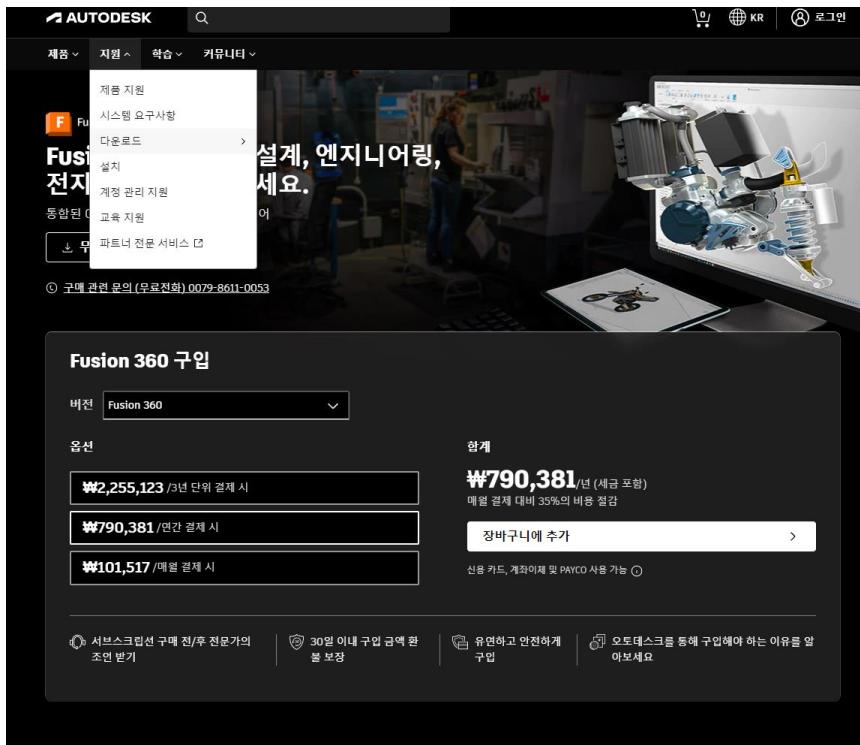


∨ ∧ ∙ ∙ ∙

Robit Fusion 360 Seminar

2023. 01



➤ Install Fusion 360

- [Fusion 360](#) (Autodesk 계정으로 로그인)
- 지원 > 다운로드 > 학생 및 교사 or 무료 체험판 다운로드

➤ Install WSL2

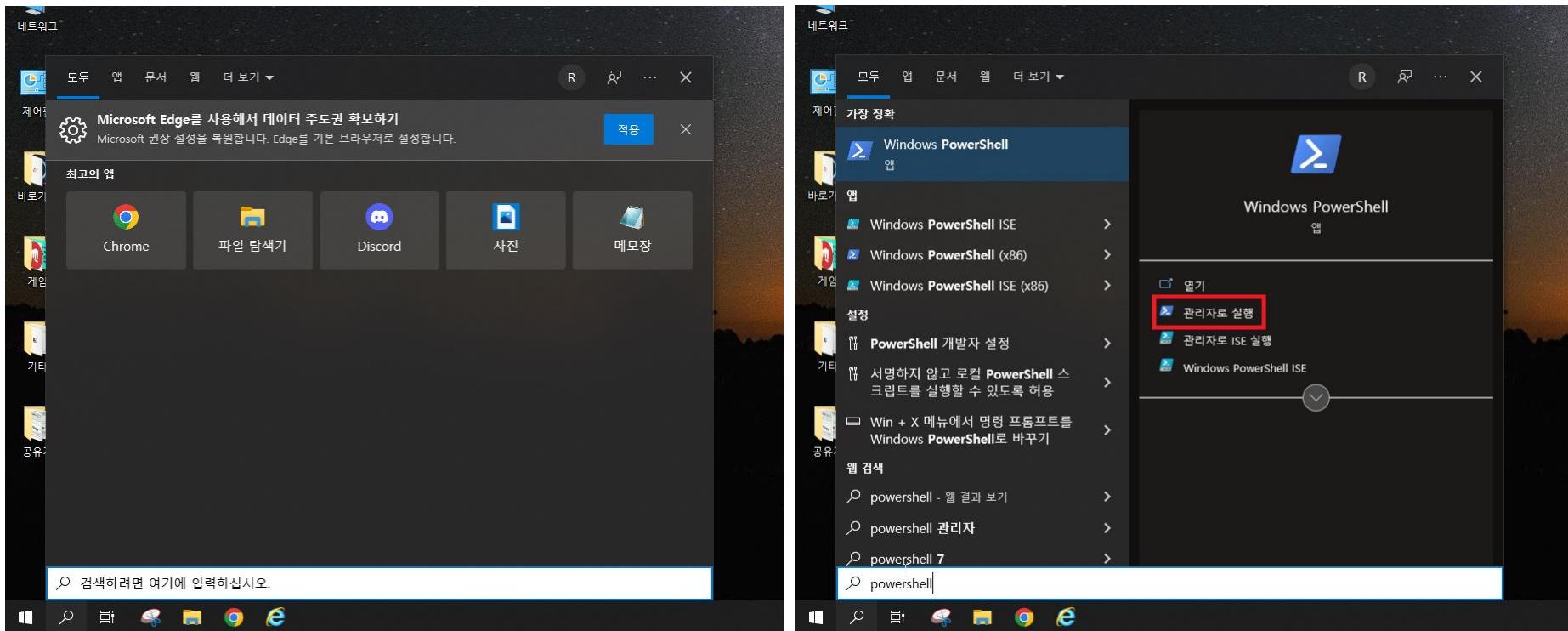
- WSL = Windows Subsystem for Linux

기능 비교

기능	WSL 1	WSL 2
Windows와 Linux 통합	✓	✓
빠른 부팅 시간	✓	✓
기존 Virtual Machines보다 작은 리소스 공간	✓	✓
현재 버전의 VMware 및 VirtualBox에서 실행	✓	✓
관리 VM	✗	✓
전체 Linux 커널	✗	✓
전체 시스템 호출 호환성	✗	✓
OS 파일 시스템 간 성능	✓	✗

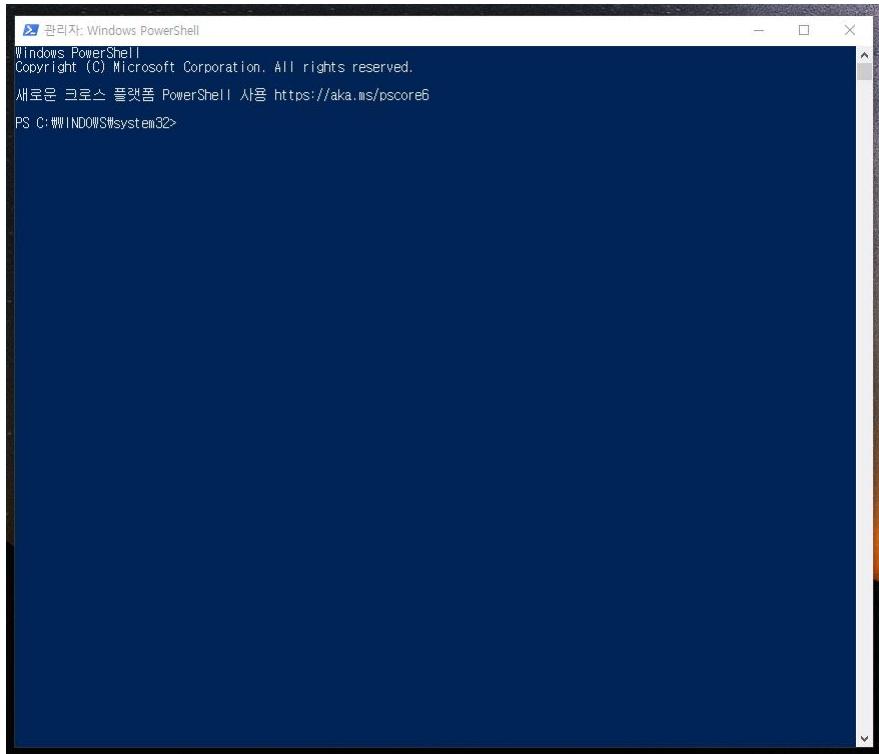
➤ Install WSL2 Step1

- 원도우 키 + S
- PowerShell 검색 -> 관리자로 실행



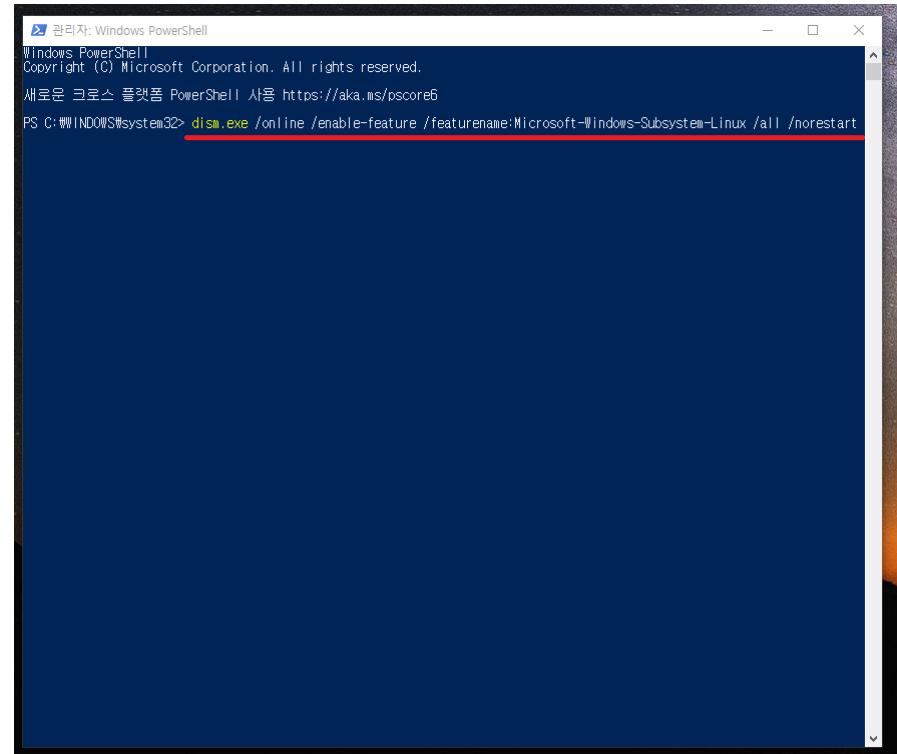
➤ Install WSL2 Step2

- **dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart**



```
관리자: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6
PS C:\WINDOWS\system32>
```

