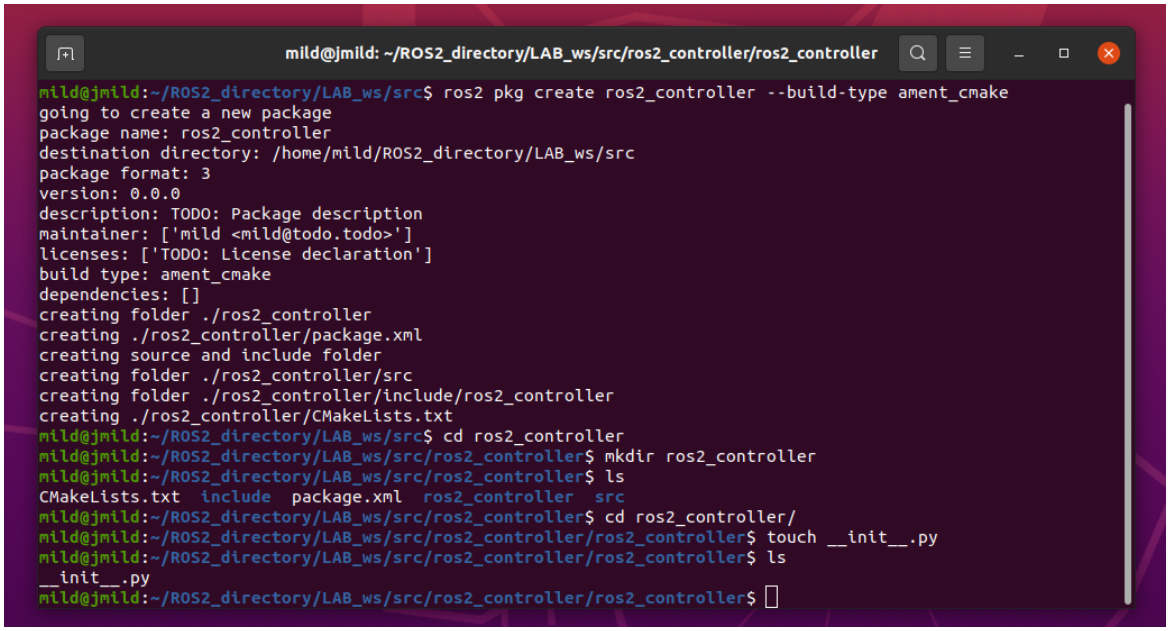
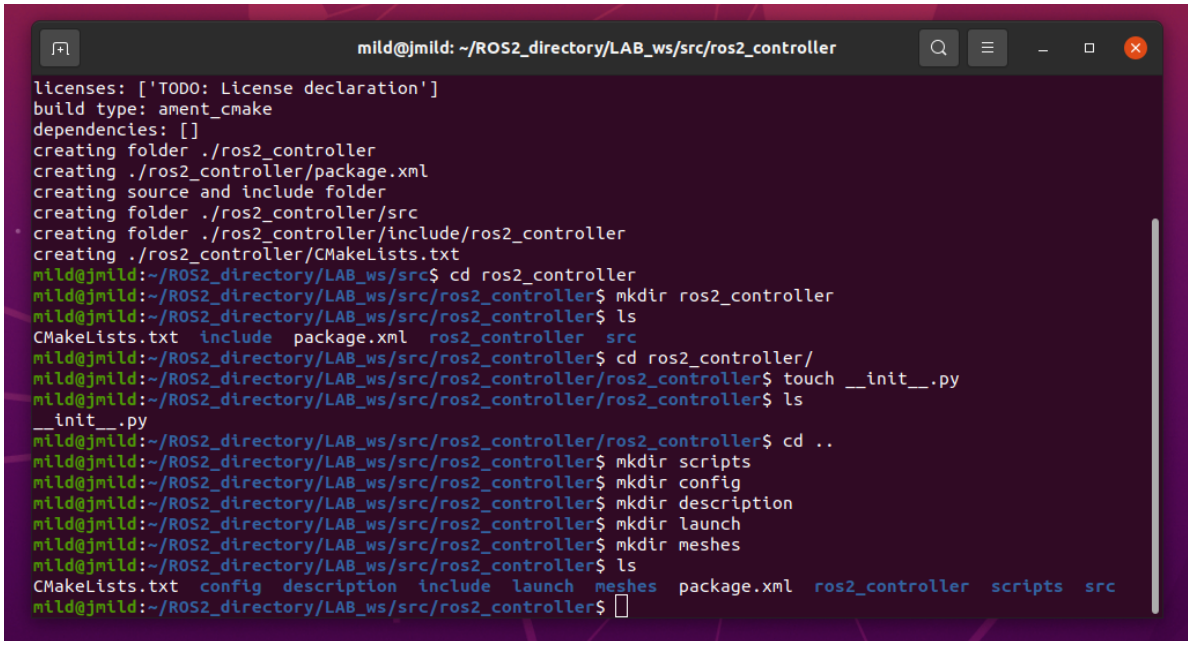
Lab3 Report_08

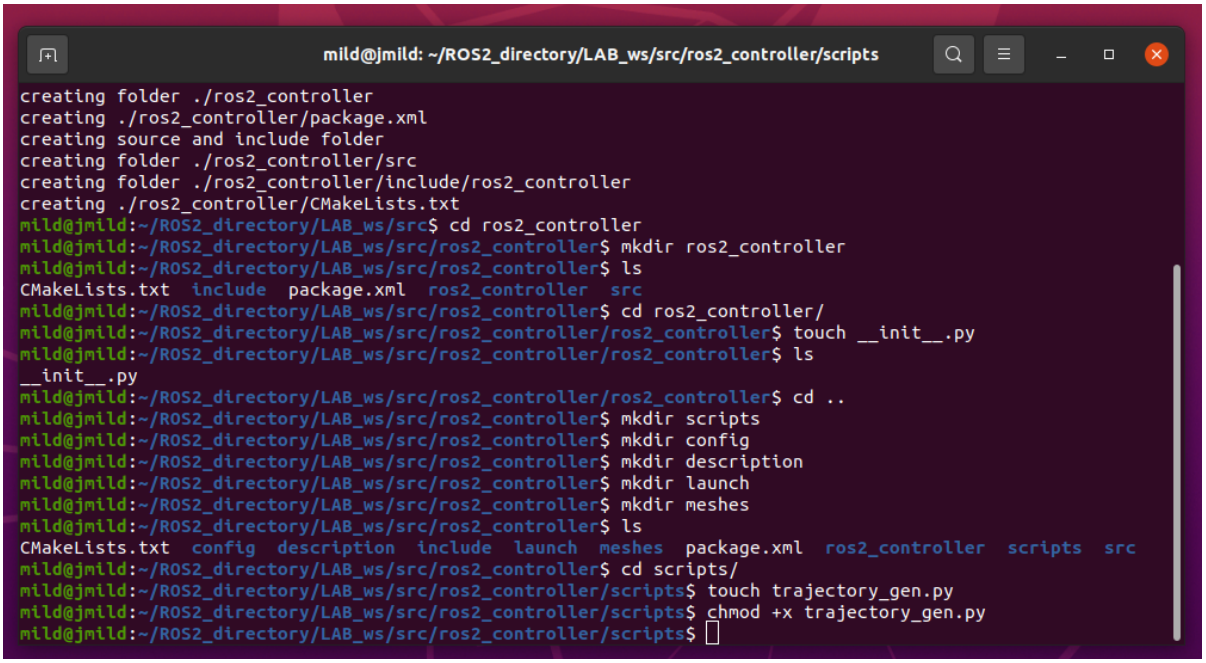
Create package: ros2_controller
Create folder: ros2_controller
Create file: __itit__.py (in folder: ros2_controller)



