

轮趣科技

N10 雷达上位机软件及 ROS 环境中 使用教程

推荐关注我们的公众号获取更新资料



版本说明:

版本	日期	内容说明
V1.0	2022/04/11	第一次发布
V2.0	2022/11/24	第二次发布

网址: www.wheeltec.net

目录

1. 上位机软件	3
1.1 安装 CH9102 芯片驱动	3
1.2 连接使用上位机软件	4
1.3 CH9102 信息修改	7
2. ROS	8
2.1 编译 ROS_SDK 包	8
2.2 Ubuntu CH9102 驱动安装	9
2.3 使用 ROS 包驱动 N10 雷达	11

1. 上位机软件

1.1 安装 CH9102 芯片驱动

① 解压资料中的“CH343SER (CH9102 的 windows 驱动).zip”文件

选中资料包中驱动软件及工具目录下对应的驱动压缩包，进行解压使用。



图 1-1-1 解压 CH9102 驱动压缩包

② 安装 CH9102 驱动

双击驱动包中的 SETUP.EXE 文件打开，然后点击安装按钮，等待安装完成后会提示驱动安装成功。

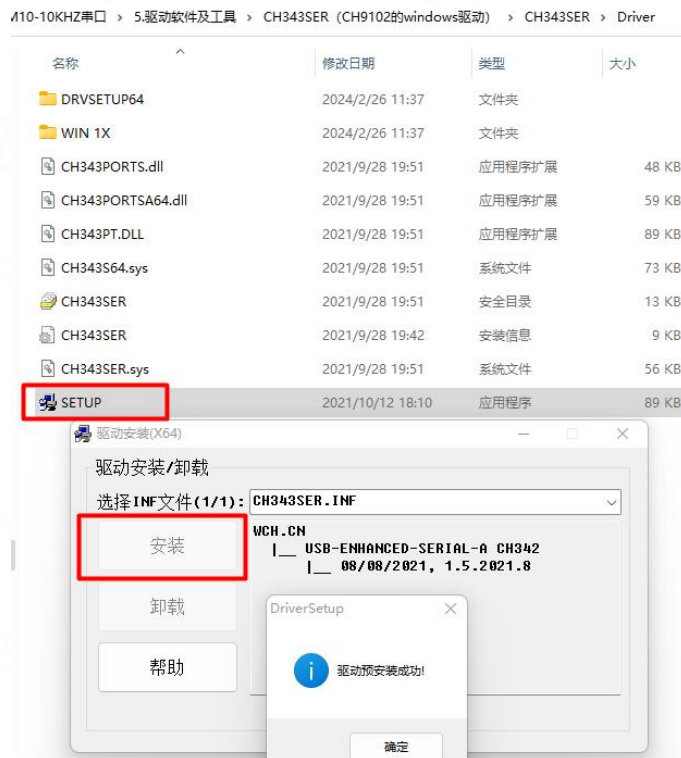


图 1-1-2 驱动安装

1.2 连接使用上位机软件

① 雷达接线

一开始到手，我们会有三个部分：雷达本体，SMH200-05H 转 Micro USB 转接模块，Micro USB 线。将他们按下图方式连接。



图 1-2-1 雷达接线

② 连接电脑运行上位机软件

将 USB 线接到电脑的 USB 口，在 windows 中右键此电脑→属性→设备管理器→端口，找到 CH9102 的端口名，其后面对应的 COM 即为雷达在 windows 中的串口号。



图 1-2-2 查看串口号

解压 windows 上位机软件中的压缩包，得到以下文件夹。

雷达客户资料V4.7_20230802 > M10-10KHZ串口 > 2.windows上位机软件

名称	修改日期	类
LSLIDAR_Single_Scan_Lidars_V1.0.4_20230504	2023/5/4 14:32	文
4. LSLIDAR_Single_Scan_上位机_V1.0.4_20230504	2023/5/5 8:44	W

图 1-2-3 解压上位机软件压缩包

进入解压后的文件夹，找到 LSLIDAR_Single_Scan_Lidars.exe 双击运行。

2.windows上位机软件 > LSLIDAR_Single_Scan_Lidars_V1.0.4_20230504

名称	修改日期	类型	大小
bearer	2023/5/4 14:29	文件夹	
iconengines	2023/5/4 14:29	文件夹	
imageformats	2023/5/4 14:29	文件夹	
ini	2023/5/4 14:29	文件夹	
NetModuleConfig_FreeBaud_new	2023/5/4 14:29	文件夹	
platforms	2023/5/4 14:29	文件夹	
qss	2023/5/4 14:29	文件夹	
styles	2023/5/4 14:29	文件夹	
translations	2023/5/4 14:29	文件夹	
D3Dcompiler_47.dll	2014/3/11 18:54	应用程序扩展	4,077 KB
libEGL.dll	2019/3/8 2:11	应用程序扩展	18 KB
libGLESv2.dll	2019/3/8 2:11	应用程序扩展	3,487 KB
LSLIDAR_Single_Scan_Lidars_V1.0.4_2...	2023/5/4 14:29	应用程序	2,164 KB
msvcp120.dll	2013/10/5 2:38	应用程序扩展	445 KB

