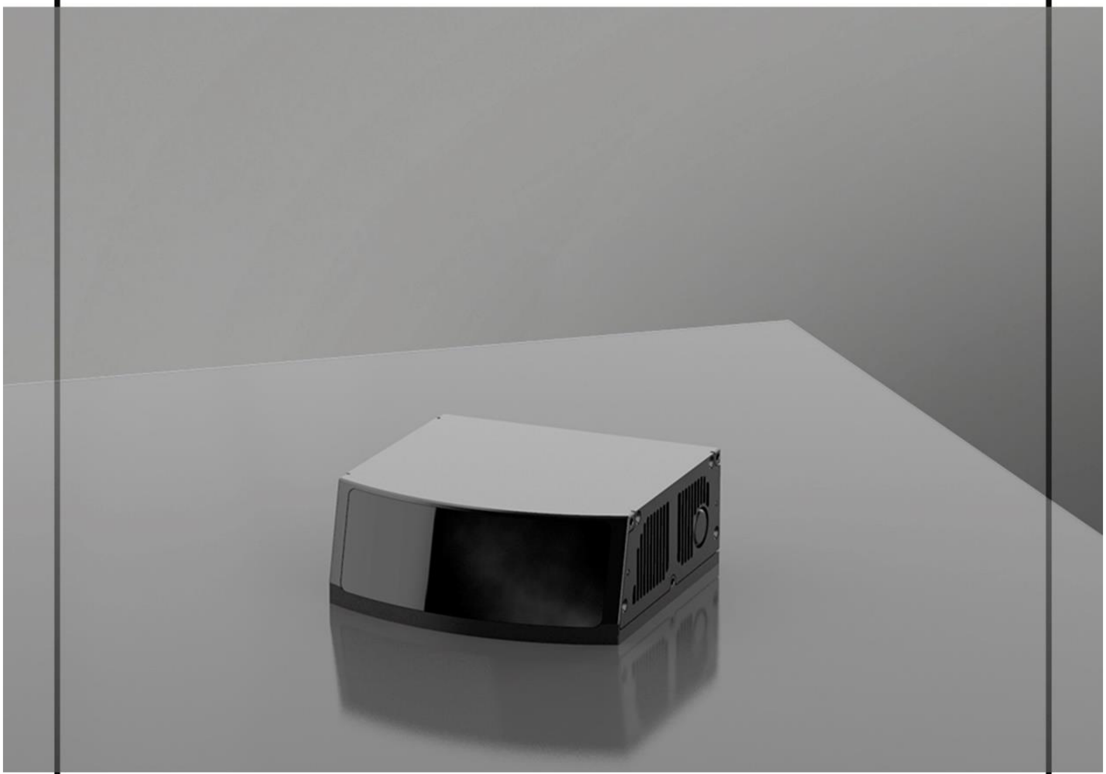
robosense® LiDAR

User Guide



*【Executive Standard】 Q/RS 001-2021

Revision History

Revision	Content	Date	Edited by
1.0	Initial issue	2020-11-26	PD
1.1	Add the X, Y and Z coordinate calculation formula in Section 2.1.2.1 Add Appendix A RSView Add Appendix B Driver & SDK Add Appendix C The use of MEMS Tool Correct some improper description	2020-12-02	PD
1.2	Update Product Specifications Table	2021-02-23	PD
1.3	Update the explanation of MSOP and DIFOP Update Figure 2 Add the Appendix D	2021-03-11	PD
1.4	Add the network demand for data transfer rate in Section 2 Update Vertical FOV footnote & MSOP data definition Add types of Interface Box in Section 3.1 Add Section 3.2 State Machine of LiDAR Add Chapter 4 Time Synchronization Update Appendix D Dimension Revise some improper description	2021-06-07	PD
1.5	Revise working voltage range	2021-07-16	PD
1.6	Revise Chapter 1 Ranging Accuracy of Product Parameters Revise Chapter 2 Default Port Configuration Revise Chapter 2.2 offset in DIFOP table and content Update Chapter 2.2 Description of DIFOP Time Synchronization Status Update Chapter 4 Time Synchronization Description, gPTP mode by default Update Appendix A RSview port setting guidance Update Appendix C Autosar Tool and Operation Guide	2021-11-26	PD
1.7	Add Executive Standard in home page	2021-12-03	PD

Table of Contents

1 Product Specifications

RS-LiDAR-M1, adopting the MEMS solid-state LiDAR technology, has achieved long measuring distance up to 200 meters (150m @ 10%), high data rate of 750,000 points/sec (single return) and 1,500,000 points/sec (dual return) data output, a horizontal FOV of 120° (-60.0°~+60.0°), a vertical FOV of 25° (-12.5°~+12.5°).

Table 1: Product Specifications

Sensor	• Time of Flight (TOF) ranging, including reflection intensity value • Ranging distance: 0.5m ~200m(150m@10% NIST)[1] • Ranging Precision: ± 5cm@1 sigma[2] • FOV(vertical): 25° (-12.5°~+12.5°)[3] • Angular resolution(vertical): average 0.2°[4] • FOV(horizontal): 120° (-60.0°~+60.0°) • Angular resolution(horizontal): average 0.2°[4] • Frame rate: 10Hz
Laser	• Class 1 eye safe • Wavelength: 905nm
Output	• ~750,000 points/second (single return mode) • ~1,500,000 points/second (dual return mode) • 1000Base-T1 Gigabit Ethernet • UDP package contains Three-dimensional space coordinates, reflection intensity, time stamp, etc.

1 The ranging capability of 150 meters is measured with the 10% NIST diffuse reflector as the target, the test results may be affected by the environment conditions, including but not limited to factors such as ambient temperature and lights;

2 The ranging precision is tested in the range of 10m~100m with 50% NIST diffuse reflector as the target. The test results may be affected by the environment conditions, including but not limited to factors such as ambient temperature and target distance. The precision value is applicable to most channels, but difference may exist between some channels.

3 The five channels of RS-LiDAR-M1 are horizontally arranged, with staggered positions vertically; The maximum envelope vertical FOV of a single channel is 25.2 °; Since the FOV of five channels are present irregularly, based on the maximum envelope principle, the vertical FOV will be calculated as 35.79 °；

4 The vertical & horizontal angular resolution is not uniform in the entire FOV, the average angular resolution is 0.2°;

•

Mechanical/electronic operation	● Power consumption: 15w[5] ● Working voltage: 9~32VDC ● Weight: about 0.73kg (not including data cable) ● Dimensions: Length 110mm * Width 108mm * Height 45m ● Protection level: IP67, IP6K9K ● Operating temperature range: -40°C~85°C(Forced convection is required for long hours of work)[6] ● Storage temperature: -40°C ~105°C

5 The device power consumption is tested when the device is working stably, and the results may be affected by external environment conditions, including but not limited to factors such as ambient temperature, target distance, target reflectivity, etc.

6 The operating temperature of the device may be affected by external environment conditions, including but not limited to factors such as solar radiation, airflow changes, etc.

2 Communication Protocol

The communication between RS-LiDAR-M1 and the computer is through Ethernet, and uses UDP protocol. There are two types of output packets: MSOP packet and DIFOP packet. All MSOP packets involved in this document are with fixed length of 1210 bytes, DIFOP packets are with fixed length of 256 bytes. In single return mode, the output data includes 6300 MSOP packets and 1 DIFOP packet, which demands the data transfer rate no less than 58.2 Mbps. In dual return mode, the rate must be no less than 116.4 Mbps. RS-LiDAR-M1 network parameters are configurable, and the factory default IP and fixed client port number are set as listed in the table below:

Table 2: Factory default network configuration

	IP Address	**MSOP Port Number**	**DIFOP Port Number**
RS-LiDAR-M1	192.168.1.200	/	/
Computer	192.168.1.102	6699	7788

The default MAC address of the LiDAR is initially set at the factory, and the MAC address of each LiDAR is unique.

When using the LiDAR, you need to set the computer's IP to the same network segment as the LiDAR, for example, 192.168.1.x (the range of x is 1~254), and the subnet mask as 255.255.255.0. If you don't know the network configuration information of the LiDAR, please set the host computer subnet mask to 0.0.0.0, connect to the LiDAR and use Wireshark to capture the LiDAR output packet for analysis.

The communication protocol between RS-LiDAR-M1 and the computer is mainly divided into two categories. See the table below for the protocol list.

- The main data stream output protocol (MSOP), encapsulates the distance, angle, reflectivity and other information measured by the LiDAR into a package and outputs it to the computer;
- LiDAR information output protocol (DIFOP), outputs various configuration information of the LiDAR currently in use to the computer.

Table 3: List of communication protocols

Protocol	**Abbreviation**	**Function**	**Type**	**Packet size**
Main Data Stream Output	MSOP	Output measured	UDP	1210 Bytes

Protocol		data		
LiDAR Information Output Protocol	DIFOP	Output device information	UDP	256 Bytes

Note: The following chapters describe and define the payload (MSOP package of 1210 bytes and DIFOP package of 256 bytes) of the protocols.

2.1 Main Data Stream Output Protocol (MSOP)

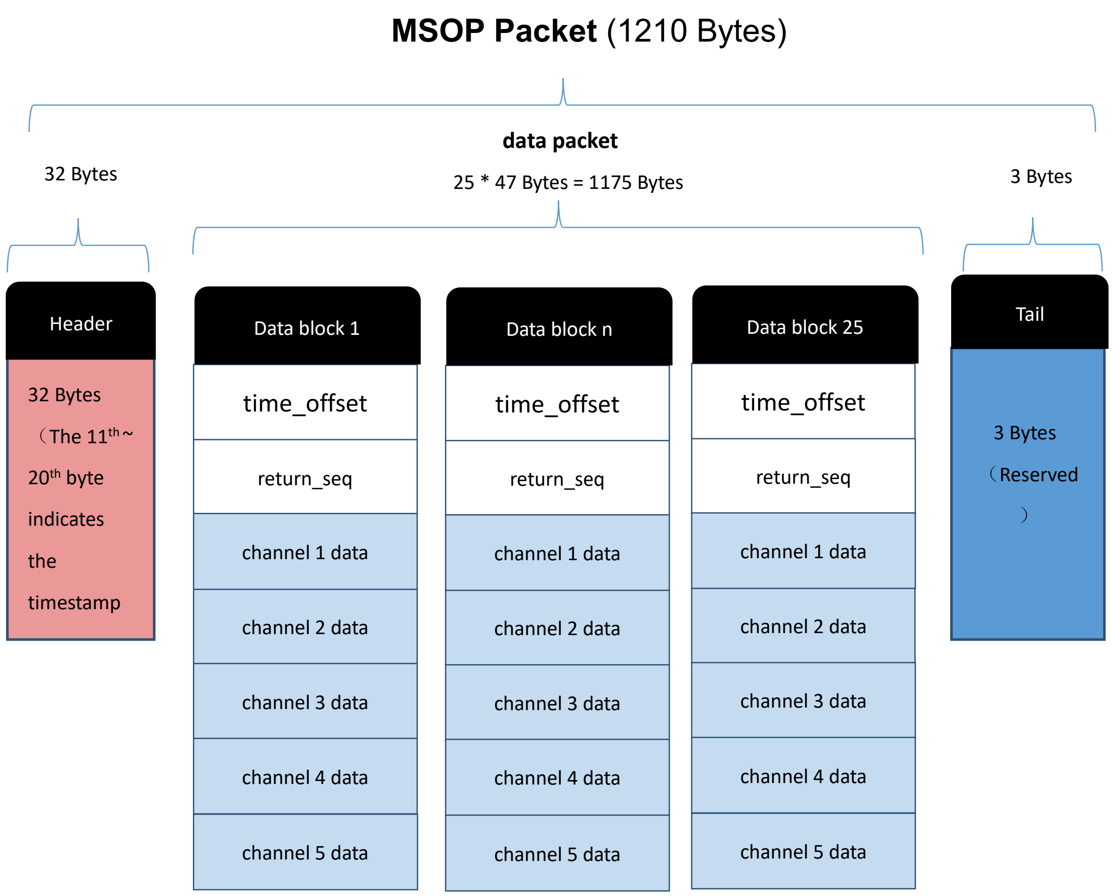
Main data Stream Output Protocol is abbreviated as MSOP.

I/O type: LiDAR output, computer analysis.

Default port number: 6699.

The MSOP packets output three-dimensional measurement related data, including laser ranging value, return reflectivity value, vertical angle, horizontal angle and time stamp. The payload length of the MSOP packet is 1210 bytes, which consists of a synchronization header of 32 bytes, a data packet of 1175 bytes (a total of 25 data blocks of 47 bytes), and a tail of 3 bytes.

The basic structure of the MSOP packet is as shown in the figure below:



•

Figure 1: MSOP Packet Structure

2.1.1 Header

The header is 32-byte long, and is used for identification of the starting position of data, packet counting, UDP communication reservation, and time stamp storage. The detailed definition is as follows:

Table 4: MSOP Header

Header（32 Bytes)				
pkt_header	pkt_psn	protocol version	wave_mode	time_sync_mode
4 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte
timestamp	reserved	lidar_type	mems_tmp	
10 Bytes	10 Bytes	1 Byte	1 Byte	

pkt_header: can be used as a packet inspection sequence, and the identification header is 0x55, 0xaa, 0x5a, 0xa5.

pkt_psn: Packets Sequence Number, packet counting in a circular counting manner, the count value of the first data packet of each frame is 1, the count value of the last data packet of each frame is the maximum value.

protocol version: version number of the UDP communication protocol

wave_mode: return mode flag, 0 means dual return mode, 1 - N/A, 2 - N/A, 3 - N/A, 4 means strongest return, 5 means last return, 6 means first return.

time_sync_mode: time synchronization mode:

0x00 currently using the LiDAR internal timing

0x01 currently using 1PPS for sub-second reset in full seconds

0x02 currently using PTP time synchronization mode

Timestamp: store timestamps. The defined timestamp is used to record the system time. The high 6 bytes are the second bits, and the low 4 bytes are the microsecond bits.

reserved: reserved bit

lidar_type: the type of LiDAR, default value is 0x10

mems_tmp: mems temperature, Temp=mems_tmp-80; namely when mems_tmp value is 0, mems temperature is -80℃; when the value is 255, the temperature is 175℃.

•

2.1.2 Data Packet

The data packet in the MSOP packet stores the data measured by the LiDAR. It has a total of 1175 bytes consisting of 25 data blocks, each data block has 47 bytes.

In single return mode, each data block represent the complete measurement data measured by a group of 5 laser channels at one time. Each data block stores the data of one transmission in the single return mode.

In dual return mode, the odd numbered MSOP packets store the data of the first return, including 25 data blocks. The even numbered MSOP packets store the data of the second return, including 25 data blocks. The first and second returns are stored by turns in sequence. The type of returned packets could be judged according to the ‘returen_seq’ value in the data block, please check Table 5 for detailed definition. Every two MSOP packets form a complete measurement. The total number of data points in a dual return mode is twice that of a single return mode.

