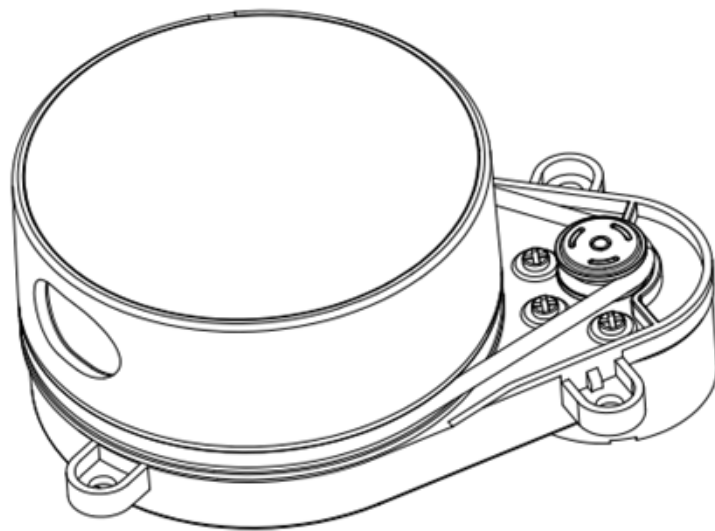


Delta-2D 雷达 SDK 使用简介

[型号: Delta-2D 2k/s、8m]

版 本：V1.0



目 录

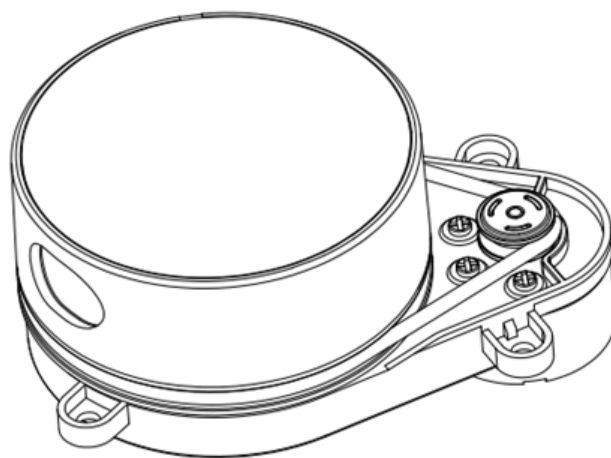
一 . SDK 简介	3
二 . SDK 编译调试之前准备工作	4
2.1 按雷达使用手册正确连线	4
2.2 安装 usb 转串口驱动	4
三 . Windows 平台 SDK	4
3.1 开发环境	4
3.2 windows sdk 解决方案	4
3.3 解决方案路径	5
3.4 解决方案工程图	5
3.5 获取扫描完整一圈点的初始信息（没有计算出角度）函数	5
3.6 建点云图项目 frame_grabber	7
3.7 雷达驱动简单运用项目 ultra_simple	8
四 . Linux 平台 SDK	9
4.1 开发环境	9
4.2 目录结构	9
4.3 sdk 提供的雷达类接口函数	10
4.4 实例运用	10
五 . Ros 平台 SDK	11
5.1 开发环境	11

5.2 目录结构.....	11
5.3 函数接口参考.....	12
5.4 启动发布者和订阅者例程.....	12
5.5 启动 rviz 建立点云图.....	13
六 . Mcu 平台 DEMO.....	13
6.1 开发环境: Stm32f103cb.....	13
6.2 工程.....	13

一 . SDK 简介

Delta-2D SDK 提供了 windows、ros、Linux、mcu 平台基础开发代码；基本内容包含接收雷达数据、解析雷达数据、将解析的雷达数据整理成扫描完整一圈的信息，存储在数组中供客户应用程序中调用。客户根据实际需要可以直接利用提供的 sdk 源码，也可以自己编写 sdk 接收解析（参考 sdk 源码和雷达通讯协议）。