图 1-2-4 运行上位机软件

打开上位机软件后，可以点击自动连接便能连上雷达。

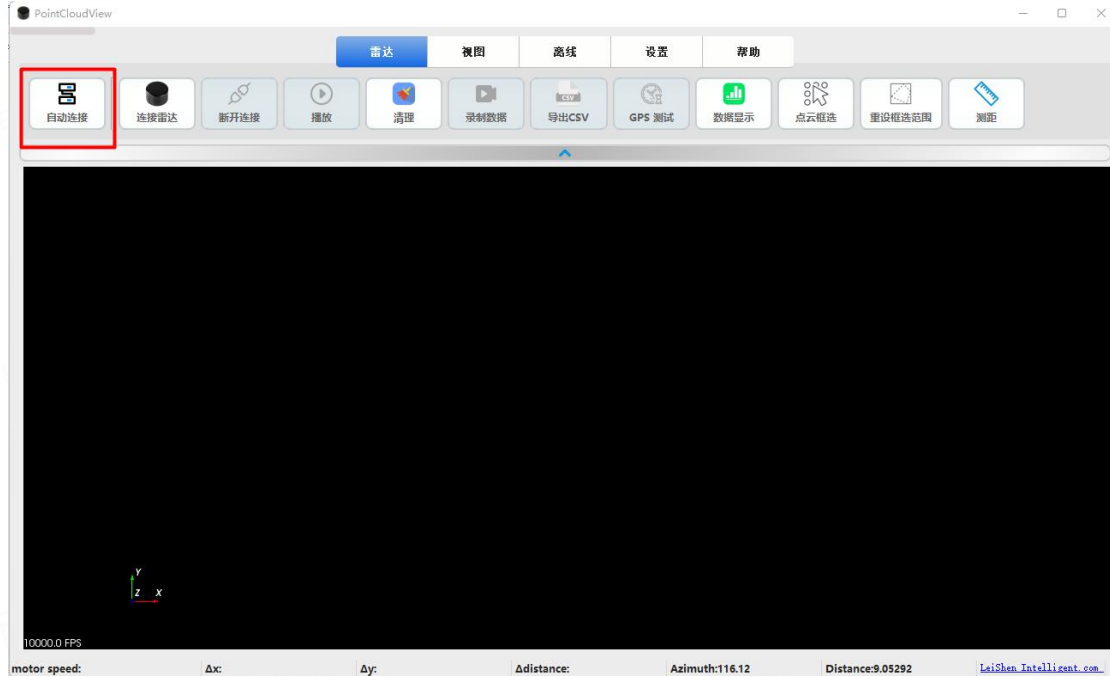


图 1-2-5 自动连接

自动连接失败或者其他情况下可选择手动连接，点击连接雷达，然后选择相应的雷达型号，选择雷达对应的串口号后，点击 ok 即可连接雷达。

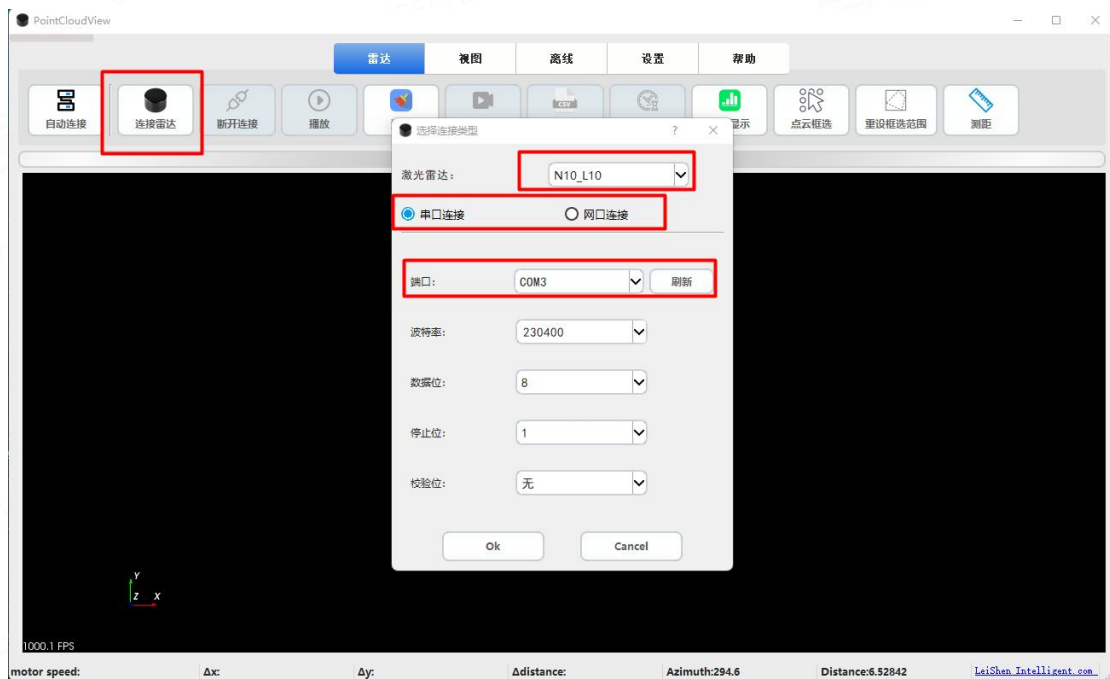


图 1-2-6 手动连接

注：上位机软件相关功能说明可以参考《上位机使用说明》。

1.3 CH9102 信息修改

点进驱动软件及工具目录下的 CH9102 修改串口号工具目录，得到以下文件，双击运行 CH34xSerCfg.exe。

< M10-10KHZ串口 > 5.驱动软件及工具 > CH9102修改串口号工具		
名称	修改日期	类型
CH34xSerCfg	2021/11/4 0:00	应用程序
CH343PT.dll	2021/10/25 0:00	应用程序扩展

