTAMP_FILTER_PTP_V2_EVENT)

```

```

Time stamping parameters for wlan0:
Capabilities:
    software-receive       (SOF_TIMESTAMPING_RX_SOFTWARE)
    software-system-clock  (SOF_TIMESTAMPING_SOFTWARE)
PTP Hardware Clock: none
Hardware Transmit Timestamp Modes: none
Hardware Receive Filter Modes: none

```

ptp						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.250	224.0.1.129	PTPv2	106	Announce Message
2	0.175128	192.168.1.250	224.0.1.129	PTPv2	86	Sync Message
3	0.175130	192.168.1.250	224.0.1.129	PTPv2	86	Follow_Up Message
4	0.999903	192.168.1.250	224.0.1.129	PTPv2	106	Announce Message
5	1.175057	192.168.1.250	224.0.1.129	PTPv2	86	Sync Message
6	1.175058	192.168.1.250	224.0.1.129	PTPv2	86	Follow_Up Message

You can use the following command to synchronize the system time with the PTP hardware clock:

For more information, please refer to [phc2sys link](#)

```
sudo phc2sys -c eth0 -s CLOCK_REALTIME -O 0
```

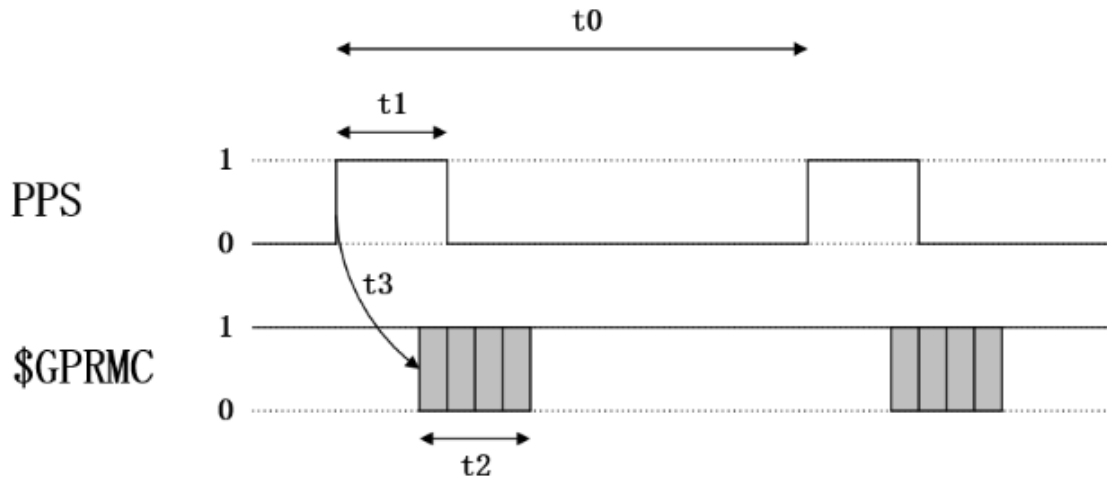
After completing the above steps, the LiDAR time and PC local system time can be synchronized through ptp4l.

GPS Time Synchronization

Principle of GPS Time Synchronization

The PPS port of the GPS clock source sends a hardware pulse (PPS signal) every second, and then the data port sends time information (GPRMC format) corresponding to the rising edge of this pulse.

After receiving the rising edge of the PPS signal and analyzing the correct time information from the GPRMC data, the Livox device will set the point cloud time as the GPS time, and keep this time base for continuous accumulation to achieve time synchronization with the GPS device.



Timing requirements for PPS signal and GPRMC signal:

time	description	effective range	recommended value
t_0	The interval between the rising edges of two adjacent second pulses	900 ms ~ 1100ms	~ 1000 ms
t_1	High level time of the second pulse	> 1us	10 ms ~ 200 ms
t_2	Transmission time of GPRMC (baud rate: 9600bps)	70 ms	70 ms
t_3	The delay of GPRMC data start sending relative to the rising edge of the pulse	0 ms ~ 900 ms	0 ms ~ 430 ms

Note: The slope of PPS pulse is recommended to be greater than 1V/us

GPS time synchronization method

In order to be compatible with other manufacturers' LiDAR hardware, Livox devices also support GPS time synchronization.

The following takes Mid-360 as an example to introduce two ways of using GPS synchronization:

1. (recommended) Synchronization via Uart: Lidar hardware is connected to PPS pulse signal, and Lidar hardware is connected to GPRMC Uart signal;
2. Synchronization via Ethernet: the hardware accesses the PPS pulse signal, and sends the GPRMC signal to Lidar through UDP packets;

Synchronization via Uart

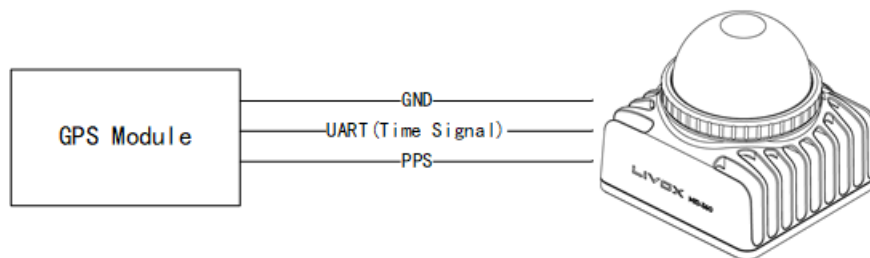
Hardware connection

If the UART output signal and PPS signal of the GPS module are at TTL (3.3v) level, directly connect the two signal lines to the corresponding external pins of the Mid360 (pin 8 "second pulse" and pin 10 "GPS input" pin), and the corresponding pin sequence of Mid360 can be found in "Livox Mid-360 User Manual";

Otherwise, the Uart signal and PPS signal need to be converted to TTL (3.3v) level before they can be connected to the corresponding pins of Mid-360.

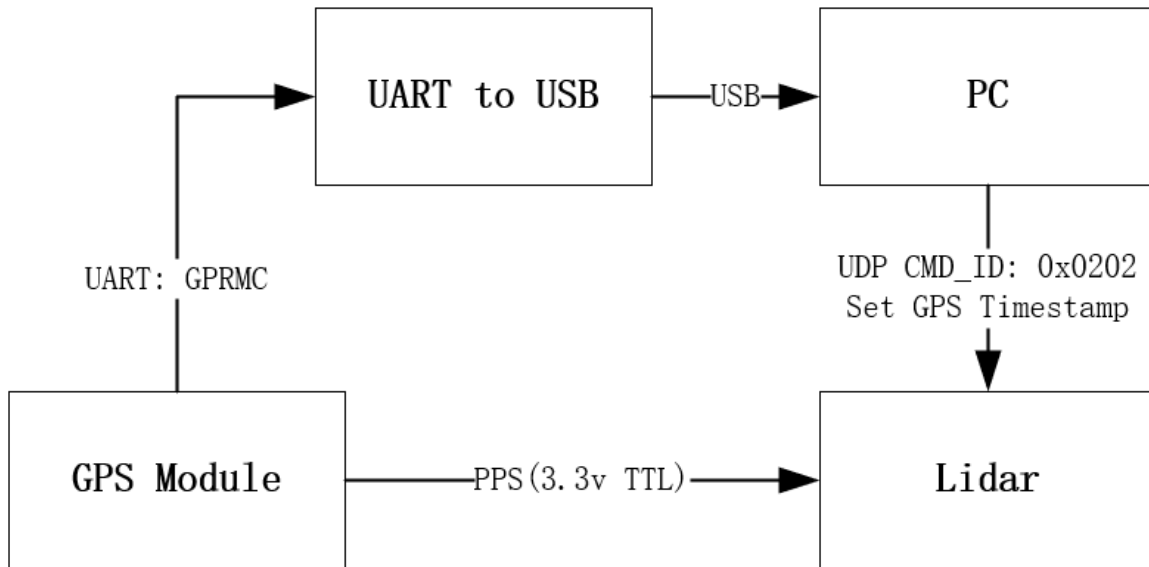
Note:

1. Adopt Uart mode, no need to configure SDK software;
2. Pay special attention to the hardware access voltage, otherwise it will cause equipment damage;
3. GPS Uart needs to be configured as: baud rate 9600, data bit 8bit, no parity;
4. The content output by GPS Uart needs to contain GPRMC or GNRMC information;
5. Pay attention to the signal quality of hardware wiring, which will seriously affect the stability and accuracy of synchronization.



Synchronization via Ethernet(not recommended)

If the GPRMC signal output by the GPS module is connected to the host computer (PC or industrial computer), after the host computer parses the time information, it can send UDP packets according to the control command protocol to set the GPS time stamp for the device;



How to view the device is using GPS synchronization

1. By viewing the timestamp_type data in the point cloud data packet header (see the product communication protocol for details)

If timestamp_type is 2, it means that the device is performing GPS time synchronization:

timestamp_type	data_type	time type
2	uint64_t	ns

2. Or in the Livox Viewer host computer, view the Sync Type of the “Settings” option;

Note: Normally, the GPS module can output PPS signal and time signal normally only after receiving GPS signal. When using, it is necessary to ensure that the GPS signal is stable; currently the time signal only supports the two formats of GPRMC/GNRMC;

OTHER PRODUCT

6.1 Single lidar data acquisition

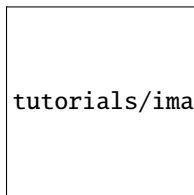
6.1.1 Use Livox Viewer to Collect Data

Take Livox Mid-40 as an example:

dependenciesUbuntu 16.04 or win10, Livox_Viewer(see driver for this part of the software installation Driver)

Hardware partLivox-Mid 40, power adapter socket 2.0 or connection box, Ethernet cable, PC.

ConnectionThe physical connection is shown in the figure below



tutorials/image/one_lidar_connect.png

If you have any questions about the connection and IP configuration, please refer to the “Connection” section of the Livox Mid user manual on the Livox official website, and the link of the [Livox Mid user manual](#)

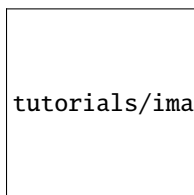
Acquisition: After completing the hardware connection, open Livox Viewer, the download address is as follows:

[Livox Viewer 0.8.0 for Windows](#)

[Livox Viewer 0.7.0 For Linux Ubuntu16.04_x64](#)


After starting the terminal, go to the root directory of the unzipped folder and run the command ‘./livox viewer.sh’ to start

If the Lidar is correctly connected and the IP configuration is completed, the Livox Viewer interface is now shown as follows, and the model and 15-digit broadcast code of the connected Livox Lidar are displayed on the left.



tutorials/image/start_Viewer_and_link_Horizon.png

Click the button to start Lidar, and then click the play button to display the point cloud image collected by Lidar, as shown below:



tutorials/image/capture.png

After the acquisition is successful, you can click the record button on the toolbar to record the lvx format file, pause playback or click this button again to end the recording.

The toolbar contains multiple buttons to set the point cloud color scheme, point style size, frame integration duration and other functions. For details, please refer to the official website Livox Viewer User Manual. The link to Livox [Livox Viewer User Manual](#)

The default format of Livox Viewer to record and save files is lvx. Select File->Save as in the menu to save the current point cloud as a csv or las file.


6.1.2 Collection Under ROS

Take Livox Mid-40 as an example:

dependenciesubuntu 16.04ROSLivox-SDKLivox_ros_driver(see the driver for the detailed process of downloading and installing this part Driver)

Hardware partLivox-Mid 40 × 1, power adapter 2.0 or connection box × 1, Ethernet cable (100M or Gigabit) × 1, PC × 1

ConnectionThe physical connection is shown in the figure below



tutorials/image/one_lidar_connect.png

If you have any questions about the connection and IP configuration, please refer to the “Connection” section of the Livox Mid User Manual on the Livox official website. [Livox Mid Manual](#)

Point Cloud Collection:

- Lvx:

First confirm that Livox-SDK has been cloned from Git


