Guideline to perform LiDAR-recordings using ROS2 Humble

Goal of this Guideline

The main goal of this guideline is to give instructions on how to view rosbags using the program Foxglove. If you want to dive deeper, the guideline also explains how to record rosbags using ROS2 Humble. There is also an explanation on how to record 3D maps using the ROS2 package RTAB-Map. The target audience are sales representatives, with zero to little knowledge about ROS2. The multiScan100 will be used as an example in this guideline.

Table of Contents

Introduction and Glossary

This guideline will show you how you can visualize already existing rosbags using the program Foxgloe. If you want to dive deeper, this guideline will also give you the basic knowledge to perform recordings with ROS2 Humble yourself. There is also a video tutorial playlist available for ROS2 on the multiScan100. You can find the playlist at the following link.
https://video.sick.com/playlist/dedicated/0_rkklzpsj/0_9tgrc1zz
If there is a video tutorial available for a topic that is covered in this guideline, the specific link will be placed in the corresponding section

```
Text that is written in this format is a command that can be run in a terminal or a path that leads to a specific file
```

You can copy these commands with “ctrl +c” and paste them into the terminal by using “ctrl + shift + v”.

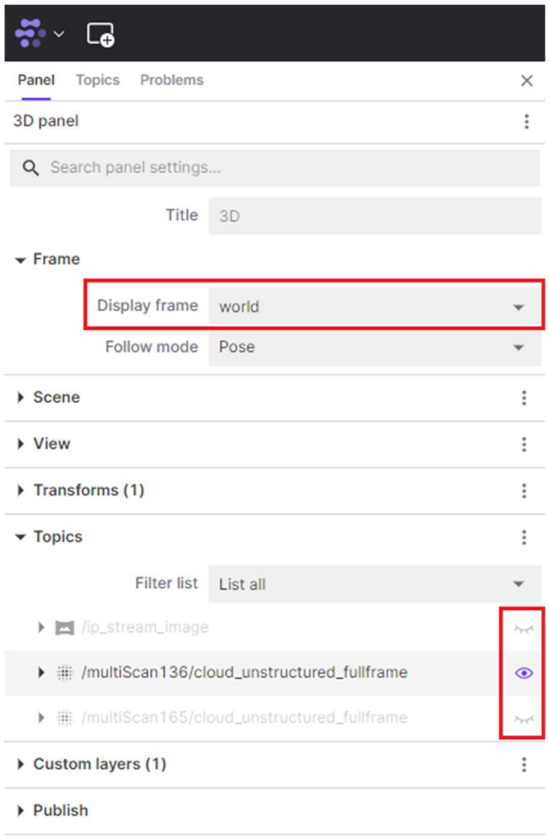
The most important terms are explained in the following glossary.

Term	Explanation
Foxglove	Free program with which you can view rosbags on every operating system you want.
ROS2 Humble Hawksbill	ROS stands for “Robot Operating System”. Humble Hawksbill is the specific distribution that is used for this guideline.
sick_scan_xd	This is the SICK driver for ROS2. It enables communication between the sensor and ROS2
Rosbag	File which can contain recordings of one or multiple sensors and cameras. Has the file formats .bag (ROS1) and .db3 (ROS2)
RTAB-Map	RTAB-Map is a downloadable ROS2 package for SLAM (Simultaneous Localization and Mapping). It can be used for creating 3D maps.
Linux Ubuntu	Linux Ubuntu is the OS (Operating System) that is used in this tutorial.
Terminal	The Terminal is the console used in Linux. It can be used to run commands.

View Rosbags on every operating system

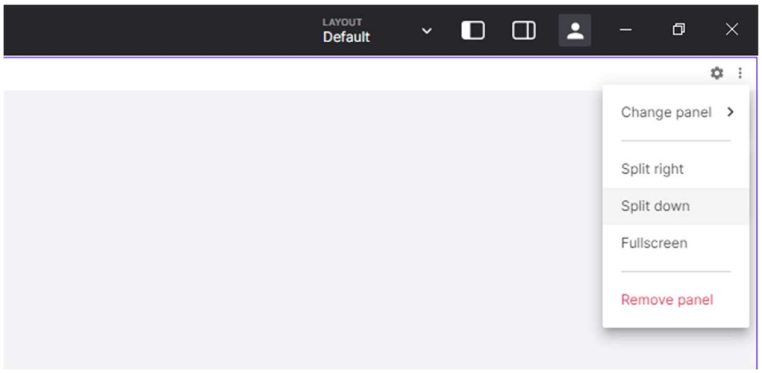
The visualization software to view rosbags is Foxglove and can be used on Windows, macOS, Linux and the Web. You can download Foxglove at the following link.
https://foxglove.dev/download

To have a look at a certain rosbag open Foxglove and click on “Open local file”. Now select the .bag (ROS1) or .db3 (ROS2) file you want to visualize. There is the possibility that not everything is configured correctly from the beginning and you see no data when opening the file.



