

# RS-LiDAR-M1 (B3) User Guide

## Revision History

| Revision | Content   | Date       | Edited by |
|----------|---|------------|-----------|
| 1.0      | 初版发行  | 2020-11-26 | PD        |
| 1.1      | 2.1.2.1节增加XYZ坐标运算公式<br>增加附录A RSView<br>增加附录B Driver&SDK<br>增加附录C MEMS Tool的使用<br>修订部分表述错误                                     | 2020-12-02 | PD        |
| 1.2      | 更新产品参数规格表   | 2021-02-23 | PD        |
| 1.3      | 更新MSOP和DIFOP内容<br>更新图2<br>增加附录D结构图纸   | 2021-03-11 | PD        |
| 1.4      | 通信协议增加数据传输速率要求说明<br>更新垂直FOV脚注及MSOP数据说明<br>增加3.1 Interface Box种类说明<br>增加3.2节 LiDAR状态机说明<br>增加章节4 时间同步<br>更新附录D结构图纸<br>修订部分表述问题 | 2021-06-07 | PD        |

## 目录

|  |    |
|--|----|
| 1 产品规格 .....                               | 1  |
| 2 通信协议 .....                               | 3  |
| 2.1 主数据流输出协议（MSOP） .....                   | 3  |
| 2.2 设备信息输出协议（DIFOP） .....                  | 9  |
| 3 LiDAR 接线及工作说明 .....                      | 11 |
| 3.1 Interface box 接线方式 .....               | 11 |
| 3.2 LiDAR 状态机说明 .....                      | 12 |
| 4 时间同步 .....                               | 13 |
| 4.1 精准时间同步协议（PTP） .....                    | 13 |
| 4.2 使用 Linuxptp 工具简单验证时间同步 .....           | 14 |
| 4.3 GPS 时间同步 .....                         | 16 |
| 附录 A RSView .....                          | 17 |
| A.1 软件功能 .....                             | 17 |
| A.2 安装 RSView .....                        | 17 |
| A.3 设置网路 .....                             | 17 |
| A.4 可视化数据 .....                            | 18 |
| A.5 保存 RS-LiDAR-M1（B3 样件）数据为 PCAP 格式 ..... | 19 |
| A.6 回放 pcap 数据 .....                       | 22 |
| 附录 B Driver & SDK .....                    | 26 |
| B.1 rs_driver 的编译与安装 .....                 | 26 |
| B.2 rlidar_sdk 的编译与安装 .....                | 28 |
| 附录 C MEMS Tool 的使用 .....                   | 35 |
| C.1 使用 MEMS Tool 与雷达建立通信 .....             | 35 |
| C.2 修改雷达 IP 及端口号 .....                     | 36 |
| C.3 固件升级 .....                             | 36 |
| 附录 D 结构图纸 .....                            | 38 |

# 1 产品规格

RS-LiDAR-M1(B3 样件)采用 MEMS 固态激光雷达方式，测量距离高达 200 米（150m @ 10%），出点数高达 750,000 点/秒（单回波模式）及 1,500,000 点/秒（双回波模式），水平测角 120°（-60.0°~+60.0°），垂直测角 25°（-12.5°~+12.5°）。

表 1: 产品规格表（B3 样件）

|           |   |
|-----------|---|
| 传感器       | <ul style="list-style-type: none"> <li>● TOF 法测距，包含反射强度信息</li> <li>● 测距：0.5m ~200m (150m@10% NIST)<sup>1</sup></li> <li>● 精度：± 3cm@1 sigma<sup>2</sup></li> <li>● 视角（垂直）：25°（-12.5°~+12.5°）<sup>3</sup></li> <li>● 角分辨率（垂直）：平均 0.2°<sup>4</sup></li> <li>● 视角（水平）：120°（-60.0°~+60.0°）</li> <li>● 角分辨率（水平）：平均 0.2°<sup>4</sup></li> <li>● 帧率：10Hz</li> </ul> |
| 激光        | <ul style="list-style-type: none"> <li>● Class 1 eye safe</li> <li>● 波长：905nm</li> </ul>  |
| 输出        | <ul style="list-style-type: none"> <li>● ~750,000 点/秒（单回波模式）</li> <li>● ~1,500,000 点/秒（双回波模式）</li> <li>● 1000Base-T1 千兆以太网</li> <li>● UDP 包中包含<br/>三维空间坐标、反射强度信息、时间戳等</li> </ul>  |
| 机械 / 电子操作 | <ul style="list-style-type: none"> <li>● 功耗：15w<sup>5</sup></li> <li>● 工作电压：9~36VDC</li> <li>● 重量：约 0.73kg（不包含数据线）</li> <li>● 尺寸：长 110mm * 宽 108mm * 高 45mm</li> </ul>  |

- 1 测距能力 150 米以 10% NIST 漫反射板作为目标，测试结果会受到环境影响，包括但不限于环境温度、光照强度等因素；
- 2 测距精度以 50% NIST 漫反射板为目标在 10m~100m 范围内测试，测试结果会受到环境影响，包括但不限于环境温度、目标物距离等因素，且精度值适用于大部分通道，部分通道之间存在差异；
- 3 RS-LiDAR-M1 整机 5 个通道水平排列，在垂直方向上会有一些位置上的错开；单个视场最大包络垂直 FoV 为 25.2°；5 个视场拼接后呈现非规则排列，按照最大包络计算则垂直 FoV 为 35.79°；
- 4 水平&垂直分辨率在整个 FOV 区域内并非均匀分布，角分辨率的均值为 0.2°；
- 5 设备功耗测试在稳定工作时测试，结果会受到外部环境影响，包括但不限于环境温度、目标物的距离、目标物反射率等因素；

|  |  |
|--|--|
|  | <ul style="list-style-type: none"><li>● 防护安全级别：IP67, IP6K9K</li><li>● 工作温度范围：-40℃~85℃（要求长时间工作需强制对流）<sup>6</sup></li><li>● 存储温度范围：-40℃~105℃</li></ul> |
|--|--|

---

<sup>6</sup> 设备运行温度可能会受到外部环境影响，包括但不限于太阳辐射、气流变化等因素；

## 2 通信协议

RS-LiDAR-M1(B3 样件)与电脑之间的通信采用以太网介质，使用 UDP 协议，输出包有两种类型：MSOP 包和 DIFOP 包。文中所有涉及 MSOP 协议包均为 1210 Bytes 定长；DIFOP 协议包均为 256 Bytes 定长。单回波模式下，每秒发射 6300 个 MSOP 包和 1 个 DIFOP 包，数据传输速率要求不低于 58.2 Mbps。双回波模式下，要求不低于 116.4 Mbps。RS-LiDAR-M1(B3 样件)网络参数可配置，出厂默认采用固定 IP 和端口号模式，按照如下表格。

表 2：出厂默认网络配置表

|             | IP 地址         | MSOP 包端口号 | DIFOP 包端口号 |
|-------------|---------------|-----------|------------|
| RS-LiDAR-M1 | 192.168.1.200 | 6699      | 7788       |
| 电脑          | 192.168.1.102 |           |            |

设备默认 MAC 地址是在工厂初始设置的，每台设备 MAC 地址唯一。

使用设备的时候，需要把电脑的 IP 设置为与设备同一网段上，例如 192.168.1.x(x 的取值范围为 1~254)，子网掩码为 255.255.255.0。若不知设备网络配置信息，请将主机子网掩码设置为 0.0.0.0 后连接设备并使用 Wireshark 抓取设备输出包进行分析。

RS-LiDAR-M1 和电脑之间的通信协议主要分两类，一览表见下表格。

- 主数据流输出协议 MSOP，将激光雷达扫描出来的距离，角度，反射率等信息封装成包输出给电脑；
- 设备信息输出协议 DIFOP，将激光雷达当前状态的各种配置信息输出给电脑。

表 3：设备协议一览表

| (协议/包) 名称                          | 简写    | 功能     | 类型  | 包大小        |
|------------------------------------|-------|--------|-----|------------|
| Main Data Stream Output Protocol   | MSOP  | 扫描数据输出 | UDP | 1210 Bytes |
| Device Information Output Protocol | DIFOP | 设备信息输出 | UDP | 256 Bytes  |

注：下面章节皆为对协议中的有效载荷（MSOP 包 1210 Bytes 和 DIFOP 包 256Bytes）部分进行描述和定义。

### 2.1 主数据流输出协议（MSOP）

主数据流输出协议：Main data Stream Output Protocol，简称：MSOP。

I/O 类型：设备输出，电脑解析。

默认端口号为 6699。

MSOP 包完成三维测量相关数据输出，包括激光测距值、回波的反射强度值、垂直角度、水平角度和时间戳。MSOP 包的有效载荷长度为 1210 字节，其中 32 Bytes 的同步帧

头 Header, 1175 Bytes 的数据块区间 (共 25 个 47 Bytes 的 data block), 3 Bytes 为帧尾。

基本结构如下图所示:

### MSOP Packet (1210 Bytes)

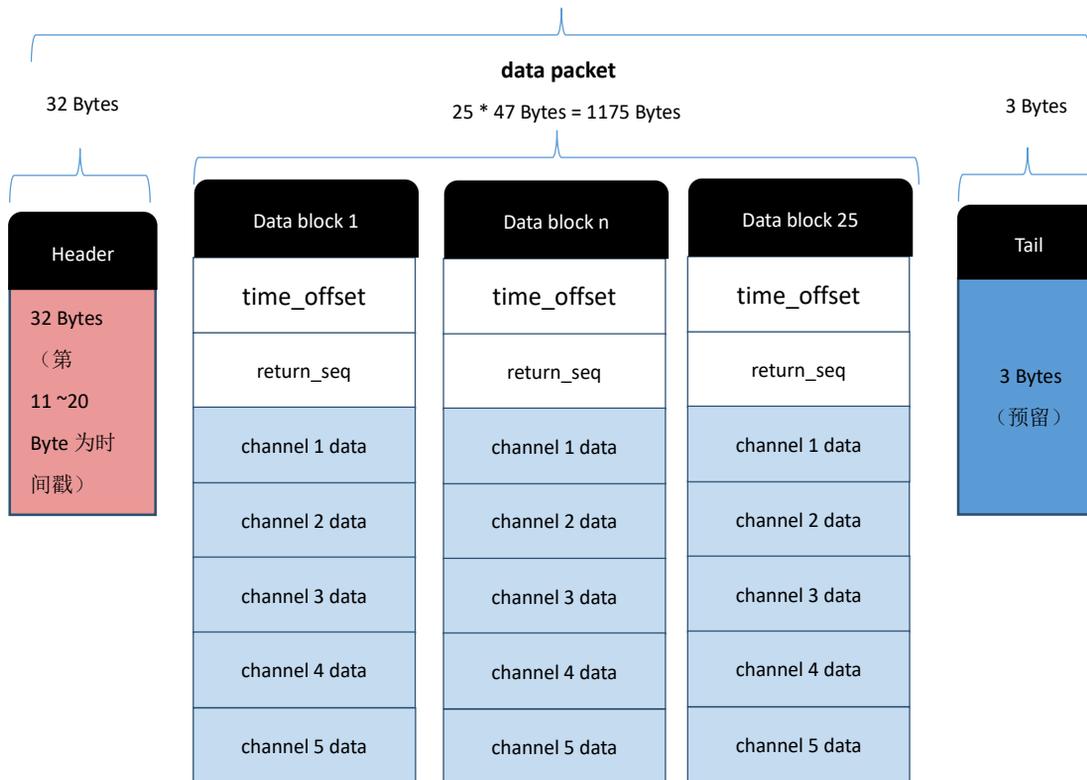


图 1: MSOP Packet 数据包定义示意图

#### 2.1.1 帧头

帧头 Header 共 32 Bytes, 用于识别出数据的开始位置, 包计数, UDP 通信预留以及存储时间戳。详细定义如下:

表 4: MSOP 包头定义

| 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:** 可作为包的检查序列，识别头为 0x55, 0xaa, 0x5a, 0xa5。