The detailed definition is as follows:

Table 5： Definition of data block in MSOP packet

data block N (47 Bytes)			
content	offset(byte)	byte	instruction
time_offset	0	1	The time offset of all points in the block relative to the timestamp of the packet, the time of this group of points equals to timestamp+time_offset
return_seq	1	1	Return sequence. In single-return mode, this flag is always 0; in dual-return mode, the first return (closer) is represented by 0x1, and the second return (further) is represented by 0x2
ch1_radius	2	2	In the polar coordinate system, the radial distance value of the channel 1 points, the distance resolution is 5mm
ch1_elevation	4	2	In the polar coordinate system, the vertical angle of the channel 1 points, the resolution is 0.01°
ch1_azimuth	6	2	In the polar coordinate system, the horizontal angle of the channel 1 points, the resolution is 0.01°
ch1_intensity	8	1	Reflection intensity value of the channel 1 points, the value range is 0~255

•

resev.	9	2	Reserved bits
ch2_radius	11	2	In the polar coordinate system, the radial distance value of the channel 2 points, the distance resolution is 5mm
ch2_elevation	13	2	In the polar coordinate system, the vertical angle of the channel 2 points, the resolution is 0.01°
ch2_azimuth	15	2	In the polar coordinate system, the horizontal angle of the channel 2 points, the resolution is 0.01°
ch2_intensity	17	1	Reflection intensity value of the channel 2points, the value range is 0~255
resev.	18	2	Reserved bits
ch3_radius	20	2	In the polar coordinate system, the radial distance value of the channel 3 points, the distance resolution is 5mm
ch3_elevation	22	2	In the polar coordinate system, the vertical angle of the channel 3 points, the resolution is 0.01°
ch3_azimuth	24	2	In the polar coordinate system, the horizontal angle of the channel 3 points, the resolution is 0.01°
ch3_intensity	26	1	Reflection intensity value of the channel 3 points, the value range is 0~255
resev.	27	2	Reserved bits
ch4_radius	29	2	In the polar coordinate system, the radial distance value of the channel 4 points, the distance resolution is 5mm
ch4_elevation	31	2	In the polar coordinate system, the vertical angle of the channel 4 points, the resolution is 0.01°
ch4_azimuth	33	2	In the polar coordinate system, the horizontal angle of the channel 4 points, the resolution is 0.01°
ch4_intensity	35	1	Reflection intensity value of the channel 4 points, the value range is 0~255
resev.	36	2	Reserved bits
ch5_radius	38	2	In the polar coordinate system, the radial distance value of the channel 5 points, the distance resolution is 5mm
ch5_elevation	40	2	In the polar coordinate system, the vertical angle of the channel 5 points, the resolution is 0.01°

•

ch5_azimuth	42	2	In the polar coordinate system, the horizontal angle of the channel 5 points, the resolution is 0.01°
ch5_intensity	44	1	Reflection intensity value of the channel 5 points, the value range is 0~255
resev.	45	2	Reserved bits

N is the N^{th} data block in any MSOP packet.

time_offset: the time offset of all points in the N^{th} block relative to the time stamp of the packet. The time of this group of points equals time stamp+time_offset.

return_seq: return sequence. In single-return mode, this flag is always 0; in dual-return mode, the first return (closer) is represented by 0x1, and the second return (further) is represented by 0x2

n is the n^{th} channel in the N^{th} data block, n=1, 2, 3, 4, 5, which contains data as follows:

chn_radius: the radial distance value of the points of channel n in the polar coordinate system, the resolution is 5mm.

chn_elevation: the vertical angle of the channel n points in polar coordinate system, the resolution is 0.01°

chn_azimuth: the horizontal angle of the channel n points in polar coordinate system, the resolution is 0.01°

chn_intensity: reflection intensity value of the channel n points, the value range is 0~255.

2.1.2.1 Channel Data Definition

The channel data is 9-byte long, with the radial distance of this channel occupying 2 bytes, the elevation angle occupying 2 bytes, the horizontal angle occupying 2 bytes, the reflection intensity value occupying 1 byte, and 2 bytes reserved.

Detailed definitions are as follows:

Table 6：Definition of channel data in data block

channel data（9 Bytes)						
chn_radius (2 Bytes)		chn_elevation (2 Bytes)		chn_azimuth (2 bytes)		chn_intensity (1 Byte)
R1 [15:8]	R2 [7:0]	E1[15:8]	E2[7:0]	A1[15:8]	A2[7:0]	Intensity[7:0]

resv. (2 Bytes)		
r1 [15:8]	r2 [7:0]	

Take the radial distance calculation as an example：

Chn_radius is 2-byte long, the unit is centimeters (cm), and the resolution is 0.5 cm.

Get the hexadecimal number of the radius value of a channel in the data packet: R1 is 0x03, R2 is 0xfc

0x03 is the high digit of the distance, converted to decimal is 3, 0xfc is the low digit of the distance, converted into decimal is 252.

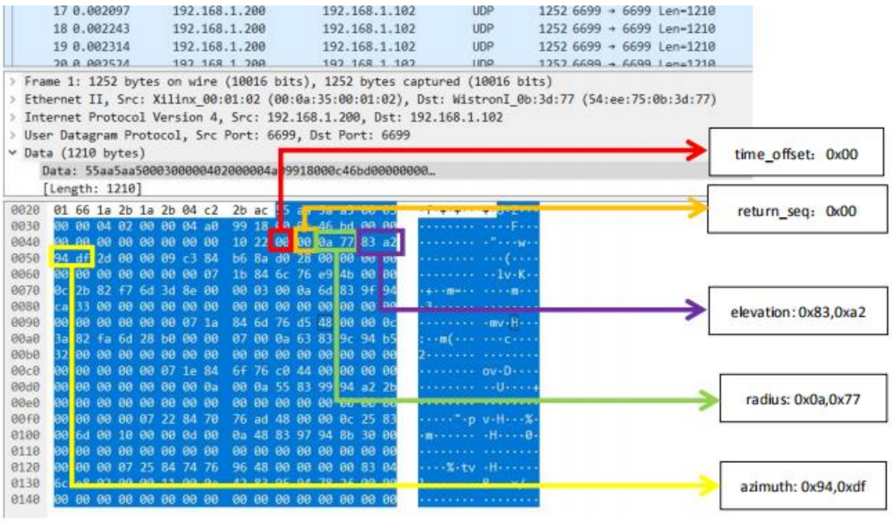
Therefore: the radial distance of this channel=R1*256+R2=3*256+252=1020.

According to the resolution of the coordinates, it is converted to meters: 1020 *0.005=5.10m.

Therefore, the radial distance of this channel in the corresponding elevation and azimuth direction is 5.1 m.

Calculation of XYZ coordinates:

Use Wireshark to capture the data packets of RS-Lidar-M1, as shown in the figure below:



Example of parameters calculation:

1. **time_offset:** data block time offset HEX: 0x00 -> DEC: 00 -> 0 μs

2. **return_seq:** HEX: 0x00 -> single return

3. **radius:** radial distance HEX: 0x0a,0x77 -> DEC: 10, 119

-> radius = (10 x256 + 119) x0.005 [m] = 13.395 m

4. **elevation:** vertical angle HEX: 0x83,0xa2 -> DEC: 131,162

-> elevation = ((131 x 256 + 162)-32768) x 0.01[degree] = 9.3°

•

5. **azimuth:** horizontal angle HEX: 0x94,0xdf -> DEC: 148,223

-> azimuth = ((148 x 256 + 223)-32768) x 0.01[degree] = 53.43°

The X, Y, Z coordinates of the point cloud can be calculated by the formula below:

$$\begin{cases} X = radius \bullet \cos(evelation) \bullet \cos(azimuth) \\ Y = radius \bullet \cos(evelation) \bullet \sin(azimuth) \\ Z = radius \bullet \sin(evelation) \end{cases}$$

$$\begin{cases} X = 13.395m \bullet \cos(9.3°) \bullet \cos(53.43°) \\ Y = 13.395m \bullet \cos(9.3°) \bullet \sin(53.43°) \\ Z = 13.395m \bullet \sin(9.3°) \end{cases}$$

Thus, the X, Y, Z coordinates of the point cloud of one transmitting in the single return mode of this channel is (7.88m,10.62m,2.17m)。

2.1.3 Tail

The Tail contains 3 bytes and are reserved bits.

2.2 LiDAR Information Output Protocol (DIFOP)

LiDAR Information Output Protocol is abbreviated as DIFOP

I/O type: LiDAR output, computer read.

Default port number: 7788.

DIFOP is an "output-only" protocol to periodically send the LiDAR serial number (S/N), firmware version information, host computer driver compatibility information, network configuration information, calibration information, operating status, and fault diagnosis information to users. By reading DIFOP, users can learn specific information of various parameters of the LiDAR currently in use.

A complete DIFOP packet consists of a synchronization header, reserved bytes and a data packet. Each DIFOP Packet is 256-byte long, including an 8-byte long synchronization header, 1 reserved byte and a 247-byte long data packet.

The basic structure of the DIFOP packet is as shown in the table below.

Table 7: Definition of DIFOP packet

Segments	Sequence No.	Attribute	Definition	Offset	Length（byte)
Header	1	Header	DIFOP identification header	0	8

	2	Reserved	Reserved bits	8	1
Data	3	Frame rate setting	Setting frame rate value，not enable yet	9	1
	4	Ethernet	Ethernet IP source address	10	4
			Ethernet IP destination address	14	4
			Ethernet IP local MAC address	18	6
			MSOP port number	24	2
			DIFOP port number	26	2
	5	FOV Setting (not enabled yet)	Horizontal FOV start angle	28	2
			Horizontal FOV end angle	30	2
			Vertical FOV start angle	32	2
			Vertical FOV end angle	34	2
	6	Version Information	Firmware version number of the motherboard programmable logic	36	5
			Firmware version number of the motherboard programming system	41	5
	7	Product SN information	Product serial number	46	6
	8	Wave_mode	Return mode setting	52	1
	9	Time information	Time synchronization mode setting	53	1
			Time synchronization status	54	1
			Time	55	10
	10	Operating status	Voltage, current, input and output signal status	65	31
	11	Diag_Inform_reserve	Diagnose Information reserved	96	29
	12	Reserved	Reserved	125	60
	13	Reserved	Reserved	185	71

Note:

•

1. The Header (DIFOP identification header) in the table is 0xa5, 0xff, 0x00, 0x5a, 0x11,0x11,0x55, 0x55, which can be used as the packet inspection sequence.

2. The LSB of the horizontal FOV is 0.01°the minimum value is 0°, and the maximum value is 120°.

3. The LSB of the vertical FOV is 0.01°, the minimum value is 0°, and the maximum value is 25 °.

4. Return mode setting: the return mode flag, 0-dual return, 1-N/A, 2-N/A, 3-N/A, 4- strongest return, 5-last return, 6-The first return.

5. Time synchronization mode setting: the default value is 0x02. 0x00 means currently using the LiDAR internal timing, 0x01 means that the 1PPS is currently used for sub-second reset in full seconds, 0x02 means currently using PTP time synchronization mode.

6. Time synchronization status: status of synchronization success. 0-unsuccess syn, 1-syn Success, 2-clock source disconnected

•

3 Interface Box Connection and State Machine

3.1 The Connection of Interface Box

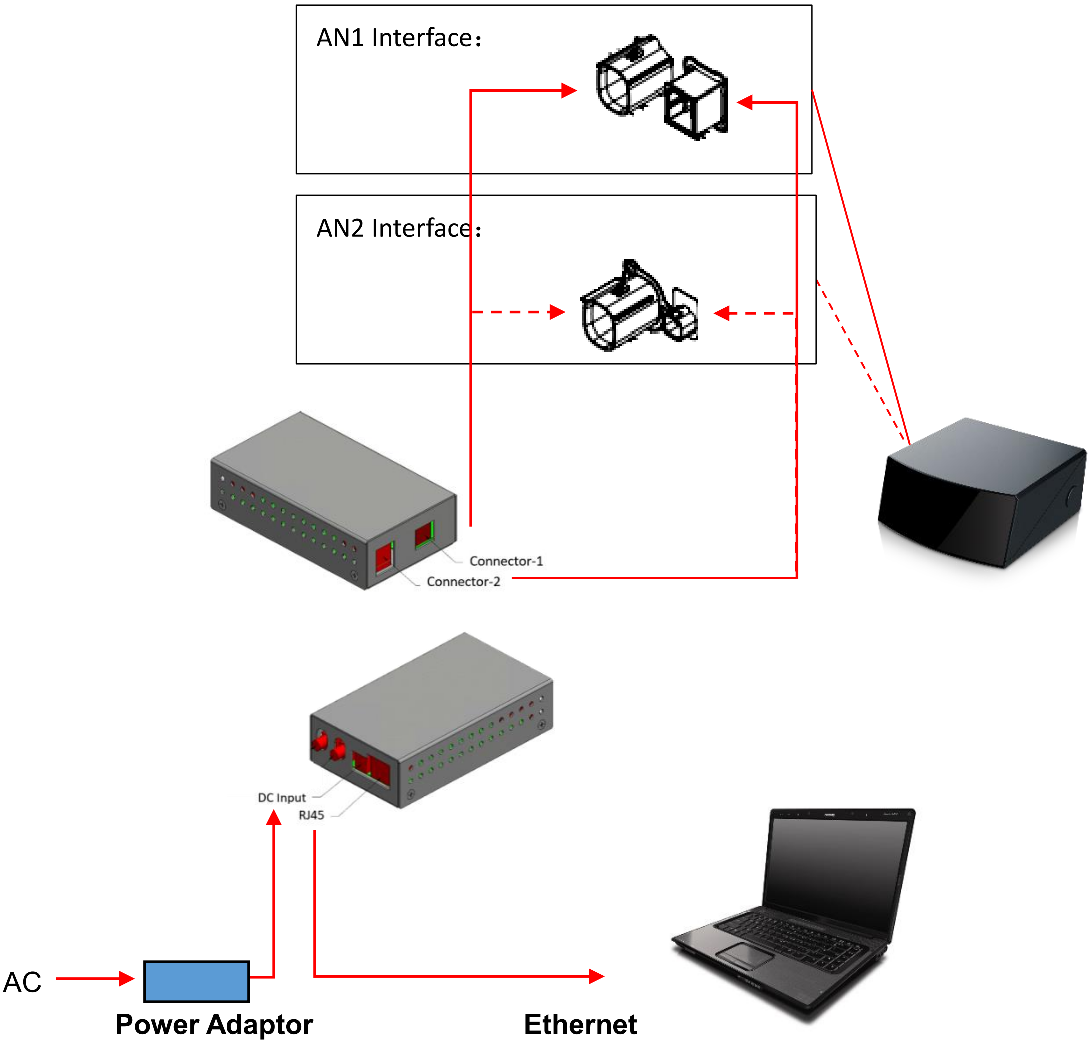


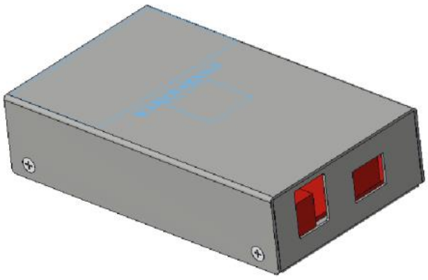
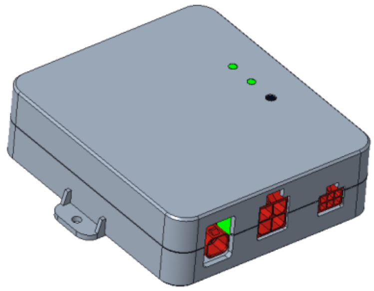
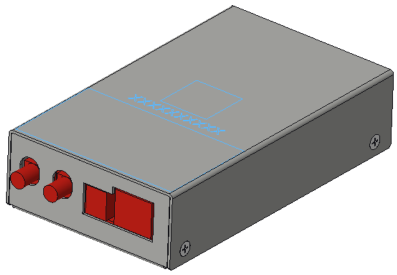
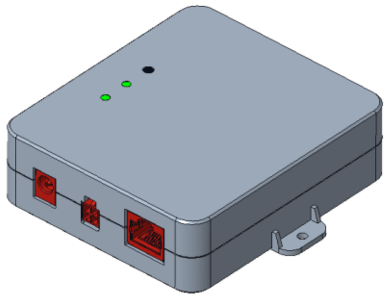
Figure 2: Image for topology of LiDAR and PC

Note：

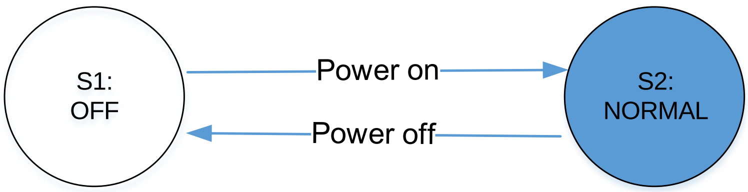
Figure 2 shows the topology of connection of Interface Box (AN1). Currently, RS-LiDAR-M1 has two versions of interfaces, namely AN1 and AN2, supporting the following two types of Interface Boxes respectively:

Connection	AN1 Interface Box	AN2 Interface Box

•

Connecting to LiDAR		
Connecting to Power Adaptor and Host Computer		

3.2 State Machine of LiDAR



Definition of State Machine：
S1：LiDAR OFF
S2：NORMAL operation

4 Time Synchronization

RS-LiDAR-M1 default firmware supports gPTP (IEEE 802.1AS) time synchronization method. Therefore, only gPTP is supported by default. If users want to apply PTP (IEEE 1588v2)time synchronization method, please contact RoboSense technical team for support.

