CygLidar D2 SLAM 2D Guide (EN)

Update: 24-05-16

Table of Contents

- **Introduction**

- **CygLiDAR D2**

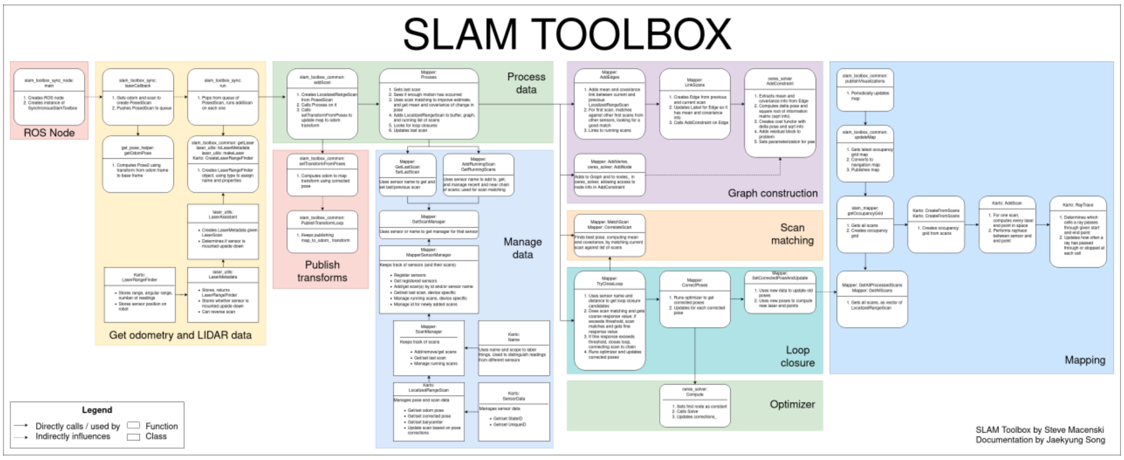
CygLiDAR D2 manual

https://www.cygbot.com/downloads

CygLiDAR D2 ROS Driver

https://github.com/CygLiDAR-ROS/cyglidar_d2

- **Slam Toolbox**



Slam Toolbox ROS Driver

https://github.com/SteveMacenski/slam_toolbox/tree/ros2

- **CygLiDAR D2 + Slam Toolbox + Create3 Operation Video**

https://www.youtube.com/watch?v=_G9JiKEF0ck

2. Configuration Settings

Test based on below :

Jetson Nano Dev Kit 4GB Ubuntu 20.04 / ROS2 Foxy

(cyglidar_d2_publisher)

GL62 7RD(Remote PC) Ubuntu 22.04 / ROS2 humble

- GeForce® GTX 1050 with 2GB GDDR5
- Intel® i5-7300HQ CPU
- Memory 8GB, DDR4-2400

(slam_toolbox / robot_description(modeling create3) / teleop_twist_keyboard)

(Robot)Create3 – **ROS2 Galactic (G4.4)**

(odometry)

- **Installation of required software and libraries**

Installation of Create3 ROS2(**humble**)

https://github.com/iRobotEducation/create3_examples

https://github.com/iRobotEducation/create3_sim

```
mkdir -p create3_examples_ws/src
cd create3_examples_ws/src
git clone https://github.com/iRobotEducation/create3_examples.git --branch humble
git clone https://github.com/iRobotEducation/create3_sim
cd ..
rosdep install --from-path src --ignore-src -yi
colcon build
```

Before starting to build packages, you must enter the commands below.