Make sure that the Display frame is set to “world” and that all the topics you want to see are toggled on. Start to play back the rosbag and now you should see the point cloud.

You can show all the point clouds in one panel or you open a new one to show two point clouds beside each other or add a camera image.



You can add or remove panels in the top right corner. Once you have all the panels you like, you can edit them with the help of the settings bar on the left side. Here you can change the perspective in which you look on the data or edit the point cloud directly like shown in the picture below.

Installation ROS2 Humble Hawksbill

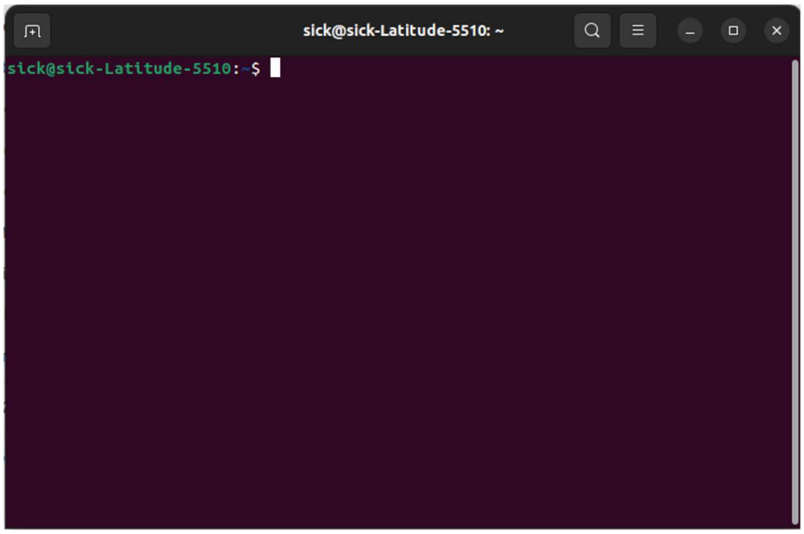
ROS2 Humble can be installed directly from the ROS2 website.

Installation on Ubuntu 22.04:
https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html

Follow the ROS2 installation guide. The installation guide is very detailed and can be followed step by step. There will also be a short example you can try to check if ROS2 is installed correctly.

You can install ROS2 Humble also on Ubuntu 20.04, although the sick_scan_xd only works with Ubuntu 22.04. Please check you Ubuntu Version like shown below.

Open a new terminal by clicking on the terminal icon or press “ctrl + alt + t”. Your terminal should look like the following picture.



To check your Ubuntu version type the following command into the terminal:

```
lsb_release -a
```

Make sure that your system runs Ubuntu 22.04. Otherwise you’ll have to update your Ubuntu Version before continuing with the guideline.

Installation sick_scan_xd

Video Tutorials:
https://video.sick.com/playlist/dedicated/0_rkklzpsj/0_ny9gfvui

https://video.sick.com/playlist/dedicated/0_rkklzpsj/0_mnpxklv3
https://video.sick.com/playlist/dedicated/0_rkklzpsj/0_obkmxq6w

To install the ROS2 Driver from SICK called sick_scan_xd you can follow the installation guide on the SICK github website.

Installation on Linux:
https://github.com/SICKAG/sick_scan_xd/blob/develop/INSTALL-ROS2.md

You can follow the installation guide **Install prebuilt binaries**, which should be the easiest way to install sick_scan_xd on Linux. In case that doesn't work you can try to follow the installation guide **Build from sources**.

If you have sick_scan_xd already installed and want to update it, you can remove it with the commands

```
sudo apt-get remove ros-humble-sick-scan-xd
sudo apt autoremove
```

and install it again afterwards.

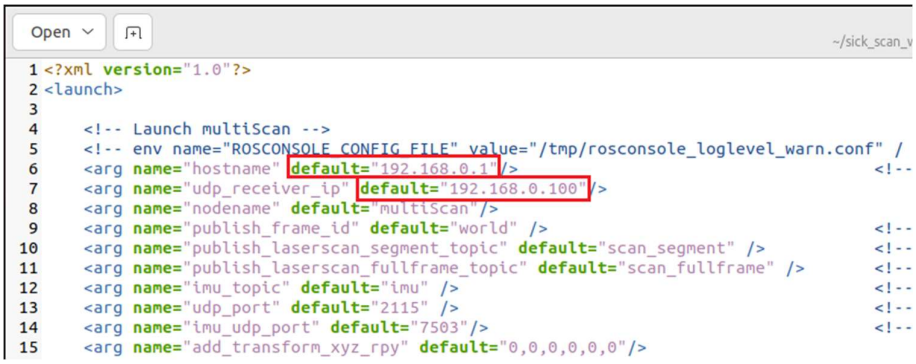
There are some things you can configure, that will make launching the sick_scan_xd driver easier. You can already configure which IP address will be used when launching the driver. To do this open the path to the sick_scan_xd launch files. The path should look similar to this.

```
[PATH TO YOUR ROS FOLDER]/install/sick_scan_xd/share/sick_scan_xd/launch
```

In this example the path looks like this:

```
~/sick_scan_ws/install/sick_scan_xd/share/sick_scan_xd/launch
```

You can open the specific launch file for each sensor now. For this example we will open the launch file of the multiScan called "sick_multiscan.launch". Here you can configure which IP address should be used so you don't have to type it every time you start the driver.