```
$ git clone https://github.com/Livox-SDK/Livox-SDK.git
```

Complete the Lidar hardware and IP configuration part, and open a terminal window under the following file path:


```
$ ../Livox_SDK/build/sample/lidar_lvx_file
```

Execute the following command to connect to Lidar and set the duration of the recorded point cloud data to 10s:

```
$ ./lidar_lvx_sample -c "Lidar's Broadcast code" -t 10
```




tutorials/image/save_lvx_data_by_SDK_01.png



tutorials/image/save_lvx_data_by_SDK_02.png

After successful execution, a lvx file can be generated under the current path



tutorials/image/save_lvx_data_by_SDK_03.png

- rosbag:

Note: Since the Horizon integrates IMU, there are two topics when using this method to record Horizon lidar data: point cloud data in CustomMsg format and IMU data in sensor_msgs/Imu format.

1.1. After connecting the lidar, open the terminal to compile and update the current ROS package environment in the workspace where livox_ros_driver is located:

```
$ catkin_make
$ source ./devel/setup.sh
```

2. Use the ROS launch file to load the Livox ROS driver:

```
$ roslaunch livox_ros_driver livox_lidar_rviz.launch
```

3. After starting rviz and displaying the lidar-screen, use the record command to record data:

```
$ rosbag record -a
```

6.2 Multi-lidars data collection (using Livox Hub)

6.2.1 Connection and Configuration

Take a Livox Mid-40 and a Livox Horizon as an example:

dependencies: ubuntu 16.04 ROS Livox-SDK Livox ROS driver see the driver for the detailed process of downloading and installing of this part

Hardware part: Livox Mid-40 × 1, Livox Horizon × 1, Livox Hub × 1, Gigabit Ethernet cable × 1, Livox Hub power cord × 1, Livox lidar battery × 1, PC × 1.

synchronization:For time stamp synchronization between multiple livox LiDARs, please refer to: [Timestamp Synchronization](#)

Wiring and IP configuration:Livox Hub uses UDP communication protocol for data communication through Ethernet, and supports two types of IP address configuration: static IP address and dynamic IP address. The factory default is to use Dynamic Host Configuration Protocol (DHCP) to assign IP addresses. According to different IP address settings, the physical connection is slightly different.

- When using a static IP address to connect, the physical connection is as shown in the figure below:



- When a dynamic IP address is used for connection, the physical connection is shown below:



Livox Hub uses Dynamic Host Configuration Protocol (DHCP) to assign IP addresses by default when it leaves the factory. If you want to use static IP to connect, you need to connect to a router for settings. The steps are as follows:

- 1.Connect the Livox Hub, Livox Lidar, PC and power supply through the router according to the dynamic IP address connection method as above figure.
- 2.Run Livox Viewer on the computer and click to open the device manager. Select Livox Hub and click to open the device properties page. Set the IP address of Livox Hub as a static IP address in the device properties page. Note that the static IP address of Livox Hub should be set to 192.168.1.X (where X is any number between 2 and 233).
- 3.After setting up, disconnect all connections to Livox Hub.
- 4.If Livox Hub is in static IP mode, the computer should also be set to static IP mode (if the computer uses dynamic IP mode, Livox Hub may not be found). The static IP address of the computer should be set to 192.168.1.X (where X is any number between 2 and 233), and it cannot be the same as the IP address of the connected Livox Hub.

If you have any questions about the connection and IP configuration of the Hub, please refer to the “Connection” section of the “Livox Hub User Manual v1.2” on the Livox official website. The download link is as follows: [Livox Hub manuel v1.2](#)

This example uses static IP configuration, as shown in the following figure:

Up to 9 Livox Lidars can be connected to the Livox Hub at the same time. In this example, a Livox Horizon and a Livox Mid-40 are connected. If you want to collect more Lidars at the same time, you can connect directly.

6.2.2 Collection Under ROS

After the connection is successful, compile in the local ws_livox path:

```
catkin_make
```

Update the current ROS package environment:

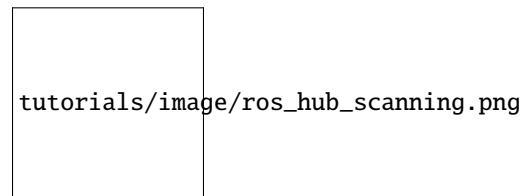
```
source ./devel/setup.bash` `
```

Use the ROS launch file to load the LIvox ros driver

```
roslaunch livox_ros_driver livox_hub_rviz.launch` `
```

All launch files in the Livox driver are located under the path: “ws_livox/src/livox_ros_driver/launch”. Different launch files have different configuration parameter values. Different launch files can be started according to different usage scenarios detailed launch configuration can be found in the description section of [Livox ROS Driver github](#) 4.1 launch configuration file

The livox_hub_rviz.launch will automatically connect to the Livox Hub device, publish pointcloud2 format point cloud data, and automatically load rviz to display real-time point cloud data, as shown in the following figure:

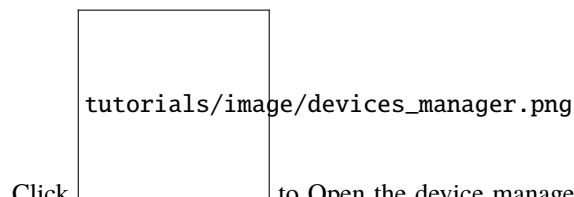


The red part of the point cloud is the image scanned by Livox Horizon, and the purple part is the image scanned by Livox Mid-40

6.2.3 Use Livox Viewer to Collect Data


After the connection is successful, open a terminal in the Livox Viewer folder and run the following command to open the Livox Viewer program:

```
./livox_viewer.sh
```



Click to Open the device manager and select Livox Hub. The IP address of Livox Hub may not match the IP address of the PC when connecting for the first time. If you have configured the Livox Hub and PC with static IP according to the above steps, you can click Network Adapter—refresh Adapter on the toolbar to refresh the configuration.

Open the Hub button and click play to display the real-time scan data of multiple lidars, as shown in the figure below:



tutorials/image/two_lidar_scanning.png

The yellow part of the point cloud is the data scanned by Livox Horizon, and the blue part of the point cloud is the data scanned by Livox Mid-40.

Similar to single lidar data recording, you can click the record button on the toolbar to record the lvx format file, pause playback or click this button again to end the recording.

6.3 Data format and conversion

6.3.1 Introduction to Livox LiDAR data format

Livox LiDAR supports multiple ways to connect and store point cloud data, and its common formats are as follows:

- Bag file: The bag file is a commonly used point cloud data storage format in the ROS system. Livox LiDAR uses the PointCloud2 format and the custom CustomMsg format. This file records each frame of point cloud data and other sensor data during the LiDAR scanning process. Use ROS to connect to Livox LiDAR to scan and store data in this format.
- lvx file: lvx file is the point cloud data stored after using Livox Viewer to connect to Livox LiDAR, which can be viewed and edited in Livox Viewer.
- csv file: comma-separated value file. After using Livox Viewer to connect to Livox LiDAR, lvx point cloud data can be saved in csv format.
- las file: It can be used to save Lidar point data records. After using Livox Viewer to connect to Livox LiDAR, point cloud data can be saved in las format.
- pcd file: can record point cloud data of a certain frame. The data recorded by Livox LiDAR cannot be saved directly in pcd format and needs to be converted.

6.3.2 Data Format Conversion

Convert lvx file to rosbag file and play

- dependencies: Ubuntu 16.04 ROS livox_ros_driver (see the driver for the detailed process of downloading and installing Driver)

livox_ros_driver supports converting lvx pointcloud data files into rosbag files.

Open the terminal under `../ws_livox` and compile

```
$ catkin_make
```

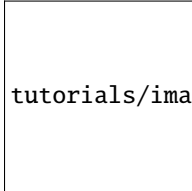
Use the following command to update the current ROS package environment:

```
$ source ../devel/setup.bash
```

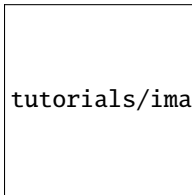
Then use the following command:

```
$ roslaunch livox_ros_driver lvx_to_rosbag.launch lvx_file_path:=""
```

Fill in the local lvx data file path in the double quotes in the above command to run successfully. After successful conversion, a rosbag format file with the same name will be generated under the above path, as shown in the following figure:



tutorials/image/lvx_to_rosbag_01.png



tutorials/image/lvx_to_rosbag_02.png

After obtaining the successfully converted test.bag file, you can use the rviz visualization tool to play, the steps are as follows:

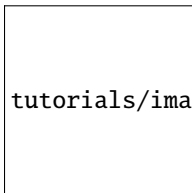
Start ROS:

```
$ roscore
```

rviz

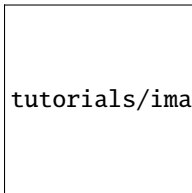
```
$ rosrn rviz rviz
```

After successfully opening rviz, you need to add PointCloud2 and select the corresponding topic. You can use the `rosbag info <filename.bag>` command to view the topic corresponding to the bag file, as shown below:



tutorials/image/rosbag_info.png

Add the corresponding topic in rviz:



tutorials/image/rviz.png

Then play rosbag to play:

```
$ rosbag play <test.bag>
```

Convert lvx file to .las or .csv file

- dependencies Livox Viewerwin or ubuntu

Livox Viewer provides a conversion tool that converts the lvx file into a las or csv file, which is convenient for users to process data. Select Tools->File Converter in the menu bar to open this tool.

Convert rosbag file to pcd file

Rosbag file with PointCloud2 format

Reference:http://wiki.ros.org/pcl_ros

dependencies Ubuntu 16.04 ROS PCL

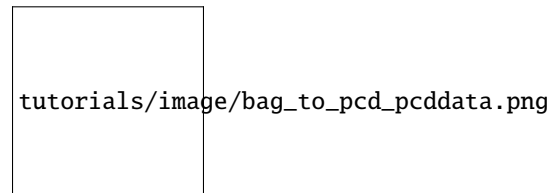
Run ros:

```
$ roscore
```

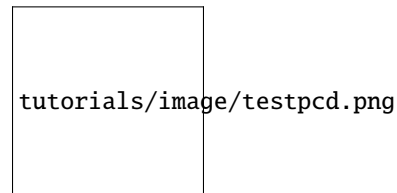
Open the terminal under the storage path of the .bag file and run the command:

```
$ rosruncat pcl_ros bag_to_pcd <input_file.bag> <topic> <output_directory>
```

Where<input_file.bag>is the name of the bag file to be converted.<topic>is the topic information of the bag file, and <output_directory>is the storage address of the pcd file after the conversion is completed.



In the corresponding folder, you can find the pcd file of each frame converted from the rosbag file, whose file name is the timestamp of the current frame:



After the conversion is completed, you can use pcl_viewer to view the pcd file.

