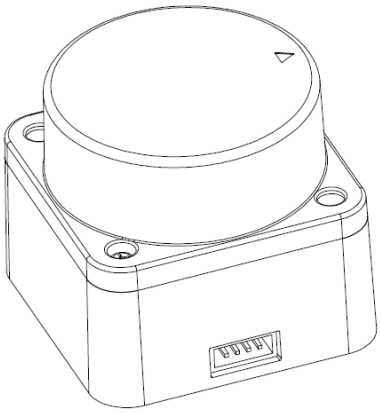
乐动机器人
MOVE SMARTER

LD06 LiDAR

Principle of DTOF

Ultimate small size , low cost

High reliability , long working life



开发手册 v1.0

目录

sales@ldrobot.com

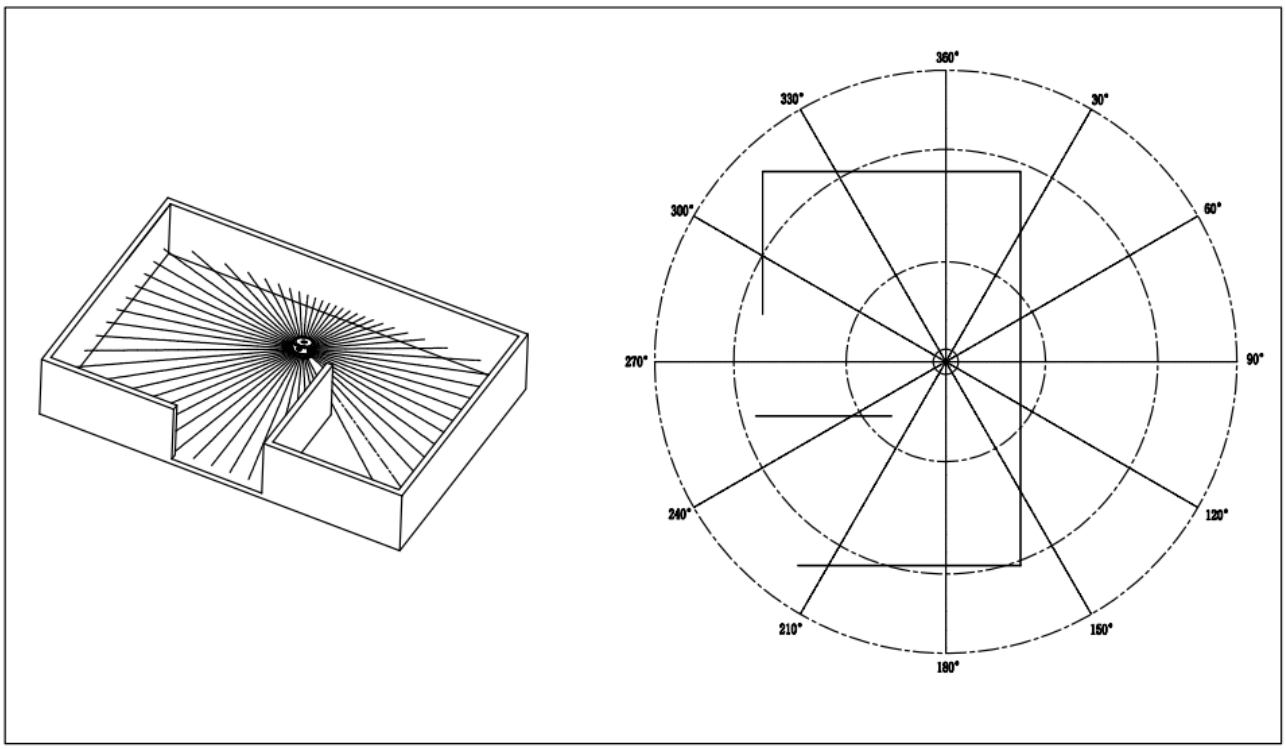
1. 产品描述

LD06 主要由激光测距核心，无线传电单元，无线通讯单元，角度测量单元、电机驱动单元和机械外壳组成。

LD06 测距核心采用 DTOF 技术，可进行每秒 4500 次的测距。每次测距时，LD06 朝前发射出红外激光，激光遇到目标物体后被反射到单光子接收单元。由此，我们获取到了激光的发出时间和单光子接收单元收到激光的时间，两者的时间差即光的飞行时间，飞行时间再结合光速即可解算出距离。