How to source your ROS2 environment

If you want to start working with ROS2, you have to source the ROS2 environment first. This means you have to start it, otherwise commands will not be recognized when typed into the terminal. You have two possibilities to do that. The first one is to manually source every time you open a new terminal. The other possibility is to automate it so its will be automatically sourced every time you open a new terminal.

How to source it manually:
To source it manually, run the following command every time you open a new terminal.
`source [PATH]/setup.bash`
Replace `[PATH]` with your individual path to the setup.bash file. This might differ depending on where you have ROS2 installed.

Some examples could be:
```
source opt/ros/humble/install/setup.bash
source sick_scan_ws/install/setup.bash
```

How to source it automatically:
Video Tutorial:
https://video.sick.com/playlist/dedicated/0_rkklzpsj/0_9tgrc1zz

To source it automatically you have to add the source command to your .bashrc file. This ensures that the source command is run every time you open a new terminal.

To open the .bashrc file run the following command.
`gedit ~/.bashrc`

Then add the command
`source [PATH]/setup.bash`
to the bottom of this file. Replace `[PATH]` with your individual path to the setup.bash file. This might differ depending on where you have ROS2 installed.

Some examples could be:
```
source opt/ros/humble/install/setup.bash
source sick_scan_ws/install/setup.bash
```

Save the file and close it again. It should look similar to this.

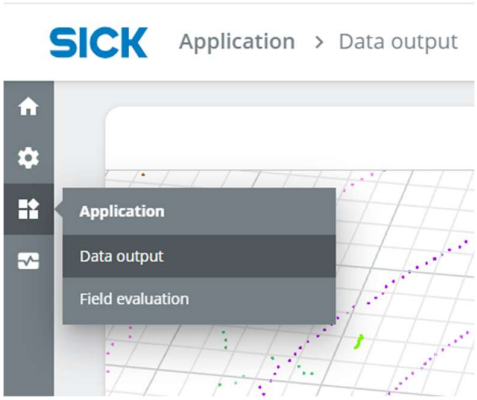
```
104 if [ -f ~/.bash_aliases ]; then
105     . ~/.bash_aliases
106 fi
107
108 # enable programmable completion features (you don't need to enable
109 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112   if [ -f /usr/share/bash-completion/bash_completion ]; then
113     . /usr/share/bash-completion/bash_completion
114   elif [ -f /etc/bash_completion ]; then
115     . /etc/bash_completion
116   fi
117 fi
118
119 source sick_scan_ws/install/setup.bash
```

Save the file and close it again. Now ROS2 sources automatically every time you open a new terminal.

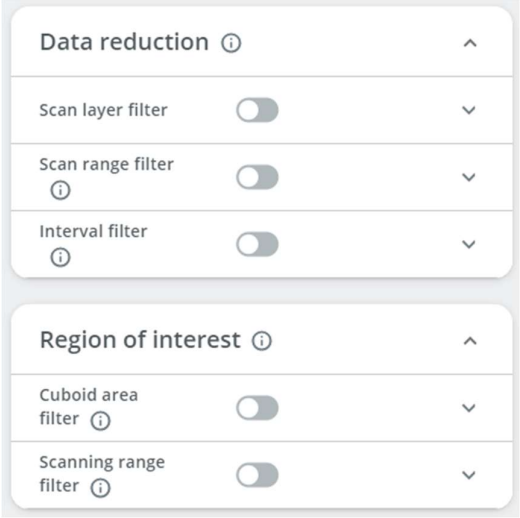
Configuring Sensor Data Output

It is also possible to configure the data coming from the sensors. A reason to do this could be a specific mounting of the sensors or the combination of two or more sensors. There are two ways you can influence the output of the sensor.

The first and easiest way is by using SOPASair. Go to the menu data output.