4.1 Precision Time Protocol

4.1.1 Time Synchronization Introduction

PTP is defined as a time-synchronization protocol. It is mainly used to achieve high-precision time synchronization between different devices through network communication, and can also be used for frequency synchronization. Compared to the existing time synchronization mechanisms, PTP has the following advantages:

1) Compared to Network Time Protocol (NTP), PTP can fulfill the requirement of time synchronization with higher precision. Generally, NTP can only achieve the sub-second level of time synchronization precision, while PTP can support sub-microsecond level.

2) Compared to Global Positioning System (GPS), PTP has advantages of lower construction and maintenance costs. Meanwhile, it also has significant meanings in national security due to independence on GPS.

PTP supports different communication protocols (CAN, Ethernet, etc.). PTP can apply two mechanisms for synchronous: end-to-end (E2E) or peer-to-peer network (P2P):

E2E mode: apply Request Response Mechanism

P2P mode: apply a peer delay mechanism (Peer Delay Mechanism).

Note: The PTP protocol provided by RoboSense only supports the L2 layer of Ethernet protocol, E2E mode.

gPTP (general precise time protocol) is a derivative protocol of PTP in Time-Sensitive Networking. gPTP shares the same synchronization mechanism with PTP -- Peer Delay Mechanism, and it applies the L2 layer of Ethernet for communication. Unlike PTP, hardware timestamps are required for gPTP, which results in stricter requirements for the switch and Master clock.

•

4.1.2 gPTP Wiring Connection

To initialize gPTP synchronization procedure, users need to prepare the following devices and finish wiring connection according to the topology below:

1) a gPTP Grand Master (plug-and-play without additional configuration);

2) Ethernet switch;

3) Slave devices supporting gPTP (RS-LiDAR-M1 and others);

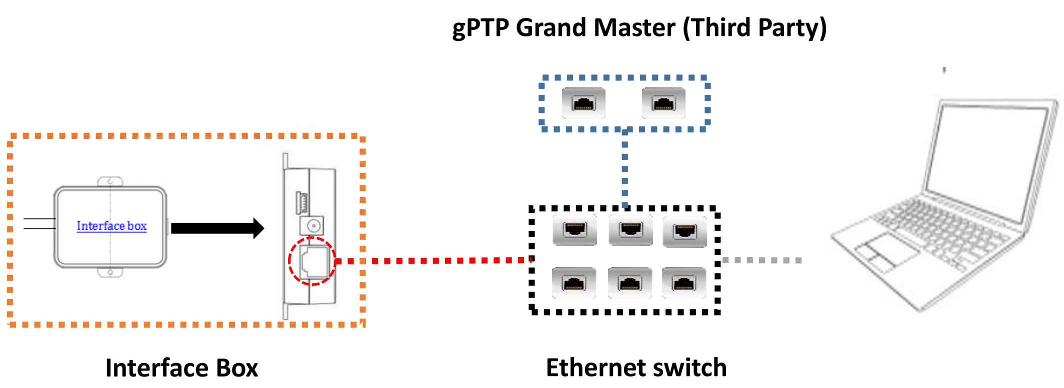


Figure 3：Topology of gPTP time synchronization

Note：

1. gPTP Grand Master device belongs to the third party, which is not included in our packing list. Users need to purchase that by themselves in advance;

2. As a Slave terminal, RS-LiDAR-M1 only obtains the time from gPTP Grand Master device with no hesitation about the accuracy of the master clock by principle. If the timestamp of LiDAR point cloud deviates from the real-time, please check whether gPTP Grand Master clock is accurate;

3. When time synchronization has been run on RS-LiDAR-M1, in case that the gPTP Grand Master is disconnected suddenly, the time stamp of LiDAR data packet will continue to stack according to LiDAR internal clock. The time of RS-LiDAR-M1 will not be reset until powered off and restarted.

4.2 Use Linuxptp tool to verify time synchronization

Please connect RS-LiDAR-M1 power cable and network cable to the Interface Box, and then to Host Computer. The Host Computer operating system (OS) must be Linux. We take Ubuntu as an example below:

1. Use the command $ifconfig to check the network card name. The name of the network

card is enp2s0 below.

```
sti@sti:~$ ifconfig
enp2s0    Link encap:Ethernet  HWaddr 54:ee:75:f0:7b:9f
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:1148564 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2786 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1436527228 (1.4 GB)  TX bytes:309309 (309.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:138110 errors:0 dropped:0 overruns:0 frame:0
          TX packets:138110 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:48448646 (48.4 MB)  TX bytes:48448646 (48.4 MB)
```

Figure 4: Find network card name

2. Use the command $ethtool - T enp2s0 (network card's name) to check whether the network card supports PTP hardware. For gPTP synchronization, hardware support is required, PTP Hardware Clock should be 1.

```
sti@sti:~$ ethtool -T enp2s0
Time stamping parameters for enp2s0:
Capabilities:
        software-transmit     (SOF_TIMESTAMPING_TX_SOFTWARE)
        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
```

Figure 5: Check PTP hardware supporting status

3. Download and install the Linuxptp tool.

$sudo git clone git://git.code.sf.net/p/linuxptp/code linuxptp

$cd linuxptp

$sudo make

$sudo make install

$reboot

4. Use ptp4l command.

Ptp4l command options:

Delay mechanism options

-A Automatic mode, E2E mode is selected automatically, and switch to P2P mode when peer to peer delay request is received

-E E2E mode, request-response delay mechanism (default)

•

-P P2P mode, peer delay mechanism

Network transmission options

-2 IEEE 802.3

-4 UDP IPV4 (default)

-6 UDP IPV6

Timestamp options

-H Hardware timestamp (default)

-S Software simulation timestamp

-L Former hardware timestamp, LEGACY HW needs to be used with PHC equipment

Other options

-f [file] Reads the configuration info from the specified file. By default, no configuration info is read.

-i [dev] Select a PTP interface device, such as eth0 (which can be specified more than once), you must specify at least one port using this option or configuration file.

-p [dev] This option is used to specify the PHC device (such as: dev / ptp0 clock device) to be used on the former Linux kernel. The default is auto, ignoring the software / LEGACY HW timestamp (this option is not recommended)

-s Slave-Only-mode, slave clock mode (override profile)

-t Transparent clock mode

-l [num] Set the logging level to 'num' and the default is 6

-m Print the message to stdout

-q Do not print messages to syslog

-v Print software version and exit

-h Help command

Use command to synchronize RS-LiDAR-M1:

(1) PTP E2E (L2 layer) command:

$sudo ptp4l -E -S -2 -m -i enp2s0 (network card name)

If PTP Hardware Clock is 1(hardware supported), you can use -H instead of -S

(2) gPTP command:

$sudo ptp4l -i enp4s0 -m -H -2 -f gptp-master.cfg

PTP Hardware Clock should be 1(hardware supported). Special note: Devices without hardware support can use -S instead of -H for gPTP synchronization simulation, and its synchronization accuracy cannot be guaranteed. And gptp-master.cfg is the gPTP master clock configuration file.

Create a new gptp-master.cfg file on the host, copy the following content in this file, and save the file:

```
# 802.1AS example configuration containing those attributes which
# differ from the defaults.    See the file, default.cfg, for the
# complete list of available options.
[global]
domainNumber            0
logSyncInterval         -3
syncReceiptTimeout      3
neighborPropDelayThresh     800
path_trace_enabled      1
follow_up_info          1
transportSpecific    0x1
ptp_dst_mac          01:80:C2:00:00:0E
#p2p_dst_mac         01:1B:19:00:00:00
network_transport L2
delay_mechanism         P2P
masterOnly           1
BMCA                 noop
asCapable                   true
inhibit_announce     1
inhibit_delay_req    1
```

4.3 GPS Time Synchronization

In case that users would like to synchronize RS-LiDAR-M1 with GPS module, it is necessary for gPTP Grand Master to receive GPS timing service at first. Please consult gPTP Grand Master device provider for the specific connectors and GPS timing service guidance. Robosense will not provide technical support except for special cases.

Figure 6: Topology of GPS Timing service synchronization

•

Appendix A RSView

This appendix explains how to use RSView to record, visualize, save and review of the data from RS-LiDAR-M1.

The original sensor data can be also captured and examined by using other free of charge tools, such as Wireshark or tcp-dump. But visualization of the 3D data through using RSView is easy to realize. User may contact RoboSense technical support for the specific RSview Version.

A.1 Software Features

RSView supports real-time visualization of 3D coordinate data from RS-LiDAR-M1. RSView also supports review of the pre-recorded data stored in “pcap” (Packet Capture) files, however, RSView doesn’t support direct importing of “.pcapng” files at the moment.

RSView displays directly the point cloud that is exchanged from the measured distance from RS-LiDAR-M1. It supports changing the display mode of point cloud according to XYZ coordinates, distance, pitch(elevation) and yaw(azimuth), etc.

Function and features of RSView are as shown below:

- Online visualization of sensor data over Ethernet
- Record of real-time data into pcap files
- Review of the recorded point cloud from pcap files
- Different visualization mode based on distance, pitch(elevation) and yaw(azimuth), etc.
- Tabular display of point cloud data
- Tool for measuring distance from visualized cloud point

A.2 Install RSView

Installation packet of RSView is suited for Windows 64-bit system and it requires no other dependent software packets. Unzip the compressed packet of RSView, the RSView.exe executable file can be found in the /bin folder.

A.3 Set Up Network

•

The sensor has set the default IP address to computer at factory, therefore, the default IP address of the computer should be set as 192.168.1.102, sub-net mask as 255.255.255.0. Besides, users should make sure that the RSView doesn't be blocked by any firewall or third party security software.

A.4 Visualization of Point Cloud

1. Connect the RS-LiDAR-M1 to PC over Ethernet cables and power supply.
2. Right click to start the RSView application with **Run as Administrator.**
3. Click on the File -> Open -> Sensor Stream (Fig A-1).

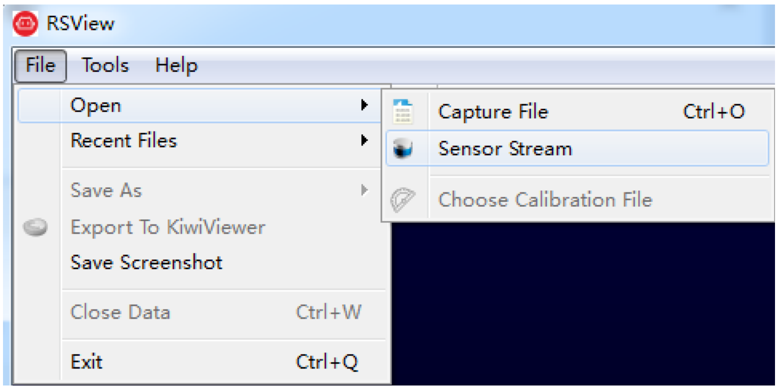


Fig A-1：Open the Sensor Stream in RSView.

4. After finishing the above 3 steps, the dialogue box "**Sensor Configuration**" shows up. In this dialogue box, the **Sensor Calibration** default contains the configuration folder named MEMSCorrectionFile_3V, select the corresponding file, click **Add** and then click the **OK** button (as shown in Fig A-2). The original point cloud data output from the RS-LiDAR-M1 is already calibrated point cloud data, therefore the value in this parameter file is void.

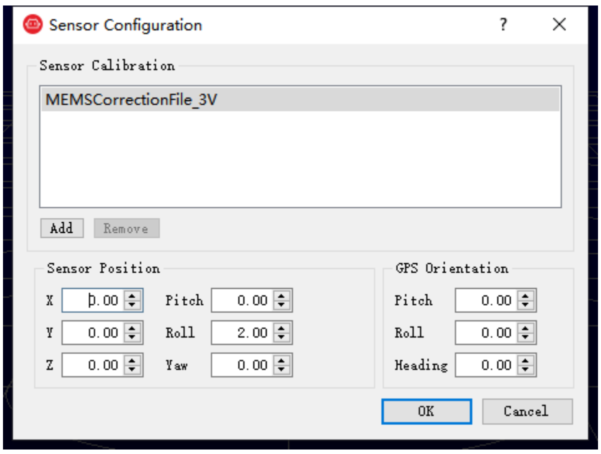


Figure A-2：Select the parameter configuration file of RS-LiDAR- M1

5. Check the MSOP and DIFOP port number: Tools > Sensor Network Configuration, choose 'use udp' and input the correct MSOP and DIFOP port number

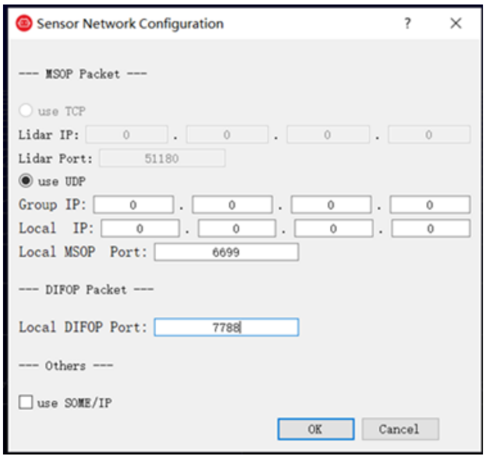


Figure A-3: RSView data port setting

6. RSView begins displaying the colored point cloud from capturing the sensor data stream from LiDAR (as shown in Fig. A-4). The stream can be paused by pressing the **Play** button, click again, the stream continues.