**pkt\_psn:** 包序列号，表示包计数，循环计数，从每帧数据的起点的包计数为 1，每帧数据的最后一个点的包计数为最大值。

**protocol version:** 表示 UDP 通信协议的版本号。

**wave\_mode:** 回波模式标志位，0-双回波，1 - N/A ， 2 - N/A， 3 - N/A， 4- 最强回波，5-最后回波， 6-第一回波。

**time\_sync\_mode :** 时间同步模式：

0x00 表示当前使用雷达内部自己计时

0x01 表示当前使用 1PPS 进行亚秒在整秒复位模式

0x02 表示当前使用 PTP 时间同步模式

**timestamp:** 用于存储时间戳，定义的时间戳用来记录系统的时间，高 6 Bytes 为秒位，低 4 Bytes 为微秒位。

**reserved:** 预留位

**lidar\_type:** 雷达类型标志位，默认值 0x10。

**mems\_tmp:** mems 温度，Temp=mems\_tmp-80;即原始值 0 代表-80 度； 255 代表 175 度。

## 2.1.2 数据块区域

数据块区间是 MSOP 包中传感器的测量值部分，共 1175 Bytes。它由 25 个 data block 组成，每个 data block 长度为 47 Bytes。

对于单回波模式，代表一组 5 个通道一次测量完整的测距数据。每一个 Data Block 就储存单次发射回波的数据。

对于双回波模式，单数个 MSOP Packet 为第一回波的数据，包含 25 个 Data Block。偶数个 MSOP Packet 为第二回波的数据，包含 25 个 Data Block。依次交替出现，可以根据 return\_seq 标志位来判定该 Packet 属于第几个回波，具体参看表 5 中说明。两个 MSOP Packet 组成一次完整的测量。双回波一次测量总点数为单回波一次测量的两倍。

详细定义如下：

表 5: MSOP 包中 data block 定义

| data block N (47 Bytes) |              |      |             |
|-------------------------|--------------|------|-------------|
| content                 | offset(byte) | byte | instruction |

|               |    |   |   |
|---------------|----|---|---|
| time_offset   | 0  | 1 | 该组 Block 里面所有的点相对于包的 timestamp 的时间偏移量，该组点的时间等于 $\text{timestamp} + \text{time\_offset}$ |
| return_seq    | 1  | 1 | 回波序列。单回波模式下，此标志位恒定为 0；双回波模式下，第一回波（距离更近的）用 0x1 表示，第二回波（距离更远的）用 0x2 表示                    |
| ch1_radius    | 2  | 2 | 极坐标系下，通道 1 的径向点距离值，距离解析分辨率 5mm  |
| ch1_elevation | 4  | 2 | 极坐标系下，通道 1 的点垂直夹角，分辨率 0.01°   |
| ch1_azimuth   | 6  | 2 | 极坐标系下，通道 1 的点水平夹角，分辨率 0.01°   |
| ch1_intensity | 8  | 1 | 通道 1 的点反射强度值，取值范围 0~255   |
| resev.        | 9  | 2 | 预留  |
| ch2_radius    | 11 | 2 | 极坐标系下，通道 2 的径向点距离值，距离解析分辨率 5mm  |
| ch2_elevation | 13 | 2 | 极坐标系下，通道 2 的点垂直夹角，分辨率 0.01°   |
| ch2_azimuth   | 15 | 2 | 极坐标系下，通道 2 的点水平夹角，分辨率 0.01°   |
| ch2_intensity | 17 | 1 | 通道 2 的点反射强度值，取值范围 0~255   |
| resev.        | 18 | 2 | 预留  |
| ch3_radius    | 20 | 2 | 极坐标系下，通道 3 的径向点距离值，距离解析分辨率 5mm  |
| ch3_elevation | 22 | 2 | 极坐标系下，通道 3 的点垂直夹角，分辨率 0.01°   |
| ch3_azimuth   | 24 | 2 | 极坐标系下，通道 3 的点水平夹角，分辨率 0.01°   |
| ch3_intensity | 26 | 1 | 通道 3 的点反射强度值，取值范围 0~255   |
| resev.        | 27 | 2 | 预留  |
| ch4_radius    | 29 | 2 | 极坐标系下，通道 4 的径向点距离值，距离解析分辨率 5mm  |
| ch4_elevation | 31 | 2 | 极坐标系下，通道 4 的点垂直夹角，分辨率 0.01°   |
| ch4_azimuth   | 33 | 2 | 极坐标系下，通道 4 的点水平夹角，分辨率 0.01°   |
| ch4_intensity | 35 | 1 | 通道 4 的点反射强度值，取值范围 0~255   |
| resev.        | 36 | 2 | 预留  |
| ch5_radius    | 38 | 2 | 极坐标系下，通道 5 的径向点距离值，距离解析分辨率 5mm  |
| ch5_elevation | 40 | 2 | 极坐标系下，通道 5 的点垂直夹角，分辨率 0.01°   |

|               |    |   |                             |
|---------------|----|---|-----------------------------|
| ch5_azimuth   | 42 | 2 | 极坐标系下，通道 5 的点水平夹角，分辨率 0.01° |
| ch5_intensity | 44 | 1 | 通道 5 的点反射强度值，取值范围 0~255     |
| resev.        | 45 | 2 | 预留                          |

N 为任一个 MSOP 包中第 N 个 data block。

**time\_offset:** 第 N 组 Block 里面所有的点相对于包的 timestamp 的时间偏移量，该组点的时间等于 timestamp+time\_offset。

**return\_seq:** 回波序列。单回波模式下，此标志位恒定为 0；双回波模式下，第一回波（距离更近的）用 0x1 表示，第二回波（距离更远的）用 0x2 表示。

n 为第 N 组 data block 中第 n 个通道，n=1, 2, 3, 4, 5，其包含数据如下：

**chn\_radius:** 极坐标系下，通道 n 的径向点距离值，距离解析分辨率 5mm。

**chn\_elevation:** 极坐标系下，通道 n 的点俯仰角，分辨率 0.01°。

**chn\_azimuth:** 极坐标系下，通道 n 的点方位角，分辨率 0.01°。

**chn\_intensity:** 通道 n 的点反射强度值，取值范围 0~255。

### 2.1.2.1 channel data 定义

Channel data 是 9 Bytes，此通道径向距离 2 Bytes、俯仰角 2 Bytes、方位角 2 Bytes、反射强度值 1 Byte、预留 2 Bytes。

详细定义如下：

表 6：data block 中通道数据定义

| channel data (9 Bytes)  |          |                            |         |                          |         |                           |
|-------------------------|----------|----------------------------|---------|--------------------------|---------|---------------------------|
| chn_radius<br>(2 Bytes) |          | chn_elevation<br>(2 Bytes) |         | chn_azimuth<br>(2 bytes) |         | chn_intensity<br>(1 Byte) |
| R1 [15:8]               | R2 [7:0] | E1[15:8]                   | E2[7:0] | A1[15:8]                 | A2[7:0] | Intensity[7:0]            |
| resv.<br>(2 Bytes)      |          |                            |         |                          |         |                           |
| r1 [15:8]               | r2 [7:0] |                            |         |                          |         |                           |

以径向距离 radius 计算为例：

chn\_radius 是 2 Bytes，单位是厘米（cm），分辨率为 0.5 cm。

获取数据包里的某通道 radius 值的十六进制数为：R1 为 0x03，R2 为 0xfc。

0x03 为距离的高位，转换为十进制为 3，0xfc 为距离的低位，转化为十进制为 252。

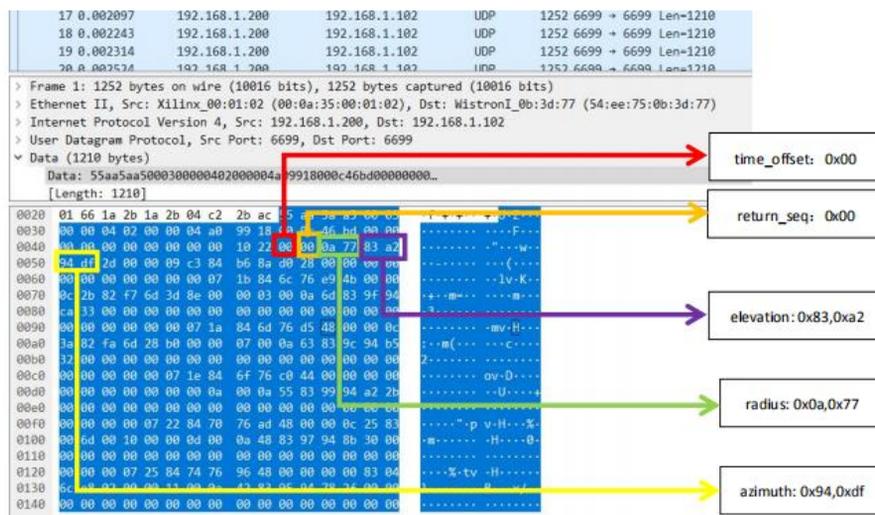
因此：此通道的径向距离=R1\*256+R2=3\*256+252=1020。

根据坐标的分辨率，转化为米：1020 \*0.005=5.10m。

因此，此通道的测距在对应 elevation 和 azimuth 方向上的径向距离是 5.1 米。

XYZ 坐标运算：

通过 Wireshark 抓取 RS-LiDAR-M1 的数据包，如下图所示：



相关参数计算样例

1. time\_offset: 数据块时间偏移量 HEX: 0x00 -> DEC: 00 -> 0  $\mu$ s
2. return\_seq: HEX: 0x00 -> 单回波
3. radius: 径向距离 HEX: 0x0a,0x77 -> DEC: 10, 119  
-> radius = (10 x 256 + 119) x 0.005 [m] = 13.395 m
4. elevation: 俯仰角 HEX: 0x83,0xa2 -> DEC: 131,162  
-> elevation = ((131 x 256 + 162)-32768) x 0.01[degree] = 9.3°
5. azimuth: 方位角 HEX: 0x94,0xdf -> DEC: 148,223  
-> azimuth = ((148 x 256 + 223)-32768) x 0.01[degree] = 53.43°

由下面的解析公式可以解析点云的 XYZ 坐标：

$$\left\{ \begin{array}{l} X = \text{radius} \cdot \cos(\text{elevation}) \cdot \cos(\text{azimuth}) \\ Y = \text{radius} \cdot \cos(\text{elevation}) \cdot \sin(\text{azimuth}) \\ Z = \text{radius} \cdot \sin(\text{elevation}) \end{array} \right.$$

$$\left\{ \begin{array}{l} X = 13.395\text{m} \cdot \cos(9.3^\circ) \cdot \cos(53.43^\circ) \\ Y = 13.395\text{m} \cdot \cos(9.3^\circ) \cdot \sin(53.43^\circ) \\ Z = 13.395\text{m} \cdot \sin(9.3^\circ) \end{array} \right.$$

由此得到该通道单回波的一次发射测量得到的点云坐标值 (7.88m,10.62m,2.17m)。

### 2.1.3 帧尾

帧尾 (Tail) 长度为 3 Bytes, 为预留位。

## 2.2 设备信息输出协议 (DIFOP)

设备信息输出协议, Device Info Output Protocol, 简称: DIFOP

I/O 类型: 设备输出, 电脑读取。

默认端口号为 7788。

DIFOP 是为了将设备序列号 (S/N)、固件版本信息、上位机驱动兼容性信息、网络配置信息、校准信息、运行状态、故障诊断信息定期发送给用户的“仅输出”协议, 用户可以通过读取 DIFOP 解读当前使用设备的各种参数的具体信息。

一个完整的 DIFOP Packet 的数据格式结构为同步帧头, 数据区, 预留。每个数据包共 256 Bytes: 包括 8 Bytes 同步帧头 Header, 1Byte 预留位, 247Bytes 的数据区。