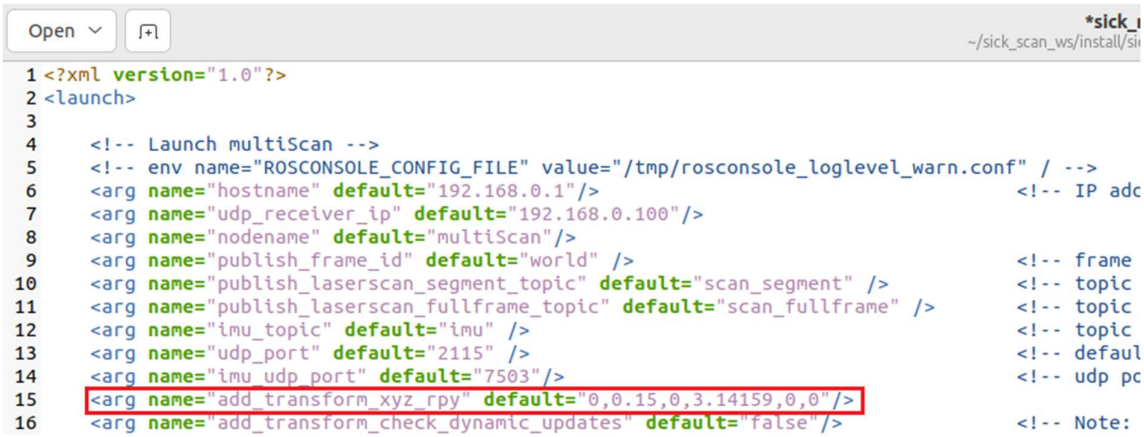


Here you can configure things like the scan layers, scan range, cuboid area and other stuff.



To save the adjustments you have to login and click on the save button.

The other way of configuring the data output is by editing the launch files of your sensor. We will go through that with one example. Let's say you have to sensors mounted upside down and 15cm besides each another. Now you have the problem that the point cloud of the two sensors doesn't match in rviz because of the different mounting and is shown upside down. Adjusting the mounting in SOPASair won't help here because it doesn't influence the output data. Open your launch file. In this example we will open the launch file of the multiScan.
`[PATH]//install/sick_scan_xd/share/sick_scan_xd/launch/sick_multiscan.launch`
Replace `[PATH]` with the path to your ROS2 workspace. Now we look at the parameters "add_transform_xyz_rpy". With this parameter we can change the coordinate system and roll, pitch and yaw. The xyz-values are given in meters and the rpy-values in radiant.



Like shown in the picture we will edit the y-value with “0.15”, which will set the point cloud 15cm to the right. Now the point clouds of our sensors will have their origin in the same place. To turn the point cloud upside down we edit roll with the value of 3.14159 (π), which represents the value 180°. In the launch file of the other sensor, we will only have to change the roll-value like we’ve just done. Now the point clouds are shown with the same origin and aren’t turned upside down because of the mounting. This is only one way of configuring the data output with the launch file. There are many other parameters that can be changed. We will not cover them in this guideline because it would take too long.

How to record rosbags

Video Tutorial:
https://video.sick.com/playlist/dedicated/0_rkklzpsj/0_45yrm6yg

Step 1: Open a new terminal and source the ROS2 environment
Open a new terminal and source your ROS2 environment as described in the chapter “How to source your ROS2 environment”.

Step 2: Launch the sick_scan_xd driver
The launch commands for all the available sensors can be found on the SICK GitHub website.
https://github.com/SICKAG/sick_scan_xd/blob/develop/USAGE.md

As an example we will use the multiScan165. You can use the commands for multiScan136, which will also work with the multiScan165. Launch the driver with the following command.

```
ros2 launch sick_scan_xd sick_multiscan.launch.py hostname:=[IP ADDRESS]
udp_receiver_ip:=[IP ADDRESS]
```

Change the IP addresses to the ones you’re using. If you already configured the IP address in the launch file you can also use the command

```
ros2 launch sick_scan_xd sick_multiscan.launch.py
```

to start the driver.

It is important that the UDP receiver you are using has an IP address that is in the same range as the sensor you’re using. If your sensor has the IP address “192.168.0.1” for example, then set

your UDP receiver to the IP address “192.168.0.100”.

```
Open                                                    ~/sick_scan_ws/i
1 <?xml version="1.0"?>
2 <launch>
3
4     <!-- Launch multiScan -->
5     <!-- env name="ROSCONSOLE_CONFIG_FILE" value="/tmp/rosconsole_loglevel_warn.conf" / --
6     <arg name="hostname" default="192.168.0.1"/>                                    <!-- I
7     <arg name="udp_receiver_ip" default="192.168.0.100"/>
8     <arg name="nodename" default="multiScan"/>
9     <arg name="publish_frame_id" default="world" />                                 <!-- f
```

If you want to record multiple sensors at the same time, simply launch each sensor in a new terminal. Don’t forget to source your ROS2 environment in every new terminal.