雷达是旋转测量一周，扫描得到周围一圈均匀分布点的信息（点的角度和距离）。sdk 就是接收解析数据，得到每一圈点的信息。一圈 360° 被平均分为 16 帧上报扫描信息（见雷达通信协议文档中命令字列表）帧，每帧给出该帧的初始角度和终止角度。16 帧数据加起来是完整一圈，一圈的总点数 = $16 \times \text{每帧的点数}$ ；每帧的总点数根据扫描信息帧计算距离个数可以得到（距离个数 = 总点数）。
每帧数据点的信息（角度和距离）：一帧中第 N 个点的距离是扫描信息帧中 N 距离值，那一帧中第 N 个点距离对应的角度 = 此帧起始角度 + $(\text{终止角度} - \text{起始角度}) / (\text{每帧的总点数} - 1) * (N - 1)$ ，这样一帧点信息（角度和距离）都有了。



二 . SDK 编译调试之前准备工作

2.1 按雷达使用手册正确连线

2.2 安装 usb 转串口驱动

雷达小板用的 cp2102, 可到官网下载驱动：
<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>；根据自己的平台下载相应的驱动文件，Linux 平台可根据官网上的文档安装

Legacy OS software and driver package download links and support information >

Download for Windows 10 Universal (v10.1.4)

Platform	Software	Release Notes
Windows 10 Universal	Download VCP (2.3 MB)	Download VCP Revision History

Download for Windows 7/8/8.1 (v6.7.6) 直接下载

Platform	Software	Release Notes
Windows 7/8/8.1	Download VCP (5.3 MB) (Default)	Download VCP Revision History
Windows 7/8/8.1	Download VCP with Serial Enumeration (5.3 MB) Learn More >	Download VCP Revision History

Download for Linux

Platform	Software	Release Notes
Linux 3.x.x and 4.x.x	Download VCP (10.0 KB)	Download Linux 3.x.x and 4.x.x VCP Revision History
Linux 2.6.x	Download VCP (10.2 KB)	Download Linux 2.6.x VCP Revision History

安装说明

*Note: The Linux 3.x.x and 4.x.x version of the driver is maintained in the current Linux 3.x.x and 4.x.x tree at www.kernel.org.

三 . Windows 平台 SDK

3.1 开发环境

开发语言：c++

编译环境：vs2017

3.2 windows sdk 解决方案

Windows sdk 是一个解决方案 sdk_and_demo, 包含：雷达驱动项目 rslidar_driver、建点云图项目 frame_grabber、雷达驱动简单运用项目