数据包的基本结构如下表所示。

表 7: DIFOP 包定义

| 段落划分   | 序号   | 属性            | 信息               | Offset | 长度 (byte) |
|--------|------|---------------|------------------|--------|-----------|
| Header | 1    | 帧头            | DIFOP 识别头        | 0      | 8         |
|        | 2    | 预留            | 预留               | 8      | 1         |
| Data   | 3    | 帧率设置          | 帧率设置值            | 9      | 1         |
|        | 4    | 以太网           | 以太网 IP 源地址       | 10     | 4         |
|        |      |               | 以太网 IP 目标地址      | 14     | 4         |
|        |      |               | 以太网 IP 本机 MAC 地址 | 18     | 6         |
|        |      |               | MSOP 端口号         | 24     | 2         |
|        |      |               | DIFOP 端口号        | 26     | 2         |
|        | 5    | FOV 设置 (暂未启用) | 水平 FOV 起始角度      | 28     | 2         |
|        |      |               | 水平 FOV 结束角度      | 30     | 2         |
|        |      |               | 垂直 FOV 起始角度      | 32     | 2         |
|        |      |               | 垂直 FOV 结束角度      | 34     | 2         |
|        | 7    | 版本信息          | 主板 PL 侧固件版本号     | 36     | 5         |
|        |      |               | 主板 PS 侧固件版本号     | 41     | 5         |
|        | 9    | 产品 SN 信息      | 产品序列号            | 46     | 6         |
|        | 11   | wave_mode     | 回波模式设置           | 52     | 1         |
| 12     | 时间信息 | 时间同步方式设置      | 53               | 1      |           |
|        |      | 时间同步状态        | 54               | 1      |           |

|  |    |      |                 |     |    |
|--|----|------|-----------------|-----|----|
|  |    |      | 时间              | 55  | 10 |
|  | 13 | 运行状态 | 电压、电流、输入输出信号等状态 | 65  | 20 |
|  | 15 | 故障诊断 | 故障诊断预留          | 85  | 40 |
|  | 17 | 设备内参 | 标定参数            | 125 | 60 |
|  | 18 | 预留   | 预留              | 185 | 71 |

**注：**

- 表格中 Header (DIFOP 识别头) 为 0xa5,0xff,0x00,0x5a,0x11,0x11,0x55,0x55, 可作为包的检查序列。
- 水平 FOV 的 LSB 为 0.01 度, 最小值 0, 最大值 120°。
- 垂直 FOV 的 LSB 为 0.01 度, 最小值 0, 最大值 25°。
- 回波模式设置: 回波模式标志位, 0-双回波, 1 - N/A, 2 - N/A, 3 - N/A, 4- 最强回波, 5-最后回波, 6-第一回波。
- 时间同步方式设置: 默认值是 0x02。0x00 表示当前使用雷达内部自己计时, 0x01 表示当前使用 1PPS 进行亚秒在整秒复位模式, 0x02 表示当前使用 PTP 时间同步模式。
- 时间同步状态: 标识时间同步是否成功的状态: 0-不成功, 1-成功。
- 标定参数: 一共 20 个参数, 每个参数由 3Bytes 组成, 第一个 Byte 用于表示符号 (0 为正, 1 为负), LSB=0.01, 顺序对应 ChannelNum.csv 文件参数前 20 个。

### 3 LiDAR 接线及工作说明

#### 3.1 Interface box 接线方式

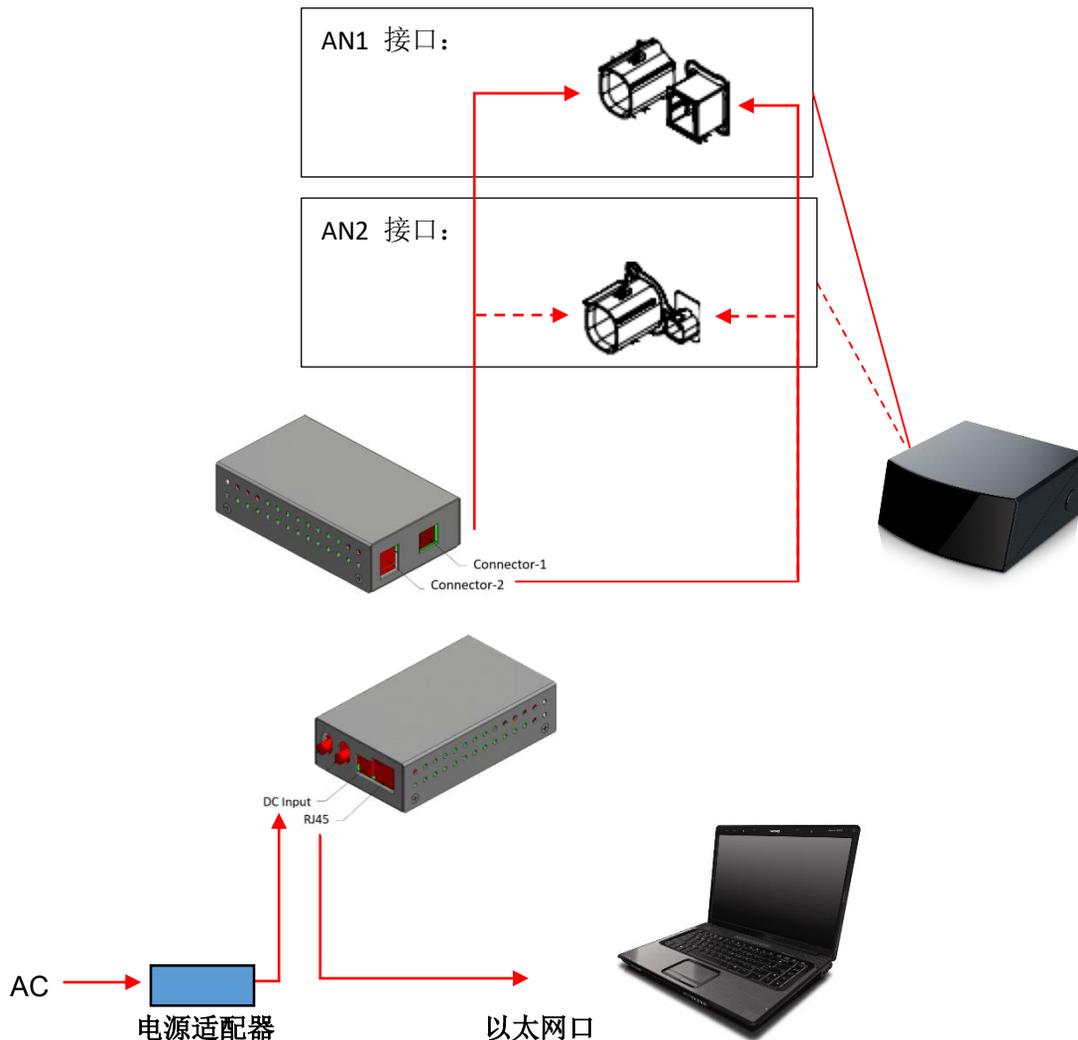
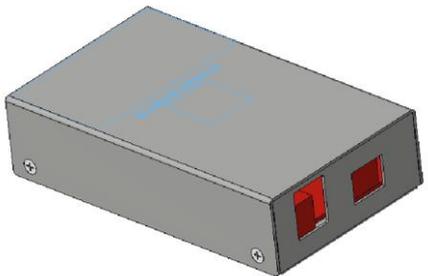
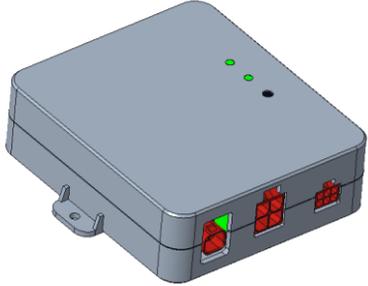
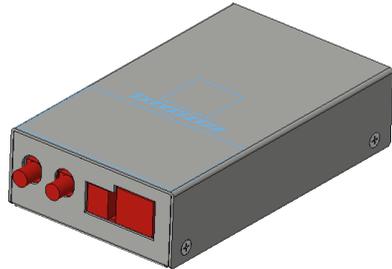
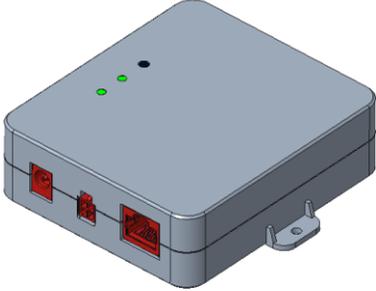


图 2：雷达设备与上位机直连拓扑图

注：

图 2 中展示的是 Interface Box (AN1) 结构图的接线方式。目前 RS-LiDAR-M1 (B3 样件) 支持 AN1 和 AN2 两种接口，分别对应以下两种 Interface Box(电源转接盒)：

| 接线说明 | AN1 Interface Box 结构图 | AN2 Interface Box 结构图 |
|------|-----------------------|-----------------------|
|------|-----------------------|-----------------------|

|                            |   |  |
|----------------------------|---|--|
| <p>连接<br/>LiDAR 侧</p>      |  |  |
| <p>连接<br/>电源及上<br/>位机侧</p> |  |  |

### 3.2 LiDAR 状态机说明



整机状态机描述：  
 S1：整机未供电  
 S2：整机正常运行

## 4 时间同步

RS-LiDAR-M1 (B3 样件) 默认固件使用 PTP 1588v2 的时间同步方式。雷达默认只支持 PTPv2, 如需要使用 gPTP 时间同步, 请联系 RoboSense 技术支持。

### 4.1 精准时间同步协议 (PTP)

PTP (Precision Time Protocol, 精确时间协议) 是一种时间同步的协议, 其本身主要用于通过网络通讯的形式实现设备之间的高精度时间同步, 也可被借用于设备之间的频率同步。相比现有的各种时间同步机制, PTP 具备以下优势:

- 1) 相比 NTP (Network Time Protocol, 网络时间协议), PTP 能够满足更高精度的时间同步要求, NTP 一般只能达到亚秒级的时间同步精度, 而 PTP 则可达到亚微秒级。
- 2) 相比 GPS (Global Positioning System, 全球定位系统), PTP 具备更低的建设和维护成本, 并且由于可以摆脱对 GPS 的依赖, 在国家安全方面也具备特殊的意义。

#### 4.1.1 PTP 接线方式

使用 PTP 同步方式, 需要做以下准备, 然后按照下图的连接方式进行连接:

- 1) 一台 PTP Master 授时主机 (即插即用, 无需额外配置);
- 2) 以太网交换机;
- 3) 支持 PTP 协议的设备 (RS-LiDAR-M1 及其他待授时设备);

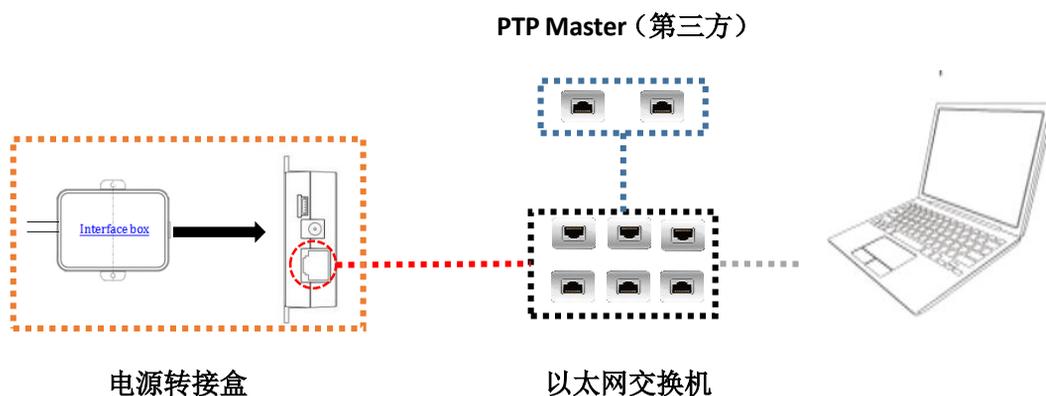


图 3: PTP 时间同步拓扑

注:

1. PTP Master 授时设备属于第三方设备, 我司出货时不包含此配件, 需要用户自行采购;
2. 我司设备作为 PTP Slave 设备只获取 PTP Master 发出的时间, 不做准确度判断。若解析雷达点云

时间与真实时间出现偏差，请检查 PTP Master 提供的时间是否准确；

3. 雷达同步之后，若 PTP Master 断开连接，点云数据包中的时间会继续按照雷达内部时钟进行叠加。

雷达断电重启后时间才会被重置。

### 4.1.2 PTP 模式介绍

PTP 支持点对点（Peer to Peer）和端到端（End to End）两种模式。对于 RS-LiDAR-M1（B3 样件）默认只支持 P2P 模式。

P2P（Peer to Peer）模式：采用端延迟机制（Peer Delay Mechanism）

E2E（End to End）模式：采用请求应答机制（Request Response Mechanism）

## 4.2 使用 Linuxptp 工具简单验证时间同步

将 RS-LiDAR-M1（B3 样件）电源线和网线与 Interface Box 相连，再与上位机相连。

上位机操作系统（OS）需为 Linux 系统，以下以 Ubuntu 为例。

1. 使用命令 `$ifconfig` 查看网卡名。如下图所示网卡名为 `enp2s0`。

```
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)
```