Step 3: Open a new terminal and source the ROS2 environment

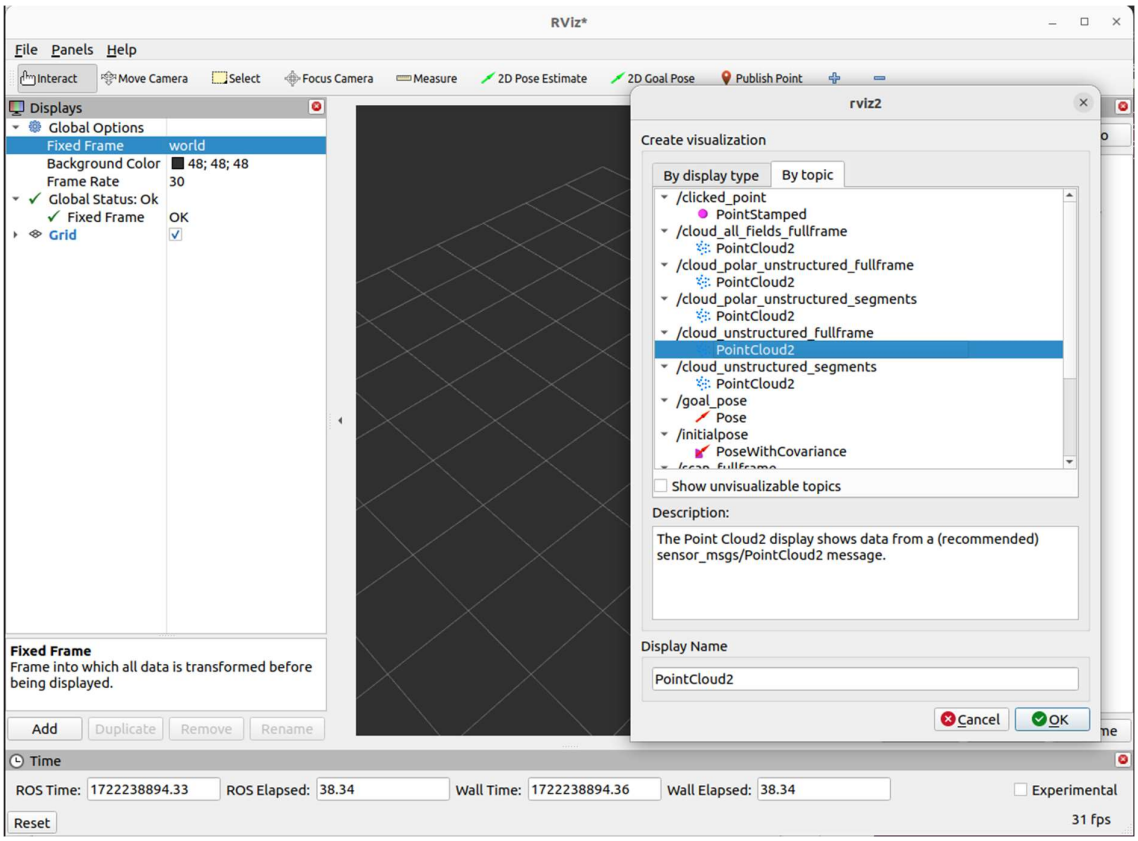
Open a new terminal and source your ROS2 environment as described in the chapter “How to source your ROS2 environment”.

Step 4: Start rviz2

Start rviz2 with the following command.

```
rviz2
```

To see the scan data you have to make some adjustments in rviz2. The first one is to change the Fixed Frame to “world”. Then click the “Add” button and select “PointCloud2” from the topic “/cloud_unstructured_fullframe”



Now you should be able to see the scan data in rviz2.

Step 5: Open a new terminal and source the ROS2 environment

Open a new terminal and source your ROS2 environment as described in the chapter "How to source your ROS2 environment".

Step 6: Choose the folder for saving your rosbag

Go to the folder you want to save the rosbag in. The following commands will help you navigate in the terminal.

Action	How to do it
Create a new folder	`mkdir [NAME]`
Go to a different folder:	`cd [PATH TO THE FOLDER]`
Go to the folder above:	`cd ../`

The terminal will show you in which folder you're currently in.

Step 7: Start recording a rosbag

There are different commands to start a rosbag record. All the important ones are shown in the following table.

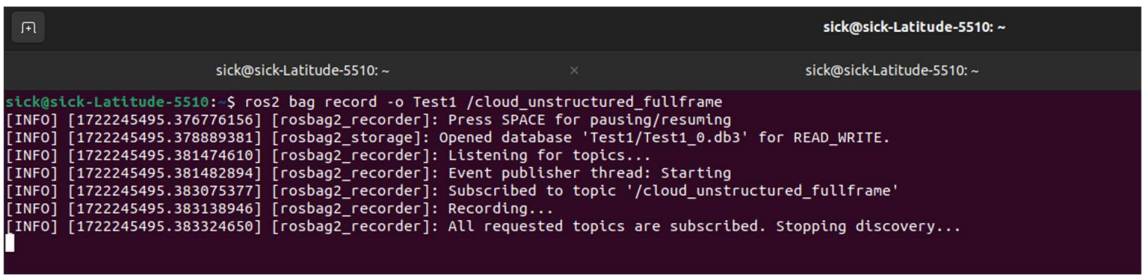
Action	How to do it
Open a list with all the commands you can use to record a rosbag	`ros2 bag record --help`
Record to one specific topic	`ros2 bag record [NAME OF THE TOPIC]`
Record to one specific topics and give it a name	`ros2 bag record -o [NAME OF THE ROSBAG] [NAME OF THE TOPIC]`
Record multiple topics	`ros2 bag record [NAME OF TOPIC1] [NAME OF TOPIC2] [NAME OF TOPIC3]`
Record all active topics	`ros2 bag record -a`