```
$ pcl_viewer <filename.pcd>
```

6.4 Lidar Calibration

In multi-sensor intelligent body navigation, multiple sensors are usually installed for data fusion and calculation. In mobile mapping and environment perception, it is usually necessary to arrange multiple Lidars to ensure a larger FOV and provide a stable data source for more robust attitude estimation algorithms and rich environment details. However, no matter what kind of sensor: camera, lidar, millimeter wave radar, they all have their own coordinate system, that is, the data generated by all sensors are based on the sensor's own coordinate system. In order to facilitate algorithm research and testing, it is necessary to convert the data obtained by the respective sensors to the same coordinate system, which is the external parameter calibration of the sensor.

6.4.1 Lidar-Lidar Calibration

Manual

The Livox_Viewer integrates the function of manually calibrating Livox LiDARs, which can easily and accurately calibrate multiple Livox LiDARs with overlapping FOVs. For details, please refer to: [Livox Viewer Manuel](#)

Automatic

Livox provides tools based on C++ to realize automatic calculation of calibration external parameters between multiple lidars. For details, please refer to [Livox calibration](#)

6.4.2 Lidar-Camera Calibration

Livox provides a method to manually calibrate the external parameters between Livox lidar and camera, and it has been verified on Mid-40, Horizon and Tele-15. It contains code for calculating internal camera parameters, obtaining calibration data, optimizing external parameters and lidar camera fusion applications. In this scheme, the corners of the calibration board are used as the calibration targets. Due to the non-repetitive scanning feature of Livox lidar, the density of the point cloud is relatively high, and it is easier to find the accurate position of the corners in the lidar point cloud. The calibration and fusion of camera lidar can also get good results. For details, please refer to: [Camera-LiDAR-Calibration Manual](#)

For the multi-sensor fusion scanning scheme, please refer to the Livox open source algorithm [livox high precision mapping](#) which shows in detail how to synchronize and calibrate the Mid-40 lidar and apx-15 inertial navigation module, and use this component for real-time high-precision mapping.

6.5 Livox SDK

Install Livox-SDK

Install Livox SDK in Linux 32-bit system and Mac

Development Guide

[Time synchronization of Livox LiDARs](#)

[Livox SDK Communication Protocol](#)

Livox Viewer

[Color Coding Strategy](#)

[livox_ros_driver application](#)

[How to use lvx file in ros](#)

[Set publish frequency](#)

[Manual Calibration](#)

Quick Links

[LivoxTech Official website](#)

SUMMARY OF OFFICIAL OPEN SOURCE MATERIALS

7.1 Drive

7.1.1 Livox SDK

Livox SDK is a software development kit designed for all products under Livox. It is developed based on the C/C++ language, follows the Livox SDK communication protocol, and provides users with an easy-to-use C language style interface. Through the Livox SDK, users can quickly connect to Livox products and receive point cloud data. See [Livox SDK](#) for details and installation.

7.1.2 Livox ROS Driver

Livox ROS driver is a brand new ROS package, specifically used to connect LiDAR products produced by Lavo. The driver can run under the ubuntu14.04/16.04/18.04 operating system with ROS environment (indigo, kinetic, melodic) installed. After testing, the hardware platforms that can run the Lanwo ROS driver include: intel x86 mainstream cpu platform, and some ARM64 hardware platforms (eg, nvida TX2/Xavier, etc.). See [Livox ROS Driver](#)

7.1.3 Livox Apollo Driver

The Apollo driver is used to convert the original data file scanned by Livox_Lidar into a standard point cloud format file and write it into the point cloud channel. For details and installation see [Livox Apollo Driver](#)

7.2 Tools

7.2.1 Livox Viewer

Livox Viewer is a software designed specifically for Livox lidar and Livox Hub, which can be used to display the point cloud data of all lidars connected to the computer in real time. Through Livox Viewer, users can easily view, record and store point cloud data for later use. Detailed instructions can be found in the Livox Viewer manual on Livox official website. The download address is as follows:

[Livox Viewer 0.7.0 \(64bit\) - Windows](#)

[Livox Viewer 0.7.0\(64bit\) - Ubuntu 16.04](#)

7.3 Open Source Algorithm

7.3.1 Livox Horizon LOAM

Livox Horizon Loam is a robust, low-equivalent, real-time odometer and drawing software package developed for Livox Lidar. The software package is mainly designed for low-speed scenes (about 5km/h) and solves many key problems such as feature extraction and selection under limited field of view and motion distortion compensation. For details see: [Livox Horizon Loam](#)

7.3.2 Livox Horizon Calibration

Livox-Horizon-based calibration steps for Horizon-Horizon/Horizon-Tele. For details see: [LOAM Horizon Calibration](#)

7.3.3 livox scanner

livox scanner is a 3D scanning solution, which mainly includes a Mid-40 lidar, Manifold 2 computing platform, and a GM6020 motor with encoder. The LiDAR and the motor are installed on the top of the tripod. The Mid-40 will rotate around the Z axis to scan different areas in the space, and the point cloud data will be fused with the motor angle to obtain a 3D image in the space. The system construction process and detailed code can be found here: [livox scanner](#)

7.3.4 livox high precision mapping

This solution uses Mid-40 lidar sensor with a detection distance of 260m, accuracy of 2cm, and non-repetitive scanning mode, combined with the high-precision position and attitude data provided by the APX-15 inertial navigation module, to achieve real-time high-precision mapping function. For details, see: [livox high precision mapping](#)

7.3.5 livox camera lidar calibration