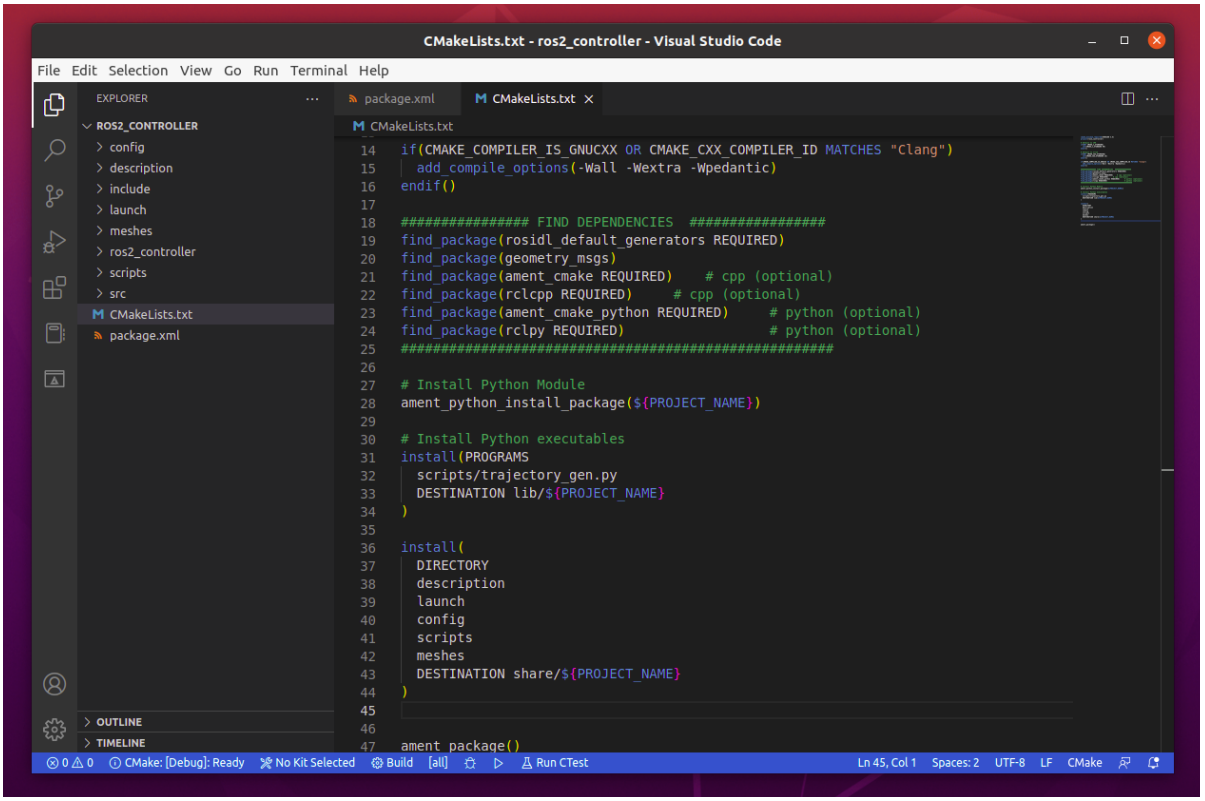
Create folder: scripts
Create folder: config
Create folder: description
Create folder: launch
Create folder: meshes



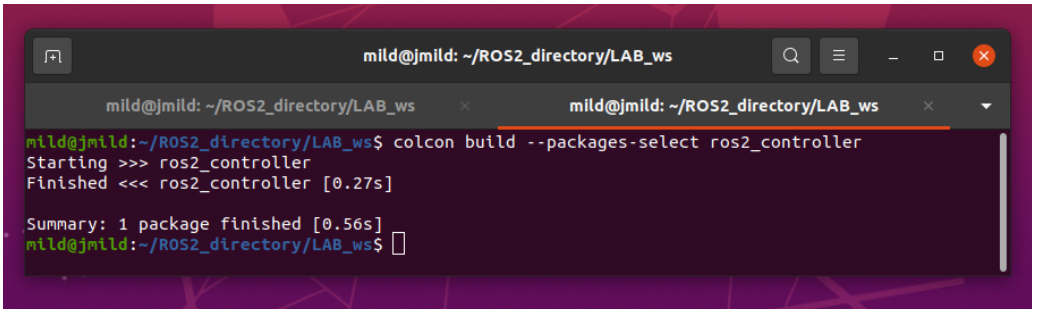
Create file: trajectory_gen.py (in folder: scripts) and adds the execute permission (chmod +x)