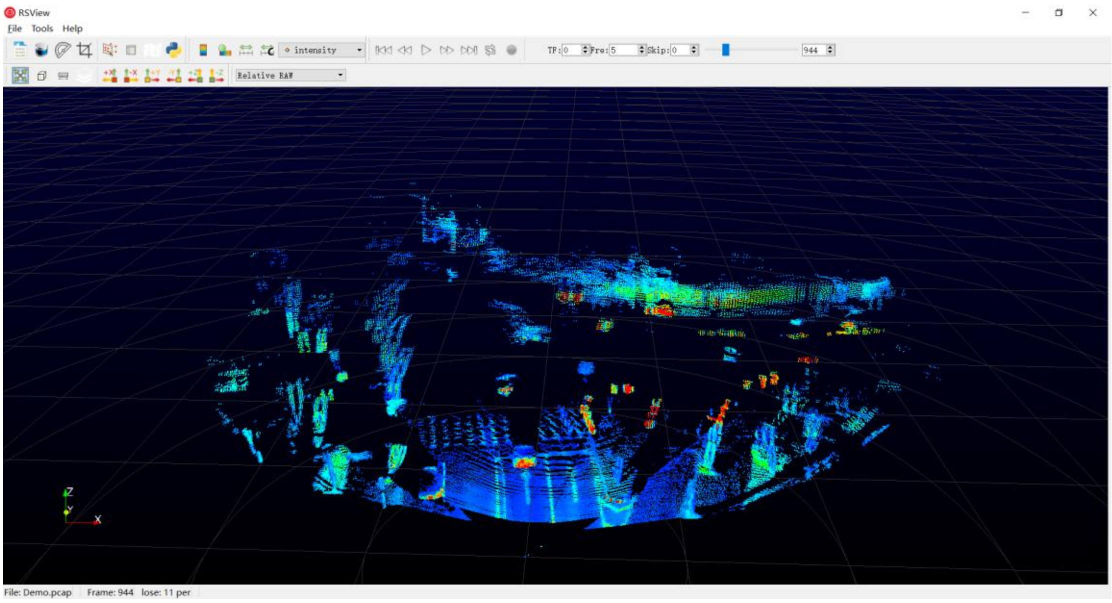


Figure A-4：RS-LiDAR-M1 Sensor Stream display

7. If there is no point cloud display, please click **Tools** and check if the MSOP and DIFOP port number are correctly set in the **Data Port Setting** window.

A.5 Save Streaming Sensor Data into PCAP File

Use RSView as the packet recording tool:

1. Click the **Record** button during real-time display (Fig. A-5).

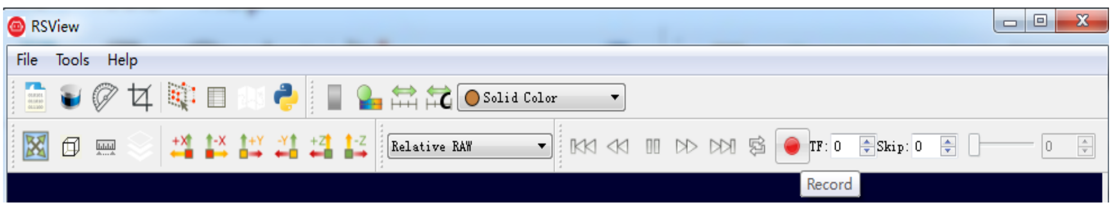


Figure A-5：RSView save button

2. In the dialogue box “**Choose Output File**”, choose the save path and file name of pcap file, click **Save** button (Fig. A-6), RSView begins writing data into pcap file. (Note: RS-LiDAR-M1 will generate enormous data, therefore, it is best to use a fast, local HDD or SSD, instead of a slow subsystem, such as USB storage LiDAR or network drive.)

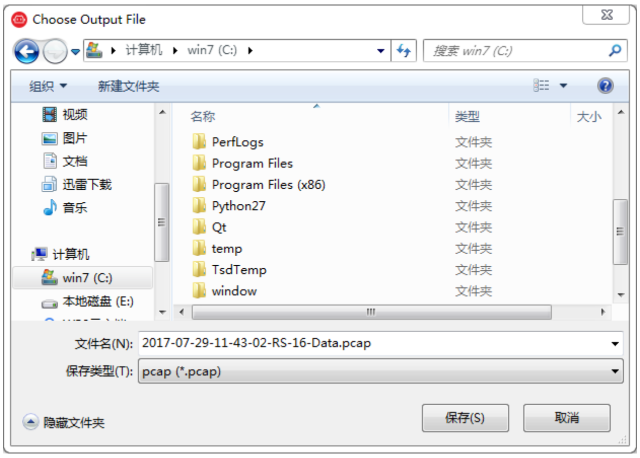


Figure A-6：RSView record data

3. Click **Record** button again to stop recording pcap packets.

Use Wireshark as the packet recording tool:

1. Download and install the wireshark software.

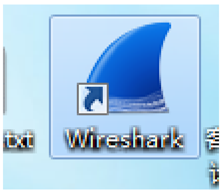


Figure A-7：Wireshark icon

2. Double click to start the wireshark application, select the name of the network card currently connected to the LiDAR and double-click it.

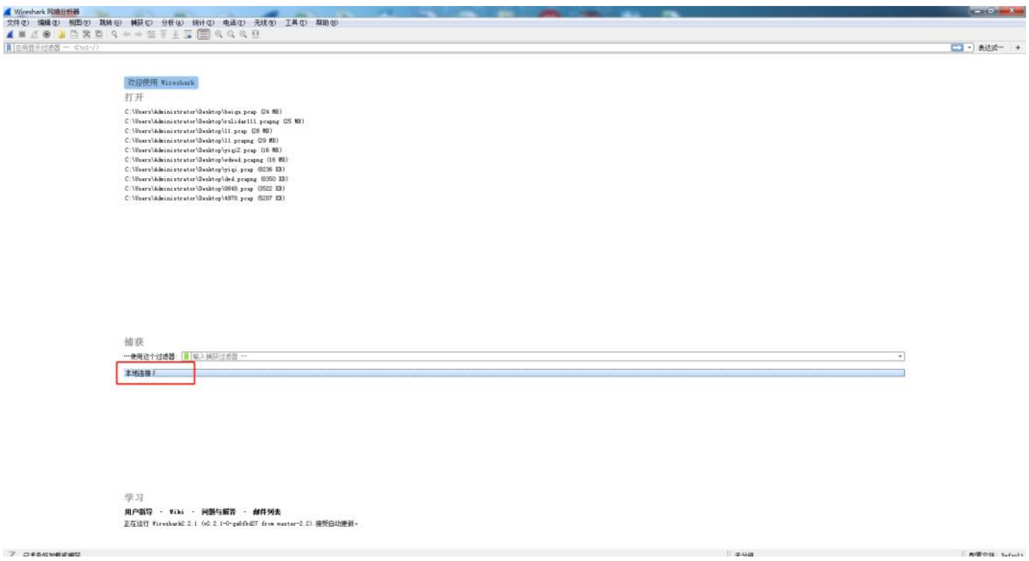


Figure A-8：Start Wireshark

3. If the figure below shows up, the connection to the LiDAR is normal. The data in the red boxes represent "LiDAR IP", "PC IP", "MSOP port number", and "DIFOP packet port number" respectively.

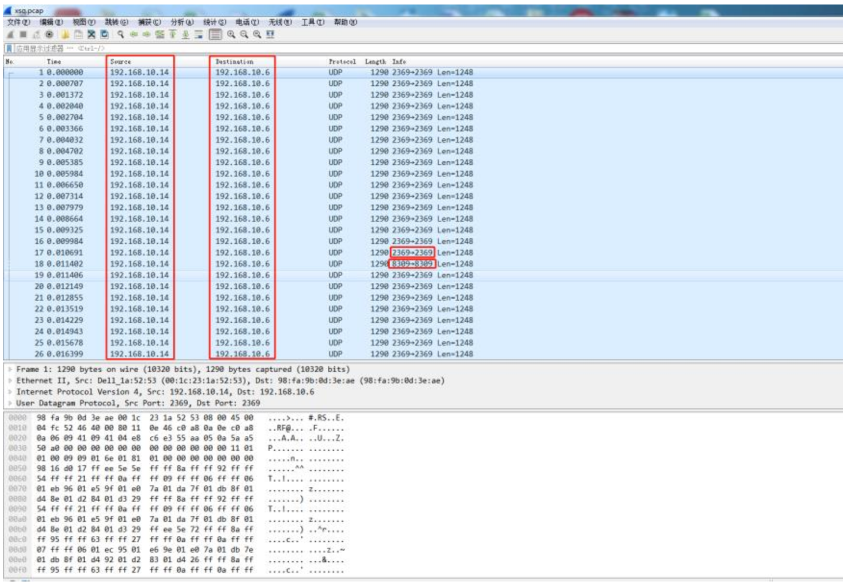


Figure A-9：Wireshark at work

4. Click **File** at the up left corner of the window, and click **Save** to save the data.

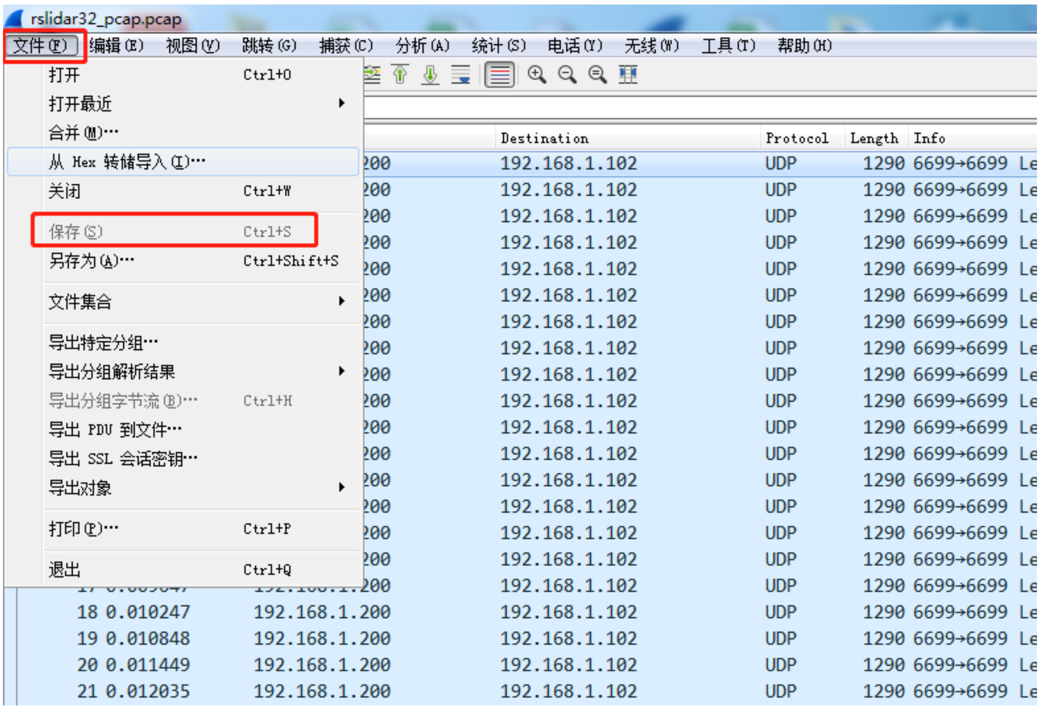


Figure A-10：Wireshark data saving

5. Enter the file name in the pop-up dialog box and select .pcap as the data format to save.

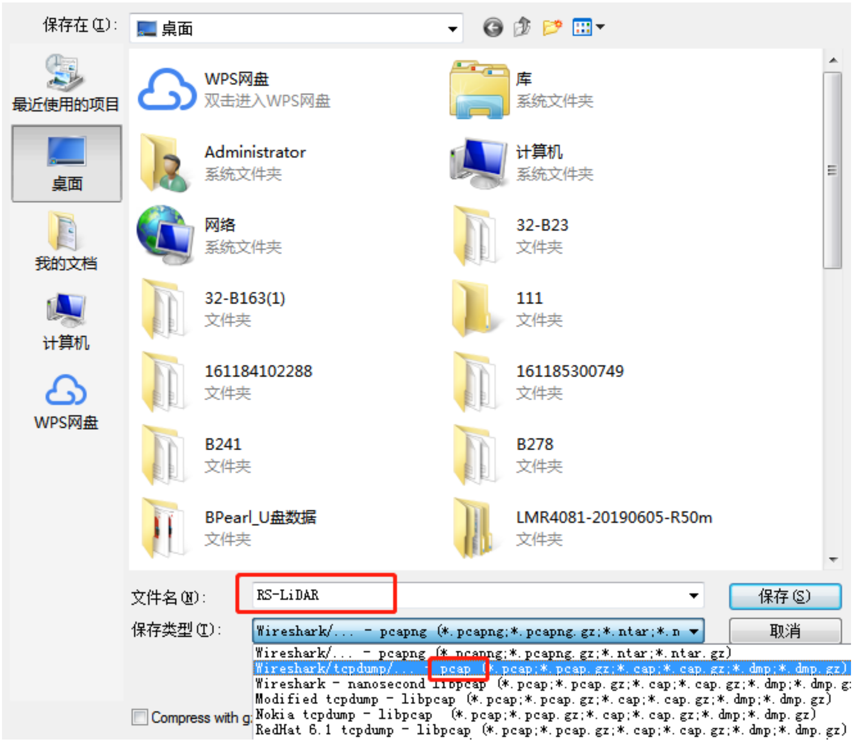


Figure A-11：Wireshark data saving

6. Now, the corresponding files can be found in the specified folder directory and you can use the RSView software or driver to view the point cloud (please refer to the product user manual for the RSView operation guide).

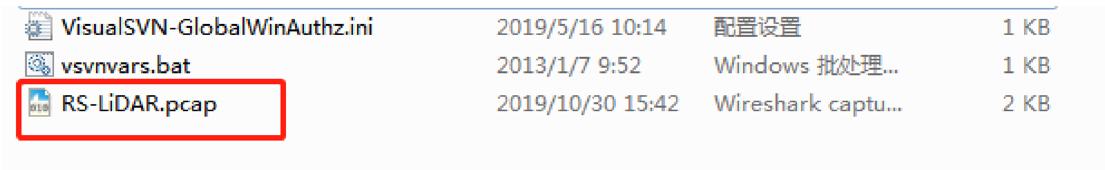


Figure A-12 Wireshark data saving

A.6 Replay Recorded Sensor Data from PCAP Files

The pcap file can be replayed or examined through RSView. User can press the **Play** button to play or pause the data play, and can also scrub the time slider to check the data at a certain time point. User can also use a mouse to click and select part of the point cloud and check them in the pop-up table. Save path of pcap file should not contain any Chinese characters.

1. Click **File -> Open** then select **Capture File.**

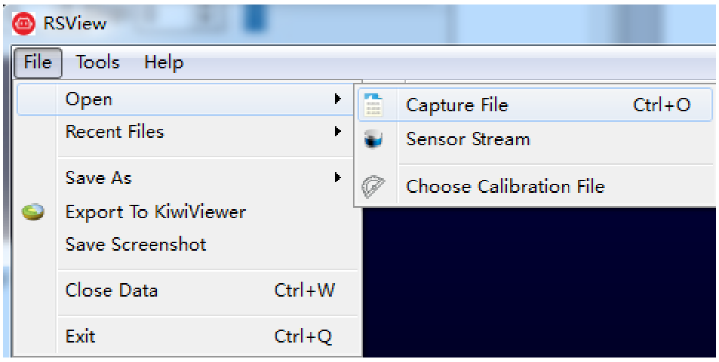


Figure A-13 RSView Open capture file

2. In the dialogue box **Open File,** please import a recorded pcap file then click **Open (O)** button.

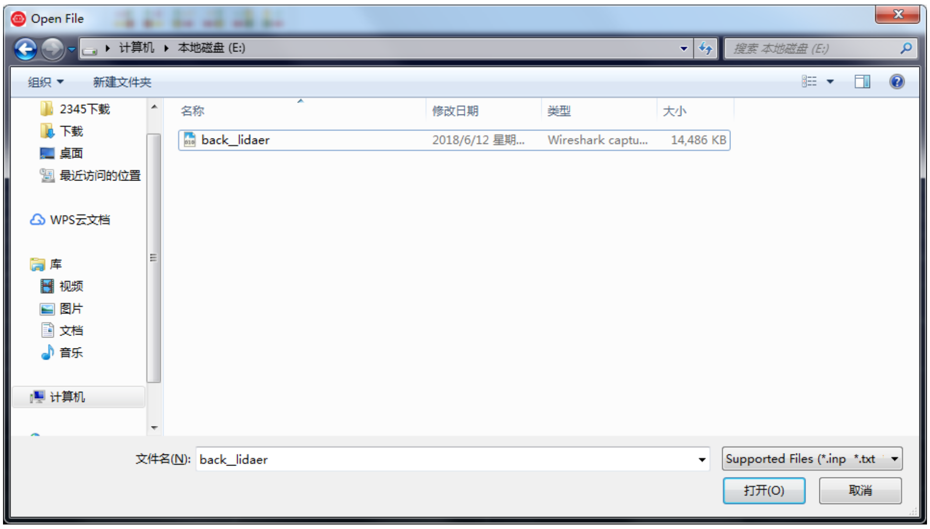


Figure A-14 Import PCAP File

3. In the dialogue box **Sensor Configuration**, add and select the right configuration file of RS-LiDAR-M1, then click **OK** button.

