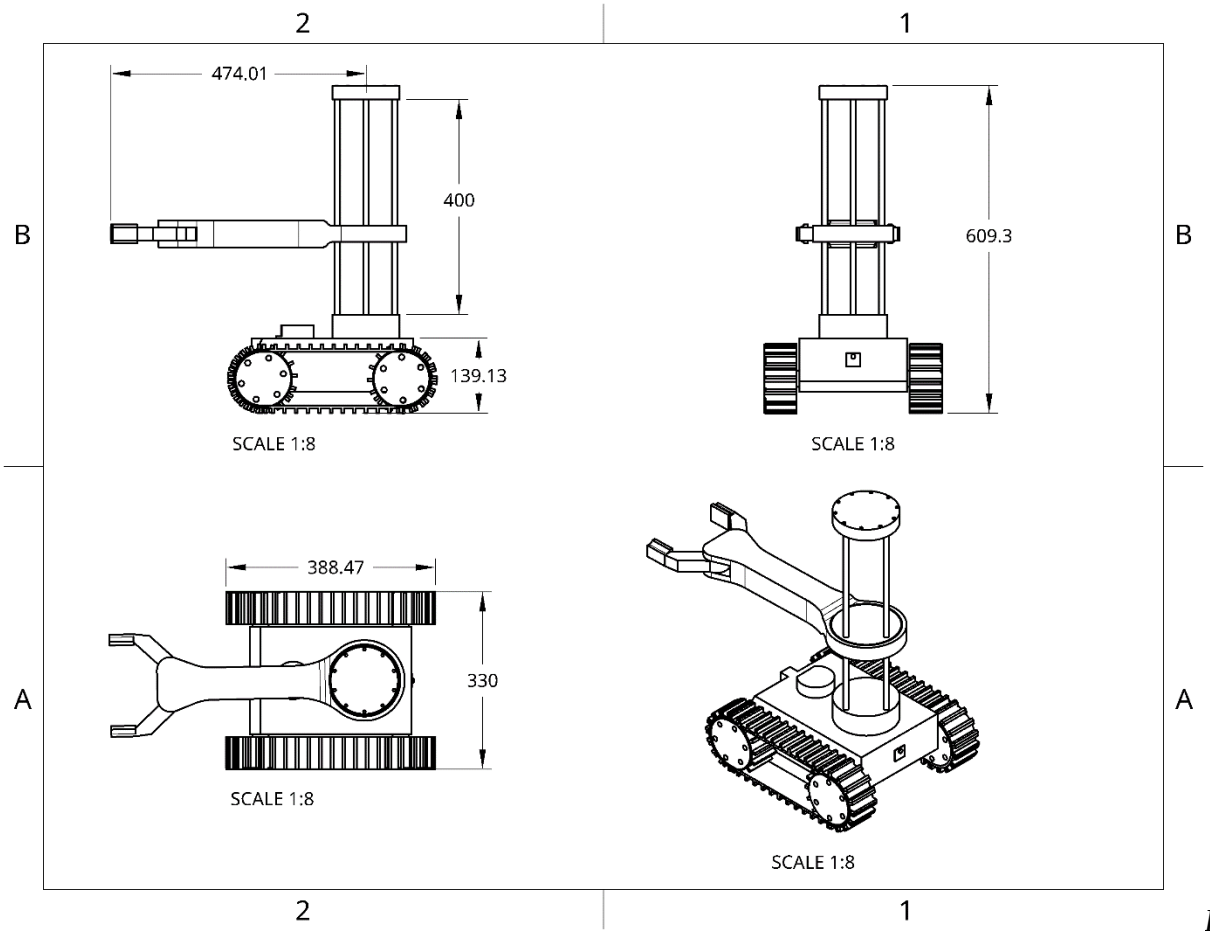
Báo cáo ROS giữa kỳ

Họ và tên: Nguyễn Văn Diễn

MSV: 22027541

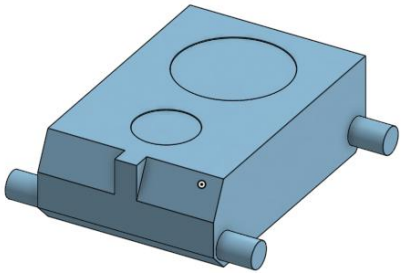
1. Dạng robot, động học, kích thước
    a. Dạng robot
        - Loại di chuyển: Bánh xích
        - Tay máy (khớp 1): Tịnh tiến
        - Tay máy (khớp 2): Xoay
        - Cảm biến: LIDAR, Camera, Encoder

    b. Động học
        - Cách di chuyển: Di chuyển tiến lùi, quay trái, quay phải
        - Bậc tự do:
        - Kích thước:
            - Thân xe:
                - Chiều dài: 388.47 mm
                - Chiều rộng: 330 mm
                - Chiều cao: 139.13 mm
            - Tay máy:
                - Khớp tịnh tiến: 400 mm
                - Khớp xoay: 474.01 mm

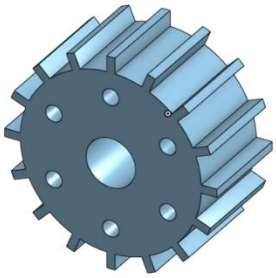


Hình 1. Bản vẽ robot

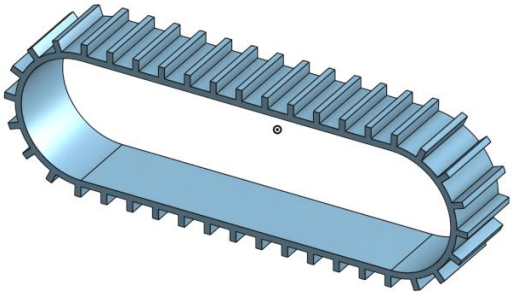
2. Thiết kế, cách đặt hệ trục toạ độ
    a. Thiết kế Onshape