图 1-3-1 解压压缩包

之后若我们的电脑有连接 CH9102 芯片的设备则会显示相关参数，我们可以通过修改对应参数栏后面的值并点击“写入配置”进行应用修改。我们雷达默认的串口号为 0001。

若没有显示相关参数，serial string 为空，则点击恢复默认值在点击写入配置，上电重新连接即可。

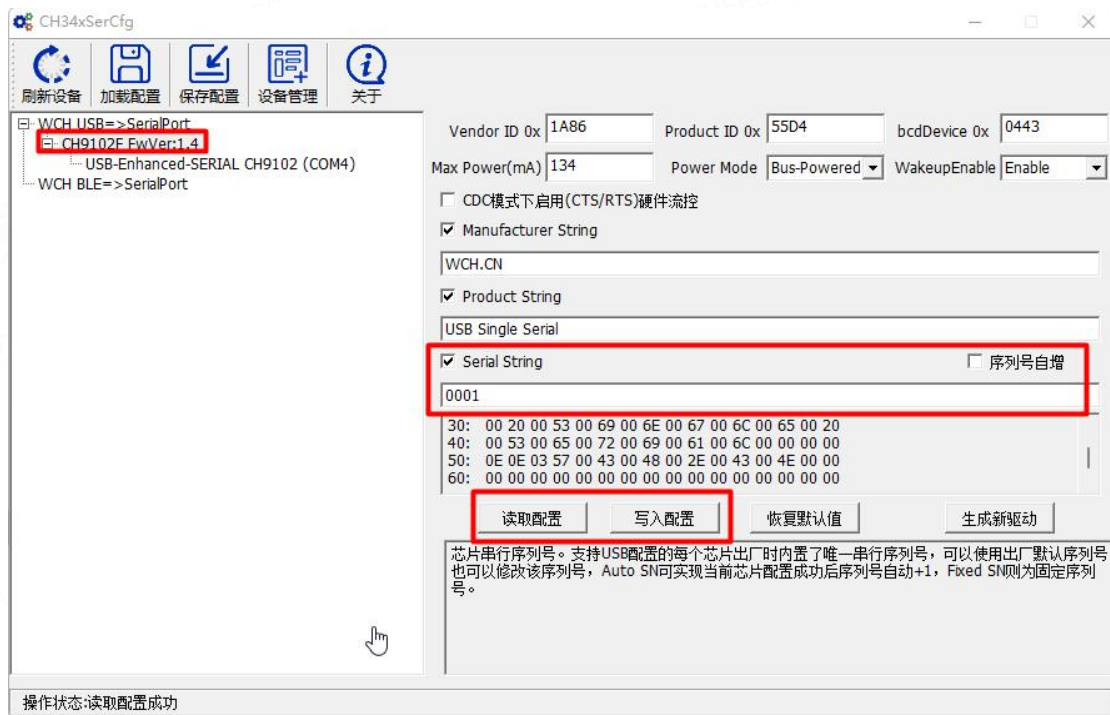


图 1-3-2 查看芯片信息及修改

2. ROS

2.1 编译 ROS_SDK 包

① 解压 ROS_SDK 压缩包

进入 ROS_SDK 目录中，解压其中的压缩包，得到 lsx10 包。

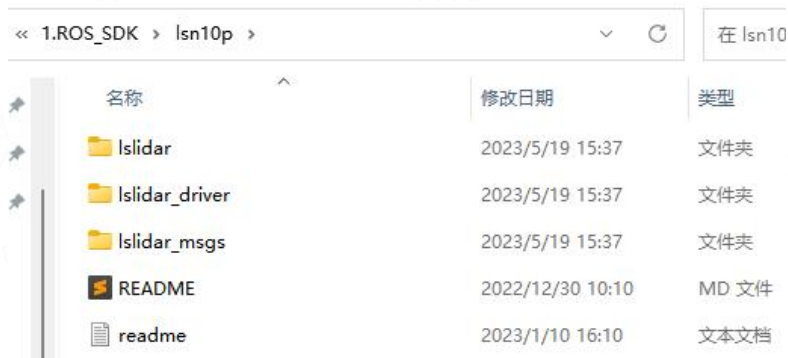


图 2-1-1 解压 ROS_SDK 压缩包

② 将功能包放入 ROS 工作空间的 src 中

将我们刚刚解压得到的 lsx10 包复制到 ros 工作空间的 src 目录中。



图 2-1-2 lsx10 放置到 ros 工作空间

③ 运行终端并编译

先联网运行以下命令安装依赖

```
sudo apt-get install libpcap-dev
```

然后返回工作空间的根目录并在该目录下打开一个终端，先执行

“catkin_make -DCATKIN_WHITELIST_PACKAGES=lslidar_msgs” 命令先编译 msgs，然后执行

“catkin_make -DCATKIN_WHITELIST_PACKAGES=lslidar_driver” 编译另外一个功能包。


```
passoni@passoni: ~/catkin_m
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
passoni@passoni:~/catkin_m$ catkin_make -DCATKIN_WHITELIST_PACKAGES=lslidar_msgs
Base path: /home/passoni/catkin_m
Source space: /home/passoni/catkin_m/src
Build space: /home/passoni/catkin_m/build
Devel space: /home/passoni/catkin_m/devel
Install space: /home/passoni/catkin_m/install
####
```