获取到距离数据后，LD06 会融合角度测量单元测量到的角度值组成点云数据，然后通过无线通讯将点云数据发送到外部接口。同时外部接口提供 PWM，使电机驱动单元驱动电机。外部控制单元获取到转速后，通过 PID 算法闭环控制到指定的转速，从而使 LD06 稳定工作。

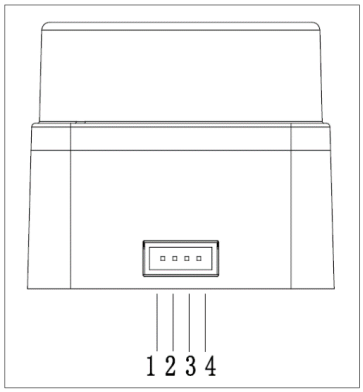
LD06 点云数据形成的环境扫描图意图如下：



sales@ldrobot.com

2. 通讯接口

LD06 使用 ZH1.5T-4P 1.5mm 连接器与外部系统连接，实现供电和数据接收，具体接口定义和参数要求见下图/表：



序号	信号名	类型	描述	最小值	典型值	最大值
1	Tx	输出	雷达数据输出	0V	3.3V	3.5V
2	PWM	输入	电机控制信号	0V	-	3.3V
3	GND	供电	电源负极	-	0V	-
4	P5V	供电	电源正极	4.5V	5V	5.5V

LD06 内部具有可无级调速的电机驱动器，可通过接口中的 PWM 信号控制电机的启、停和转速。由于每个产品电机的个体差异，占空比设置为典型值时实际转速可能会有差异，如要精确控制电机转速，需根据接收数据中的转速信息进行闭环控制。

LD06 的数据通讯采用标准异步串口(UART)单向发送，其传输参数如下表所示：

波特率	数据长度	停止位	奇偶校验位	流控制
230400	8 Bits	1	无	无

3. 通讯协议

3.1. 数据包格式

LD06 采用单向通讯，稳定工作后，即开始发送测量数据包，不需要发送任何指令。测量数据包格式如下图所示。

起始符	数据长度	雷达转速		起始角度		数据	结束角度		时间戳		CRC 校验
54H	1 Byte	LSB	MSB	LSB	MSB	……	LSB	MSB	LSB	MSB	1 Byte

- 起始符：长度 1 Byte，值固定为 0x54，表示数据包的开始；

sales@ldrobot.com

- 数据长度：长度 1 Byte，高三位预留，低五位表示一个包的测量点数，目前固定为 12；
- 雷达转速：长度 2 Byte，单位为度每秒；
- 起始角度：长度 2 Byte，单位为 0.01 度；
- 数据：一个测量数据长度为 3 个字节，详细解析请见下一小节；
- 结束角度：长度 2 Byte，单位为 0.01 度；
- 时间戳：长度 2 Byte，单位为 ms，最大为 30000，到达 30000 会重新计数；
- CRC 校验：前面所有数据的校验和；

数据结构参考如下：

```
#define ANGLE_PER_FRAME 12
#define HEADER 0x54
typedef struct   __attribute__((packed))
{
    uint8_t             header;
    uint8_t             ver_len;
    uint16_t            speed;
    uint16_t            start_angle;
    LidarPointStructDef point[POINT_PER_PACK];
    uint16_t            end_angle;
    uint16_t            timestamp;
    uint8_t             crc8;
}LiDARFrameTypeDef;
```