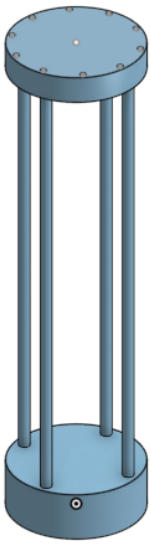
Hình 2. Thân xe



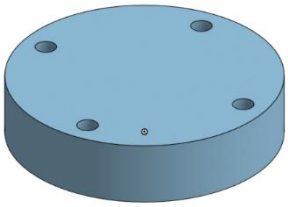
Hình 3. Bánh xe



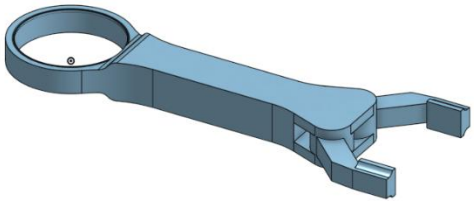
Hình 4. Xích



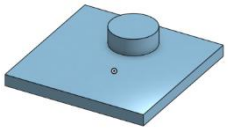
Hình 5. Khâu 1



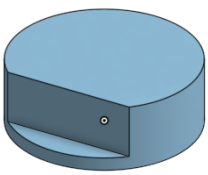
Hình 6. Khớp



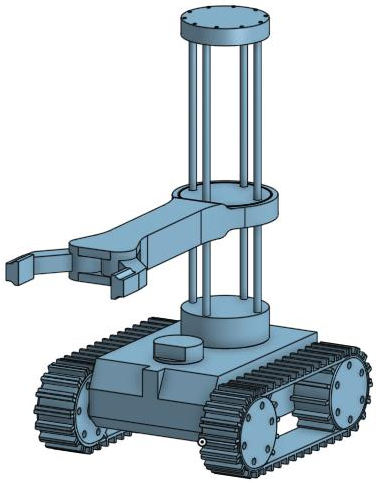
Hình 7. Khâu 2



Hình 8. Camera



Hình 9. LIDAR



Hình 10. Robot hoàn chỉnh

b. Đặt hệ trục toạ độ
   - Hệ trục toạ độ có gốc tại trung tâm hình học của robot. Trục x hướng về phía trước của robot, trục y hướng sang trái, trục z hướng lên trên.
   - Mỗi bánh xe có hệ toạ độ riêng với gốc tại tâm bánh, trục z trùng với trục quay của bánh.

3. Mô tả file Xacro, liên kết của các link, các cảm biến, mô tả Gazebo
   a. Mô tả file Xacro
      - Gồm 8 file xacro trong thư mục **description**:
        - **macros.xacro**: Gồm các macro

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro" name="macros">
    <!-- Macro định nghĩa một link với quán tính, hình học và màu sắc -->
    <xacro:macro name="link" params="link_name
                                     x1 y1 z1 roll1 pitch1 yaw1
                                     mass
                                     ixx ixy ixz iyy iyz izz
                                     r g b a
                                     x2 y2 z2 roll2 pitch2 yaw2">
        <link name="${link_name}">
            <inertial>
                <!-- Vị trí và hướng của khối lượng quán tính -->
                <origin xyz="${x1} ${y1} ${z1}" rpy="${roll1} ${pitch1} ${yaw1}"/>
                <mass value="${mass}"/> <!-- Khối lượng của link -->
                <!-- Ma trận quán tính -->
                <inertia ixx="${ixx}" ixy="${ixy}" ixz="${ixz}" iyy="${iyy}" iyz="${iyz}"
izz="${izz}"/>
            </inertial>
            <visual>
                <!-- Vị trí và hướng của hình học hiển thị -->
                <origin xyz="${x2} ${y2} ${z2}" rpy="${roll2} ${pitch2} ${yaw2}"/>
```

```
                <geometry>
                    <!-- File STL biểu diễn hình học hiển thị -->
                    <mesh filename="package://xe_tang/meshes/${link_name}.stl"/>
                </geometry>
                <material name="${link_name}_material">
                    <!-- Màu sắc RGBA của link -->
                    <color rgba="${r} ${g} ${b} ${a}"/>
                </material>
            </visual>
            <collision>
                <!-- Vị trí và hướng của hình học va chạm -->
                <origin xyz="${x2} ${y2} ${z2}" rpy="${roll2} ${pitch2} ${yaw2}"/>
                <geometry>
                    <!-- File STL biểu diễn hình học va chạm -->
                    <mesh filename="package://xe_tang/meshes/${link_name}.stl"/>
                </geometry>
            </collision>
        </link>
    </xacro:macro>

    <!-- Macro định nghĩa một khớp với loại khớp, vị trí và giới hạn (nếu có) -->
    <xacro:macro name="joint" params="joint_name type
                                      x1 y1 z1 roll1 pitch1 yaw1
                                      parent_link
                                      child_link
                                      x2:=0 y2:=0 z2:=0
                                      effort:=0 velocity:=0 lower:=0 upper:=0">
        <joint name="${joint_name}" type="${type}">
            <!-- Vị trí và hướng của khớp -->
            <origin xyz="${x1} ${y1} ${z1}" rpy="${roll1} ${pitch1} ${yaw1}"/>
            <parent link="${parent_link}"/> <!-- Link cha -->
            <child link="${child_link}"/> <!-- Link con -->
            <!-- Nếu là khớp quay hoặc tịnh tiến, thêm trục và giới hạn -->
            <xacro:if value="${type == 'revolute' or type == 'prismatic'}">
                <axis xyz="${x2} ${y2} ${z2}"/>
                <limit effort="${effort}" velocity="${velocity}" lower="${lower}"
upper="${upper}"/>
            </xacro:if>
            <!-- Nếu là khớp liên tục, chỉ thêm trục và giới hạn lực/tốc độ -->
            <xacro:if value="${type == 'continuous'}">
                <axis xyz="${x2} ${y2} ${z2}"/>
                <limit effort="${effort}" velocity="${velocity}"/>
            </xacro:if>
        </joint>
    </xacro:macro>

    <!-- Macro định nghĩa truyền động cho base (dùng giao diện vận tốc) -->
    <xacro:macro name="base_transmission" params="joint_name actuator_name">
        <transmission name="${joint_name}_transmission">
            <type>transmission_interface/SimpleTransmission</type> <!-- Loại truyền động
đơn giản -->
            <actuator name="${actuator_name}">
                <hardwareInterface>hardware_interface/VelocityJointInterface</hardwareInte
rface> <!-- Giao diện vận tốc -->
                <mechanicalReduction>1</mechanicalReduction> <!-- Tỷ lệ giảm tốc: 1 -->
            </actuator>
            <joint name="${joint_name}">
                <hardwareInterface>hardware_interface/VelocityJointInterface</hardwareInte
rface>
            </joint>
        </transmission>
    </xacro:macro>
```

```
    <!-- Macro định nghĩa truyền động cho cánh tay (dùng giao diện lực) -->
    <xacro:macro name="arm_transmission" params="joint_name actuator_name">
        <transmission name="${joint_name}_transmission">
            <type>transmission_interface/SimpleTransmission</type> <!-- Loại truyền động
đơn giản -->
            <actuator name="${actuator_name}">
                <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterf
ace> <!-- Giao diện lực -->
                <mechanicalReduction>1</mechanicalReduction> <!-- Tỷ lệ giảm tốc: 1 -->
            </actuator>
            <joint name="${joint_name}">
                <hardwareInterface>hardware_interface/EffortJointInterface</hardwareInterf
ace>
            </joint>
        </transmission>
    </xacro:macro>

    <!-- Macro cấu hình thuộc tính Gazebo: vật liệu và ma sát -->
    <xacro:macro name="gazebo" params="reference material mu1 mu2">
        <gazebo reference="${reference}">
            <material>${material}</material> <!-- Vật liệu hiển thị trong Gazebo -->
            <mu1>${mu1}</mu1> <!-- Hệ số ma sát 1 -->
            <mu2>${mu2}</mu2> <!-- Hệ số ma sát 2 -->
        </gazebo>
    </xacro:macro>
</robot>
```

- **arm_controller.xacro**: Điều khiển tay máy

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro" name="arm_controller">
    <!-- Plugin điều khiển Gazebo ROS Control -->
    <plugin name="arm_control" filename="libgazebo_ros_control.so">
        <!-- Không gian tên gốc cho tay máy -->
        <robotNamespace>/</robotNamespace>
    </plugin>

    <!-- Định nghĩa truyền động cho khớp tịnh tiến với động cơ lower_motor -->
    <xacro:arm_transmission joint_name="khop_joint" actuator_name="lower_motor"/>
    <!-- Định nghĩa truyền động cho khớp xoay với động cơ upper_motor -->
    <xacro:arm_transmission joint_name="khau_2_joint" actuator_name="upper_motor"/>
</robot>
```