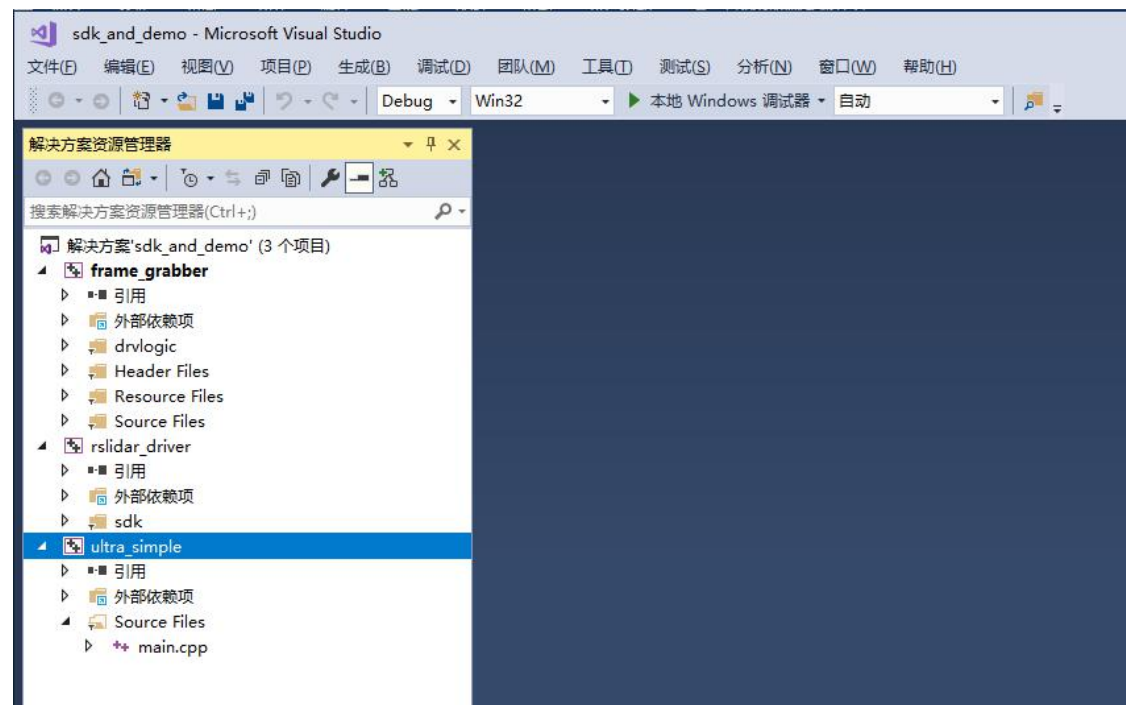
ultra_simple 3 个项目。

3.3 解决方案路径

地磁盘 (E:) > 项目 > Lidar资料 > Delta-2D > Delta-2D_Windows_Demo > sdk > workspaces > vc10

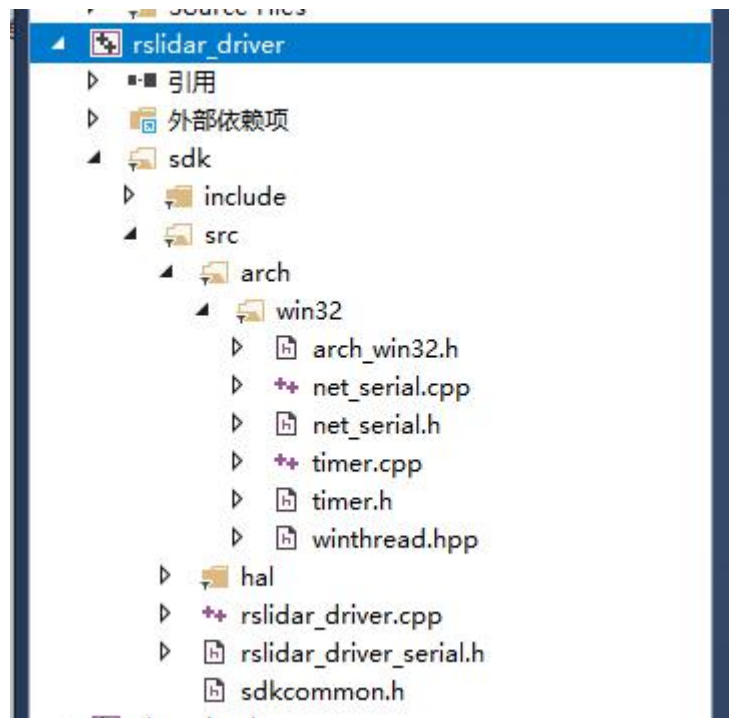
名称	修改日期	类型	大小
frame_grabber	2020/12/9 14:57	文件夹	
rslidar_driver	2020/12/9 14:57	文件夹	
ultra_simple	2020/12/9 14:57	文件夹	
sdk_and_demo.sln	2018/12/26 10:47	Visual Studio Sol...	3 KB

3.4 解决方案工程图



3.5 雷达驱动项目 rslidar_driver

提供打开串口，串口接收，串口解析、设置转速、雷达复位、开始扫描、停止扫描、判读扫描结果等接口函数。



3.5.1 打开配置串口函数：

```
u_result RSlidarDriverSerialImpl::connect(const char * port_path,
_u32 baudrate, _u32 flag)
```

参数：port_path 配置串口端口号 baudrate：波特率

返回值：RESULT_ALREADY_DONE 串口已打开

RESULT_INVALID_DATA 打开失败

3.5.2 关闭串口函数

```
void RSlidarDriverSerialImpl::disconnect() :
```

3.5.3 复位雷达函数

```
u_result RSlidarDriverSerialImpl::resetlidar(_u32 timeout)
```

参数：timeout 接收雷达应答超时时间

3.5.4 获取扫描完整一圈点的初始信息（没有计算出角度）函数

```
u_result RSlidarDriverSerialImpl::grabScanData(LIDAR_MEASURE_INFO_T
```

* nodebuffer, size_t & count, _u32 timeout)