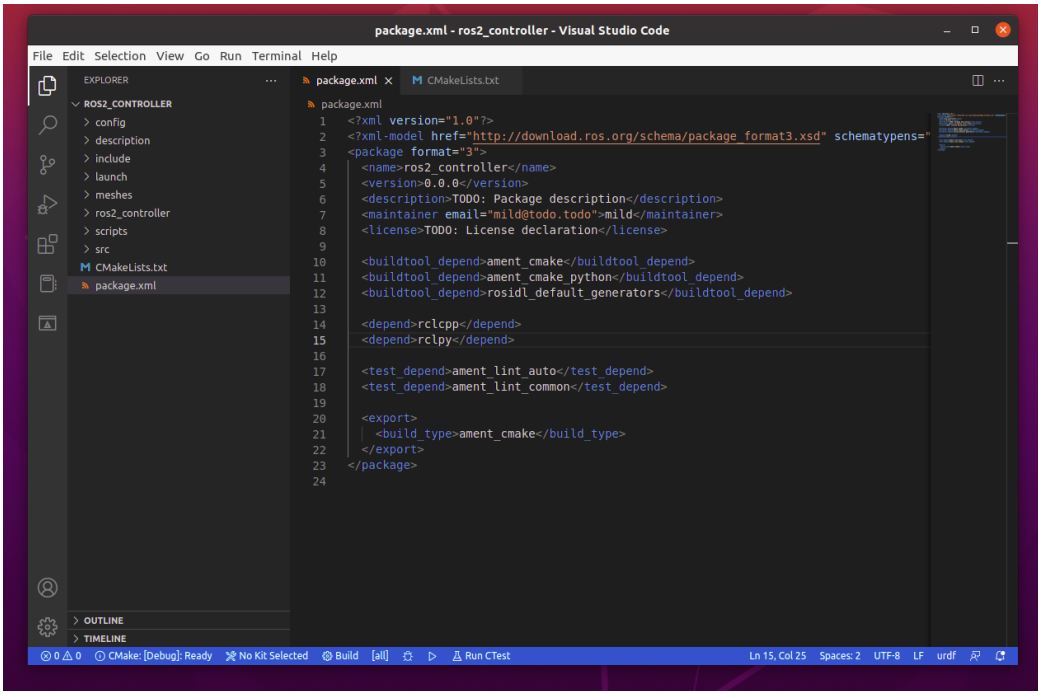


Configure CMakeLists.txt



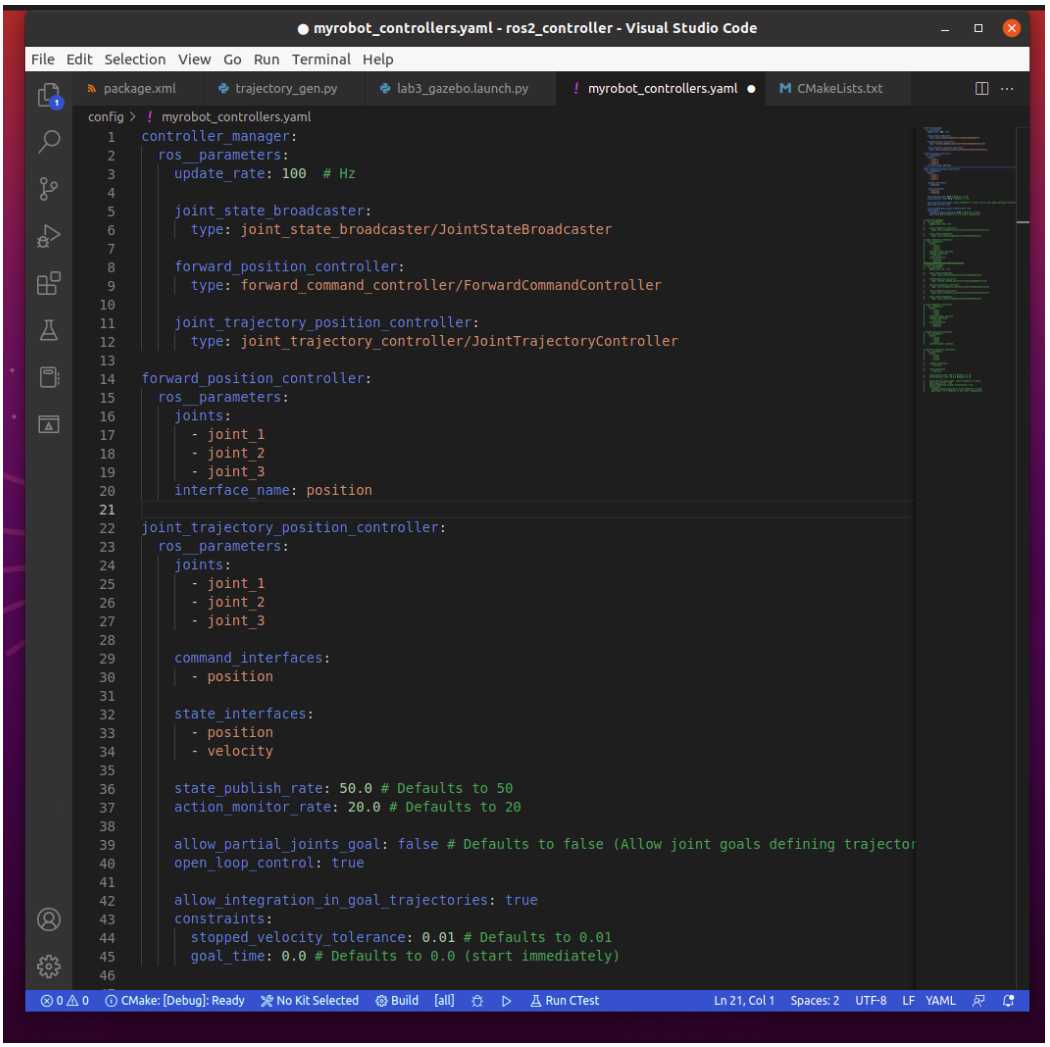
Configure package.xml





colcon build package: ros2_controller

Writing ROS Joint Trajectory Controller YAML File for Robotic Arm to Control its Joints.

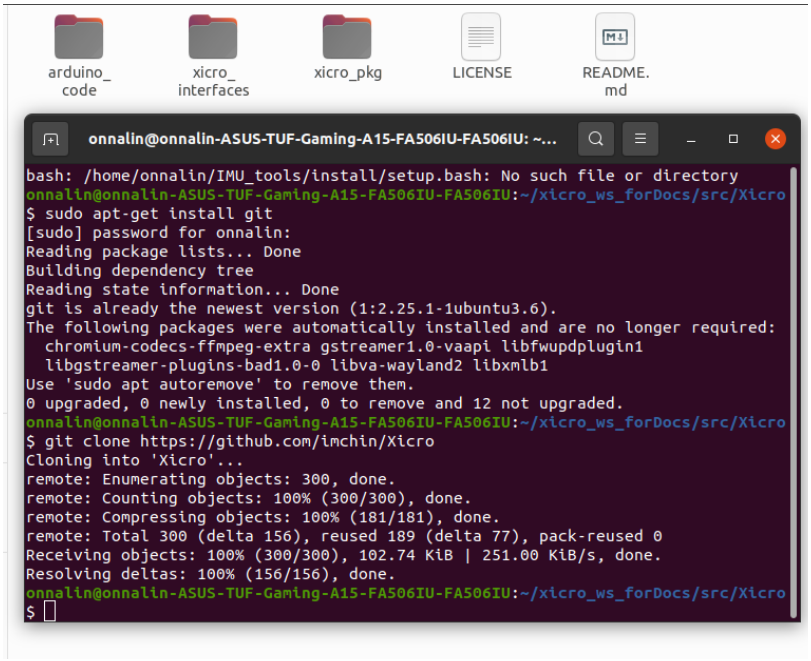


Connect IMU with Library XICRO

Git: https://github.com/imchin/Xicro

-Create Workspace for Install Xicro Package

1. Create Workspace: xicro_ws
2. Make Directory: mkdir Xicro
3. Clone Git: sudo apt-get install git and git clone https://github.com/imchin/Xicro