图 4：查找网卡名示意图

2. 使用命令 `$ethtool -T enp2s0`（网卡名），可以查看此网卡是否支持 PTP 硬件。

```
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
```

图 5：检查 PTP 硬件支持情况示意图

3. 下载并安装 `linuxptp` 工具。

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

```
$cd linuxptp
```

```
$sudo make
```

```
$sudo make install
```

```
$reboot
```

4. `ptp4l` 命令的使用。

简单同步 RS-LiDAR-M1 (B3 样件) 使用命令:

```
$sudo ptp4l -P -S -4 -m -i enp2s0 (网卡名)
```

命令选项介绍:

#### 延迟机制选项

-A 自动模式, 自动选择 E2E 延迟机制, 当收到对等延迟请求时切换到 P2P。

-E E2E 模式, 请求应答延迟机制 (默认)

-P P2P 模式, 端延迟机制

#### 网络传输选项

-2 IEEE 802.3

-4 UDP IPV4 (默认)

-6 UDP IPV6

#### 时间戳选项

-H 硬件时间戳 (默认)

-S 软件模拟时间戳

-L 老的硬件时间戳, LEGACY HW 需要配合 PHC 设备使用。

#### 其他选项

-f [file] 从指定文件 `file` 中读取配置。默认情况下不读取任何配置文件。

-i [dev] 选择 PTP 接口设备, 例如 `eth0` (可多次指定) 必须至少使用此选项或配置文件指定一个端口。

-p [dev] 此选项用于在旧 Linux 内核上指定要使用的 PHC 设备 (例如 `/dev/ptp0` 时钟设备), 默认为 `auto`, 忽略软件/LEGACY HW 时间戳(不推荐使用此选项)

-s `slaveOnly mode`, 从时钟模式 (覆盖配置文件)

.

- t 透明时钟模式
- l [num] 将日志记录级别设置为'num', 默认是 6
- m 将消息打印到 `stdout`
- q 不打印消息到 `syslog`
- v 打印软件版本并退出
- h 帮助命令

### 4.3 GPS 时间同步

如需要将 RS-LIDAR-M1 (B3 样件) 与 GPS 模块同步。首先需要使 GPS 模块给 PTP Master 授时, 具体接口与授时方式需要与 PTP Master 提供方明确。除特殊需求外, RoboSense 将不提供相关技术支持。



图 6: GPS 同步拓扑简图

## 附录 A RSView

在本附录中将展示如何使用 RSView 获取、可视化、保存和回放 RS-LiDAR-M1(B3 样件)数据。

对于从 RS-LiDAR- M1 (B3 样件) 得到的原始数据, 可以使用一些免费工具去检测, 例如 Wireshark 和 tcp-dump。但对于可视化这些数据, 使用 RSView 是更为便捷和容易实现的方式, 具体的 RSView 版本可以联系 RoboSense 技术人员。

### A.1 软件功能

RSView 提供将 RS-LiDAR- M1 (B3 样件) 数据进行实时可视化的功能。RSView 也能回放保存为 pcap 文件格式的数据, 但是还不能支持.pcapng 格式的文件。

RSView 将 RS-LiDAR- M1 (B3 样件) 得到距离测量值显示为一个点。它能够支持多种自定义颜色来显示数据, 例如 XYZ 坐标、径向距离、pitch (Elevation) 和 yaw (azimuth) 角度等。

RSView 所包含的功能:

- 通过以太网实时显示数据
- 将实时数据记录保存为 PCAP 文件
- 从记录的 PCAP 文件中回放
- 不同类型可视化模式, 例如距离、pitch (elevation) 和 yaw(azimuth)等等
- 用表格显示点的数据
- 测量距离工具

### A.2 安装 RSView

RSView 的安装文件支持 Windows 的 64 位操作系统, 安装前不需要安装其他依赖软件。解压 RSView 的压缩包, 在/bin 文件夹下得到 RSView.exe 可执行文件。

### A.3 设置网路

雷达在出厂时设定的发送电脑的 IP 地址, 因此默认情况下需要设定计算机的静态 IP 的地址为 192.168.1.102, 子网掩码为 255.255.255.0。此外还需要确保 RSView 没有被防火墙或第三方安全软件给禁止。

## A.4 可视化数据

1. RS-LiDAR- M1（B3 样件）接通电源，并用网线和电脑连接。
2. 右键使用管理员权限运行打开 RSView 软件。
3. 点击 **File > Open** 并且选择 **Sensor Stream** (图 A-1)。

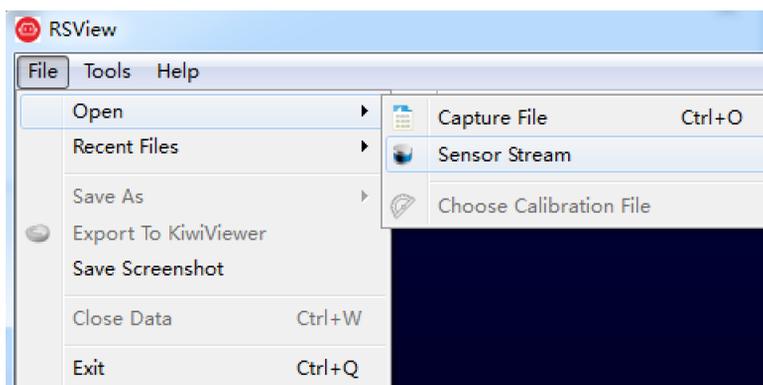


图 A-1: 打开 RSView 实时数据显示

4. 在弹出的 **Sensor Configuration** 窗口中，**Sensor Calibration** 默认包含一个命名为 MEMSCorrectionFile\_3V 的雷达参数，选择此参数后点击 OK。RS-LiDAR-M1（B3 样机）原始点云输出的已经是校准过的点云，所以此参数文件里的值为空。

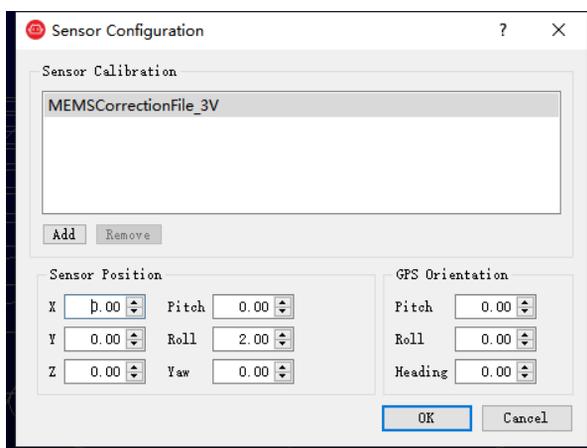


图 A-2: 选择 RS-LiDAR- M1（B3 样件）参数配置文件

5. 确认 MSOP 和 DIFOP 端口号是否正确：Tools > Data Port Setting

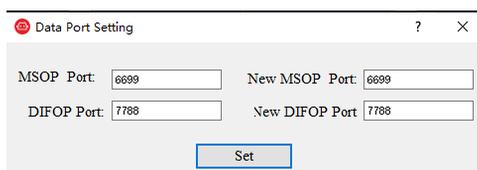


图 A-3: RSView 雷达端口设置示意图

6. RSView 开始显示实时采集到数据(图 A-4)。可以通过点击 **Play** 按钮暂停，再点击一次可以继续显示。

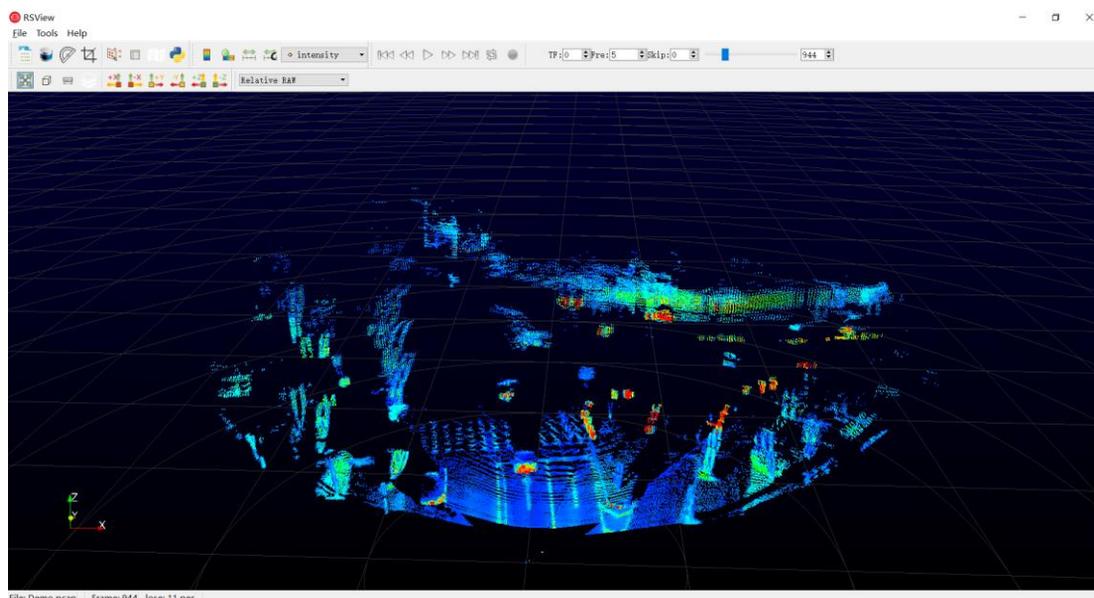


图 B-4: RS-LiDAR-M1(B3 样件)点云图像

7. 如没有图像显示请在工具栏 Tool 工具下 Data Port Setting 中查看 MSOP 和 DIFOP 端口是否配置正确。

## A.5 保存 RS-LiDAR- M1（B3 样件）数据为 PCAP 格式

采用 **RSView** 作为录包工具：

1. 在实时显示数据时点击 **Record** 按钮(图 A-5)。



图 A-5: RSView 保存按钮

2. 在弹出的 **Choose Output File** 对话框中，选择需要保存的路径和保存的文件名后，点击“保存 (S)”按钮(图 B-6)。RSView 将开始将数据包文件写入目标 pcap 文件中。(注意：RS-LiDAR- M1（B3 样件）将会产生大量的数据，随着记录时间变长，目标 pcap 文件将会变大。因此最好将记录文件保存到 HDD 或者 SSD 中，而不是保存到较慢的 USB 设备或者用网络保存)。

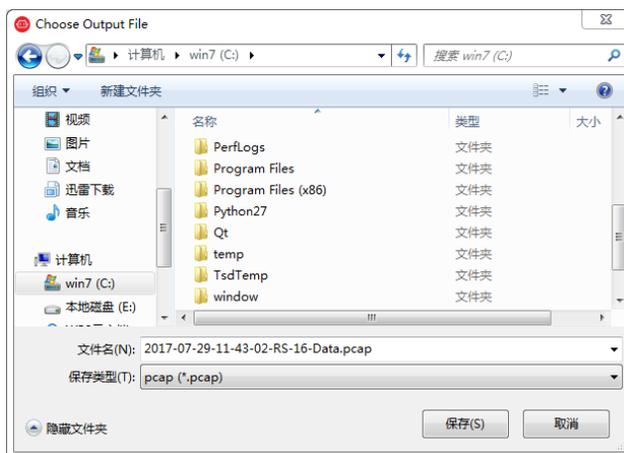


图 A-6: RSVIEW 保存对话框

3. 再次点击 **Record** 按钮停止保存 pcap 数据。

采用 **Wireshark** 作为录包工具:

1. 下载安装 wireshark 软件。



图 A-7: Wireshark 图标

2. 双击启动 **wireshark** 软件，启动后选择当前连接雷达的网卡名称双击进去。

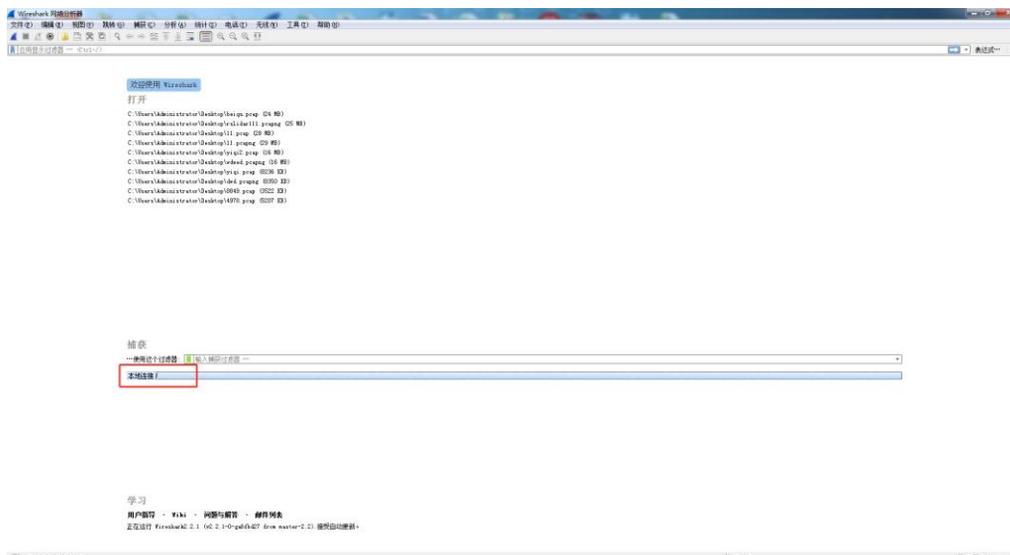


图 A-8: Wireshark 启动

3. 看到下图数据则说明和雷达连接正常，红色框内数据分别代表“雷达 IP”，“PC IP”，“MSOP 端口号”，“DIFOP 包端口号”。

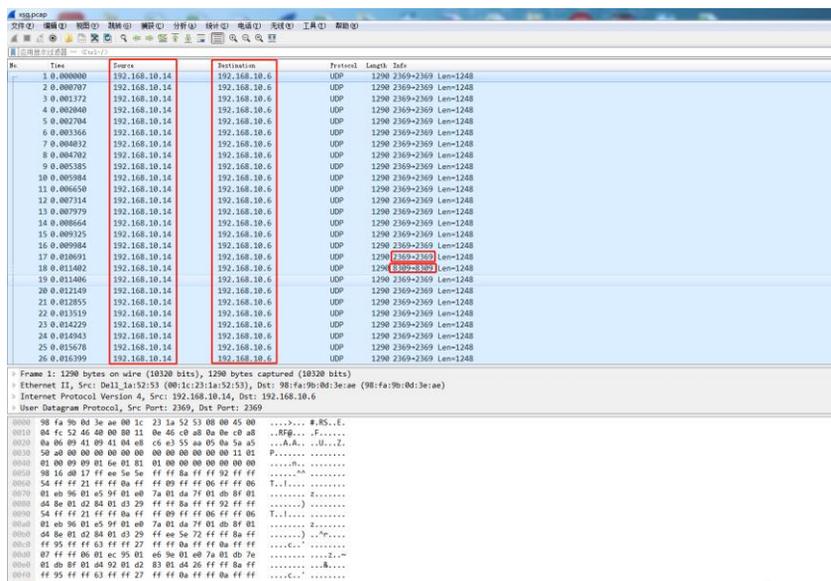


图 A-9: Wireshark 工作

4. 软件左上角找到“文件-保存”。

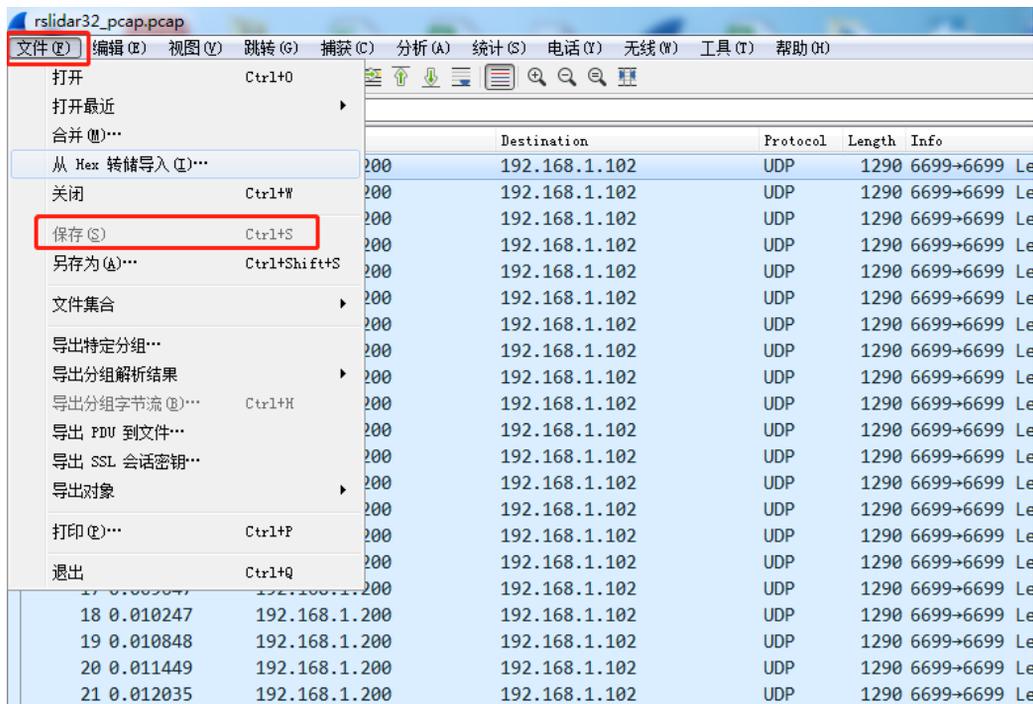


图 A-10: Wireshark 数据保存

5. 在弹出的对话框中输入文件名称，数据格式选择.pcap 保存即可。

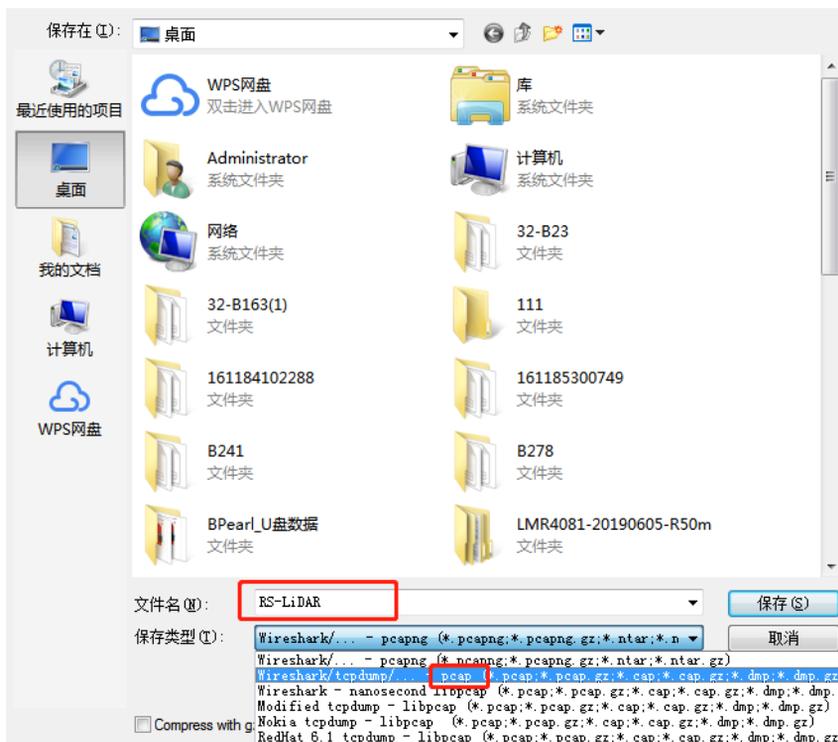


图 A-11: Wireshark 数据保存

6. 此时可以在指定的文件夹目录下找到对应的文件，此时可以使用我们的 RSView 软件或者驱动去查看点云（RSView 操作请参照产品用户手册）。

|                              |                  |                    |      |
|------------------------------|------------------|--------------------|------|
| VisualSVN-GlobalWinAuthz.ini | 2019/5/16 10:14  | 配置设置               | 1 KB |
| vsvnvars.bat                 | 2013/1/7 9:52    | Windows 批处理...     | 1 KB |
| <b>RS-LiDAR.pcap</b>         | 2019/10/30 15:42 | Wireshark captu... | 2 KB |

图 A-12 Wireshark 数据保存

## A.6 回放 pcap 数据

可以使用 RSView 对 RS-LiDAR- M1（B3 样件）保存的数据 pcap 文件进行回放或者测试。您可以使用 **Play** 按钮去播放或者选择数据中个别帧。也可以用鼠标选择 3D 点云数据中的一部分，然后打开表格进行分析。pcap 文件的保存路径不可以有中文。

1. 点击 **File > Open** 并且选择 **Capture File**。

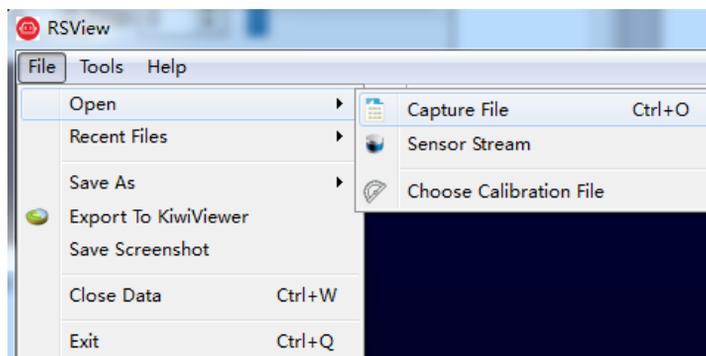


图 A-13 打开 pcap 记录文件

2. 弹出 Open File 对话框，选择一个记录的 pcap 文件并且点击“打开 (O)”按钮。

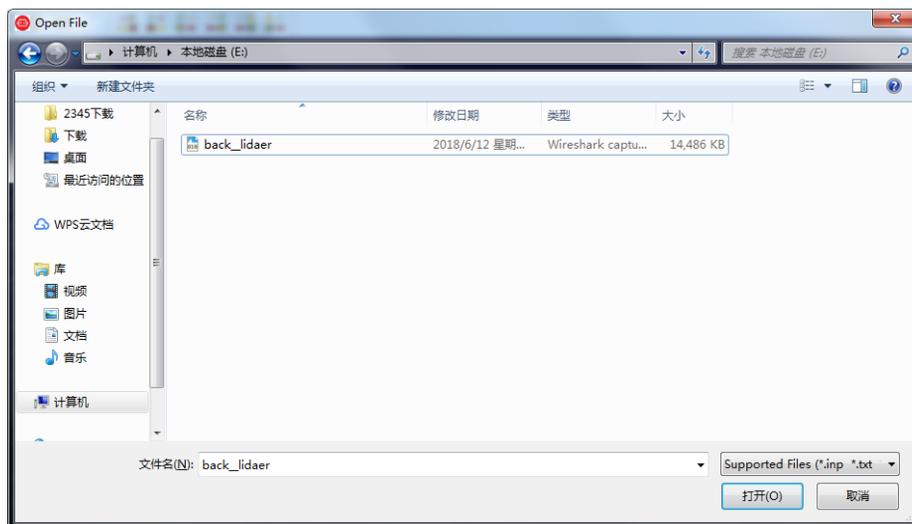


图 A-14 打开 pcap 记录文件

3. 弹出 Sensor Configuration 对话框，添加并选择正确的 RS-LiDAR- M1 (B3 样件) 的配置文件并点击 **OK** 按钮。

4. 点击 **Play** 按钮可以播放或者暂停数据。使用 **Scrub** 滑动工具前后滑动可以选择数据中不同位置的帧，此工具和 **Record** 按钮在同一个工具栏内(图 A-15)。

