1. MoveIt2 configuration

Note: The factory image of this course has been configured. If you use the factory image system, you can skip the moveit configuration course.

1.1. Install MoveIt2

```
sudo apt install ros-humble-moveit*
```

The Jetson Orin series motherboard can run Moveit2 related cases directly on the motherboard, and the overall fluency is acceptable!

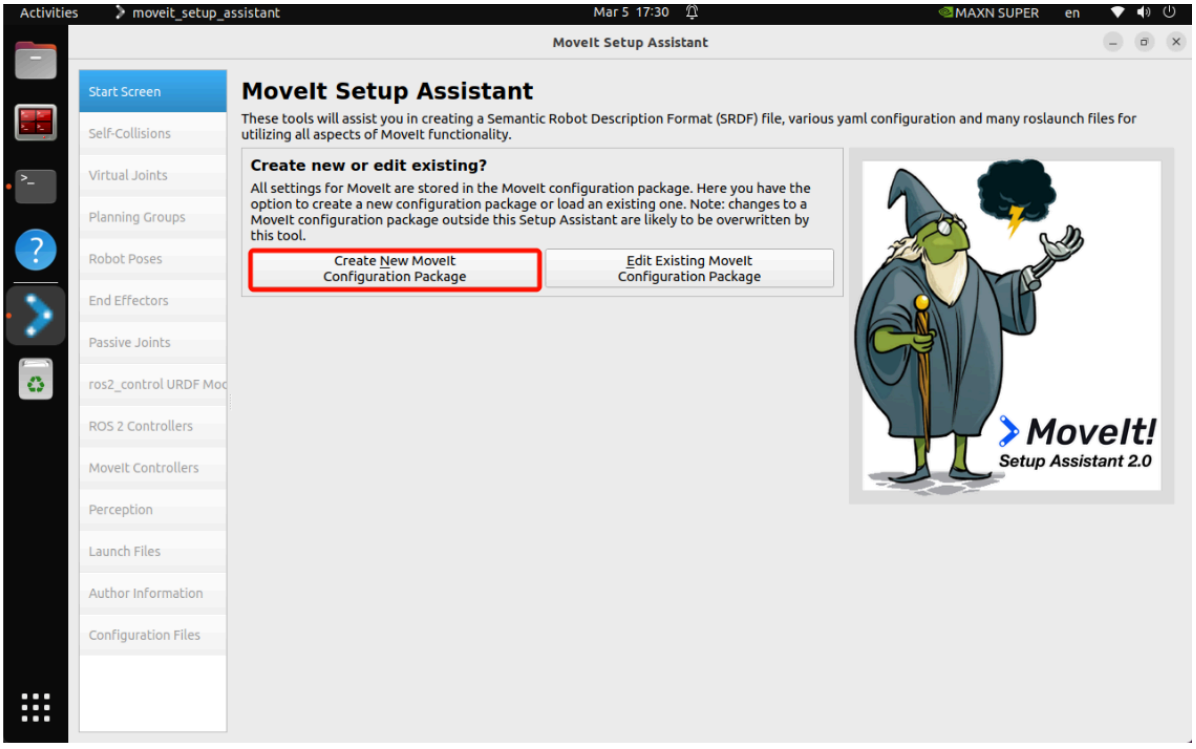
Open another terminal.

1.2. Start the assistant

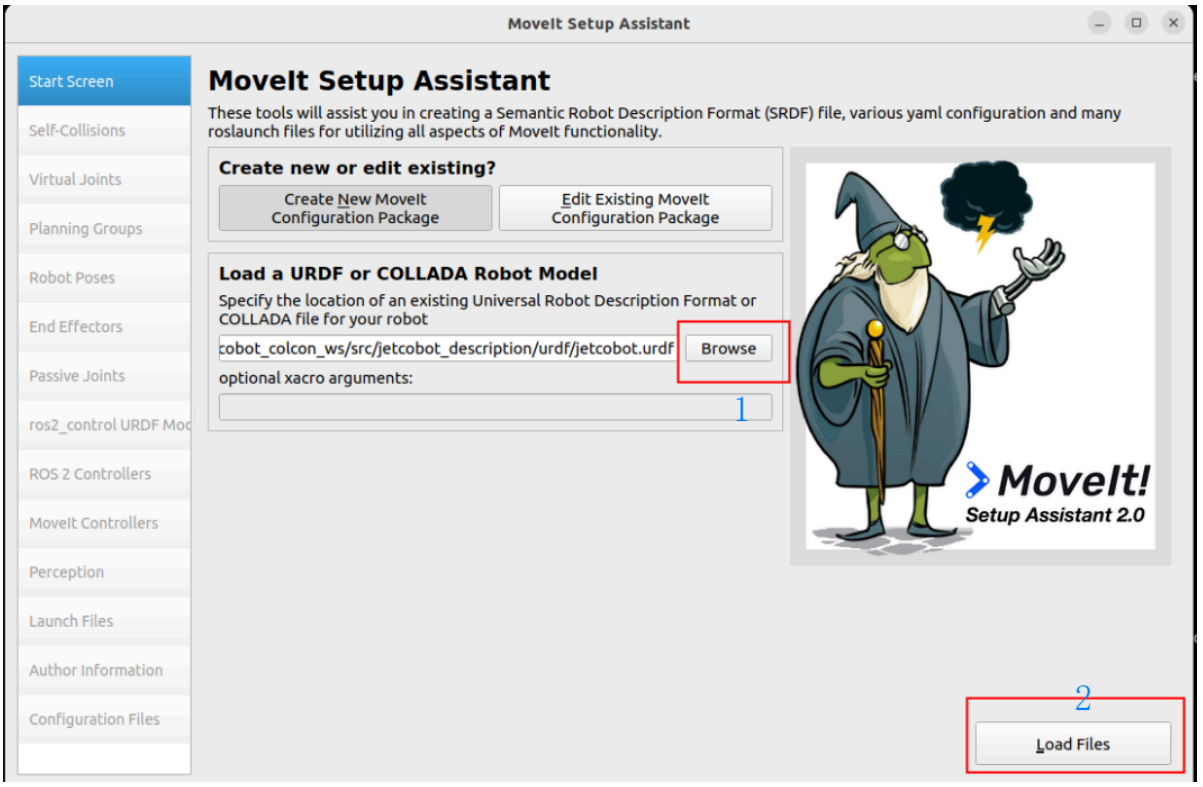
Start MoveIt Setup Assistant:

```
ros2 launch moveit_setup_assistant setup_assistant.launch.py
```

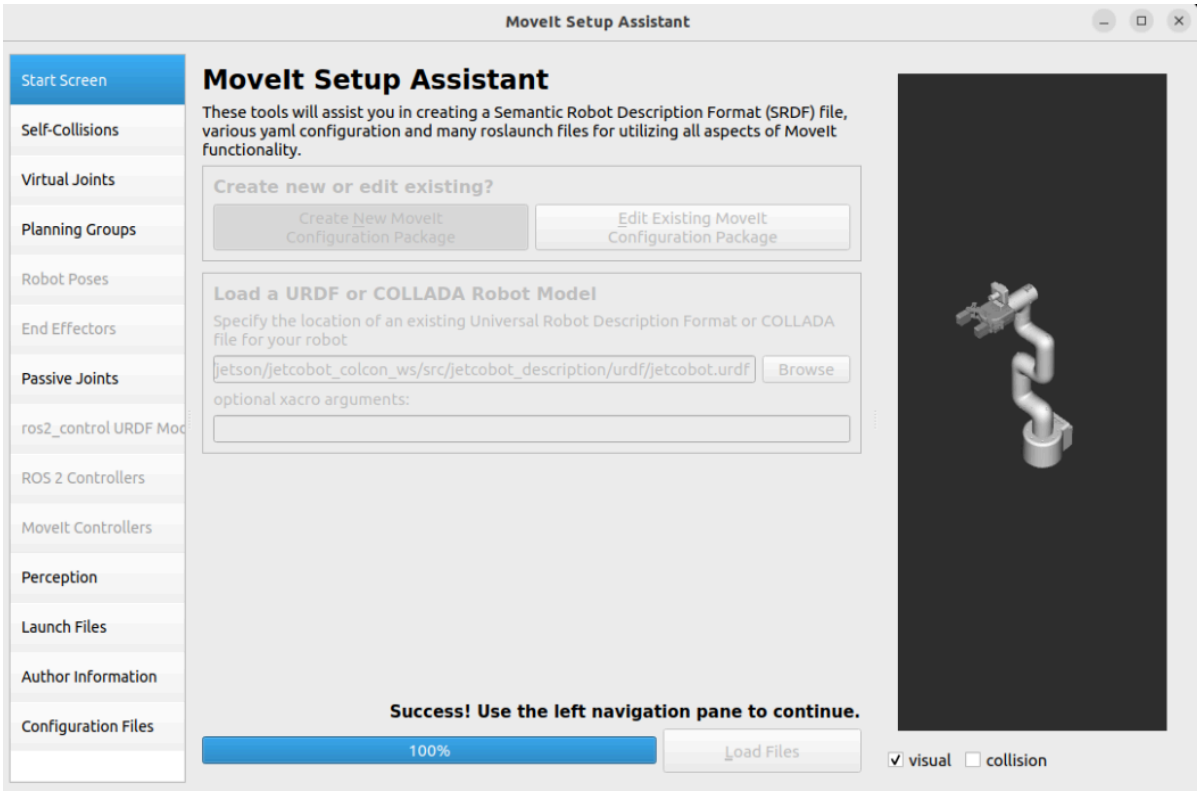
For the first time, you can choose to create a new MoveIt configuration package. For later simple modifications, you can choose to compile the existing MoveIt software package:



To create a new MoveIt configuration package, you need to import the robot's URDF model: Simply importing the URDF model file into the MoveIt assistant will result in an error. You need Use the compiled software package (jetcobot_description) URDF file path



After selecting the URDF file, click Load File, and the MoveIt assistant will display the robot model:

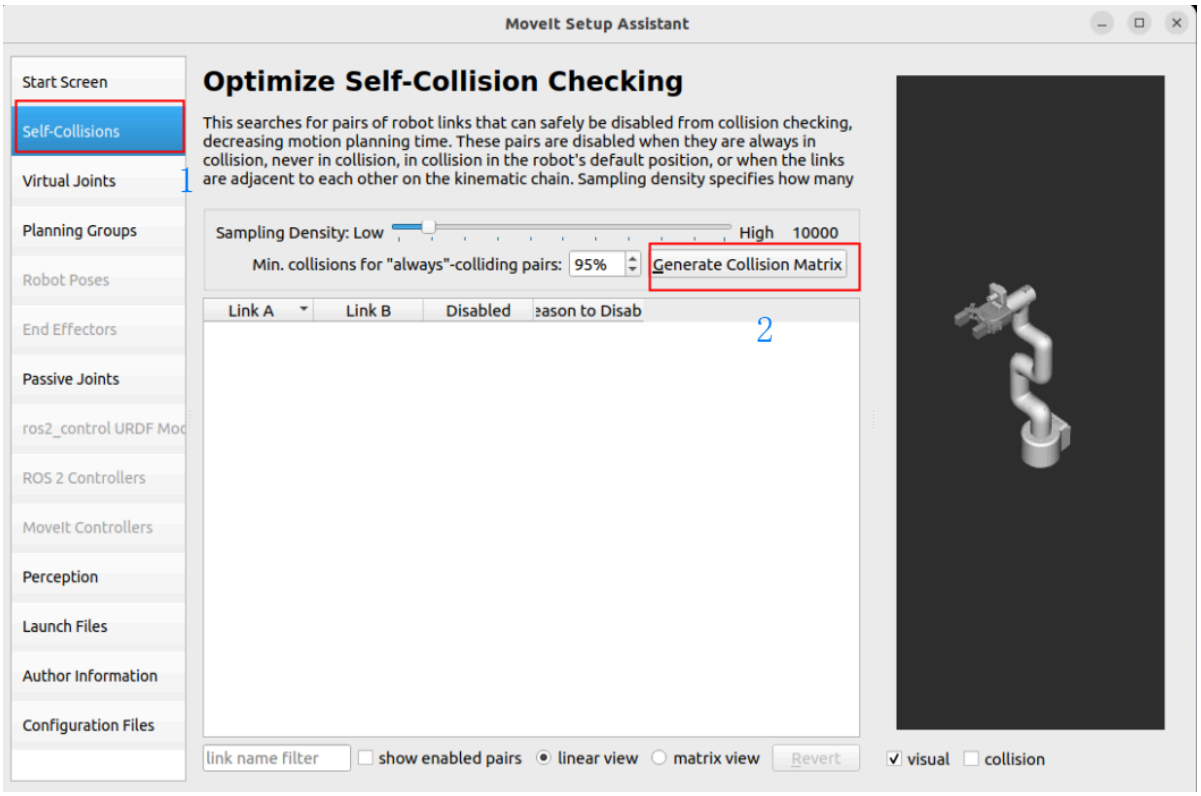


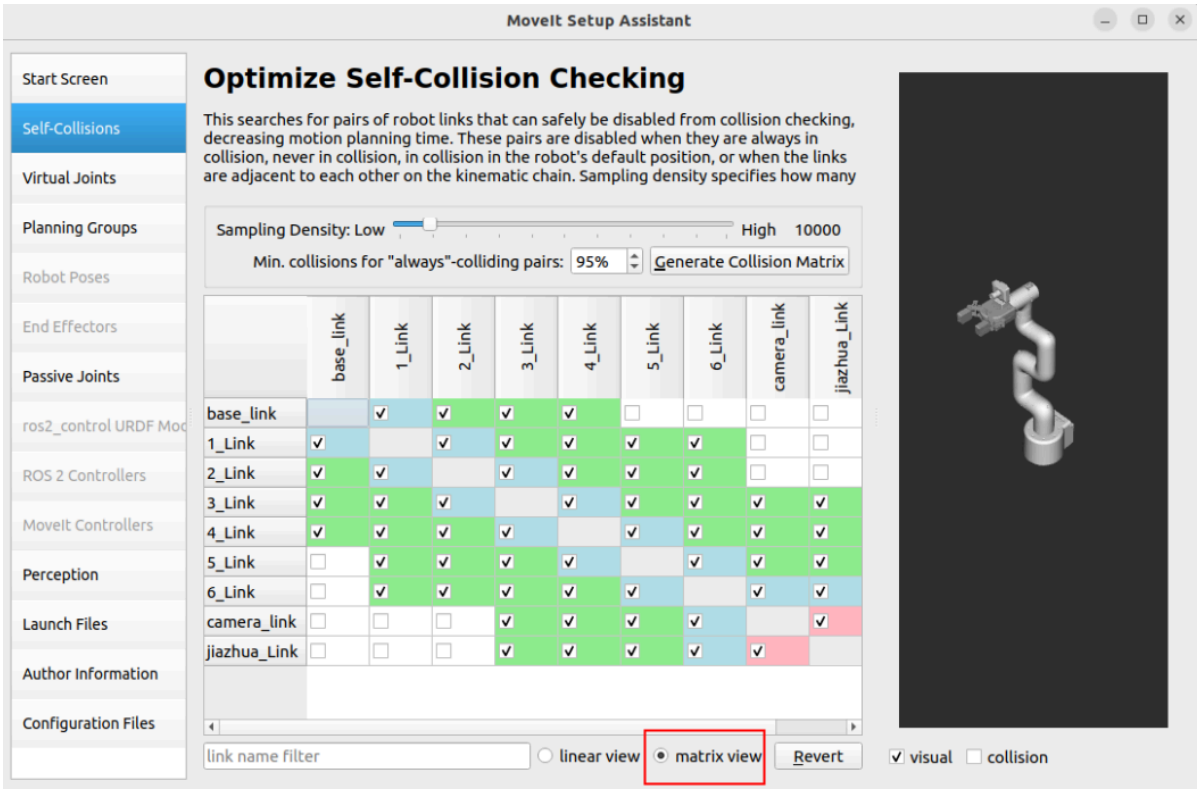
1.3, Collision Detection

The self-collision matrix is a function used to optimize motion planning. Its main function is to generate a matrix to describe whether collisions may occur between the various links in the robot model.

This matrix can help the motion planner avoid unnecessary collision detection when planning the path, thereby improving planning efficiency.