4. colcon build your workspace at: cd ~/xicro_ws/src

-Setup parameter in yaml

First step before Setup

pip3 uninstall serial

pip3 install pyserial

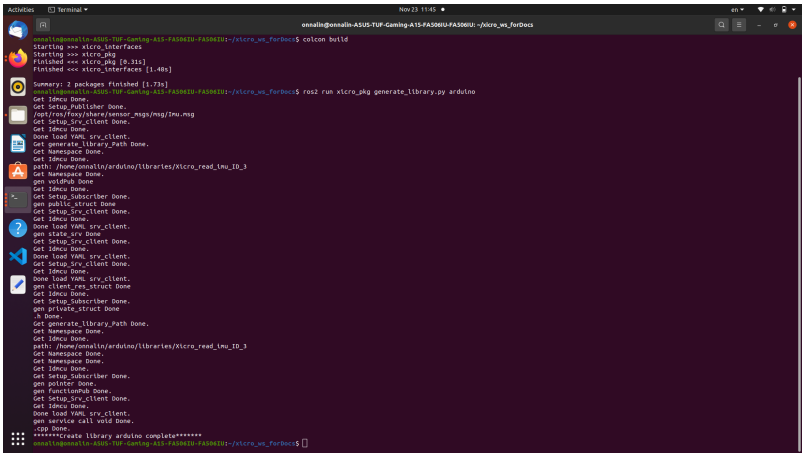
pip3 install numpy

1. Go to config: cd ~/xicro_ws/src/Xicro/xicro_pkg/config
2. Open code .yaml: code setup_xicro.yaml

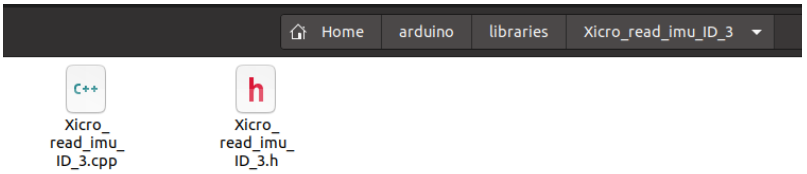
Before Create xicro library ,Check your arduino IDE (This lab we use arduino Version 18)

- Create xicro library

1. colcon build your workspace
2. Generate library: ros2 run xicro_pkg generate_library.py arduino

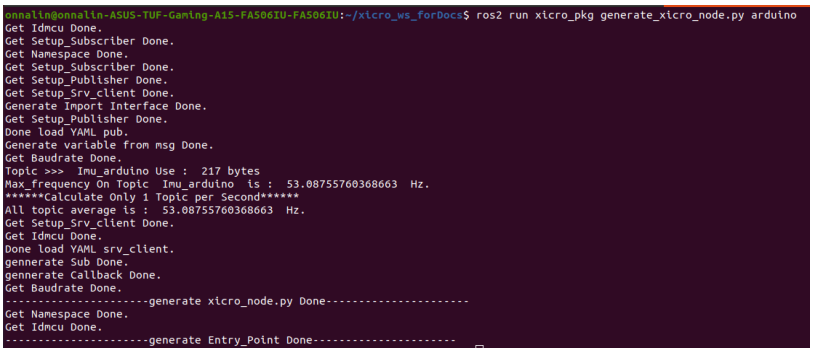


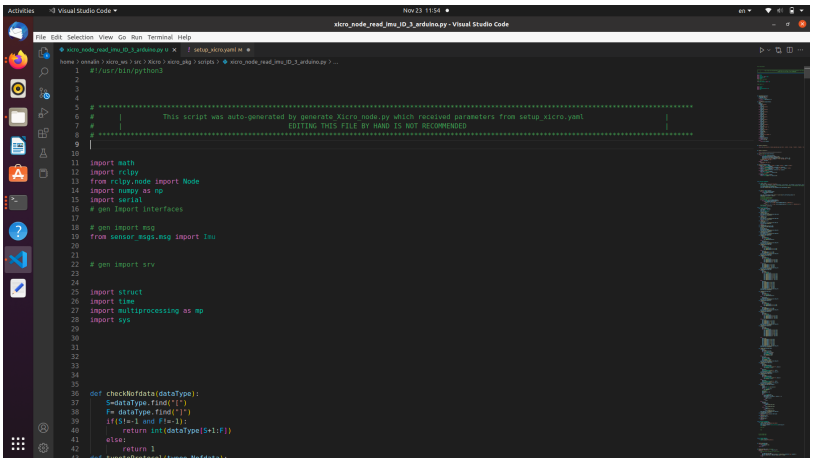
Check file generate library



-Generate Node of Xicro

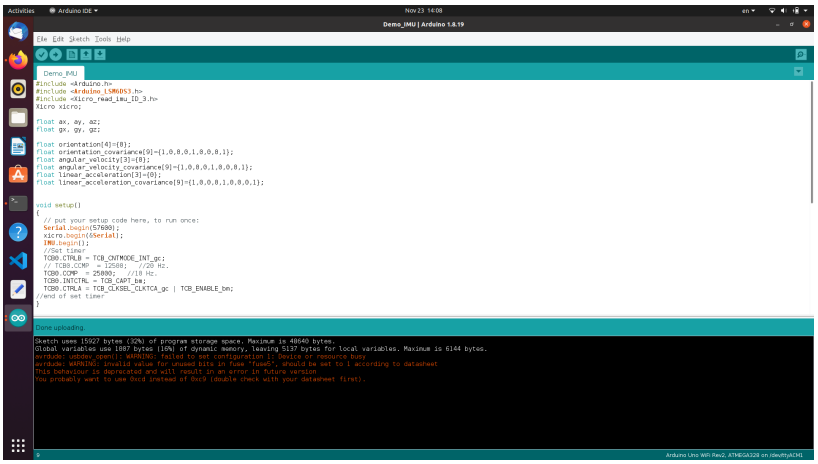
1. run on your workspace: ros2 run xicro_pkg generate_xicro_node_py arduino