参数：nodebuffer 存储扫描一圈点的初始信息 count 扫描一圈的总点数

Timeout 扫描一圈超时时间

返回值：RESULT_OK 表示成功扫描完整一圈

RESULT_OPERATION_FAIL 表示扫描失败

RESULT_OPERATION_TIMEOUT 表示扫描超时

3.5.5 将一圈点的初始信息转化成完整点信息(距离、角度)

u_result

RSlidarDriverSerialImpl::ascendScanData(LIDAR_MEASURE_INFO_T *

nodebuffer, size_t count)

参数：nodebuffer 存储扫描一圈点的完整信息 count 扫描一圈的总点数

Timeout 扫描一圈超时时间

RESULT_OK 打开成功

3.6 建点云图项目 frame_grabber

建立点云图，可以获取当前转速，每个点角度和距离

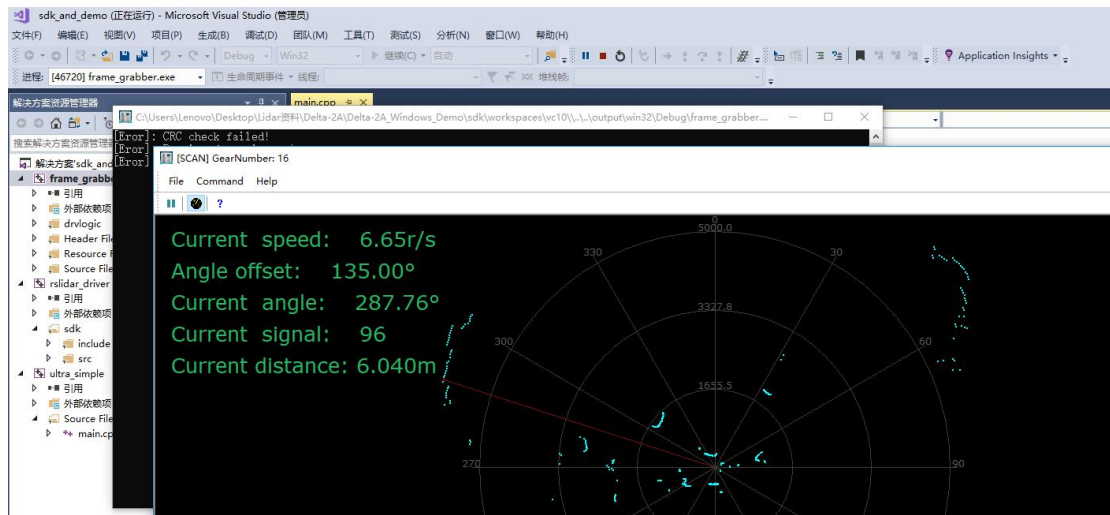


3.6.1 开发环境：c++

3.6.2 MFC 创建的界面

3.6.3 该项目实例依赖 rslidar_driver 项目中的接口

3.6.4 编译调试结果



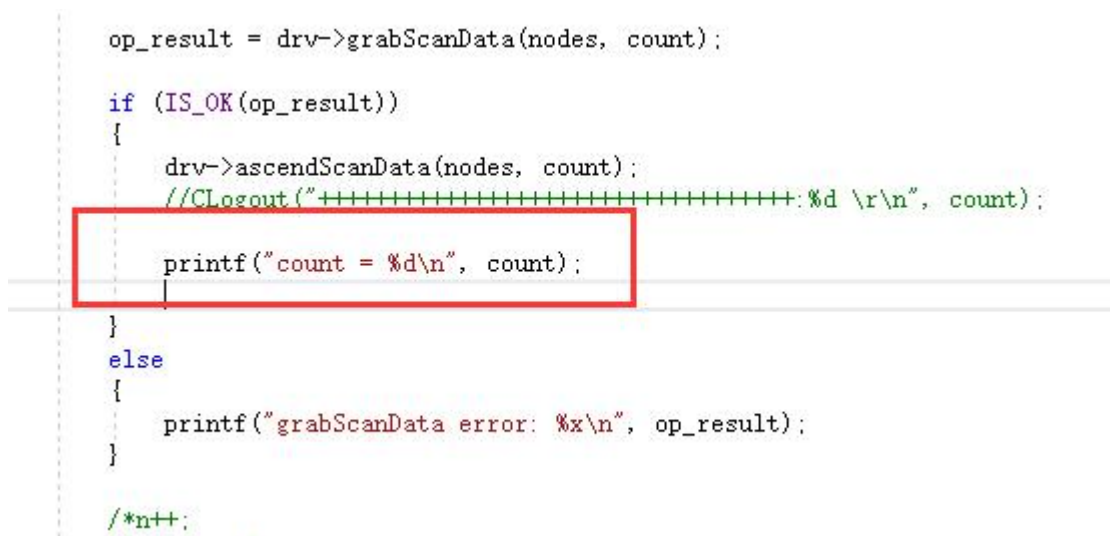
3.7 雷达驱动简单运用项目 ultra_simple

可以参考该项目，运用在实际的项目中