4. Click **Play** button to play or pause 3D point cloud data streaming. Using the Scrub tool to select the interested frame. The **Scrub** tool and the **Record** button are in the same toolbar(Fig. A-15).

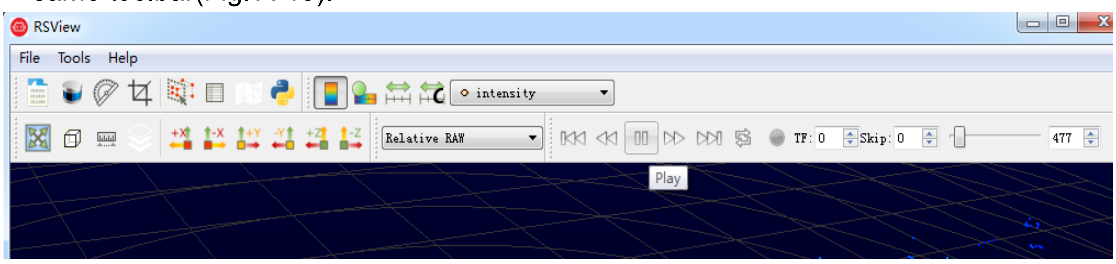


Figure A-15 RSView **Play** button and **Scrub** tool

5. In order to inspect partial relevant point cloud data from a closer aspect, please scrub to an interested frame and click the **Spreadsheet** button (Fig A-16). A data table will be displayed on the right side. It displays all data points in the frame.

•

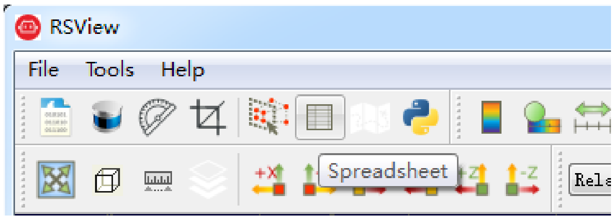


Figure A-16 RSView Spreadsheet

6. You can adjust the width of each column of the table, or sort for clearer inspection.

Showing: Data Attribute: Point Data Precision: 3

	Point ID	Point_X	Point_Y	Point_Z	distance_m	intensity	laser_id	pitch	yaw
0	0	7.947	10.617	1.888	13.395	45	0	9.300	53.430
1	1	10.905	5.676	2.231	12.495	40	1	12.060	27.680
2	2	0.000	0.000	0.000	0.000	1	2	-327.680	-327.680
3	3	8.250	-3.523	1.498	9.095	75	3	11.320	-23.270
4	4	10.390	-11.478	1.696	15.575	142	4	7.590	-48.030
5	5	7.956	10.549	1.873	13.345	51	0	9.270	53.220
6	6	0.000	0.000	0.000	0.000	1	1	-327.680	-327.680
7	7	0.000	0.000	0.000	0.000	1	2	-327.680	-327.680
8	8	8.233	-3.550	1.499	9.090	72	3	11.330	-23.470
9	9	10.397	-11.571	1.713	15.650	176	4	7.620	-48.240
10	10	7.965	10.481	1.858	13.295	50	0	9.240	53.010
11	11	0.000	0.000	0.000	0.000	1	1	-327.680	-327.680
12	12	0.000	0.000	0.000	0.000	1	2	-327.680	-327.680
13	13	8.237	-3.587	1.506	9.110	68	3	11.350	-23.680
14	14	0.000	0.000	0.000	0.000	1	4	-327.680	-327.680
15	15	7.958	10.401	1.840	13.225	43	0	9.210	52.820
16	16	0.000	0.000	0.000	0.000	1	1	-327.680	-327.680
17	17	0.000	0.000	0.000	0.000	1	2	-327.680	-327.680
18	18	8.243	-3.622	1.512	9.130	72	3	11.360	-23.870

Figure A-17 RSView Spreadsheet display

7. Click **Show only selected elements** in the spreadsheet, only the data of selected points will be displayed. If no point is selected, there will be no data shown in table (Fig. A-18).

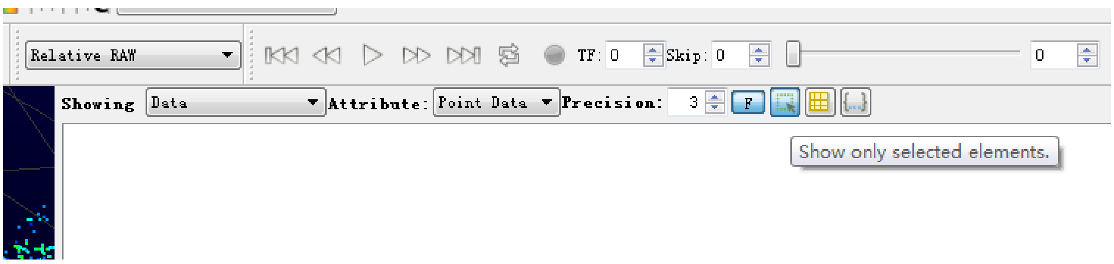


Figure A-18 RSView **show only selected elements** tool

8. Click the **Select All Points** tool, your mouse will turn into a data point selection tool (Figure A-19).

•

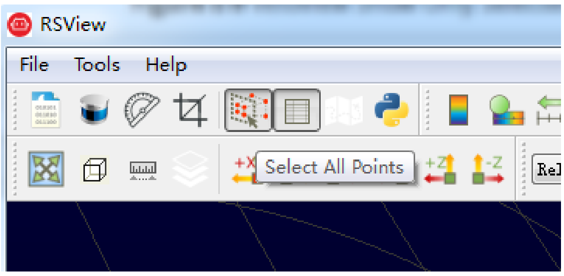


Figure A-19 RSView Select All Points tool

9. In the 3D point cloud display space, use the mouse to draw a rectangle to frame some data points. The data of these points will be displayed in the **Spreadsheet** and these points will turn pink in the point cloud display space (Figure A-20).

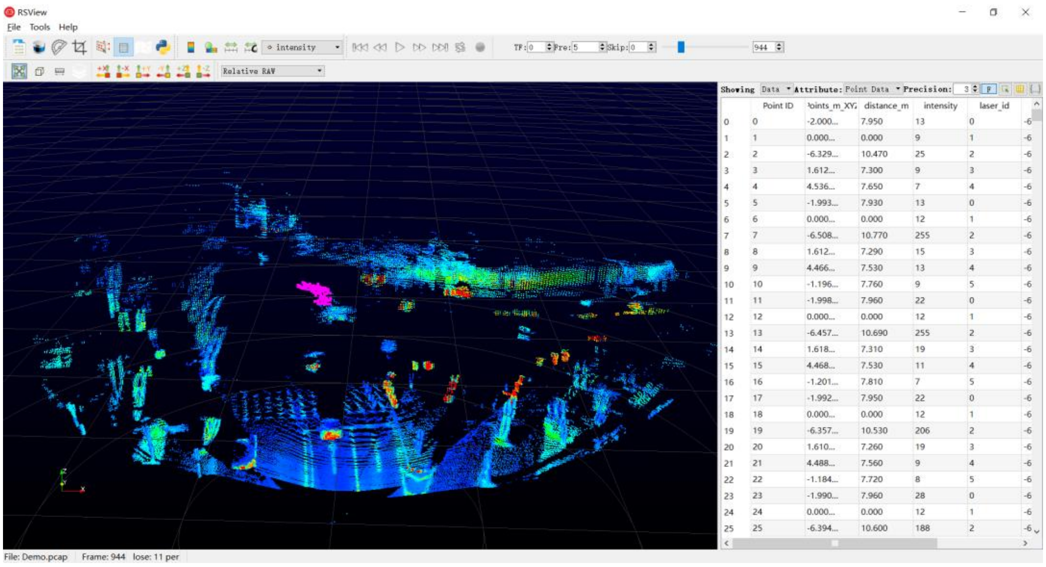


Figure A-20 RSView List Selected Points

10. Any selected point can be saved through the **output csv data** tool at the Spreadsheet toolbar (see Figure A-21).

Figure A-21 RSView export selected points to csv file

Appendix B Driver & SDK

B.1 Compile and Install rs_ driver

RS Driver provides a cross-platform LiDAR driver kernel for RoboSense LiDAR products, which is convenient for users to re-develop and use. The driver kernel of v1.3.0 and later versions already support analysis and transformation of RS-LiDAR-M1 point cloud. Users can download the rs_driver package from our official account on GitHub: https://github.com/RoboSense-LiDAR/rs_driver

rs_driver currently supports the following systems and compilers:

- Windows
  - MSVC (VS2017 & VS2019 tested)
  - Mingw-w64 (x86_64-8.1.0-posix-seh-rt_v6-rev0 tested)
- Ubuntu (16.04, 18.04, 20.04)
  - gcc (4.8+)

B.1.1 Install Dependent Libraries

rs_driver depends on the following third-party libraries, which need to be installed before compilation:

- Boost
- Pcap
- PCL (not required, can be ignored if visualization tools are not needed)
- Eigen3 (not required, can be ignored if built-in coordinate transformation is not needed)

Install the above dependent libraries in Ubuntu:

$sudo apt-get install libboost-dev libpcap-dev libpcl-dev libeigen3-dev

Install the above dependent libraries in Windows:

- Boost

The Boost library needs to be compiled from source code under Windows, please refer to the official guide (https://www.boost.org/doc/libs/1_67_0/more/getting_started/windows.html) After compiling and installing, add the path of Boost to the system environment variable BOOST_ROOT, see Figure B-1 below. If you use MSVC, you can also choose to directly download the pre-compiled installation package of the corresponding version.

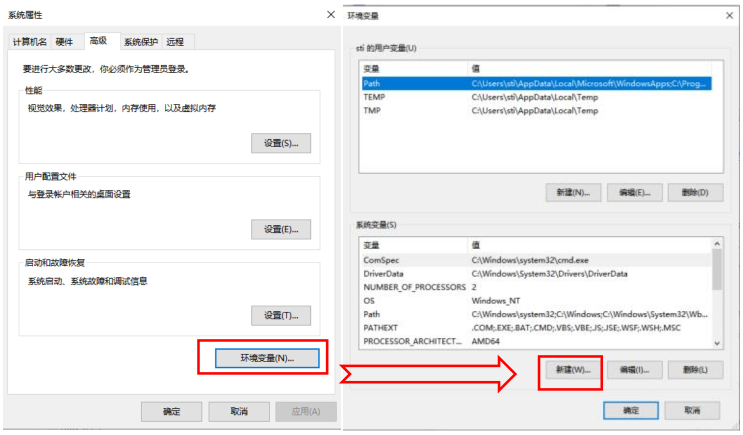


Figure B – 1: Add environment variables

➢ Pcap

First, install the Pcap runtime library (https://www.winpcap.org/install/bin/WinPcap_4_1_3.exe).

Download the developer package (https://www.winpcap.org/install/bin/WpdPack_4_1_2.zip) to any location,

Then, add the path of WpdPack_4_1_2/WpdPack to the environment variable PATH, as shown in Figure B-1.

➢ PCL (not required, can be ignored if visualization tools are not needed)

(1) MSVC

If you are going to use the MSVC compiler, please install the official installation package provided by PCL.

Select "Add PCL to the system PATH for xxx" during installation:

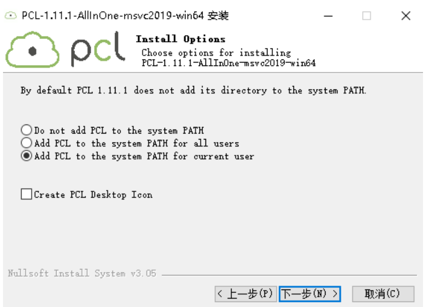


Figure B - 1: Set up PCL

(2) Mingw-w64

PCL does not provide the official mingw compilation library, users need to compile PCL from source code and install it according to the official tutorial.

B.1.2 Use of rs_Driver

B.1.2.1 rs_Driver Installation and Use

Take the Linux environment as an example for driver compilation (rs_driver currently does not support installation and use in windows system), execute the following codes to install the driver

•

$cd rs_driver
$mkdir build && cd build
$cmake .. && make -j4
$sudo make install

B.1.2.2 Use as a Submodule

When rs_driver is used as a submodule, the following commands need to be added to the CMakeLists.txt file. (add rs_driver as a submodule to the project, use the find_package() instruction to find rs_driver, and then link the relevant library)

```
add_subdirectory(${PROJECT_SOURCE_DIR}/rs_driver)
find_package(rs_driver REQUIRED)
include_directories(${rs_driver_INCLUDE_DIRS})
target_link_libraries(project ${rs_driver_LIBRARIES})
```

B.1.3 Demo Programs & Visualization Tools

B.1.3.1 Demo Programs

rs_driver provides two demo programs. Users can refer to the demo programs to write code, call interfaces and store them in rs_driver/demo:

demo_online.cpp
demo_pcap.cpp

To compile the two demo programs, users can add the parameters when executing the CMake configuration.

$cmake -DCOMPILE_DEMOS=ON ..

B.1.3.2 Visualization Tools

rs_driver provides a point cloud visualization tool based on PCL, which is stored in rs_driver/tool:

rs_driver_viewer.cpp

To compile the two demo programs, users can add the parameters when executing the CMake configuration.

$cmake -DCOMPILE_TOOLS=ON ..

B.1.4 Coordinate Transformation

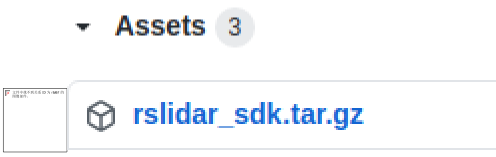
rs_driver provides a built-in coordinate transformation feature, which can directly output the point cloud after coordinate transformation, which saves users the time-consuming extra operations of coordinate transformation on the point cloud. If you want to enable this feature, add the parameters when executing CMake configuration:

$cmake -DENABLE_TRANSFORM=ON ..

B.2 Compile and Install rslidar_sdk

rslidar_sdk is the LiDAR driver software package of RoboSense in the Ubuntu environment, including the LiDAR driver kernel, ROS expansion function, ROS2 expansion function, and Protobuf-UDP communication expansion features. Users without secondary development needs, or users who want to directly use ROS or ROS2 for secondary development, can directly use this software package to view the point cloud with the RVIZ visualization tool that comes with ROS or ROS2. For users who have further secondary development needs and want to integrate the LiDAR driver into their own projects, please refer to the relevant documentation of the LiDAR driver kernel and directly use the kernel rs_driver for secondary development.

