

雷达建图测试

1.环境介绍

系统版本: ubuntu20.04

ROS版本: Noetic

旭日x3派作为从机, 虚拟机作为主机

2.环境配置

注: 如果已经进行过旭日x3派的环境配置的可以跳过旭日x3派环境配置, 直接从ubuntu20.04虚拟机环境配置开始

旭日x3派环境配置

1.安装Noetic, 这一步可以自行百度

2.配置YDLIDAR-X3雷达sdk环境

2.1 直接到这个地址去下载sdk文件 <https://github.com/YDLIDAR/YDLidar-SDK>,也可以使用git命令下载 `git clone https://github.com/YDLIDAR/YDLidar-SDK.git`

2.2 安装cmake

```
sudo apt install cmake pkg-config
```

2.3编译和安装

进入YDLidar-SDK文件夹后如果没有build文件夹可以先创建一个build文件夹然后利用下面的命令编译安装!

```
mkdir build #创建build文件夹, 如果原本就有的可以不用创建, 把build文件夹里的文件删除就可以了
cd build
cmake ..
make
sudo make install #这个条命令也会自动安装python库
```

3.下载编译YDLIDAR-X3雷达官方提供的ROS包

3.1官方提供的ros包可以直接在这里下载: https://github.com/YDLIDAR/ydlidar_ros_driver,也可以用Git命令下载 `git clone https://github.com/YDLIDAR/ydlidar_ros_driver.git`

3.2创建ros工作空间并编译

```
mkdir -p ~/ros_test/src #这个~/ros_test/src表示会在当前用户的主目录下创建一个ros_test的文件夹, 还会在这个文件夹里创建一个src的文件夹
```

文件夹创建好, 把上面下载的ydlidar_ros_driver文件夹复制到src文件夹下

然后在ros_test文件夹下打开终端输入 `catkin_make` 然后按回车就会开始编译了

如果提示下面这个错误就是上面的sdk没有安装好，需要重新安装sdk

```
CMake Error at ydlidar_ros_driver/CMakeLists.txt:9 (find_package):
  By not providing "Findydlidar_sdk.cmake" in CMAKE_MODULE_PATH this project
  has asked CMake to find a package configuration file provided by
  "ydlidar_sdk", but CMake did not find one.

Could not find a package configuration file provided by "ydlidar_sdk" with
any of the following names:

  ydlidar_sdkConfig.cmake
  ydlidar_sdk-config.cmake

Add the installation prefix of "ydlidar_sdk" to CMAKE_PREFIX_PATH or set
"ydlidar_sdk_DIR" to a directory containing one of the above files.  If
"ydlidar_sdk" provides a separate development package or SDK, be sure it
has been installed.

-- Configuring incomplete, errors occurred!
See also "/home/tony/ydlidar_ws/build/CMakeFiles/CMakeOutput.log".
See also "/home/tony/ydlidar_ws/build/CMakeFiles/CMakeError.log".
Makefile:2112: recipe for target 'cmake_check_build_system' failed
make: *** [cmake_check_build_system] Error 1
Invoking "make cmake_check_build_system" failed
tony@ubuntu:~/ydlidar_ws$
```

编译成功后重新加载一下包的环境设置！

```
source ./devel/setup.sh
#当然也可以直接写进当前用户下的.bashrc文件中，这样就不用每次都需要手动加载环境了具体的方式为
echo "source ~/ros_test/devel/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

3.3绑定端口

```
chmod 0777 ~/ros_test/src/ydlidar_ros_driver/startup/*
sudo sh ~/ros_test/src/ydlidar_ros_driver/startup/initenv.sh
```

端口绑定完成后重新插入激光雷达，然后启动雷达测试一下

```
cd ~/ros_test/
source ./devel/setup.bash #重新加载环境变量，在.bashrc文件中配置了的可以直接启动雷达节点
#启动雷达测试
roslaunch ydlidar_ros_driver x2.launch
```

看到以下提示表示雷达启动成功