You can use the following command to see which topics are currently active.

```
ros2 topic list
```

Example:

```
ros2 bag record -o Test1 /cloud_unstructured_fullframe
```



If the rosbag is recorded successfully the terminal should look like that. You can go back to the rviz2 window to look at the data you're currently recording.

Step 8: Stop recording a rosbag

To stop the recording of a topic and save the rosbag go to the terminal in which your rosbag is recorded. Now press the “ctrl + c”. This will cancel the current task performed by the terminal. The rosbag is saved to the folder you are currently in.

Step 9: Play back your rosbag

To play back a rosbag make sure that rviz2 is open and no sensor is currently publishing data. You can stop a sensor from publishing data by going to the terminal in which you started the launch file and press “ctrl + c”. Now, go the folder your rosbag is saved in. Use the following command to play back a rosbag.

```
ros2 bag play [NAME OF THE ROSBAG]
```

Installation RTAB-Map

RTAB-Map can be installed by using the following command.

```
sudo apt install ros-humble-rtabmap-ros
```

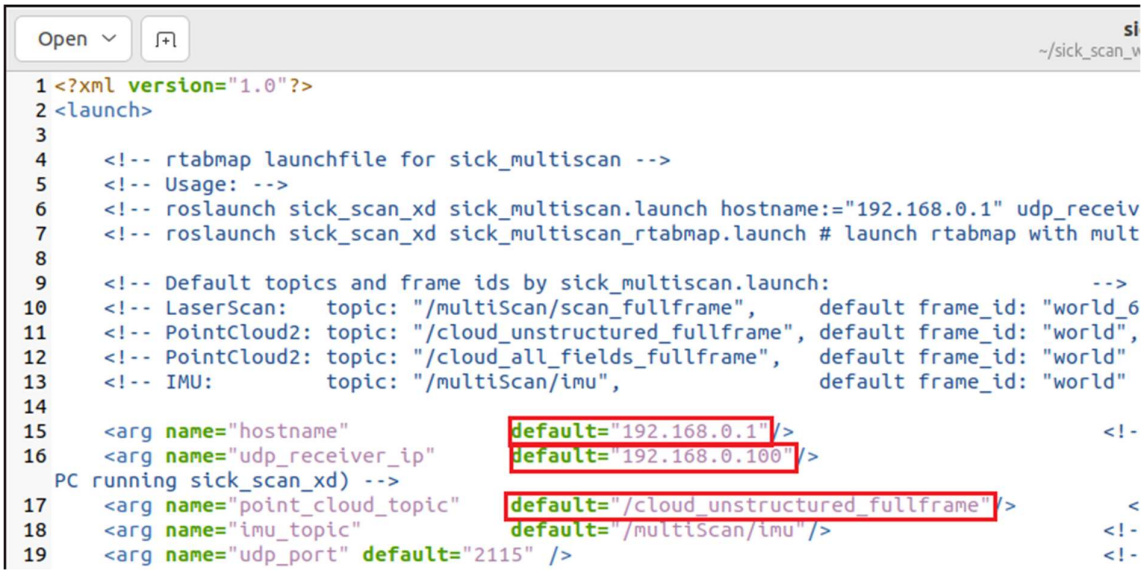
You can also configure the launch file for the RTAB-Map package. To do this open the path to the sick_scan_xd launch files. The path should look similar to this.

```
[PATH TO YOUR ROS FOLDER]/install/sick_scan_xd/share/sick_scan_xd/launch
```

In this example the path looks like this:

```
~/sick_scan_ws/install/sick_scan_xd/share/sick_scan_xd/launch
```

Now open the launch file ”sick_multiscan_rtabmap.launch”. You can add the IP addresses you want to use. Also change the “point_cloud_topic” to “/cloud_unstructured_fullframe” as shown in the picture below.