图 2-1-3 编译功能包

④ 编译成功

执行完上一步后，系统会开始编译功能包并在终端显示相应的编译进度信息。
若中间无报错最后显示为 100%则编译成功。

```
passoni@passoni: ~/catkin_m
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[ 50%] Built target sensor_msgs_generate_messages_lisp
Scanning dependencies of target geometry_msgs_generate_messages_eus
[ 50%] Built target geometry_msgs_generate_messages_eus
Scanning dependencies of target roscpp_generate_messages_py
[ 50%] Built target roscpp_generate_messages_py
Scanning dependencies of target geometry_msgs_generate_messages_nodejs
[ 50%] Built target geometry_msgs_generate_messages_nodejs
Scanning dependencies of target std_srvs_generate_messages_eus
[ 50%] Built target std_srvs_generate_messages_eus
Scanning dependencies of target roscpp_generate_messages_lisp
[ 50%] Built target roscpp_generate_messages_lisp
Scanning dependencies of target bond_generate_messages_eus
[ 50%] Built target bond_generate_messages_eus
Scanning dependencies of target rosgenmsg_generate_messages_cpp
[ 50%] Built target rosgenmsg_generate_messages_cpp
Scanning dependencies of target tf_generate_messages_py
[ 50%] Built target tf_generate_messages_py
Scanning dependencies of target dynamic_reconfigure_generate_messages_cpp
[ 50%] Built target dynamic_reconfigure_generate_messages_cpp
Scanning dependencies of target topic_tools_generate_messages_cpp
[ 50%] Built target topic_tools_generate_messages_cpp
Scanning dependencies of target nodelet_generate_messages_lisp
[ 50%] Built target nodelet_generate_messages_lisp
Scanning dependencies of target lslidar_driver
[ 62%] Building CXX object lslidar_driver/CMakeFiles/lslidar_driver.dir/src/lslidar_driver.cc.o
[ 75%] Linking CXX shared library /home/passoni/catkin_m/devel/lib/liblslidar_driver.so
[ 75%] Built target lslidar_driver
Scanning dependencies of target lslidar_driver_node
[ 87%] Building CXX object lslidar_driver/CMakeFiles/lslidar_driver_node.dir/src/lslidar_driver_node.cc.o
[100%] Linking CXX executable /home/passoni/catkin_m/devel/lib/lslidar_driver/lslidar_driver_node
[100%] Built target lslidar_driver_node
passoni@passoni:~/catkin_m$
```

图 2-1-4 编译成功

2. 2 Ubuntu CH9102 驱动安装

可以通过在终端执行 `lsmod|grep ch343` 命令确认系统是否已经有该驱动模块。

① 解压驱动压缩包

首先解压“ch343ser_linux（CH9102 的 linux 驱动）.zip”到 Ubuntu 系统中。

M10雷达客户资料V5.3_20240221 > M10-10KHZ串口 > 5.驱动软件及工具

名称	修改日期	类型
CH9102修改串口号工具	2024/2/26 10:20	文件夹
CH343SER (CH9102的windows驱动)	2022/1/20 20:56	WinRAR
ch343ser_linux (CH9102的linux驱动)	2022/5/11 16:00	WinRAR
ch9102_udev	2022/6/10 15:49	SH 文件
CH9102使用手册	2022/6/10 15:25	WPS PDF
tty_uart-master	2022/4/27 15:13	WinRAR

图 2-2-1 解压驱动压缩包

② 移除 cdc_acm 模块

首先运行 `lsmod | grep cdc_acm` 检查是否有加载 `cdc_acm` 模块，此模块一般为 Linux 自带模块，可用于注册相应 USB 设备驱动等。如果显示已加载到内核的模块有 `cdc_acm`，则运行 `sudo rmmod cdc_acm` 先将其 `rmmod`，如果未显示相应模块则无需 `rmmod`。