3.7.1 该项目实例依赖 rslidar_driver 项目中的接口，帮助客户调用这些接口在自己项目中运用。

3.7.2 打印信息：扫描了一圈后 打印扫描一圈的总点数。



3.7.3 编译调试

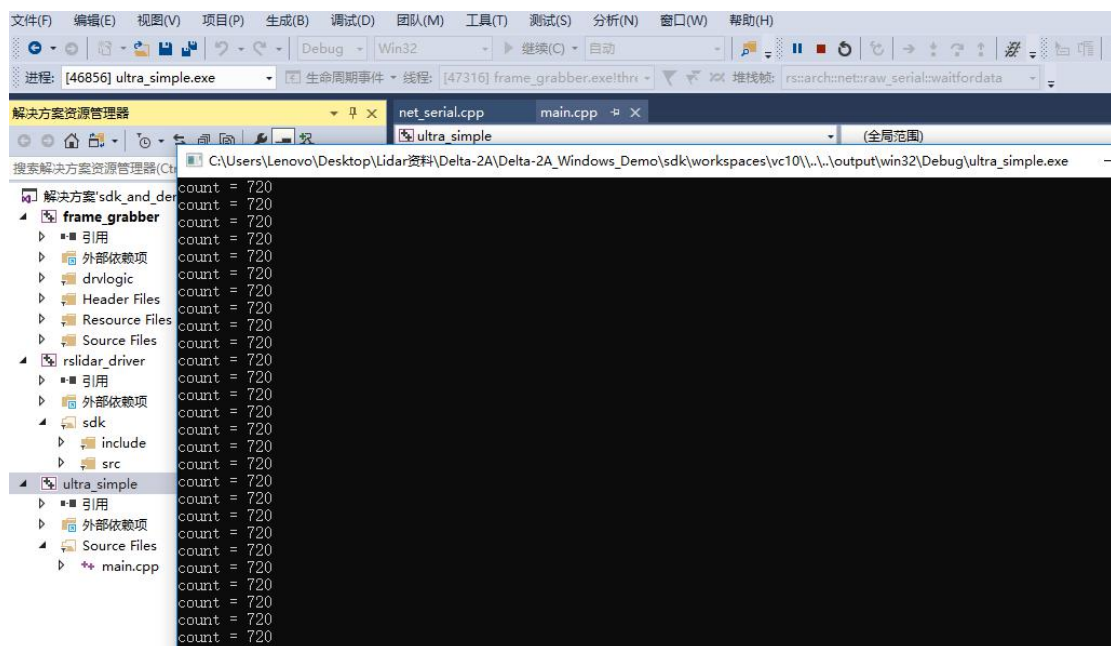
3.7.3.1 在 main.c 中修改成正确 COM 端口

```

if (!opt_com_path) {
#ifdef _WIN32
    // use default com port
    opt_com_path = "\\\\.\\com5";
#else
    opt_com_path = "/dev/ttyUSB0";
#endif
}

```

3.7.3.2 调试运行结果



四 . Linux 平台 SDK

4.1 开发环境

4.1.1 系统： Ubuntu16.0 、 c++

4.1.2 编译工具：Cmake

4.2 目录结构

此电脑 > 本地磁盘 (E:) > 项目 > Lidar资料 > Delta-2D > Delta_2D_linux				
名称	修改日期	类型	大小	
app	2020/12/9 14:57	文件夹		
src	2020/12/9 14:57	文件夹		
CMakeLists	2020/10/16 17:20	文本文档	1 KB	
READ_ME	2020/10/16 17:18	文件	1 KB	