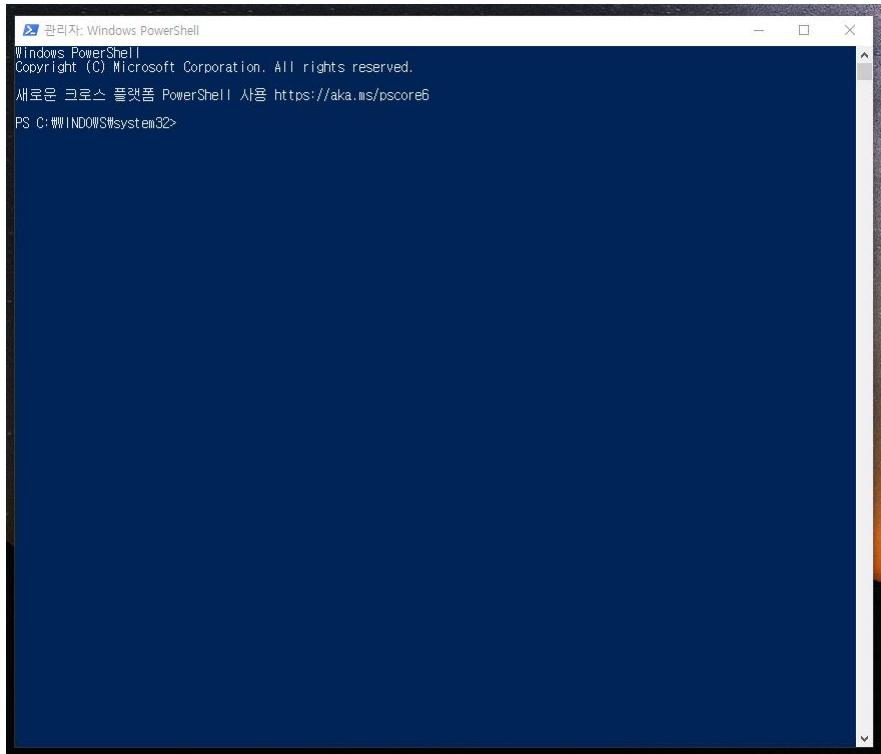


```
관리자: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6
PS C:\WINDOWS\system32> dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

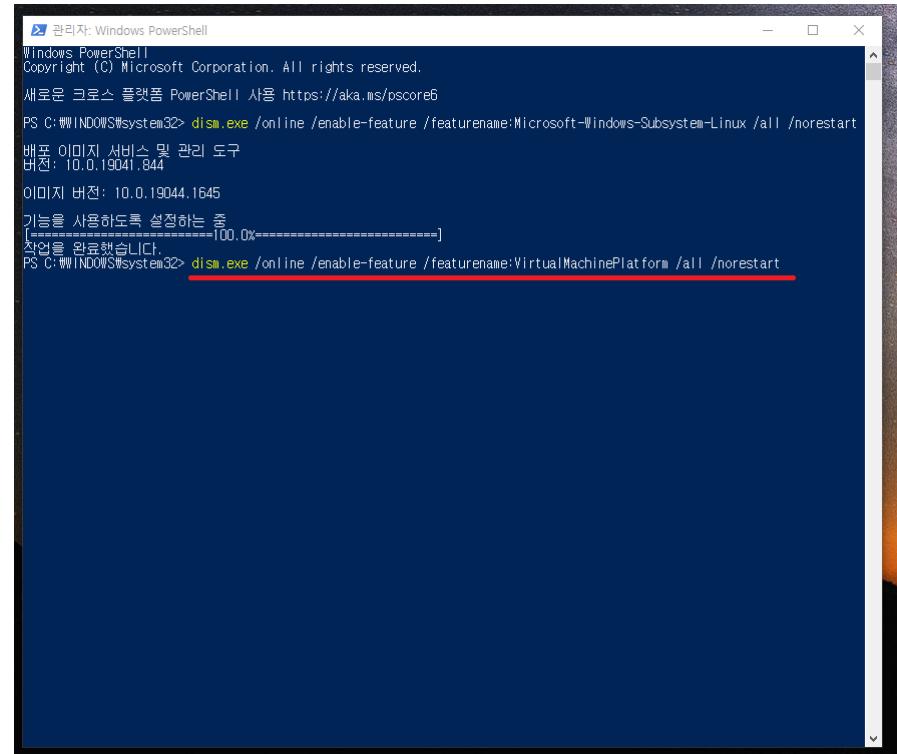
➤ Install WSL2 Step3

- `dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart`



```
관리자: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6
PS C:\WINDOWS\system32>
```

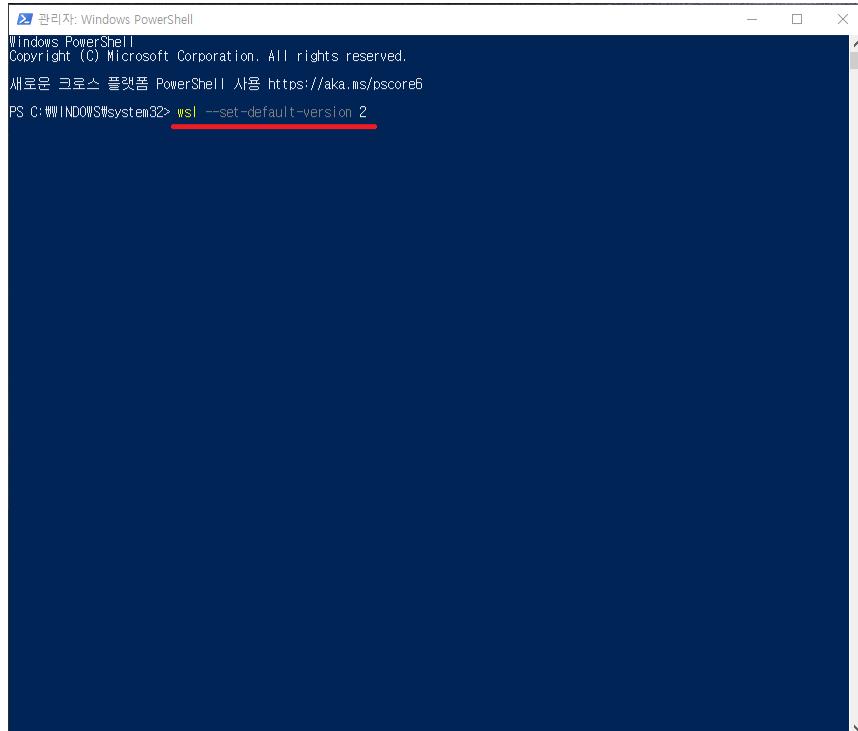


```
관리자: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

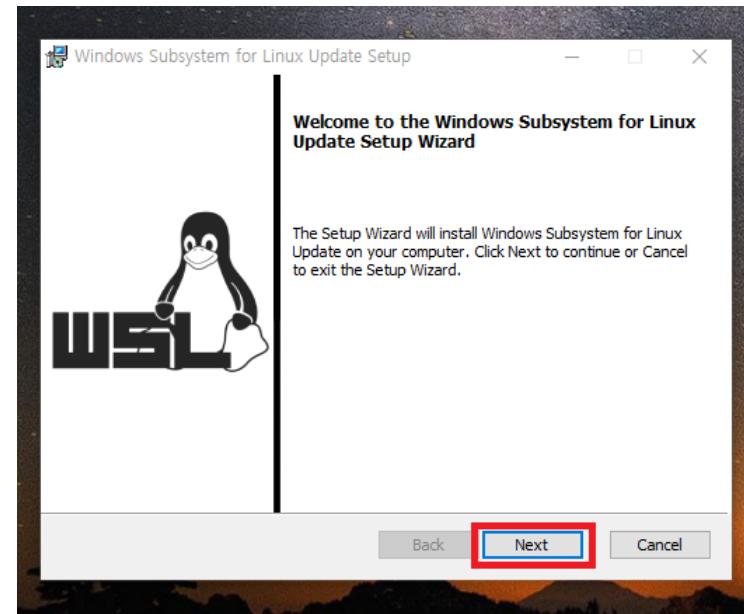
새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6
PS C:\WINDOWS\system32> dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
배포 이미지 서비스 및 관리 도구
버전: 10.0.19041.844
이미지 버전: 10.0.19044.1645
기능을 사용하도록 설정하는 중 [=====100.0%=====]
작업을 완료했습니다.
PS C:\WINDOWS\system32> dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

➤ Install WSL2 Step4

- 컴퓨터를 재부팅 후 PowerShell 실행
- wsl --set-default-version 2 -> wsl2로 기본설정 변경
- 윈도우용 리눅스 커널 설치

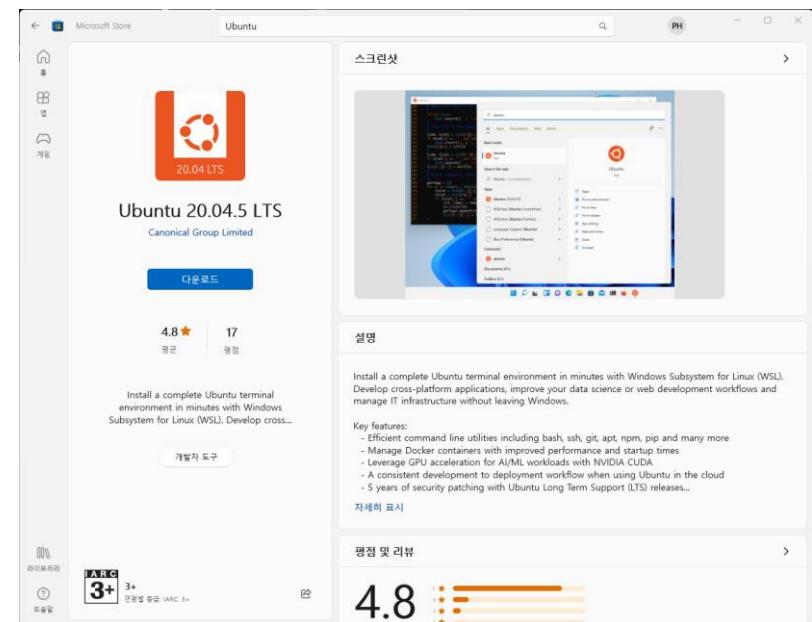
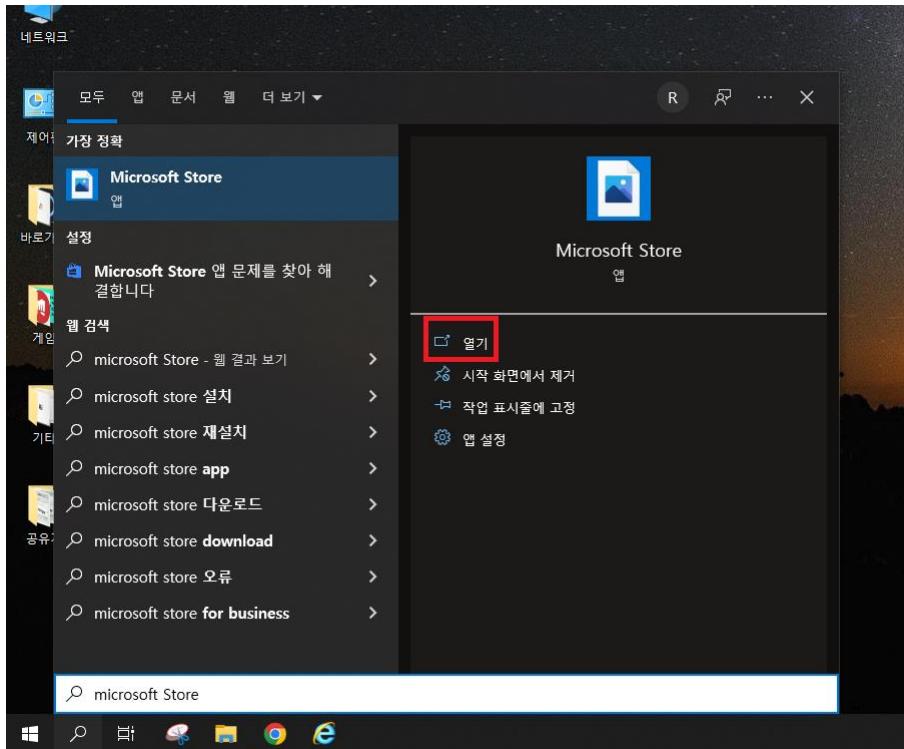


관리자: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
새로운 클로스 플랫폼 PowerShell 사용 <https://aka.ms/powershell>
PS C:\WINDOWS\system32> wsl --set-default-version 2



