

# 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 |
|      |            |       |

网址：[www.wheeltec.net](http://www.wheeltec.net)

# 目录

|                              |    |
|------------------------------|----|
| 1. 上位机软件 .....               | 3  |
| 1.1 安装 cp2102 芯片驱动 .....     | 3  |
| 1.2 连接使用上位机软件 .....          | 4  |
| 1.3 CP2102 信息修改 .....        | 6  |
| 2. ROS .....                 | 8  |
| 2.1 编译 ROS_SDK 包 .....       | 8  |
| 2.2 Ubuntu CP2102 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 cp2102 芯片驱动

### ① 解压资料中的“CP2102usb 驱动.zip”文件

选中资料包中 CP2102 芯片目录下对应的驱动压缩包，进行解压使用。



图 1-1-1 解压 CP2102 驱动压缩包

### ② 安装 CP2102 驱动

根据系统的位数来选择相应位数的驱动进行安装。



图 1-1-2 选择相应位数的驱动

双击打开安装程序后点击下一页→接受协议→下一页进行驱动安装，安装完成后会显示以下界面，这时电脑就能识别到我们的 CP2102 的 usb 了。



图 1-1-3 驱动安装完成

## 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 CP210x USB 的端口号，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。

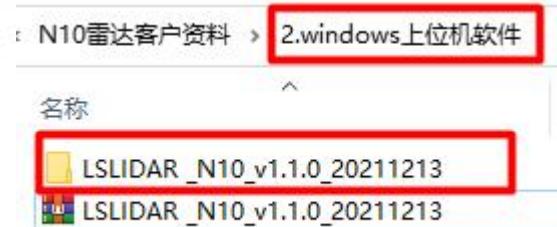


图 1-2-3 解压上位机软件压缩包

进入解压后的文件夹，找到 LSLIDAR\_N10\_v1.1.0\_20211213.exe 双击运行。



图 1-2-4 运行上位机软件

打开上位机软件后，找到 COM 选择的下拉窗口，选择我们刚刚查看到的串口号。



图 1-2-5 选择对应串口

之后点击左上角的“雷达连接”，即可连接到我们的雷达，并在上位机软件中显示雷达点云数据。

注：上位机软件相关功能说明可以参考《N10 规格书 v1.0》第 15~26 页。

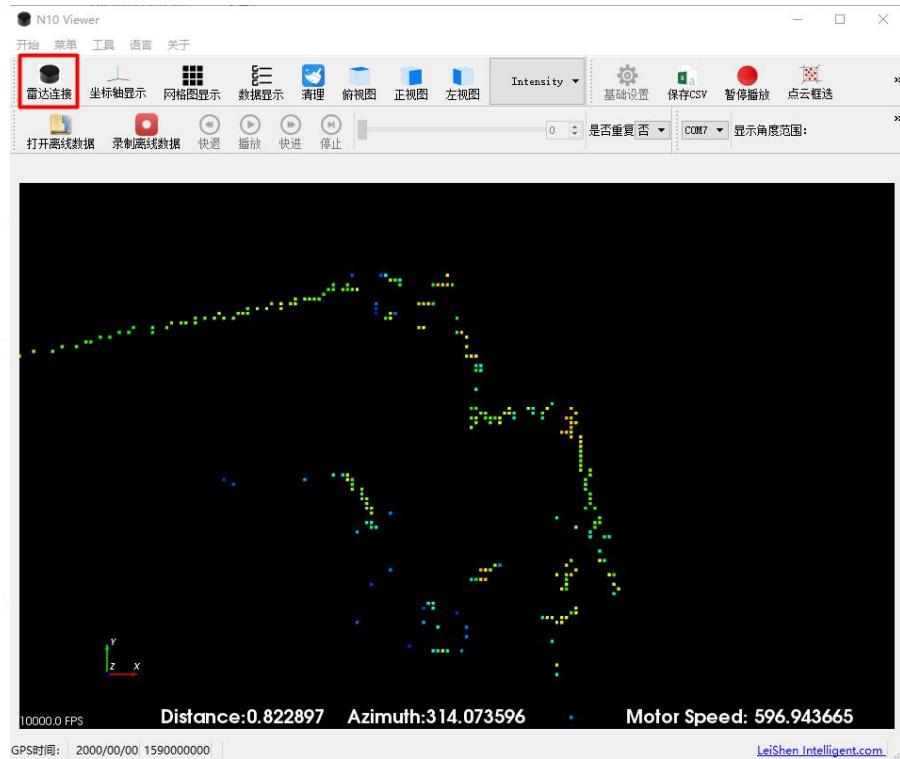


图 1-2-6 雷达连接上位机软件

## 1. 3 CP2102 信息修改

解压 CP2102 芯片目录下的“修改端口号软件”，得到以下文件。

| N10雷达客户资料 > 5.驱动软件及工具 > CP2102芯片驱动及参数修改软件 |                 |            |
|---|-----------------|------------|