CRC 校验计算方式如下：

```
static const uint8_t CrcTable[256] =
{
    0x00, 0x4d, 0x9a, 0xd7, 0x79, 0x34, 0xe3,
    0xae, 0xf2, 0xbf, 0x68, 0x25, 0x8b, 0xc6, 0x11, 0x5c, 0xa9, 0xe4, 0x33,
    0x7e, 0xd0, 0x9d, 0x4a, 0x07, 0x5b, 0x16, 0xc1, 0x8c, 0x22, 0x6f, 0xb8,
    0xf5, 0x1f, 0x52, 0x85, 0xc8, 0x66, 0x2b, 0xfc, 0xb1, 0xed, 0xa0, 0x77,
    0x3a, 0x94, 0xd9, 0x0e, 0x43, 0xb6, 0xfb, 0x2c, 0x61, 0xcf, 0x82, 0x55,
    0x18, 0x44, 0x09, 0xde, 0x93, 0x3d, 0x70, 0xa7, 0xea, 0x3e, 0x73, 0xa4,
    0xe9, 0x47, 0x0a, 0xdd, 0x90, 0xcc, 0x81, 0x56, 0x1b, 0xb5, 0xf8, 0x2f,
    0x62, 0x97, 0xda, 0x0d, 0x40, 0xee, 0xa3, 0x74, 0x39, 0x65, 0x28, 0xff,
    0xb2, 0x1c, 0x51, 0x86, 0xcb, 0x21, 0x6c, 0xbb, 0xf6, 0x58, 0x15, 0xc2,
```


sales@ldrobot.com

```
    0x8f, 0xd3, 0x9e, 0x49, 0x04, 0xaa, 0xe7, 0x30, 0x7d, 0x88, 0xc5, 0x12,
    0x5f, 0xf1, 0xbc, 0x6b, 0x26, 0x7a, 0x37, 0xe0, 0xad, 0x03, 0x4e, 0x99,
    0xd4, 0x7c, 0x31, 0xe6, 0xab, 0x05, 0x48, 0x9f, 0xd2, 0x8e, 0xc3, 0x14,
    0x59, 0xf7, 0xba, 0x6d, 0x20, 0xd5, 0x98, 0x4f, 0x02, 0xac, 0xe1, 0x36,
    0x7b, 0x27, 0x6a, 0xbd, 0xf0, 0x5e, 0x13, 0xc4, 0x89, 0x63, 0x2e, 0xf9,
    0xb4, 0x1a, 0x57, 0x80, 0xcd, 0x91, 0xdc, 0x0b, 0x46, 0xe8, 0xa5, 0x72,
    0x3f, 0xca, 0x87, 0x50, 0x1d, 0xb3, 0xfe, 0x29, 0x64, 0x38, 0x75, 0xa2,
    0xef, 0x41, 0x0c, 0xdb, 0x96, 0x42, 0x0f, 0xd8, 0x95, 0x3b, 0x76, 0xa1,
    0xec, 0xb0, 0xfd, 0x2a, 0x67, 0xc9, 0x84, 0x53, 0x1e, 0xeb, 0xa6, 0x71,
    0x3c, 0x92, 0xdf, 0x08, 0x45, 0x19, 0x54, 0x83, 0xce, 0x60, 0x2d, 0xfa,
    0xb7, 0x5d, 0x10, 0xc7, 0x8a, 0x24, 0x69, 0xbe, 0xf3, 0xaf, 0xe2, 0x35,
    0x78, 0xd6, 0x9b, 0x4c, 0x01, 0xf4, 0xb9, 0x6e, 0x23, 0x8d, 0xc0, 0x17,
    0x5a, 0x06, 0x4b, 0x9c, 0xd1, 0x7f, 0x32, 0xe5, 0xa8
};
uint8_t CalCRC8(uint8_t *p, uint8_t len)
{
    uint8_t crc = 0;
    uint16_t i;

    for (i = 0; i < len; i++)
    {
        crc = CrcTable[(crc ^ *p++) & 0xff];
    }

    return crc;
}
```

3.2. 测量数据解析

每个测量数据点由 2 个字节长度的距离值和 1 个字节长度的置信度值组成，如下图所示。

起始符	数据长度	雷达转速		起始角度		数据	结束角度		时间戳		CRC 校验
54H	2CH	LSB	MSB	LSB	MSB	……	LSB	MSB	LSB	MSB	1Byte

测量点 1			测量点 2			…	测量点 n		
距离值		置信度	距离值		置信度		距离值		置信度
LSB	MSB	1 Byte	LSB	MSB	1 Byte	…	LSB	MSB	1 Byte