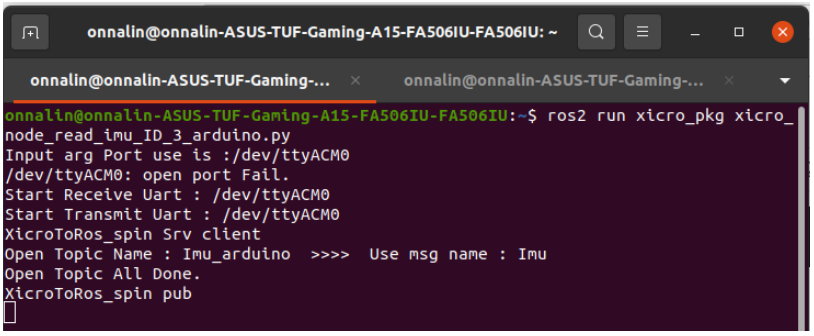


-Connect Xicro with Arduino

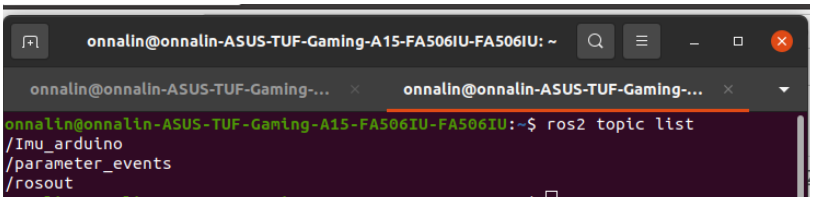
1. Upload this Code On Board



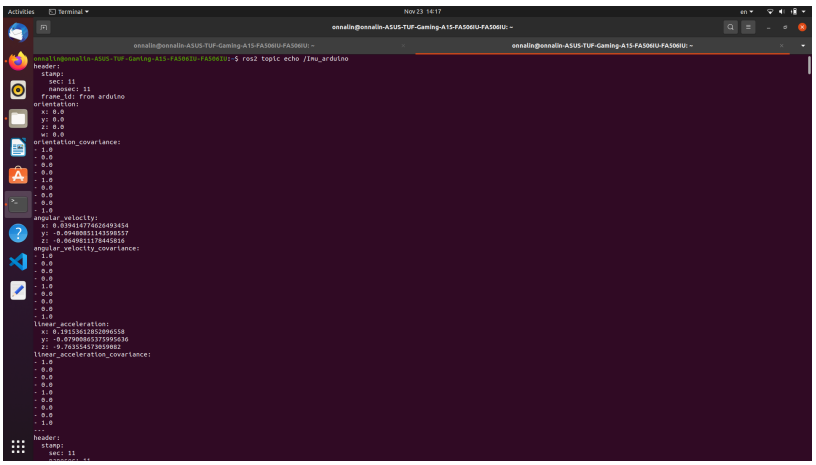
2. run command: ros2 run xicro_pkg xicro_node_read_imu_ID_3_arduino.py



3. Check Topic: ros2 topic list



4. Echo data on topic as your interest: ros2 topic echo /topic_name
   This lab interest about topic /Imu_arduino

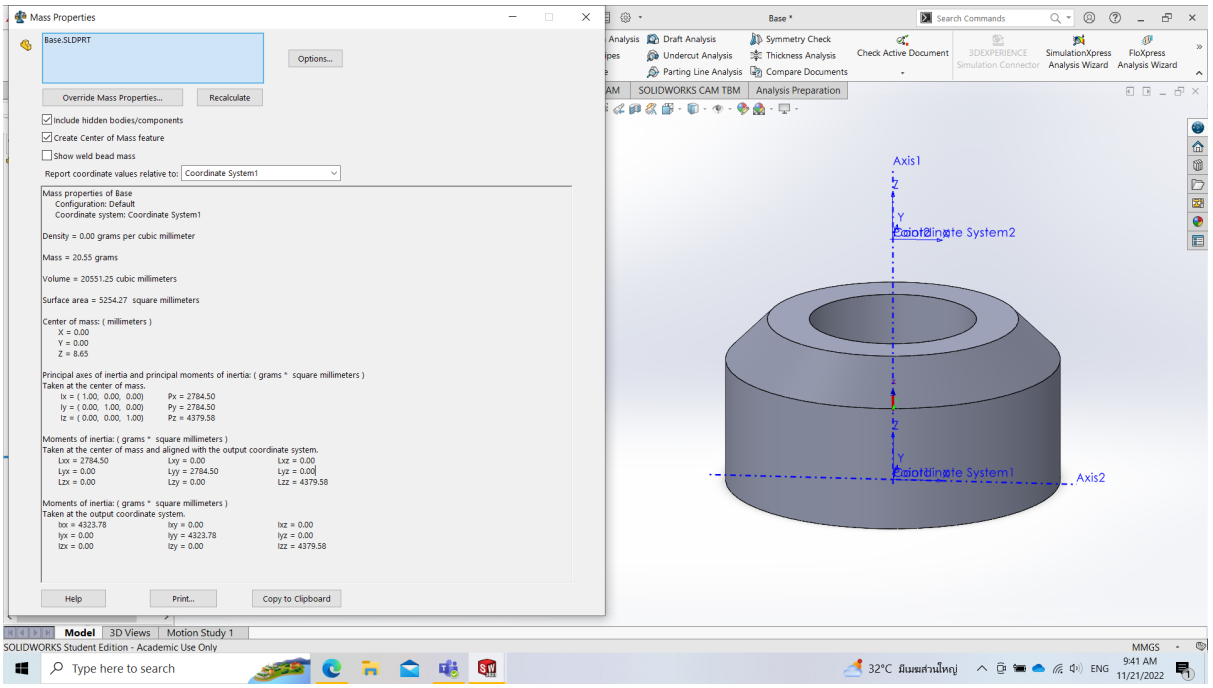


Now we can use topic /Imu_arduino for subscribe or something if you need.