| 名称  | 修改日期            | 类型         |
| CP2102usb驱动                               | 2022/4/11 9:57  | 文件夹        |
| 修改端口号软件                                   | 2022/4/11 9:56  | 文件夹        |
| CP2102usb驱动                               | 2020/9/23 20:21 | WinRAR ZIP |
| 修改端口号软件                                   | 2020/9/28 11:15 | WinRAR ZIP |

图 1-3-1 解压压缩包

打开解压后的文件夹，双击运行 CP21xxCustomizationUtility.exe。

| N10雷达客户资料 > 5.驱动软件及工具 > CP2102芯片驱动及参数修改软件 > 修改端口号软件 |                 |                  |        |
|---|-----------------|------------------|--------|
| 名称  | 修改日期            | 类型               | 大小     |
| common  | 2020/9/23 11:37 | 文件夹              |        |
| configuration                                       | 2020/9/23 11:37 | 文件夹              |        |
| features  | 2020/9/23 11:37 | 文件夹              |        |
| p2  | 2020/9/23 11:37 | 文件夹              |        |
| plugins   | 2020/9/23 11:37 | 文件夹              |        |
| .eclipseproduct                                     | 2014/11/11 3:25 | ECLIPSEPRODUC... | 1 KB   |
| artifacts   | 2014/11/11 3:26 | XML 文档           | 6 KB   |
| CP21xxCustomizationUtility                          | 2014/11/11 3:26 | 应用程序             | 312 KB |
| CP21xxCustomizationUtility                          | 2014/11/11 3:26 | 配置设置             | 1 KB   |

图 1-3-2 运行 CP21xxCustomizationUtility.exe

之后若我们的电脑有连接 CP2102 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 Value 值并点击“Program Device”进行应用修改。