The self-collision matrix can be automatically generated by clicking the MoveIt Setup Assistant option:

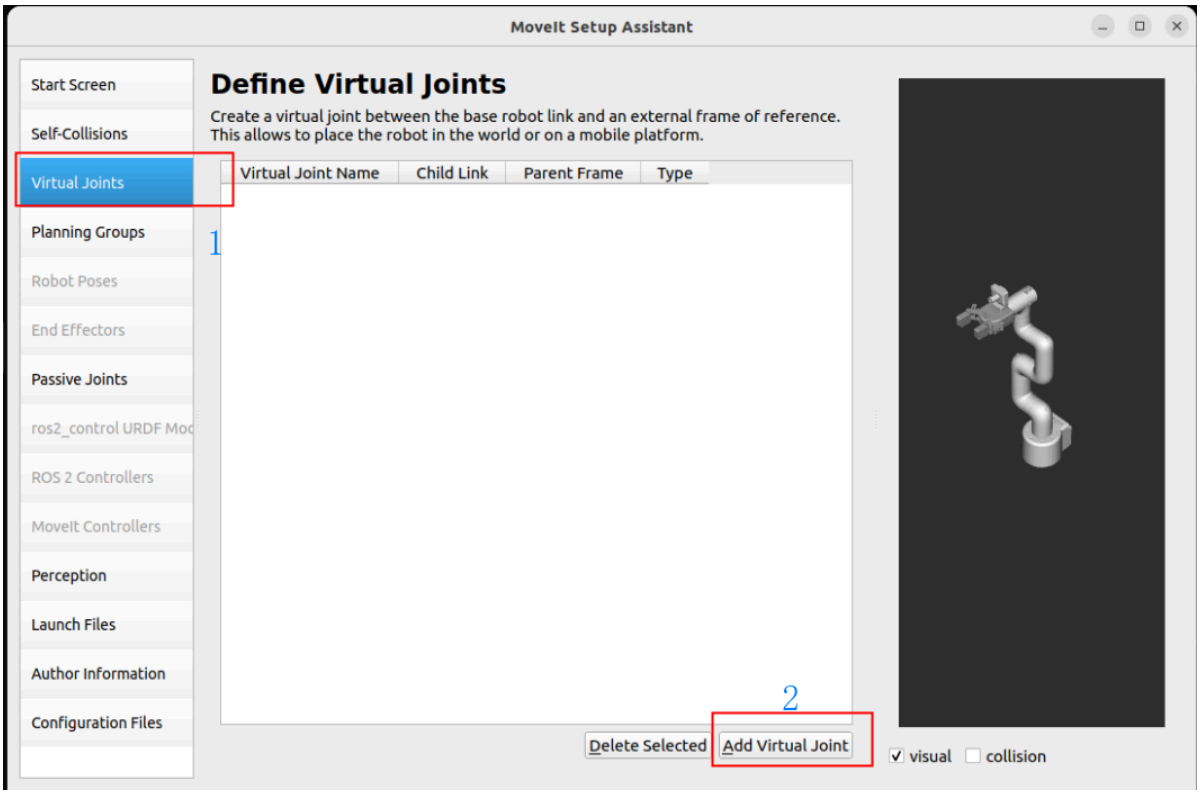




1.4. Virtual joints

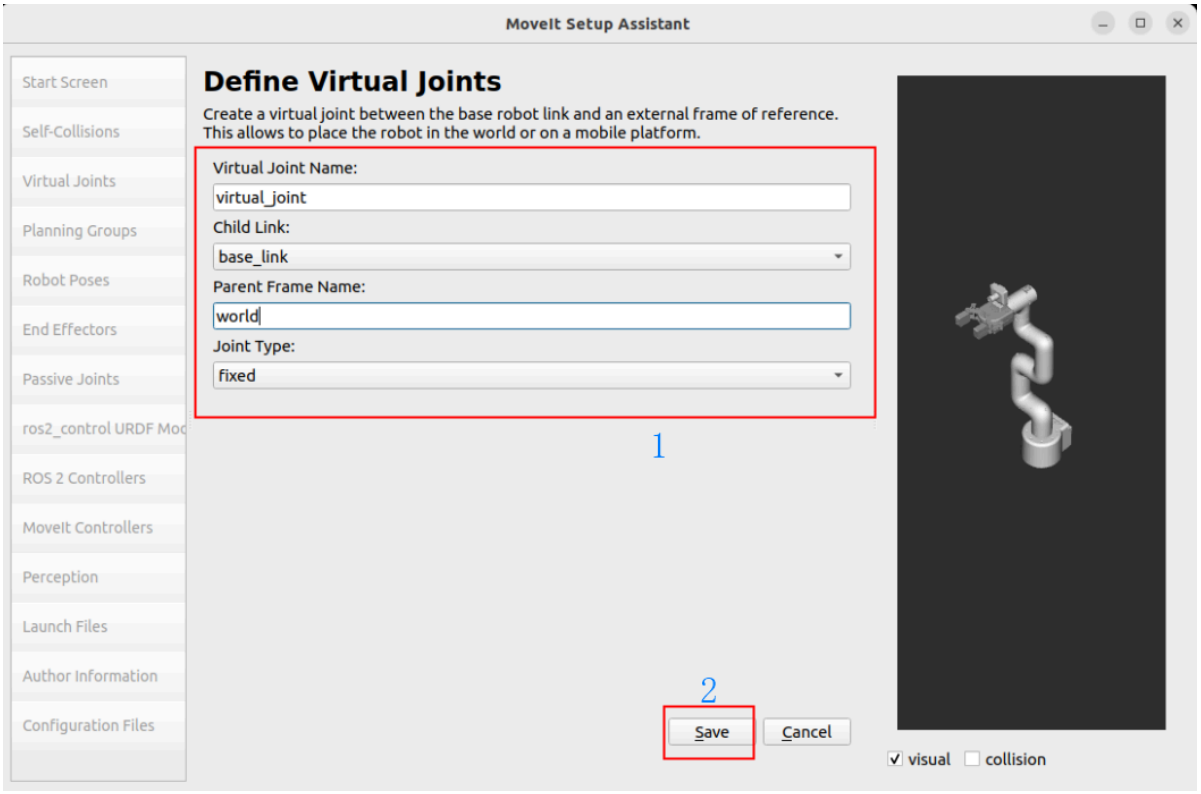
A virtual joint is a tool used to define the relationship between a robot model and the outside world.

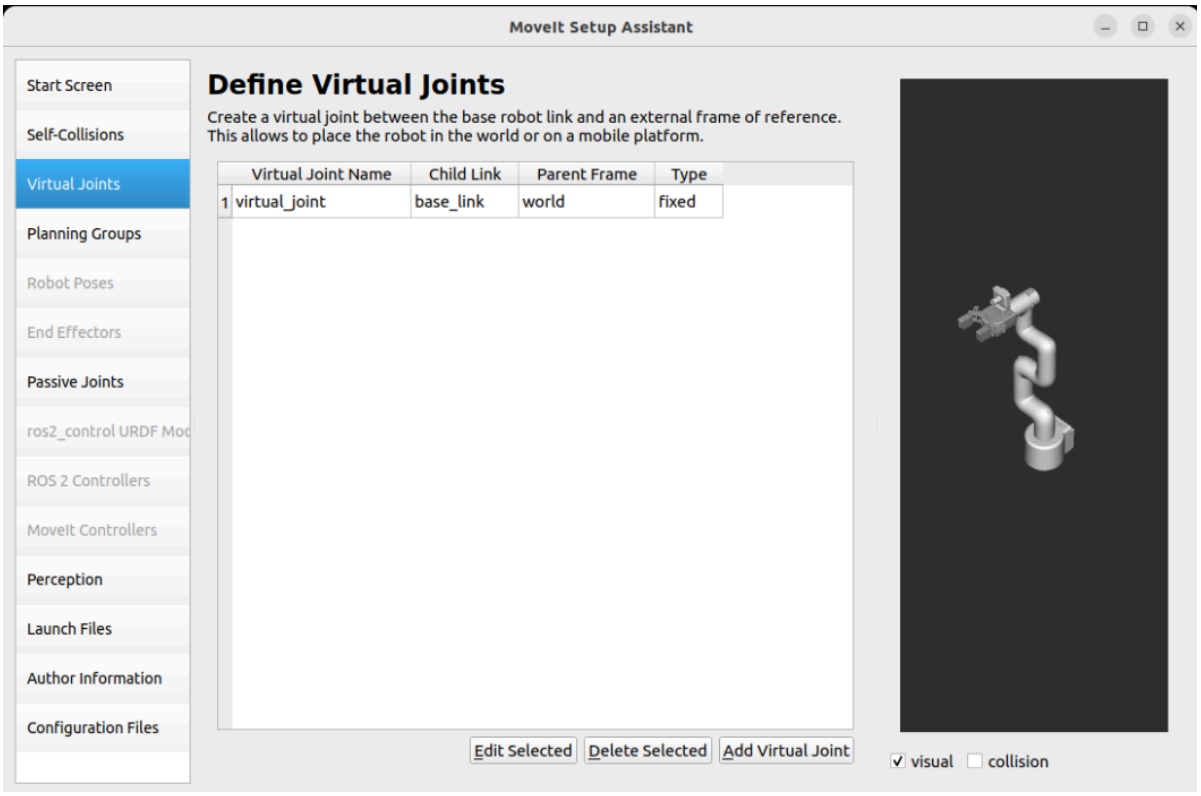
Virtual joints are usually used to describe the connection between the robot base and a fixed reference system (such as the world coordinate system).



Fix the robot base link `base_link` and `world`:

- Virtual Joint Name: virtual_joint
- Child Link: base_link
- Parent Frame Name: world
- Joint Type: fixed

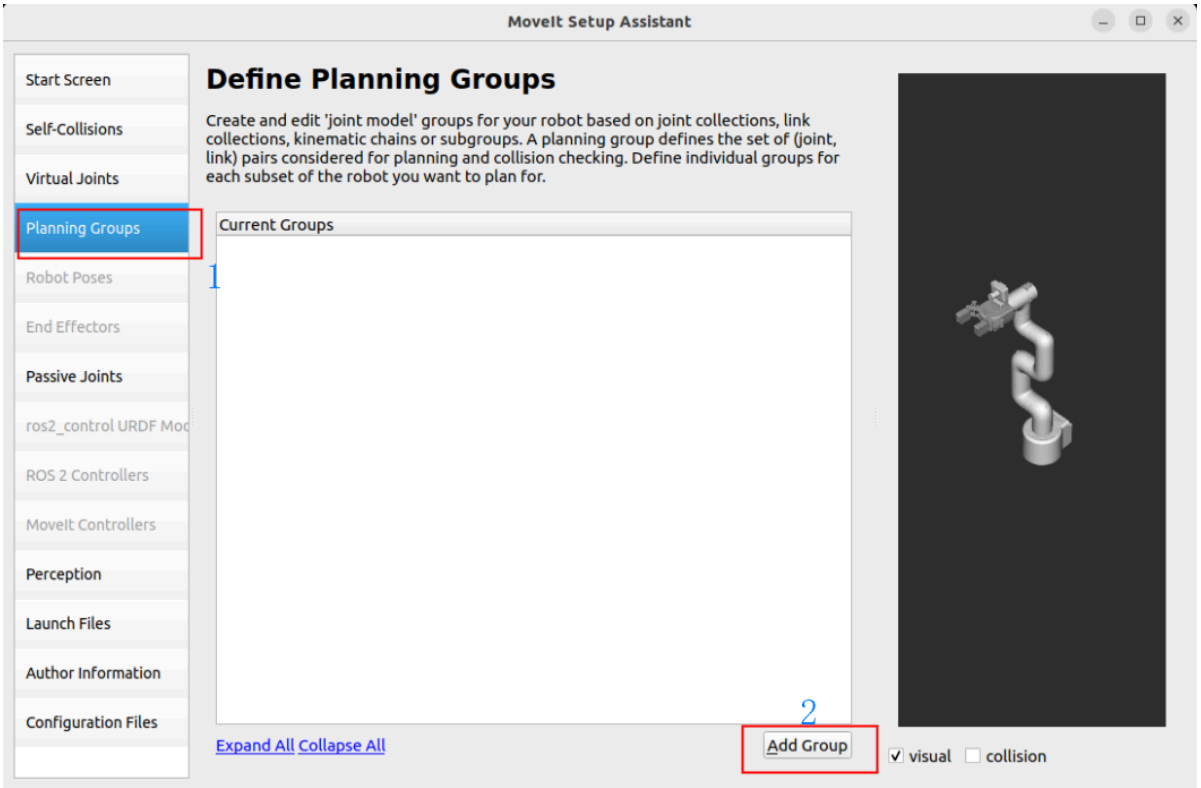




1.5, Motion Planning Group

The planning group is a key step in configuring the robot motion planning.

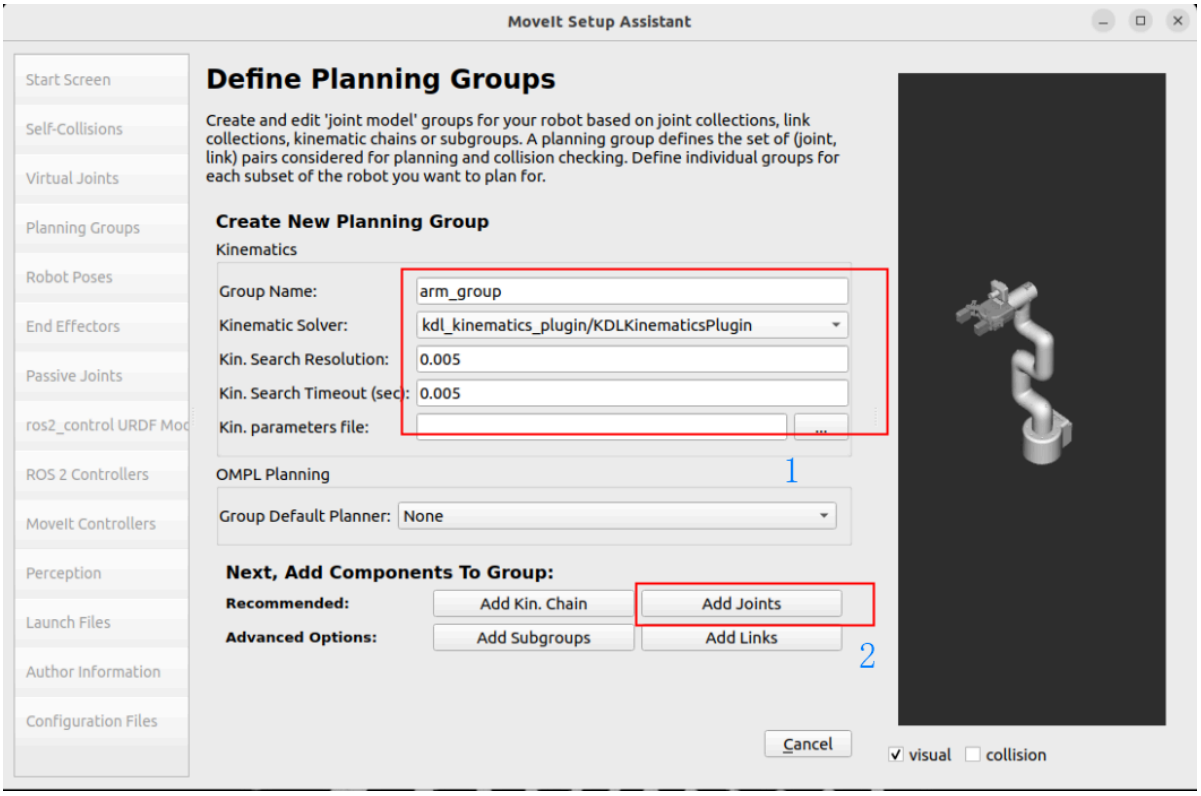
The planning group defines which joints and links in the robot can move together and how to plan their motion.