- **base_controller.xacro**: Điều khiển robot

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro" name="base_controller">
    <gazebo>
        <!-- Plugin điều khiển Gazebo ROS Control -->
        <plugin name="base_control" filename="libgazebo_ros_control.so">
            <!-- Không gian tên gốc cho robot -->
            <robotNamespace>/</robotNamespace>
        </plugin>
    </gazebo>

    <!-- Truyền động cho bánh xe 4, sử dụng động cơ trái 1 -->
    <xacro:base_transmission joint_name="banh_xe_4_joint"
actuator_name="left_wheel_motor_1"/>
    <!-- Truyền động cho bánh xe 3, sử dụng động cơ trái 2 -->
```

```
    <xacro:base_transmission joint_name="banh_xe_3_joint"
actuator_name="left_wheel_motor_2"/>
    <!-- Truyền động cho bánh xe 2, sử dụng động cơ phải 1 -->
    <xacro:base_transmission joint_name="banh_xe_2_joint"
actuator_name="right_wheel_motor_1"/>
    <!-- Truyền động cho bánh xe 1, sử dụng động cơ phải 2 -->
    <xacro:base_transmission joint_name="banh_xe_1_joint"
actuator_name="right_wheel_motor_2"/>
</robot>
```

- **camera.xacro**: Cấu hình camera

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro" name="camera">
    <!-- Định nghĩa link camera với thông số vị trí, quán tính và màu sắc -->
    <xacro:link link_name="camera"
                x1="5.48234e-19" y1="-0.000645903" z1="-0.00126913" roll1="0" pitch1="0"
yaw1="0"
                mass="0.00147364"
                ixx="7.56624e-08" ixy="0" ixz="0" iyy="7.05662e-08" iyz="-2.12341e-09"
izz="1.43344e-07"
                r="0" g="1.0" b="0" a="1.0"
                x2="1.9877e-18" y2="0" z2="0" roll2="3.14159" pitch2="0" yaw2="0"/>

    <!-- Link quang học của camera, không có thuộc tính vật lý cụ thể -->
    <link name="camera_optical"/>

    <!-- Thuộc tính Gazebo cho link camera: vật liệu và ma sát -->
    <xacro:gazebo reference="camera" material="Gazebo/Green" mu1="0.2" mu2="0.2"/>

    <!-- Cấu hình cảm biến camera trong Gazebo cho link camera_optical -->
    <gazebo reference="camera_optical">
        <sensor name="camera" type="camera">
            <!-- Tần suất cập nhật dữ liệu camera: 30 Hz -->
            <update_rate>30.0</update_rate>
            <camera>
                <!-- Góc nhìn ngang của camera: ~80 độ -->
                <horizontal_fov>1.3962634</horizontal_fov>
                <image>
                    <!-- Độ phân giải ảnh: 1000x1000 pixel, định dạng RGB -->
                    <width>1000</width>
                    <height>1000</height>
                    <format>R8G8B8</format>
                </image>
                <clip>
                    <!-- Khoảng cách cắt gần và xa của camera -->
                    <near>0.02</near>
                    <far>300</far>
                </clip>
                <noise>
                    <!-- Nhiễu Gaussian với độ lệch chuẩn 0.007 -->
                    <type>gaussian</type>
                    <mean>0.0</mean>
                    <stddev>0.007</stddev>
                </noise>
            </camera>
            <!-- Plugin điều khiển camera trong Gazebo -->
            <plugin name="camera_controller" filename="libgazebo_ros_camera.so">
                <alwaysOn>true</alwaysOn> <!-- Camera luôn bật -->
                <updateRate>0.0</updateRate> <!-- Tần suất cập nhật do sensor quyết định -
->
```

```
                <cameraName>camera_pi</cameraName> <!-- Tên camera trong ROS -->
                <imageTopicName>image_raw</imageTopicName> <!-- Topic xuất bản ảnh thô -->
                <cameraInfoTopicName>camera_info</cameraInfoTopicName> <!-- Topic xuất bản
thông tin camera -->
                <frameName>camera_optical</frameName> <!-- Khung tham chiếu của camera -->
                <hackBaseline>0.07</hackBaseline> <!-- Khoảng cách baseline (giả lập
stereo nếu cần) -->
                <!-- Các tham số méo ống kính, hiện tại không áp dụng -->
                <distortionK1>0.0</distortionK1>
                <distortionK2>0.0</distortionK2>
                <distortionK3>0.0</distortionK3>
                <distortionT1>0.0</distortionT1>
                <distortionT2>0.0</distortionT2>
            </plugin>
        </sensor>
    </gazebo>

    <!-- Khớp cố định gắn camera vào base_link -->
    <xacro:joint joint_name="camera_joint" type="fixed"
                 x1="0.151437" y1="3.96457e-17" z1="0.0991321" roll1="-1.5708" pitch1="0"
yaw1="1.5708"
                 parent_link="base_link"
                 child_link="camera"/>

    <!-- Khớp cố định giữa camera và khung quang học -->
    <xacro:joint joint_name="camera_optical_joint" type="fixed"
                 x1="0" y1="0" z1="0" roll1="1.5708" pitch1="1.5708" yaw1="0"
                 parent_link="camera"
                 child_link="camera_optical"/>
</robot>
```

- **encoder.xacro**: Cấu hình encoder

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro" name="encoder">
    <gazebo>
        <!-- Plugin Gazebo mô phỏng dữ liệu của encoder -->
        <plugin name="differential_drive_controller"
filename="libgazebo_ros_diff_drive.so">
            <!-- Tần suất cập nhật của plugin: 100 Hz -->
            <updateRate>100</updateRate>
            <!-- Khớp bánh xe trái -->
            <leftJoint>left_wheel_joint</leftJoint>
            <!-- Khớp bánh xe phải -->
            <rightJoint>right_wheel_joint</rightJoint>
            <!-- Khoảng cách giữa hai bánh xe: 0.135 m -->
            <wheelSeparation>0.135</wheelSeparation>
            <!-- Đường kính bánh xe: 0.129 m -->
            <wheelDiameter>0.129</wheelDiameter>
            <!-- Mô-men xoắn tối đa của bánh xe: 20 Nm -->
            <wheelTorque>20</wheelTorque>
            <!-- Topic nhận lệnh vận tốc (twist) từ ROS -->
            <commandTopic>cmd_vel</commandTopic>
            <!-- Topic xuất bản dữ liệu odometry -->
            <odometryTopic>odom</odometryTopic>
            <!-- Khung tham chiếu cho odometry -->
            <odometryFrame>odom</odometryFrame>
            <!-- Khung tham chiếu của robot (thường là thân robot) -->
            <robotBaseFrame>base_link</robotBaseFrame>
            <!-- Xuất bản trạng thái khớp bánh xe lên ROS -->
            <publishWheelJointState>true</publishWheelJointState>
```