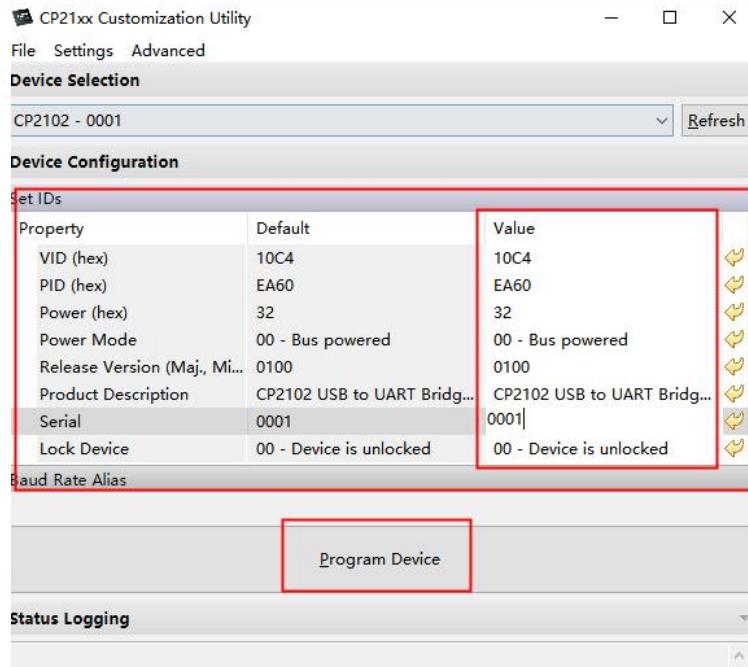


图 1-3-3 查看芯片信息及修改

## 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 lsn10 包。



图 2-1-1 解压 ROS\_SDK 压缩包

#### ② 将 lsn10 放入 ROS 工作空间的 src 中

将我们刚刚解压得到的 lsn10 包复制到 ros 工作空间的 src 目录中。



图 2-1-2 lsn10 放置到 ros 工作空间

#### ③ 运行终端并编译

返回工作空间的根目录并在该目录下打开一个终端，执行

“catkin\_make -DCATKIN\_WHITELIST\_PACKAGES=lsn10”命令进行指定 lsn10 功能包编译。

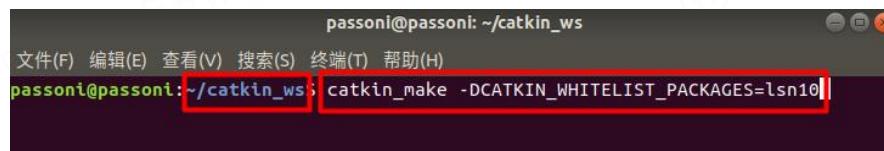


图 2-1-3 编译功能包

#### ④ 编译成功

执行完上一步后，系统会开始编译 lsn10 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_ws
[ 25%] Generating Euslisp manifest code for lsn10
[ 25%] Built target lsn10_generate_messages_eus
Scanning dependencies of target rosgraph_msgs_generate_messages_py
[ 25%] Built target rosgraph_msgs_generate_messages_py
Scanning dependencies of target rosgraph_msgs_generate_messages_nodejs
[ 25%] Built target rosgraph_msgs_generate_messages_nodejs
Scanning dependencies of target std_srvs_generate_messages_cpp
[ 25%] Built target std_srvs_generate_messages_cpp
Scanning dependencies of target rosgraph_msgs_generate_messages_eus
[ 25%] Built target rosgraph_msgs_generate_messages_eus
Scanning dependencies of target rosgraph_msgs_generate_messages_lisp
[ 25%] Built target rosgraph_msgs_generate_messages_lisp
Scanning dependencies of target sensor_msgs_generate_messages_cpp
[ 25%] Built target sensor_msgs_generate_messages_cpp
Scanning dependencies of target sensor_msgs_generate_messages_eus
[ 25%] Built target sensor_msgs_generate_messages_eus
Scanning dependencies of target rosgraph_msgs_generate_messages_cpp
[ 25%] Built target rosgraph_msgs_generate_messages_cpp
Scanning dependencies of target rospp_generate_messages_lisp
[ 25%] Built target rospp_generate_messages_lisp
Scanning dependencies of target lsn10
[ 50%] Building CXX object CMakeFiles/lsn10.dir/src/lsiosr.cpp.o
[ 75%] Building CXX object lsn10/CMakeFiles/lsn10.dir/src/lsn10.cpp.o
[100%] Linking CXX executable /home/passoni/catkin_ws/devel/lib/lsn10/lsn10
[100%] Built target lsn10
Scanning dependencies of target lsn10_generate_messages
[100%] Built target lsn10_generate_messages
passoni@passoni:~/catkin_ws$
```

图 2-1-4 编译成功

## 2. 2 Ubuntu CP2102 驱动安装

一般情况下 ubuntu 系统都会自带该驱动模块，不需要手动安装，该步骤主要针对系统中没有该驱动的用户。可以通过在终端执行 lsmod|grep cp210x 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“Ubuntu CP2102 驱动”目录下的压缩包，得到以下文件。

| N10雷达客户资料 > 5.驱动软件及工具 > Ubuntu CP2102驱动 > Linux_3.x.x-4.x.x_VCP_Driver_Source |                 |      |       |
|---|-----------------|------|-------|
| 名称  | 修改日期            | 类型   | 大小    |
| cp210x.c  | 2021/1/29 19:12 | C 文件 | 78 KB |
| cp210x_gpio_example.c   | 2021/1/29 19:12 | C 文件 | 2 KB  |
| cp210x_gpio_example_gpiolib.c   | 2021/1/29 19:12 | C 文件 | 4 KB  |
| CP210x_VCP_Linux_4.x_Release_Notes  | 2021/1/29 19:16 | 文本文档 | 3 KB  |
| Makefile  | 2020/8/19 17:03 | 文件   | 1 KB  |

图 2-2-1 解压驱动压缩包

### ② 打开解压得到的 txt 文档

打开解压得到的 txt 文档，并参照“Ubuntu:”后面四步进行驱动安装。