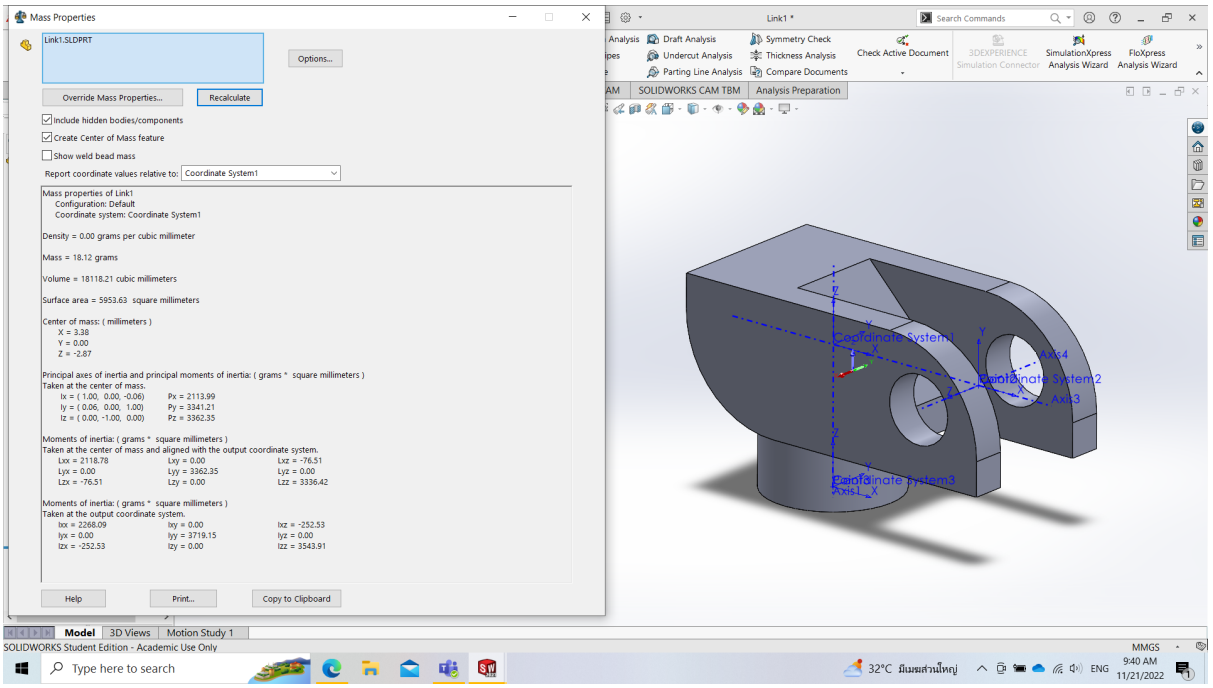
Xacro file with Gazebo

1. Design and Import Mass Properties from Solidworks

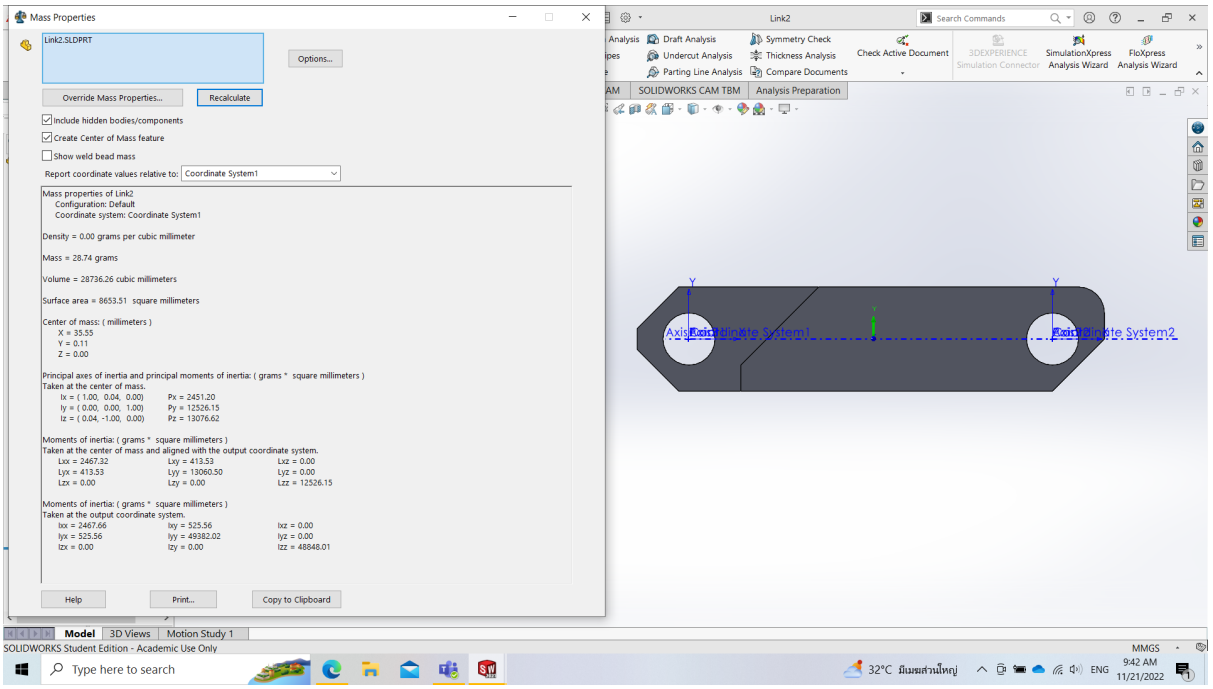
link_0



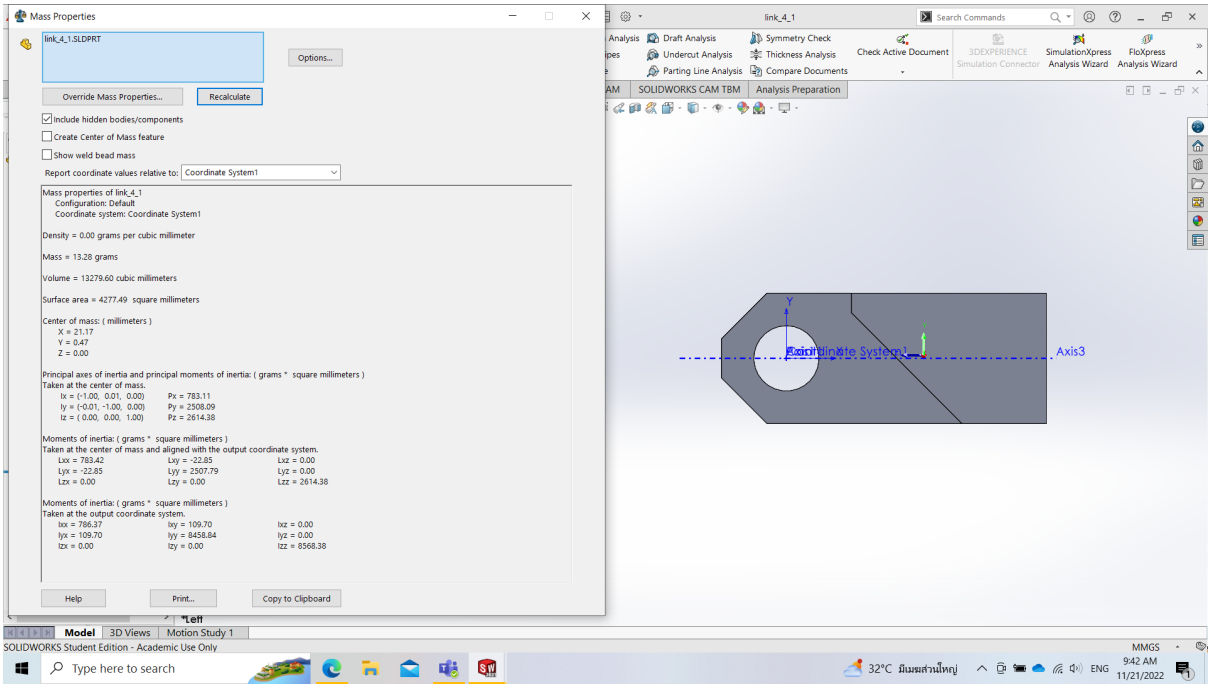
link_1



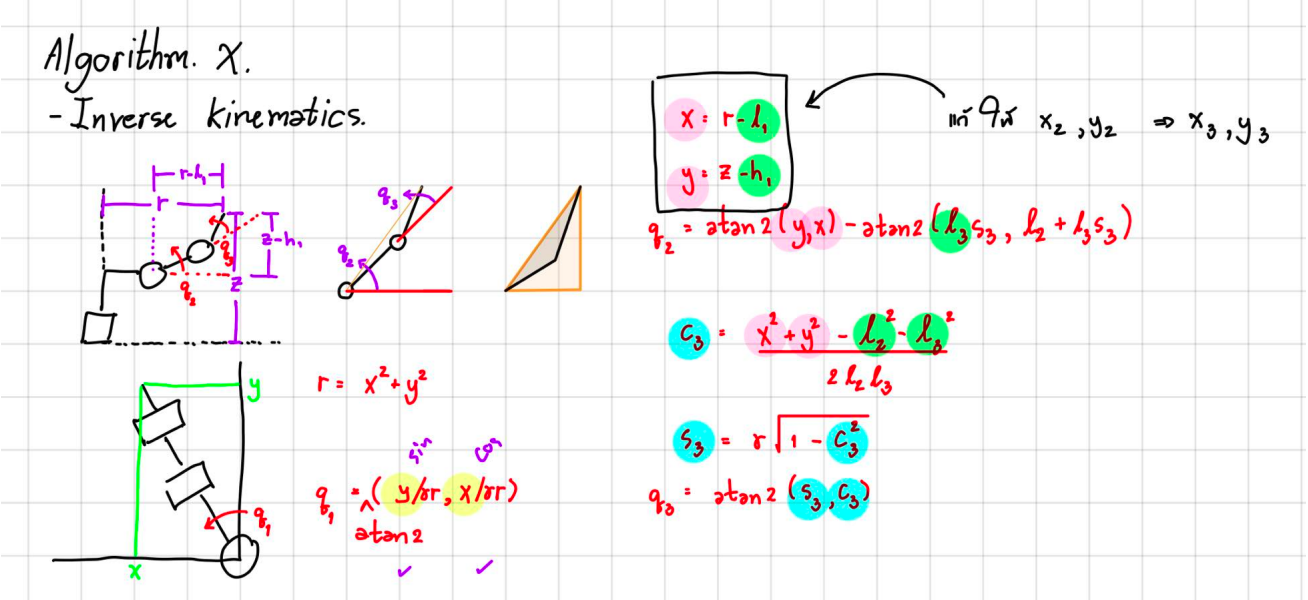
link_2


Mass Properties
Link2.SLDPRT
Options...
Override Mass Properties...
Recalculate
Include hidden bodies/components
Create Center of Mass feature
Show weld bead mass
Report coordinate values relative to: Coordinate System1
Mass properties of Link2
Configuration: Default
Coordinate system: Coordinate System1
Density = 0.00 grams per cubic millimeter
Mass = 28.74 grams
Volume = 28736.26 cubic millimeters
Surface area = 8653.51 square millimeters
Center of mass: (millimeters)
X = 35.55
Y = 0.11
Z = 0.00
Principal axes of inertia and principal moments of inertia: (grams * square millimeters)
Taken at the center of mass.
Ix = (1.00, 0.04, 0.00) Px = 2451.20
Iy = (0.00, 0.00, 1.00) Py = 12526.15
Iz = (0.04, -1.00, 0.00) Pz = 13076.62
Moments of inertia: (grams * square millimeters)
Taken at the center of mass and aligned with the output coordinate system.
Lxx = 2467.32 Lxy = 413.53 Lxz = 0.00
Lyx = 413.53 Lyy = 13060.50 Lyz = 0.00
Lzx = 0.00 Lzy = 0.00 Lzz = 12526.15
Moments of inertia: (grams * square millimeters)
Taken at the output coordinate system.
Ixx = 2467.66 Ixy = 525.56 Ixz = 0.00
Iyx = 525.56 Iyy = 49382.02 Iyz = 0.00
Izx = 0.00 Izy = 0.00 Izz = 48848.01
Help
Print...
Copy to Clipboard
Link2
Search Commands
Draft Analysis
Undercut Analysis
Parting Line Analysis
Symmetry Check
Thickness Analysis
Compare Documents
Check Active Document
SOLIDWORKS CAM TBM
Analysis Preparation
SimulationXpress Analysis Wizard
FloXpress Analysis Wizard
Model
3D Views