```
        </plugin>
    </gazebo>
</robot>
```

- **lidar.xacro**: Cấu hình LIDAR

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro" name="lidar">
    <!-- Định nghĩa link cho LIDAR với thông số vị trí, quán tính và màu sắc -->
    <xacro:link link_name="lidar"
                x1="0.00218942" y1="-6.24511e-05" z1="-0.0123115" roll1="0" pitch1="0"
yaw1="0"
                mass="0.0941187"
                ixx="3.50951e-05" ixy="-1.59755e-07" ixz="5.54001e-07" iyy="2.9499e-05"
iyz="-1.58024e-08" izz="5.46193e-05"
                r="0" g="0" b="0" a="1.0"
                x2="-1.249e-16" y2="2.23093e-17" z2="0" roll2="3.14159" pitch2="0" yaw2="-
0.0285163"/>

    <!-- Cấu hình Gazebo cho link LIDAR: vật liệu, ma sát và cảm biến -->
    <gazebo reference="lidar">
        <material>Gazebo/Black</material> <!-- Vật liệu màu đen trong Gazebo -->
        <mu1>0.2</mu1> <!-- Hệ số ma sát 1 -->
        <mu2>0.2</mu2> <!-- Hệ số ma sát 2 -->

        <!-- Cảm biến tia (ray) cho LIDAR -->
        <sensor name="laser" type="ray">
            <visualize>true</visualize> <!-- Hiển thị tia quét trong Gazebo -->
            <update_rate>30</update_rate> <!-- Tần suất cập nhật: 30 Hz -->
            <ray>
                <scan>
                    <horizontal>
                        <samples>360</samples> <!-- Số mẫu quét: 360 (quét toàn vòng) -->
                        <resolution>1</resolution> <!-- Độ phân giải góc: 1 độ/mẫu -->
                        <min_angle>0</min_angle> <!-- Góc quét tối thiểu: 0 radian -->
                        <max_angle>6.28319</max_angle> <!-- Góc quét tối đa: 2π radian
(360 độ) -->
                    </horizontal>
                </scan>
                <range>
                    <min>0.12</min> <!-- Khoảng cách tối thiểu: 0.12 m -->
                    <max>10</max> <!-- Khoảng cách tối đa: 10 m -->
                    <resolution>0.015</resolution> <!-- Độ phân giải khoảng cách: 0.015 m
-->
                </range>
            </ray>
            <!-- Plugin tích hợp LIDAR với ROS -->
            <plugin name="laser_controller" filename="libgazebo_ros_laser.so">
                <topicName>scan</topicName> <!-- Topic xuất bản dữ liệu quét -->
                <frameName>lidar</frameName> <!-- Khung tham chiếu của LIDAR -->
                <gaussianNoise>0.01</gaussianNoise> <!-- Nhiễu Gaussian với độ lệch chuẩn
0.01 -->
            </plugin>
        </sensor>
    </gazebo>

    <!-- Khớp cố định gắn LIDAR vào base_link -->
    <xacro:joint joint_name="lidar_joint" type="fixed"
                 x1="-0.0675634" y1="4.37765e-17" z1="0.138132" roll1="3.14159" pitch1="0"
yaw1="0"
                 parent_link="base_link"
```

```
                child_link="lidar"/>
</robot>
```

- **robot.xacro**: Mô tả hình dạng vật lý của robot và tay máy

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro" name="robot">
    <!-- Định nghĩa link thân chính (base_link) của robot -->
    <xacro:link link_name="base_link"
                x1="0.00402455" y1="4.46693e-17" z1="0.0886727" roll1="0" pitch1="0"
yaw1="0"
                mass="7.43294"
                ixx="0.0323519" ixy="0" ixz="-0.000431382" iyy="0.0606346" iyz="0"
izz="0.0810268"
                r="0.175" g="0.175" b="0.175" a="1.0"
                x2="0.00243663" y2="4.16334e-17" z2="0.0391321" roll2="0" pitch2="0"
yaw2="0"/>

    <!-- Định nghĩa bánh xe 1 -->
    <xacro:link link_name="banh_xe_1"
                x1="-5.25222e-18" y1="2.39474e-17" z1="0.00456292" roll1="0" pitch1="0"
yaw1="0"
                mass="0.700837"
                ixx="0.000816204" ixy="1.75689e-20" ixz="1.65089e-45" iyy="0.000816204"
iyz="4.68061e-46" izz="0.00121262"
                r="0.1" g="0.1" b="0.1" a="1.0"
                x2="-6.93889e-18" y2="2.77556e-17" z2="-0.025" roll2="-1.5708" pitch2="-
3.64847e-42" yaw2="2.45344e-17"/>

    <!-- Khớp liên tục cho bánh xe 1 gắn với base_link -->
    <xacro:joint joint_name="banh_xe_1_joint" type="continuous"
                 x1="-0.127563" y1="0.14" z1="0.0641321" roll1="-1.5708" pitch1="1.42583"
yaw1="3.14159"
                 parent_link="base_link"
                 child_link="banh_xe_1"
                 x2="0" y2="0" z2="1" effort="10" velocity="10"/>

    <!-- Định nghĩa bánh xe 2 -->
    <xacro:link link_name="banh_xe_2"
                x1="1.68667e-18" y1="-1.7686e-17" z1="0.00456292" roll1="0" pitch1="0"
yaw1="0"
                mass="0.700837"
                ixx="0.000816204" ixy="4.60524e-20" ixz="1.48737e-45" iyy="0.000816204"
iyz="-2.27194e-45" izz="0.00121262"
                r="0.1" g="0.1" b="0.1" a="1.0"
                x2="0" y2="-1.38778e-17" z2="-0.025" roll2="-1.5708" pitch2="-1.90898e-42"
yaw2="4.22367e-17"/>

    <!-- Khớp liên tục cho bánh xe 2 gắn với base_link -->
    <xacro:joint joint_name="banh_xe_2_joint" type="continuous"
                 x1="0.132437" y1="0.14" z1="0.0641321" roll1="-1.5708" pitch1="1.05145"
yaw1="3.14159"
                 parent_link="base_link"
                 child_link="banh_xe_2"
                 x2="0" y2="0" z2="1" effort="10" velocity="10"/>

    <!-- Định nghĩa bánh xe 3 -->
    <xacro:link link_name="banh_xe_3"
                x1="1.68667e-18" y1="-3.38768e-19" z1="0.00456292" roll1="0" pitch1="0"
yaw1="0"
                mass="0.700837"
```

```
                ixx="0.000816204" ixy="-1.97574e-20" ixz="0" iyy="0.000816204" iyz="0"
izz="0.00121262"
                r="0.1" g="0.1" b="0.1" a="1.0"
                x2="0" y2="3.46945e-18" z2="-0.025" roll2="-1.5708" pitch2="0" yaw2="-
1.73484e-17"/>