```

process[roscpp]: started with pid [4151]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 22dbb772-fe9f-11ec-8034-0011225ad20c
process[rosout-1]: started with pid [4166]
started core service [/rosout]
process[ydlidar_lidar_publisher-2]: started with pid [4169]
process[base_link_to_laser4-3]: started with pid [4170]
[ INFO] [1657272000.976910264]: YDLIDAR ROS Driver Version: 1.0.
YDLidar SDK initializing
YDLidar SDK has been initialized
[YDLIDAR]:SDK Version: 1.0.6
LiDAR successfully connected
[YDLIDAR]:Lidar running correctly ! The health status: good
LiDAR init success, Elapsed time 623 ms
[CYdLidar] Succeeded to start scan mode, Elapsed time 1062 ms
[YDLIDAR] Calc Sample Rate: 3K
[YDLIDAR] Fixed Size: 720
[YDLIDAR] Sample Rate: 3K
[YDLIDAR] Calc Sample Rate: 3K
[YDLIDAR] Fixed Size: 720
[YDLIDAR] Sample Rate: 3K
[YDLIDAR]:Single Fixed Size: 270
[YDLIDAR]:Sample Rate: 3K
[YDLIDAR INFO] Single Channel Current Sampling Rate: 3K
[YDLIDAR INFO] Now YDLIDAR is scanning .....
[YDLIDAR] Connection established in [/dev/ydlidar][115200]:
Firmware version: 2.1
Hardware version: 0
Model: F2
Serial: 2019020100002056

```

测试成功后可以使用ctrl+c结束运行

ubuntu20.04虚拟机环境配置

1.安装Noetic，这一步可以自行百度

2.配置ubuntu20.04雷达sdk环境（参考上面旭日x3派YDLIDAR-X3雷达sdk环境）

3.下载编译YDLIDAR-X3雷达官方提供的ROS包（参考上面旭日x3派YDLIDAR-X3雷达官方提供的ROS包）

4.下载laser_scan_matcher工具包

```

#用git命令下载工具包
git clone https://github.com/ccny-ros-pkg/scan_tools.git
#有兴趣的也可以自己去看看这个项目，项目地址：
https://github.com/CCNYRoboticsLab/scan_tools

```

```

#用下面的命令安装gmapping
sudo apt install ros-noetic-gmapping
#安装成功后进入ros_test文件夹
cd ~/ros_test
#编译
#如果在编译过程中提示找不到csm包可以用这个命令安装：sudo apt-get install ros-noetic-csm
catkin_make
#编译完成后重新加载一下
source ./devel/setup.bash

```

然后把下面的文件复制替换到 laser_scan_matcher 文件夹下的 demo 文件夹里的 demo_gmapping.launch 文件

```
<launch>

#### set up data playback from bag #####

<param name="/use_sim_time" value="false"/>

#### publish an example base_link -> laser transform #####

<node pkg="tf" type="static_transform_publisher" name="odom_to_base_footprint"

  args="0.0 0.0 0.0 0.0 0.0 0.0 /odom /base_footprint 40" />

<node pkg="tf" type="static_transform_publisher" name="robot_stat_publisher"
  args="0.0 0.0 0.0 0.0 0.0 0.0 /base_footprint /base_link 40" />

#### start rviz #####

<node pkg="rviz" type="rviz" name="rviz"
  args="-d $(find laser_scan_matcher)/demo/demo_gmapping.rviz"/>

#### start the laser scan_matcher #####

<node pkg="laser_scan_matcher" type="laser_scan_matcher_node"
  name="laser_scan_matcher_node" output="screen">

  <param name="fixed_frame" value = "odom"/>
  <param name="max_iterations" value="10"/>

  <param name="base_frame" value="base_link"/>
  <param name="use_odom" value="false"/>
  <param name="publy_pose" value="true"/>
  <param name="publy_tf" value="true"/>

</node>

#### start gmapping #####

<node pkg="gmapping" type="slam_gmapping" name="slam_gmapping"
output="screen">
  <param name="map_udpate_interval" value="1.0"/>
  <param name="maxUrange" value="5.0"/>
  <param name="sigma" value="0.1"/>
  <param name="kernelSize" value="1"/>
  <param name="lstep" value="0.15"/>
  <param name="astep" value="0.15"/>
  <param name="iterations" value="1"/>
  <param name="lsigma" value="0.1"/>
  <param name="ogain" value="3.0"/>
  <param name="lskip" value="1"/>
  <param name="srr" value="0.1"/>

```

```
<param name="srt" value="0.2"/>
<param name="str" value="0.1"/>
<param name="stt" value="0.2"/>
<param name="linearUpdate" value="1.0"/>
<param name="angularUpdate" value="0.5"/>
<param name="temporalUpdate" value="0.4"/>
<param name="resampleThreshold" value="0.5"/>
<param name="particles" value="10"/>
<param name="xmin" value="-5.0"/>
<param name="ymin" value="-5.0"/>
<param name="xmax" value="5.0"/>
<param name="ymax" value="5.0"/>
<param name="delta" value="0.02"/>
<param name="llsamplerange" value="0.01"/>
<param name="llsamplestep" value="0.05"/>
<param name="lasamplerange" value="0.05"/>
<param name="lasamplestep" value="0.05"/>
</node>

</launch>
```

多机通信配置

利用ifconfig命令查看旭日x3派和虚拟机的ip地址，注意需要旭日X3派和虚拟机是在同一个局域网下能正常ping通

我这里查到的虚拟机的ip地址是192.168.2.89

旭日x3派的的ip地址是：192.168.2.205

我这里是用旭日x3派作为主机，虚拟机为从机。

然后打开虚拟机和x3派用户文件夹下面的 .bashrc 文件