You can download the rsliar_sdk.tar.gz package from our official account on GitHub: https://github.com/RoboSense-LiDAR/rslidar_sdk/releases



Note: Downloading the source code will not include the rs_driver parsing kernel, users need to download and add it manually.

B.2.1 Install Dependent Libraries

B.2.1.1 ROS Environment

To use the LiDAR driver in ROS environment, users need to install ROS related dependent libraries

Ubuntu 16.04 - ROS kinetic desktop-full

Ubuntu 18.04 - ROS melodic desktop-full

Installation method: refer to http://wiki.ros.org

If ROS kinetic desktop-full or ROS melodic desktop-full is installed, other dependent libraries of compatible versions should also be installed at the same time, so there is no need to reinstall them to avoid problems caused by multiple version conflicts. Therefore, it is strongly recommended to install the desktop-full version, which will save a lot of time to install and configure the libraries one by one.

B.2.1.2 ROS2 Environment

To use the LiDAR driver in ROS2 environment, users need to install ROS2 related dependent libraries

Ubuntu 16.04 - not supported

Ubuntu 18.04 - ROS2 Eloquent desktop

Installation method: refer to https://index.ros.org/doc/ros2/Installation/Eloquent/Linux-Install-Debians/

Note: Please avoid installing ROS and ROS2 on the same computer at the same time, this may cause conflicts! You also need to install the Yaml library manually.

B.2.2 Compile and Run rslidar_sdk

rslidar_sdk can be compiled and run in three different ways.

B.2.1.1 Direct Compilation

Follow the instructions below, users can directly compile and run the program. Direct

•

compilation can access some ROS features (excluding ROS2), but it requires users to manually start roscore before the program starts, after the roscore starts, users need to manually open rviz to view the visualized point cloud results.

The compilation commands are as follows:

$cd rslidar_sdk
$mkdir build && cd build
$cmake .. && make -j4
$./rslidar_sdk_node

B.2.1.2 Compilation Dependent on ROS-catkin

1. Open the CMakeLists.txt file in the project and change the set(COMPILE_METHOD ORIGINAL) at the top of the file to set(COMPILE_METHOD CATKIN).

```
#=======================================
# Compile setup (ORIGINAL,CATKIN,COLCON)
#=======================================
set(COMPILE_METHOD CATKIN)
```

2. Rename the package_ros1.xml file in the rslidar_sdk project directory to package.xml.
3. Create a new folder as the workspace, then create a new folder named src, and put the rslidar_sdk project into the src folder.
4. Return to the workspace directory, execute the following command to compile and run. (if you use .zsh, replace the second command with source devel/setup.zsh)

$catkin_make
$source devel/setup.bash
$roslaunch rslidar_sdk start.launch

B.2.1.3 Compilation Dependent on ROS2-colcon

1. Open the CMakeLists.txt file in the project and change the set(COMPILE_METHOD ORIGINAL)at the top of the file to set(COMPILE_METHOD COLCON)。

```
#=======================================
# Compile setup (ORIGINAL,CATKIN,COLCON)
#=======================================
set(COMPILE_METHOD COLCON)
```

2. Rename the package_ros2.xml file in the rslidar_sdk project directory to package.xml.
3. Create a new folder as the workspace, then create a new folder named src, and put the rslidar_sdk project into the src folder.
4. Download the LiDAR packet message definition in the ROS2 environment through the link, and put the rslidar_msg project in the newly created src folder alongside the rslidar_sdk.
5. Return to the workspace directory and execute the following command to compile and run. (if you use .zsh, replace the second command with source install/setup.zsh)

$colcon build
$source install/setup.bash
$ros2 launch rslidar_sdk start.py

•

B.2.3 Parameters

This project has only one parameter file config.yaml, which is stored in the rslidar_sdk/config folder. The entire parameter file can be divided into two parts, a common part and a LiDAR part. In the case of multiple LiDARs, the common part parameters are applicable to all LiDAR sensors, where the LiDAR part parameters need to be set separately according to the actual status of each LiDAR.

Note: The parameter file config.yaml has strict requirements for indentation! Please ensure that the indentation at the beginning of each line remains consistent after modifying the parameters!

B.2.3.1 Common Part Parameters

This part of parameters is used to set the message source of the LiDAR and decides whether to publish the results.

```
common:
  msg_source: 1                          # LiDAR message source type
  send_packet_ros: false
  send_point_cloud_ros: false
  send_packet_proto: false
  send_point_cloud_proto: false
  pcap_path: /home/robosense/lidar.pca     #Absolute address when playing offline
PCAP packets
```

msg_source:

1 -- Connect to LiDAR online. For more details, please refer to **Reading LiDAR data online and sending to ROS.**

2 -- Parse ROS or ROS2 packets offline. For more details, please refer to **Recording ROS data packets & parsing ROS data packets offline**.

3 -- Parse the pcap packet offline. For more details, please refer to **Parsing Pcap packets offline and sending to ROS.**

4 -- The LiDAR message source is the packet message of Protobuf-UDP

5 -- The LiDAR message source is the point cloud message of Protobuf-UDP

send_packet_ros：

true-- LiDAR packet messages will be sent through ROS or ROS2, false-- forbidden.

Since the LiDAR ROS packet message is a custom ROS message of RoboSense, users cannot directly echo the topic to view the specific content of the message. In fact, the packet is mainly used to record offline ROS packets, because the volume of the packet is smaller than the point cloud.

send_point_cloud_ros：

true – LiDAR point cloud message will be sent through ROS or ROS2, false—forbidden.

The point cloud message type is officially defined by ROS as sensor_msgs/PointCloud2, so users can directly use Rviz to view the point cloud. At the same time, users can also choose to record the point cloud directly when recording the packet, but the volume of the packet will be very large, so we recommend recording the packet message when recording the ROS packet offline.

send_packet_proto：

true – LiDAR packet message will be sent through Protobuf-UDP, false – forbidden.

send_point_cloud_proto：

true – LiDAR point cloud message will be sent through Protobuf-UDP, false – forbidden.

We recommend sending packet messages instead of point cloud messages, because point cloud messages are too large and has higher requirements on bandwidth.

pcap_path：

If msg_dource = 3, please ensure that this parameter is set to the correct absolute path of the pcap package.

B.2.3.2 LiDAR Part Parameters

This part of parameters needs to be set for each LiDAR according to their specific status.

```
lidar:
  - driver:
      lidar_type: RSM1
      frame_id: /rslidar
      msop_port: 6699
      difop_port: 7788
      start_angle: 0
      end_angle: 360
      min_distance: 0.2
      max_distance: 200
      use_lidar_clock: false
    ros:
      ros_recv_packet_topic: /rslidar_packets
      ros_send_packet_topic: /rslidar_packets
      ros_send_point_cloud_topic: /rslidar_points
    proto:
      point_cloud_recv_port: 60021
      point_cloud_send_port: 60021
      msop_recv_port: 60022
      msop_send_port: 60022
      difop_recv_port: 60023
      difop_send_port: 60023
      point_cloud_send_ip: 127.0.0.1
      packet_send_ip: 127.0.0.1
```

lidar_type: the currently supported LiDAR types are listed in the sdk file in the README folder. **RS-LiDAR-M1** belongs to type **RSM1**。

frame_id: the frame_id of point cloud messages.

msop_port, difop_port: The msop port number and difop port number of the point cloud. If cannot receive messages, please first check whether these two parameters are configured correctly.

•

start_angle, end_angle: This parameter is temporarily disabled for RS-LiDAR-M1. The start angle and end angle of the point cloud message are set here as software shielding, and the volume of the point cloud per frame cannot be reduced. Only the points outside the area are set as NAN points. The range of the starting angle and ending angle should be between 0 and 360°. (The starting angle can be greater than the ending angle).

min_distance, max_distance: The minimum distance and maximum distance of the point cloud display are set here as software shielding. The volume of the point cloud per frame cannot be reduced, and only the points outside the area are set as NAN points.

use_lidar_clock: true - use LiDAR time as message timestamp; false - use system time as message timestamp.

B.2.3.3 Example of Multiple LiDAR Sensors

Connect 2 RS-LiDAR-M1 LiDAR sensors online and send the point cloud to ROS.

Attention: Indentation of LiDAR part parameters

```
common:
  msg_source: 1                          #Use online data messages
  send_packet_ros: false
  send_point_cloud_ros: true             # Send point cloud rslidar_points data
  send_packet_proto: false
  send_point_cloud_proto: false
  pcap_path: /home/robosense/lidar.pcap
lidar:
  - driver:
      lidar_type: RSM1
      frame_id: /rslidar
      msop_port: 6699
      difop_port: 7788
      start_angle: 0
      end_angle: 360
      min_distance: 0.2
      max_distance: 200
      use_lidar_clock: false
    ros:
      ros_recv_packet_topic: /middle/rslidar_packets
      ros_send_packet_topic: /middle/rslidar_packets
      ros_send_point_cloud_topic: /middle/rslidar_points
    proto:
      point_cloud_recv_port: 60021
      point_cloud_send_port: 60021
      msop_recv_port: 60022
      msop_send_port: 60022
      difop_recv_port: 60023
      difop_send_port: 60023
      point_cloud_send_ip: 127.0.0.1
```

```
      packet_send_ip: 127.0.0.1
  - driver:
      lidar_type: RSBP
      frame_id: /rslidar
      msop_port: 1990
      difop_port: 1991
      start_angle: 0
      end_angle: 360
      min_distance: 0.2
      max_distance: 200
      use_lidar_clock: false
    ros:
      ros_recv_packet_topic: /left/rslidar_packets
      ros_send_packet_topic: /left/rslidar_packets
      ros_send_point_cloud_topic: /left/rslidar_points
    proto:
      point_cloud_recv_port: 60024
      point_cloud_send_port: 60024
      msop_recv_port: 60025
      msop_send_port: 60025
      difop_recv_port: 60026
      difop_send_port: 60026
      point_cloud_send_ip: 127.0.0.1
      packet_send_ip: 127.0.0.1
```

B.2.4 Coordinate Transformation

rslidar_sdk provides a built-in coordinate transformation feature, which can directly output the point cloud after coordinate transformation, which significantly saves the user's time-consuming operation of coordinate transformation on the point cloud. This section will guide users how to use the built-in coordinate transformation feature of rslidar_sdk to output the point cloud after coordinate transformation.

B.2.4.1 Dependencies

To enable the coordinate transformation feature, the following dependencies need to be installed:

- Eigen3

Command installation method:

$sudo apt-get install libeigen3-dev

B.2.4.2 Compilation

To enable the coordinate transformation function, the ENABLE_TRANSFORM option needs to be set as **ON** when compiling the program.

•

1. Direct compilation

$cmake -DENABLE_TRANSFORM=ON

$make -j4

2.ROS compilation

$catkin_make -DENABLE_TRANSFORM=ON

2. ROS2 compilation

$colcon build --cmake-args '-DENABLE_TRANSFORM=ON'

B.2.4.3 Set the Coordinate Transformation Parameters

The coordinate transformation parameters are the LiDAR part hidden parameters, including x, y, z, roll, pitch, and yaw. Here is an example of the parameter file, which can be configured by the user according to the actual situation.

```
common:
  msg_source: 1
  send_packet_ros: false
  send_point_cloud_ros: true
  send_packet_proto: false
  send_point_cloud_proto: false
  pcap_path: /home/robosense/lidar.pcap
lidar:
  - driver:
      lidar_type: RS128
      frame_id: /rslidar
      msop_port: 6699
      difop_port: 7788
      start_angle: 0
      end_angle: 360
      min_distance: 0.2
      max_distance: 200
      use_lidar_clock: false
      x: 1
      y: 0
      z: 2.5
      roll: 0.1
      pitch: 0.2
      yaw: 1.57
```

•

Appendix C Autosar Tool

C.1 Tool Introduction

The little-robo tool is a customized Autosar tool developed by RoboSense for the Automotive-Grade LiDAR RS-LiDAR-M1. This tool is only for M1 0210 platform products based on the AutoSar architecture. Developed with UDS diagnostic DoIP protocol, this tool can be used to obtain basic information of LiDAR, such as source IP, target IP, MSOP Port, DIFOP Port, motherboard PS/PL firmware version, product serial number, etc. This tool also supports ROI (Gaze function) on/off switch control, DID information query, firmware upgrade and other functions.

C.2 Operation System Requirements

The little-robo tool supports the environment of windows/ubuntu16.04/ubuntu18.04/ ubuntu20.04. The corresponding version of the tool can be obtained by contacting RS technical support.

C.3 Operation Instructions

C.3.1 Function Description

The little-robo tool supports two modes:

① State without connection: Unlink mode

② State with connection: Link mode (also called Linked mode)

Two modes can be switched by clicking the [Link] & [Unlink] Button.

Different buttons will light up in different modes, Usually, set buttons are colored and clickable, and buttons without setting functions are green and non-clickable.

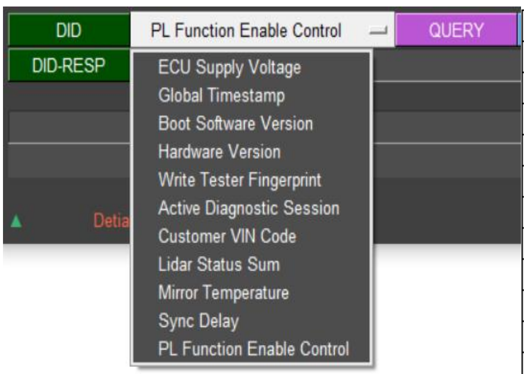
The Link mode supports the following functions:

1. Modify the internal network configuration parameters of the connected ECU (specifically LiDAR)

① Support to modify the IP address and port inside ECU;

② Support to modify the subnet mask and routing inside ECU.

2. Support DID query; click the [QUERY] button to search the DID query options, and the result will be shown in DID-RESP



DID Name	Remark
ECU Supply Voltage	系统供电电压
Global Timestamp	全局时间戳
Boot Software Version	底层软件版本
Hardware Version	硬件版本
Write Tester Fingerprint	写入测试者指纹
Active Diagnostic Session	主动诊断会话
Customer VIN Code	车架号
Lidar Status Sum	雷达状态描述
Mirror Temperature	振镜温度
Sync Delay	同步延迟
PL Function Enable Control	ROI使能控制

Figure C - 1: DID query options & description

•

3. Support reading some default ECU parameters, such as SN, PS, and PL version
4. Support bin and hex firmware format flashing
5. Support Reset function (LiDAR soft-reboot function)

C.3.2 Interface Description

A license is required for the first-time running the software. As shown in Figure 2. Please use the following license serial number:

robosense.autosar.mems.team

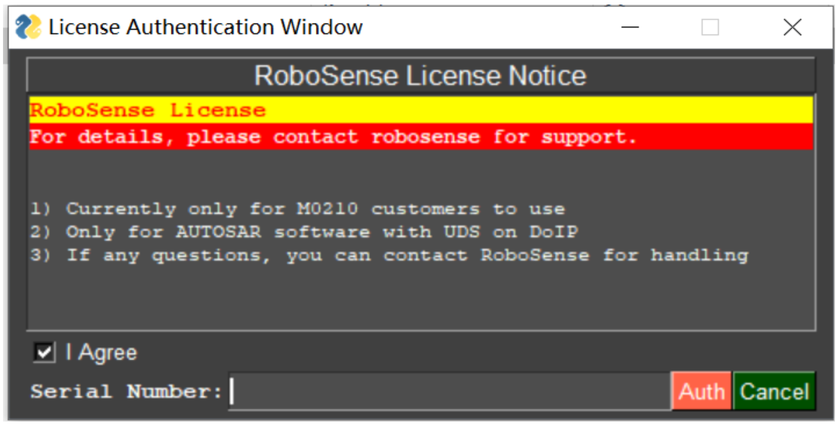


Figure C - 2: RoboSense License Notice

After entering the License Serial Number, check [I Agree], and click [Auth] to enter the tool operation interface, as shown in Figure 3.