```
The bundle contains:
* cp210x.c
* Makefile
* cp210x_gpio_example.c
* cp210x_gpio_example_gpiolib.c
* build.sh
* CP210x_VCP_Linux_4.x_Release_Notes.txt

Build instructions:
Ubuntu:
1. make (your cp210x driver)
2. cp cp210x.ko to /lib/modules/<kernel-version>/kernel/drivers/usb/serial
3. insmod /lib/modules/<kernel-version>/kernel/drivers/usb/serial/usbserial.ko
4. insmod cp210x.ko
```

图 2-2-2 打开 txt 文档说明

### ③ 安装驱动

将我们解压得到的驱动文件夹整个复制到 ubuntu 的 home 路径下。



图 2-2-3 复制驱动文件夹到 ubuntu

打开一个终端进入到我们刚刚复制过来的驱动文件夹中，并执行 make 指令进行编译。

```
passoni@passoni:~$ cd Linux_3.x.x_4.x.x_VCP_Driver_Source/
passoni@passoni:~/Linux_3.x.x_4.x.x_VCP_Driver_Source$ make
make -C /lib/modules/`uname -r`/build M=/home/passoni/Linux_3.x.x_4.x.x_VCP_Driver_Source/modules
make[1]: 进入目录“/usr/src/linux-headers-5.4.0-84-generic”
  CC [M]  /home/passoni/Linux_3.x.x_4.x.x_VCP_Driver_Source/cp210x.o
Building modules, stage 2.
MODPOST 1 modules
  CC [M]  /home/passoni/Linux_3.x.x_4.x.x_VCP_Driver_Source/cp210x.mod.o
  LD [M]  /home/passoni/Linux_3.x.x_4.x.x_VCP_Driver_Source/cp210x.ko
make[1]: 离开目录“/usr/src/linux-headers-5.4.0-84-generic”
passoni@passoni:~/Linux_3.x.x_4.x.x_VCP_Driver_Source$
```

图 2-2-4 执行 make 编译驱动包

编译完成之后，文件夹内会生成几个新的文件。我们继续在这个终端执行一个复制文件的命令

sudo cp cp210x.ko /lib/modules/<kernel-version>/kernel/drivers/usb/serial

其中<kernel-version>是系统发行编号，可以通过 uname -r 指令查看。

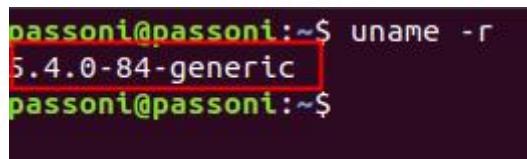


图 2-2-5 查看系统发行编号

在得到系统发行编号之后，我们就可以执行上面那条复制指令了。

```
passoni@passoni:~/Linux_3.x.x_4.x.x_VCP_Driver_Source$ sudo cp cp210x.ko /lib/modules/5.4.0-84-generic/kernel/drivers/usb/serial
passoni@passoni:~/Linux_3.x.x_4.x.x_VCP_Driver_Source$
```

图 2-2-6 复制 cp210x.ko 到指定目录

接下来继续在终端执行以下命令进行模块的载入：

sudo insmod /lib/modules/<kernel-version>/kernel/drivers/usb/serial/usbserial.ko

```
passoni@passoni:~/Linux_3.x.x_4.x.x_VCP_Driver_Source$ sudo insmod /lib/modules/5.4.0-84-generic/kernel/drivers/usb/serial/usbserial.ko
passoni@passoni:~/Linux_3.x.x_4.x.x_VCP_Driver_Source$
```

图 2-2-7 insmod 载入 usbserial.ko 模块

最后再执行 cp210x.ko 模块的载入命令

