ROS-Anbindung

# ROS-Bridge

Die ROS-Bridge ist ein ROS-Paket, dieses kann über den Paketmanager installiert werden.

sudo apt-get install ros-#ROS\_VERSION#-ros-gz-bridge

Die Bridge unterstützt nicht alle Message- und Topic-Typen. Eine Liste mit Unterstützen befindet sich unter folgendem Link:

<https://github.com/gazebosim/ros_gz/tree/ros2/ros_gz_bridge>

Der Aufruf der ROS-Bridge hat immer den gleichen Aubau

ros2 run ros\_gz\_bridge parameter\_bridge <TOPIC>@<ROS\_TYPE><DIRECTION><GZ\_TYPE>

<TOPIC>

Hier wird die Topic oder Message angegeben, die über die Bridge laufen soll.

<ROS\_TYPE>

Das ist die Message-Art, die im ROS-Netzwerk angezeigt wird.

<DIRECTION>

Die Richtung wird angegeben in der, der Nachrichtenaustausch durchgeführt werden soll. „@“ ist für eine bidirektionale Verbindung. „[“ überträgt nur Nachrichten aus dem GZ-Netzwerk in das ROS-Netzwerk. „]“ ist die entgegengesetzte Übertragungsrichtung.

<GZ\_TYPE>

Das ist die Message-Art, die im GZ-Netzwerk angezeigt wird.

# Umsetzung mit eigener Welt

Für die ROS-Anbindung wird die folgende Welt verwendet:

<https://github.com/JuSteif/GazeboLeitfaden/blob/main/gz_project/project_world_camera_lidar64.sdf>

Der Roboter in dieser Welt verfügt über vier Topics, die Daten für Sensoren und Aktoren enthalten. Dazu gehören die LiDAR-, Kamera- und Bewegungsdaten.

Der LiDAR-Sensor publisht einen LaserScan seiner Umgebung, dieser kann mit dem folgendem Terminalbefehl in eine ROS-Topic übersetzt werden.

ros2 run ros\_gz\_bridge parameter\_bridge /lidar@sensor\_msgs/msg/LaserScan[gz.msgs.LaserScan

Eine weitere Topic des LiDAR-Sensors ist die PointCloud. Diese ermöglicht das verwenden mehrerer Layer des Sensors.

ros2 run ros\_gz\_bridge parameter\_bridge /lidar/points@sensor\_msgs/msg/PointCloud2[gz.msgs.PointCloudPacked

Die Kamera veröffentlicht in Bild auf der Topic /cam

ros2 run ros\_gz\_bridge parameter\_bridge /cam@sensor\_msgs/msg/Image[gz.msgs.Image

Die letzte Topic ist /cmd\_vel, diese steuert den Roboter. Sie ist vom Typ Twist und hat im Gegensatz zu den anderen Topics die umgekehrte Richtung, da der Roboter aus dem ROS-Netzwerk gesteuert werden soll.

ros2 run ros\_gz\_bridge parameter\_bridge /model/vehicle\_blue/cmd\_vel@geometry\_msgs/msg/Twist]gz.msgs.Twist

Die Visualisierung der Sensordaten erfolgt in Rviz2

rviz2

Dort muss der „Fixed Frame“ geändert werden in „vehicle\_blue/chassis/gpu\_lidar“. Der LiDAR benötigt diesen um die Punkte korrekt darzustellen. Im Entity-Tree auf der linken Seite kann man unter „Add“ neue Topics visualisieren. Wenn man im Fenster den Reiter „by Topic“ auswählt, kann man sich den LaserScan und die PointCloud der Topic /lidar visualisieren oder das RawImage der /cam anzeige lassen. Die Topic /cmd\_vel lässt sich mit dem Teleop\_Twist\_Keyboard ansprechen.

ros2 run teleop\_twist\_keyboard teleop\_twist\_keyboard

# Zusammenfassen mehrerer Bridges

Wenn mehrere Bridges gestartet werden soll, kann man eine Launch-File schreiben. Diese wird in eine YAML-File geschrieben. Ein Beispiel für die Bridge des Kamerabildes hat folgenden Aufbau.

- gz\_topic\_name: "/cam"

ros\_topic\_name: "/cam"

ros\_type\_name: "sensor\_msgs/msg/Image"

gz\_type\_name: "gz.msgs.Image"

Ein Beispiel mit allen vier Bdriges aus dem Kapitel „Umsetzung mit eigener Welt“ befindet sich unter folgendem Link:

<https://github.com/JuSteif/GazeboLeitfaden/blob/main/gz_project/bridge_runner.yaml>