距离值的单位为 mm。置信度反映的是光反射强度，强度越高，置信度值越大；强度越低，置信度值越小。6m 以内的白色物体，置信度典型值在 200 左右。

sales@ldrobot.com

每个点的角度值是通过起始角度和结束角度线性插值得来，其角度计算方法如下：

step = (end_angle – start_angle)/(len – 1);

angle = start_angle + step*i;

其中 len 为数据包的长度，i 的取值范围为[0 , len)。

3.3. 参考示例

假设我们收到了如下所示的一段数据

54 2C 68 08 AB 7E E0 00 E4 DC 00 E2 D9 00 E5 D5 00 E3 D3 00 E4 D0 00 E9 CD 00 E4 CA 00 E2 C7 00 E9 C5 00 E5 C2 00 E5 C0 00 E5 BE 82 3A 1A 50

我们对其解析如下：

起始符	数据长度	雷达转速		起始角度		数据	结束角度		时间戳		CRC 校验
54H	2CH	68H	08H	AB	7E	……	BE	82	3A	1A	50

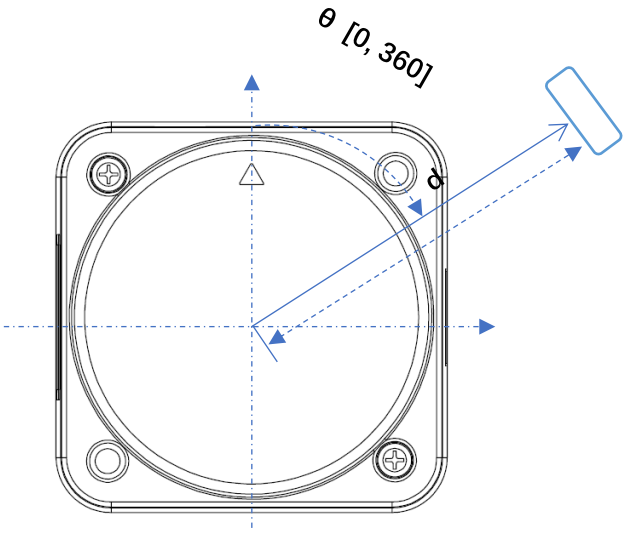
0868H = 2152°/s　　7EABH=32427 即 324.27°　　82BEH= 33470 即 334.7°

测量点 1			测量点 2			…	测量点 12		
距离值		置信度	距离值		置信度		距离值		置信度
E0	00	E4	DC	00	E2	…	B0	00	EA
00E0H= 224 mm 置信度 228			00DCH=220 mm 置信度 226				00B0H=176 mm 置信度 234		

sales@ldrobot.com

4. 坐标系

LD06 使用左手坐标系，旋转中心为坐标原点，传感器的正前方定义为零度方向，旋转角度沿着顺时针方向增大。具体如下图所示：



需要注意的是，雷达传输过来的原始测量数据并不是使用该坐标系的，需要进行换算。换算过程，请参考 SDK 中的 Transform()函数：

```
Points2D SlTransform::Transform(const Points2D &data)
{
    Points2D tmp2;
    for (auto n : data)
    {
        /*Filter out invalid data*/
        if (n.distance == 0)
        {
            continue;
        }
        /*transfer the origin to the center of lidar circle*/
        /*The default direction of radar rotation is clockwise*/
        /*transfer to the right-hand coordinate system*/
        float right_hand = (360.f - n.angle);
        double x = n.distance + offset_x;
        double y = n.distance * 0.11923 + offset_y;
        double d = sqrt(x * x + y * y);
```


sales@ldrobot.com

```
            double shift = atan(y / x) * 180.f / 3.14159;
            /*Choose whether to use the right-hand system according to the flag*/
            double angle;
            if (to_right_hand)
                angle = right_hand + shift;
            else
                angle = n.angle - shift;

            if (angle > 360)
            {
                angle -= 360;
            }
            if (angle < 0)
            {
                angle += 360;
            }
            tmp2.push_back(PointData(angle, d, n.confidence,x,y));
        }
        return tmp2;
}
```


sales@ldrobot.com

5. ROS SDK 使用说明

ROS（Robot Operating System，简称“ROS”）是一个适用于机器人的开源的元操作系统。它提供了操作系统应有的服务，包括硬件抽象，底层设备控制，常用函数的实现，进程间消息传递，以及包管理。它也提供用于获取、编译、编写、和跨计算机运行代码所需的工具和库函数。ROS 各个版本的安装步骤请参考 ROS 官方网址。*http://wiki.ros.org/kinetic/Installation*