```
sudo insmod cp210x.ko
```

```
passoni@passoni:~/Linux_3.x.x_4.x.x_VCP_Driver_Source$ sudo insmod cp210x.ko
passoni@passoni:~/Linux_3.x.x_4.x.x_VCP_Driver_Source$
```

图 2-2-8 insmod 载入 cp210x.ko 模块

通过 lsmod|grep cp210x 命令查看已安装驱动，发现 cp210x 说明我们的驱动安装成功。

```
passoni@passoni:~$ lsmod|grep cp210x
cp210x           40960  0
usbserial        49152  1 cp210x
```

图 2-2-9 lsmod 查看已安装驱动

## 2.3 使用 ROS 包驱动 N10 雷达

### ① 将雷达连接至 Ubuntu 系统

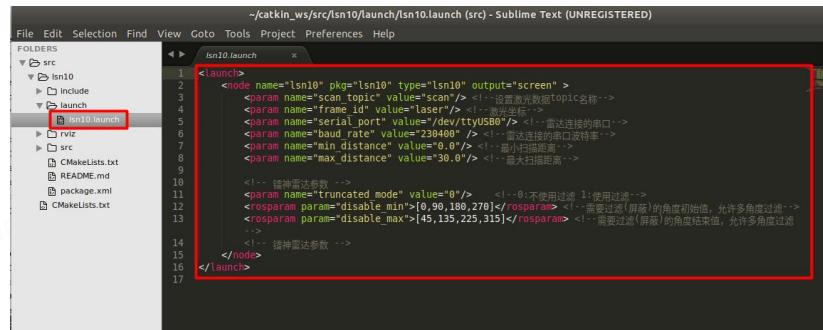
将雷达连接至 ubuntu 系统，通过 lsusb 命令查看已连接的 usb 设备，发现 CP210x 的设备说明连接成功。

```
passoni@passoni:~$ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 002: ID 10c4:a60 Cygnal Integrated Products, Inc. CP210X UART Bridge / myAVR mySmartUSB light
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 002: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 002 Device 003: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
passoni@passoni:~$
```

图 2-3-1 lsusb 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 launch 文件夹下的 lsn10.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。



```

~/catkin_ws/src/lsn10/launch/lsn10.launch (src) - Sublime Text (UNREGISTERED)

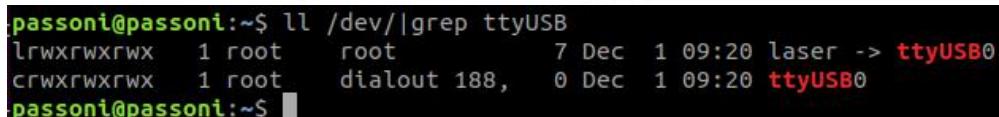
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  ▾ lsn10
    ▾ include
    ▾ launch
      lsn10.launch
    ▾ rviz
    CMakeLists.txt
    README.md
    package.xml
  CMakeLists.txt

1 <launch>
2   <node name="lsn10" pkg="lsn10" type="lsn10" output="screen" >
3     <param name="scan_topic" value="scan"/> <!-- 设置激光数据topic名称-->
4     <param name="frame_id" value="laser"/> <!-- 激光坐标-->
5     <param name="serial_port" value="/dev/ttyUSB0"/> <!-- 雷达连接的串口-->
6     <param name="baud_rate" value="230400"/> <!-- 雷达连接的波特率-->
7     <param name="min_distance" value="0.0"/> <!-- 最小扫描距离-->
8     <param name="max_distance" value="30.0"/> <!-- 最大扫描距离-->
9
10    <!-- 错神雷达参数 -->
11    <param name="truncated_mode" value="0"/> <!-- 0:不使用过滤 1:使用过滤-->
12    <rosparam param="disable_min">[0,90,180,270]</rosparam> <!-- 需要过滤(屏蔽)的角度初始值, 允许多角度过滤-->
13    <rosparam param="disable_max">[45,135,225,315]</rosparam> <!-- 需要过滤(屏蔽)的角度结束值, 允许多角度过滤-->
14
15  </node>
16 </Launch>
17

```

图 2-3-2 查看 lsn10.launch

用 lsn10.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 ttyUSB 命令查看。一般情况下串口名为 ttyUSB0。这里我通过创建串口别名将其串口名重映射为 laser（这一步不是必须的）。