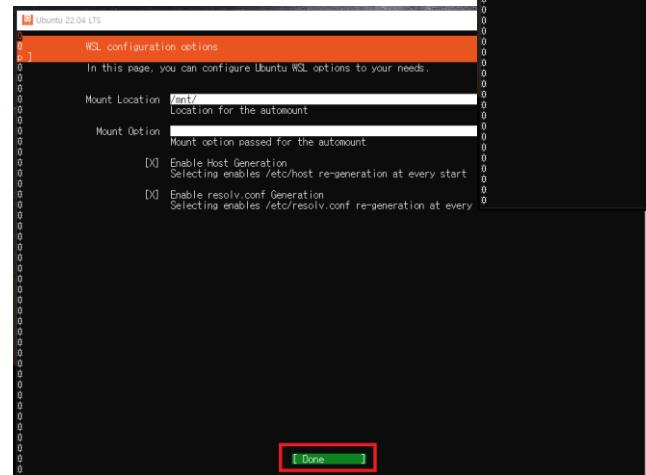
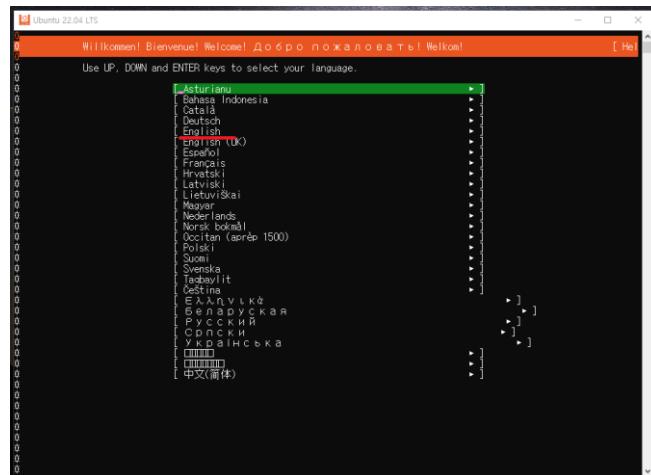
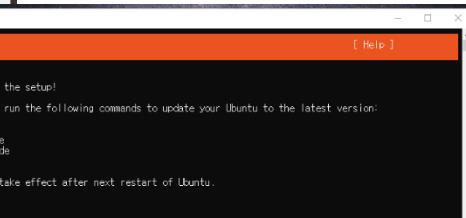
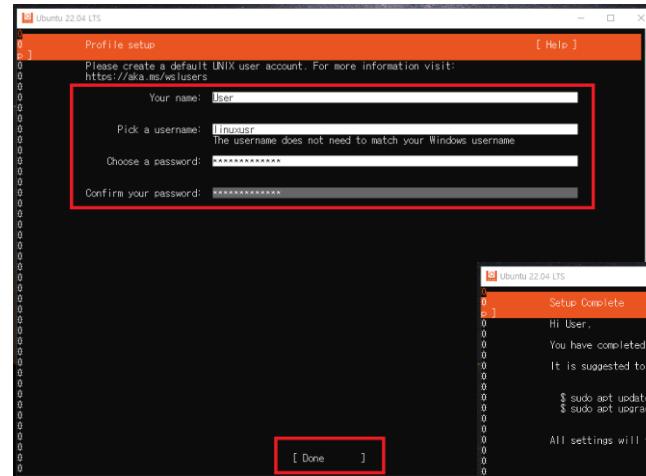
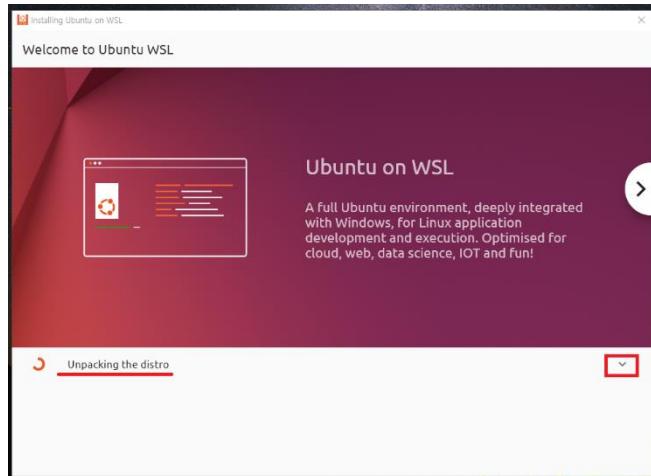
➤ Install WSL2 Step5

- Microsoft Store 열기 -> Ubuntu 검색
- Ubuntu 20.04 설치



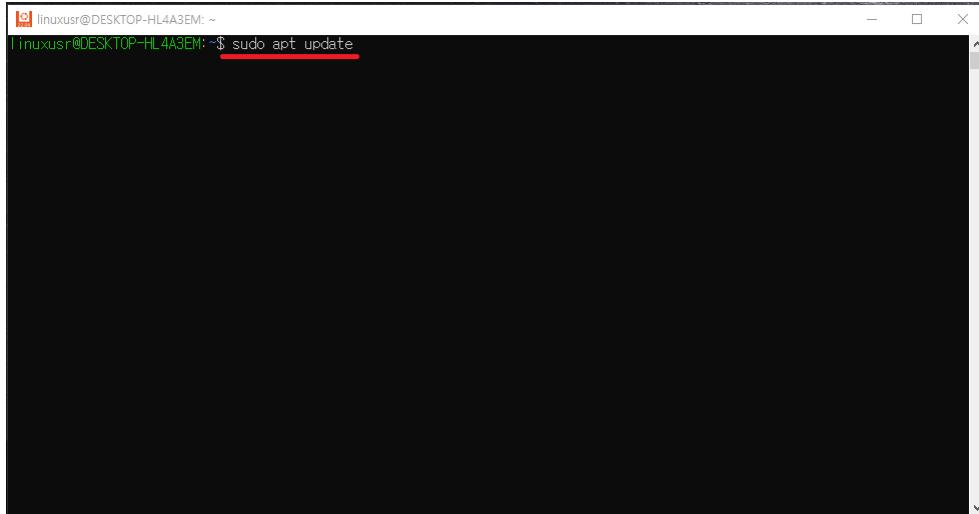
➤ Install WSL2 Step6

- 언어 선택 -> English
- Name & Password -> 편한대로..
- Reboot Now -> Enter 키를 눌러 WSL 재부팅

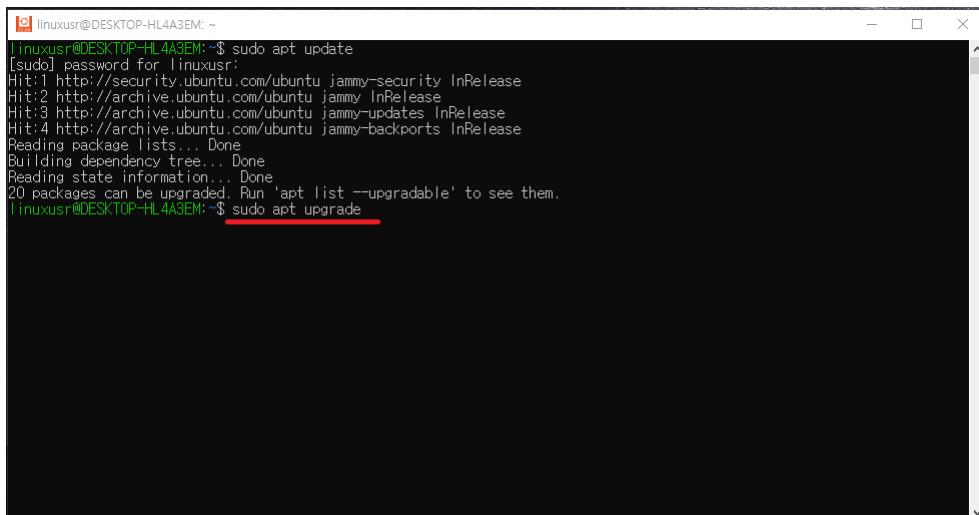


➤ Install WSL2 Step7

- sudo apt update
- sudo apt upgrade



```
linuxusr@DESKTOP-HL4A3EM: ~
linuxusr@DESKTOP-HL4A3EM: ~$ sudo apt update
```



```
linuxusr@DESKTOP-HL4A3EM: ~
linuxusr@DESKTOP-HL4A3EM: ~$ sudo apt update
[sudo] password for linuxusr:
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
20 packages can be upgraded. Run 'apt list --upgradable' to see them.
linuxusr@DESKTOP-HL4A3EM: ~$ sudo apt upgrade
```

사전 설치 메뉴얼



- Install ROS
- roscore
- rviz

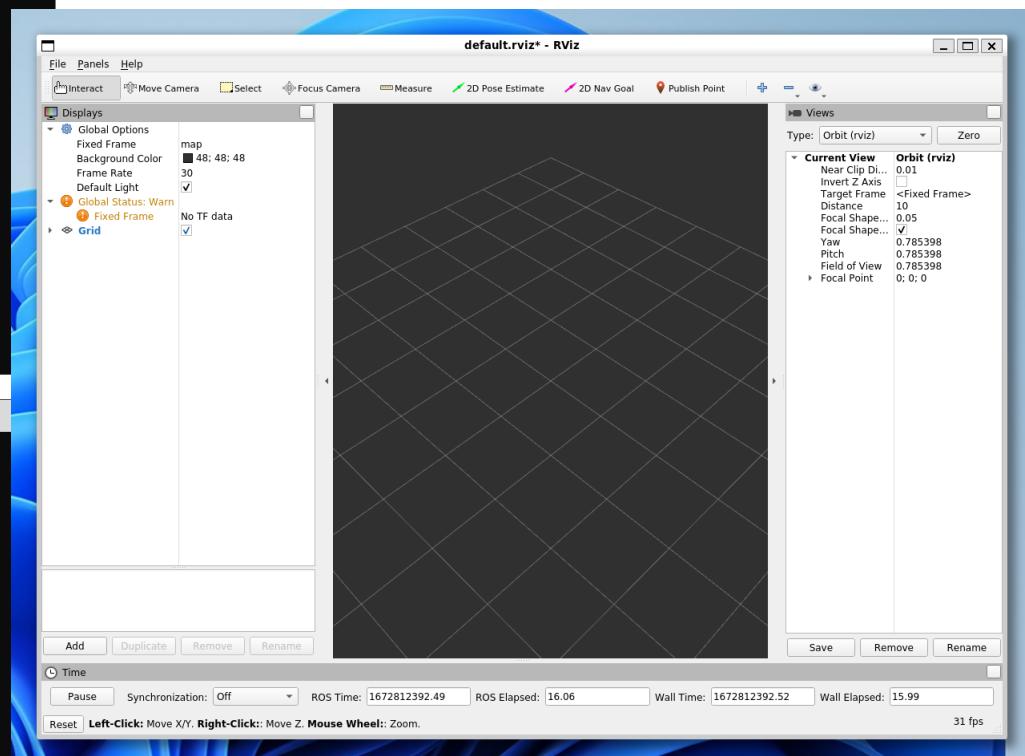
```
roscore http://127.0.0.1:11311/ x phj@DESKTOP-H0V58J5:~ x + v
phj@DESKTOP-H0V58J5:~$ roscore
... logging to /home/phj/.ros/log/d6ec169c-8bf5-11ed-be7e-00155dbc7e6c/roslaunch-DESKTOP-H0V58J5-300.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://127.0.0.1:44889/
ros_comm version 1.15.15

SUMMARY
=====
PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.15

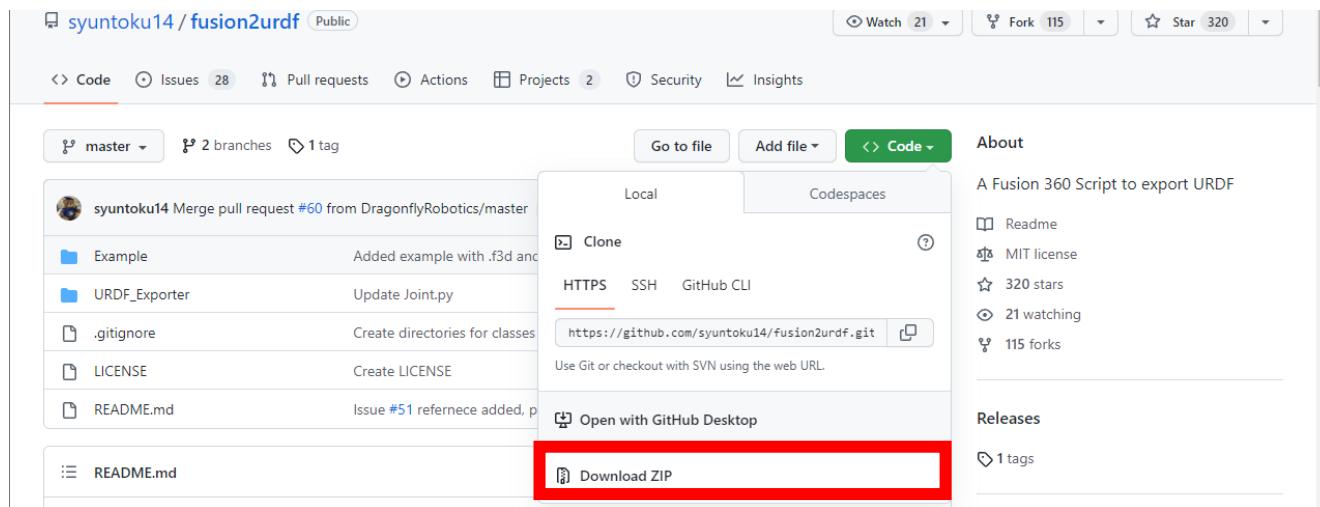
NODES
auto-starting new master
process[master]: started with pid [308]
ROS_MASTER_URI=http://127.0.0.1:11311/

setting /run_id to d6ec169c-8bf5-11ed-be7e-00155dbc7e6c
process[rosout-1]: started with pid [318]
started core service [/rosout]
```