    <!-- Khớp liên tục cho bánh xe 3 gắn với base_link -->
    <xacro:joint joint_name="banh_xe_3_joint" type="continuous"
                 x1="-0.127563" y1="-0.14" z1="0.0641321" roll1="-1.5708"
pitch1="0.295982" yaw1="0"
                 parent_link="base_link"
                 child_link="banh_xe_3"
                 x2="0" y2="0" z2="1" effort="10" velocity="10"/>

    <!-- Định nghĩa bánh xe 4 -->
    <xacro:link link_name="banh_xe_4"
                x1="2.94422e-17" y1="-3.80821e-18" z1="0.00456292" roll1="0" pitch1="0"
yaw1="0"
                mass="0.700837"
                ixx="0.000816204" ixy="-4.37987e-22" ixz="-1.61979e-46" iyy="0.000816204"
iyz="1.99461e-45" izz="0.00121262"
                r="0.1" g="0.1" b="0.1" a="1.0"
                x2="2.77556e-17" y2="0" z2="-0.025" roll2="-1.5708" pitch2="7.68415e-43"
yaw2="1.4056e-18"/>

    <!-- Khớp liên tục cho bánh xe 4 gắn với base_link -->
    <xacro:joint joint_name="banh_xe_4_joint" type="continuous"
                 x1="0.132437" y1="-0.14" z1="0.0641321" roll1="1.5708" pitch1="0.033338"
yaw1="3.14159"
                 parent_link="base_link"
                 child_link="banh_xe_4"
                 x2="0" y2="0" z2="1" effort="10" velocity="10"/>

    <!-- Định nghĩa xích 1 -->
    <xacro:link link_name="xich_1"
                x1="2.90278e-16" y1="-4.8039e-17" z1="-0.03" roll1="0" pitch1="0" yaw1="0"
                mass="2.81002"
                ixx="0.00760602" ixy="1.11093e-06" ixz="0" iyy="0.0406757" iyz="0"
izz="0.0465957"
                r="0.1" g="0.1" b="0.1" a="1.0"
                x2="3.40439e-16" y2="-5.55112e-17" z2="0" roll2="1.5708" pitch2="0"
yaw2="0"/>

    <!-- Khớp cố định cho xích 1 gắn với base_link -->
    <xacro:joint joint_name="xich_1_joint" type="fixed"
                 x1="0.0025" y1="-0.105" z1="0.064" roll1="-1.5708" pitch1="1.82546e-16"
yaw1="0"
                 parent_link="base_link"
                 child_link="xich_1"/>

    <!-- Định nghĩa xích 2 -->
    <xacro:link link_name="xich_2"
                x1="-0.0432788" y1="-0.055" z1="-0.03" roll1="0" pitch1="0" yaw1="0"
                mass="2.81002"
                ixx="0.00760602" ixy="-1.11093e-06" ixz="0" iyy="0.0406757" iyz="0"
izz="0.0465957"
                r="0.1" g="0.1" b="0.1" a="1.0"
                x2="-0.0432788" y2="-0.055" z2="-0.06" roll2="-1.5708" pitch2="0"
yaw2="0"/>

    <!-- Khớp cố định cho xích 2 gắn với base_link -->
    <xacro:joint joint_name="xich_2_joint" type="fixed"
```

```
                x1="-0.0406" y1="0.105" z1="0.0091" roll1="-1.5708" pitch1="0"
yaw1="3.14159"
                parent_link="base_link"
                child_link="xich_2"/>

    <!-- Định nghĩa khâu 1 (đế của tay máy) -->
    <xacro:link link_name="khau_1"
                x1="-8.22159e-20" y1="-9.32786e-20" z1="-0.186482" roll1="0" pitch1="0"
yaw1="0"
                mass="1.23111"
                ixx="0.0507892" ixy="0" ixz="0" iyy="0.0507892" iyz="0" izz="0.00254127"
                r="1.0" g="1.0" b="1.0" a="1.0"
                x2="0" y2="3.33585e-18" z2="0" roll2="3.14159" pitch2="0" yaw2="0"/>

    <!-- Khớp cố định cho khâu 1 gắn với base_link -->
    <xacro:joint joint_name="khau_1_joint" type="fixed"
                x1="0.0649366" y1="4.67039e-17" z1="0.138132" roll1="3.14159" pitch1="0"
yaw1="0"
                parent_link="base_link"
                child_link="khau_1"/>

    <!-- Định nghĩa khớp -->
    <xacro:link link_name="khop"
                x1="7.63332e-19" y1="8.07171e-20" z1="-0.015" roll1="0" pitch1="0"
yaw1="0"
                mass="0.485965"
                ixx="0.000545572" ixy="7.21303e-49" ixz="0" iyy="0.000545572" iyz="0"
izz="0.00101825"
                r="0.13" g="0.44" b="0.70" a="1.0"
                x2="0" y2="-3.9801e-18" z2="-0.03" roll2="0" pitch2="0" yaw2="1.3617e-
45"/>

    <!-- Khớp tịnh tiến cho khâu 1 và khớp -->
    <xacro:joint joint_name="khop_joint" type="prismatic"
                x1="0" y1="2.80737e-25" z1="-0.210059" roll1="-3.14159" pitch1="6.59901e-
17" yaw1="-1.96037e-29"
                parent_link="khau_1"
                child_link="khop"
                x2="0" y2="0" z2="1" effort="100" velocity="10" lower="-0.135059"
upper="0.44"/>

    <!-- Định nghĩa khâu 2 -->
    <xacro:link link_name="khau_2"
                x1="0.228819" y1="-6.22477e-10" z1="-0.0150176" roll1="0" pitch1="0"
yaw1="0"
                mass="1.61532"
                ixx="0.0023024" ixy="-2.36835e-10" ixz="-6.71781e-06" iyy="0.0271882"
iyz="1.68742e-11" izz="0.0288969"
                r="1.0" g="1.0" b="1.0" a="1.0"
                x2="0" y2="-3.90313e-18" z2="-0.03" roll2="-5.99742e-33" pitch2="-
6.51514e-33" yaw2="-3.10412e-18"/>

    <!-- Khớp liên tục cho khớp và khâu 2 -->
    <xacro:joint joint_name="khau_2_joint" type="continuous"
                x1="0" y1="8.97939e-27" z1="0" roll1="-9.25178e-17" pitch1="2.59191e-16"
yaw1="-3.12038"
                parent_link="khop"
                child_link="khau_2"
                x2="0" y2="0" z2="1" effort="10" velocity="10"/>