Planning robot arm group: arm_group

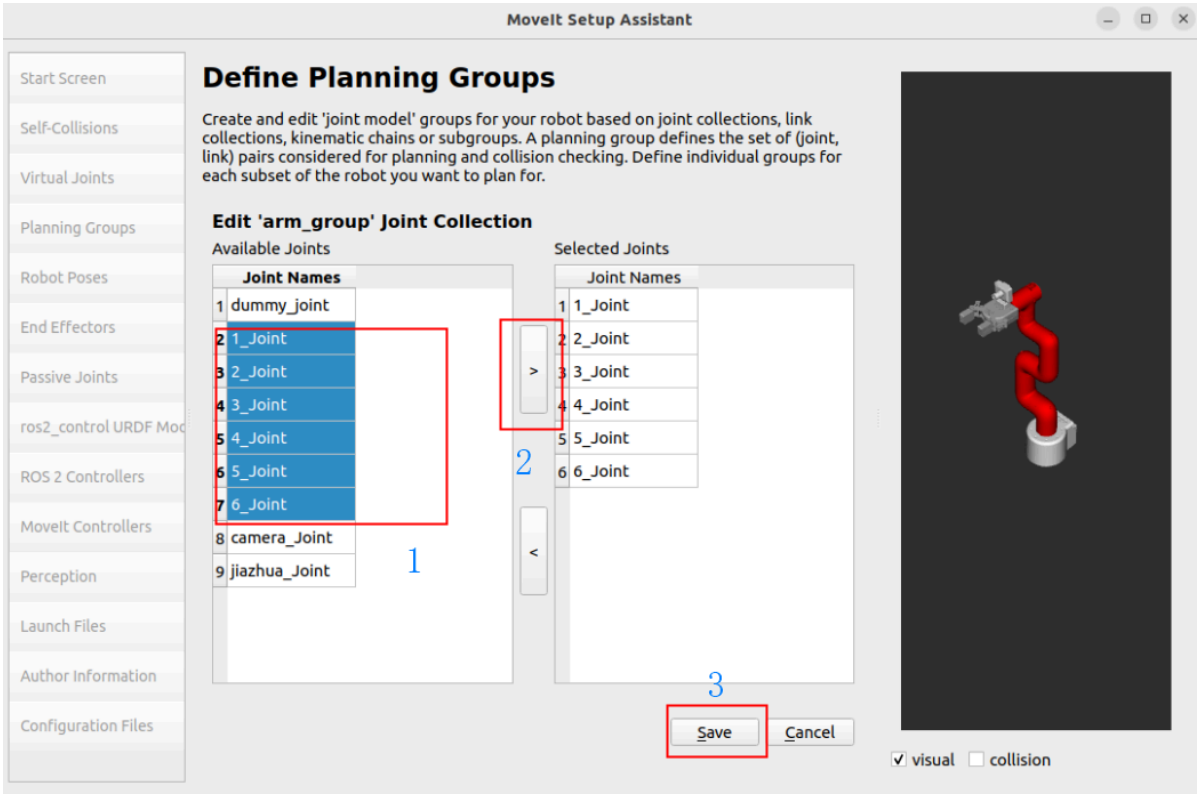
- Group Name: arm_group
- Kinematic Solver: kdl_kinematics_plugin/KDLKinematicsPlugin
- Kin. Search Resolution (sampling density of joint space): 0.005

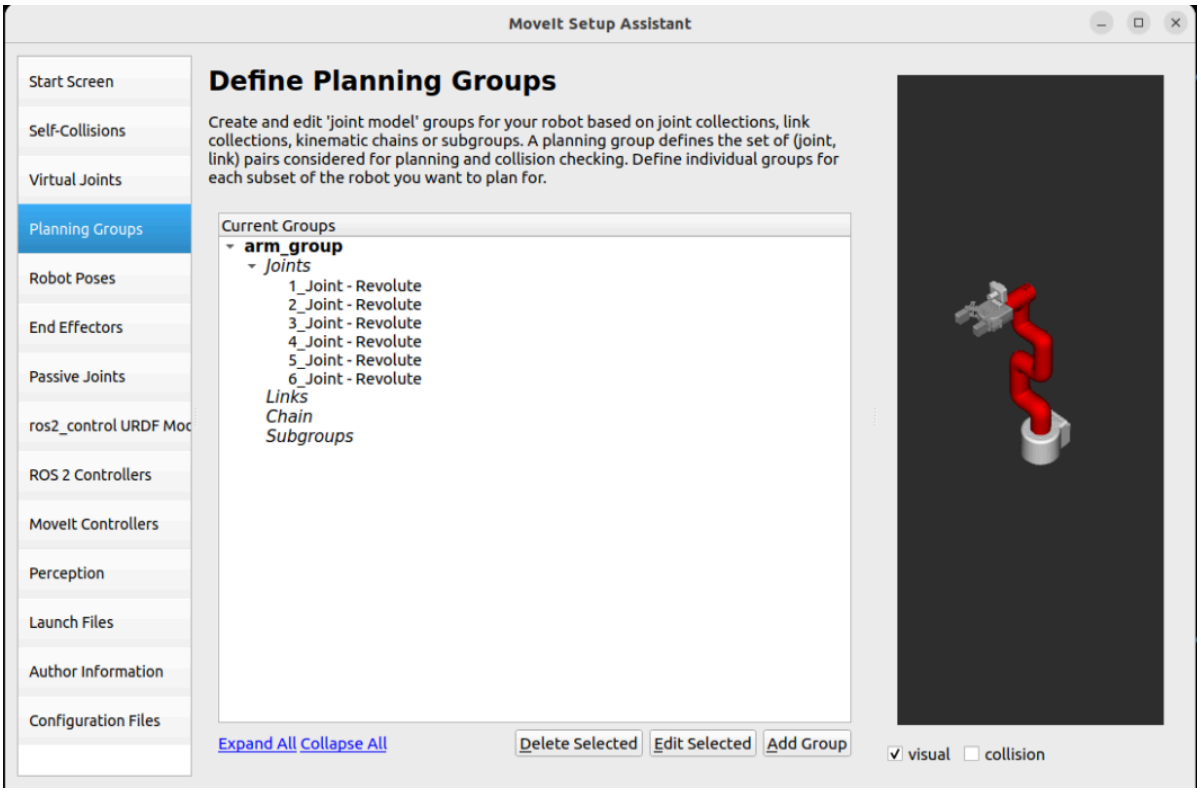
- Kin. Search Timeout (solution time): 0.005



- Add Joints: Select the joints of the robot arm

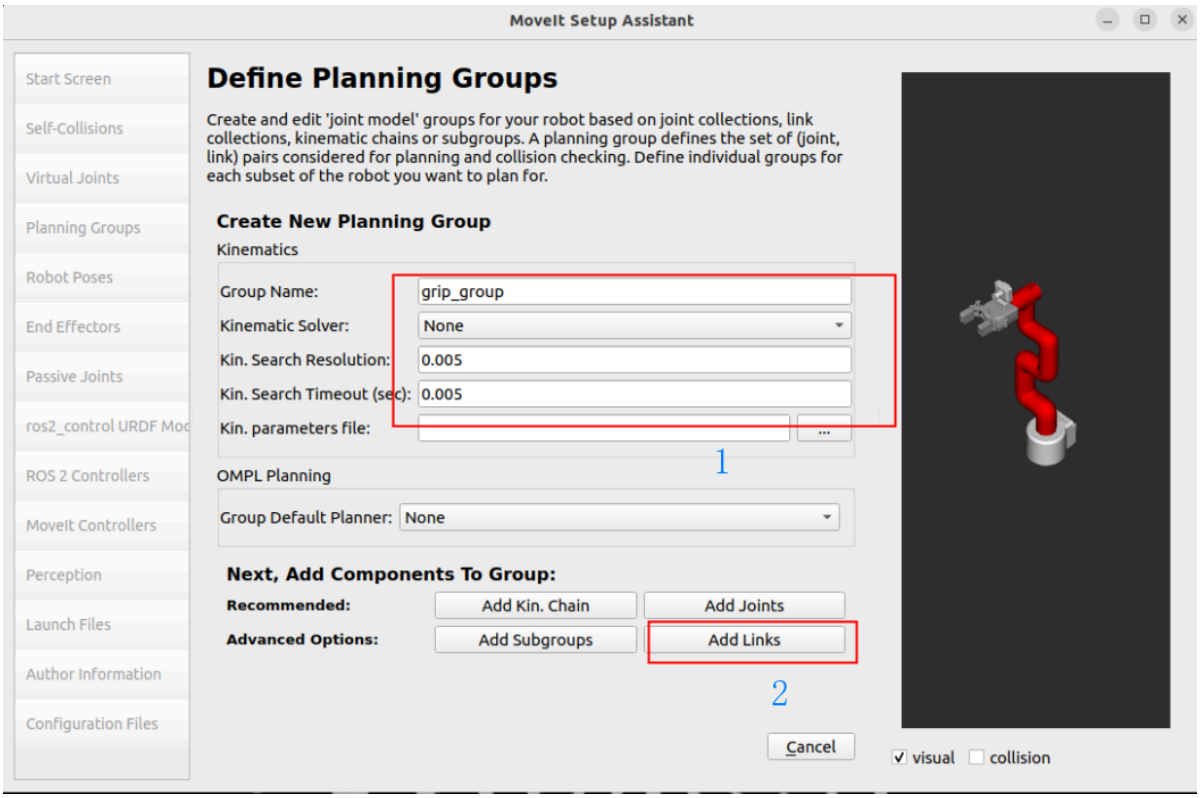
Joints: 1_Joint, 2_Joint, 3_Joint, 4_Joint, 5_Joint, 6_Joint

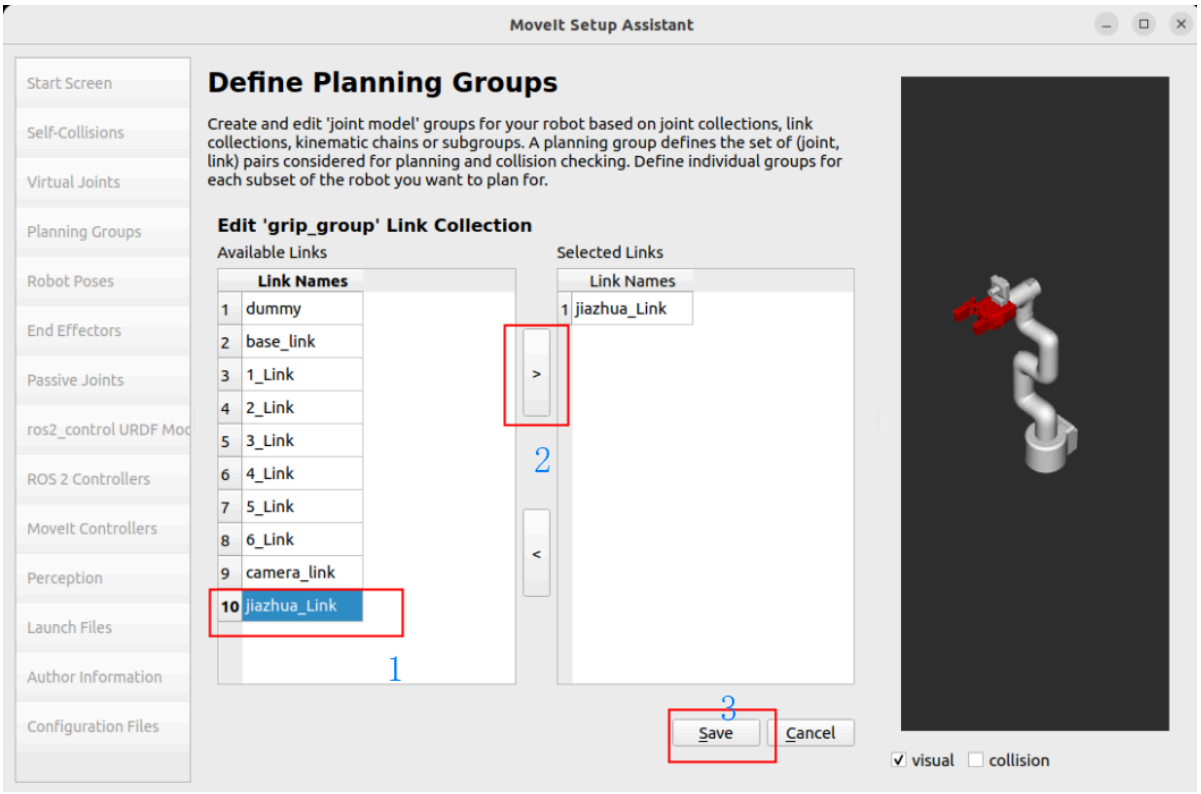


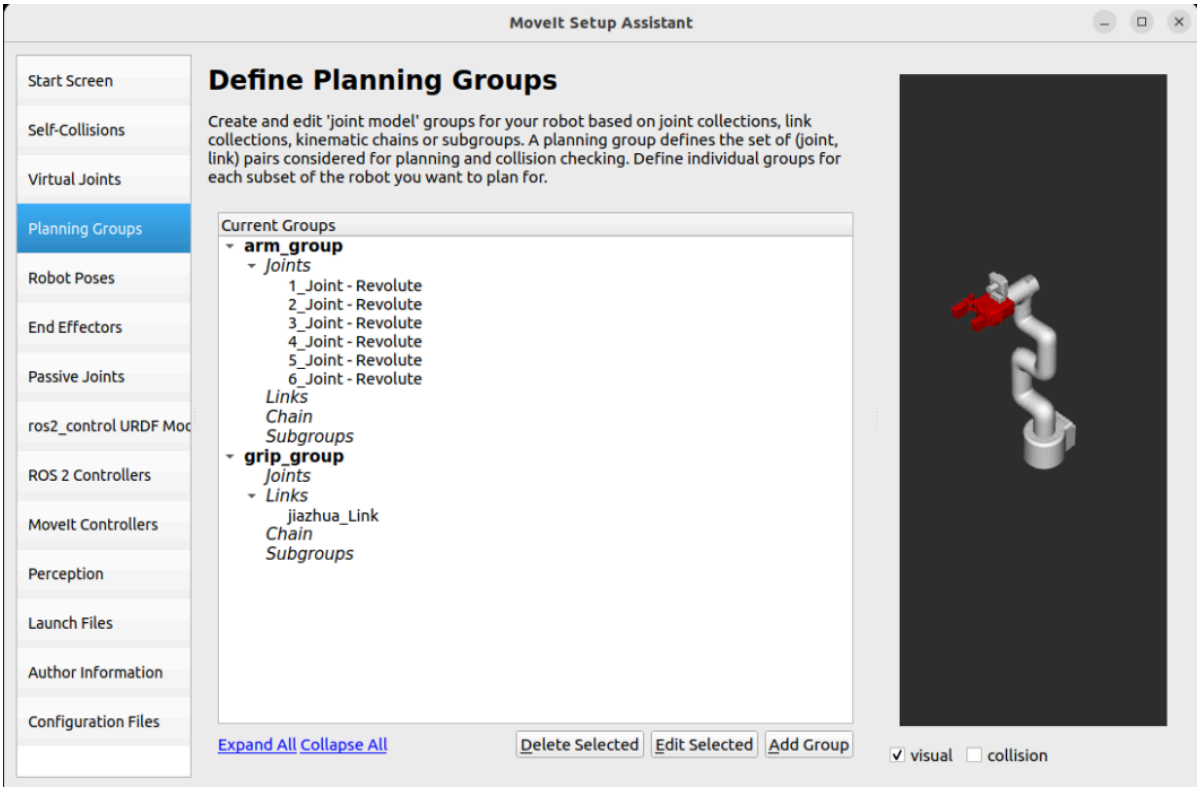


Plan gripper group: grip_group

- Group Name: grip_group
- Kinematic Solver: None
- Kin. Search Resolution: 0.005
- Kin. Search Timeout (solving time): 0.005