```

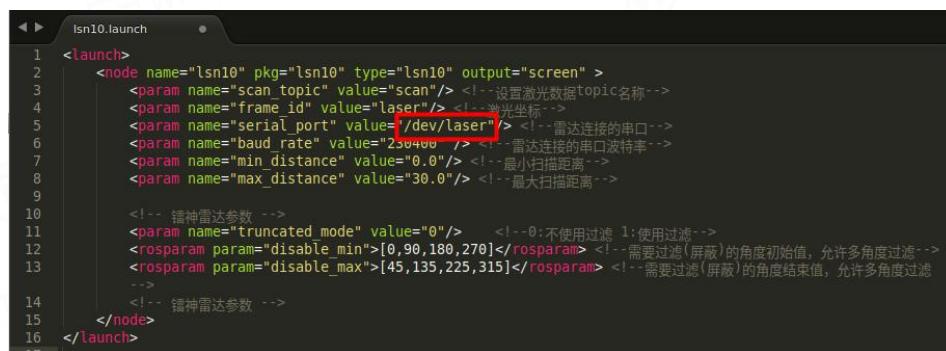
passoni@passoni:~$ ll /dev/|grep ttyUSB
lrwxrwxrwx 1 root root 7 Dec 1 09:20 laser -> /dev/ttyUSB0
crwrxrwxrwx 1 root dialout 188, 0 Dec 1 09:20 /dev/ttyUSB0
passoni@passoni:~$ 

```

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 lsn10.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 laser。



```

lsn10.launch

1 <launch>
2   <node name="lsn10" pkg="lsn10" type="lsn10" output="screen" >
3     <param name="scan_topic" value="scan"/> <!-- 设置激光数据topic名称-->
4     <param name="frame_id" value="laser"/> <!-- 激光坐标-->
5     <param name="serial_port" value="/dev/laser"/> <!-- 雷达连接的串口-->
6     <param name="baud_rate" value="230400"/> <!-- 雷达连接的波特率-->
7     <param name="min_distance" value="0.0"/> <!-- 最小扫描距离-->
8     <param name="max_distance" value="30.0"/> <!-- 最大扫描距离-->
9
10    <!-- 错神雷达参数 -->
11    <param name="truncated_mode" value="0"/> <!-- 0:不使用过滤 1:使用过滤-->
12    <rosparam param="disable_min">[0,90,180,270]</rosparam> <!-- 需要过滤(屏蔽)的角度初始值, 允许多角度过滤-->
13    <rosparam param="disable_max">[45,135,225,315]</rosparam> <!-- 需要过滤(屏蔽)的角度结束值, 允许多角度过滤-->
14
15  </node>
16 </Launch>
17

```

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 lsn10.launch 中，先将 truncated\_mode 参数置 1，表明需要屏蔽雷达角度。然后根据自己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，

如下图，代表屏蔽雷达  $0\text{~}45^\circ$ 、 $90^\circ\text{~}135^\circ$ 、 $180^\circ\text{~}225^\circ$ 、 $270^\circ\text{~}315^\circ$  角度范围。支持多角度屏蔽。