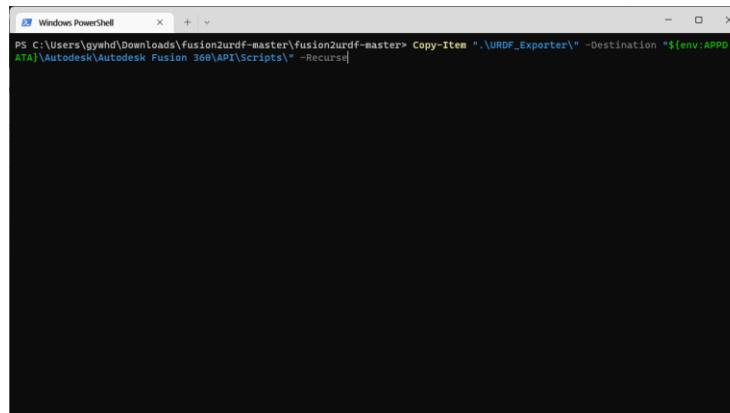
```
roscore http://127.0.0.1:11311/ x phj@DESKTOP-H0V58J5:~ x + v
phj@DESKTOP-H0V58J5:~$ rviz
```

➤ Download [fusion2urdf](#)



➤ Open PowerShell

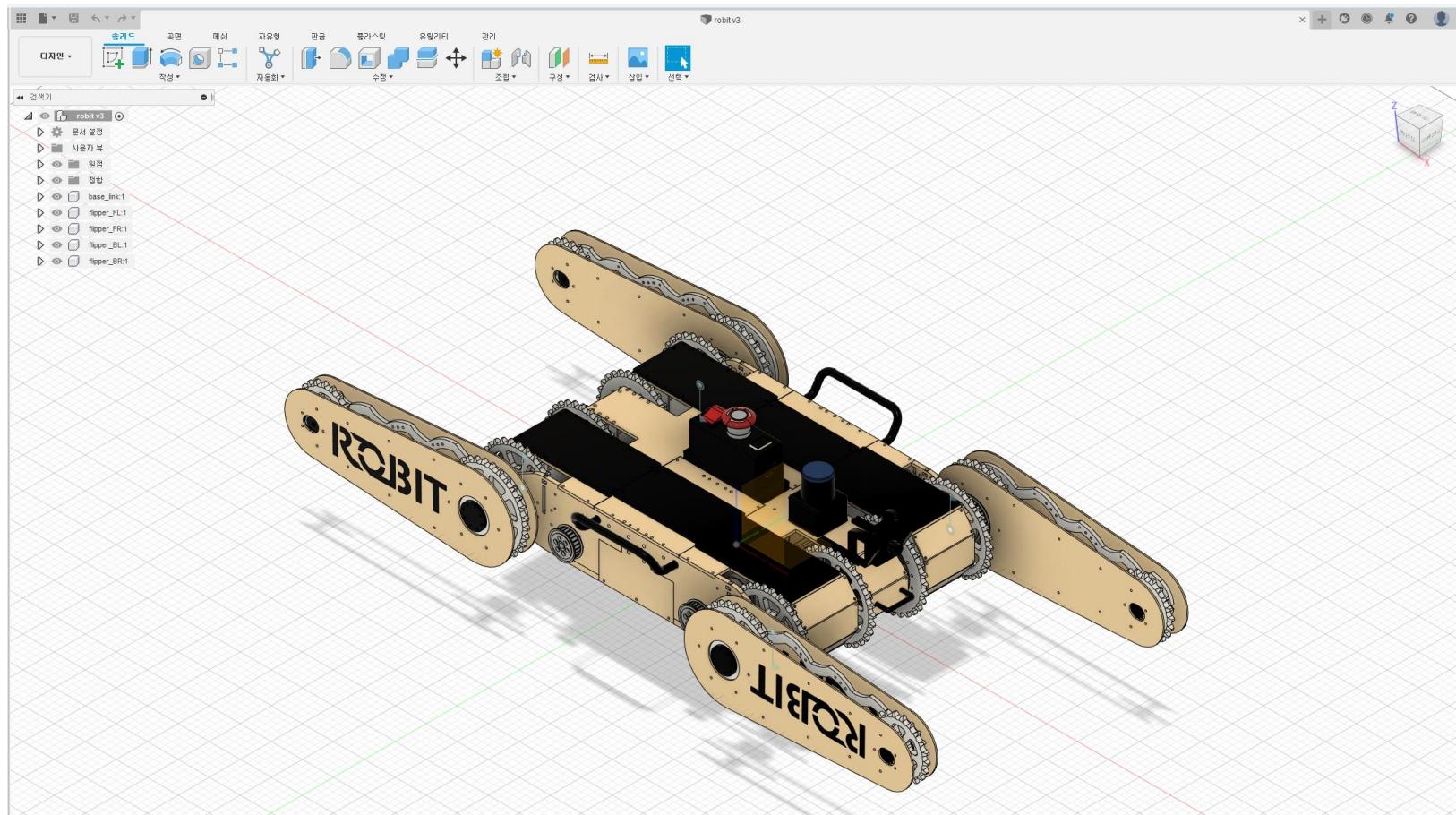
- Copy-Item ".\URDF_Exporter\" -Destination "\${env:APPDATA}\Autodesk\Autodesk Fusion 360\API\Scripts\" -Recurse



```
PS C:\Users\gywhd\Downloads\fusion2urdf-master\fusion2urdf-master> Copy-Item ".\URDF_Exporter\" -Destination "${env:APPDATA}\Autodesk\Autodesk Fusion 360\API\Scripts\" -Recurse
```

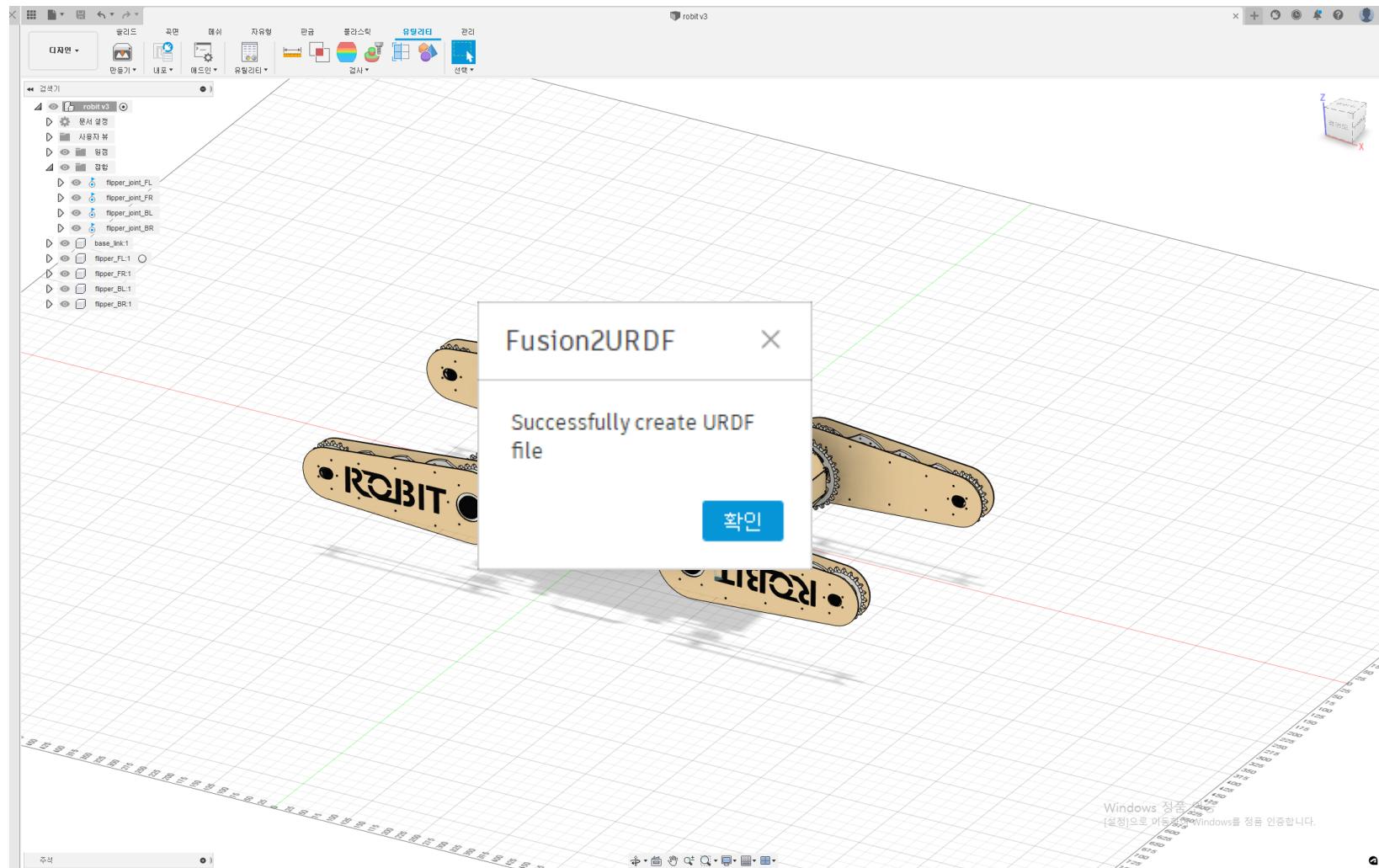
Simple Urdf

- Check origin (pos / rot)
- Link must be component
- Make Joint(접합)
- Set first link name as **base_link**
- Save before make URDF



Simple Urdf

➤ 유틸리티 -> 애드인 -> URDF_Exporter (실행)