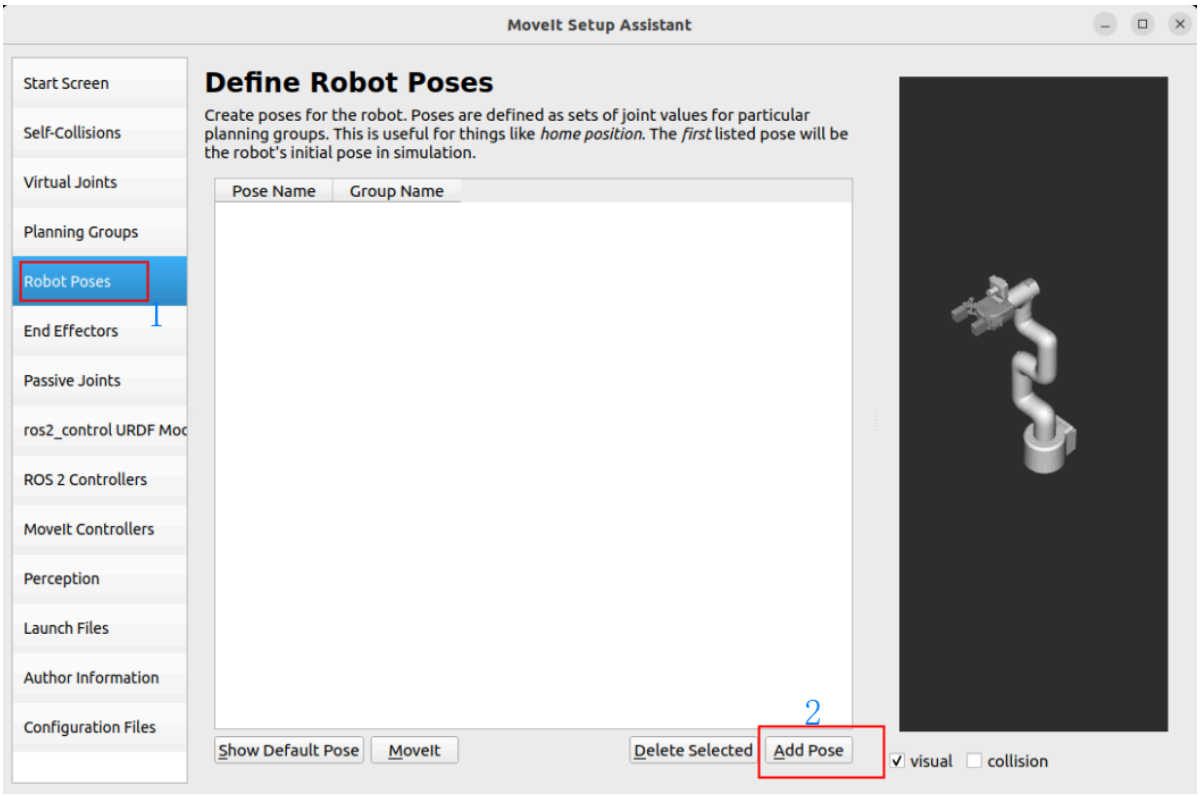




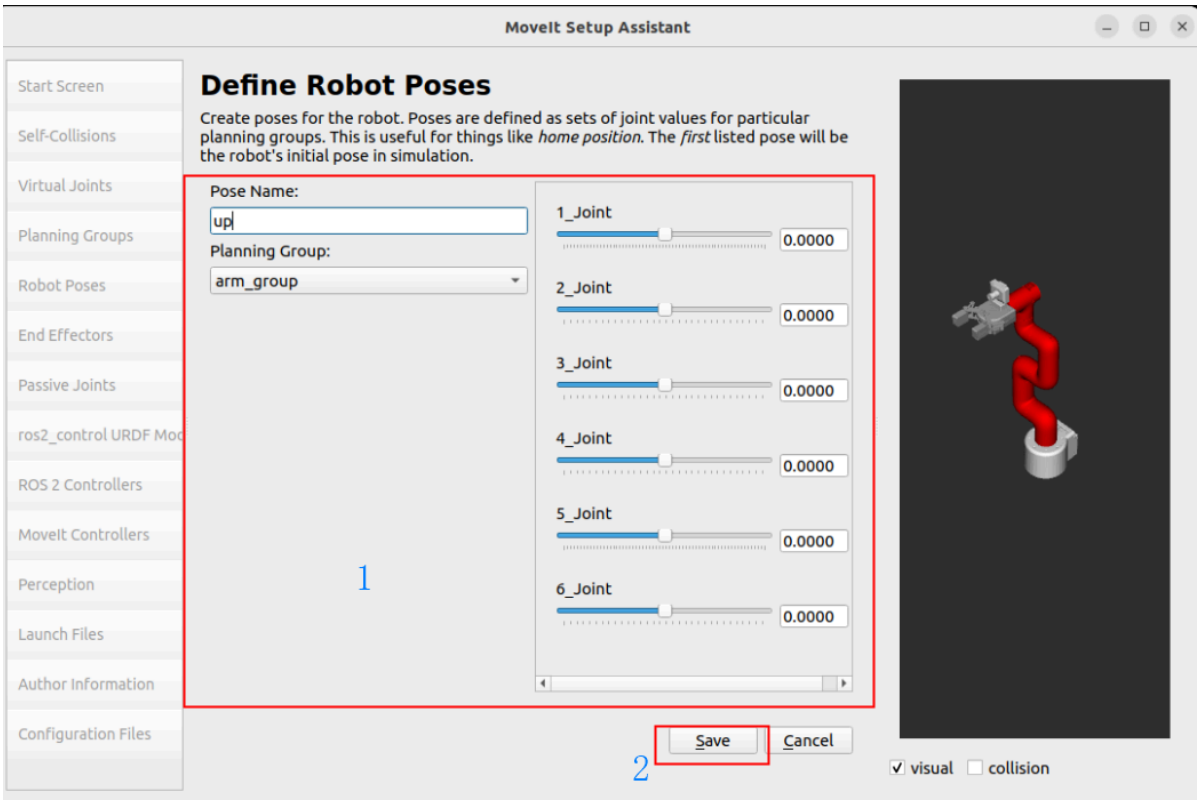


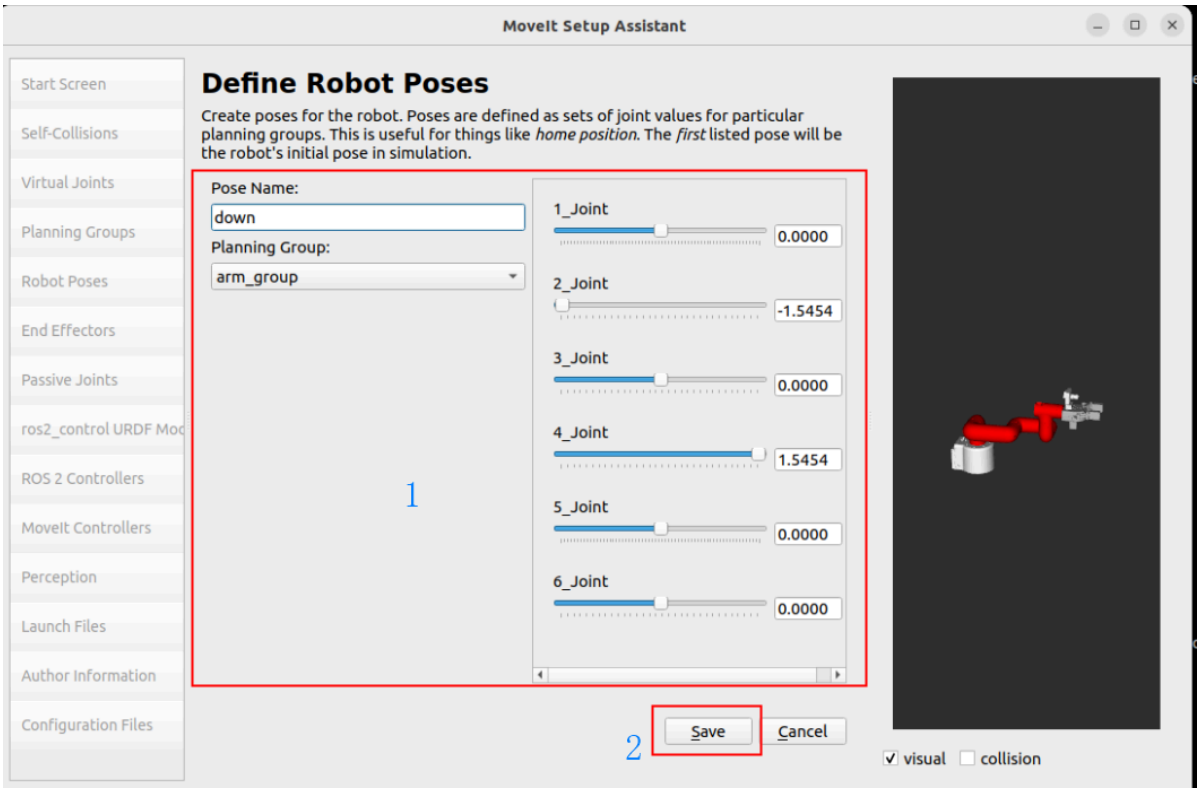
1.6. Robot position

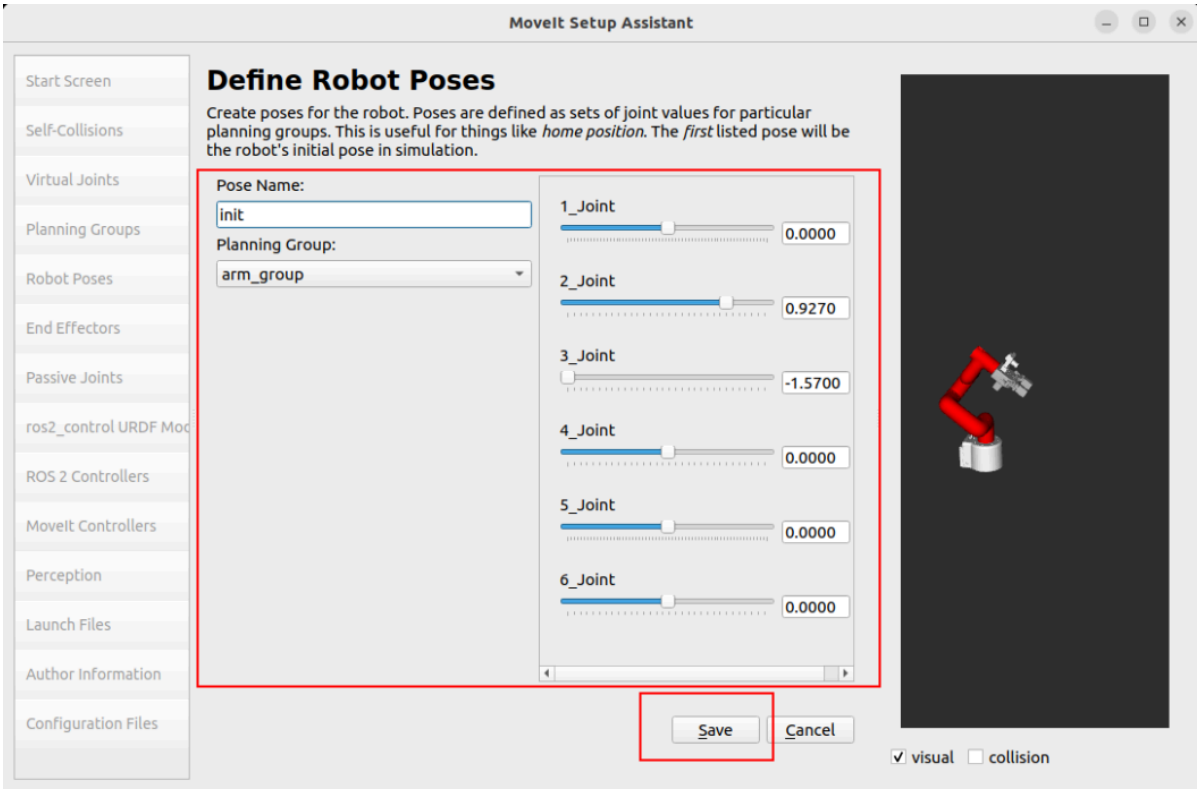
Set three sets of predefined robot arm positions to the robot configuration:

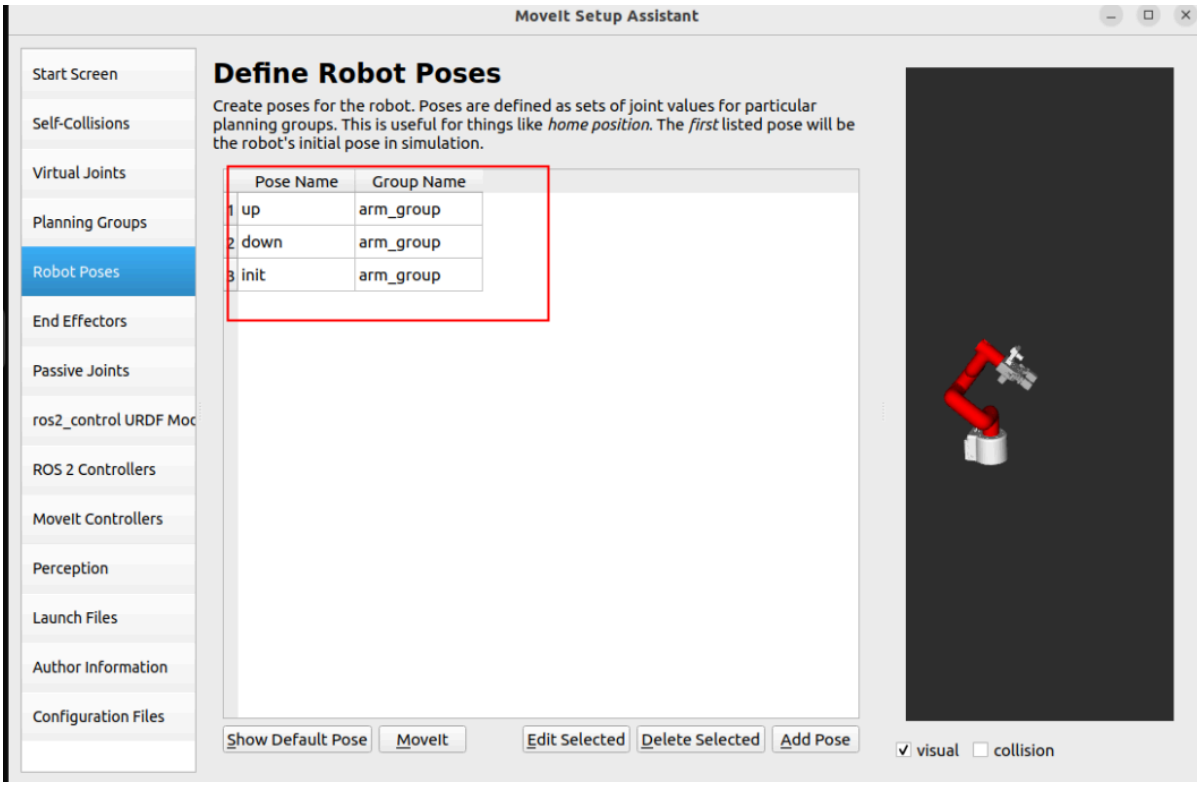


You can manually move the slider to set the state of each joint: up, down, init



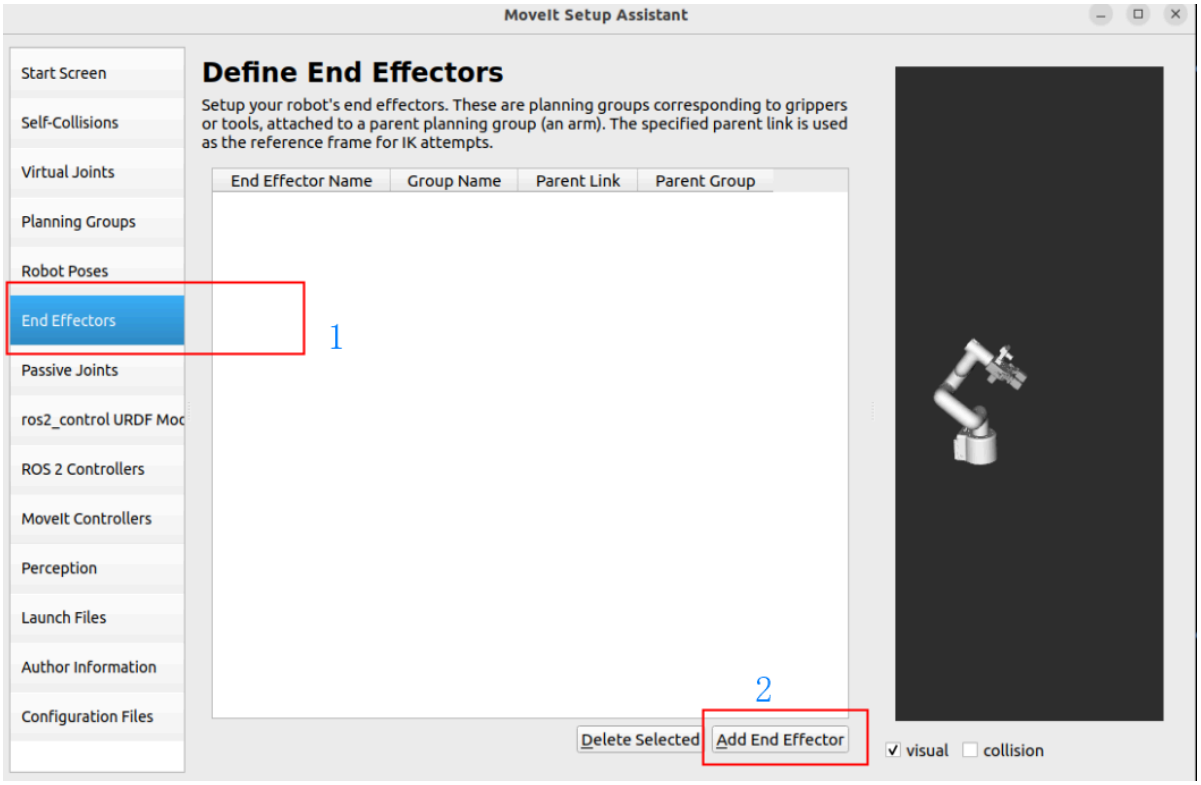

MoveIt Setup Assistant
Start Screen
Self-Collisions
Virtual Joints
Planning Groups
Robot Poses
End Effectors
Passive Joints
ros2_control URDF Mod
ROS 2 Controllers
MoveIt Controllers
Perception
Launch Files
Author Information
Configuration Files
Define Robot Poses
Create poses for the robot. Poses are defined as sets of joint values for particular planning groups. This is useful for things like home position. The first listed pose will be the robot's initial pose in simulation.
Pose Name:
down
Planning Group:
arm_group
1_Joint
0.0000
2_Joint
-1.5454
3_Joint
0.0000
4_Joint
1.5454
5_Joint
0.0000
6_Joint
0.0000
1
2
Save
Cancel
visual
collision


MoveIt Setup Assistant
Start Screen
Self-Collisions
Virtual Joints
Planning Groups
Robot Poses
End Effectors
Passive Joints
ros2_control URDF Mod
ROS 2 Controllers
MoveIt Controllers
Perception
Launch Files
Author Information
Configuration Files
Define Robot Poses
Create poses for the robot. Poses are defined as sets of joint values for particular planning groups. This is useful for things like home position. The first listed pose will be the robot's initial pose in simulation.
Pose Name:
init
Planning Group:
arm_group
1_Joint
0.0000
2_Joint
0.9270
3_Joint
-1.5700
4_Joint
0.0000
5_Joint
0.0000
6_Joint
0.0000
Save
Cancel
visual
collision