    <!-- Thuộc tính Gazebo cho các link -->
```

```
    <xacro:gazebo reference="base_link" material="Gazebo/DarkGray" mu1="0.2" mu2="0.2"/>
<!-- Thân robot: xám đậm, ma sát thấp -->
    <xacro:gazebo reference="banh_xe_1" material="Gazebo/FlatBlack" mu1="1.0" mu2="1.0"/>
<!-- Bánh xe 1: đen, ma sát cao -->
    <xacro:gazebo reference="banh_xe_2" material="Gazebo/FlatBlack" mu1="1.0" mu2="1.0"/>
<!-- Bánh xe 2: đen, ma sát cao -->
    <xacro:gazebo reference="banh_xe_3" material="Gazebo/FlatBlack" mu1="1.0" mu2="1.0"/>
<!-- Bánh xe 3: đen, ma sát cao -->
    <xacro:gazebo reference="banh_xe_4" material="Gazebo/FlatBlack" mu1="1.0" mu2="1.0"/>
<!-- Bánh xe 4: đen, ma sát cao -->
    <xacro:gazebo reference="xich_1" material="Gazebo/FlatBlack" mu1="1.0" mu2="1.0"/> <!-
- Xích 1: đen, ma sát cao -->
    <xacro:gazebo reference="xich_2" material="Gazebo/FlatBlack" mu1="1.0" mu2="1.0"/> <!-
- Xích 2: đen, ma sát cao -->
    <xacro:gazebo reference="khau_1" material="Gazebo/White" mu1="0.2" mu2="0.2"/> <!--
Khâu 1: trắng, ma sát thấp -->
    <xacro:gazebo reference="khop" material="Gazebo/SkyBlue" mu1="0.2" mu2="0.2"/> <!--
Khớp: xanh nhạt, ma sát thấp -->
    <xacro:gazebo reference="khau_2" material="Gazebo/White" mu1="0.2" mu2="0.2"/> <!--
Khâu 2: trắng, ma sát thấp -->
</robot>
```

- **xe_tang.urdf.xacro**: Gồm tất cả các file Xacro trên, được gọi trong file launch để chạy

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro" name="xe-tang">
    <!-- Bao gồm file macro định nghĩa các hàm tái sử dụng (link, joint, arm_transmission,
base_transmission, gazebo) -->
    <xacro:include filename="macros.xacro"/>
    <!-- Bao gồm file cấu hình điều khiển cánh tay robot -->
    <xacro:include filename="arm_controller.xacro"/>
    <!-- Bao gồm file cấu hình điều khiển di chuyển robot -->
    <xacro:include filename="base_controller.xacro"/>
    <!-- Bao gồm file cấu hình camera của robot -->
    <xacro:include filename="camera.xacro"/>
    <!-- Bao gồm file cấu hình encoder cho odometry và điều khiển bánh xe -->
    <xacro:include filename="encoder.xacro"/>
    <!-- Bao gồm file cấu hình LIDAR cho quét môi trường -->
    <xacro:include filename="lidar.xacro"/>
    <!-- Bao gồm file định nghĩa cấu trúc chính của robot -->
    <xacro:include filename="robot.xacro"/>
</robot>
```

- Danh sách các link:

odom → base_link → banh_xe_1

→ banh_xe_2

→ banh_xe_3

→ banh_xe_4

→ xich_1

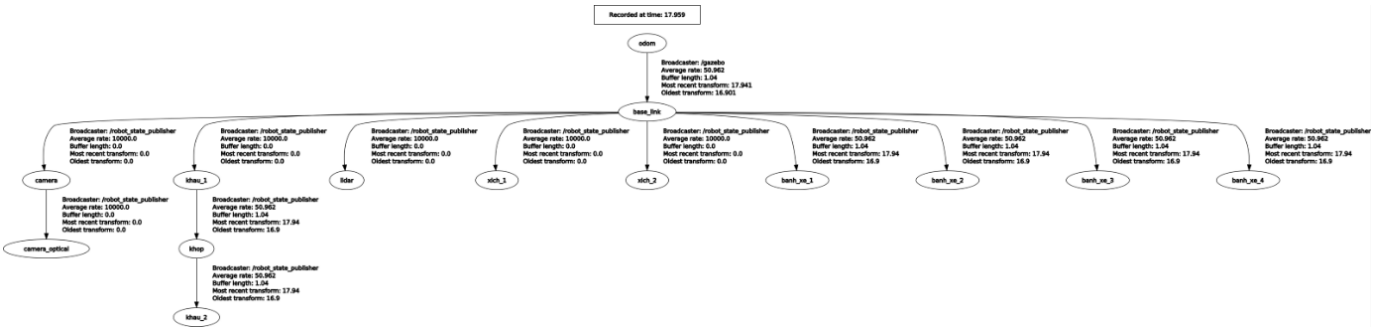
→ xich_2

→ lidar

→ camera → camera_optical

→ khau_1 → khop → khau_2

- Chạy lệnh `rosrun rqt_tf_tree rqt_tf_tree` hiện cây TF:



- Các cảm biến:
  - LIDAR:
    - Được gắn cố định vào thân chính của robot (base_link) thông qua khớp *lidar_joint*.
    - Sử dụng cảm biến tia *ray*, mô phỏng các tia laser để đo khoảng cách.
    - Thêm nhiễu Gaussian để mô phỏng nhiễu trong thực tế.
    - Sử dụng plugin *libgazebo_ros_laser.so* để mô phỏng hoạt động của LIDAR.
    - Dữ liệu quét được xuất bản lên topic */scan* dưới dạng message *sensor_msgs/LaserScan* và được RViz nhận để hiển thị các điểm quét.
  - Camera:
    - Được gắn cố định vào thân chính của robot (base_link) thông qua khớp *camera_joint*.
    - Sử dụng plugin *libgazebo_ros_camera.so*, hình ảnh thô được xuất bản lên topic */camera_pi/image_raw*, được RViz nhận để hiển thị hình ảnh từ camera, giúp quan sát môi trường từ góc nhìn của robot.
  - Encoder:
    - Được mô phỏng thông qua plugin *libgazebo_ros_diff_drive.so*, plugin nhận lệnh vận tốc từ topic */cmd_vel* và xuất bản lên topic */odom* dưới dạng

nav_msgs/Odometry bao gồm vị trí, hướng và vận tốc của robot trong hệ tọa độ *odom*.

- Ngoài ra, plugin xuất bản trạng thái của các khớp bánh xe lên topic */joint_states*, cung cấp thông tin về vị trí và vận tốc của bánh xe.

4. Mô tả cơ chế điều khiển trên Gazebo
    - Cơ chế điều khiển robot trên Gazebo được thực hiện thông qua plugin *libgazebo_ros_control.so*, được tích hợp trong các file Xacro. Plugin này cho phép sử dụng ROS Control để điều khiển bánh xe của robot thông qua *effort_controllers* và *diff_drive_controller*.
    - File cấu hình **arm_controller.yaml** và **base_controller.yaml** định nghĩa các controller để điều khiển bánh xe và các khớp của tay máy.
    - Các controller được khởi động bởi node */controller_spawner* thông qua file launch. Robot nhận lệnh vận tốc từ topic */diff_drive_controller/cmd_vel* và lực từ các topic */joint_1_position_controller/command* và */joint_2_position_controller/command* được Gazebo áp dụng vào mô phỏng vật lý.
    - Các transmission có chức năng định nghĩa các khớp trong Gazebo với controller để điều khiển.
    - Node điều khiển **teleop_keyboard.py** đăng ký vào các topic trên để có thể điều khiển robot và tay máy thông qua bàn phím.
    - Thêm plugin *libgazebo_ros_diff_drive.so* để tích hợp odometry, dùng để mô phỏng dữ liệu của encoder.