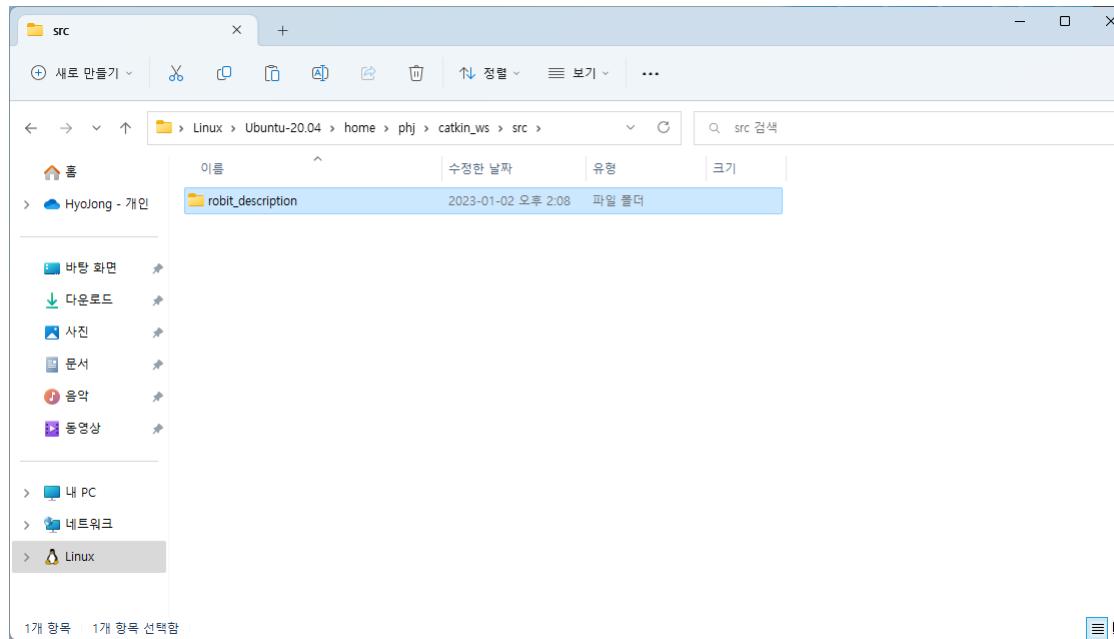


➤ Enable software rendering

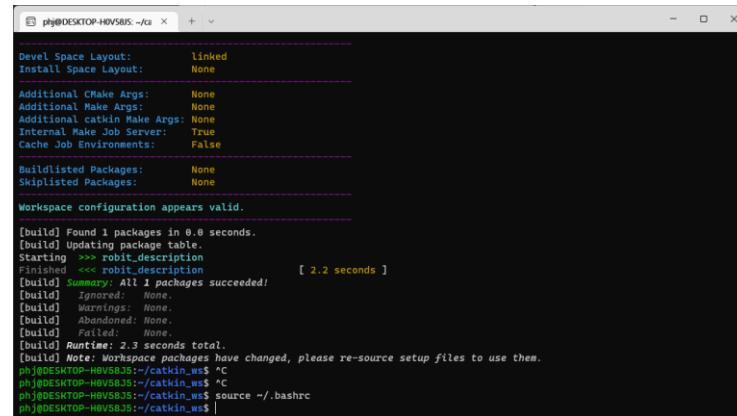
- code ~/.bashrc (install VSCode) or vim ~/.bashrc
- Add export LIBGL_ALWAYS_SOFTWARE=1

```
132 #RVIZ
133 export LIBGL_ALWAYS_SOFTWARE=1
134
```

➤ Move description file to catkin_ws/src



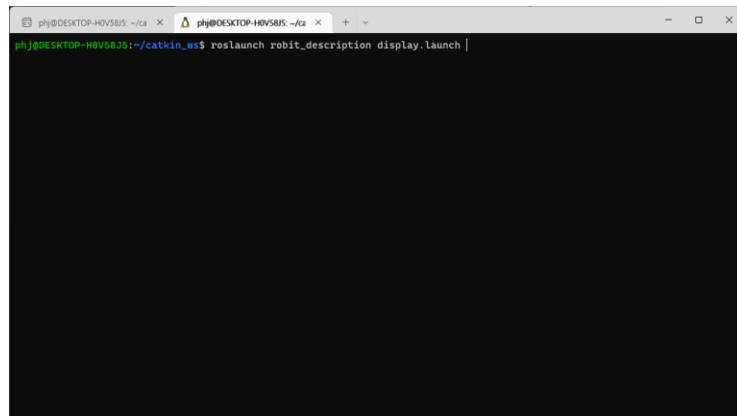
- Enable software rendering
 - catkin build or catkin_make
 - source ~/.bashrc



```
phj@DESKTOP-H0V58JS:~/ca x + v
Devel Space Layout: linked
Install Space Layout: None
Additional CMake Args: None
Additional Make Args: None
Additional catkin Make Args: None
Internal Make Job Server: True
Cache Job Environments: False
Buildlisted Packages: None
Skiplisted Packages: None
Workspace configuration appears valid.

[build] Found 1 packages in 0.0 seconds.
[build] Updating workspace...
Starting >>> robit_description
Finished <<< robit_description [ 2.2 seconds ]
[build] Summary: All 1 packages succeeded!
[build] Ignored: None.
[build] Warnings: None.
[build] Abandoned: None.
[build] Failed: None.
[build] Runtime: 2.3 seconds total.
[build] If your package's CMakeLists.txt file has changed, please re-source setup files to use them.
phj@DESKTOP-H0V58JS:~/catkin_ws $ ^C
phj@DESKTOP-H0V58JS:~/catkin_ws $ ^C
phj@DESKTOP-H0V58JS:~/catkin_ws $ source ~/.bashrc
phj@DESKTOP-H0V58JS:~/catkin_ws $ |
```

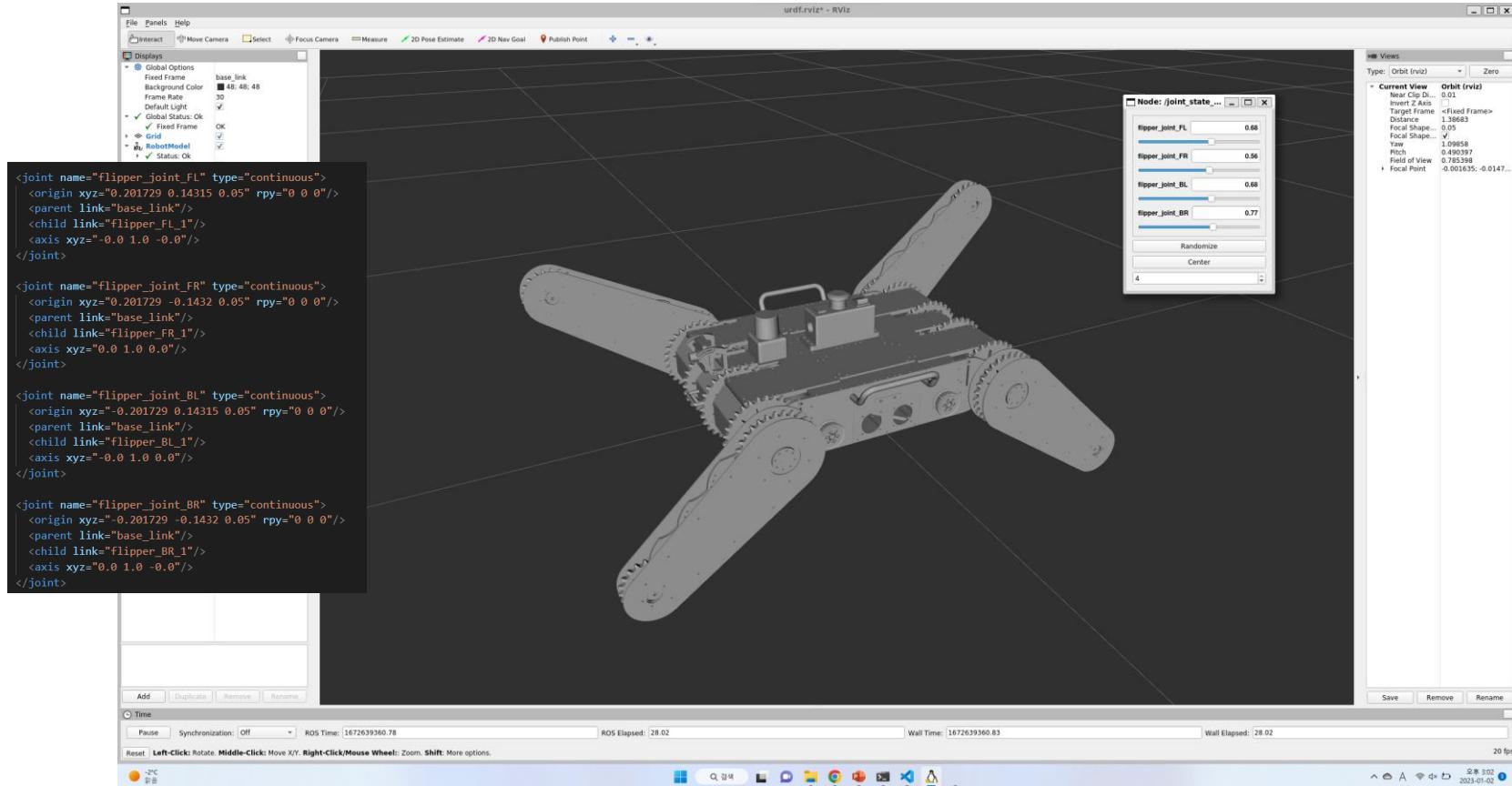
- rosrun robit_description display.launch



```
phj@DESKTOP-H0V58JS:~/ca x phj@DESKTOP-H0V58JS:~/ca x + v
phj@DESKTOP-H0V58JS:~/catkin_ws$ rosrun robit_description display.launch |
```