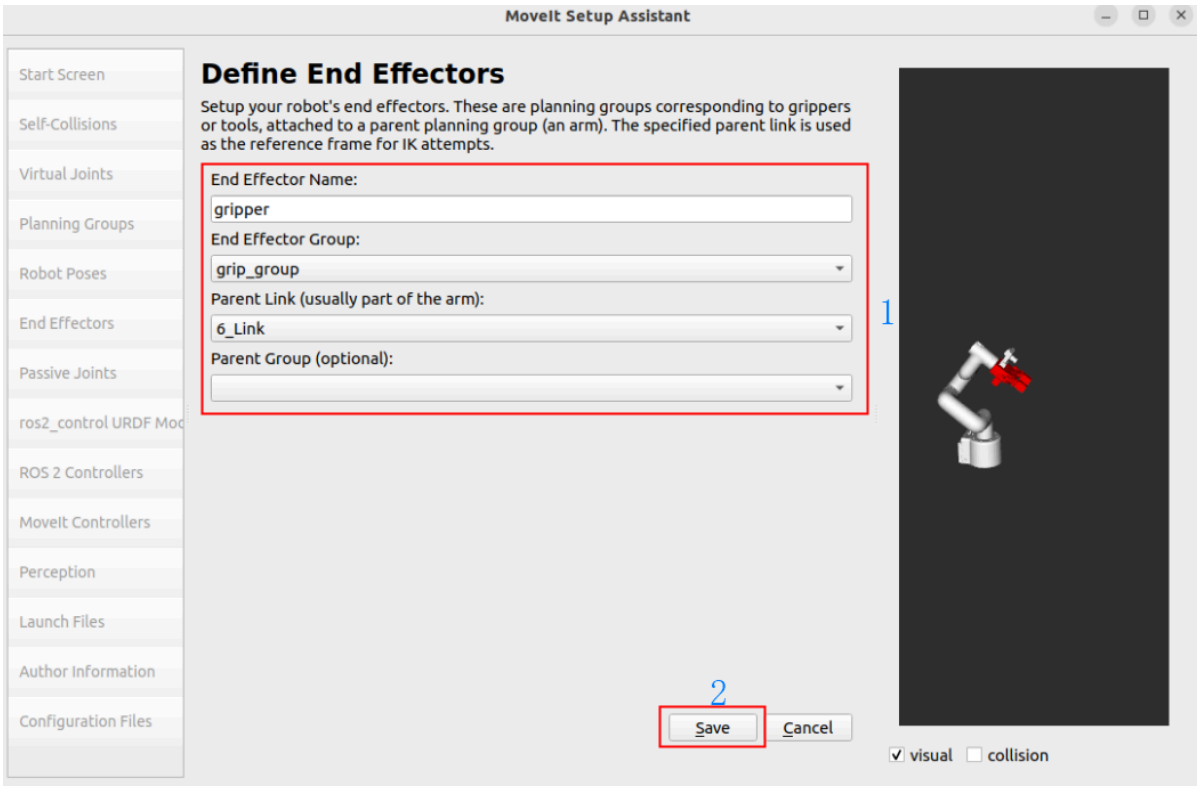
1.7. End Effector

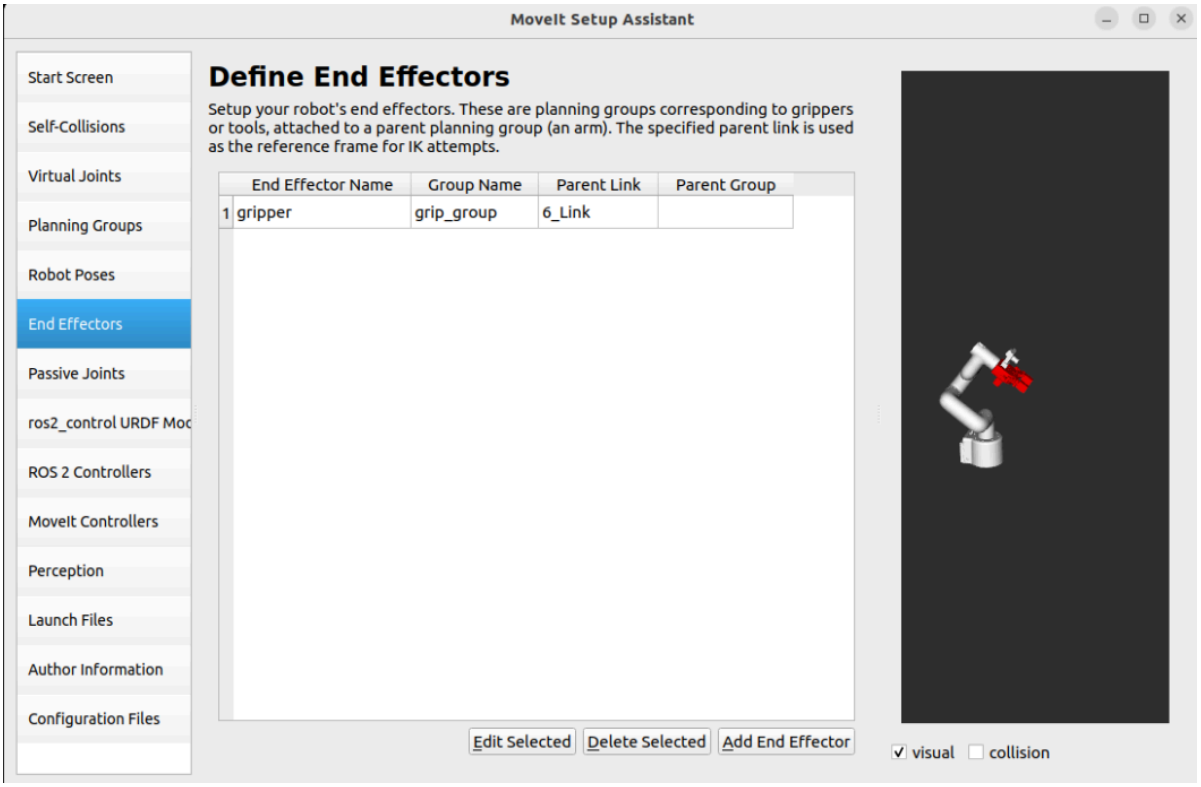
Set the gripper group as the end effector:



End Effector: gripper

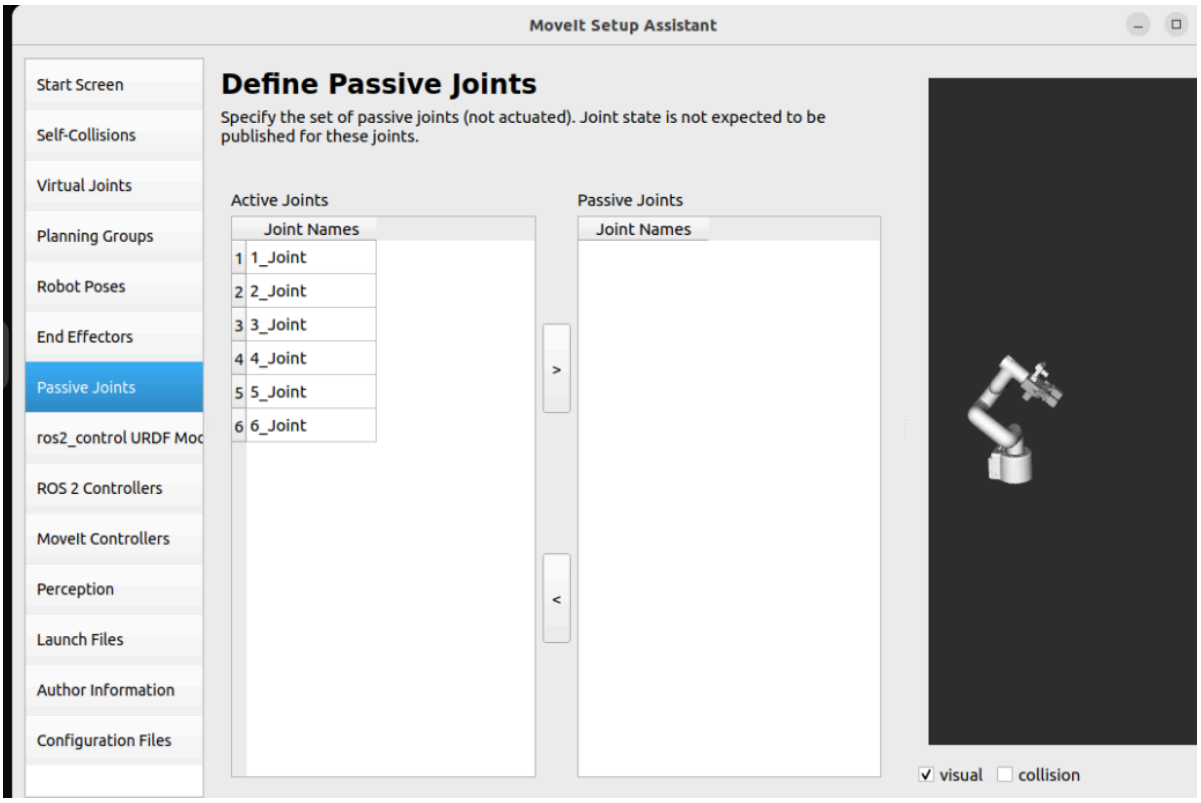
- End Effector Name: gripper
- End Effector Group: gripp_group
- Parent Link: Arm5_Link
- Parent Group: None





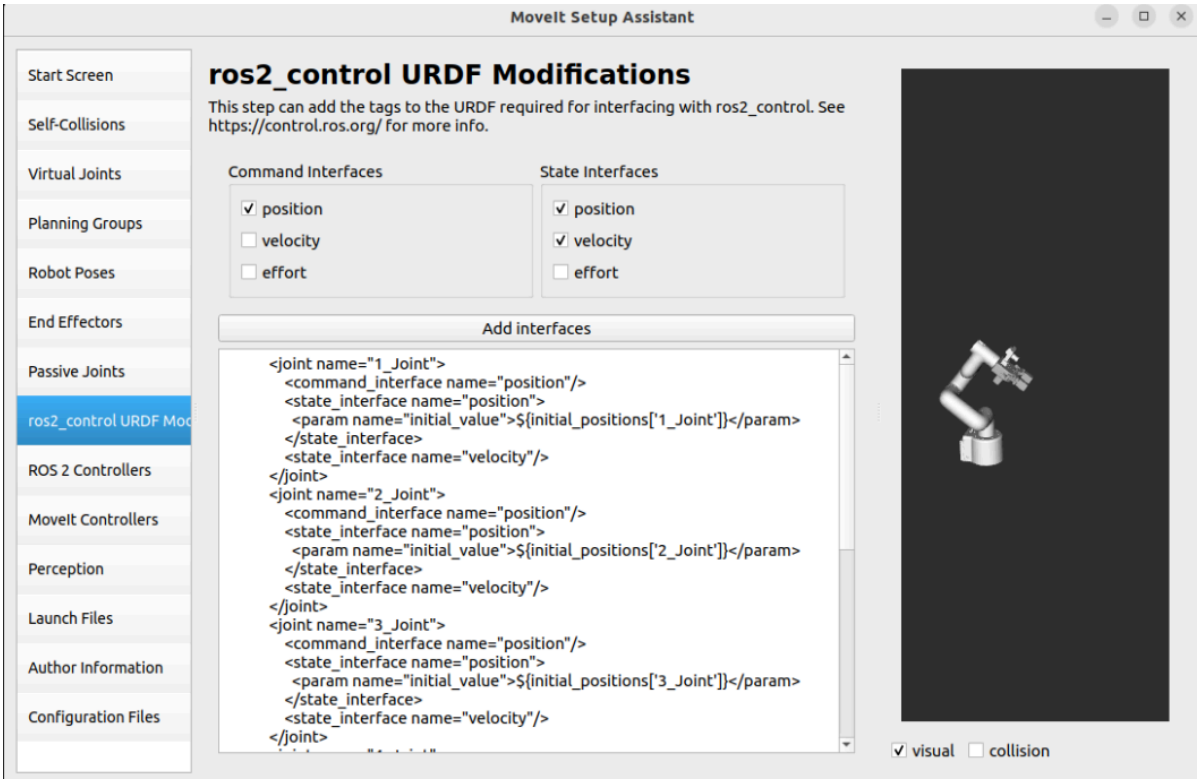
1.8. Passive Joints

Passive joints are non-actuated joints that cannot be directly controlled: if the robot arm does not have a passive joint, it will be skipped directly.



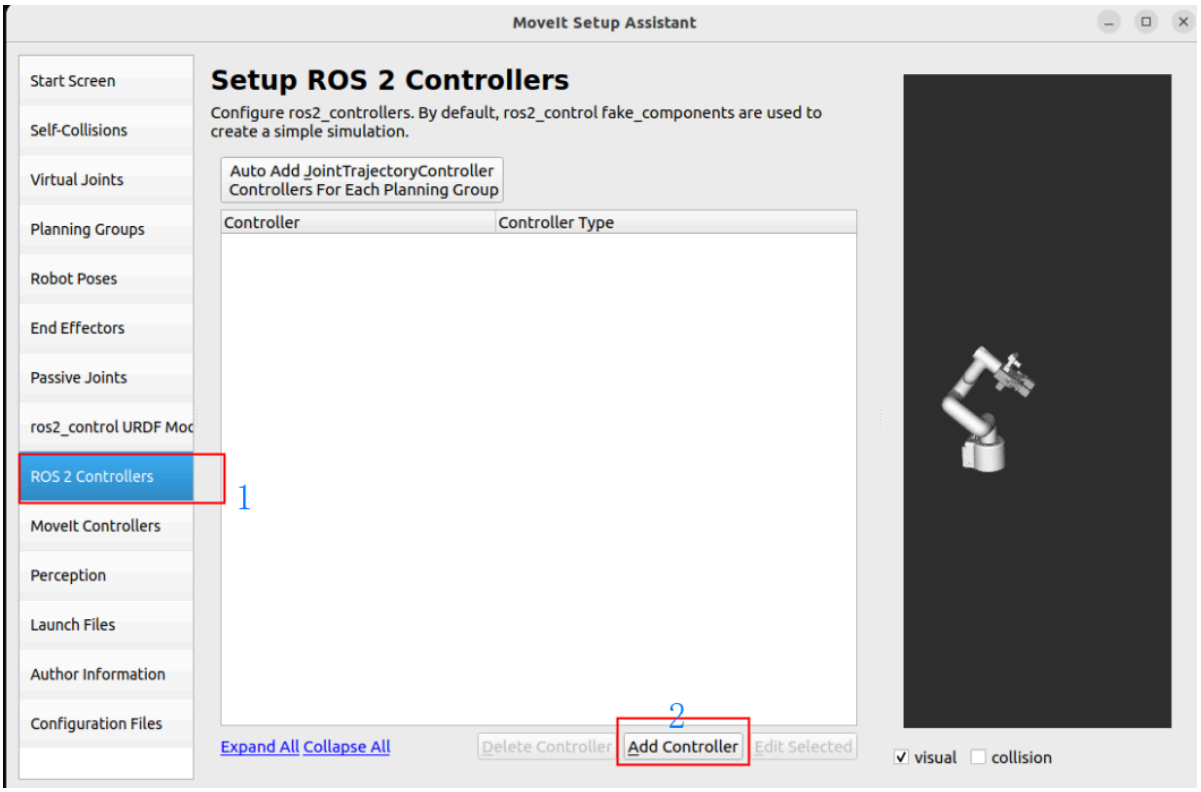
1.9, URDF file

The MoveIt Setup Assistant automatically sets up the command interface and state interface for each joint: use the default generated information.