5. Các thành phần chính của code, cấu trúc của dự án
    - Cấu trúc của dự án:

```
xe_tang/
├── config/                    # Chứa các file cấu hình và RViz
│   ├── arm_controller.yaml
│   ├── base_controller.yaml
│   └── xe_tang.rviz
├── description/               # Chứa các file Xacro
│   ├── arm_controller.xacro
│   ├── base_controller.xacro
│   ├── camera.xacro
│   ├── encoder.xacro
│   ├── lidar.xacro
│   ├── macros.xacro
```

```
│   ├── robot.xacro
│   └── xe_tang.urdf.xacro
├── launch/                    # Chứa file launch
│   └── empty.world.launch
├── meshes/                    # Chứa file 3D
│   └── các file STL
├── scripts/                   # Chứa script điều khiển
│   ├── encoder.py
│   └── teleop_keyboard.py
├── worlds/                    # Chứa file world tạo môi trường trong Gazebo
│   └── empty.world
├── Bao_cao.pdf
├── CMakeLists.txt             # File xây dựng package
├── README.md
└── package.xml                # File mô tả package ROS
```

- Các thành phần chính của code:
  - **arm_controller.yaml**:

```
joint1_position_controller:
  type: effort_controllers/JointPositionController
  joint: khop_joint
  pid: {p: 100.0, i: 20.0, d: 30.0}
joint2_position_controller:
  type: effort_controllers/JointPositionController
  joint: khau_2_joint
  pid: {p: 100.0, i: 1.0, d: 10.0}
```

  - **base_controller.yaml**:

```
joint_state_controller:
    type: joint_state_controller/JointStateController
    publish_rate: 50
diff_drive_controller:
    type: diff_drive_controller/DiffDriveController
    left_wheel: ['banh_xe_4_joint', 'banh_xe_3_joint']
    right_wheel: ['banh_xe_2_joint', 'banh_xe_1_joint']
    pose_covariance_diagonal: [0.001, 0.001, 0.001, 0.001, 0.001, 0.01]
    twist_covariance_diagonal: [0.001, 0.001, 0.001, 0.001, 0.001, 0.01]
    publish_rate: 50
    wheel_separation: 0.135
    wheel_radius: 0.064075
    cmd_vel_timeout: 0.25
    velocity_rolling_window_size: 10
```

  - **empty.world.launch**:

```
<launch>
  <!-- Tải file cấu hình YAML cho bộ điều khiển cánh tay robot -->
  <rosparam command="load" file="$(find xe_tang)/config/arm_controller.yaml"/>
  <!-- Tải file cấu hình YAML cho bộ điều khiển base (di chuyển) của robot -->
  <rosparam command="load" file="$(find xe_tang)/config/base_controller.yaml"/>

  <!-- Sử dụng thời gian mô phỏng từ Gazebo -->
  <param name="/use_sim_time" value="true"/>

  <!-- Khởi chạy Gazebo với một thế giới trống -->
  <include file="$(find gazebo_ros)/launch/empty_world.launch">
```

```
    <!-- Đường dẫn đến file thế giới trống -->
    <arg name="world_name" value="$(find xe_tang)/worlds/empty.world"/>
    <!-- Không tạm dừng mô phỏng khi khởi chạy -->
    <arg name="paused" value="false"/>
    <!-- Đồng bộ thời gian mô phỏng với ROS -->
    <arg name="use_sim_time" value="true"/>
    <!-- Hiển thị giao diện đồ họa Gazebo -->
    <arg name="gui" value="true"/>
    <!-- Chạy Gazebo với giao diện (không headless) -->
    <arg name="headless" value="false"/>
    <!-- Không bật chế độ debug -->
    <arg name="debug" value="false"/>
  </include>

  <!-- Tạo mô tả robot từ file xacro và lưu vào tham số robot_description -->
  <param name="robot_description"
         command="$(find xacro)/xacro $(find xe_tang)/description/xe_tang.urdf.xacro"/>

  <!-- Tải mô hình robot vào Gazebo từ tham số robot_description -->
  <node name="spawn_model" pkg="gazebo_ros" type="spawn_model"
        args="-urdf -model xe_tang -param robot_description"/>

  <!-- Xuất bản trạng thái robot (TF transforms) từ mô tả robot -->
  <node name="robot_state_publisher" pkg="robot_state_publisher"
type="robot_state_publisher"/>

  <!-- Khởi chạy RViz với file cấu hình để trực quan hóa robot -->
  <node name="rviz" pkg="rviz" type="rviz" args="-d $(find xe_tang)/config/xe_tang.rviz"
required="true"/>

  <!-- Khởi chạy bộ điều khiển cho cánh tay robot (joint1 và joint2) -->
  <node name="arm_controller_spawner" pkg="controller_manager" type="spawner"
        respawn="true" output="screen"
        args="joint1_position_controller
              joint2_position_controller"/>

  <!-- Khởi chạy bộ điều khiển cho base robot (joint state và differential drive) -->
  <node name="base_controller_spawner" pkg="controller_manager" type="spawner"
        respawn="true" output="screen"
        args="joint_state_controller
              diff_drive_controller"/>

  <!-- Khởi chạy node Python để lắng nghe dữ liệu encoder -->
  <node name="encoder_listener" pkg="xe_tang" type="encoder.py" output="screen"/>
</launch>
```

- **encoder.py**:

```
#!/usr/bin/env python3

import rospy
from sensor_msgs.msg import JointState

def joint_state_callback(msg):
    # In vị trí của các khớp (mô phỏng encoder)
    for i, name in enumerate(msg.name):
        position = msg.position[i]  # Vị trí (radian)
        velocity = msg.velocity[i]  # Vận tốc (rad/s)
        rospy.loginfo(f"Joint: {name}, Position: {position:.3f} rad, Velocity:
{velocity:.3f} rad/s")
```

```
def listener():
    # Khởi tạo node
    rospy.init_node('encoder_listener')

    # Đăng ký subscriber để lắng nghe /joint_states
    rospy.Subscriber('/joint_states', JointState, joint_state_callback)

    # Giữ node chạy
    rospy.spin()

if __name__ == '__main__':
    try:
        listener()
    except rospy.ROSInterruptException:
        pass
```