```

1 <launch>
2   <node name="lsn10" pkg="lsn10" type="lsn10" output="screen">
3     <param name="scan_topic" value="scan"/> <!-- 设置激光数据topic名称-->
4     <param name="frame_id" value="laser"/> <!-- 激光坐标-->
5     <param name="serial_port" value="/dev/laser"/> <!-- 雷达连接的串口-->
6     <param name="baud_rate" value="230400"/> <!-- 雷达连接的串口波特率-->
7     <param name="min_distance" value="0.0"/> <!-- 最小扫描距离-->
8     <param name="max_distance" value="30.0"/> <!-- 最大扫描距离-->
9
10    <!-- 错神雷达参数 -->
11    <param name="truncated_mode" value="1"/> <!-- 0:不使用过滤 1:使用过滤-->
12    <rosparam param="disable_min">[0,90,180,270]</rosparam> <!-- 需要过滤(屏蔽)的角度初始值, 允许多角度过滤-->
13    <rosparam param="disable_max">[45,135,225,315]</rosparam> <!-- 需要过滤(屏蔽)的角度结束值, 允许多角度过滤-->
14    <!-- 错神雷达参数 -->
15  </node>
16 </launch>
17

```

图 2-3-5 ROS 中 N10 角度屏蔽

有一点需要注意的是，在 ROS 中，逆时针为正，所以在 ros 中对应的雷达角度及坐标系如图所示。

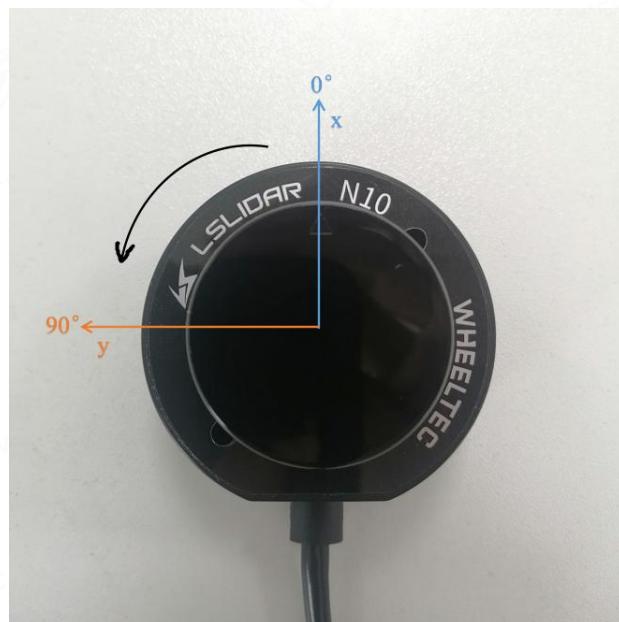


图 2-3-6 ROS 中 N10 角度及坐标系

#### 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 lsn10 lsn10.launch

成功启动后终端会出现以下字样。

```
/home/passoni/catkin_ws/src/lxn10/launch/ln10.launch http://localhost:11311
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
* /ln10/frame_id: laser
* /ln10/max_distance: 30.0
* /ln10/min_distance: 0.0
* /ln10/scan_topic: scan
* /ln10/serial_port: /dev/ttyUSB0
* /ln10/truncated_mode: 0
* /rosdistro: melodic
* /rosversion: 1.14.10

NODES
/
  lsn10 (lsn10/ln10)

auto-starting new master
process[master]: started with pid [2690]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 092093fa-b95a-11ec-95f8-000c2948ab90
process[rosout-1]: started with pid [2701]
started core service [/rosout]
process[lsn10-2]: started with pid [2708]
port = /dev/ttyUSB0, baud_rate = 230400
open_port /dev/ttyUSB0 OK !
```

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 lsn10.launch 启动雷达后，/lsn10 节点便会在 ros 中发布一个/scan 话题。

```
passoni@passoni:~$ rostopic info /scan
Type: sensor_msgs/LaserScan

Publishers:
* /lsn10 (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](http://docs.ros.org/en/api/sensor_msgs/html/msg/LaserScan.html)