This solution provides a method to manually calibrate the external parameters between Livox lidar and camera, which has been verified on Mid-40, Horizon and Tele-15. It contains code for calculating internal camera parameters, obtaining calibration data, optimizing external parameters and lidar camera fusion applications. In this scheme, the corners of the calibration board are used as the calibration targets. Due to the non-repetitive scanning feature of Livox lidar, the density of the point cloud is relatively high, and it is easier to find the accurate position of the corners in the lidar point cloud. The calibration and fusion of camera lidar can also get good results. For details see: [livox camera lidar calibration](#)

7.3.6 Horizon Highway SLAM

Horizon is a high-performance lidar independently developed by Livox for L3 and L4 autonomous driving. Horizon Highway SLAM is a stable, low-drift and real-time SLAM software package developed based on Horizon for high-speed sports scenes. The algorithm is suitable for speed scenes of 0~80km/h, and solves the technical problems of feature extraction in extremely narrow FOV, motion distortion compensation and multi-sensor fusion to avoid scene degradation. For details see: [Horizon Highway SLAM](#)

7.3.7 Livox-Relocalization

Livox-Relocalization is a relocation software package for Livox Lidar developed by Livox. It can help users load the map collected with Mid-40 and use the map to relocate the lidar location information. For details see: [Livox-Relocalization](#)

7.4 Documentation and Manuals

7.4.1 Livox Viewer User Manual

Download link: [Livox Viewer](#)

7.4.2 Extension cord adapter box installation model

Download link: [model](#)

7.4.3 Livox point cloud characteristics

Download link: [cloud characteristics](#)

7.4.4 Livox Avia

- Livox Avia doc link: [Livox Avia doc](#)

7.4.5 Livox Horizon

- Horizon user manual download link: [Livox Horizon User Manuel](#)
- Horizon Quick Start Guide download link: [Livox Horizon Quick Start Guide](#)
- Livox Horizon doc link: [Livox Horizon doc](#)

7.4.6 Livox Mid

- Livox Series User Manual v1.0 download link : [Livox Mid Series User Manual v1.2](#)
- Livox Mid-40 Quick Start Guide v1.4 download link : [Livox Mid-40 Quick Start Guide v1.4 multi](#)
- Livox Mid-100 Quick Start Guide v1.4 download link: [Livox Mid-100 Quick Start Guide v1.4 multi](#)
- Mid-40 3D Model and FOV Shape download link: [MID-40 3D Model and FOV Shape](#)
- Mid-100 3D Model and FOV Shape download link: [MID-100 3D Model and FOV Shape](#)
- Livox Mid-70 doc link: [Livox Mid-70 doc](#)

7.4.7 Livox Tele-15

- Tele-15 User Manual download link: [Tele-15 User Manual](#)
- Tele-15 Quick Start Guide download link: [Tele-15 Quick Start Guide](#)
- Tele-15 3D Model and FOV Shape download link: [Tele-15 3D Model and FOV Shape](#)
- Tele-15 W/O Dissipation Module 3D Model download link: [Tele-15 W/O Dissipation Module 3D Model](#)
- Livox Tele-15 doc link: [Livox Tele-15 doc](#)

7.4.8 Livox Hub

- Livox Hub User Manual v1.2 download link: [Livox Hub User Manual v1.2](#)
- Livox Hub Quick Start Guide v1.0 download link [Livox Hub Quick Start Guide](#)
- Livox Hub 3D model download link [Livox Hub 3D model](#)

7.5 Point Cloud Data

7.5.1 Livox Horizon point cloud data

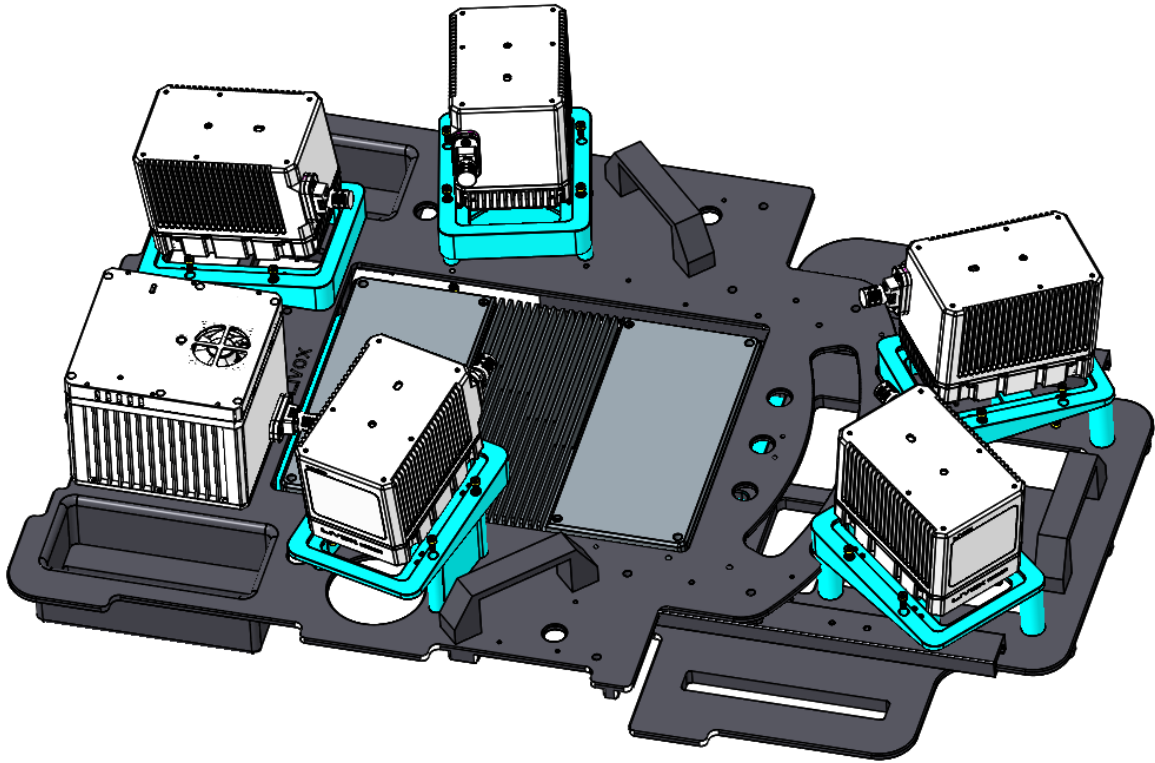
- Data package 1 download link [Livox Horizon Point Cloud Data 1](#)
- Data package 2 download link [Livox Horizon Point Cloud Data 2](#)

7.5.2 Livox Mid-100 Point Cloud Data

- Data package 1 download link: [Livox Mid-100 Point Cloud Data 1](#)
- Data package 2 download link: [Livox Mid-100 Point Cloud Data 2](#)

7.6 Livox Vehicle Platform

- Vehicle Platform for Livox lidar



download link: [Vehicle Platform DOC](#)

SUMMARY OF UNOFFICIAL OPEN SOURCE MATERIALS

8.1 Paper

Low-cost Retina-like Robotic Lidar Based on Incommensurable Scanning

- Abstract High performance lidars are essential in autonomous robots such as self-driving cars, automated ground vehicles and intelligent machines. Traditional mechanical scanning lidars offer superior performance in autonomous vehicles, but the potential mass application is limited by the inherent manufacturing difficulty. We propose a robotic lidar sensor based on incommensurable scanning that allows straightforward mass production and adoption in autonomous robots. Some unique features are additionally permitted by this incommensurable scanning. Similar to the fovea in human retina, this lidar features a peaked central angular density, enabling in applications that prefers eye-like attention. The incommensurable scanning method of this lidar could also provide a much higher resolution than conventional lidars which is beneficial in robotic applications such as sensor calibration. Examples making use of these advantageous features are demonstrated.
- [link:Low-cost Retina-like Robotic Lidar Based on Incommensurable Scanning](#)

Loam_livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV

- Abstract LiDAR odometry and mapping (LOAM) has been playing an important role in autonomous vehicles, due to its ability to simultaneously localize the robot's pose and build high-precision, high-resolution maps of the surrounding environment. This enables autonomous navigation and safe path planning of autonomous vehicles. In this paper, we present a robust, real-time LOAM algorithm for LiDARs with small FoV and irregular samplings. By taking effort on both front-end and back-end, we address several fundamental challenges arising from such LiDARs, and achieve better performance in both precision and efficiency compared to existing baselines. To share our findings and to make contributions to the community, we open source our codes on Github.
- [linkLoam_livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV](#)
- [codeGithub](#)

CamVox: A Low-cost and Accurate Lidar-assisted Visual SLAM System