本手册使用的是 ubuntu16.04 系统，安装的 ROS 版本为 kinetic。

5.1. 设置权限

首先，将雷达接上我们的转接模块(CP2102 串口转接模块)，将模块接入电脑。然后，在 ubuntu 系统下打开一个终端，输入 `ls /dev/ttyUSB*` 查看串口设备是否接入。若检测到串口设备，则使用 `sudo chmod 777 /dev/ttyUSB*` 命令给其赋予最高权限，即给文件拥有者、群组、其他用户读写和执行权限。

```
pi@pi:~$ ls /dev/ttyUSB*
/dev/ttyUSB0
pi@pi:~$ sudo chmod 777 /dev/ttyUSB*
pi@pi:~$
```

5.2. 编译 sdk

进入 sdk_ld_sllidar_ros 文件夹，然后使用 catkin_make 编译源文件，编译过程如下图所示。

```
pi@pi: ~/sdk_ld_sllidar_ros
pi@pi:~$ cd sdk_ld_sllidar_ros/
pi@pi:~/sdk_ld_sllidar_ros$ catkin_make
Base path: /home/pi/sdk_ld_sllidar_ros
Source space: /home/pi/sdk_ld_sllidar_ros/src
Build space: /home/pi/sdk_ld_sllidar_ros/build
Devel space: /home/pi/sdk_ld_sllidar_ros/devel
Install space: /home/pi/sdk_ld_sllidar_ros/install
Creating symlink "/home/pi/sdk_ld_sllidar_ros/src/CMakeLists.txt" pointing to "/
opt/ros/kinetic/share/catkin/cmake/toplevel.cmake"
####
#### Running command: "cmake /home/pi/sdk_ld_sllidar_ros/src -DCATKIN_DEVEL_PREF
IX=/home/pi/sdk_ld_sllidar_ros/devel -DCMAKE_INSTALL_PREFIX=/home/pi/sdk_ld_slli
dar_ros/install -G Unix Makefiles" in "/home/pi/sdk_ld_sllidar_ros/build"
####
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
```


sales@ldrobot.com

编译成功后会显示如下界面：

```
Scanning dependencies of target ldlidar
[ 16%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/main.cpp.o
[ 33%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/transform.cpp.o
[ 50%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/slbf.cpp.o
[ 66%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/cmd_interface_linux.cpp.o
[ 83%] Building CXX object ldlidar/CMakeFiles/ldlidar.dir/src/lipkg.cpp.o
[100%] Linking CXX executable /home/pi/sdk_ld_sllidar_ros/devel/lib/ldlidar/ldlidar
[100%] Built target ldlidar
```

5.3. 运行程序

编译完成后需要将编译的文件加入环境变量，命令为 source devel/setup.bash，该命令是临时给终端加入环境变量，意味着您如果另外打开新的终端，也需要进入 sdk_ld_sllidar_ros 路径下之执行添加环境变量命令。添加环境变量后，roslaunch 命令则可以找到相应的 ros 包和 launch 文件，运行命令为 roslaunch ldlidar ld06.launch。