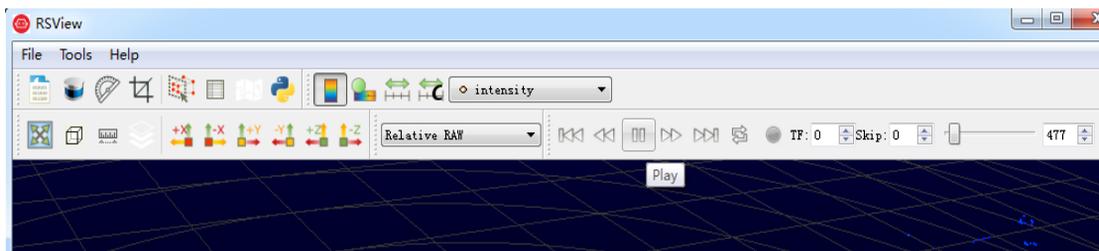


图 A-15 RSView Play 按钮和 Scrub 滑动工具

5. 为了得到更为具体的分析，选择一帧您感兴趣的数据并且点击 **Spreadsheet** 按钮(图 A-16)。一个侧边栏数据表将会显示在软件中右边，在表中包含了这一帧所有的数据。



图 A-16 RSVIEW 表格工具

6. 可以调整表格每列的宽度，或者排序得到更直观的数据显示。

| 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  |  |

图 A-17 RSVIEW 表格显示

7. 点击 Spreadsheet 中的 **Show only selected elements**，可以得到所选择点对应的数据，当然如果没有选择，表格将为空(图 A-18)。

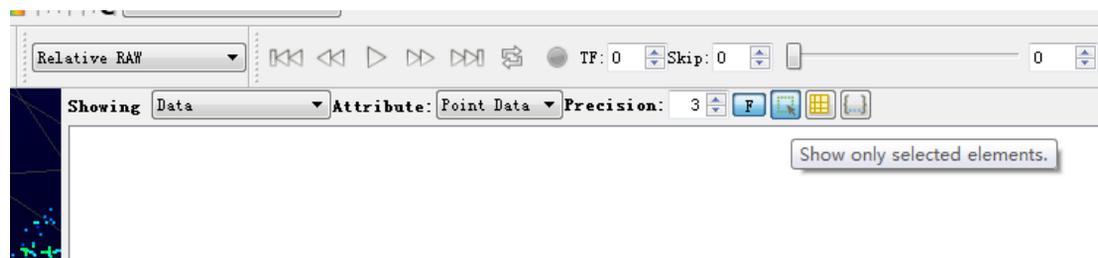


图 A-18 RSVIEW show only elements 工具

8. 点击 **Select All Points** 工具，这使得您的鼠标变成一个数据点选择工具（图 A-19）。

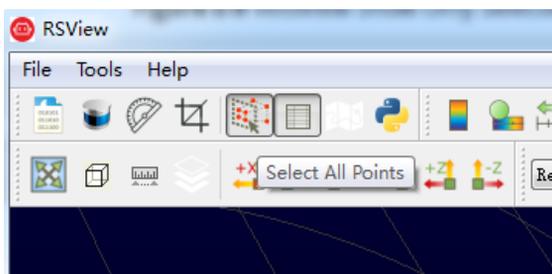


图 A-19 RSView Select All Points 工具

9. 在 3D 数据显示空间中，使用鼠标画一个长方形框住一些数据点，这些点的数将会在 Spreadsheet 被选择，并且在图中会变成粉红色(图 A-20)。

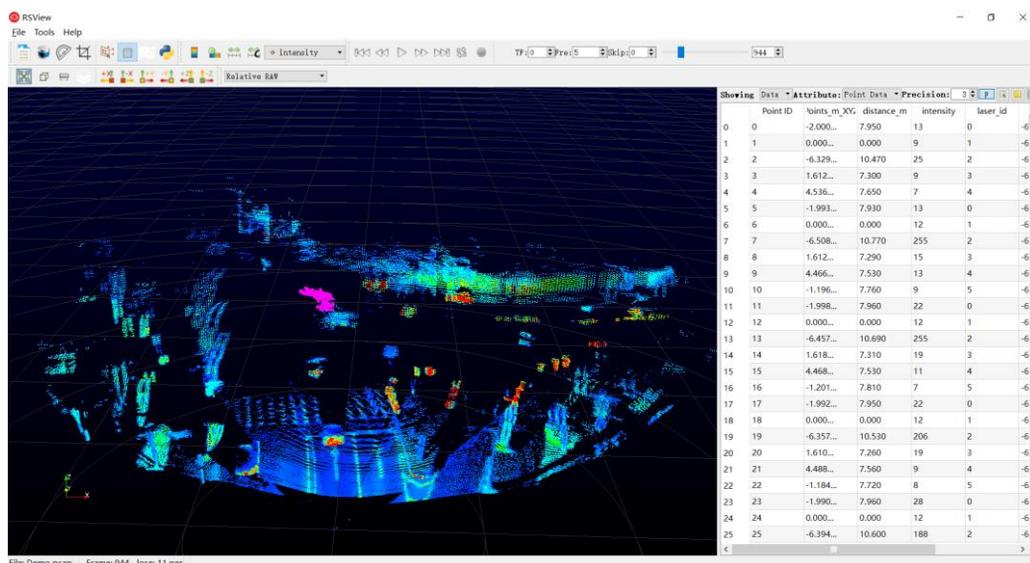


图 A-20 RSView List Selected Points

10. 任何被选中的点都可以通过 Spreadsheet 工具栏 output csv data 保存（见图 A-21）。

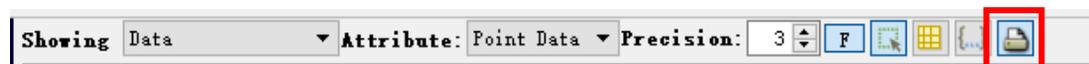


图 A-21 RSView 输出选中点到 csv 文件

## 附录 B Driver & SDK

### B.1 rs\_driver 的编译与安装

RS Driver 为速腾聚创激光雷达提供跨平台的雷达驱动内核，方便用户二次开发使用。v1.3.0 的驱动内核及之后的版本已支持 RS-LiDAR-M1(B3 样件)的点云解析及变换。

可以在官方 GitHub 账号上下载 rs\_driver 包:

[https://github.com/RoboSense-LiDAR/rs\\_driver](https://github.com/RoboSense-LiDAR/rs_driver)

rs\_driver 目前支持下列系统和编译器:

- Windows
  - MSVC (VS2017 & VS2019 已测试)
  - Mingw-w64 (x86\_64-8.1.0-posix-seh-rt\_v6-rev0 已测试)
- Ubuntu (16.04, 18.04, 20.04)
  - gcc (4.8+)

#### B.1.1 依赖库的安装

rs\_driver 依赖下列的第三方库，在编译之前需要先安装:

- Boost
- Pcap
- PCL (非必须，不需要可视化工具可忽略)
- Eigen3 (非必须，不需要内置坐标变换可忽略)

在 Ubuntu 中安装以上依赖库:

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

在 Windows 中安装以上依赖库:

- Boost

Windows 下需要从源码编译 Boost 库，请参考官方指南 ([https://www.boost.org/doc/libs/1\\_67\\_0/more/getting\\_started/windows.html](https://www.boost.org/doc/libs/1_67_0/more/getting_started/windows.html))。编译安装完成之后，将 Boost 的路径添加到系统环境变量 BOOST\_ROOT，见下图 B-1。如果使用 MSVC，也可以选择直接下载相应版本的预编译的安装包。

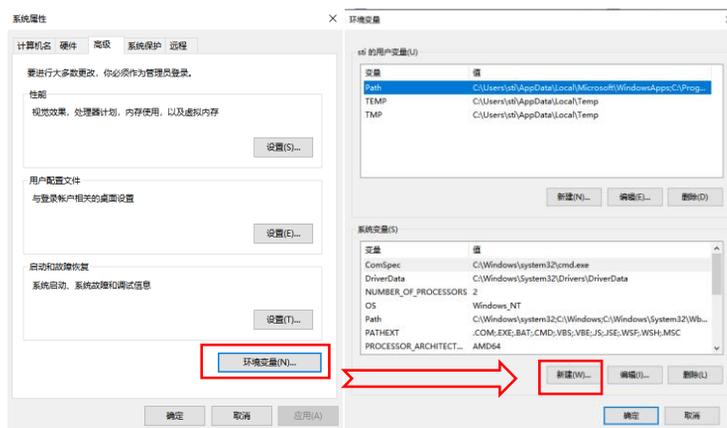


图 B - 1: 环境变量添加示意图

### ➤ Pcap

首先，安装 Pcap 运行库（[https://www.winpcap.org/install/bin/WinPcap\\_4\\_1\\_3.exe](https://www.winpcap.org/install/bin/WinPcap_4_1_3.exe)）。然后，下载开发者包（[https://www.winpcap.org/install/bin/WpdPack\\_4\\_1\\_2.zip](https://www.winpcap.org/install/bin/WpdPack_4_1_2.zip)）到任意位置，然后将 WpdPack\_4\_1\_2/WpdPack 的路径添加到环境变量 PATH，见图 B-1。

➤ PCL (非必须，如果不需要可视化工具可忽略)

#### (1) MSVC

如果使用 MSVC 编译器，可使用 PCL 官方提供的安装包安装。

安装过程中选择 “Add PCL to the system PATH for xxx”:

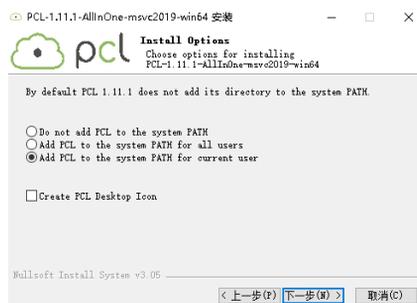


图 B - 2: PCL 设置界面

#### (2) Mingw-w64

PCL 官方并没有提供 mingw 编译的库，所以需要按照官方教程，从源码编译 PCL 并安装。

## B.1.2 使用方式

### B.1.2.1 rs\_Driver 安装使用

驱动编译以 Linux 环境为例（在 windows 中，rs\_driver 暂不支持安装使用），按顺序执行以下代码，安装驱动：

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

### B.1.2.2 作为子模块使用

在作为子模块使用时，需要添加如下命令到 CMakeLists.txt 文件中（将 rs\_driver 作为子模块添加到工程内，使用 find\_package() 指令找到 rs\_driver，然后链接相关库）。

```
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 示例程序 & 可视化工具

### B.1.3.1 示例程序

rs\_driver 提供了两个示例程序，用户可参考示例程序编写代码调用接口，存放于 rs\_driver/demo 中：

demo\_online.cpp

demo\_pcap.cpp

若希望编译这两个示例程序，执行 CMake 配置时加上参数：

```
$cmake -DCOMPILER_DEMOS=ON ..
```

### B.1.3.2 可视化工具

rs\_driver 提供了一个基于 PCL 的点云可视化工具，存放于 rs\_driver/tool 中：

rs\_driver\_viewer.cpp

若希望编译可视化工具，执行 CMake 配置时加上参数：

```
$cmake -DCOMPILER_TOOLS=ON ..
```

### B.1.4 坐标变换

rs\_driver 提供了内置的坐标变换功能，可以直接输出经过坐标变换后的点云，节省了用户对点云进行坐标变换的额外操作耗时。若希望启用此功能，执行 CMake 配置时加上参数：