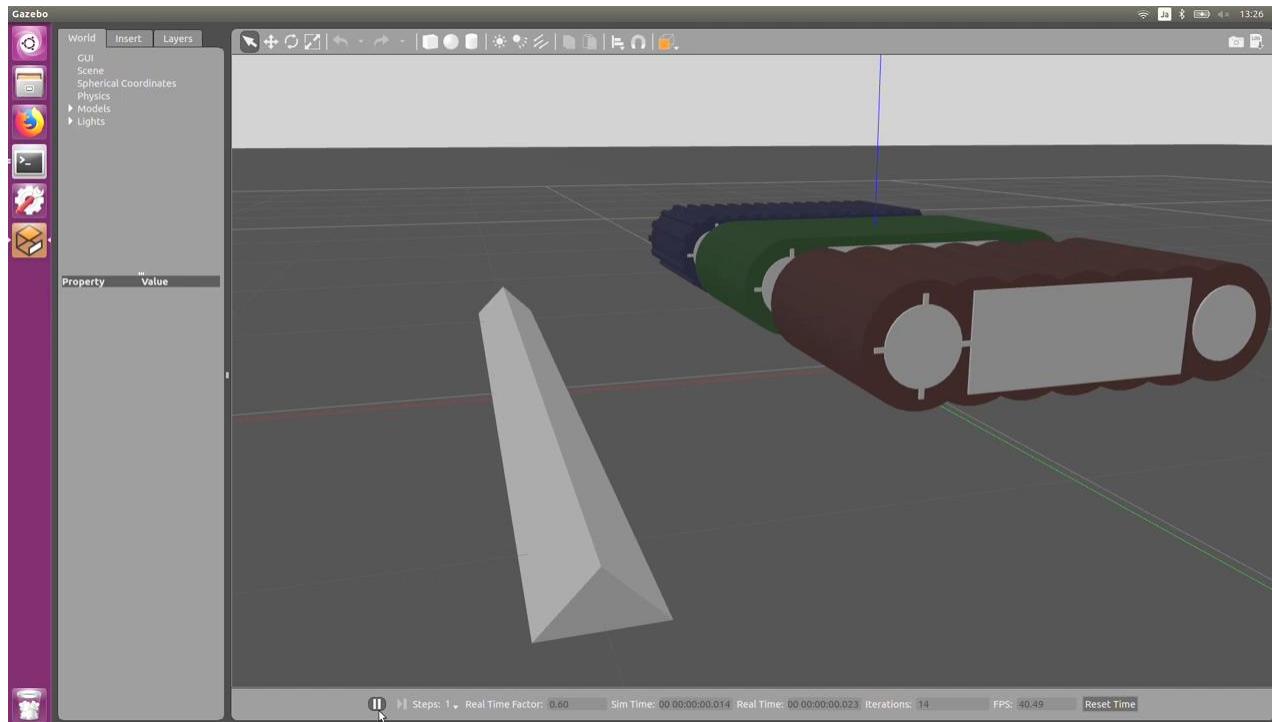
➤ Check your robot

- Edit joint dir <axis xyz="0.0 1.0 0.0"/> or <axis xyz="0.0 -1.0 0.0"/>



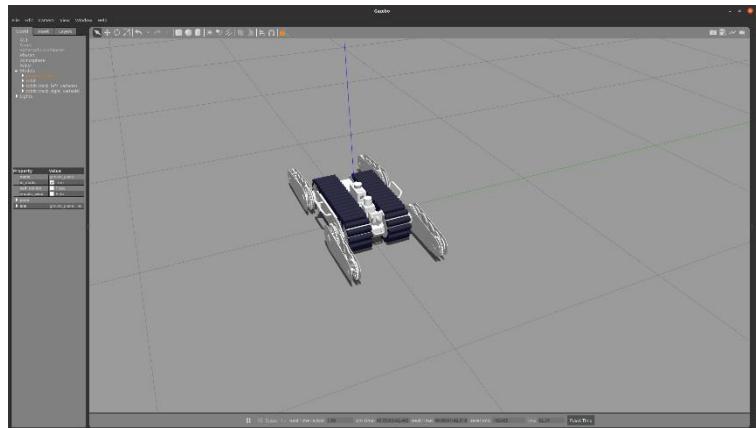
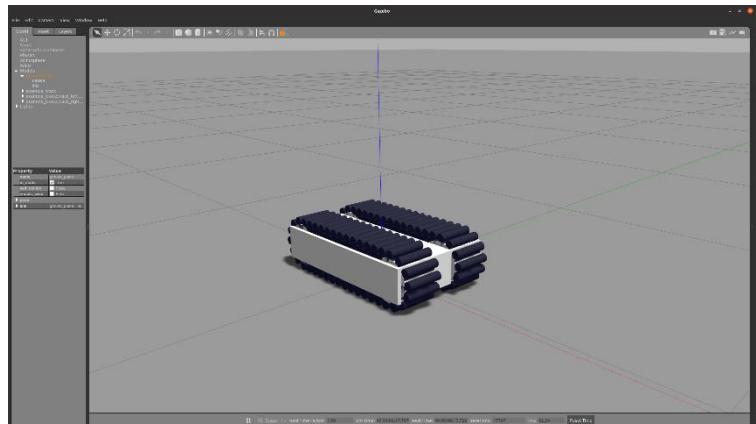
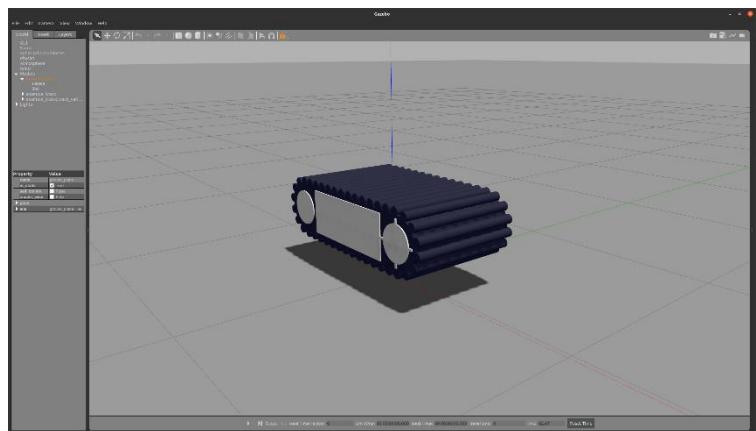
➤ Make track

- cd catkin_ws/src
- git clone https://github.com/phj5024/gazebo_continuous_track
- git clone https://github.com/phj5024/gazebo_continuous_track_example.git
- catkin build or catkin_make
- source ~/.bashrc
- sudo apt install ros-noetic-ros-controllers
- roslaunch gazebo_continuous_track_example example_track_all_world.launch
- rostopic pub /cmd_vel std_msgs/Float64 "data: 0.5"



Track Urdf

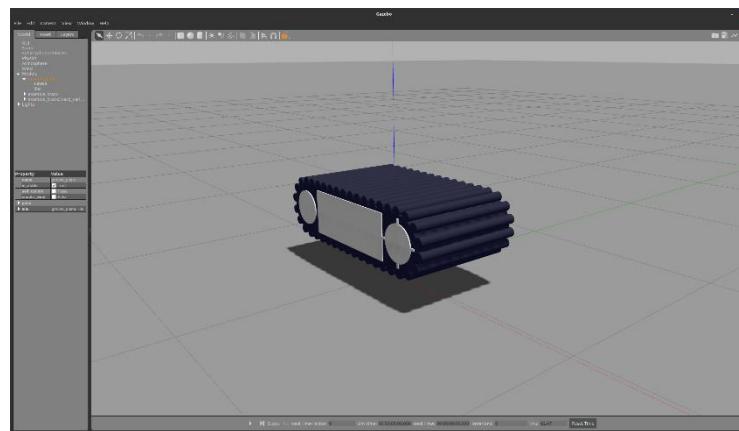
➤ Make track



➤ Make track

- Box element to cylinder
- Open urdf_xacro -> example_track.urdf.xacro
 - body -> base_link
- Open urdf_xacro -> example_track_gazebo.urdf.xacro

```
19   <xacro:make_track name="track" mass="0.4" length="0.7" radius="0.115" width="0.5" parent="base_link" sprocket_joint="sprocket_axle" pitch_diameter="0.24">
20     <xacro:insert_block name="track_material" />
21   <pattern>
22     <elements_per_round>40</elements_per_round>
23     <element>
24       <xacro:make_cylinder_element pose="0 0 0.01 ${pi} / 2 0 0" length="0.5" radius="0.02">
25         <xacro:insert_block name="track_material" />
26       </xacro:make_cylinder_element>
27     </element>
28   </pattern>
29 </xacro:make_track>
```



- rosrun gazebo_continuous_track_example example_track_world.launch
- rostopic pub /example_track/sprocket_velocity_controller/command std_msgs/Float64 "data: 0.5"

➤ Make track

- Left Track
- Open urdf_xacro -> example_track.urdf.xacro
- Measure robot in Fusion 360

```

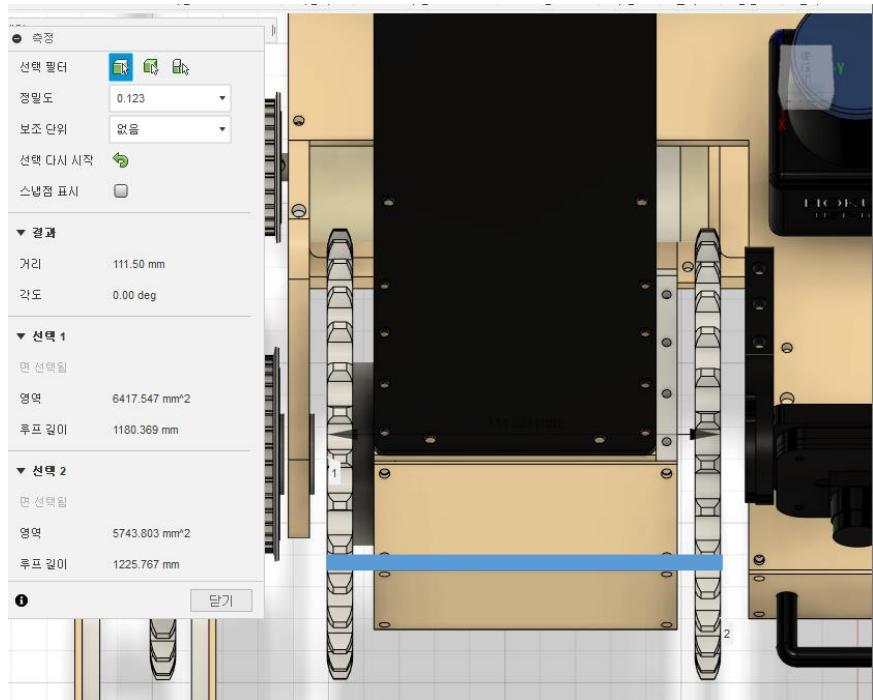
10
11 <xacro:property name="track_width" value="0.1115"/>
12 <xacro:property name="track_radius" value="0.064"/>
13

```

```

29 <!-- base_link to be wrapped by the track -->
30
31 <link name="base_link">
32   <xacro:make_box_inertia mass="1" size_x="0.5" size_y="0.325" size_z="0.1" />
33
34   <visual>
35     <geometry>
36       <box size="0.5 0.325 0.1" />
37     </geometry>
38   </visual>
39
40   <collision>
41     <geometry>
42       <box size="0.5 0.325 0.1" />
43     </geometry>
44   </collision>
45 </link>
46

```



➤ Make track

- Left Track
 - Open urdf_xacro -> example_track.urdf.xacro
 - Change sprocket -> sprocket_left & radius and length
 - Change joint xyz

```
47 <!-- sprocket driving the track -->
48
49 <link name="sprocket_left">
50   <xacro:make_wheel_inertia mass="0.2" radius="${track_radius}" length="${track_width}" />
51
52   <visual>
53     <origin xyz="0 0 0" rpy="${pi / 2} 0 0" />
54     <geometry>
55       <cylinder radius="${track_radius}" length="${track_width}" />
56     </geometry>
57   </visual>
58   <visual>
59     <geometry>
60       <box size="0.01 ${track_width} 0.1" />
61     </geometry>
62   </visual>
63   <visual>
64     <geometry>
65       <box size="0.1 ${track_width} 0.01" />
66     </geometry>
67   </visual>
68
69   <collision>
70     <origin xyz="0 0 0" rpy="${pi / 2} 0 0" />
71     <geometry>
72       <cylinder radius="${track_radius}" length="${track_width}" />
73     </geometry>
74   </collision>
75 </link>
76
77 <joint name="sprocket_left_axle" type="continuous">
78   <origin xyz="0.2 0.095 0" rpy="0 0 0" />
79   <parent link="base_link" />
80   <child link="sprocket_left" />
81   <axis xyz="0 1 0" />
82 </joint>
```

➤ Make track

- Left Track
- Open urdf_xacro -> example_track.urdf.xacro
- Change idler -> idler_left & radius and length
- Change joint xyz

```
94      <!-- idler on the other side of the sprocket -->
95
96      <link name="idler_left">
97          <xacro:make_wheel_inertia mass="0.2" radius="${track_radius}" length="${track_width}" />
98
99          <visual>
100             <origin xyz="0 0 0" rpy="${pi / 2} 0 0" />
101             <geometry>
102                 <cylinder radius="${track_radius}" length="${track_width}" />
103             </geometry>
104         </visual>
105
106         <collision>
107             <origin xyz="0 0 0" rpy="${pi / 2} 0 0" />
108             <geometry>
109                 <cylinder radius="${track_radius}" length="${track_width}" />
110             </geometry>
111         </collision>
112     </link>
113
114     <joint name="idler_left_axle" type="fixed">
115         <origin xyz="-0.2 0.095 0" rpy="0 0 0" />
116         <parent link="base_link" />
117         <child link="idler_left" />
118     </joint>
```

➤ Make track

- Left Track
- Open urdf_xacro -> example_track_gazebo.urdf.xacro
- Change track -> track_left & radius and length
- Add x="0" y="0.095" (track_left pos)