- Abstract Combining lidar in camera-based simultaneous localization and mapping (SLAM) is an effective method in improving overall accuracy, especially at a large scale outdoor scenario. Recent development of low-cost lidars (e.g. Livox lidar) enable us to explore such SLAM systems with lower budget and higher performance. In this paper we propose CamVox by adapting Livox lidars into visual SLAM (ORB-SLAM2) by exploring the lidars' unique features. Based on the non-repeating nature of Livox lidars, we propose an automatic lidar-camera calibration method that will work in uncontrolled scenes. The long depth detection range also benefit a more efficient mapping. Comparison of CamVox with visual SLAM (VINS-mono) and lidar SLAM (LOAM) are evaluated on the same dataset to demonstrate the performance. We open sourced our hardware, code and dataset on GitHub.
- [link:CamVox: A Low-cost and Accurate Lidar-assisted Visual SLAM System](#)

- [codeGithub](#)

A fast, complete, point cloud based loop closure for LiDAR odometry and mapping

- AbstractThis paper presents a loop closure method to correct the long-term drift in LiDAR odometry and mapping (LOAM). Our proposed method computes the 2D histogram of keyframes, a local map patch, and uses the normalized cross-correlation of the 2D histograms as the similarity metric between the current keyframe and those in the map. We show that this method is fast, invariant to rotation, and produces reliable and accurate loop detection. The proposed method is implemented with careful engineering and integrated into the LOAM algorithm, forming a complete and practical system ready to use. To benefit the community by serving a benchmark for loop closure, the entire system is made open source on Github.
- [linkA fast, complete, point cloud based loop closure for LiDAR odometry and mapping](#)
- [codeGithub](#)

BALM: Bundle Adjustment for Lidar Mapping

- AbstractWe propose a framework for bundle adjustment (BA) on sparse lidar points and incorporate it to a lidar odometry and mapping (LOAM) to lower the drift. A local BA on a sliding window of keyframes has been widely used in visual SLAM and has proved to be very effective in lowering the drift. But in lidar mapping, BA method is hardly used because the sparse feature points (e.g., edge and plane) in a lidar point-cloud make the exact feature matching impossible. Our method is to enforce feature points lie on the same edge or plane by minimizing the eigenvalue of the covariance matrix. To speed up the optimization, we derive the analytical derivatives, up to second order, in closed form. Moreover, we propose a novel adaptive voxelization method to search feature correspondence efficiently. The proposed formulations are incorporated into a LOAM back-end for map refinement. Results show that, although as a back-end, the local BA can be solved very efficiently, even in real-time at 10Hz when optimizing 20 scans of point-cloud. The local BA also considerably lowers the LOAM drift. Our implementation of the BA optimization and LOAM are open-sourced to benefit the community.
- [linkBALM: Bundle Adjustment for Lidar Mapping](#)
- [codeGithub](#)

A decentralized framework for simultaneous calibration, localization and mapping with multiple LiDARs

- AbstractLiDAR is playing a more and more essential role in autonomous driving vehicles for objection detection, self localization and mapping. A single LiDAR frequently suffers from hardware failure (e.g., temporary loss of connection) due to the harsh vehicle environment (e.g., temperature, vibration, etc.), or performance degradation due to the lack of sufficient geometry features, especially for solid-state LiDARs with small field of view (FoV). To improve the system robustness and performance in self-localization and mapping, we develop a decentralized framework for simultaneous calibration, localization and mapping with multiple LiDARs. Our proposed framework is based on an extended Kalman filter (EKF), but is specially formulated for decentralized implementation. Such an implementation could potentially distribute the intensive computation among smaller computing devices or resources dedicated for each LiDAR and remove the single point of failure problem. Then this decentralized formulation is implemented on an unmanned ground vehicle (UGV) carrying 5 low-cost LiDARs and moving at 1.3m/s in urban environments. Experiment results show that the proposed method can successfully and simultaneously estimate the vehicle state (i.e., pose and velocity) and all LiDAR extrinsic parameters. The localization accuracy is up to 0.2% on the two datasets we collected. To share our findings and to make contributions to the community, meanwhile enable the readers to verify our work.
- [linkA decentralized framework for simultaneous calibration, localization and mapping with multiple LiDARs](#)
- [codeGithub](#)

Initial Investigation of a Low-Cost Automotive LIDAR System

- AbstractThis investigation focuses on the performance assessment of a low-cost automotive LIDAR, the Livox Mid-40 series. The work aims to examine the qualities of the sensor in terms of ranging, repeatability and accuracy. Towards these aims a series of experiments were carried out based on previous research of low-cost sensor accuracy, LIDAR accuracy investigation and TLS calibration experiments. The Livox Mid-40 series offers

the advantage of a long-range detection beyond 200m at a remarkably low cost. The preliminary results of the tests for this sensor indicate that it can be used for reality capture purposes such as to obtain coarse as-built plans and volume calculations to mention a few. Close-range experiments were conducted in an indoor laboratory setting. Long-range experiments were performed outdoors towards a building façade. Reference values in both setups were provided with a Leica RTC 360 terrestrial LIDAR system. In the close-range experiments a cross section of the point cloud shows a significant level of noise in the acquired data. At a stand-off distance of 5 m the length measurement tests reveal deviations of up to 11 mm to the reference values. Range measurement was tested up to 130 meters and shows ranging deviations of up to 25 millimetres. The authors recommend further investigation of the issues in radiometric behaviour and material reflectivity. Also, more knowledge about the internal components is needed to understand the causes of the concentric ripple effect observed at close ranges. Another aspect that should be considered is the use of targets and their design as the non-standard scan pattern prevents automated detection with standard commercial software.