```
$cmake -DENABLE_TRANSFORM=ON ..
```

## B.2 rlidar\_sdk 的编译与安装

rslidar\_sdk 为速腾聚创在 Ubuntu 环境下的雷达驱动软件包，包括了雷达驱动内核，ROS 拓展功能，ROS2 拓展功能，Protobuf-UDP 通信拓展功能。对于没有二次开发需求的用户，或是想直接使用 ROS 或 ROS2 进行二次开发的用户，可直接使用本软件包，配合 ROS 或 ROS2 自带的 RVIZ 可视化工具即可查看点云。对于有更深一步二次开发需求，想将雷达驱动集成到自己工程内的客户，请参考雷达驱动内核的相关文档，直接使用内核 rs\_driver 进行二次开发。

可以在官方 GitHub 账号上下载 rslidar\_sdk.tar.gz 包，可以在官方 GitHub 账号上下载：

[https://github.com/RoboSense-LiDAR/rslidar\\_sdk/releases](https://github.com/RoboSense-LiDAR/rslidar_sdk/releases)

Assets 3



注意：下载 source code 将不包含 rs\_driver 解析内核，需要手动下载添加。

### B.2.1 依赖库的安装

#### B.2.1.1 ROS 环境

若需在 ROS 环境下使用雷达驱动，则需安装 ROS 相关依赖库：

Ubuntu 16.04 - ROS kinetic desktop-full

Ubuntu 18.04 - ROS melodic desktop-full

安装方式：参考 <http://wiki.ros.org>

如果安装了 ROS kinetic desktop-full 版或 ROS melodic desktop-full 版，那么兼容版本其他依赖库也应该同时被安装了，所以不需要重新安装它们以避免多个版本冲突引起的问题，因此，强烈建议安装 desktop-full 版，这将节省大量的时间来逐个安装和配置库。

### B.2.1.2 ROS2 环境

若需在 ROS2 环境下使用雷达驱动，则需安装 ROS2 相关依赖库

Ubuntu 16.04 - 不支持

Ubuntu 18.04 - ROS2 Eloquent desktop

安装方式: 参考 <https://index.ros.org/doc/ros2/Installation/Eloquent/Linux-Install-Debian/>

**注意: 请避免在同一台电脑上同时安装 ROS 和 ROS2, 这可能会产生冲突! 同时还需要手动安装 Yaml 库。**

### B.2.2 编译与运行

rslidar\_sdk 可以使用三种不同的编译和运行方式。

#### B.2.1.1 直接编译

按照如下指令即可编译运行程序。直接编译也可以使用 ROS 相关功能(不包括 ROS2), 但需要在程序启动前手动启动 roscore, 启动后手动打开 rviz 才能看到可视化点云结果。

编译命令如下:

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

#### B.2.1.2 依赖于 ROS-catkin 编译

1. 打开工程内的 CMakeLists.txt 文件, 将文件顶部的 set(COMPILER\_METHOD ORIGINAL) 改为 set(COMPILER\_METHOD CATKIN)。

```
#=====
```

```
# Compiler setup (ORIGINAL,CATKIN,COLCON)
```

```
#=====
```

```
set(COMPILER_METHOD CATKIN)
```

2. 将 rslidar\_sdk 工程目录下的 package\_ros1.xml 文件重命名为 package.xml。

3. 新建一个文件夹作为工作空间, 然后再新建一个名为 src 的文件夹, 将 rslidar\_sdk 工程放入 src 文件夹内。

4. 返回工作空间目录, 执行以下命令即可编译&运行(若使用 .zsh, 将第二句指令替换为 source devel/setup.zsh)。

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

#### B.2.1.3 依赖于 ROS2-colcon 编译

1. 打开工程内的 CMakeLists.txt 文件, 将文件顶部的 set(COMPILER\_METHOD ORIGINAL) 改为 set(COMPILER\_METHOD COLCON)。

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

```
set(COMPILE_METHOD COLCON)
```

2. 将 rslidar\_sdk 工程目录下的 package\_ros2.xml 文件重命名为 package.xml。
3. 新建一个文件夹作为工作空间，然后再新建一个名为 src 的文件夹，将 rslidar\_sdk 工程放入 src 文件夹内。
4. 通过链接下载 ROS2 环境下的雷达 packet 消息定义，将 rslidar\_msg 工程也放在刚刚新建的 src 文件夹内，与 rslidar\_sdk 并列。
5. 返回工作空间目录，执行以下命令即可编译&运行(若使用 .zsh,将第二句指令替换为 source install/setup.zsh)。

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

### B.2.3 参数介绍

本工程只有一份参数文件 config.yaml，储存于 rslidar\_sdk/config 文件夹内。整个参数文件可以被分为两部分，common 部分以及 lidar 部分。在多雷达情况下，common 部分的参数设置将会被所有雷达共享，而 lidar 部分需要根据每台雷达实际情况分别进行设置。  
**注意：参数文件 config.yaml 对缩进有严格的要求！请确保修改参数之后每行开头的缩进仍保持一致！**

#### B.2.3.1 Common 部分参数

此部分用于设置雷达的消息来源，以及是否将结果发布。

```
common:
  msg_source: 1                #雷达数据消息来源类型
  send_packet_ros: false
  send_point_cloud_ros: false
  send_packet_proto: false
  send_point_cloud_proto: false
  pcap_path: /home/robosense/lidar.pca    #播放离线 PCAP 包时的绝对地址
```

#### msg\_source:

- 1 -- 连接在线雷达。更多使用细节请参考在线读取雷达数据发送到 ROS。
- 2 -- 离线解析 ROS 或 ROS2 的 packet 包。更多使用细节请参考 录制 ROS 数据包&离线解析 ROS 数据包。
- 3 -- 离线解析 pcap 包。更多使用细节请参考离线解析 Pcap 包发送到 ROS。
- 4 -- 雷达消息来源为 Protobuf-UDP 的 packet 消息
- 5 -- 雷达消息来源为 Protobuf-UDP 的点云消息。

#### send\_packet\_ros:

true -- 雷达 packet 消息将通过 ROS 或 ROS2 发出，false -- 禁止。

由于雷达 ROS packet 消息为速腾聚创自定义 ROS 消息，因此用户无法直接 echo 话题查看消息具体内容。实际上 packet 主要用于录制离线 ROS 包，因为 packet 的体积小于点云。

**send\_point\_cloud\_ros:**

true -- 雷达点云消息将通过 ROS 或 ROS2 发出, false -- 禁止。

点云消息类型为 ROS 官方定义的点云类型 `sensor_msgs/PointCloud2`, 因此用户可以直接使用 Rviz 查看点云。同时, 用户也可以选择录包时直接录制点云, 但这样做包的体积会非常大, 因此我们建议离线录制 ROS 包时录制 `packet` 消息。

**send\_packet\_proto:**

true -- 雷达 `packet` 消息将通过 Protobuf-UDP 发出, false -- 禁止。

**send\_point\_cloud\_proto:**

true -- 雷达点云消息将通过 Protobuf-UDP 发出, false -- 禁止。

我们建议发送 `packet` 消息而不是点云, 因为点云消息体积过大, 对带宽有较高的要求。

**pcap\_path:**

如果 `msg_source = 3`, 请确保此参数设置为正确的 `pcap` 包的绝对路径。

**B.2.3.2 lidar 部分参数**

本部分需要根据不同的雷达进行设置。

```
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:** 目前支持的雷达型号已在 `sdk` 文件夹 README 中列出。RS-LiDAR-M1 选择类型 **RSM1**。

**frame\_id:** 点云消息的 `frame_id`。

**msop\_port, difop\_port:** 点云的 `msop` 端口号和 `difop` 端口号。若收不到消息, 请优先确认这两个参数是否配置正确。

**start\_angle, end\_angle:** 此参数对 RS-LiDAR-M1 暂未启用，点云消息的起始角度和结束角度，此处设置为软件屏蔽，无法减小每帧点云的体积，只会将区域外的点设置为 NAN 点。起始角和结束角的范围应在 0~360°之间。(起始角可以大于结束角)。

**min\_distance, max\_distance:** 点云显示的最小距离和最大距离，此处设置为软件屏蔽，无法减小每帧点云的体积，只会将区域外的点设置为 NAN 点。

**use\_lidar\_clock:** true -- 使用雷达时间作为消息时间戳；false -- 使用系统时间作为消息时间戳。

### B.2.3.3 多雷达示例

在线连接 2 台 RS-LiDAR-M1 雷达，并发送点云到 ROS。

注意：lidar 部分参数的缩进

```

common:
  msg_source: 1                #使用在线数据消息
  send_packet_ros: false
  send_point_cloud_ros: true   #发出点云 rslidar_points 数据
  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_rcv_packet_topic: /left/rslidar_packets
ros_send_packet_topic: /left/rslidar_packets
ros_send_point_cloud_topic: /left/rslidar_points

proto:
point_cloud_rcv_port: 60024
point_cloud_send_port: 60024
msop_rcv_port: 60025
msop_send_port: 60025
difop_rcv_port: 60026
difop_send_port: 60026
point_cloud_send_ip: 127.0.0.1
packet_send_ip: 127.0.0.1
```

## B.2.4 坐标转换功能

`rslidar_sdk` 提供了内置的坐标变换功能，可以直接输出经过坐标变换后的点云，显著节省了用户对点云进行坐标变换的操作耗时。本节将指导用户如何使用 `rslidar_sdk` 的内置坐标变换功能，输出经过坐标变换后的点云。

### B.2.4.1 依赖介绍

若希望启用坐标变换功能，需要安装以下依赖：

#### ➤ Eigen3

命令安装方式：

```
$sudo apt-get install libeigen3-dev
```

### B.2.4.2 编译

若希望启用坐标变换的功能，在编译程序时需要将 `ENABLE_TRANSFORM` 选项设置为 ON.

#### 1.直接编译

```
$cmake -DENABLE_TRANSFORM=ON ..
```

```
$make -j4
```

#### 2.ROS 编译

```
$catkin_make -DENABLE_TRANSFORM=ON
```

## 2. ROS2 编译

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

### B.2.4.3 坐标转换参数设置

坐标变换参数为 lidar 部分的隐藏参数 x, y, z, roll, pitch ,yaw，此处为参数文件的一个示例，用户可根据实际情况配置。

```
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
```

## 附录 C MEMS Tool 的使用

此附录将介绍怎样使用 `mems_flash` 工具修改雷达的 IP 地址，雷达上位机 IP 地址（在非广播地址下，雷达只会给具有此 IP 的上位机发送数据），获取雷达版本号。

### C.1 使用 MEMS Tool 与雷达建立通信

#### 1. 打开小工具

在 windows 下打开小工具所在的文件夹，运行 `mems_flash.exe` 可执行文件。

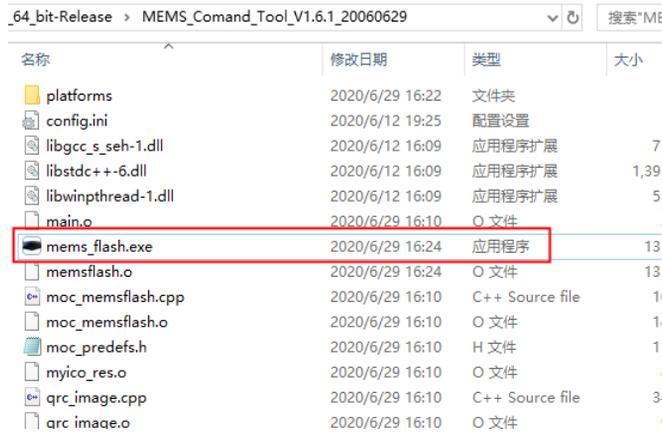


图 C - 1: MEMS Tool 在文件夹中示意图

在 Ubuntu 下，进入小工具文件夹，在文件夹空白处鼠标右键打开选项，点击在终端打开（open in terminal）。在此路径下打开的终端内输入并执行命令：

`./mems_flash.sh`

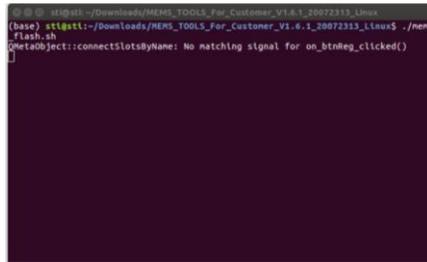


图 C - 2: 打开小工具命令示意图

2. 在小工具的 LidarAddress 输入当前雷达 IP: 192.168.1.102 (默认), MSOP 端口号: 6699 (默认), 之后鼠标左键点击连接按钮 (Connect)。LiDAR 地址和 MSOP 端口号可以通过 Wireshark 或其它抓包工具查看。



图 C - 3: 小工具雷达连接示意图

3. 连接成功后本机 PS 和 PL 版本将会显示。

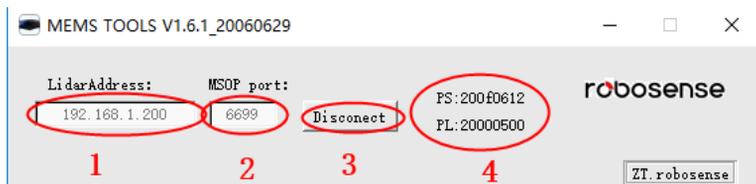


图 C - 4: 固件版本显示示意图

## C.2 修改雷达 IP 及端口号

本节内容将介绍如何修改雷达 IP，上位机目的 IP，MSOP 端口号和 DIFOP 端口号。  
 注意：请不要修改雷达 MAC 地址及 SN。

1. 连接 MEMS Tool 与雷达（按照 C.1 节所述）。
  2. 获取当前机器配置（鼠标左键点击 GetPara）。
- 注意：在每次修改雷达网络配置参数时，都要先获取一次雷达配置参数。
3. 在下图区域改变雷达参数后，鼠标左键点击设置参数（SetPara）。
  4. 鼠标左键点击断开连接（Disconnect）后，断电重新上电检查更改是否成功。