Motion Study 1
SOLIDWORKS Student Edition - Academic Use Only
Type here to search
32°C
ENG
9:42 AM
11/21/2022

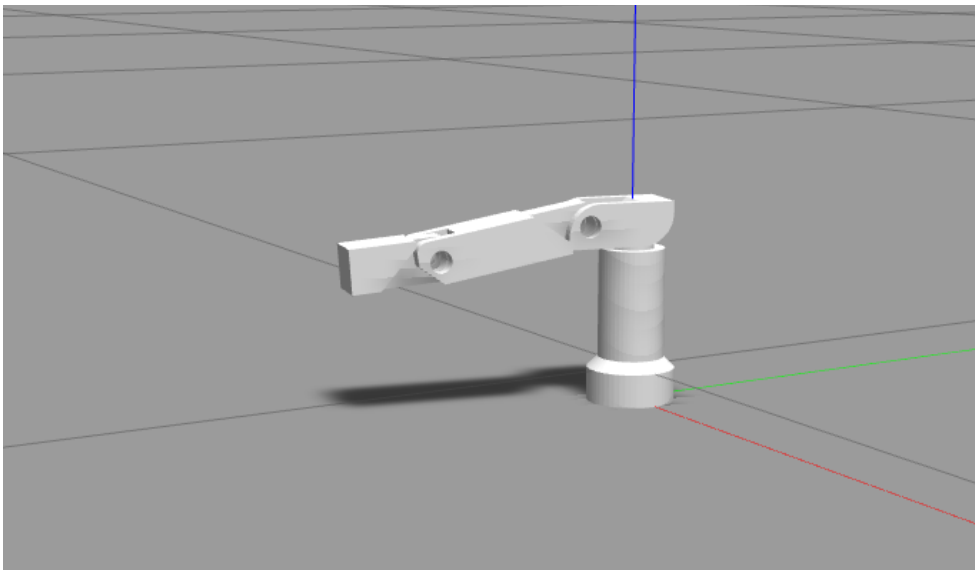
link_3


Mass Properties
link_4_1.SLDPRT
Options...
Override Mass Properties...
Recalculate
Include hidden bodies/components
Create Center of Mass feature
Show weld bead mass
Report coordinate values relative to: Coordinate System1
Mass properties of link_4_1
Configuration: Default
Coordinate system: Coordinate System1
Density = 0.00 grams per cubic millimeter
Mass = 13.28 grams
Volume = 13279.60 cubic millimeters
Surface area = 4277.49 square millimeters
Center of mass: (millimeters)
X = 21.17
Y = 0.47
Z = 0.00
Principal axes of inertia and principal moments of inertia: (grams * square millimeters)
Taken at the center of mass.
Ix = (-1.00, 0.01, 0.00) Px = 783.11
Iy = (-0.01, -1.00, 0.00) Py = 2508.09
Iz = (0.00, 0.00, 1.00) Pz = 2614.38
Moments of inertia: (grams * square millimeters)
Taken at the center of mass and aligned with the output coordinate system.
Lxx = 783.42 Lxy = -22.85 Lxz = 0.00
Lyx = -22.85 Lyy = 2507.79 Lyz = 0.00
Lzx = 0.00 Lzy = 0.00 Lzz = 2614.38
Moments of inertia: (grams * square millimeters)
Taken at the output coordinate system.
Ixx = 786.37 Ixy = 109.70 Ixz = 0.00
Iyx = 109.70 Iyy = 8458.84 Iyz = 0.00
Izx = 0.00 Izy = 0.00 Izz = 8568.38
Help
Print...
Copy to Clipboard
link_4_1
Search Commands
Draft Analysis
Undercut Analysis
Parting Line Analysis
Symmetry Check
Thickness Analysis
Compare Documents
Check Active Document
SOLIDWORKS CAM TBM
Analysis Preparation
SimulationXpress Analysis Wizard
FloXpress Analysis Wizard
Axis3
Model
3D Views
Motion Study 1
SOLIDWORKS Student Edition - Academic Use Only
Type here to search
32°C
ENG
9:42 AM
11/21/2022

