

Thema: ROS2 humable

Von: Stephan Reckziegel
Studiengang: Technische Informatik; 5.Semester

Abgabedatum: 16. Januar 2026

Inhaltsverzeichnis

1. Vorbereitung	3
1.1. Installation von ROS2	3
1.2. Arbeitsverzeichnis erstellen	3
1.3. Universal Robots ROS 2 Pakete klonen	3
1.4. Abhängigkeiten installieren	3
1.5. Workspace Bauen	3
1.6. 7. Test: UR5e Description laden (z. B. in RViz)	3
1.7. 8. (Optional) Mit echtem Roboter verbinden	3
1.8. 9. Ergebnis	3
1.9. Bonus: MoveIt 2 integrieren (optional)	4
1.10. Verbindungen	4

1. Vorbereitung

1.1. Installation von ROS2

ROS2 Installieren

```
sudo apt install ros-humble-desktop ROS2-sorcen
```

```
source /opt/ros/humble/setup.bash
```

1.2. Arbeitsverzeichnis erstellen

```
mkdir -p /ros2_ws/src
```

```
cd /ros2_ws/src
```

1.3. Universal Robots ROS 2 Pakete klonen

Offizielle Treiber holen

```
https://github.com/UniversalRobots/Universal_Robots_ROS2_Driver git clone -b humble
```

```
https://github.com/UniversalRobots/Universal_Robots_ROS2_Driver.git ur_ros2_driver
```

Beschreibung (URDF / Meshes / MoveIt):

```
git clone -b humble https://github.com/UniversalRobots/Universal_Robots_ROS2_  
Description.git ur_description
```

Wird Iron oder Jazzy verwendet muss man den Branch (-b iron oder -b jazzy) angeben

1.4. Abhängigkeiten installieren

Dazu muss man zum Workspace oder Hauptverzeichnis zurückkehren

```
cd /ros2_ws
```

```
rosdep update
```

```
rosdep install --from-paths src --ignore-src -r -y
```

1.5. Workspace Bauen

```
colcon build --symlink-install
```

Wenn der Build fertig ist, musst du dein Setup sourcen: **source install/setup.bash**

Typisch ist es, das in `/.bashrc` einzutragen:

```
echo "source /ros2_ws/install/setup.bash" >> /.bashrc
```

1.6. 7. Test: UR5e Description laden (z. B. in RViz)

```
ros2 launch ur_description view_ur.launch.py ur_type:=ur5e
```

1.7. 8. (Optional) Mit echtem Roboter verbinden

```
ros2 launch ur_ros2_driver ur_control.launch.py ur_type:=ur5e robot_ip:=192.168.1.30
```

1.8. 9. Ergebnis

```
/ros2_ws/
```

```
|— src/  
| |— ur_ros2_driver/  
| |— ur_description/  
|— build/  
|— install/  
|— log/
```

Und du kannst:

- Simulation starten (RViz / Gazebo)
- Den echten UR5e ansteuern
- MoveIt2 oder ROS2 Control einbinden

1.9. Bonus: MoveIt 2 integrieren (optional)

Für Planung und Bewegung:

sudo apt install ros-humble-moveit

und dann:

ros2 launch ur_moveit_config ur5e_moveit.launch.py

1.10. Verbindungen

1.10.1. Normal Verbinden

- Umgebungsvariablen von Ros2 in das Terminal laden
source /opt/ros/humble/setup.bash
- Verbindungsaufbau mit dem Roboter
**ros2 launch ur_robot_driver ur_control.launch.py ur_type:=ur5e robot_ip:=192.168.1.7
initial_trajectory_controller:=scaled_joint_trajectory_controller launch_rviz:=false
headless_mode:=false**

ros2 launch: Ros2-Pakete und Knoten von ROS starten **ur_robot_driver:** Name des Paketes, das gestartet werden soll **ur_control.launch.py:** Launchdatei **ur_type:=ur5e:** Argument für die Launchdatei als den Typ des Roboters **robot_ip:=** IP: Ip-Adresse des Roboters
initial_trajectory_controller: Typ des Trajektor-Datei **launch_rviz:=false:** Ob das Programm "rviz" für die Virtualisierung starten oder nicht **headless_mode:=false:** Mit oder ohne Grafische Oberfläche starten

- Die Datei external.urp ausführen

1.10.2. Node bauen, starten und mit Roboter verbinden

- Umgebungsvariablen in das Terminal laden
source /opt/ros/humble/setup.bash
- Workspace neu bauen
colcon build --packages-select ur_move
- Terminal neu vorbereiten
source install/setup.bash
- Die entsprechende Node ausführen
python3 src/ur_move/ur_move/test_trajectory.py

1.10.3. Troubleshooting

- Per SSH-Verbinden
ssh root@192.168.1.7 => easybot
 - Ping
ping <IP-Adresse>
 - alte Builds löschen
rm -rf ws-build
- >bild/, install/, log/ werden gelöscht
- Controller reaktivieren
Fallst der Controller nicht richtig aktiviert wurde oder neu gesetzt wurde
ros2 control set_controller_state scaled_joint_trajectory_controller active
 - Controller überprüfen
ros2 control list_controllers
 - Controller sind nicht gleichzeitig aktiviert
ros2 control set_controller_state joint_trajectory_controller inactive