4.2.1 src:存放串口类 , 时间类 , 雷达接口类(解析、开始扫描、停止等接口)

4.2.2 app:运用雷达类的接口，实现一个调用实例

4.2.3 README:编译过程

4.2.4 CMakeList.txt: cmake 配置

4.3 sdk 提供的雷达类接口函数：

4.3.1 初始化串口函数

```
bool C3iroboticsLidar::initilize(CDeviceConnection *device_connect)
```

参数：device_connect 串口类指针 可以设置串口端口 波特率

4.3.2 接收和解析串口数据函数

TLidarGrabResult C3iroboticsLidar::getScanData()

返回值：LIDAR_GRAB_ING：正在扫描

LIDAR GRAB SUCCESS : 扫描了一圈 ok

LIDAR GRAB ERRO : 扫描出错

4.4 实例运用：在 app/node.c 中利用雷达接口实现雷达简单运用，打印雷达

每扫描一圈的总点数，可根据需要修改。

4.4.1 编译：按 READ ME 提示编译

4.4.2 运行结果：打印扫描一圈的总点数

[illegible]

五 . Ros 平台 SDK

5.1 开发环境

Ubuntu16.04LTS + kinetic

5.2 目录结构

此电脑 > 本地磁盘 (E:) > 项目 > Lidar资料 > Delta-2D > Delta_2D_ros

名称	修改日期	类型	大小
launch	2020/12/9 14:57	文件夹	
rviz	2020/12/9 14:57	文件夹	
sdk	2020/12/9 14:57	文件夹	
src	2020/12/9 14:57	文件夹	
CMakeLists	2018/9/4 7:25	文本文档	1 KB
package	2018/9/4 7:27	XML 文档	1 KB
README.md	2019/2/24 11:02	MD 文件	1 KB

5.2.1 src: node.cpp 发布者节点代码 ; client.cpp 订阅者节点代码。

5.2.2 sdk: 存放串口类 , 时间类 , 雷达接口类 (解析、开始扫描、停止等接口) 。

5.2.3 launch: delta_lidar.launch 启动发布者节点配置
view_delta_lidar.launch 启动 rviz 节点配置。

5.2.4 rviz:创建点云图配置。

5.2.5 README.md:编译过程 启动节点过程。

5.3 函数接口参考 : Linux 函数接口解析

5.4 启动发布者和订阅者例程 :通过 ros 系统中激光雷达数据结构 (如下图) 通讯 , 实现雷达数据简单应用。可参考 , 应用到自己的方案中。

原 ros中激光雷达的消息类型 (sensor_msgs/LaserScan Message) 说明

2018年07月11日 10:46:07 ultimate1212 阅读数: 1474

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/ultimate1212/article/details/80995545>