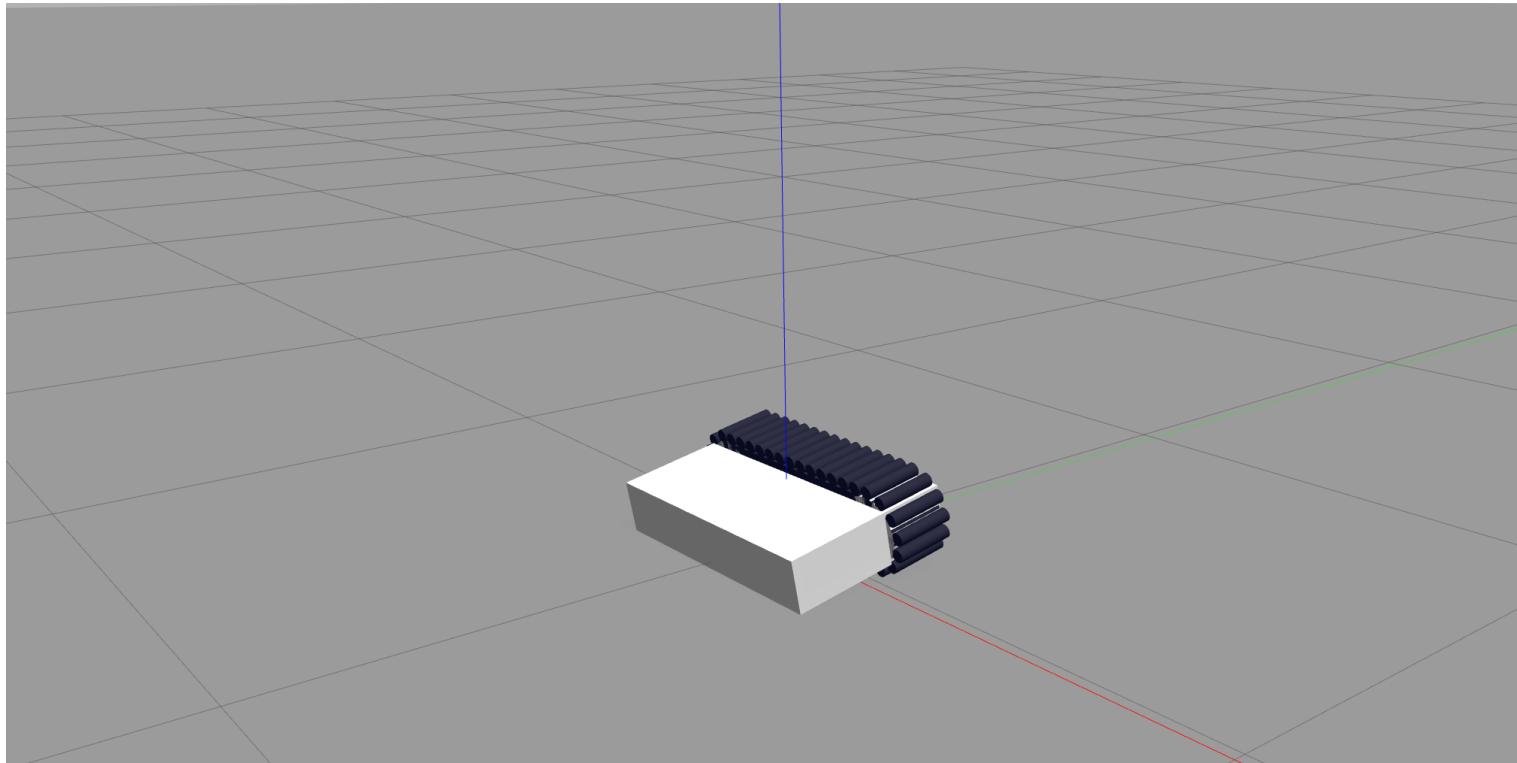
```
19  <xacro:make_track name="track_left" x="0" y="0.095" mass="0.4" length="0.4" radius="${track_radius}" width="${track_width}" parent="base_link" sprocket_joint="sprocket_left_axle" pitch_diameter="0.24">
20    <xacro:insert_block name="track_material" />
21    <pattern>
22      <elements_per_round>40</elements_per_round>
23      <element>
24        <xacro:make_cylinder_element pose="0 0 0.01 ${pi / 2} 0 0" length="${track_width}" radius="0.0125">
25          |   <xacro:insert_block name="track_material" />
26        </xacro:make_cylinder_element>
27      </element>
28    </pattern>
29  </xacro:make_track>
```

- Open launch -> example_track_world.launch
- Change sprocket -> sprocket_left

```
34  <!-- Load the URDF into the ROS Parameter Server -->
35  <param name="robot_description" command="xacro --inorder '$(find gazebo_continuous_track_example)/urdf_xacro/example_track.urdf.xacro'" />
36
37  <rosparam>
38    joint_state_controller:
39      type: joint_state_controller/JointStateController
40      publish_rate: 10
41    sprocket_left_velocity_controller:
42      type: velocity_controllers/JointVelocityController
43      joint: sprocket_left_axle
44  </rosparam>
45
46  <!-- load and start the controllers -->
47  <node name="controller_starter" pkg="controller_manager" type="controller_manager" args="spawn joint_state_controller sprocket_left_velocity_controller" output="screen" />
```

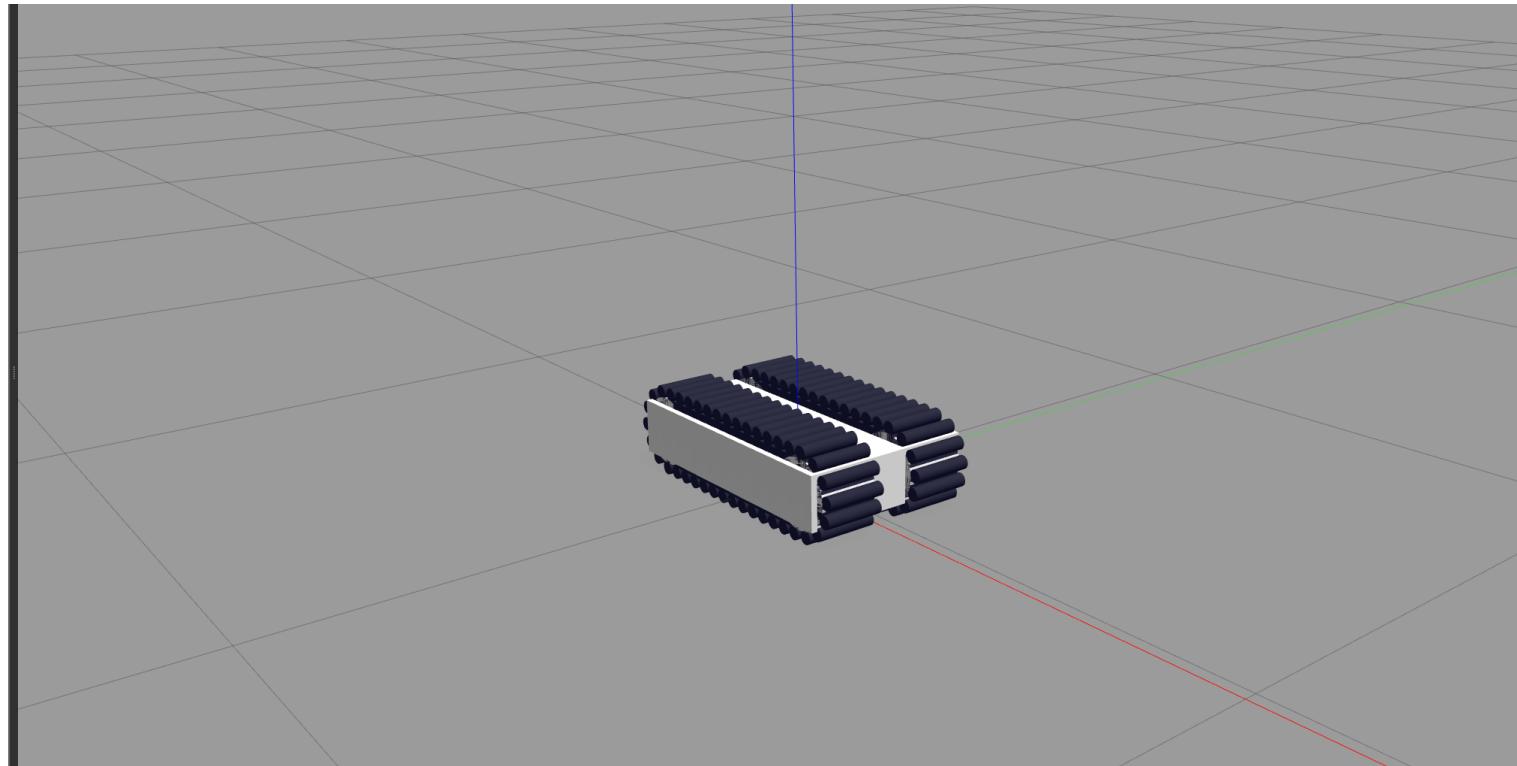
➤ Make track

- catkin build or catkin_make
- roslaunch gazebo_continuous_track_example example_track_world.launch
- rostopic pub /example_track/sprocket_left_velocity_controller/command std_msgs/Float64 "data: 0.5"



➤ Make left track & right track

- rostopic pub /example_track/sprocket_left_velocity_controller/command std_msgs/Float64 "data: 0.5"
- rostopic pub /example_track/sprocket_right_velocity_controller/command std_msgs/Float64 "data: 0.5"