How to record 3D-Maps using RTAB-Map

Step 1: Start RTAB-Map

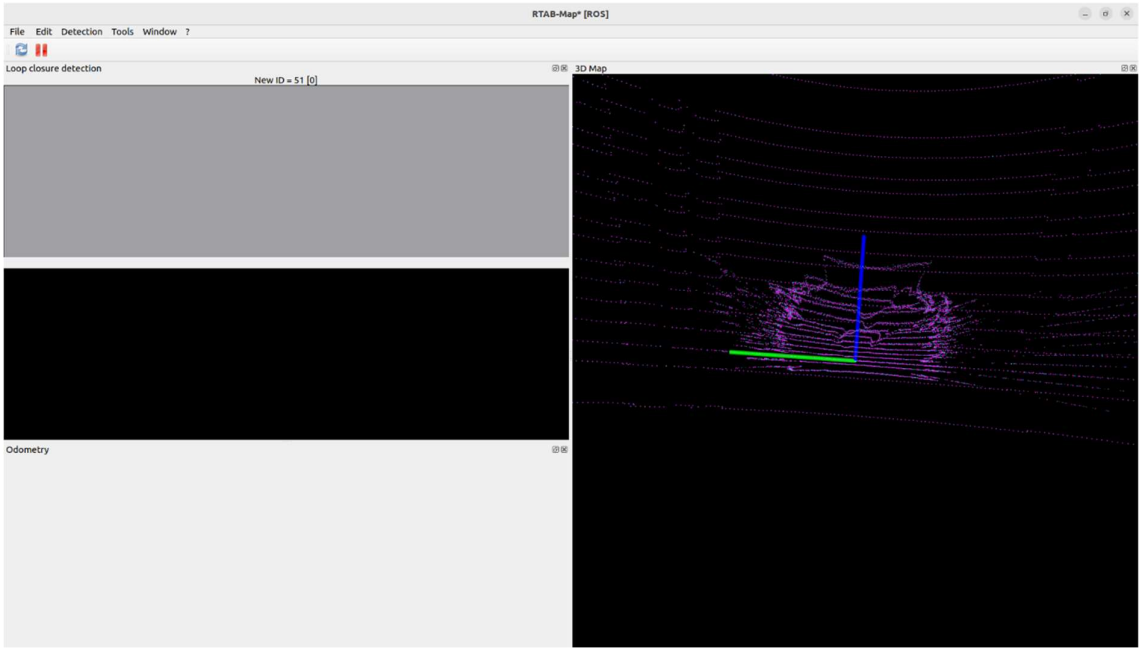
You can start RTAB-Map with the following command.
ros2 launch sick_scan_xd sick_multiscan_rtabmap.launch.py hostname:=192.168.0.1 udp_receiver_ip:=192.168.0.100

If you already configured the launch file with the correct IP addresses, you can also use the command.
ros2 launch sick_scan_xd sick_multiscan_rtabmap.launch.py

Now RTAB-Map should be open and ready to start.

Step 2: Create the map

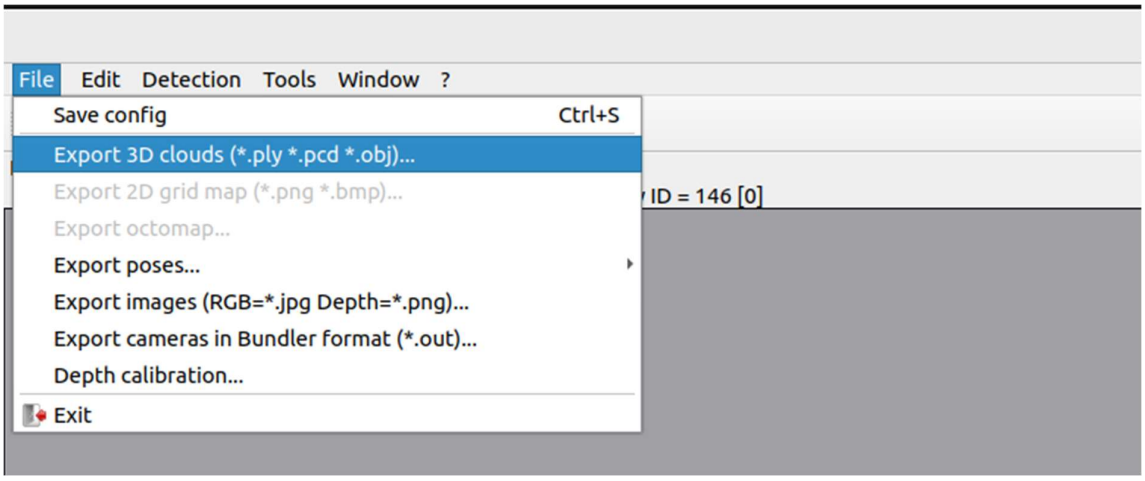
If RTAB-Map is started correctly you should already see the point cloud of the multiScan100.