- [linkInitial Investigation of a Low-Cost Automotive LIDAR System](#)

Towards high-performance solid-state-lidar-inertial odometry and mapping

- **Abstract**We present a novel tightly-coupled LiDAR-inertial odometry and mapping scheme for both solid-state and mechanical LiDARs. As frontend, a feature-based lightweight LiDAR odometry provides fast motion estimates for adaptive keyframe selection. As backend, a hierarchical keyframe-based sliding window optimization is performed through marginalization for directly fusing IMU and LiDAR measurements. For the Livox Horizon, a newly released solid-state LiDAR, a novel feature extraction method is proposed to handle its irregular scan pattern during preprocessing. LiLi-OM (Livox LiDAR-inertial odometry and mapping) is real-time capable and achieves superior accuracy over state-of-the-art systems for both LiDAR types on public data sets of mechanical LiDARs and in experiments using the Livox Horizon. Source code and recorded experimental data sets are available on Github.

- [linkTowards High-Performance Solid-State-LiDAR-Inertial Odometry and Mapping](#)
- [codeGithub](#)

Accuracy Assessment and Calibration of Low-Cost Autonomous LIDAR Sensors

- **Abstract**A number of low-cost, small form factor, high resolution lidar sensors have recently been commercialized in an effort to fill the growing needs for lidar sensors on autonomous vehicles. These lidar sensors often report performance as range precision and angular accuracy, which are insufficient to characterize the overall quality of the point clouds returned by these sensors. Herein, a detailed geometric accuracy analysis of two representative autonomous sensors, the Ouster OSI-64 and the Livox Mid-40, is presented. The scanners were analyzed through a rigorous least squares adjustment of data from the two sensors using planar surface constraints. The analysis attempts to elucidate the overall point cloud accuracy and presence of systematic errors for the sensors over medium (<40 m) ranges.

- [linkAccuracy Assessment and Calibration of Low-Cost Autonomous LIDAR Sensors](#)

UAV LiDAR Point Cloud Segmentation of A Stack Interchange with Deep Neural Networks

- **Abstract**Stack interchanges are essential components of transportation systems. Mobile laser scanning (MLS) systems have been widely used in road infrastructure mapping, but accurate mapping of complicated multi-layer stack interchanges are still challenging. This study examined the point clouds collected by a new Unmanned Aerial Vehicle (UAV) Light Detection and Ranging (LiDAR) system to perform the semantic segmentation task of a stack interchange. An end-to-end supervised 3D deep learning framework was proposed to classify the point clouds. The proposed method has proven to capture 3D features in complicated interchange scenarios with stacked convolution and the result achieved over 93% classification accuracy. In addition, the new low-cost semi-solid-state LiDAR sensor Livox Mid-40 featuring a incommensurable rosette scanning pattern has demonstrated its potential in high-definition urban mapping.

- [linkUAV LiDAR Point Cloud Segmentation of A Stack Interchange with Deep Neural Networks](#)

FAST-LIO: A Fast, Robust LiDAR-inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter

- **Abstract** This paper presents a computationally efficient and robust LiDAR-inertial odometry framework. We fuse LiDAR feature points with IMU data using a tightly-coupled iterated extended Kalman filter to allow robust navigation in fast-motion, noisy or cluttered environments where degeneration occurs. To lower the computation load in the presence of large number of measurements, we present a new formula to compute the Kalman gain. The new formula has computation load depending on the state dimension instead of the measurement dimension. The proposed method and its implementation are tested in various indoor and outdoor environments. In all tests, our method produces reliable navigation results in real-time: running on a quadrotor onboard computer, it fuses more than 1,200 effective feature points in a scan and completes all iterations of an iEKF step within 25 ms. Our codes are open-sourced online.
- [linkFAST-LIO: A Fast, Robust LiDAR-inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter](#)
- [codeGithub](#)

VIO-UWB-Based Collaborative Localization and Dense Scene Reconstruction within Heterogeneous Multi-Robot Systems

- **Abstract** Effective collaboration in multi-robot systems requires accurate and robust estimation of relative localization: from cooperative manipulation to collaborative sensing, and including cooperative exploration or cooperative transportation. This paper introduces a novel approach to collaborative localization for dense scene reconstruction in heterogeneous multi-robot systems comprising ground robots and micro-aerial vehicles (MAVs). We solve the problem of full relative pose estimation without sliding time windows by relying on UWB-based ranging and Visual Inertial Odometry (VIO)-based egomotion estimation for localization, while exploiting lidars onboard the ground robots for full relative pose estimation in a single reference frame. During operation, the rigidity eigenvalue provides feedback to the system. To tackle the challenge of path planning and obstacle avoidance of MAVs in GNSS-denied environments, we maintain line-of-sight between ground robots and MAVs. Because lidars capable of dense reconstruction have limited FoV, this introduces new constraints to the system. Therefore, we propose a novel formulation with a variant of the Dubins multiple traveling salesman problem with neighborhoods (DMTSPN) where we include constraints related to the limited FoV of the ground robots. Our approach is validated with simulations and experiments with real robots for the different parts of the system.