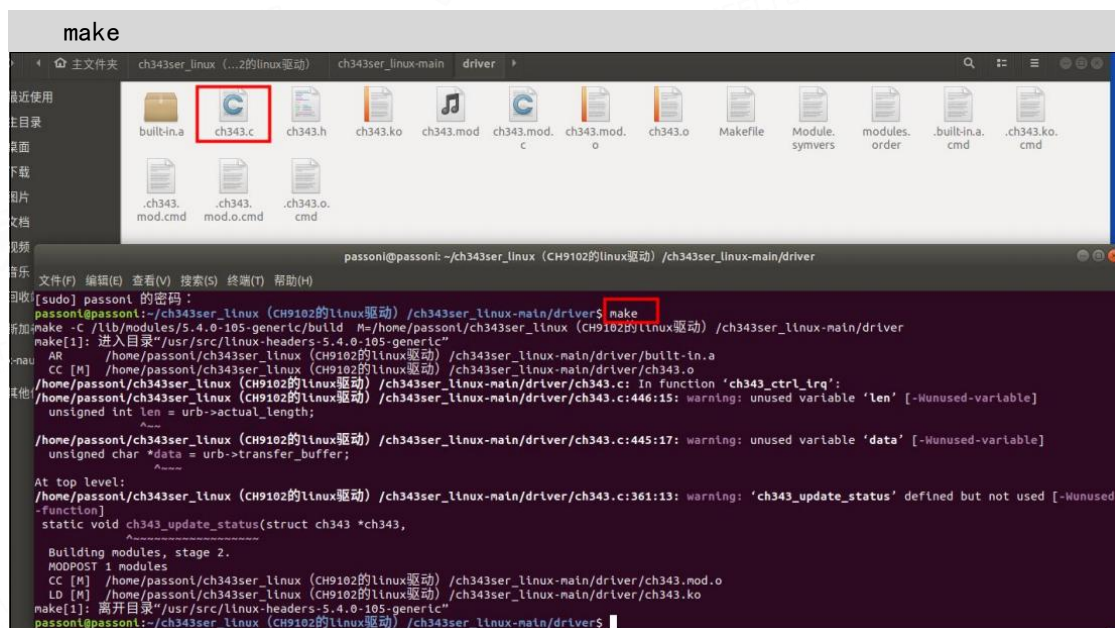


```
passoni@passoni: ~/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
passoni@passoni:~$ cd ~/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver
passoni@passoni:~/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver$ lsmod | grep cdc_acm
cdc_acm          40960      0
passoni@passoni:~/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver$ sudo rmmod cdc_acm
[sudo] passoni 的密码:
passoni@passoni:~/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver$
```

图 2-2-2 移除 cdc_acm 说明

③ 安装驱动

在 `driver` 路径下终端输入 `make` 根据文件夹内的编译规则对 `.c` 文件进行编译，会相应生成 `.ko` 等文件如图 1-2-5:



```
make
passoni@passoni:~/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver
[sudo] passoni 的密码:
passoni@passoni:~/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver$ make
make -C /lib/modules/5.4.0-105-generic/build M=/home/passoni/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver
make[1]: 进入目录"/usr/src/linux-headers-5.4.0-105-generic"
AR /home/passoni/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver/built-in.a
CC [M] /home/passoni/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver/ch343.o
/home/passoni/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver/ch343.c: In function 'ch343_ctrl_irq':
/home/passoni/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver/ch343.c:446:15: warning: unused variable 'len' [-Wunused-variable]
    unsigned int len = urb->actual_length;
                   ^~~~
/home/passoni/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver/ch343.c:445:17: warning: unused variable 'data' [-Wunused-variable]
    unsigned char *data = urb->transfer_buffer;
                   ^~~~~
At top level:
/home/passoni/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver/ch343.c:361:13: warning: 'ch343_update_status' defined but not used [-Wunused-function]
static void ch343_update_status(struct ch343 *ch343,
~~~~~
Building modules, stage 2.
MODPOST 1 modules
CC [M] /home/passoni/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver/ch343.mod.o
LD [M] /home/passoni/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver/ch343.ko
make[1]: 离开目录"/usr/src/linux-headers-5.4.0-105-generic"
passoni@passoni:~/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver$
```

图 2-2-3 编译 ch343.c

完成编译后需要将模块载入，具体细节可参照 `Makefile`，这里不再展开

```
sudo make load
```

模块载入后再将文件安装到指定位置:

```
sudo make install
```

```

passoni@passoni:~/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver$ sudo make load
insmod ch343.ko
passoni@passoni:~/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver$ sudo make install
make -C /lib/modules/5.4.0-105-generic/build M=/home/passoni/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver
make[1]: 进入目录"/usr/src/linux-headers-5.4.0-105-generic"
Building modules, stage 2.
MODPOST 1 modules
make[1]: 离开目录"/usr/src/linux-headers-5.4.0-105-generic"
mkdir -p /lib/modules/5.4.0-105-generic/kernel/drivers/usb/serial/
cp -f /ch343.ko /lib/modules/5.4.0-105-generic/kernel/drivers/usb/serial/
depmod -a
passoni@passoni:~/ch343ser_linux (CH9102的linux驱动) /ch343ser_linux-main/driver$
  
```

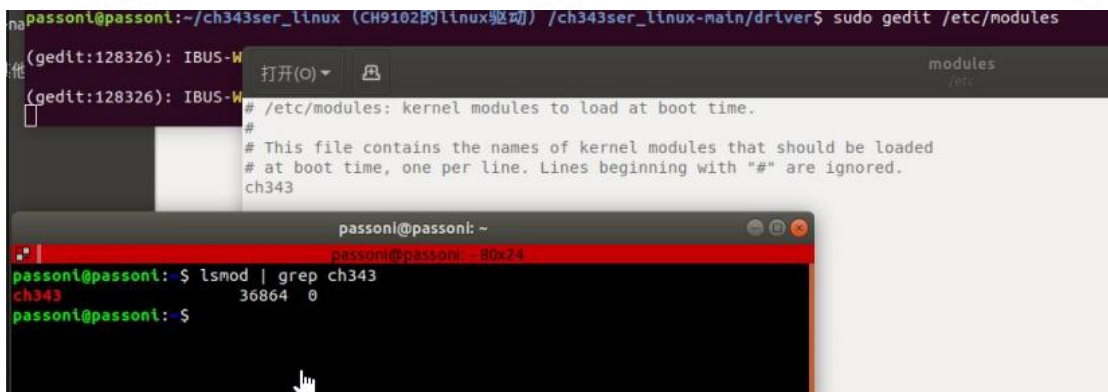
图 2-2-4 载入模块并安装

为了方便后续系统启动会自动加载 ch343.ko，需要在/etc/modules 文件里加入 ch343 在终端输入：