- **teleop_keyboard.py**:

```
#!/usr/bin/env python3

from __future__ import print_function
import threading
import rospy
from geometry_msgs.msg import Twist
from std_msgs.msg import Float64
import sys
from select import select
import termios
import tty

# Thông báo hướng dẫn điều khiển robot qua bàn phím
msg = """
Dieu khien xe:
w - tien
s - lui
a - quay trai
d - quay phai

Dieu khien tay may:
i - nang
k - ha
j - quay trai
l - quay phai

f - thoat chuong trinh
"""

# Từ điển ánh xạ phím điều khiển di chuyển (x: quay, th: tiến/lùi)
moveBindings = {
    'w': (0, 1),  # Tiến
    's': (0, -1), # Lùi
    'a': (1, 0),  # Quay trái
    'd': (-1, 0)  # Quay phải
}

# Từ điển ánh xạ phím điều khiển khớp (joint1: nâng/hạ, joint2: quay)
jointBindings = {
    'i': (1, 0),  # Nâng joint1
    'k': (-1, 0), # Hạ joint1
    'j': (0, 1),  # Quay trái joint2
    'l': (0, -1), # Quay phải joint2
```

```
}

# Lớp luồng xuất bản dữ liệu điều khiển
class PublishThread(threading.Thread):
    def __init__(self, rate):
        super(PublishThread, self).__init__()
        # Publisher cho vận tốc base (differential drive)
        self.pub_base = rospy.Publisher('/diff_drive_controller/cmd_vel', Twist,
queue_size=10)
        # Publisher cho vị trí joint1 (cánh tay)
        self.pub_joint1 = rospy.Publisher('/joint1_position_controller/command', Float64,
queue_size=10)
        # Publisher cho vị trí joint2 (cánh tay)
        self.pub_joint2 = rospy.Publisher('/joint2_position_controller/command', Float64,
queue_size=10)

        # Khởi tạo các biến điều khiển
        self.x = 0.0        # Vận tốc góc (quay)
        self.th = 0.0       # Vận tốc tuyến tính (tiến/lùi)
        self.joint1_pos = 0.0  # Vị trí joint1
        self.joint2_pos = 0.0  # Vị trí joint2
        self.speed = 0.0    # Hệ số tốc độ tuyến tính
        self.turn = 0.0     # Hệ số tốc độ góc
        self.condition = threading.Condition()  # Điều kiện đồng bộ luồng
        self.done = False   # Cờ dừng luồng

        # Giới hạn vị trí joint1
        self.joint1_min = 0.0
        self.joint1_max = 0.5
        self.joint_step = 0.05  # Bước thay đổi vị trí khớp

        # Tính timeout dựa trên tần suất (rate)
        if rate != 0.0:
            self.timeout = 1.0 / rate
        else:
            self.timeout = None

        self.start()  # Bắt đầu luồng

    # Chờ subscriber kết nối trước khi xuất bản
    def wait_for_subscribers(self):
        while not rospy.is_shutdown() and self.pub_base.get_num_connections() == 0:
            rospy.sleep(0.5)
        if rospy.is_shutdown():
            raise Exception("Got shutdown request before subscribers connected")

    # Cập nhật giá trị điều khiển
    def update(self, x, th, joint1_pos, joint2_pos, speed, turn):
        self.condition.acquire()  # Khóa luồng
        self.x = x
        self.th = th
        # Giới hạn joint1_pos trong khoảng min/max
        self.joint1_pos = max(self.joint1_min, min(joint1_pos, self.joint1_max))
        self.joint2_pos = joint2_pos
        self.speed = speed
        self.turn = turn
        self.condition.notify()  # Thông báo cập nhật
        self.condition.release()  # Mở khóa luồng

    # Dừng luồng và đặt tất cả giá trị về 0
    def stop(self):
        self.done = True
```

```
        self.update(0, 0, self.joint1_pos, self.joint2_pos, 0, 0)
        self.join()

    # Hàm chạy luồng: xuất bản dữ liệu liên tục
    def run(self):
        twist = Twist()  # Tin nhắn Twist cho base
        while not self.done:
            self.condition.acquire()  # Khóa luồng
            self.condition.wait(self.timeout)  # Chờ timeout hoặc thông báo

            # Gán giá trị vận tốc cho Twist
            twist.linear.x = self.x * self.speed
            twist.angular.z = self.th * self.turn
            twist.linear.y = twist.linear.z = twist.angular.x = twist.angular.y = 0

            self.condition.release()  # Mở khóa luồng

            # Xuất bản dữ liệu
            self.pub_base.publish(twist)
            self.pub_joint1.publish(self.joint1_pos)
            self.pub_joint2.publish(self.joint2_pos)

        # Dừng robot khi thoát
        twist.linear.x = 0
        twist.angular.z = 0
        self.pub_base.publish(twist)

# Lấy phím nhấn từ bàn phím với timeout
def getKey(settings, timeout):
    tty.setraw(sys.stdin.fileno())  # Chuyển terminal sang chế độ raw
    rlist, _, _ = select([sys.stdin], [], [], timeout)  # Chờ đầu vào
    key = sys.stdin.read(1) if rlist else ''  # Đọc phím nếu có
    termios.tcsetattr(sys.stdin, termios.TCSADRAIN, settings)  # Khôi phục cài đặt
terminal
    return key

# Lưu cài đặt terminal hiện tại
def saveTerminalSettings():
    return termios.tcgetattr(sys.stdin)

# Khôi phục cài đặt terminal
def restoreTerminalSettings(old_settings):
    if old_settings:
        termios.tcsetattr(sys.stdin, termios.TCSADRAIN, old_settings)

# Chương trình chính
if __name__ == "__main__":
    settings = saveTerminalSettings()  # Lưu cài đặt terminal
    rospy.init_node('keyboard_control')  # Khởi tạo node ROS

    # Lấy tham số từ ROS parameter server
    speed = rospy.get_param("~speed", 10.0)  # Tốc độ tuyến tính mặc định
    turn = rospy.get_param("~turn", 10.0)    # Tốc độ góc mặc định
    repeat = rospy.get_param("~repeat_rate", 0.0)  # Tần suất lặp
    key_timeout = rospy.get_param("~key_timeout", 0.5)  # Thời gian chờ phím

    pub_thread = PublishThread(repeat)  # Khởi tạo luồng xuất bản
    x = 0  # Vận tốc góc ban đầu
    th = 0  # Vận tốc tuyến tính ban đầu
    joint1_pos = 0.0  # Vị trí joint1 ban đầu
    joint2_pos = 0.0  # Vị trí joint2 ban đầu
    status = 0  # Trạng thái chương trình
```

```
    try:
        pub_thread.wait_for_subscribers()  # Chờ subscriber kết nối
        pub_thread.update(x, th, joint1_pos, joint2_pos, speed, turn)  # Cập nhật giá trị
ban đầu

        print(msg)  # In hướng dẫn điều khiển

        # Vòng lặp chính: xử lý phím nhấn
        while not rospy.is_shutdown():
            key = getKey(settings, key_timeout)  # Lấy phím nhấn

            if key in moveBindings:  # Điều khiển di chuyển
                x = moveBindings[key][0]
                th = moveBindings[key][1]
            elif key in jointBindings:  # Điều khiển khớp
                joint1_pos += jointBindings[key][0] * pub_thread.joint_step
                joint2_pos += jointBindings[key][1] * pub_thread.joint_step
            elif key == 'f' or key == '\x03':  # Thoát bằng 'f' hoặc Ctrl+C
                break
            else:
                if key == '' and x == 0 and th == 0:  # Không làm gì nếu không có phím
                    continue
                x = 0  # Đặt lại vận tốc nếu phím không hợp lệ
                th = 0

            pub_thread.update(x, th, joint1_pos, joint2_pos, speed, turn)  # Cập nhật giá
trị điều khiển

    except Exception as e:
        print(e)  # In lỗi nếu có

    finally:
        pub_thread.stop()  # Dừng luồng xuất bản
        restoreTerminalSettings(settings)  # Khôi phục cài đặt terminal
```