- [linkVIO-UWB-Based Collaborative Localization and Dense Scene Reconstruction within Heterogeneous Multi-Robot Systems](#)
- [codeGithub](#)

A Survey of Simultaneous Localization and Mapping with an Envision in 6G Wireless Networks

- **Abstract** Simultaneous Localization and Mapping (SLAM) achieves the purpose of simultaneous positioning and map construction based on self-perception. The paper makes an overview in SLAM including Lidar SLAM, visual SLAM, and their fusion. For Lidar or visual SLAM, the survey illustrates the basic type and product of sensors, open source system in sort and history, deep learning embedded, the challenge and future. Additionally, visual inertial odometry is supplemented. For Lidar and visual fused SLAM, the paper highlights the multi-sensors calibration, the fusion in hardware, data, task layer. The open question and forward thinking with an envision in 6G wireless networks end the paper. The contributions of this paper can be summarized as follows: the paper provides a high quality and full-scale overview in SLAM. It's very friendly for new researchers to hold the development of SLAM and learn it very obviously. Also, the paper can be considered as a dictionary for experienced researchers to search and find new interesting orientation.
- [linkA Survey of Simultaneous Localization and Mapping with an Envision in 6G Wireless Networks](#)

Review on 3D Lidar Localization for Autonomous Driving Cars

- **Abstract** LiDAR sensors are becoming one of the most essential sensors in achieving full autonomy for self driving cars. LiDARs are able to produce rich, dense and precise spatial data, which can tremendously help in localizing and tracking a moving vehicle. In this paper, we review the latest finding in 3D LiDAR localization for autonomous driving cars, and analyse the results obtained by each method, in an effort to guide the research community towards the path that seems to be the most promising.
- [linkReview on 3D Lidar Localization for Autonomous Driving Cars](#)

ACSC: Automatic Calibration for Non-repetitive Scanning Solid-State LiDAR and Camera Systems

- Abstract Recently, the rapid development of Solid-State LiDAR (SSL) enables low-cost and efficient obtainment of 3D point clouds from the environment, which has inspired a large quantity of studies and applications. However, the non-uniformity of its scanning pattern, and the inconsistency of the ranging error distribution bring challenges to its calibration task. In this paper, we proposed a fully automatic calibration method for the non-repetitive scanning SSL and camera systems. First, a temporal-spatial-based geometric feature refinement method is presented, to extract effective features from SSL point clouds; then, the 3D corners of the calibration target (a printed checkerboard) are estimated with the reflectance distribution of points. Based on the above, a target-based extrinsic calibration method is finally proposed. We evaluate the proposed method on different types of LiDAR and camera sensor combinations in real conditions, and achieve accuracy and robustness calibration results. The code is available at this [https URL](#).
- [linkACSC: Automatic Calibration for Non-repetitive Scanning Solid-State LiDAR and Camera Systems](#)
- [codeGithub](#)

Autonomous Dam Surveillance Robot System Based on Multi-Sensor Fusion

- Abstract Dams are important engineering facilities in the water conservancy industry. They have many functions, such as flood control, electric power generation, irrigation, water supply, shipping, etc. Therefore, their long-term safety is crucial to operational stability. Because of the complexity of the dam environment, robots with various kinds of sensors are a good choice to replace humans to perform a surveillance job. In this paper, an autonomous system design is proposed for dam ground surveillance robots, which includes general solution, electromechanical layout, sensors scheme, and navigation method. A strong and agile skid-steered mobile robot body platform is designed and created, which can be controlled accurately based on an MCU and an onboard IMU. A novel low-cost LiDAR is adopted for odometry estimation. To realize more robust localization results, two Kalman filter loops are used with the robot kinematic model to fuse wheel encoder, IMU, LiDAR odometry, and a low-cost GNSS receiver data. Besides, a recognition network based on YOLO v3 is deployed to realize real-time recognition of cracks and people during surveillance. As a system, by connecting the robot, the cloud server and the users with IOT technology, the proposed solution could be more robust and practical.
- [linkAutonomous Dam Surveillance Robot System Based on Multi-Sensor Fusion](#)

DATASET

9.1 Livox Simu-dataset v1.0 CN

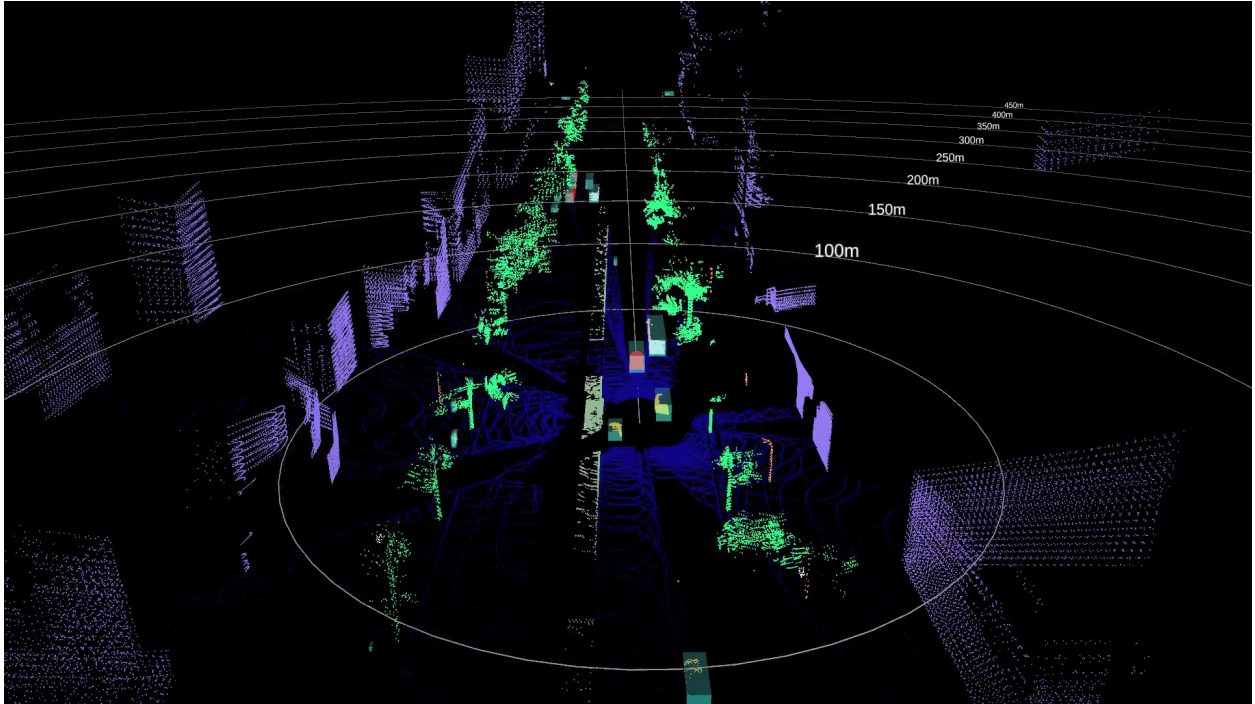
Introduction:

Livox Simu-dataset contains point cloud data and corresponding annotations generated based on the autonomous driving simulator, and supports 3D object detection and point cloud semantic segmentation tasks. The sensor contains 5 Horizon lidars and 1 Tele-15 lidar. The entire dataset contains 14,445 frames of 360° Lidar point cloud data, 3D bounding box annotations for 6 types of objects, and semantic annotations for 14 categories. The simulated scene contains a variety of vehicle and pedestrian models and corresponding traffic flow. In addition, a variety of traffic lights, traffic signs, partitions (isolated railings, green belts, isolated piers, etc.), trees and buildings make the entire scene closer to the actual driving scene.

Simulated scene:



Annotation demo:



Data format and download link:

PLEASE READ THESE TERMS CAREFULLY BEFORE DOWNLOADING THE DATASET. DOWNLOADING OR USING THE DATASET MEANS YOU ACCEPT THESE TERMS: [GNU General Public License v3.0](#)

[Readme](#)

[Dataset](#)

[Data rosbag](#)

Object detection demo:

[Livox_detection_simu](#)

CHAPTER TEN

FAQ

If you have any suggestions or comments on this document, please contact us on [Livox-SDK Github](#) [Livox LiDARs Google Groups](#) or send your questions to Livox's email "cs@livoxtech.com".