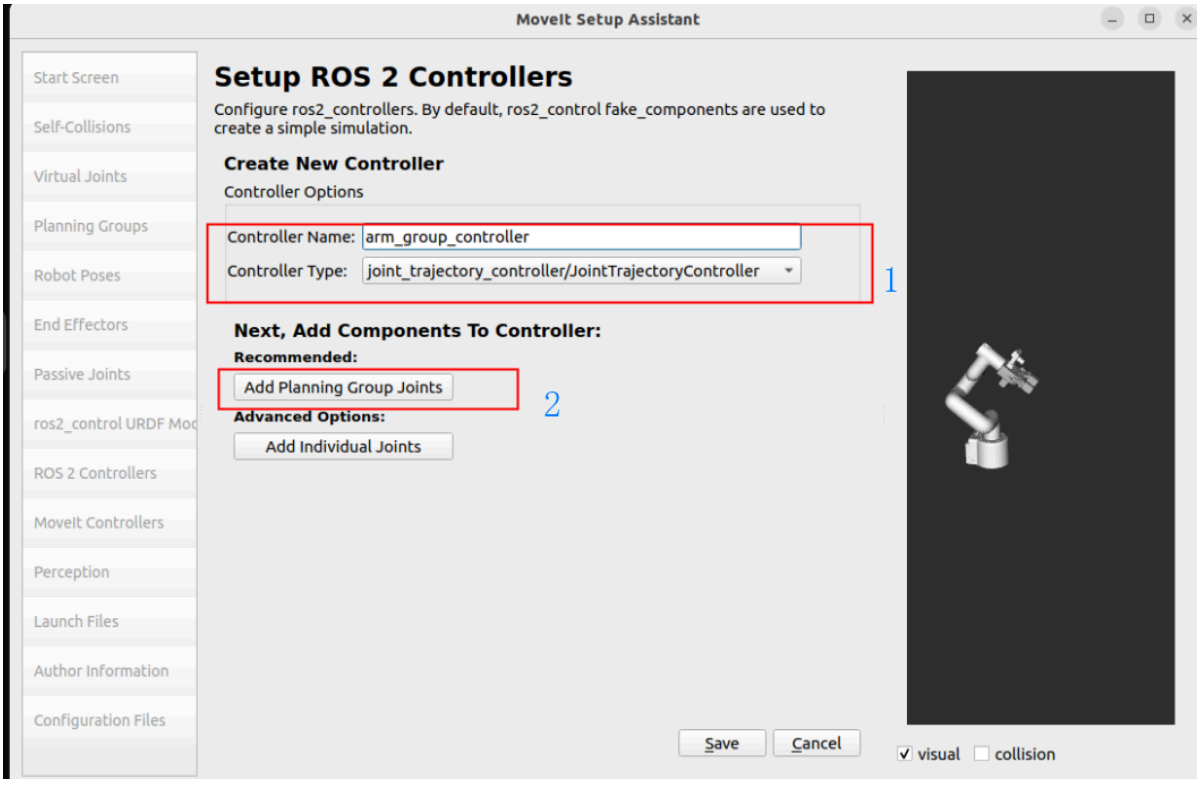
1.10, ROS2 Controllers

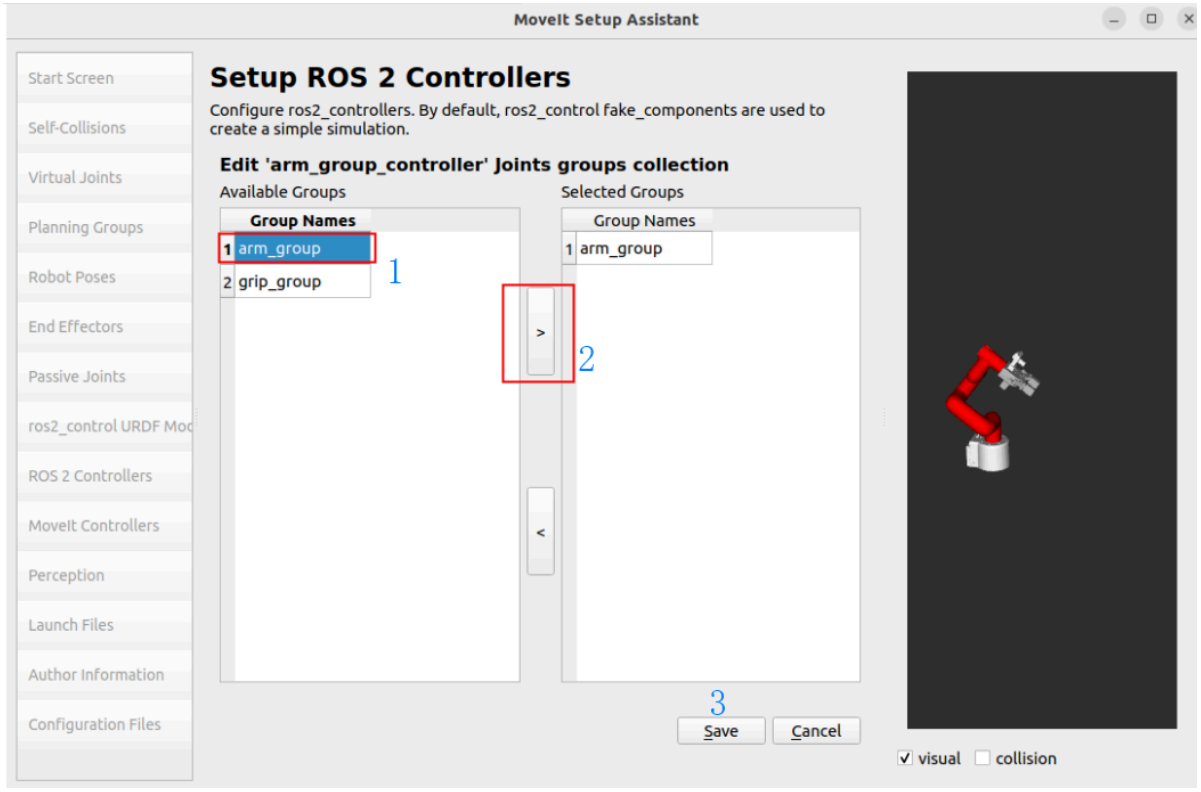
ROS2 Controllers is a framework for real-time robot control that can be used to automatically generate simulation controllers to drive robot joints.

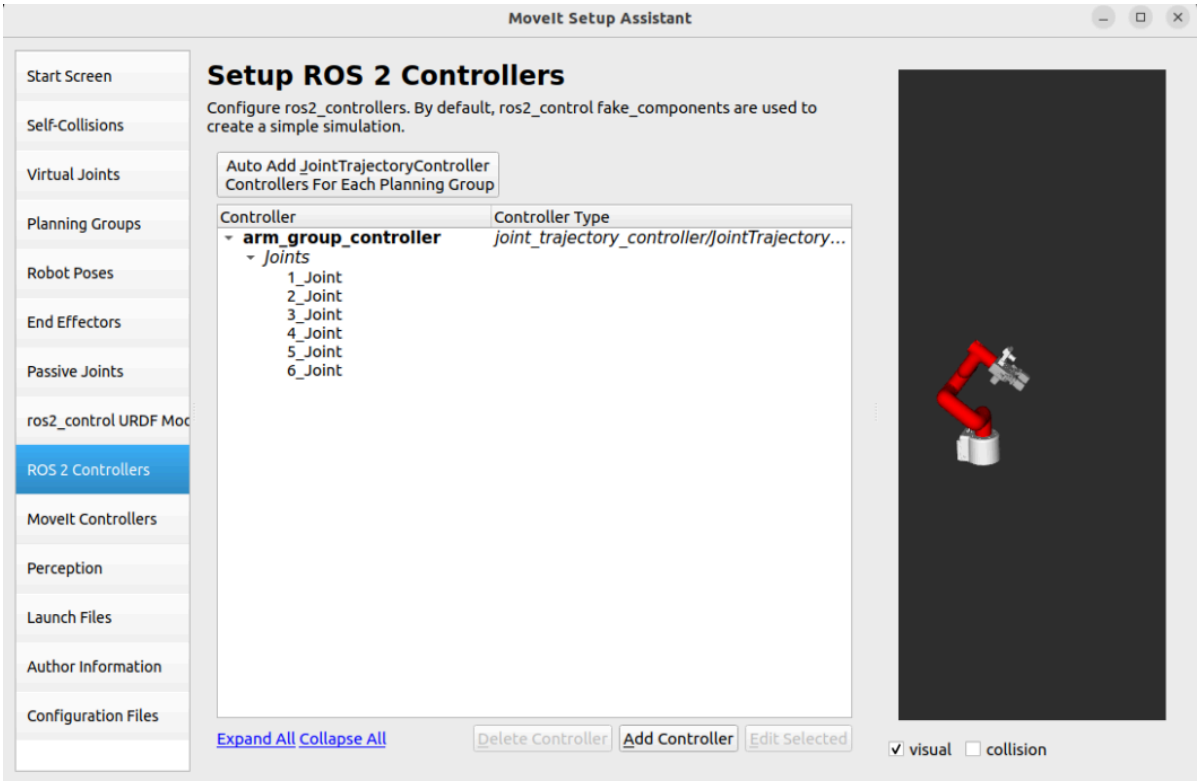


Add robot arm controller:

- Controller Name: arm_group_controller
- Controller Type: joint_trajectory_controller/JointTrajectoryController
- Add Planning Group Joints: arm_group



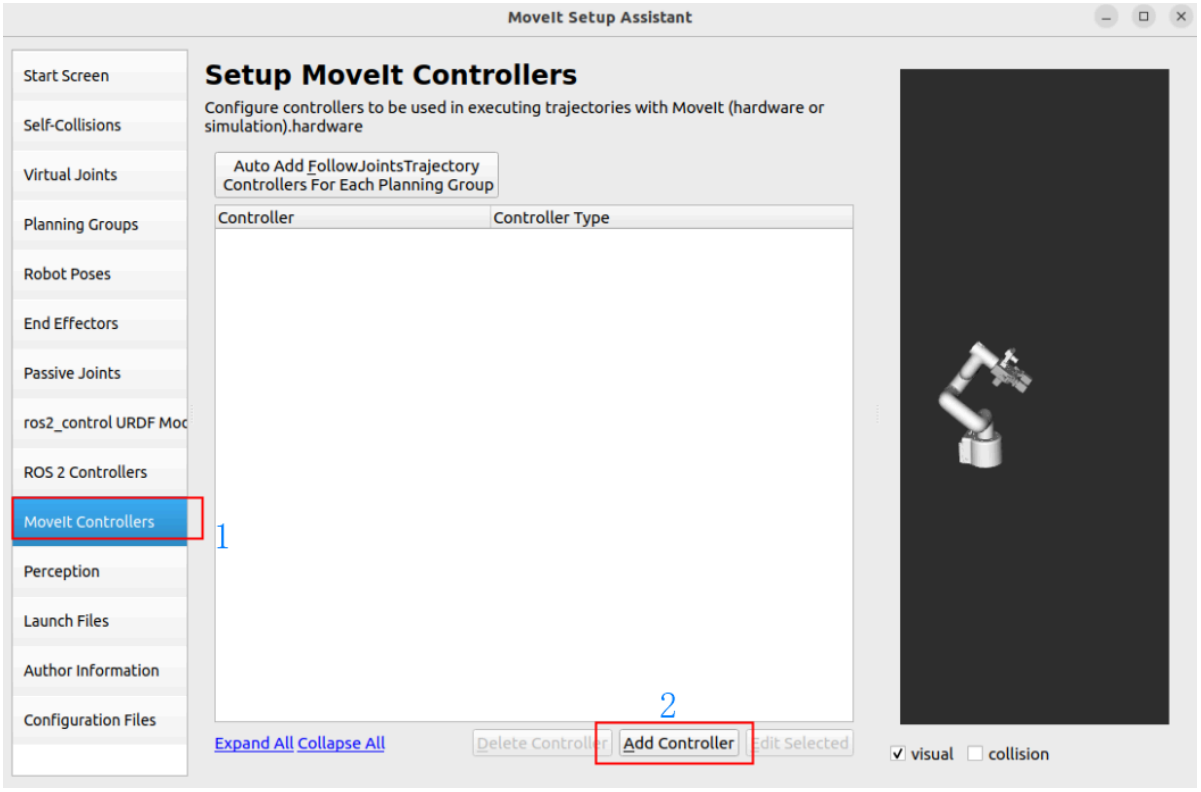




1.11, MoveIt Controllers

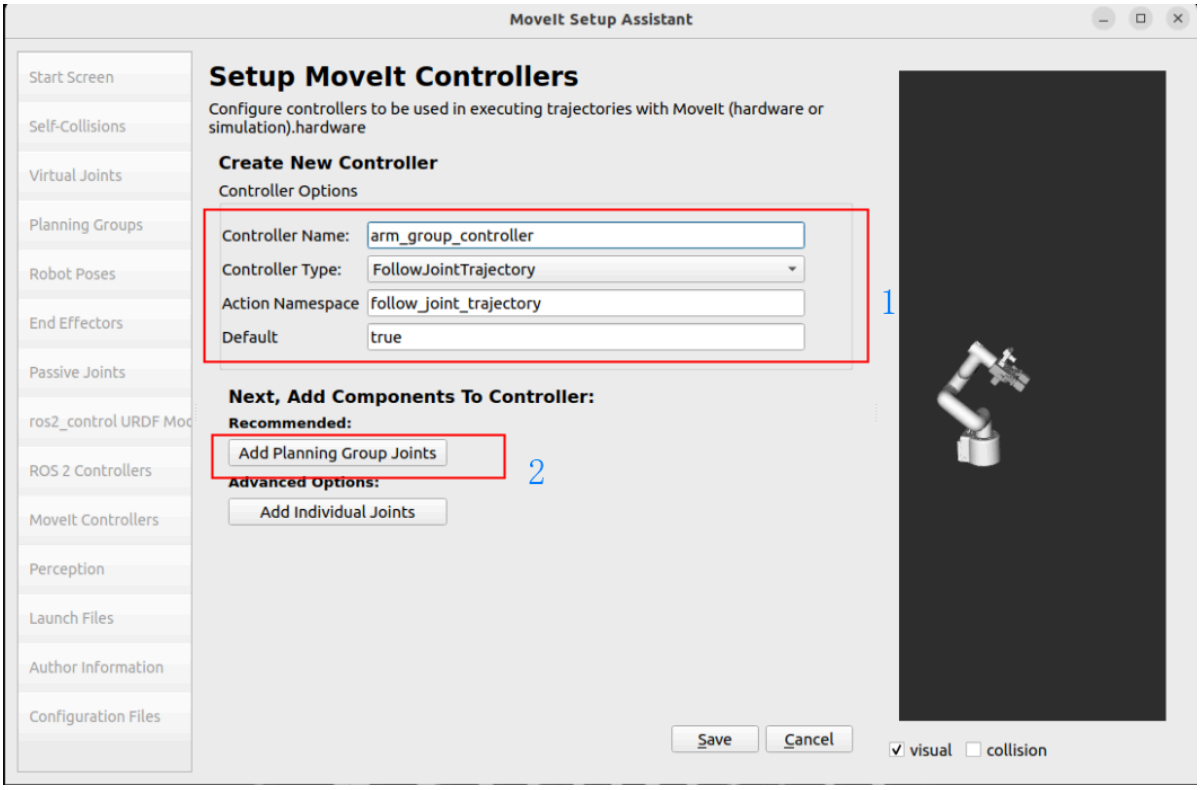
MoveIt requires a trajectory controller with the FollowJointTrajectoryAction interface to execute the planned trajectory, which sends the generated trajectory to the robot ROS2 controller.

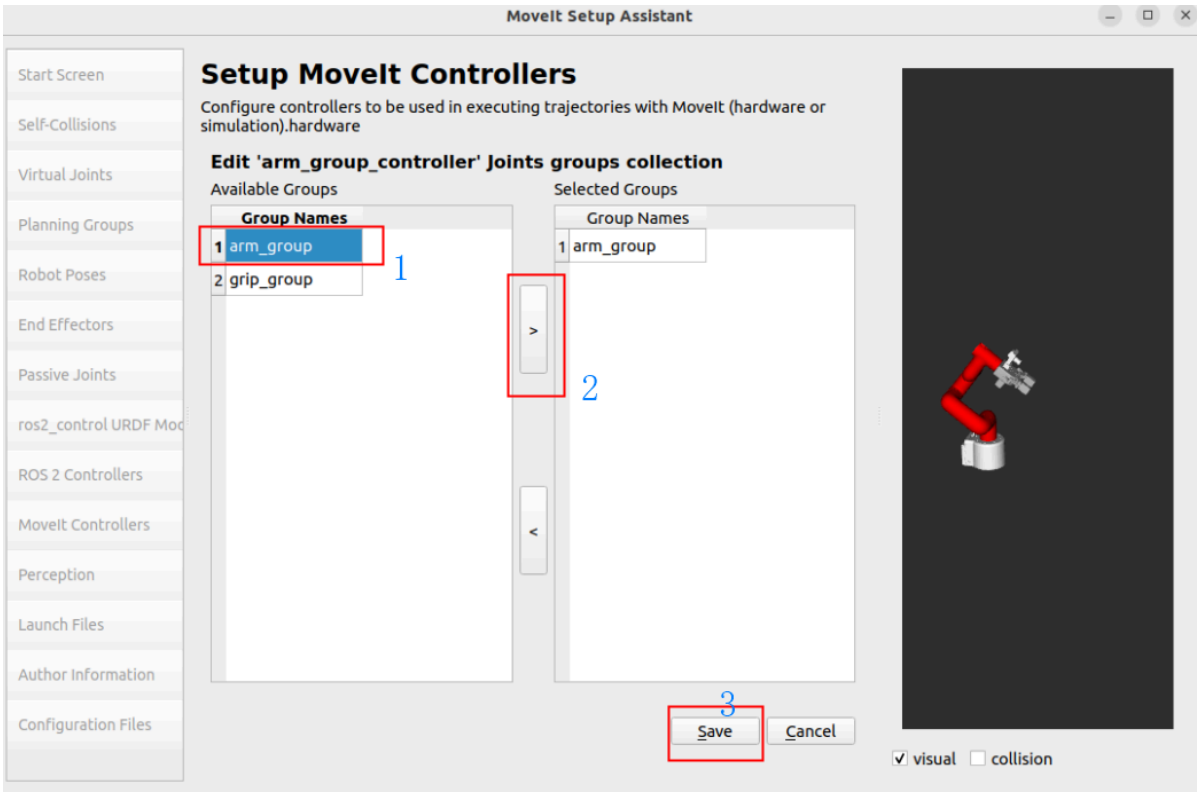
Added MoveIt Controllers need to ensure that the controller name matches the name configured in ROS2 Controllers.