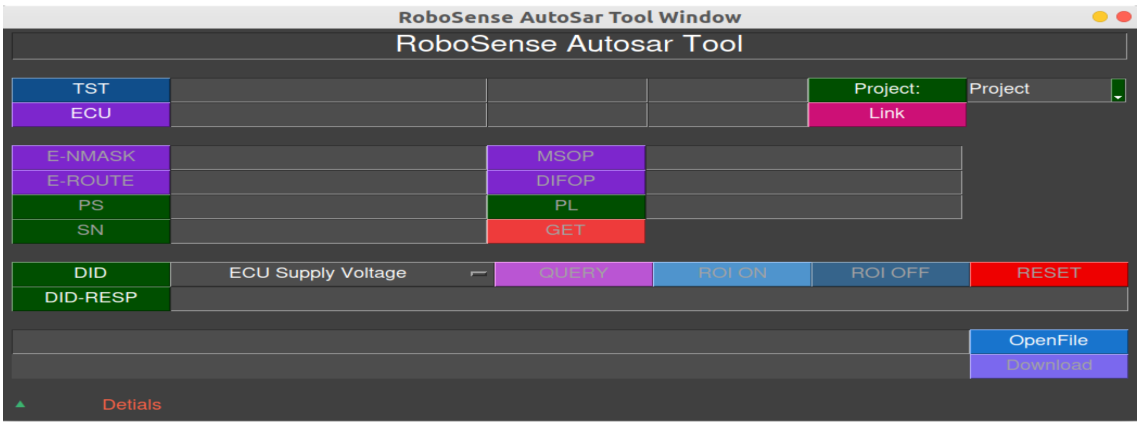


Figure C - 3: Operation initial interface (Unlink mode)

•

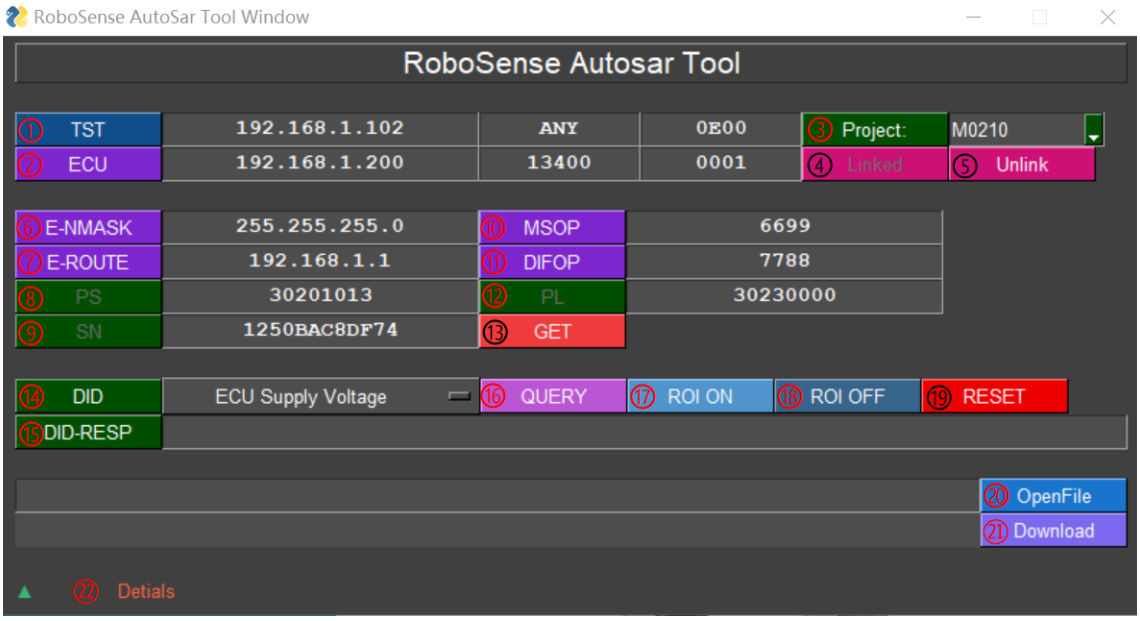


Figure C - 4: LiDAR basic information interface (Linked mode)

①LiDAR target IP is 192.168.1.102 by default. It can be modified. After entering the desired IP address in the box, click **[TST]** to confirm, and then restart the LiDAR to complete the modification;

②LiDAR source IP is 192.168.1.200 by default. It can be modified. After entering the desired IP address in the box, click **[ECU]** to confirm, and then restart the LiDAR to complete the modification;

③Project option, select M0210 from the drop-down menu;

④Click to enter Link mode;

⑤Click to enter Unlink mode;

⑥LiDAR subnet mask is 255.255.255.0 by default. It can be modified. After entering the desired IP address in the box, click **[E-NMASK]** to confirm, and then restart the LiDAR to complete the modification;

⑦LiDAR routing address is 192.168.1.1 by default. It can be modified. After entering the desired IP address in the box, click **[E-ROUTE]** to confirm, and then restart the LiDAR to complete the modification;

⑧Get the PL firmware version of the LiDAR motherboard;

⑨Get the product serial number of LiDAR;

⑩LiDAR MSOP port number is 6699 by default. It can be modified. After entering the desired port number in the box, click **[MSOP]** to confirm, and then restart the LiDAR to complete the modification;

⑪LiDAR DIFOP port number is 7788 by default. It can be modified. After entering the desired port number in the box, click **[DIFOP]** to confirm, and then restart the LiDAR to complete the modification;

⑫Get the PS firmware version of the LiDAR motherboard;

⑬Click **[GET]** to get the current LiDAR's information, including subnet mask, routing address, MSOP Port, DIFOP Port, PS & PL firmware version and product serial number;

⑭DID query, drop down to select query items, please refer to chapter 3.1 Function Description for more details;

⑮DID-RESP, return DID value;

⑯QUERY, confirm the query;

•

⑰ROI ON, click to turn on the ROI function;
⑱ROI OFF, click to turn off the ROI function;
⑲RESET, click to soft restart the LiDAR;
⑳OpenFile, open the firmware path;

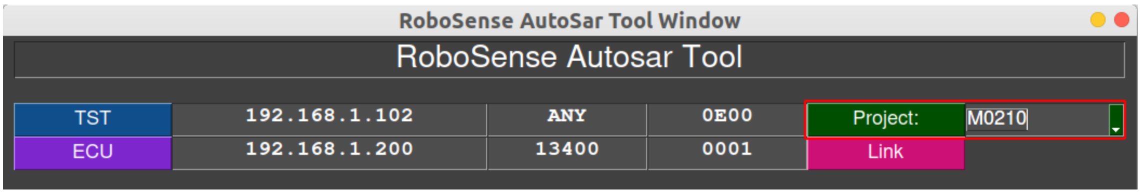
㉑Download, start the download program;

㉒Detail is the output terminal of operation information results.

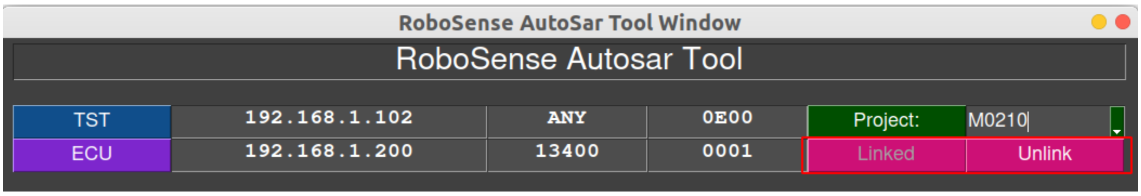
C.3.3 Tool Usage

C.3.3.1 Get LiDAR Information

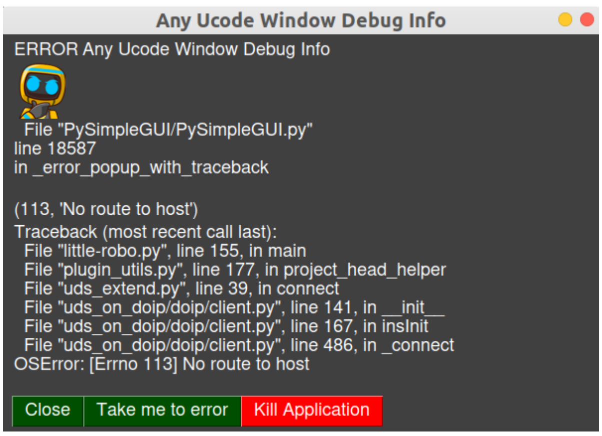
Step1. Open the tool. Double-click the program to enter the basic interface, click the drop-down option of the Project button at the top right, and select M0210, the interface is as follows:



Step2. Enter Link mode. Please confirm the source IP and target IP of the LiDAR before clicking **[Link]**. If the LiDAR is in the default state, click **[Link]** to enter Link mode. If [Unlink] appears on the right of [Link], it has entered Link mode. The interface is as follows:



If the source IP and target IP of the LiDAR is not in the default state, they need to be modified before entering the Link mode, otherwise, errors will occur. The error interface is as follows:

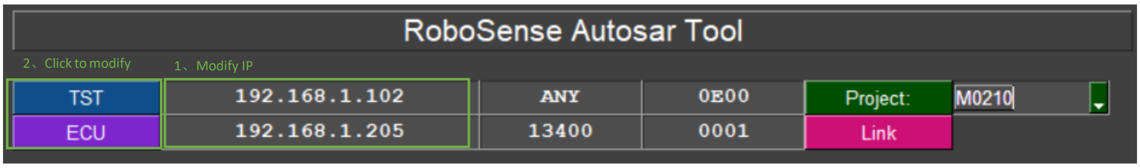


To modify the IP, enter the desired IP in the edit box and click the corresponding option on the

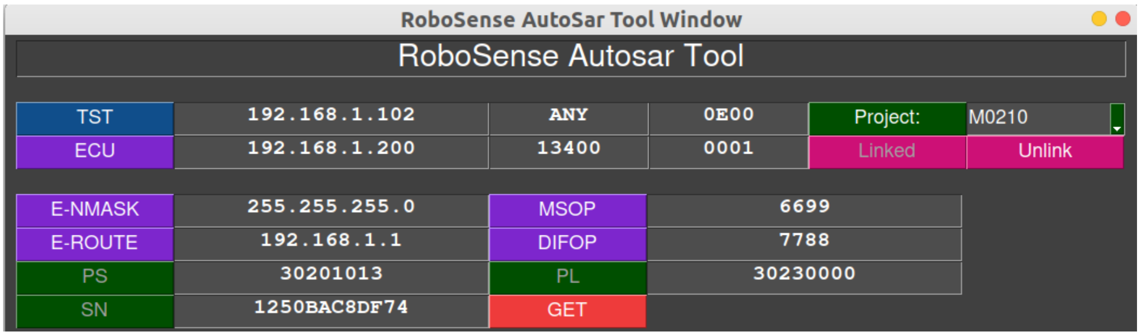
•

left to confirm;

Example: If the LiDAR IP is 192.168.1.205, please first change the ECU IP address to 192.168.1.205, and click **[ECU]** to confirm, and then click **[Link]**. if Unlink pops up, it has entered Link mode.



Step3. Get LiDAR information. After entering Link mode, click **[GET]** to get basic LiDAR information.



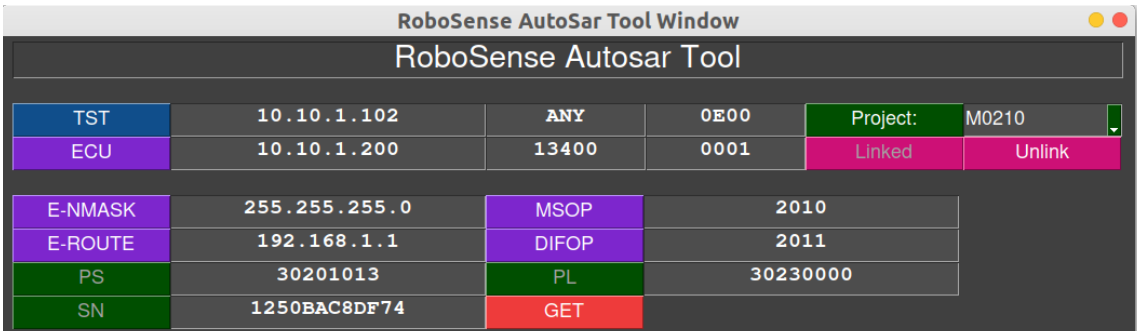
C.3.3.2 Set Up LiDAR IP and Port

Please perform this operation in Link mode, refer to [3.3.1 Get LIDAR Information] for more details.

Operation example: to modify the default LIDAR information to the following state:

LiDAR Info	Description
LiDAR IP（ECU）	10.10.1.200
PC IP(TST)	10.10.1.102
MSOP Port	2010
DIFOP Port	2011

Step1. After entering the Link mode, edit the IP and Port to be set in the corresponding boxes of **[ECU] [TST] [MSOP] [DIFOP]**, and click **[ECU] [TST] [MSOP] [DIFOP]** button after editing to confirm;



Step2. Click **[RESET]** to soft restart the LiDAR; (Whenever the LiDAR information is modified, the LiDAR should be restarted. You can click RESET to soft restart or re-power the LiDAR again!)

Step3. After restarting, re-enter the link mode to confirm, or use Wireshark packet-capturing tool to confirm.

No.	Time	Source	Destination	Protocol	Length	Info
58	0.110185	10.10.1.200	10.10.1.102		298	49153 → 2011 Len=256
59	0.113216	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210
60	0.113217	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210
61	0.113217	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210
62	0.113217	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210
63	0.114214	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210
64	0.114214	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210
65	0.114245	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210
66	0.114245	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210
67	0.114245	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210
68	0.114245	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210
69	0.114245	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210
70	0.114438	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210
71	0.114438	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210
72	0.114634	10.10.1.200	10.10.1.102		1252	49152 → 2010 Len=1210

C.3.3.3 ROI Function On/Off Switch

1. ROI on/off function:

Click **[ROI ON]** button to turn on ROI; click **[ROI OFF]** button to turn off ROI.