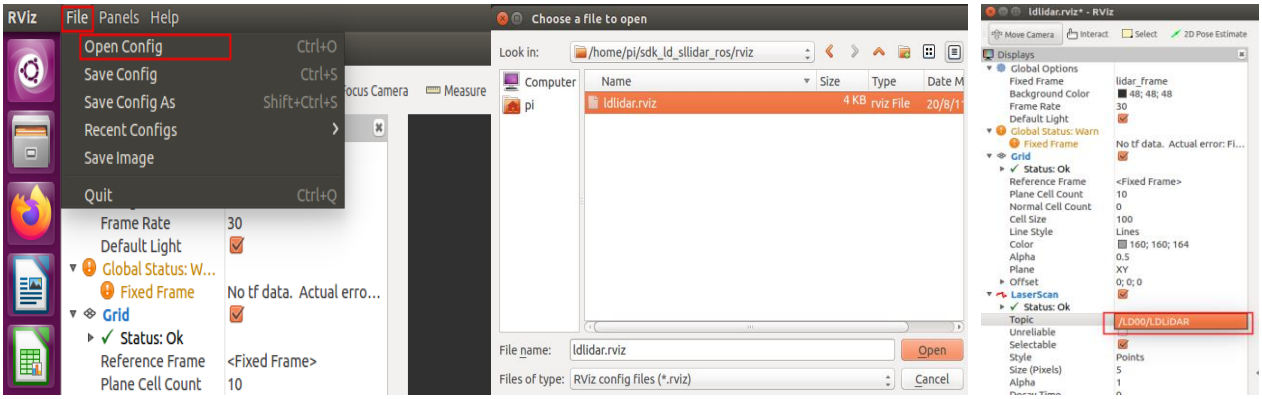
```
pi@pi:~/sdk_ld_sllidar_ros$ source devel/setup.bash
pi@pi:~/sdk_ld_sllidar_ros$ roslaunch ldlidar ld00.launch
... logging to /home/pi/.ros/log/0fad1b9c-de11-11ea-b9bb-000c29fc5655/roslaunch-pi-3191.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://pi:42991/

SUMMARY
========

PARAMETERS
 * /product: LD00
 * /rosdistro: kinetic
 * /rosversion: 1.12.14

NODES
  /
    LD00 (ldlidar/ldlidar)

auto-starting new master
process[master]: started with pid [3201]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 0fad1b9c-de11-11ea-b9bb-000c29fc5655
process[rosout-1]: started with pid [3214]
started core service [/rosout]
process[LD00-2]: started with pid [3222]
/dev/ttyUSB0    CP2102 USB to UART Bridge Controller
FOUND LiDAR_LD00
```