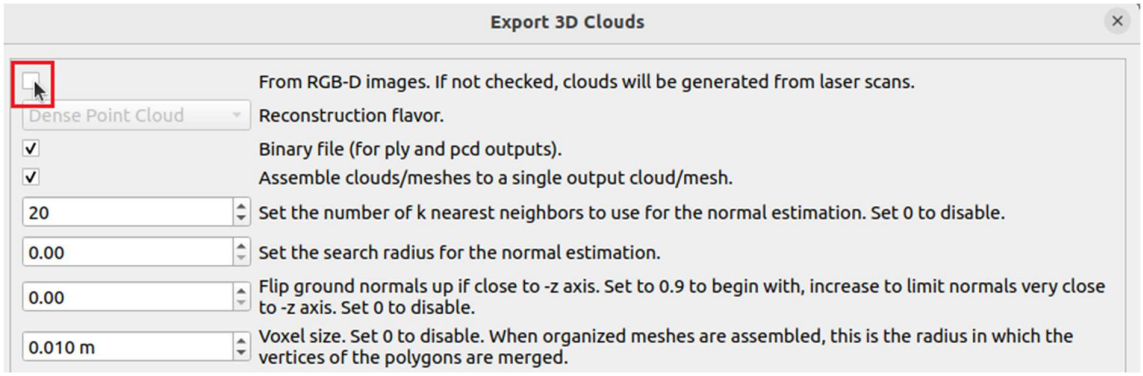
Now you can walk/drive around to create a 3D map of your surroundings. You can also zoom out to have a better view of the map you're currently creating. Once you are finished and want to stop recording, you can press the red pause button and no more information will be added to the map.

Step 3: Save the map

After stopping the recording with the red pause button, you can save your map. Start by exporting the 3D clouds.



Now click on the first box to deactivate the dense point cloud.



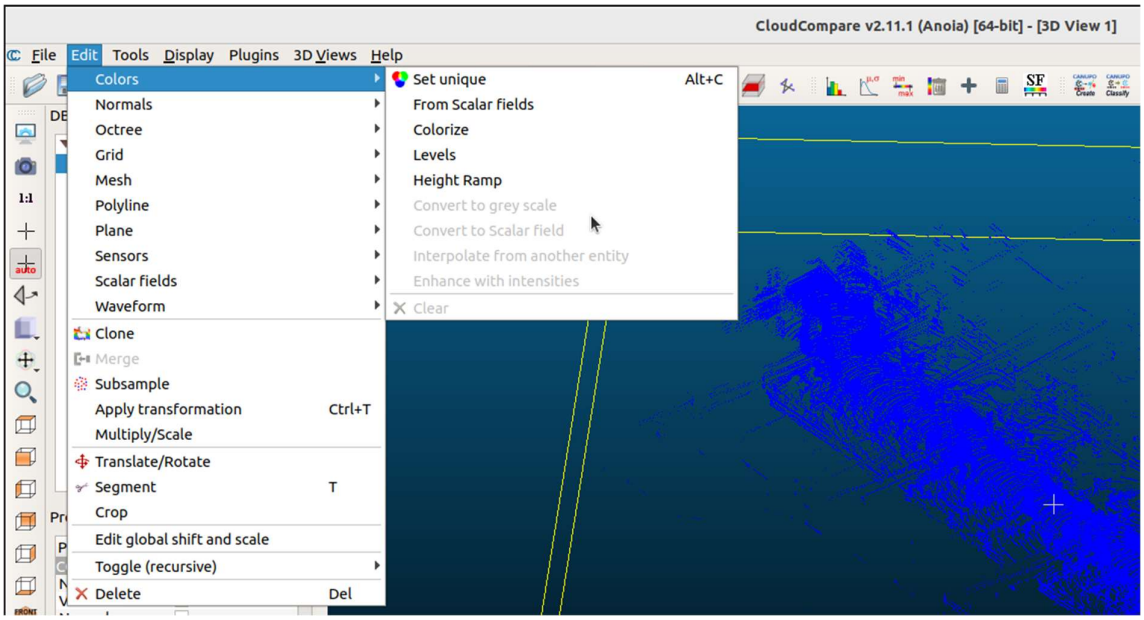
After that you can click on save. It is important that you save it as a pcd-file. Type ".pcd" at the end of the file name, for example "Test1.pcd".

Step 4: View the map

To have a look on the map you can use the visualization tool CloudCompare. You can download it at the following link.
https://www.danielgm.net/cc/

Alternatively, you can download it via Ubuntu Software.

Once you've installed it you can open your pcd-files and have a look at them. For better visualization select the map and then change the color how you like it.


CloudCompare v2.11.1 (Anoia) [64-bit] - [3D View 1]
File
Edit
Tools
Display
Plugins
3D Views
Help
Colors
Normals
Octree
Grid
Mesh
Polyline
Plane
Sensors
Scalar fields
Waveform
Clone
Merge
Subsample
Apply transformation
Ctrl+T
Multiply/Scale
Translate/Rotate
Segment
T
Crop
Edit global shift and scale
Toggle (recursive)
Delete
Del
Set unique
Alt+C
From Scalar fields
Colorize
Levels
Height Ramp
Convert to grey scale
Convert to Scalar field
Interpolate from another entity
Enhance with intensities
Clear