```
sudo gedit /etc/modules
```

新增 ch343 在最后一行后保存并退出重启系统，则系统会默认自动加载 ch343 驱动，重启后在终端输入命令查看是否自动加载：

```
lsmod | grep ch343
```



```

(gedit:128326): IBUS-W
(gedit:128326): IBUS-W
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
ch343

passoni@passoni: ~
passoni@passoni: ~$ lsmod | grep ch343
ch343          36864  0
passoni@passoni: ~$
  
```

图 2-2-5 编辑/etc/modules 文件

2.3 使用 ROS 包驱动 N10 雷达

① 将雷达连接至 Ubuntu 系统

至此已经安装完 CH9102 的 Linux 驱动，安装完成后记得重启系统检查设备能否正常加载，在终端输入命令可查看设备已识别成 ttyCH343USB。

```
ll /dev | grep ttyCH343USB
```

```

passoni@passoni:~$ ll /dev | grep ttyCH343USB
crwxrwxrwx  1 root  dialout 170,  0 Jun 10 11:56 ttyCH343USB0
  
```

图 2-3-1 查看已连接设备

② 配置 N10 ROS 驱动包

这里我用 sublime 打开 N10 的 ROS 包，方便查看文件目录结构和文件内容。我们打开 lsx10/lslidar_driver/launch 文件夹下的 lslidar_serial.launch 进行查看，这个 launch 文件也是我们之后要运行的启动雷达的文件。

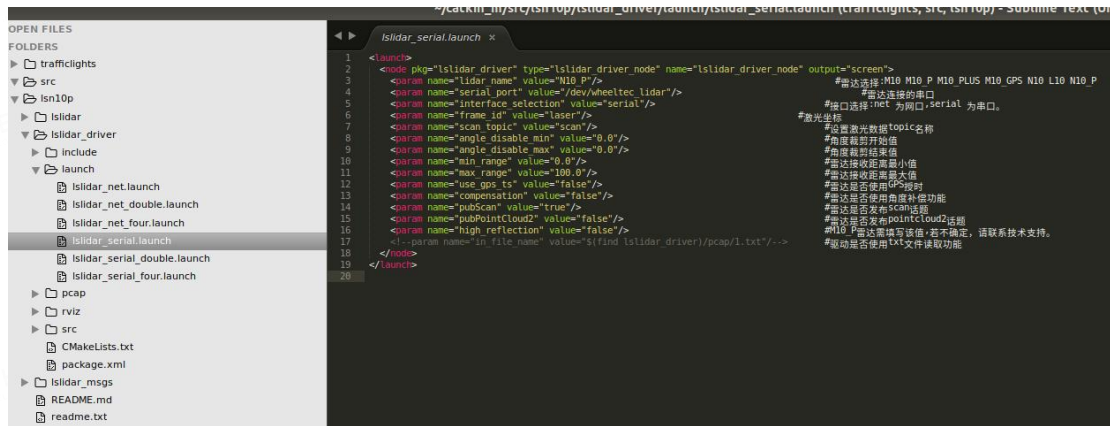


图 2-3-2 查看 lslidar_serial.launch

用 lslidar_serial.launch 开启 N10 雷达前需要先确认雷达在 ubuntu 中的串口。我们可以通过 `ll /dev|grep ttyCH343USB` 命令查看。一般情况下串口名为 `ttyCH343USB`。这里我通过创建串口别名将其串口名重映射为 `wheeltec_lidar`（这一步不是必须的）。

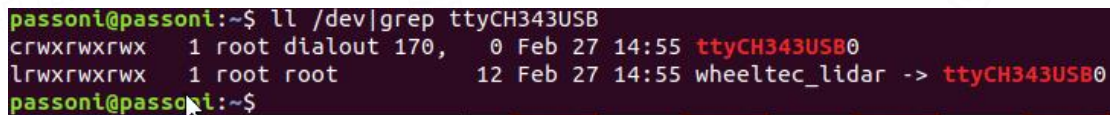


图 2-3-3 查看串口名

注：串口别名的创建可以参考雷达资料包根目录中的“`wheeltec_udev.sh`”脚本。本质是通过设备属性来定位设备并为其创建命名规则。

接下来将 lslidar_serial.launch 中对应的 `serial_port` 的值改为雷达的串口名，串口名即为我们上图查看到的，这个参数一般不需要修改，一般默认的串口名就是“`/dev/ttyUSB0`”。这里我进行了串口别名，所以也可以将其改为 `wheeltec_lidar`。

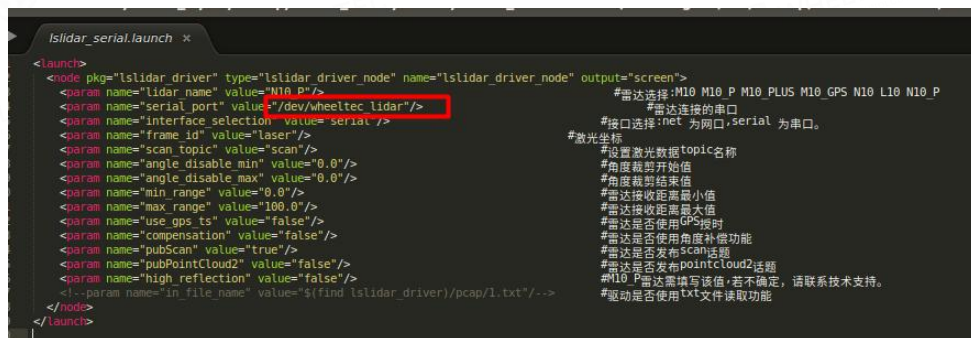


图 2-3-4 修改串口参数

③ 屏蔽雷达角度

根据自己实际需要选择屏蔽雷达角度。（此步非必须）

在 `lslidar_serial.launch` 中，先将 `truncated_mode` 参数置 1，表明需要屏蔽雷达角度。然后根据需要屏蔽的角度设置 `disable_min` 和 `disable_max` 参数，`disable_min` 参数代表需要屏蔽的角度初始值，`disable_max` 参数代表需要屏蔽的角度结束值，如 `disable_min=[0,90,180,270]`，`disable_max=[0,90,180,270]`，代表屏蔽雷达 $0\sim 45^\circ$ 、 $90^\circ\sim 135^\circ$ 、 $180^\circ\sim 225^\circ$ 、 $270^\circ\sim 315^\circ$ 角度范围。支持多角度屏蔽。