最近在做一些视觉和激光数据融合的项目，但是对激光数据的结构不是太了解，因此查了很多相关的内容，记录以下。

下图是在<http://wiki.ros.org>中截取的图片：

File: **sensor_msgs/LaserScan.msg**

Raw Message Definition

```
# Single scan from a planar laser range-finder
#
# If you have another ranging device with different behavior (e.g. a sonar
# array), please find or create a different message, since applications
# will make fairly laser-specific assumptions about this data

Header header          # timestamp in the header is the acquisition time of
                        # the first ray in the scan.
                        #
                        # in frame frame_id, angles are measured around
                        # the positive Z axis (counterclockwise, if Z is up)
                        # with zero angle being forward along the x axis

float32 angle_min      # start angle of the scan [rad]
float32 angle_max      # end angle of the scan [rad]
float32 angle_increment # angular distance between measurements [rad]

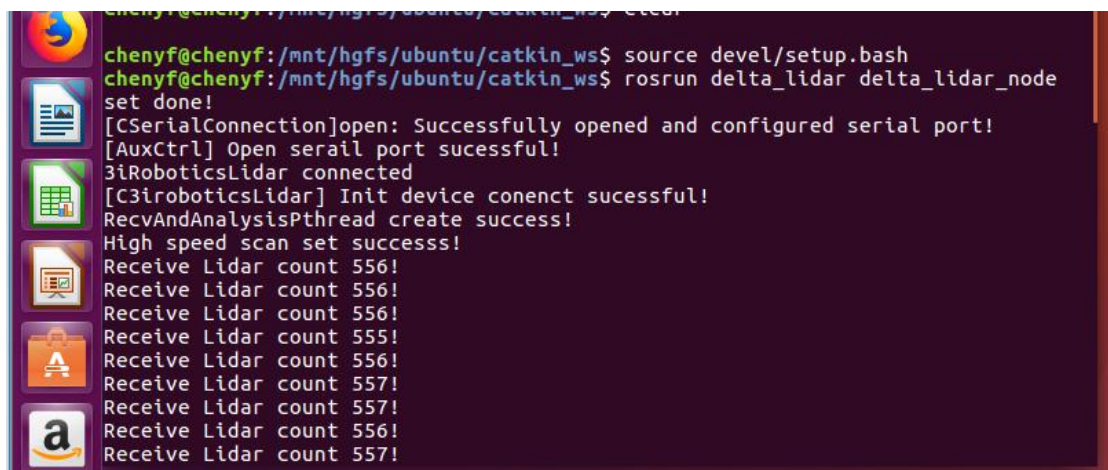
float32 time_increment  # time between measurements [seconds] - if your scanner
                        # is moving, this will be used in interpolating position
                        # of 3d points
float32 scan_time       # time between scans [seconds]

float32 range_min       # minimum range value [m]
float32 range_max       # maximum range value [m]

float32[] ranges        # range data [m] (Note: values < range_min or > range_max should be discarded)
float32[] intensities   # intensity data [device-specific units]. If your
```

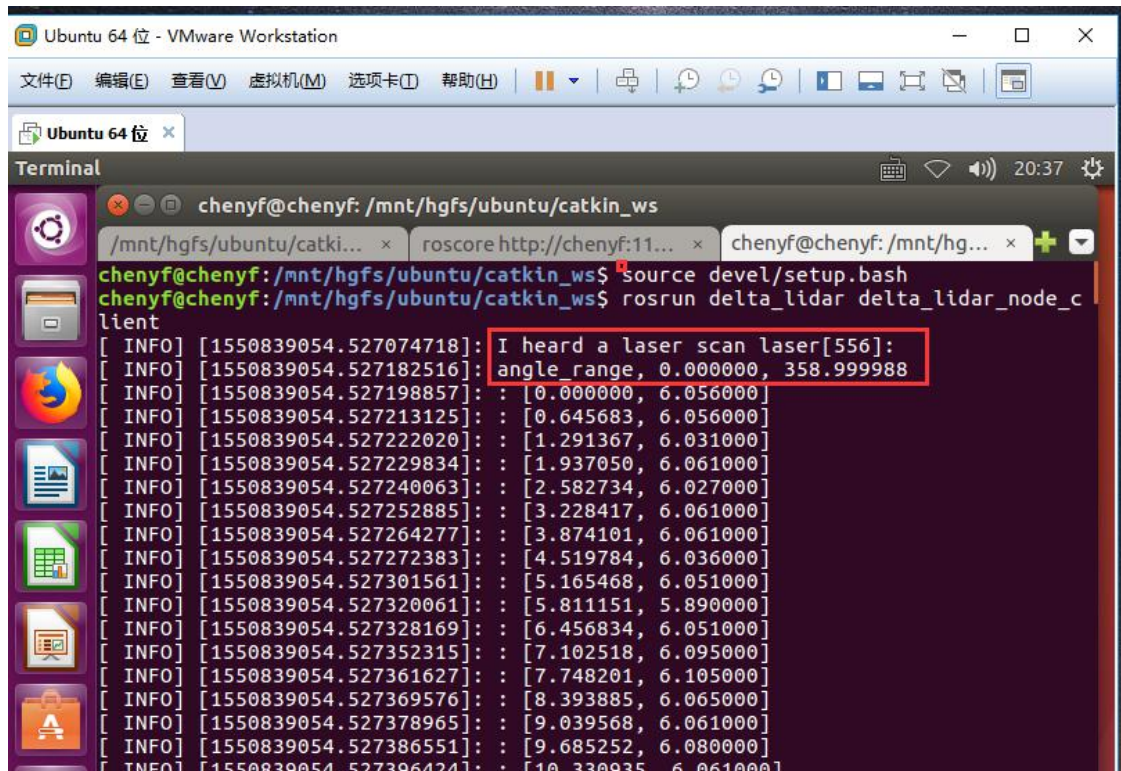
5.4.1 编译过程参考：README.md:

5.4.2 启动发布者：一个终端启动，在发布节点中 会有调试打印信息，扫描到一圈的点数。

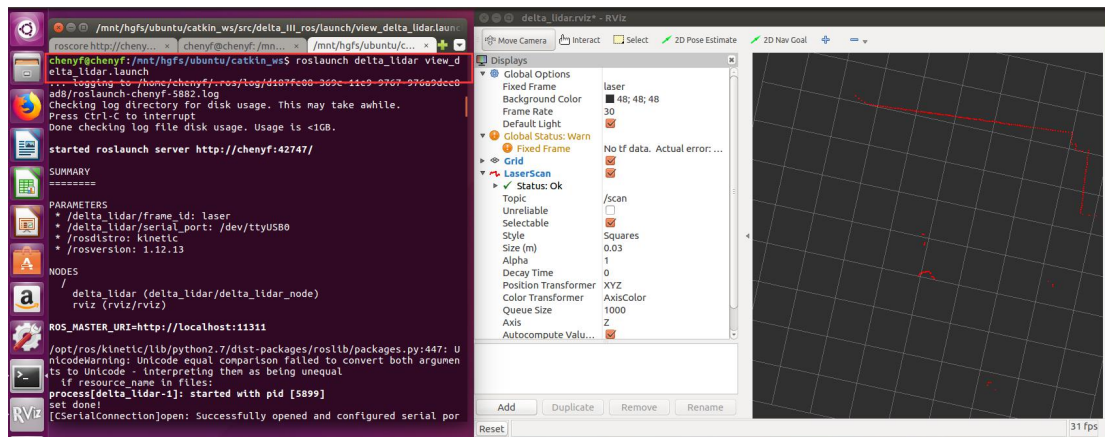


```
chenyf@chenyf: /mnt/hgfs/ubuntu/catkin_ws$ source devel/setup.bash
chenyf@chenyf: /mnt/hgfs/ubuntu/catkin_ws$ rosrn delta_lidar delta_lidar_node
set done!
[CSerialConnection]open: Successfully opened and configured serial port!
[AuxCtrl] Open serial port successful!
3iRoboticsLidar connected
[C3iRoboticsLidar] Init device connect successful!
RecvAndAnalysisPthread create success!
High speed scan set success!
Receive Lidar count 556!
Receive Lidar count 556!
Receive Lidar count 556!
Receive Lidar count 555!
Receive Lidar count 556!
Receive Lidar count 557!
Receive Lidar count 557!
Receive Lidar count 556!
Receive Lidar count 557!
```

5.4.3 启动订阅者：另一个终端中启动。当订阅者，接收到发布者的消息时



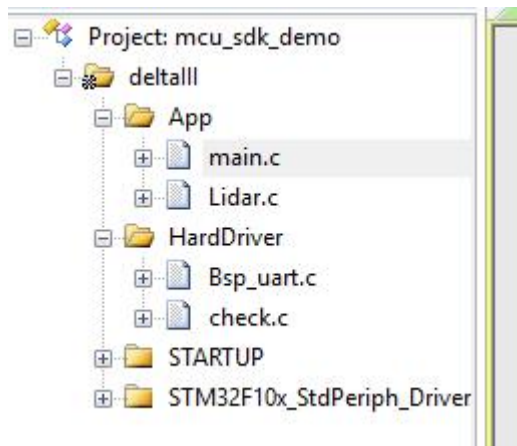
5.5 启动 rviz 建立点云图



六 . Mcu 平台 DEMO

6.1 开发环境: Stm32f103cb : 客户根据自己需要 , 移植到其他 mcu 运用。

6.2 工程 :



Bsp_uart.c :串口驱动，串口中断接收

Lidar.c：雷达解析接口

Main.c：启动雷达，设置转速，运用雷达接收解析得到扫描数据，打印扫描一圈的总点数。