2. ROI function status query

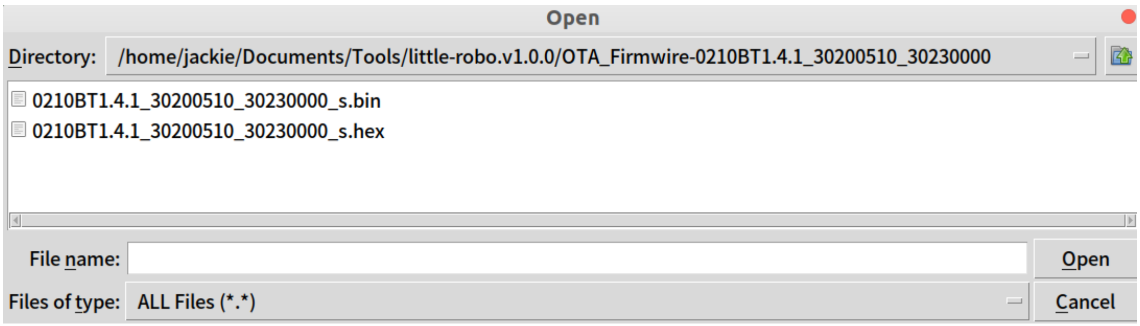
The ROI status can be queried via DID Query;

Select **[PL Function Enable Control]** and click **[QUERY]** button;

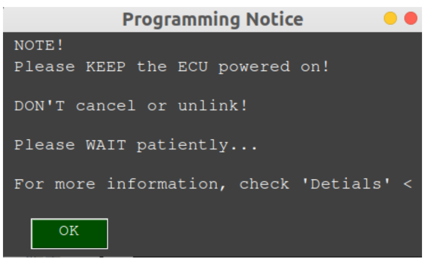
DID-RSEP value of 00 indicates ROI Function off, DID-RSEP value of 01 indicates ROI Function on.

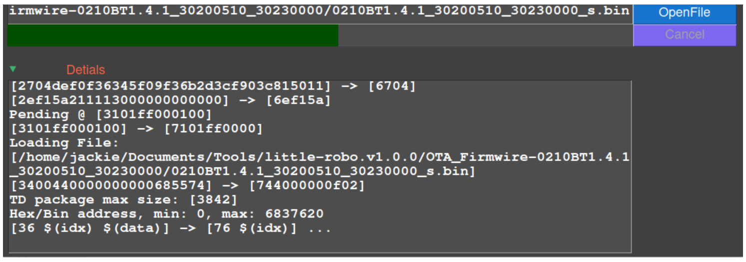
C.3.3.4 Firmware Flashing

Step1. Click **[OpenFile]**, select the firmware to be flashed (bin and hex files are supported)

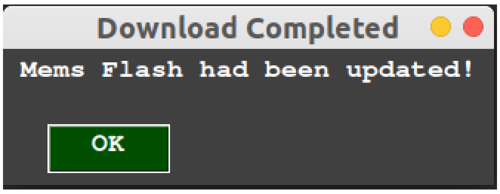


Step2. Click **[Download]** to pop up the Programming Notice window, click **[OK]** to start the upgrade. You can view the progress bar and Details information when waiting for the upgrade.

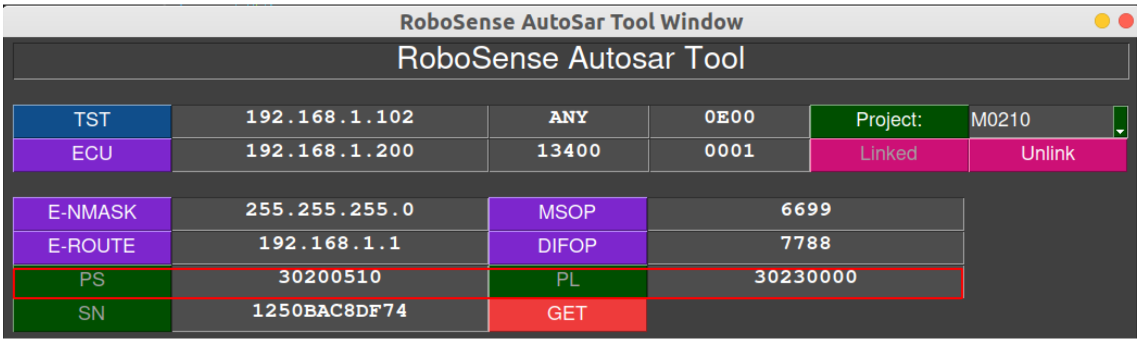




Step3. After the upgrade is done, the upgrade completion window pops up, click OK to complete the upgrade;



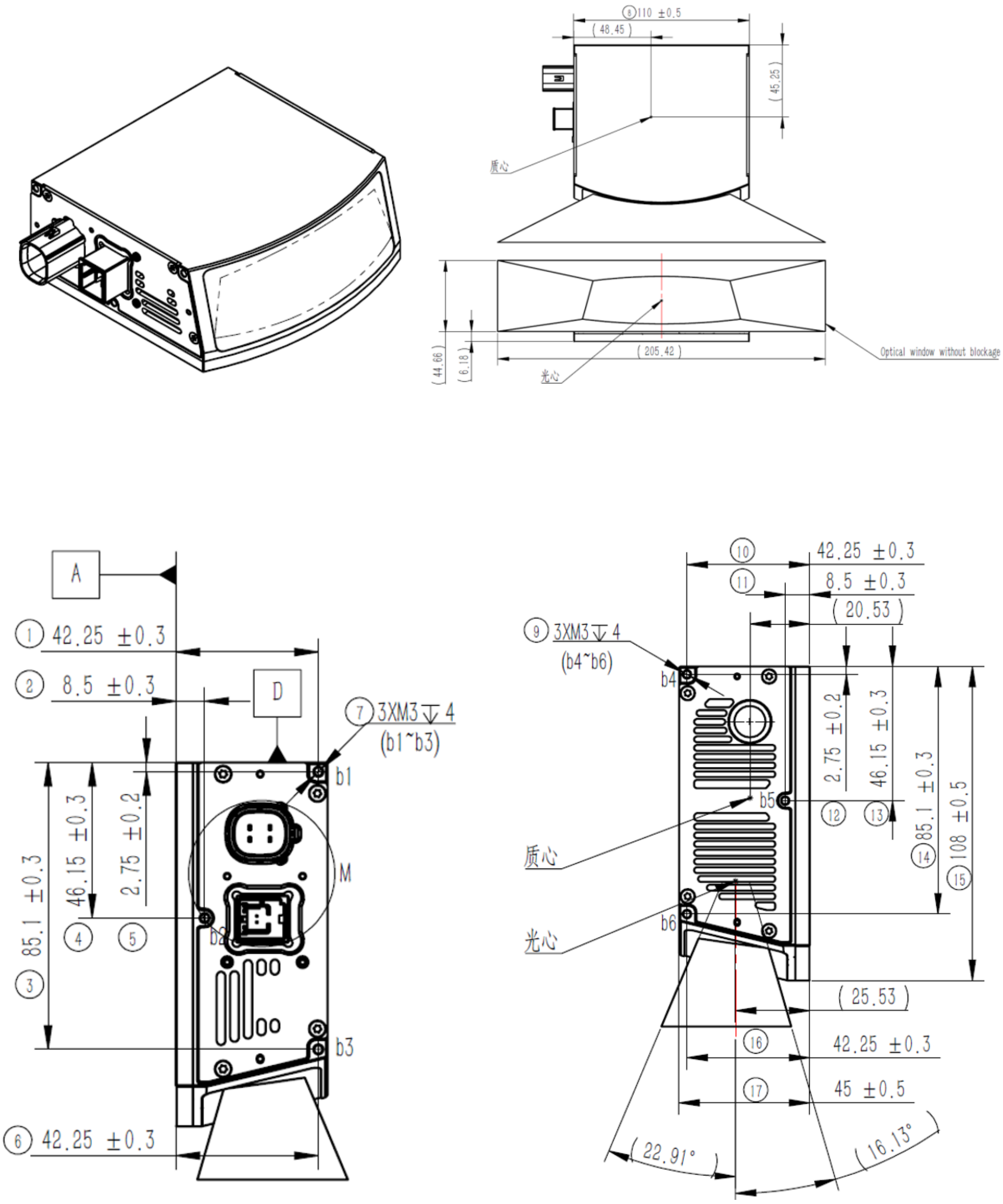
Step4. After the upgrade is done, LiDAR will start automatically. Re-enter the Link mode after booting and get the version information for confirmation.

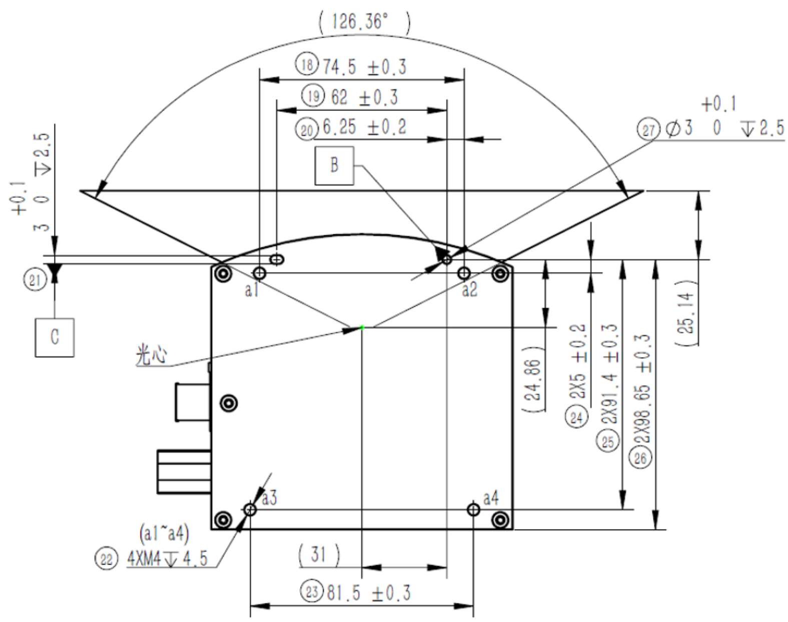


•

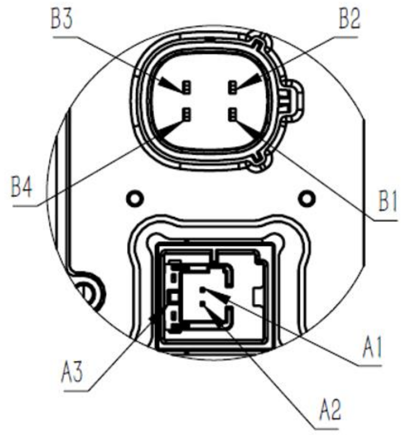
Appendix D Dimension

Drawing of LiDAR with interface AN1：



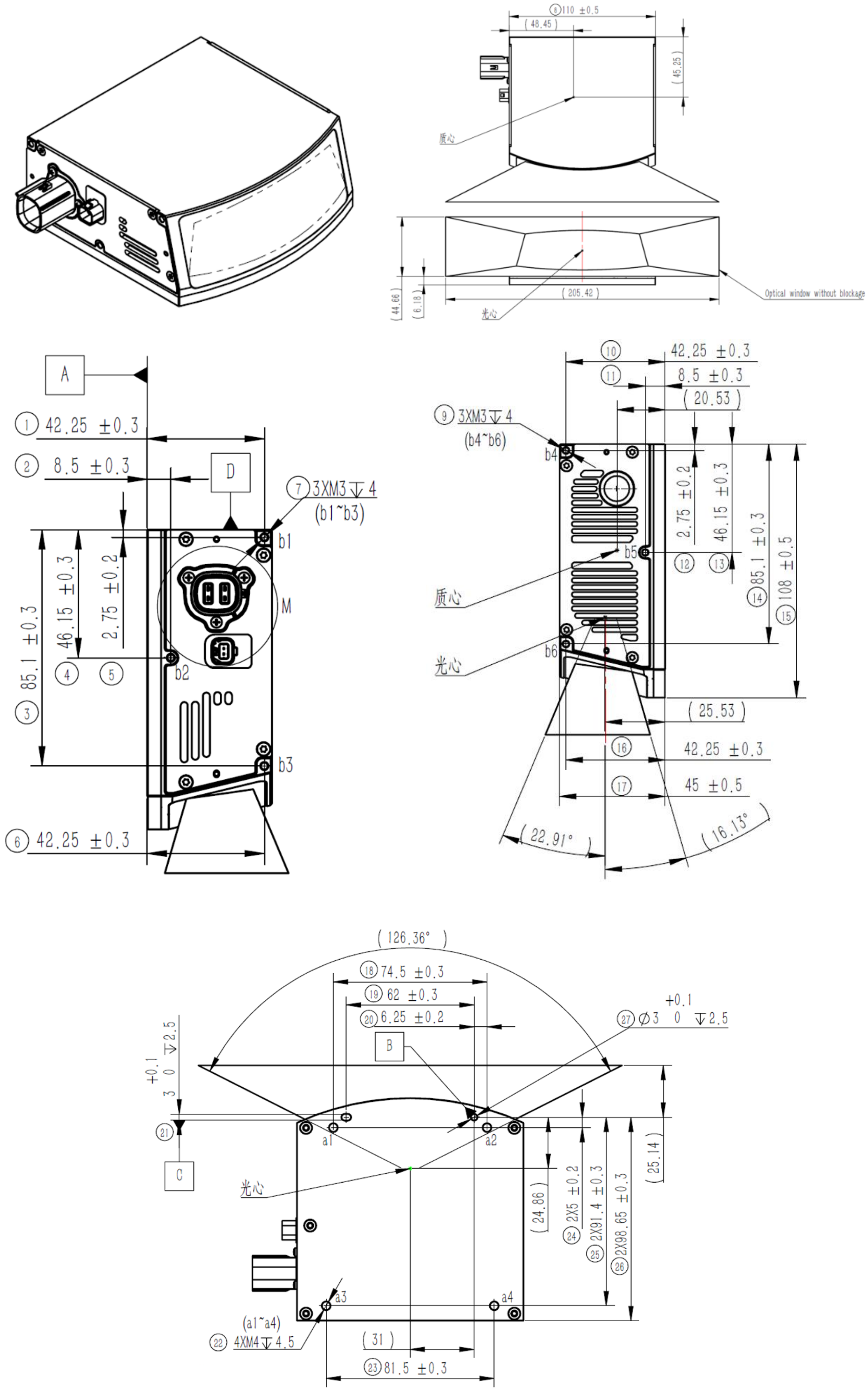


Definition of AN1-Pins:



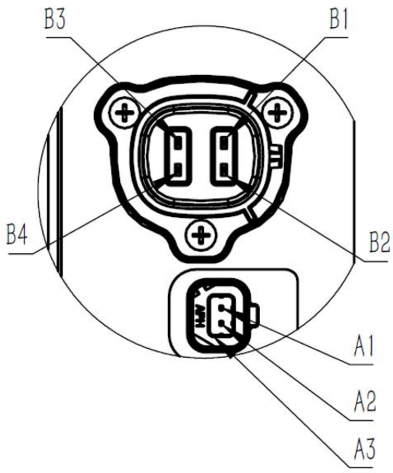
Definition of connector pin		
Pin Number	Signal Name	Connector Name
B1	VBAT	MOLEX-334824001
B2	GND	
B3	WakeuP	
B4	—	
A1	1000Base T1 P	Amphenol NTBM11V1U01110T
A2	1000Base T1 N	
A3	GND	

Drawing of LiDAR with interface AN2：



•

Definition of AN2-Pins:



Definition of connector pin		
Pin Number	Signal Name	Connector Name
B1	GND	LJV C-HSPPSNXS24T-A
B2	VBAT	
B3	WakeuP	
B4	——	
A1	1000Base T1 P	Amphenol NTHBV11A1001ST
A2	1000Base T1 N	
A3	GND	

•

0755-86325830

Smart Sensor, Safer World

深 圳 市 速 腾 聚 创 科 技 有 限 公 司
Shenzhen Suteng Innovation Technology Co., LTD.

Address: 深圳市南山区留仙大道 3370 号南山智园崇文园区 3 栋 10-11 层　10-11/F, Block 3, Chongwen Garden, Nanshan IPark, 3370 Liuxian Avenue, Nanshan District, Shenzhen, China

Web：www.robosense.ai　Tel: 0755-8632-5830　Email: service@robosense.cn