Voici un guide tout-doux, étape par étape, pour intégrer ton modèle LIMO Pro dans Gazebo avec ROS 2, même si tu débutes complètement. À la fin, tu trouveras des liens vers des ressources pour approfondir chaque aspect.

**1. Créer le dossier du modèle (limo\_description)**

1. **Quoi ?**  
   Un « package » ROS 2, c’est simplement un dossier structuré où tu mets tout ce qui concerne un composant : code, fichiers de configuration, descriptions de robot, etc.
2. **Pourquoi ?**  
   Gazebo et ROS s’attendent à retrouver, dans un package, des sous-dossiers bien nommés (comme urdf/ pour le robot, launch/ pour les scripts de lancement).
3. **Comment ?**
   * Va dans le dossier src/ de ton workspace (c’est là où ROS 2 cherche les packages).
   * Tape la commande :
   * ros2 pkg create limo\_description --build-type ament\_cmake --dependencies xacro gazebo\_ros
   * Cela crée un dossier limo\_description/ avec des fichiers de base (package.xml, CMakeLists.txt).

**2. Préparer la description du robot (URDF/XACRO)**

1. **Quoi ?**
   * **URDF** : un format XML qui décrit la forme (géométrie), la masse et les articulations de ton robot.
   * **XACRO** : un « macro-URDF » qui te permet d’écrire moins de lignes et de réutiliser des paramètres.
2. **Pourquoi ?**  
   Gazebo lit les fichiers URDF pour savoir à quoi ressemble ton robot (boîtes, cylindres, maillages 3D) et comment ses pièces sont reliées.
3. **Comment ?**
   * Dans limo\_description/, crée un dossier urdf/.
   * Dans urdf/, crée un fichier limo\_pro.urdf.xacro et colle ce squelette minimal :
   * <?xml version="1.0"?>
   * <robot name="limo\_pro" xmlns:xacro="http://ros.org/wiki/xacro">
   * <!-- Paramètres réutilisables -->
   * <xacro:property name="wheel\_radius" value="0.15"/>
   * <!-- Corps principal -->
   * <link name="base\_link">
   * <visual><geometry><box size="0.6 0.4 0.2"/></geometry></visual>
   * <!-- collision et inertial ici -->
   * </link>
   * <!-- Ex. roue avant gauche -->
   * <link name="wheel\_fl">
   * <visual><geometry><cylinder radius="${wheel\_radius}" length="0.05"/></geometry></visual>
   * </link>
   * <joint name="joint\_fl" type="continuous">
   * <parent link="base\_link"/>
   * <child link="wheel\_fl"/>
   * <origin xyz="0.3 0.2 -0.1"/>
   * <axis xyz="0 1 0"/>
   * </joint>
   * </robot>
   * Tu pourras ensuite dupliquer les blocs pour les autres roues, ou remplacer box/cylinder par tes propres fichiers STL dans un dossier meshes/.

**3. Dire à CMake d’installer tes fichiers**

1. **Quoi ?**  
   Quand tu fais colcon build, ROS compile et installe tout dans un dossier install/. Il faut indiquer à CMake quels dossiers copier là-dedans.
2. **Comment ?**
   * Ouvre limo\_description/CMakeLists.txt.
   * Ajoute, après les find\_package(...) :
   * install(
   * DIRECTORY urdf meshes launch
   * DESTINATION share/${PROJECT\_NAME}
   * )
   * Vérifie qu’en bas il y a ament\_package().

**4. Écrire un fichier de lancement pour Gazebo**

1. **Quoi ?**  
   Un script Python (.launch.py) qui dit à ROS 2 :
   * « Démarre Gazebo »
   * « Fais apparaître mon robot dedans »
2. **Comment ?**
   * Dans limo\_description/launch/, crée spawn\_limo.launch.py.
   * Colle ce code simple :
   * from launch import LaunchDescription
   * from launch.actions import ExecuteProcess
   * from launch\_ros.actions import Node
   * from ament\_index\_python.packages import get\_package\_share\_directory
   * import os
   * def generate\_launch\_description():
   * pkg = get\_package\_share\_directory('limo\_description')
   * xacro\_file = os.path.join(pkg, 'urdf', 'limo\_pro.urdf.xacro')
   * return LaunchDescription([
   * ExecuteProcess(
   * cmd=['gazebo', '--verbose', '-s', 'libgazebo\_ros\_factory.so'],
   * output='screen'
   * ),
   * Node(
   * package='gazebo\_ros',
   * executable='spawn\_entity.py',
   * arguments=['-entity', 'limo\_pro', '-file', xacro\_file],
   * output='screen'
   * ),
   * ])

**5. Compiler et lancer la simulation**

1. **Compiler**  
   Retourne à la racine de ton dépôt (le workspace) et fais :
2. colcon build --packages-select limo\_description
3. **Charger l’environnement**  
   Chaque fois que tu ouvres un nouveau terminal :
4. source install/setup.bash
5. **Lancer**
6. ros2 launch limo\_description spawn\_limo.launch.py

— tu devrais voir Gazebo se lancer et ton robot apparaître dans le monde.

**Ressources pour aller plus loin**

* **ROS 2 Beginner Tutorials**  
  <https://docs.ros.org/en/jazzy/Tutorials.html>
* **Understanding URDF and XACRO**  
  <https://wiki.ros.org/urdf/Tutorials>
* **Gazebo with ROS 2**  
  <https://gazebosim.org/tutorials?tut=ros2_overview>
* **Launch system in ROS 2**  
  <https://docs.ros.org/en/jazzy/Tutorials/Launch.html>

Ces pages te guideront, pas à pas, avec des exemples concrets et des explications plus détaillées. Bon courage pour la suite !