-Calculate Inverse Kinematics



ภาพการทดลอง



CODE: urdf.xacro
นำค่า จาก CAD มาใส่ใน urdf.xacro

```
<?xml version="1.0"?>
<robot xmlns:xacro="http://www.ros.org/wiki/xacro"  name="robot">

	<!-- This is an example of a URDF. -->
	<!-- As we move through the file, new things to note will be pointed out. -->
	<!-- It's not meant an example of GOOD design, but an example of some of the various features of URDF/xacro. -->

	<!-- This will include all the contents of example_include.xacro first. Go check it out! -->
	<xacro:include filename="example_include.xacro" />

	<!-- This first link called "world" is empty -->
	<link name="world"></link>

	<!-- A simple fixed joint from our empty world link, to our base. -->
	<!-- The base origin is offset from the world origin. -->
	<joint name="base_joint" type="fixed">
	<origin xyz="0 0 0" rpy="0 0 0"/>
	<parent link="world"/>
	<child link="link_0"/>
	</joint>

	<!-- base_link is a large rectangular plate. Some things to note: -->
	<!-- - We set the visual origin Z to half the box height, so that the link origin sits at the bottom of the box -->
	<!-- - We set the collision to be identical to the visual -->
	<!-- - We specified the colour manually (but still need to enter a name) -->
	<!-- - We specified all the inertial parameters manually -->
	<link name="link_0">
	<visual>
	<origin xyz="0 0 0.0" rpy="0 0 0"/>
	<geometry>
		<mesh filename="file:///$(find ros2_controller)/meshes/link_0.STL"/>
	</geometry>
	<material name="green">
		<color rgba="0.5 0.1 0.5 1"/>
	</material>
	</visual>
	<collision>
	<origin xyz="0 0 0.0" rpy="0 0 0"/>
	<contact_coefficients mu="0.5" kp="1000.0" kd="1.0"/>
	<geometry>
		<mesh filename="file:///$(find ros2_controller)/meshes/link_0.STL"/>
	</geometry>
```

```
        </collision>
        <inertial>
        <origin xyz="0 0 0.00865" rpy="0 0 0"/>
        <mass value="0.02055" />
        <inertia ixx="21.735" ixy="0.0" ixz="0.0" iyy="21.735" iyz="0.0" izz="7.570" />
        </inertial>
        </link>

        <joint name="joint_0" type="fixed">
        <parent link="link_0"/>
        <child link="link_1"/>
        <origin rpy="0 0 0.0" xyz="0 0 0.08"/>
        <axis xyz="0 0 1"/>
        <limit lower="-3.14" upper="3.14" effort="1000.0" velocity="1000.0"/>
        </joint>

        <link name="link_1">
        <visual>
        <origin xyz="0 0 0.0" rpy="0 0 0"/>
        <geometry>
                <mesh filename="file:///$(find ros2_controller)/meshes/link_1.STL"/>
        </geometry>
        <material name="green">
                <color rgba="0.2 1 0.2 1"/>
        </material>
        </visual>
        <collision>
        <origin xyz="0 0 0.0" rpy="0 0 0.0"/>
        <contact_coefficients mu="0.2" kp="1000.0" kd="1.0"/>
        <geometry>
                <mesh filename="file:///$(find ros2_controller)/meshes/link_1.STL"/>
        </geometry>
        </collision>
        <inertial>
        <origin xyz="0.00338 0 -0.00287" rpy="0 0 0"/>
        <mass value="0.01812" />
        <inertia ixx="2.118" ixy="0.0" ixz="-0.07651" iyy="3.362" iyz="0.0" izz="3.336"
/>
        </inertial>
        </link>

        <joint name="joint_1" type="revolute">
        <parent link="link_0"/>
        <child link="link_1"/>
        <origin rpy="0 0 0.0" xyz="0 0 0.08"/>
        <axis xyz="0 0 1"/>
        <limit lower="-3.14" upper="3.14" effort="1500.0" velocity="1000.0"/>
        <dynamics damping="0" friction="0"/>
```

```
</joint>

<link name="link_2">
<visual>
<origin xyz="0 0 0.0" rpy="0 0 0"/>
<geometry>
        <mesh filename="file:///$(find ros2_controller)/meshes/link_2.STL"/>
</geometry>
<material name="green">
        <color rgba="0.2 1 0.2 1"/>
</material>
</visual>
<collision>
<contact_coefficients mu="0" kp="1000.0" kd="1.0"/>
<origin xyz="0 0 0.0" rpy="0 0 0"/>
<geometry>
</geometry>
</collision>
<inertial>
<origin xyz="0.03555 0.00011 0.0" rpy="0 0 0.0"/>
<mass value="0.02874" />
<inertia ixx="2.467" ixy="0.413" ixz="0.0" iyy="13.060" iyz="0.0" izz="12.526" />
</inertial>
</link>

<joint name="joint_2" type="revolute">
<parent link="link_1"/>
<child link="link_2"/>
<origin rpy="1.57 0 0" xyz="0.025 0 0"/>
<axis xyz="0 0 1"/>
<limit lower="-3.14" upper="3.14" effort="1000.0" velocity="1000.0"/>
<dynamics damping="0" friction="0"/>
</joint>

<link name="link_3">
<visual>
<origin xyz="0 0 0" rpy="0 0 0"/>
<geometry>
        <mesh filename="file:///$(find ros2_controller)/meshes/link_3.STL"/>
</geometry>
<material name="green">
        <color rgba="0.2 1 0.2 1"/>
</material>
</visual>
<collision>
<contact_coefficients mu="0" kp="1000.0" kd="1.0"/>
<origin xyz="0 0 0.0" rpy="0 0 0"/>
<geometry>
```

```
        </geometry>
        </collision>
        <inertial>
        <origin xyz="0.02117 0 0.00047" rpy="0 0 0"/>
        <mass value="0.01328" />
        <inertia ixx="0.783" ixy="-0.0228" ixz="0.0" iyy="2.507" iyz="0.0" izz="2.614" />
        </inertial>
        </link>

        <joint name="joint_3" type="revolute">
        <parent link="link_2"/>
        <child link="link_3"/>
        <origin rpy="0 0.0 1.57" xyz="0.07 0 0"/>
        <axis xyz="0 0 -1"/>
        <limit lower="-3.14" upper="3.14" effort="1000.0" velocity="1000.0"/>
        <dynamics damping="10" friction="5"/>
        </joint>

        <joint name="joint_eff" type="fixed">
        <parent link="link_3"/>
        <child link="end_effector"/>
        <origin rpy="-1.57 0 1.57" xyz="0.4 0 0"/>
        <axis xyz="0 0 1"/>
        </joint>

        <link name="end_effector">
        </link>

        <xacro:include filename="example_gazebo.xacro" />


</robot>
```