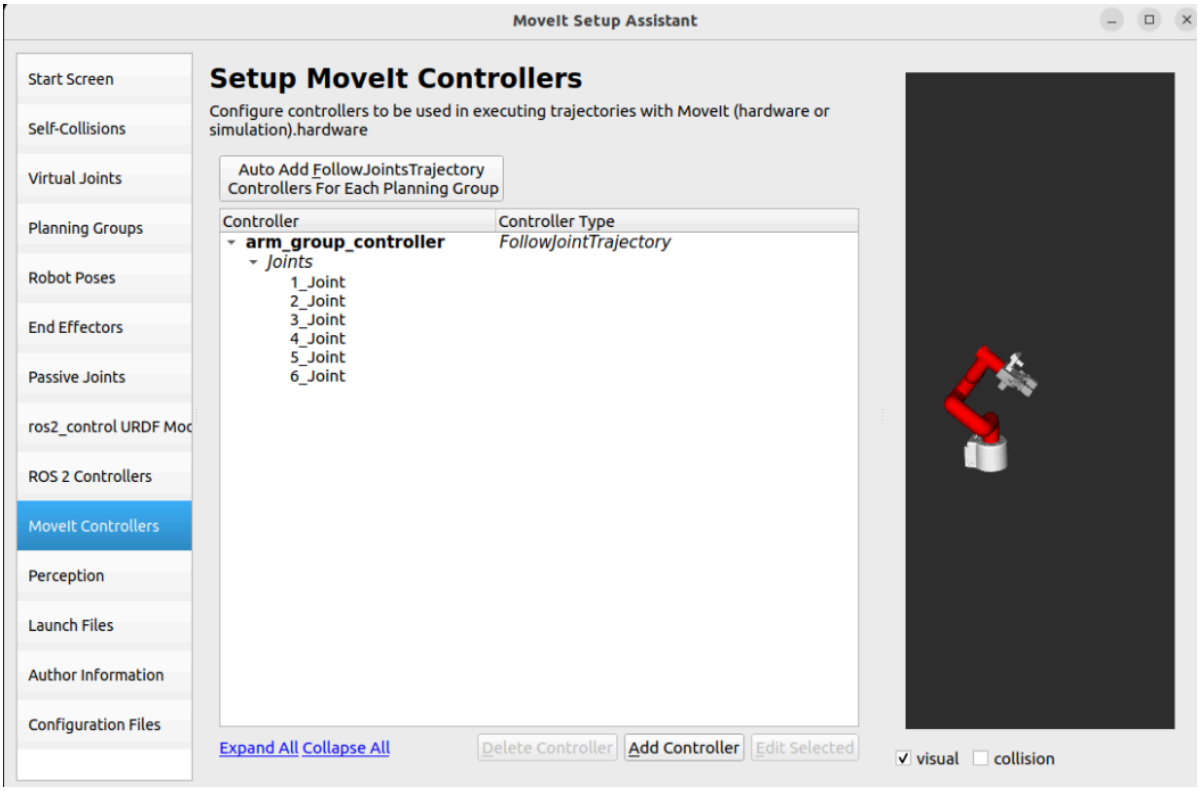


Add robot arm controller:

- Controller Name: arm_group_controller
- Controller Type: FollowJointTrajectory
- Add Planning Group Joints: arm_group

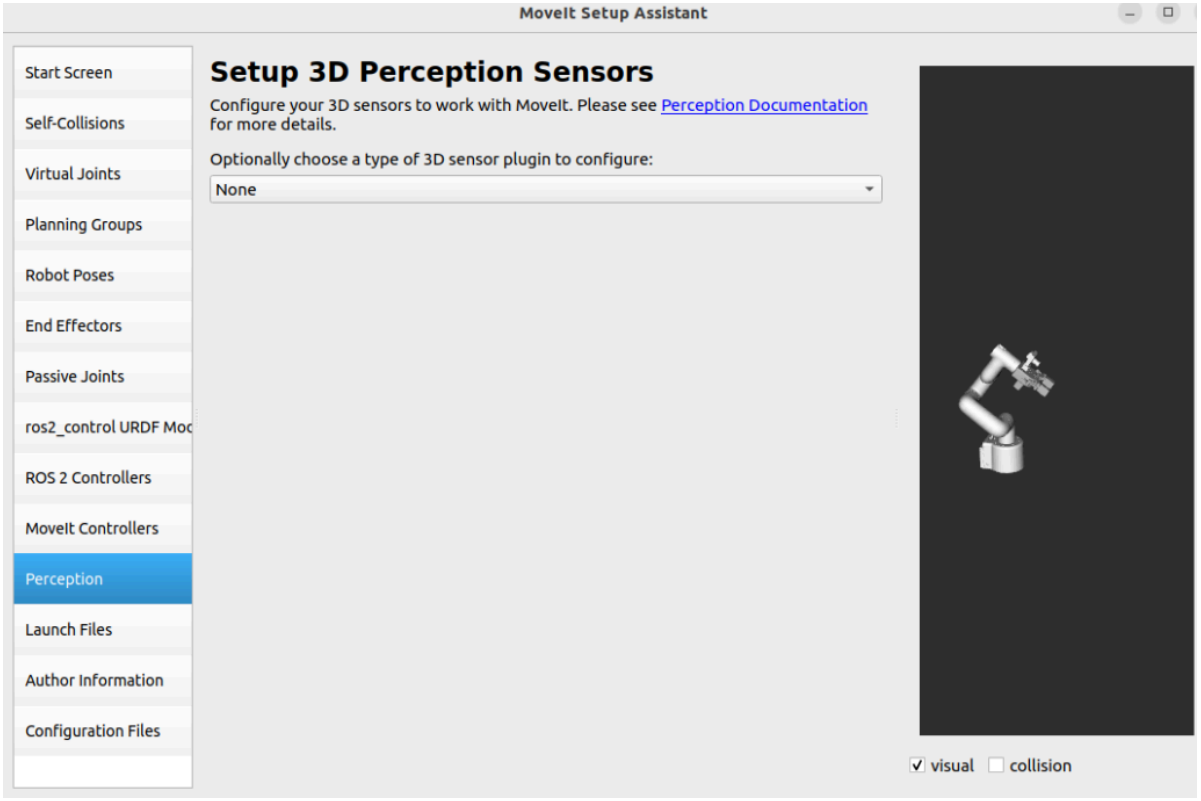






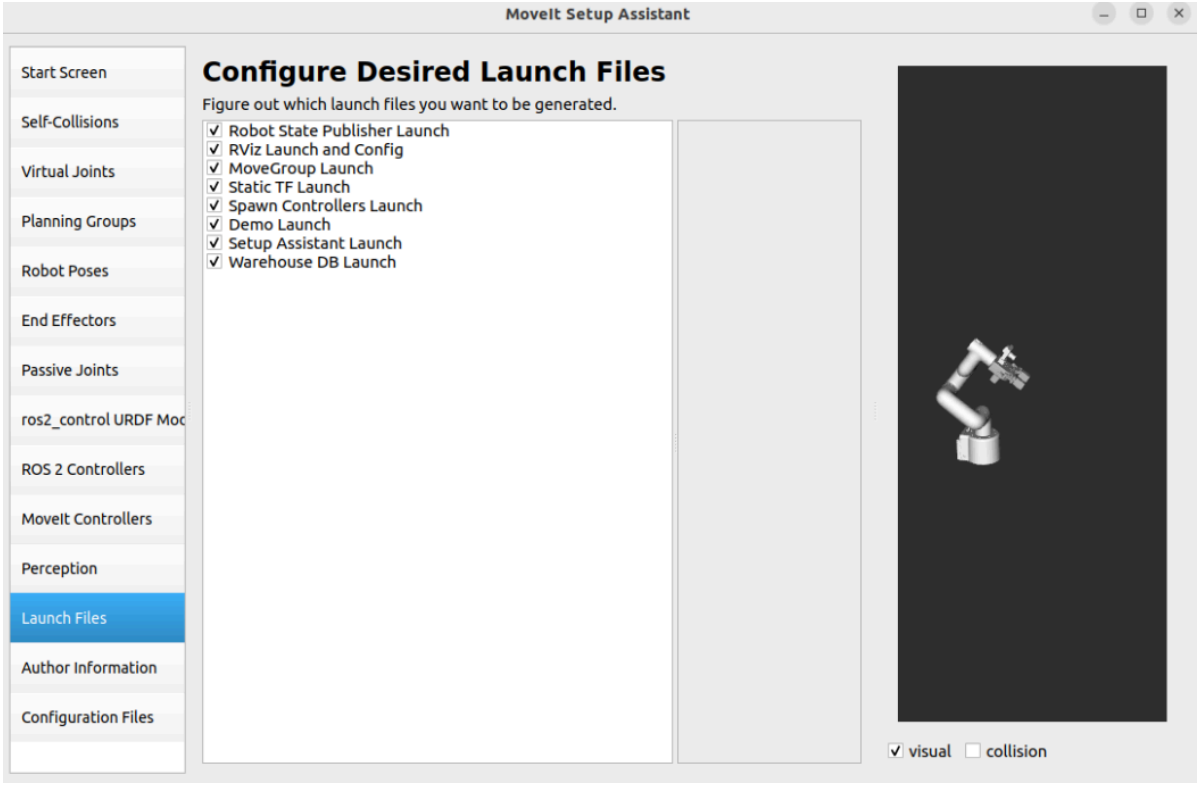
1.12. Sensors

Configure the settings of the 3D sensor used by the robot: No configuration is required.



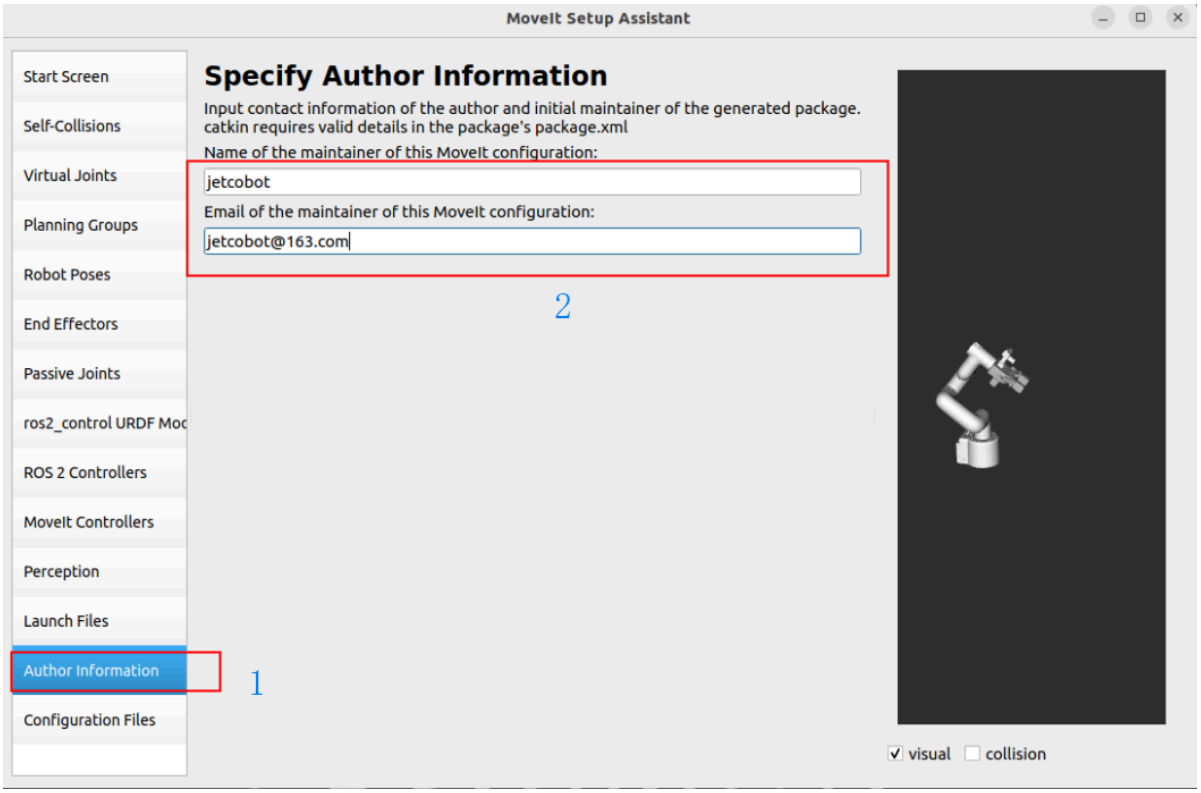
1.13. Startup File

Configure the automatically generated startup file: Use the default options.



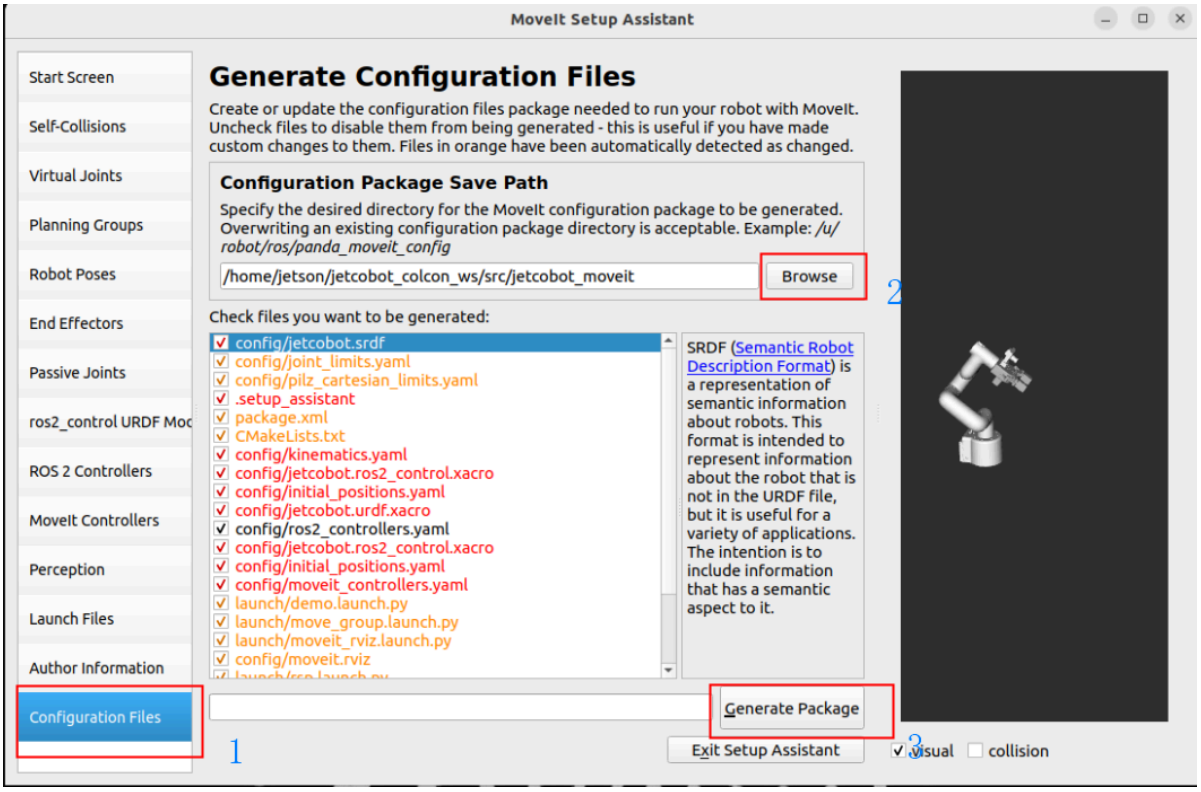
1.14. Author Information

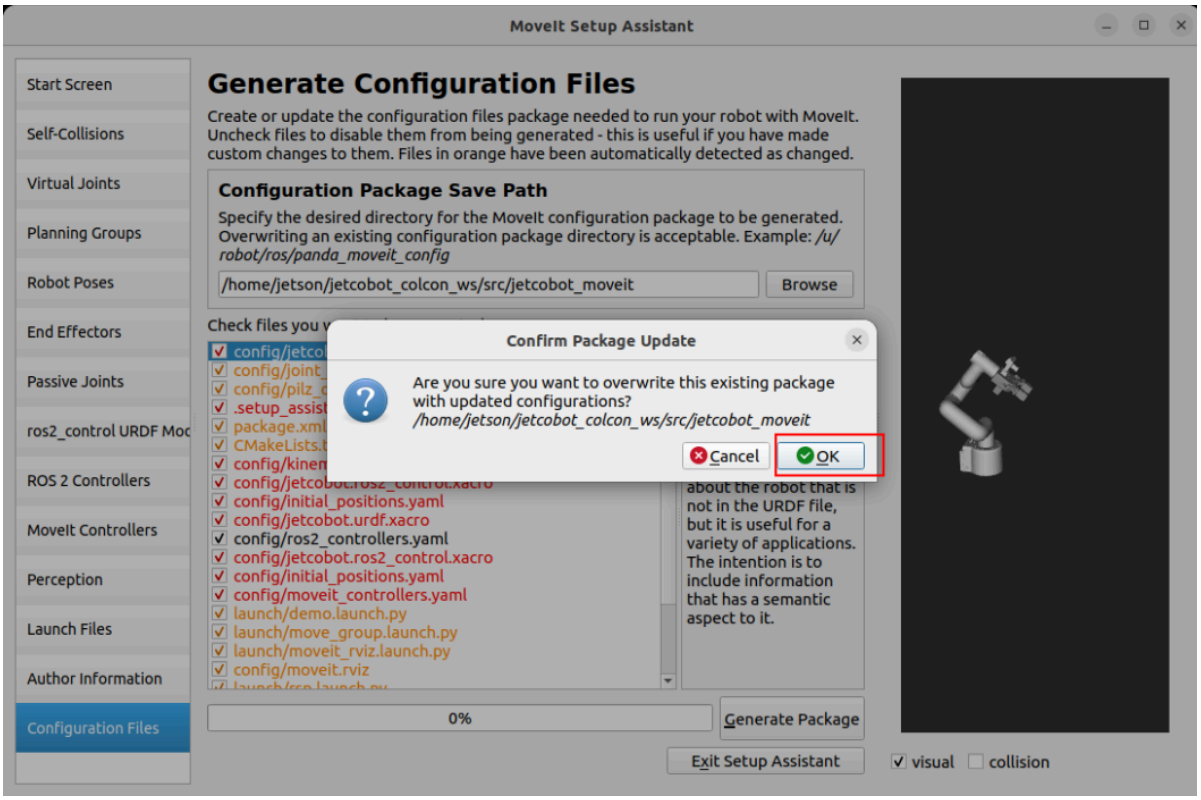
Add author information for package generation:

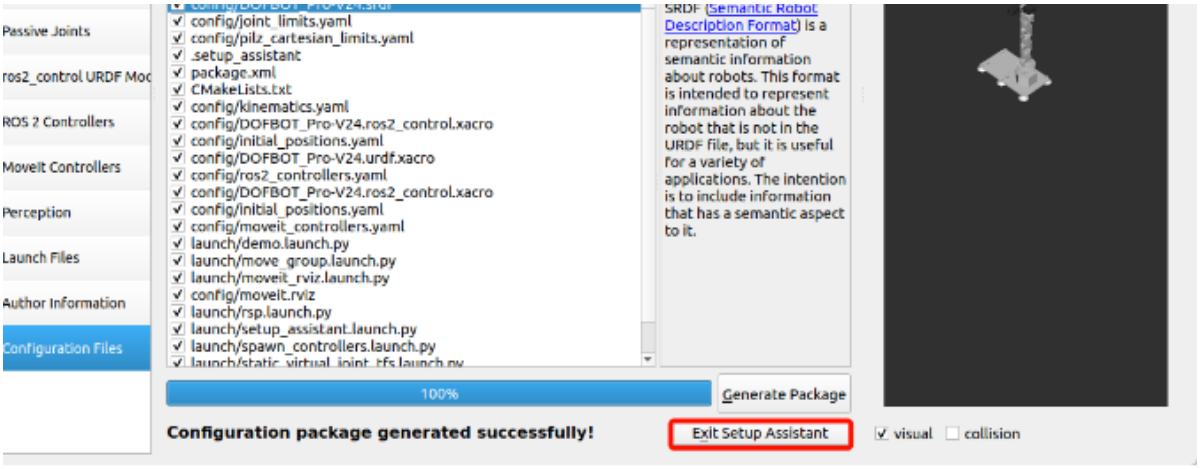


1.15. Generate configuration

Generate configuration files to the specified folder:

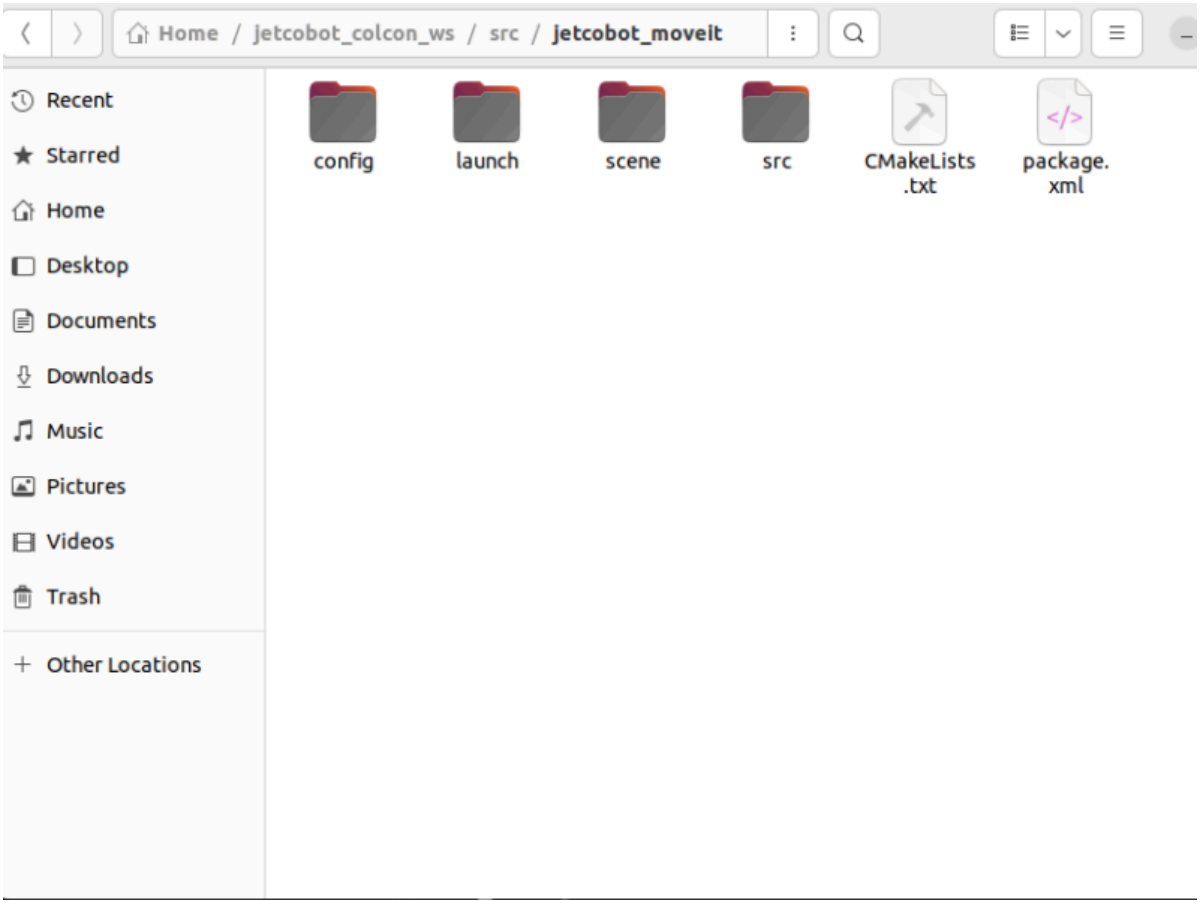






2. Configuration files

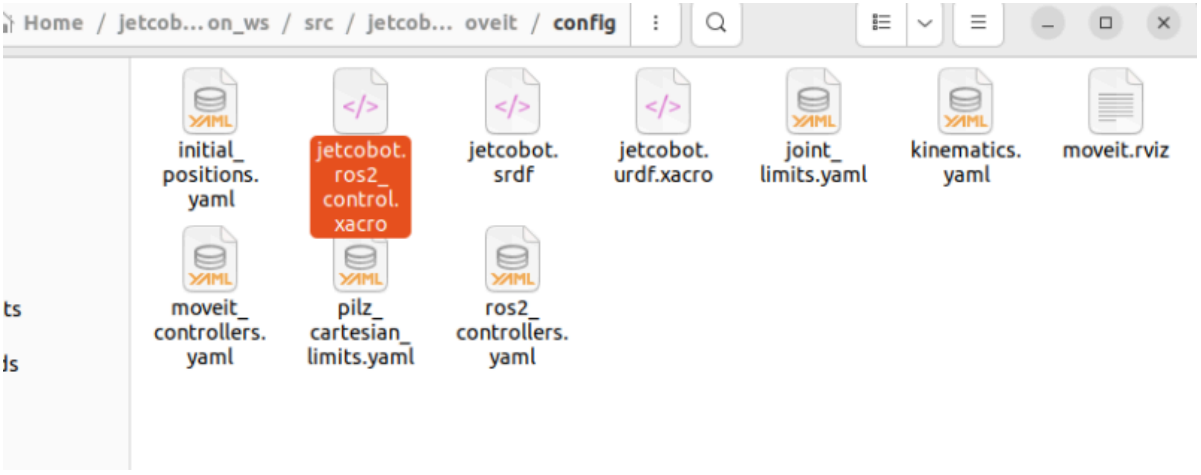
Enter the `dofbot_pro_moveit` folder:



3. Configuration verification

MoveIt Setup The configuration file generated by Assistant needs to be modified to remove the warnings at startup and the problem of model not loading.

3.1, modify the configuration file



Modify the `jetcobot.ros2_control.xacro` file: modify the function of loading yaml

```
load_yaml(initial_positions_file)['initial_positions']
Change to
xacro.load_yaml(initial_positions_file)['initial_positions']
```

jetcobot.ros2_control.xacro
~/jetcobot_colcon_ws/src/jetcobot_moveit/config

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro">
    <xacro:macro name="jetcobot_ros2_control" params="name initial_positions_file">
        <xacro:property name="initial_positions" value="${xacro.load_yaml(initial_positions_file)['initial_positions']}"/>

        <ros2_control name="${name}" type="system">
            <hardware>
                <!-- By default, set up controllers for simulation. This won't work on real hardware -->
                <plugin>mock_components/GenericSystem</plugin>
            </hardware>
            <joint name="1_Joint">
                <command_interface name="position"/>
                <state_interface name="position">
                  <param name="initial_value">${initial_positions['1_Joint']}</param>
                </state_interface>
                <state_interface name="velocity"/>
            </joint>
            <joint name="2_Joint">
                <command_interface name="position"/>
                <state_interface name="position">
                  <param name="initial_value">${initial_positions['2_Joint']}</param>
                </state_interface>
                <state_interface name="velocity"/>
            </joint>
            <joint name="3_Joint">
                <command_interface name="position"/>
                <state_interface name="position">
                  <param name="initial_value">${initial_positions['3_Joint']}</param>
                </state_interface>
                <state_interface name="velocity"/>
            </joint>
            <joint name="4_Joint">
                <command_interface name="position"/>
                <state_interface name="position">
                  <param name="initial_value">${initial_positions['4_Joint']}</param>
                </state_interface>
                <state_interface name="velocity"/>
            </joint>
            <joint name="5_Joint">
                <command_interface name="position"/>
                <state_interface name="position">
                  <param name="initial_value">${initial_positions['5_Joint']}</param>
                </state_interface>
```

3.2, compile the software package

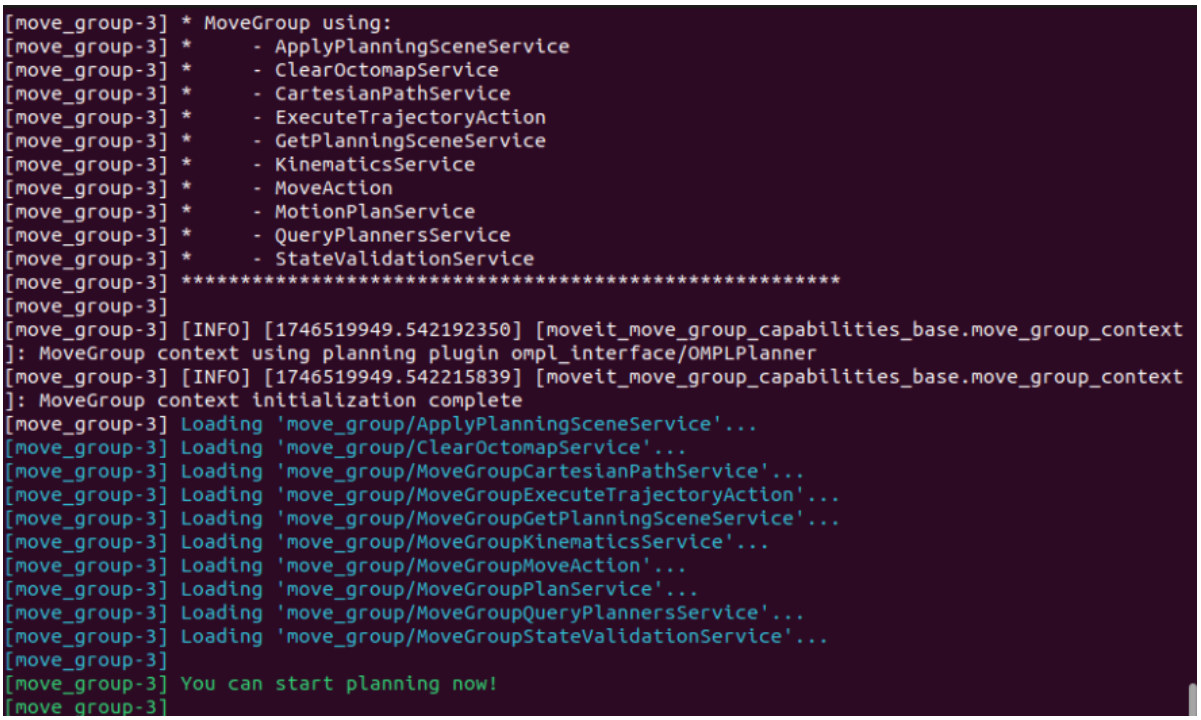
```
cd ~/jetcobot_colcon_ws/
colcon build
source install/setup.bash
```

3.3、Start MoveIt

```
ros2 launch jetcobot_moveit demo.launch.py
```

Starting the simulation is slow. When the terminal displays `You can start planning now!` or the robot arm displays a trackball (the trackball is a newly added sphere on the robot arm), the loading is complete.

```
jetson@yahboom:~/jetcobot_colcon_ws$ ros2 launch jetcobot_moveit demo.launch.py
[INFO] [launch]: All log files can be found below /home/jetson/.ros/log/2025-05-06-16-25-46-805207-y
ahboom-15529
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [static_transform_publisher-1]: process started with pid [15530]
[INFO] [robot_state_publisher-2]: process started with pid [15532]
[INFO] [move_group-3]: process started with pid [15534]
[INFO] [rviz2-4]: process started with pid [15536]
[INFO] [ros2_control_node-5]: process started with pid [15538]
[INFO] [spawner-6]: process started with pid [15540]
[INFO] [spawner-7]: process started with pid [15542]
[static_transform_publisher-1] [INFO] [1746519947.927030757] [static_transform_publisher0]: Spinning
 until stopped - publishing transform
[static_transform_publisher-1] translation: ('0.000000', '0.000000', '0.000000')
[static_transform_publisher-1] rotation: ('0.000000', '0.000000', '0.000000', '1.000000')
[static_transform_publisher-1] from 'world' to 'base_link'
[robot_state_publisher-2] [INFO] [1746519947.927861727] [robot_state_publisher]: got segment 1_Link
[robot_state_publisher-2] [INFO] [1746519947.928102023] [robot_state_publisher]: got segment 2_Link
[robot_state_publisher-2] [INFO] [1746519947.928125320] [robot_state_publisher]: got segment 3_Link
[robot_state_publisher-2] [INFO] [1746519947.928134664] [robot_state_publisher]: got segment 4_Link
[robot_state_publisher-2] [INFO] [1746519947.928142536] [robot_state_publisher]: got segment 5_Link
[robot_state_publisher-2] [INFO] [1746519947.928149192] [robot_state_publisher]: got segment 6_Link
[robot_state_publisher-2] [INFO] [1746519947.928155401] [robot_state_publisher]: got segment base_li
nk
[robot_state_publisher-2] [INFO] [1746519947.928161929] [robot_state_publisher]: got segment camera_
link
```



References

MoveIt2 Humble: https://moveit.picknik.ai/humble/index.html

MoveIt2: https://moveit.picknik.ai/main/index.html