启动成功后，将看到如上红色方框信息。程序输出 FOUND LiDAR_LD06，则表明找到 LD06 设备并且 LD06 的 ros 结点启动成功。

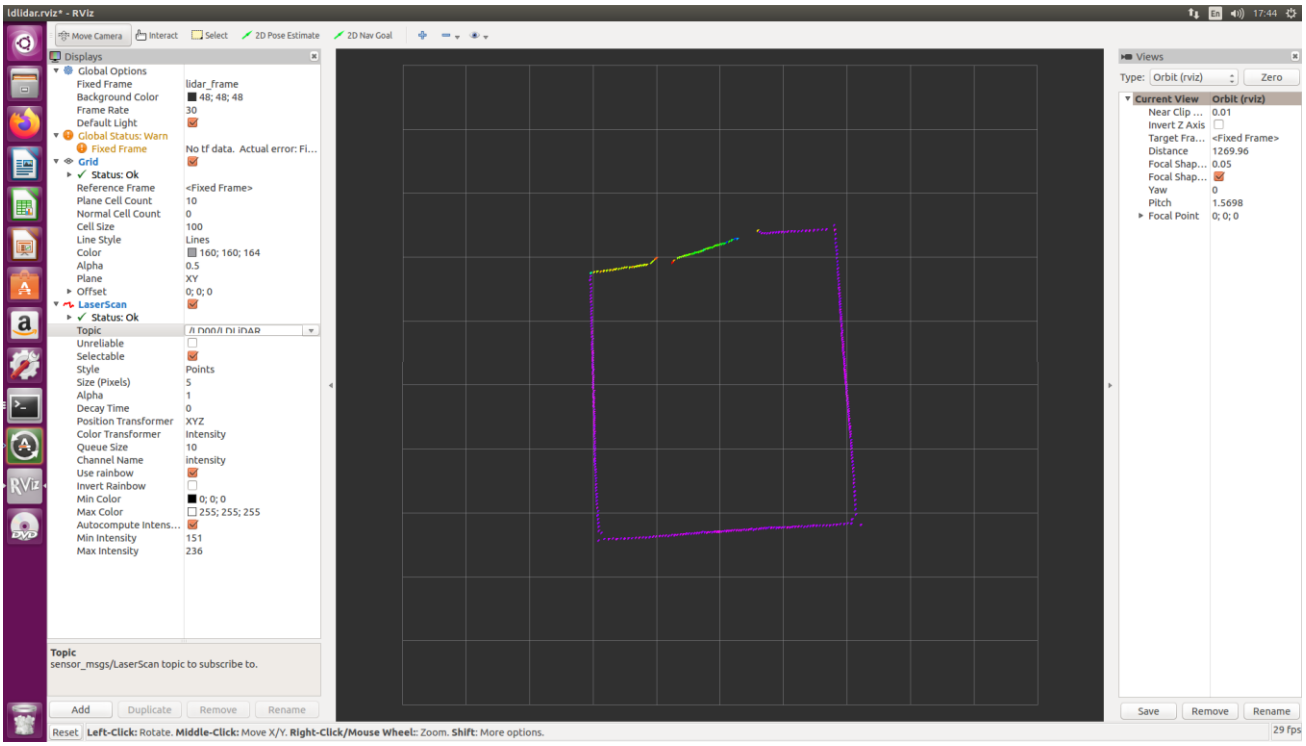
sales@ldrobot.com

5.4. rviz 显示

rviz 是 ROS 下一款常用的 3D 可视化工具，雷达数据可以在其中展现。在 roslaunch 运行成功后，打开新的终端，输入 `rosrun rviz rviz`,依次点击 file->open->Config ,然后选择 sdk_ld_sllidar_rosrviz/ldlidar.rviz 文件，打开配置文件 ldlidar.rviz 后，点击 LaserScan 的 Topic 中选择/LD06/LDLiDAR。

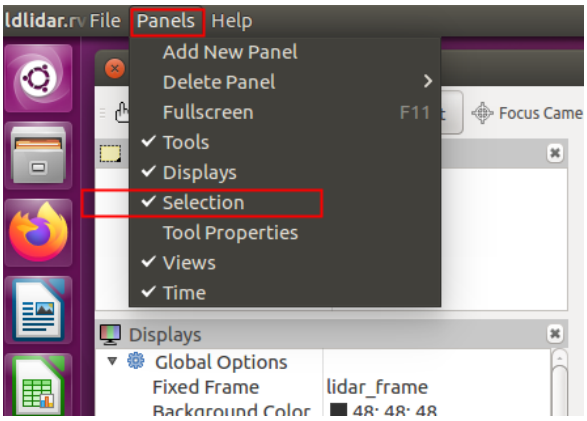


由此，雷达图在 rivz 可正常显示。

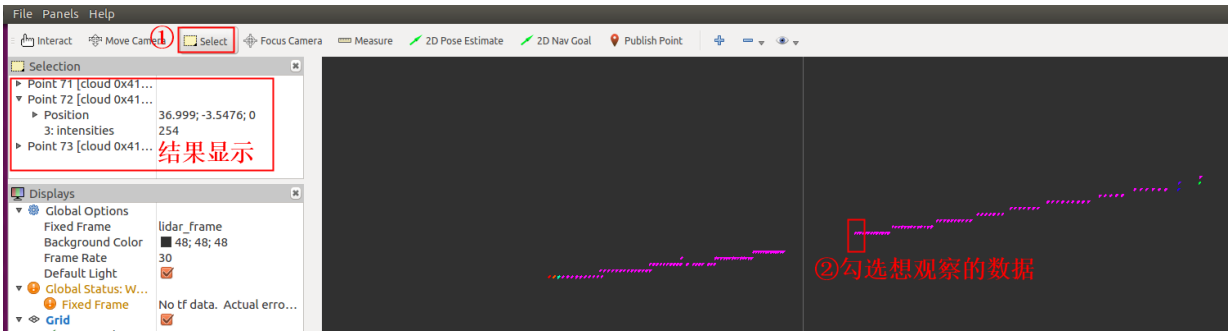


如果您需要观察雷达图中特定点的数据，您可以依次点击 Panel->Selection。

sales@ldrobot.com



然后点击 select，勾选想要观察的数据，左上角窗口会显示当前点的坐标值(Position)和置信度(intensities)信息。



sales@ldrobot.com

6. 修订记录

版本	修订日期	修订内容
1.0	2020-09-01	初始创建

sales@ldrobot.com