

Guia Completo: Navegação com Nav2 no Raspberry Pi 4 com ROS 2

Este guia descreve como configurar o pacote Navigation2 (Nav2) em um Raspberry Pi 4 com ROS 2 Humble, incluindo o controle de relés com gpod, geração de mapas com SLAM e envio de metas de navegação.

1. Instalar o ROS 2 Humble no Raspberry Pi 4

```
sudo apt update && sudo apt upgrade -y
sudo apt install curl gnupg2 lsb-release

sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o
/usr/share/keyrings/ros-archive-keyring.gpg

echo "deb [signed-by=/usr/share/keyrings/ros-archive-keyring.gpg]
http://packages.ros.org/ros2/ubuntu $(lsb_release -cs) main" | sudo tee
/etc/apt/sources.list.d/ros2.list

sudo apt update
sudo apt install ros-humble-desktop

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

2. Instalar Nav2 e SLAM Toolbox

```
sudo apt install ros-humble-navigation2 ros-humble-nav2-bringup
sudo apt install ros-humble-slam-toolbox
```

3. Criar mapa com SLAM Toolbox

```
ros2 launch slam_toolbox online_async_launch.py

# Após mapear o ambiente, salve o mapa:
ros2 run nav2_map_server map_saver_cli -f ~/mapa
```

4. Estrutura de Arquivos Necessária

```
~/ros2_ws/src/meu_robot/
■■■ launch/nav2.launch.py
■■■ nav2_params.yaml
■■■ mapa.yaml + mapa.pgm
```

5. Exemplo nav2.launch.py

```
from launch import LaunchDescription
from launch_ros.actions import Node

def generate_launch_description():
    return LaunchDescription([
        Node(
            package='nav2_bringup',
            executable='bringup_launch.py',
            name='nav2_bringup',
            output='screen',
            parameters=[
                {'use_sim_time': False},
                '/home/pi/ros2_ws/src/meu_robot/nav2_params.yaml'
            ],
            arguments=['map:=/home/pi/ros2_ws/src/meu_robot/mapa.yaml']
        )
    ])
```

6. Exemplo nav2_params.yaml (parcial)

```
amcl:
  ros__parameters:
    use_sim_time: false

bt_navigator:
  ros__parameters:
    use_sim_time: false
    default_bt_xml_filename: "navigate_w_replanning_and_recovery.xml"

controller_server:
  ros__parameters:
    controller_frequency: 10.0
```

7. Compilar e Rodar

```
cd ~/ros2_ws
colcon build
source install/setup.bash

ros2 launch meu_robot nav2.launch.py
```

8. Enviar ponto de destino para o robô

```
ros2 topic pub /goal_pose geometry_msgs/PoseStamped '
{
  header: {frame_id: "map"},
```

```
pose: {  
  position: {x: 1.0, y: 2.0, z: 0.0},  
  orientation: {x: 0.0, y: 0.0, z: 0.0, w: 1.0}  
}
```

9. Integração com controle de relé

Crie um nó ROS 2 que escuta o tópico `/navigate_to_pose/result`. Ao detectar sucesso na navegação, acione os relés com `gpod`.

```
ros2 topic echo /navigate_to_pose/result
```