sudo apt install ros-humble-slam-toolbox

sudo apt install ros-humble-irobot-create-msgs

sudo apt install ros-humble-control-msgs

sudo apt install ros-humble-joint-state-publisher

3. Configuration and Running ROS Packages for SLAM Mapping

- Slam Toolbox's Configuration (**mapper_params_online_async.yaml**)

```yaml
slam_toolbox:
  ros__parameters:
    # Plugin params
    solver_plugin: solver_plugins::CeresSolver
    ceres_linear_solver: SPARSE_NORMAL_CHOLESKY
    ceres_preconditioner: SCHUR_JACOBI
    ceres_trust_strategy: LEVENBERG_MARQUARDT
    ceres_dogleg_type: TRADITIONAL_DOGLEG
    ceres_loss_function: None

    # ROS Parameters
    odom_frame: odom
    map_frame: map
    base_frame: base_footprint
    scan_topic: /scan
    mode: mapping #localization

    debug_logging: false
    throttle_scans: 1
    transform_publish_period: 0.02
    map_update_interval: 1.0
    resolution: 0.02
    max_laser_range: 7.0
    minimum_time_interval: 0.5
    transform_timeout: 0.2
    tf_buffer_duration: 15.0
    stack_size_to_use: 160000000
    enable_interactive_mode: true

    # General Parameters
    use_scan_matching: true
    use_scan_barycenter: true
    minimum_travel_distance: 0.2
    minimum_travel_heading: 0.5
    scan_buffer_size: 10
    scan_buffer_maximum_scan_distance: 4.0
```

```
link_match_minimum_response_fine: 0.1
link_scan_maximum_distance: 2.5
loop_search_maximum_distance: 3.0
do_loop_closing: true
loop_match_minimum_chain_size: 10
loop_match_maximum_variance_coarse: 3.0
loop_match_minimum_response_coarse: 0.4
loop_match_minimum_response_fine: 0.45

# Correlation Parameters - Correlation Parameters
correlation_search_space_dimension: 0.5
correlation_search_space_resolution: 0.01
correlation_search_space_smear_deviation: 0.1

# Correlation Parameters - Loop Closure Parameters
loop_search_space_dimension: 8.0
loop_search_space_resolution: 0.05
loop_search_space_smear_deviation: 0.03

# Scan Matcher Parameters
distance_variance_penalty: 0.5
angle_variance_penalty: 0.001

fine_search_angle_offset: 0.00349
coarse_search_angle_offset: 0.349
coarse_angle_resolution: 0.0349
minimum_angle_penalty: 0.05
minimum_distance_penalty: 0.5
use_response_expansion: true
```

- Running SLAM 2D

(PC)

1) Running slam_toolbox package

ros2 launch create3_lidar_slam slam_toolbox_launch.py

2) visualizing a modeling create3 robot

ros2 launch irobot_create_common robot_description.py

3) controlling the create3 robot

ros2 run teleop_twist_keyboard teleop_twist_keyboard

(Jetson)

ros2 launch cyglidar_d2_ros2 cyglidar.launch.py

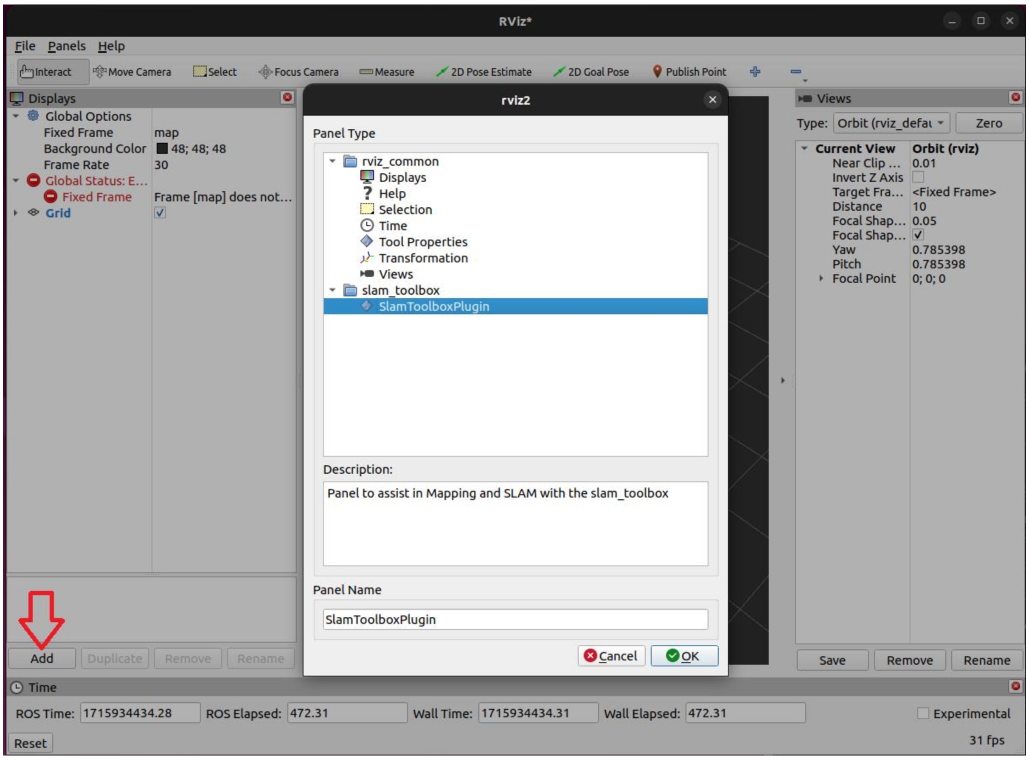
+) Before launching, please **revise the sensor's position,** like below. (need TF transforming)

```
tf_node = launch_ros.actions.Node(
        package = 'tf2_ros', executable = "static_transform_publisher", name="to_laserframe",
        arguments = ["{your sensor's X position from base_footprint}",
                     "{your sensor's Y position from base_footprint}",
                     "{your sensor's Z position from base_footprint}",
                     "0", "0", "0", "base_footprint", "laser_frame"] )
```

4. Save and Use map data

Using SlamToolboxPlugin, you can save map data.

1. Run Rviz2 and Click the 'Add' button like below. And add the 'SlamToolboxPlugin'



2. In the 'SlamToolboxPlugin', Click the 'Save Map' button.