

---

# Robotik Projekt

- Aufgabenblatt 4 -

---

## Kamera einrichten

Die TurtleBot3 Roboter sind mit einer 5MP Weitwinkelkamera mit 160° Öffnungswinkel [1] ausgestattet. Um das Netzwerk beim gleichzeitigen Betrieb der Roboter nicht zu überlasten, ist die Auflösung im Treiber auf 640x480 Pixel eingestellt. Bei Bedarf könnte aber die Verwendung einer höheren Auflösung geprüft werden. Das Bild wird sowohl komprimiert wie auch unkomprimiert gepublished. Nutzen Sie im Rahmen der Lehrveranstaltung ausschließlich das komprimierte Bild auf dem Topic `"/image_raw/compressed"`. Dafür ist die Installation der image-transport-plugins notwendig:

```
sudo apt install ros-humble-image-transport-plugins
```

Überprüfen Sie die korrekte Konfiguration, in dem Sie sich das Bild mit `rqt_image_view` anzeigen lassen. Wählen Sie oben links im Dropdown-Menü entsprechend das Topic `"/image_raw/compressed"`. Anschließend sollte das live-Bild des Roboters sichtbar sein.

```
ros2 run rqt_image_view rqt_image_view
```

## Daten aufzeichnen

Mit `ros2 bag` gibt es in ROS die Möglichkeit, Topics aufzuzeichnen und später wieder abzuspielen. Insbesondere im Bereich der Bildverarbeitung kann es hilfreich sein, Veränderungen an der Software gegen aufgezeichnete Daten zu testen, da teilweise schon geringe Änderungen in der realen Welt (z.B. Helligkeit oder Schatten) zu starken Unterschieden in den Ergebnissen führen können. Darüber hinaus besteht mit aufgezeichneten Daten die Möglichkeit, unabhängig von der physischen Verfügbarkeit der Roboter zu arbeiten.

Das entsprechende Tutorial zu `ros2 bag` finden Sie hier:

<https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Recording-And-Playing-Back-Data/Recording-And-Playing-Back-Data.html>

Kurzfassung des Tutorials; Daten aufzeichnen:

```
ros2 bag record /topic1 /topic2 /topicn
```

Daten abspielen

```
ros2 bag play name_rosbag
```

Informationen zu den aufgezeichneten Daten

```
ros2 bag info name_rosbag
```

Berücksichtigen Sie beim Abspielen der Daten den Zustand des Roboters! Ist dieser angeschaltet und wurde z.B. das Topic `/cmd_val` aufgezeichnet, wird sich der Roboter beim Abspielen bewegen. Auch bei den Sensordaten kann es zu ungewollten Effekten kommen, wenn sowohl die gespeicherten wie auch die live-Daten auf dem gleichen Topic gepublished werden. Alternativ zum Ausschalten des Roboters können Sie auch die `ROS_DOMAIN_ID` ändern.

## Node per Launchfile starten

Mit `launch_ros` besteht die Möglichkeit in ROS, eine Konfiguration für den Start von Nodes zu erstellen. Damit lassen sich u.a. mehrere Nodes mit einem Befehl starten sowie die zu übergebenden Parameter festzulegen. Das entsprechende Tutorial dazu finden Sie hier: <https://docs.ros.org/en/humble/Tutorials/Intermediate/Launch/Creating-Launch-Files.html>

## Linienverfolgung per Kamera

Schreiben Sie eine Node in Python, sodass der Roboter einer Linie per Kamera folgt. Der Roboter soll sich beim Start der Node auf der Linie befinden. Die Umsetzung soll möglichst einfach gehalten werden. Soweit sinnvoll sollen Parameter konfiguriert werden können, mindestens jedoch die zur Regelung relevanten Werte.

Der Angedachte Lösungsweg sieht folgende Schritte vor, um die Komplexität möglichst gering zu halten:

- Auswertung lediglich einer Bildzeile ohne vorherige Entzerrung
- Konvertierung als Graustufenbild
- Interpretation des hellsten Pixels als Linie (bei weißer Linie auf schwarzem Untergrund)
- Unterscheidung von lediglich drei Fällen, wo sich die Linie gerade befindet (links, mittig, rechts) und entsprechende Reaktion darauf (drehen oder fahren). Die Umsetzung einer eleganteren Regelungsstrategie ist nicht gefordert
- Möglichkeit der Anpassung der Größe dieser drei Bereiche per Parameter, um ggf. kleine Verbesserungen beim Fahrverhalten erreichen zu können
- keine Behandlung von Störungen wie z.B. durch spiegelnden Untergrund

Für die Abgabe gibt es in der Lessingstraße eine weiße Linie auf schwarzem (teilweise reflektierenden) Grund, ggf. andere gewünschte Kombinationen müssten selbst organisiert werden. Der Start soll mit `ros launch` erfolgen. Der Code soll (spätestens) zur Abgabe dokumentiert im Git liegen.

Eine Inspiration zur Nutzung der Kamera finden Sie zu Beginn der fünften Übung hier: [https://github.com/gerolicht/robotik\\_projekt/](https://github.com/gerolicht/robotik_projekt/)

[1] <https://www.reichelt.de/de/de/raspberry-pi-kamera-5mp-160--rpi-wwcam-p148944.html?r=1>