```

lslidar_serial.launch
<launch>
<node pkg="lslidar_driver" type="lslidar_driver_node" name="lslidar_driver_node" output="screen">
  <param name="lidar_name" value="N10_P"/>
  <param name="serial_port" value="/dev/wheeltec_lidar"/>
  <param name="interface_selection" value="serial"/>
  <param name="frame_id" value="laser"/>
  <param name="scan_topic" value="scan"/>
  <param name="angle_disable_min" value="0.0"/>
  <param name="angle_disable_max" value="0.0"/>
  <param name="min_range" value="0.0"/>
  <param name="max_range" value="100.0"/>
  <param name="use_gps_ts" value="false"/>
  <param name="compensation" value="false"/>
  <param name="pubScan" value="true"/>
  <param name="pubPointCloud2" value="false"/>
  <param name="high_reflection" value="false"/>
  <!-- param name="in_file_name" value="$(find lslidar_driver)/pcap/1.txt"/-->
</node>

  <!-- 请神N10P雷达参数 -->
  <param name="lslidar_driver_node/truncated_mode" value="0"/>
  <rosparam param="lslidar_driver_node/disable_min">[120]</rosparam>
  <rosparam param="lslidar_driver_node/disable_max">[240]</rosparam>
  <!-- 请神N10P雷达参数 -->
</launch>

```

图 2-3-5 ROS 中 N10 角度屏蔽

多角度屏蔽时，以雷达正前方为 0° 度角方向，扫描角度顺时针增加，如图所示。

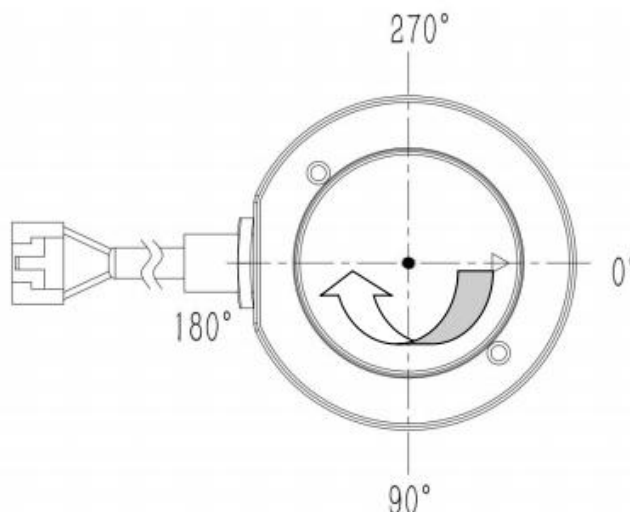


图 2-3-6 ROS 中 N10 角度及坐标系

④ 启动雷达

在配置完 `launch` 文件后，我们打开一个终端，执行以下命令：

roslaunch lslidar_driver lslidar_serial.launch

成功启动后终端会出现以下字样。

```
/home/passoni/catkin_m/src/lslidar/lslidar_driver/launch/lslidar_serial.launch http://192.168.0.136:11311
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
* /lslidar_driver_node/compensation: False
* /lslidar_driver_node/frame_id: laser
* /lslidar_driver_node/high_reflection: False
* /lslidar_driver_node/interface_selection: serial
* /lslidar_driver_node/lidar_name: N10_P
* /lslidar_driver_node/max_range: 100.0
* /lslidar_driver_node/min_range: 0.0
* /lslidar_driver_node/pubPointCloud2: False
* /lslidar_driver_node/pubScan: True
* /lslidar_driver_node/scan_topic: scan
* /lslidar_driver_node/serial_port: /dev/wheeltec_lidar
* /lslidar_driver_node/use_gps_ts: False
* /roscdistro: melodic
* /rosversion: 1.14.10