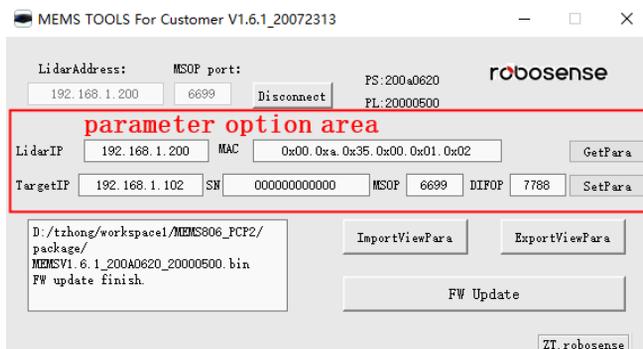


图 C - 5: 修改网络配置示意图

## C.3 固件升级

雷达升级所需的固件命名为 *MEMSV1.6.1\_200A0620\_20000500.bin*。工具会检查固件版本的格式，请不要随意改变固件命名。

1. 连接 MEMS Tool 与雷达（按照 C.1 节所述）。
2. 确认当前 PL 和 PS 版本号，不确定升级固件是否匹配，请咨询 RoboSense 相关技术人员。
3. 鼠标左键点击“FW Update”后，在跳出对话框后选择需要加载升级的 BIN 文件，鼠标左键点击打开（open）开始刷写寄存器。

注意：保持雷达与主机的通信及雷达供电。通信及供电不稳会导致刷写失败，有反厂刷机风险。

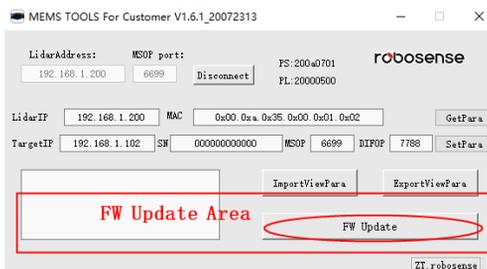


图 C - 6: 固件升级示意图

4. 等待大约 1 分钟直到升级成功

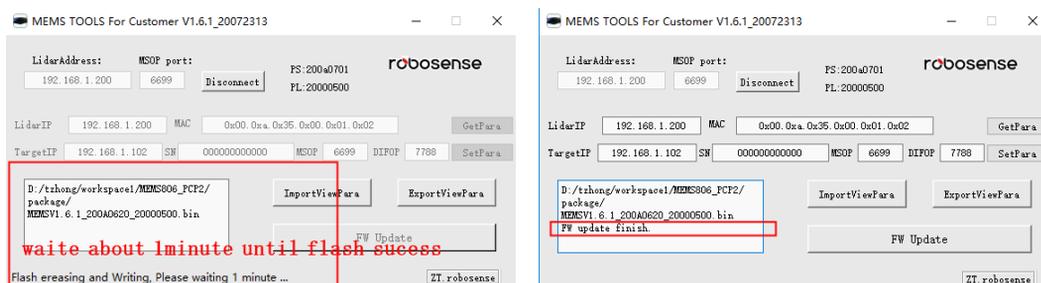
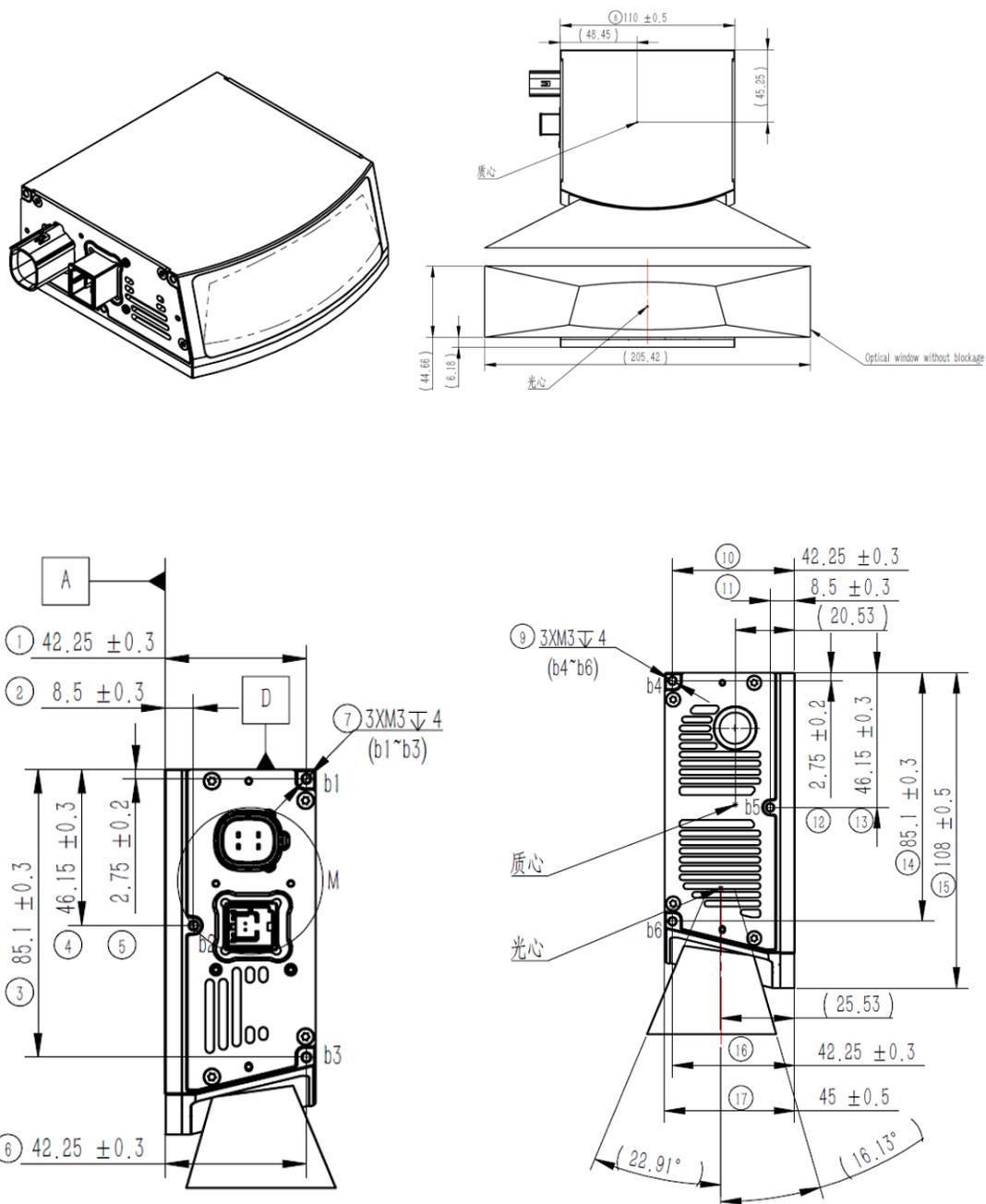


图 C - 7： 固件升级完成示意图

5. 鼠标左键点击断开连接 (Disconnect) 后，断电重新上电检查升级是否成功。

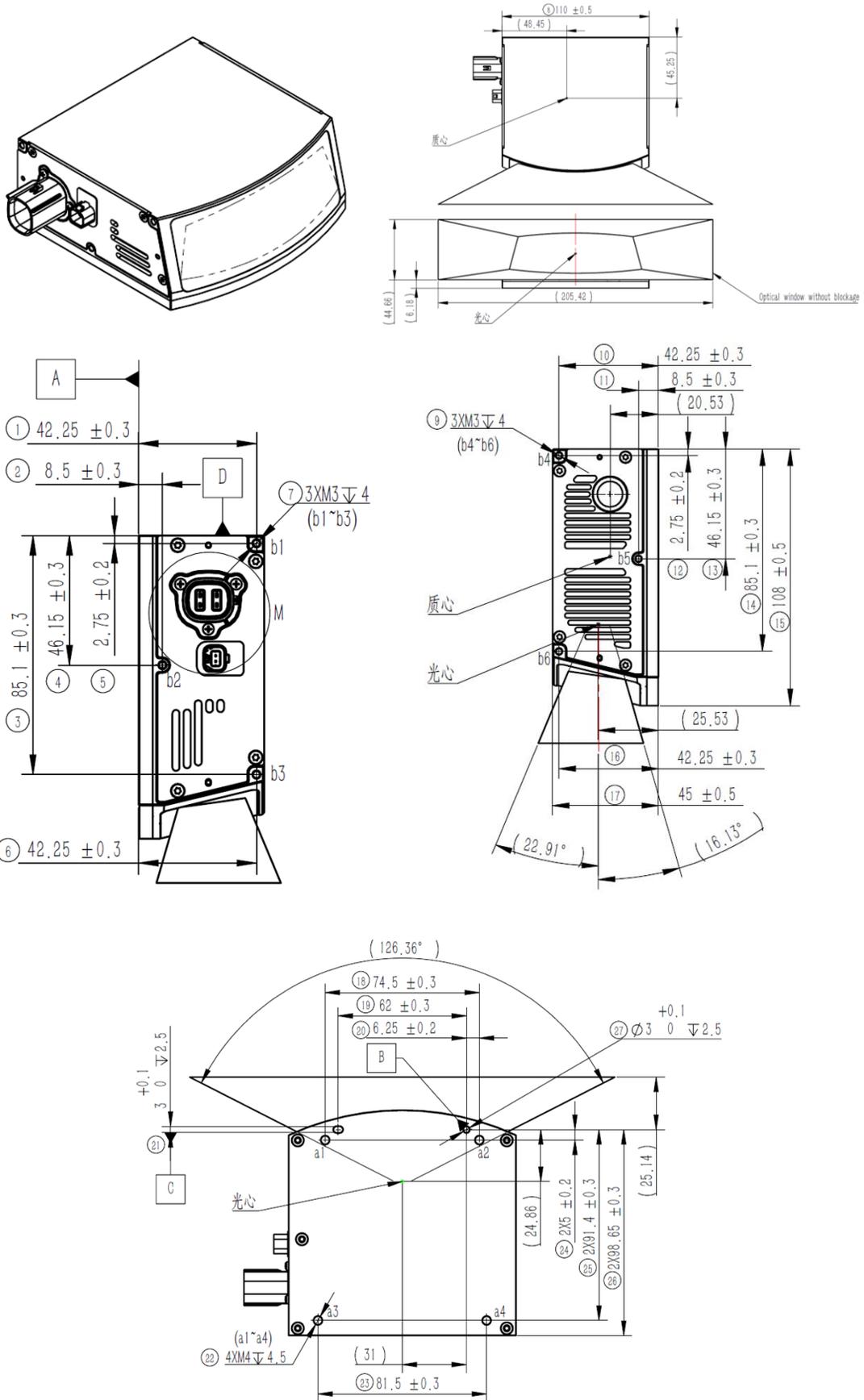
## 附录 D 结构图纸

AN1 接口雷达结构图纸:

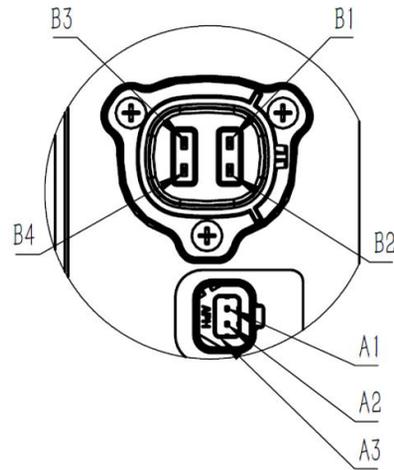




AN2 接口雷达结构图纸:



AN2-Pin 脚定义说明:



| Definition of connector pin |               |                            |
|-----------------------------|---------------|----------------------------|
| Pin Number                  | Signal Name   | Connector Name             |
| B1                          | GND           | LJV C-<br>HSPPSNXS24T-A    |
| B2                          | VBAT          |                            |
| B3                          | WakeuP        |                            |
| B4                          | —             |                            |
| A1                          | 1000Base T1 P | Amphenol<br>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](http://www.robosense.ai)

Tel: 0755-8632-5830

Email: [service@robosense.cn](mailto:service@robosense.cn)