

## Nutzung von micro-RoS zwischen einem Raspberry Pi 5 und einem ESP32

### 1. Einführung in micro-RoS

#### Was ist micro-RoS?

micro-RoS ist eine für Mikrocontroller optimierte Version von RoS2 (Robot Operating System 2). Sie erweitert das RoS2-Ökosystem, indem sie es auf ressourcenbeschränkte eingebettete Systeme bringt. micro-RoS basiert auf **Micro XRCE-DDS**, einem kompakten DDS-Client, der für den Einsatz auf Geräten mit begrenztem Speicher und Rechenleistung entwickelt wurde.

micro-RoS bietet eine standardisierte Möglichkeit zur Kommunikation zwischen Mikrocontrollern und einem RoS2-System. Es ermöglicht Mikrocontrollern, RoS2-Nachrichten zu senden und zu empfangen, wodurch sie vollständig in ein RoS2-Ökosystem integriert werden können.

#### Hauptmerkmale von micro-RoS

- **Optimiert für Mikrocontroller:** Kann auf Plattformen wie ESP32, STM32 oder Arduino ausgeführt werden.
- **Verwendet DDS für Kommunikation:** Nutzt das DDS-Protokoll, das auch in RoS2 verwendet wird, um eine nahtlose Integration zu gewährleisten.
- **Ermöglicht RoS2-Funktionalität auf Embedded-Systemen:** Unterstützung für Topics, Services, Parameter und Actions auf Mikrocontrollern.
- **Speichereffizient und Echtzeitfähig:** Läuft mit wenigen Kilobyte RAM und unterstützt Echtzeitkommunikation.

#### Vergleich: RoS2 vs. micro-RoS

Merkmal	RoS2	micro-RoS
Plattform	PCs, SBCs (z. B. Raspberry Pi)	Mikrocontroller (z. B. ESP32, STM32)
Kommunikation	DDS (Data Distribution Service)	Micro XRCE-DDS (kompakte DDS-Version)
Ressourcenverbrauch	Hoch (mehrere MB RAM)	Sehr niedrig (wenige KB RAM)
Unterstützte Features	Vollständige RoS2-Funktionalität	Begrenzte RoS2-Funktionalität für Embedded-Systeme

#### Architektur von micro-RoS

Ein typisches micro-RoS-Setup besteht aus drei Hauptkomponenten:

1. **micro-RoS-Agent (läuft auf Raspberry Pi 5):** Vermittelt die Kommunikation zwischen dem Mikrocontroller (ESP32) und dem RoS2-Ökosystem.
2. **micro-RoS-Client (läuft auf ESP32):** Führt RoS2-Knoten auf einem Mikrocontroller aus und kommuniziert mit dem Agenten über UART, WiFi oder Ethernet.
3. **RoS2-Host (z. B. Raspberry Pi 5 oder PC):** Führt reguläre RoS2-Knoten aus, die mit micro-RoS-Knoten interagieren.

### 2. micro-RoS Befehle

Hier eine Übersicht der wichtigsten micro-RoS-Befehle:

Befehl	Beispiel	Anwendungsgebiet	Erläuterung
<code>micro_ros_agent serial --dev /dev/ttyUSB0 -b 115200</code>	<code>micro_ros_agent serial --dev /dev/ttyUSB0 -b 115200</code>	Startet den micro-RoS-Agenten über eine serielle Verbindung.	Verwendet, um micro-RoS über eine serielle Verbindung mit dem Host zu verbinden.
<code>ros2 topic pub</code>	<code>ros2 topic pub /led_toggle std_msgs/msg/Bool '{data: true}'</code>	Publish einer Nachricht	Sendet eine Nachricht von ROS2 an einen micro-RoS-Knoten.
<code>ros2 topic echo</code>	<code>ros2 topic echo /sensor_data</code>	Abonnement von micro-RoS Nachrichten	Zeigt die vom micro-RoS-Client gesendeten Nachrichten an.
<code>ros2 service call</code>	<code>ros2 service call /reset_motor std_srvs/srv/Empty</code>	Serviceaufruf	Ruft einen Service auf einem micro-RoS-Node auf.
<code>ros2 action send_goal</code>	<code>ros2 action send_goal /move_robot my_msgs/action/Move '{x: 5, y: 10}'</code>	Actionsteuerung	Sendet eine asynchrone Aktion an micro-RoS.
<code>ros2 param list</code>	<code>ros2 param list /esp32_node</code>	Parameterverwaltung	Listet alle verfügbaren Parameter eines micro-RoS-Nodes auf.

### 3. Installation von micro-RoS auf Raspberry Pi 5 und ESP32

#### 3.1 Installation auf dem Raspberry Pi 5

##### 1. ROS2 auf dem Raspberry Pi 5 installieren

Falls ROS2 noch nicht installiert ist, installiere **ROS2 Jazzy**:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y ros-jazzy-ros-base
```

Source das ROS2-Setup:

```
echo "source /opt/ros/jazzy/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

## 2. micro-ROS-Agent installieren


```
sudo apt install python3-pip
pip install micro_ros_agent
```

Starte den micro-ROS-Agenten:

```
micro_ros_agent serial --dev /dev/ttyUSB0 -b 115200
```

## Fazit

Mit diesem Setup kann ein **Raspberry Pi 5** ROS2-Commands an einen **ESP32** senden, der **Motoren und LEDs** steuert. Der Gamepad-Controller ermöglicht dabei die einfache Steuerung.

 **ROS2 trifft Embedded-Systeme – Willkommen in der Zukunft der Robotik!**