```
#打开.bashrc文件
sudo gedit ~/.bashrc

#虚拟机上在的.bashrc文件后面添加上面这些内容！
export ROS_HOSTNAME=192.168.2.89
export ROS_MASTER_URI=http://192.168.2.205:11311

#旭日x3派上在文件后面添加上面这些内容！
export ROS_HOSTNAME=192.168.2.205
export ROS_MASTER_URI=http://192.168.2.205:11311
```

添加保存好后关掉终端然后再重新打开个新的终端！

进行建图测试

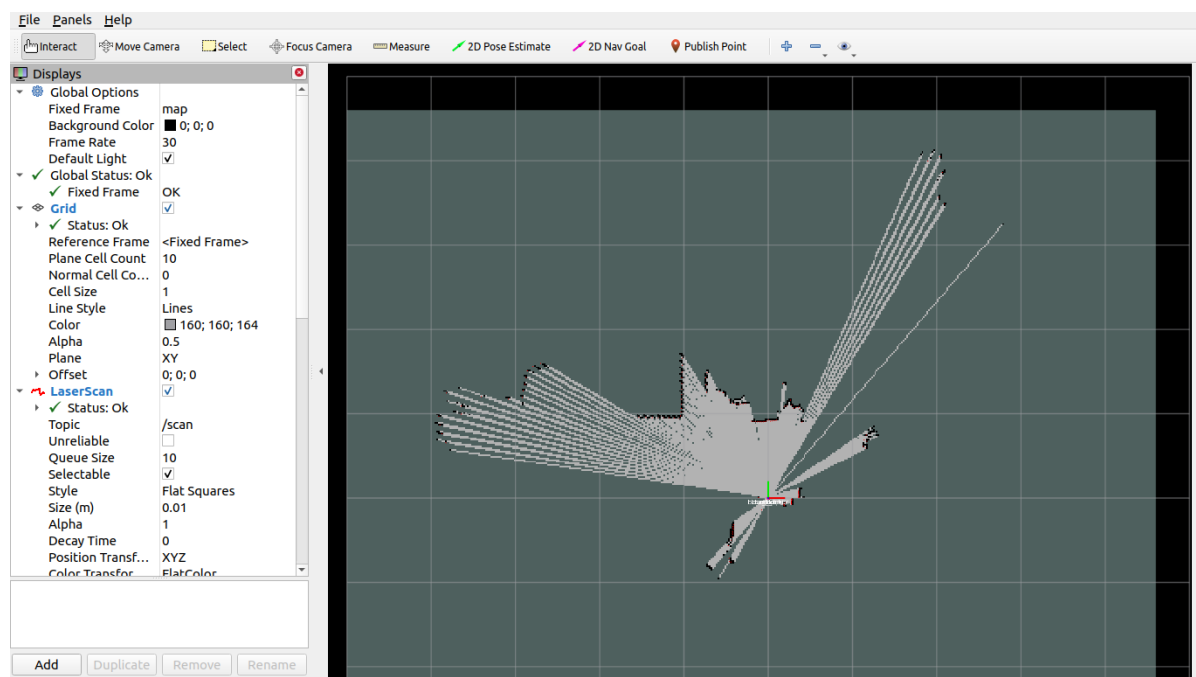
旭日x3派上的操作

```
#在终端上用cd命令进入ros_test工作空间
cd ~/ros_test/
source ./devel/setup.bash #重新加载环境变量，在.bashrc文件中配置了的可以直接启动雷达节点
#启动雷达测试
roslaunch ydlidar_ros_driver x2.launch
```

虚拟机上的操作

```
#在终端上用cd命令进入ros_test工作空间
cd ~/ros_test/
source ./devel/setup.bash #重新加载环境变量，在.bashrc文件中配置了的可以直接启动
demo_gmapping.launch文件
#启动建图测试
roslaunch laser_scan_matcher demo_gmapping.launch
```

运行后就可以看到虚拟机端的rviz上的画面了



注，如果运行`roslaunch laser_scan_matcher demo_gmapping.launch`命令后提示设备时间不同步的问题可手动同步一下时间具体方式如下：

在虚拟机端安装ntp: `sudo apt-get install ntp`

停止ntp服务 `sudo service ntp stop`

编辑ntp.conf文件 `sudo gedit /etc/ntp.conf`

在文件最后添加这几行然后保存

```
#使该网段内的机器可以与本机进行同步（本机IP 192.168.2.89）
restrict 192.168.2.89 mask 255.255.255.0 nomodify notrap
#local clock
server 127.127.1.0
#使本机的硬件时间和本机的ntp服务进行时间同步
fudge 127.127.1.0 stratum 10
```

重新启动ntp服务 `sudo service ntp restart`

在旭日旭日x3派上安装 ntpdate: `sudo apt-get install ntpdate`

进行时间同步: `sudo /usr/sbin/ntpdate 192.168.2.89`

然后再重新启动X3雷达和建图。