NODES
/
  lslidar_driver_node (lslidar_driver/lslidar_driver_node)

auto-starting new master
process[roscout-1]: started with pid [46957]
ROS_MASTER_URI=http://192.168.0.136:11311

setting /run_id to 003ad2cc-158f-11ee-9e47-000c291e1698
process[roscout-1]: started with pid [46968]
started core service [/roscout]
process[lslidar_driver_node-2]: started with pid [46971]
[ INFO] [1687941446.559956376]: Lidar is N10_P
[ INFO] [1687941446.561657190]: Opening PCAP file
port = /dev/wheeltec_lidar, baud_rate = 460800
open_port /dev/wheeltec_lidar OK !
[ INFO] [1687941446.569194314]: Initialised lslidar without error
```

图 2-3-7 成功启动雷达

⑤ 雷达数据查看

在 ROS 中雷达数据的展示形式可以是文本形式，也可以是点云图像形式。

当运行 lslidar_serial.launch 启动雷达后，/lslidar_serial_node 节点便会在 ros 中发布一个 /scan 话题。

```
passoni@passoni
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
passoni@passoni:~$ rostopic info /scan
Type: sensor_msgs/LaserScan

Publishers:
* /lslidar (http://localhost:46479/)

Subscribers: None
```

图 2-3-8 查看 /scan 话题发布者

我们可以通过执行 rostopic echo /scan 命令直接进行雷达数据的查看。/scan 话题中的消息类型是 LaserScan，该消息类型的具体描述可以参考以下链接：

http://docs.ros.org/en/api/sensor_msgs/html/msg/LaserScan.html

图 2-3-9 查看雷达话题数据

[illegible]

图 2-3-10 修改 Fixed Frame

The screenshot shows the 'rviz' application window. On the left, the 'Displays' panel lists 'Global Options', 'Fixed Frame', 'Background Color', 'Frame Rate', 'Default Light', 'Global Status: Warning', and 'Fixed Frame'. The 'Fixed Frame' is set to 'world'. The main window shows a 3D view of a robot in a simulated environment. A 'Create visualization' dialog box is open, showing a tree of topics. The 'By topic' tab is selected, and the 'LaserScan' topic under the '/scan' topic is highlighted. The 'Description' field shows that the visualization displays data from a sensor_msgs::LaserScan message. The 'Display Name' field is empty. At the bottom, there are buttons for 'Add', 'Duplicate', 'Remove', 'Rename', 'Cancel', and 'OK'. The 'OK' button is highlighted with a red box.

图 2-3-11 添加 LaserScan

成功添加 LaserScan 后我们便可以在 Rviz 中看到这样的雷达点云图像。

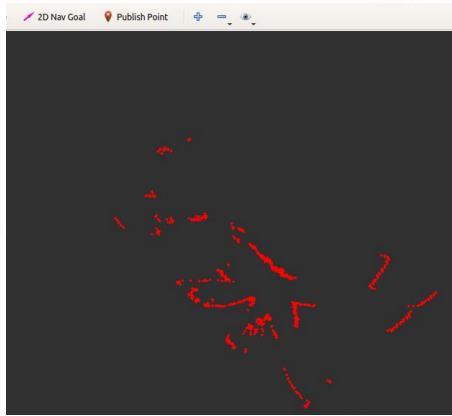


图 2-3-12 Rviz 显示雷达点云

⑥ 雷达数据在 ros 的应用

当/lslidar_serial_node 节点通过话题发布 N10 雷达的数据之后，我们就可以通过订阅它发出的/scan 话题在不同的功能中获取使用雷达的数据。比如建图、导航避障、雷达跟随等等。下图就是一个在 python 中订阅/scan 话题的示例。

```
def __init__(self):
    self.lastScan=None
    self.winSize = rospy.get_param('~winSize')
    self.deltaDist = rospy.get_param('~deltaDist')
    self.scanSubscriber = rospy.Subscriber('scan', LaserScan, self.registerScan)
    self.positionPublisher = rospy.Publisher('object_tracker/current_position', Pos
    self.infoPublisher = rospy.Publisher('object_tracker/info', StringMsg, queue_s
```

图 2-3-13 订阅/scan 话题

话题名默认情况下为/scan，但我们也可以通过手动修改 lslidar_serial.launch 中的 scan_topic 的值来修改启动后发布的雷达话题名。

```
<launch>

<node pkg="lslidar_driver" type="lslidar_driver_node" name="lslidar_driver_node">
  <param name="lidar_name" value="N10 P"/>
  <param name="serial_port" value="/dev/wheeltec_lidar"/>
  <param name="interface_selection" value="serial"/>
  <param name="frame_id" value="laser"/>
  <param name="scan_topic" value="scan"/>
  <param name="angle_disable_min" value="0.0"/>
  <param name="angle_disable_max" value="0.0"/>
  <param name="min_range" value="0.0"/>
  <param name="max_range" value="100.0"/>
  <param name="use_gps_ts" value="false"/>
  <param name="compensation" value="false"/>
  <param name="pubScan" value="true"/>
  <param name="pubPointCloud2" value="false"/>
  <param name="high_reflection" value="false"/>
  <!--param name="in_file_name" value="$(find lslidar_driver)/pcap/1.txt"/-->
</node>
```

图 2-3-14 修改雷达话题名