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

header:
  seq: 1611
  stamp:
    secs: 1649655850
    nsecs: 406630537
  frame_id: "laser"
angle_min: 0.0
angle_max: 6.28318548203
angle_increment: 0.0140249673277
time_increment: 0.00021969748964
scan_time: 0.0992204770446
range_min: 0.0
range_max: 30.0
ranges: [0.6830000281333923, 0.6830000281333923, 0.7139999866485596, 0.7139999866485596, 0.7599999904632568, 0.7749999761581421, 0.791000008530668, 0.8220000267028809, 0.8529999852180481, 0.8690000176429749, inf, 0.8989999890327454, 0.930000071525574, 0.9610000252723694, 0.976999980926514, 1.0080000162124634, 1.037999876022339, 1.8910000324249268, 1.8760000467300419, 1.8760000467300419, 1.876000467300415, 0.3790000081062317, inf, 0.8989999861717224, inf, inf, 0.2980000078678131, 0.3149999976158142, 0.34700000286102295, 0.37900008081062317, 0.395000001072883606, 0.4269999861717224, 0.45899999141693115, 0.4909999966621399, 0.52499999761581421, 0.5400000214576721, 0.5720000267028809, 0.6050000000000000]
```

图 2-3-9 查看雷达话题数据

除了直接通过话题查看文本形式的雷达数据，我们也可以通过 rviz 来查看雷达的点云图像。直接在终端输入 rviz 并执行。打开 rviz 后，先将 Fixed Frame 后面对应的值修改为 lsn10.launch 中 frame\_id 对应的 value 值。



图 2-3-10 修改 Fixed Frame

之后点击 Rviz 左下角的 Add 按键，在弹出的窗口中点击 By topic 选中 /scan 话题下的 LaserScan 并点击 OK。

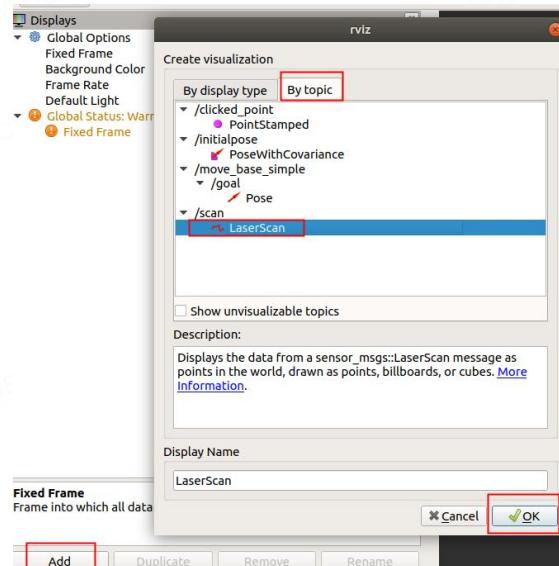


图 2-3-11 添加 LaserScan

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。

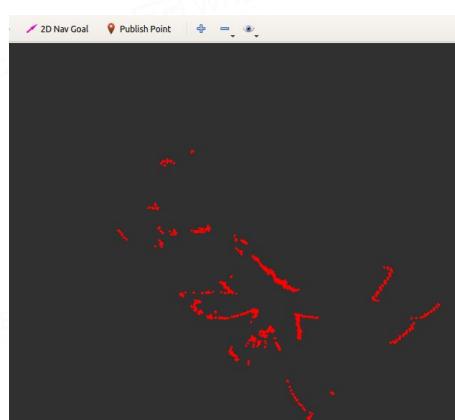


图 2-3-12 Rviz 显示雷达点云

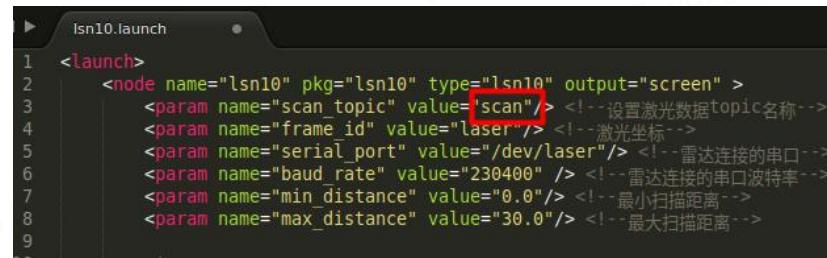
## ⑥ 雷达数据在 ros 的应用

当/lsn10 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。

```
f __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_si
```

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 lsn10.launch 中的 scan\_topic 的值来修改启动后发布的雷达话题名。



```
lsn10.launch
1 <launch>
2   <node name="lsn10" pkg="lsn10" type="lsn10" output="screen" >
3     <param name="scan_topic" value="scan" /> <!--设置激光数据topic名称-->
4     <param name="frame_id" value="laser"/> <!--激光坐标-->
5     <param name="serial_port" value="/dev/laser"/> <!--雷达连接的串口-->
6     <param name="baud_rate" value="230400" /> <!--雷达连接的串口波特率-->
7     <param name="min_distance" value="0.0"/> <!--最小扫描距离-->
8     <param name="max_distance" value="30.0"/> <!--最大扫描距离-->
```

图 2-3-14 修改雷达话题名