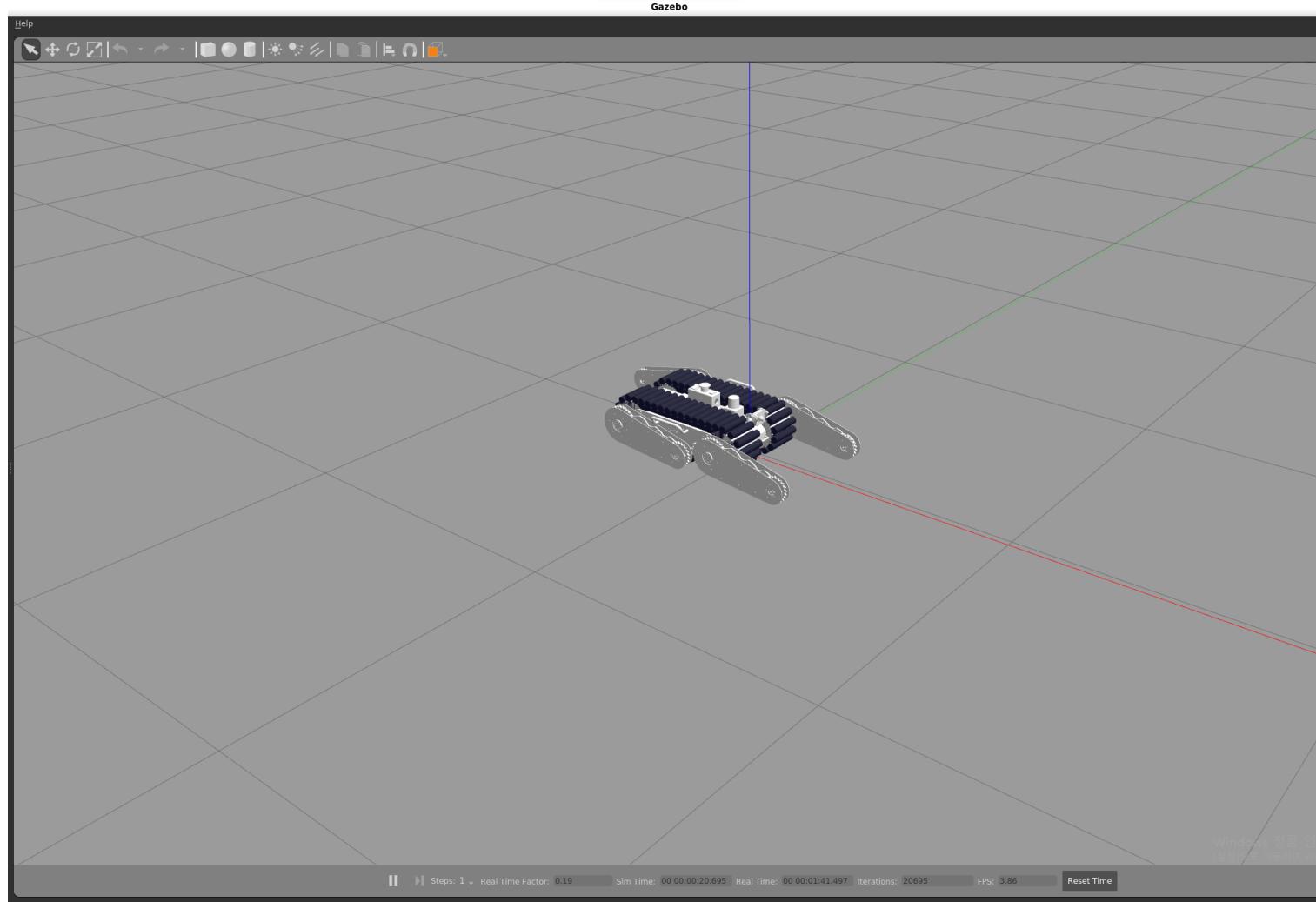
➤ Track Urdf to Custom Robot

- CMakeLists.txt generate urdf & sdf model for gazebo
- We copy urdf code to robit.xacro, robit.trans, robit.gazebo
- Open gazebo_continuous_track_example/urdf/example_track_gazebo.urdf
- Copy script to robit.xacro & remove duplicated part (ex. base_link, <robot>, ...)
- controller.yaml robit_controller -> #robit_controller & add sprocket controller
- controller.launch remove ns="robit" & add args="**sprocket_left_velocity_controller**
sprocket_left_velocity_controller ... "

```
1 #robit_controller:
2   # Publish all joint states -----
3   joint_state_controller:
4     type: joint_state_controller/JointStateController
5     publish_rate: 50
6
7     1   <launch>
8       # Position Cont 2
9       flipper_joint_FL 3   <rosparam file="$(find robit_description)/launch/controller.yaml" command="load"/>
10      type: effort_ 4   <node name="controller_spawner" pkg="controller_manager" type="spawner" respawn="false" output="screen" args="sprocket_left_velocity_controller sprocket_right_velocity_controller">
11      joint: flippe 5   <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher" respawn="false" output="screen">
12      pid: {p: 100.0 6   |   <remap from="/joint_states" to="/robit/joint_states"/>
13      flipper_joint_FL 7   </node>
14      type: effort_ 8   </launch>
15      joint: flipper_joint_FR
16      pid: {p: 100.0, i: 0.01, d: 10.0}
17      flipper_joint_BL_position_controller:
18        type: effort_controllers/JointPositionController
19        joint: flipper_joint_BL
20        pid: {p: 100.0, i: 0.01, d: 10.0}
21      flipper_joint_BR_position_controller:
22        type: effort_controllers/JointPositionController
23        joint: flipper_joint_BR
24        pid: {p: 100.0, i: 0.01, d: 10.0}
25
26      sprocket_left_velocity_controller:
27        type: velocity_controllers/JointVelocityController
28        joint: sprocket_left_axle
29      sprocket_right_velocity_controller:
30        type: velocity_controllers/JointVelocityController
31        joint: sprocket_right_axle
```

➤ Track Urdf to Custom Robot

- rosrun robot_description gazebo.launch
- Real Time Factor is remarkably low -> collision



Track Urdf

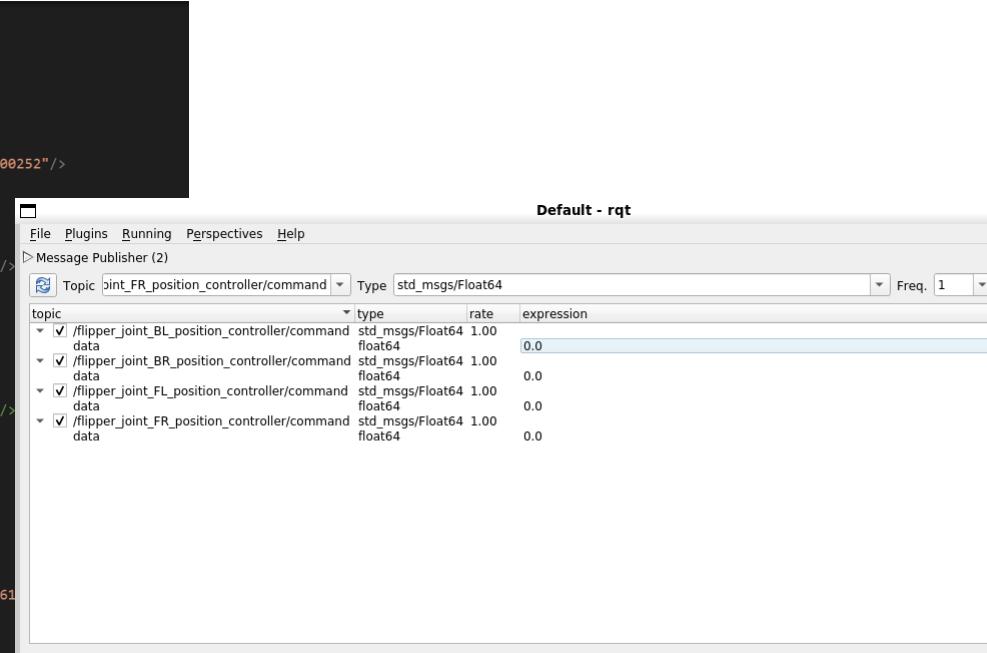
➤ Track Urdf to Custom Robot

- Comment collision base_link and flipper(4)
- rosrun robit_description gazebo.launch
- rosrun robit_description controller.launch
- rqt -> plugins -> Topics -> Message Publisher

```

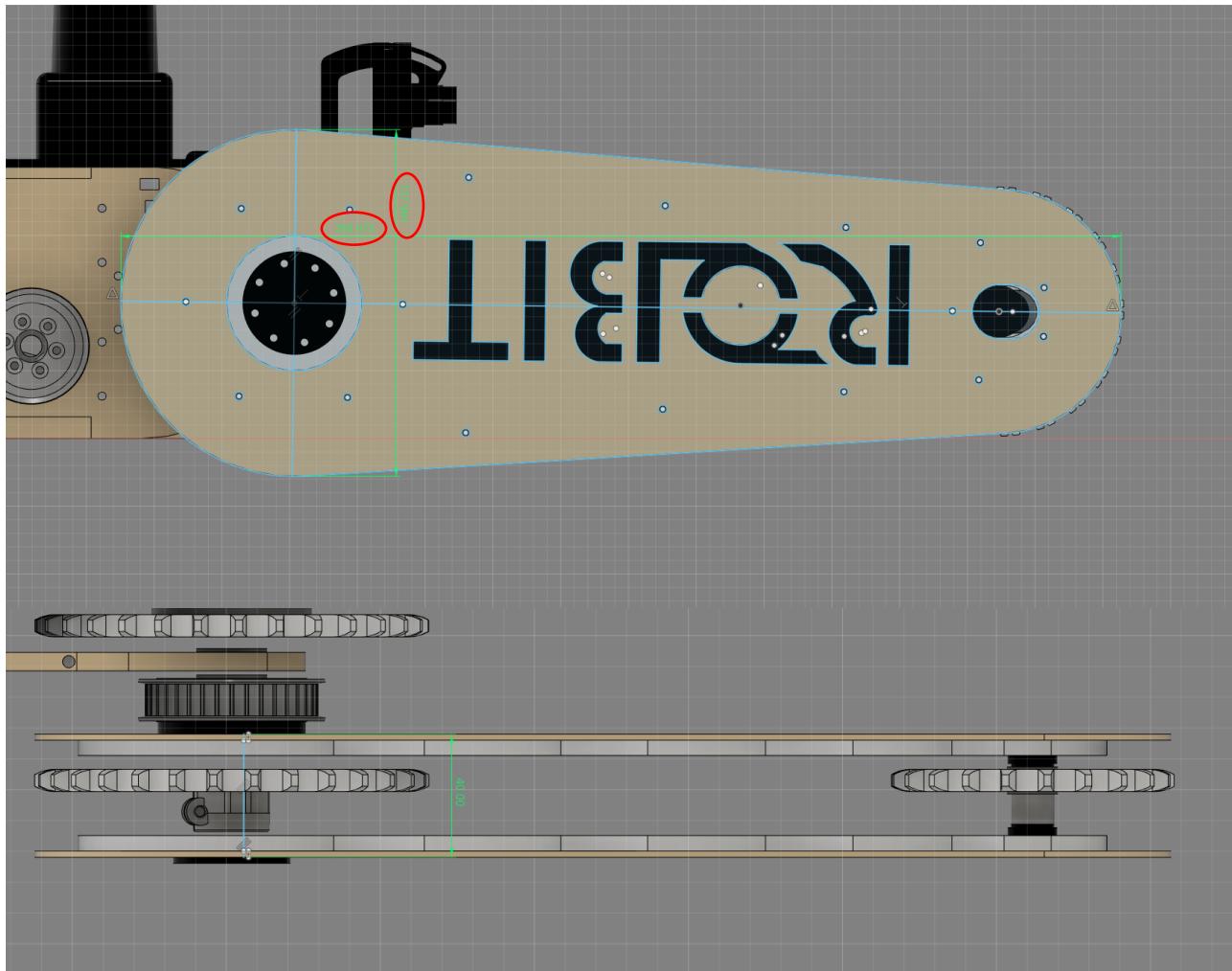
4 <xacro:include filename="$(find robit_description)/urdf/materials.xacro" />
5 <xacro:include filename="$(find robit_description)/urdf/robit.trans" />
6 <xacro:include filename="$(find robit_description)/urdf/robit.gazebo" />
7 <link name="base_link">
8   <inertial>
9     <origin xyz="0.005359823158541233 -0.008604658420625445 0.053376909536265" rpy="0 0 0"/>
10    <mass value="6.517447775880915"/>
11    <inertia ixx="0.000945" iyy="0.157265" izz="0.215644" ixy="0.000214" iyz="-0.000126" ixz="-0.00252"/>
12  </inertial>
13  <visual>
14    <origin xyz="0 0 -0.05" rpy="0 0 0"/>
15    <geometry>
16      <mesh filename="package://robit_description/meshes/base_link.stl" scale="0.001 0.001 0.001"/>
17    </geometry>
18    <material name="silver"/>
19  </visual>
20  <!-- <collision>
21    <origin xyz="0 0 -0.05" rpy="0 0 0"/>
22    <geometry>
23      <mesh filename="package://robit_description/meshes/base_link.stl" scale="0.001 0.001 0.001"/>
24    </geometry>
25  </collision> -->
26 </link>
27
28 <link name="flipper_FL">
29   <inertial>
30     <origin xyz="0.08395375283162221 0.051302889963027565 -0.0010078517627294392" rpy="0 0 0"/>
31     <mass value="1.2818842189231356"/>
32     <inertia ixx="0.001659" iyy="0.015451" izz="0.015081" ixy="-0.000706" iyz="9e-06" ixz="0.000161">
33   </inertial>
34   <visual>
35     <origin xyz="-0.201729 -0.14315 -0.05" rpy="0 0 0"/>
36     <geometry>
37       <mesh filename="package://robit_description/meshes/flipper_FL.stl" scale="0.001 0.001 0.001"/>
38     </geometry>
39     <material name="silver"/>
40   </visual>
41   <!-- <collision>
42     <origin xyz="-0.201729 -0.14315 -0.05" rpy="0 0 0"/>
43     <geometry>
44       <mesh filename="package://robit_description/meshes/flipper_FL.stl" scale="0.001 0.001 0.001"/>
45     </geometry>
46   </collision> -->
47 </link>
48

```



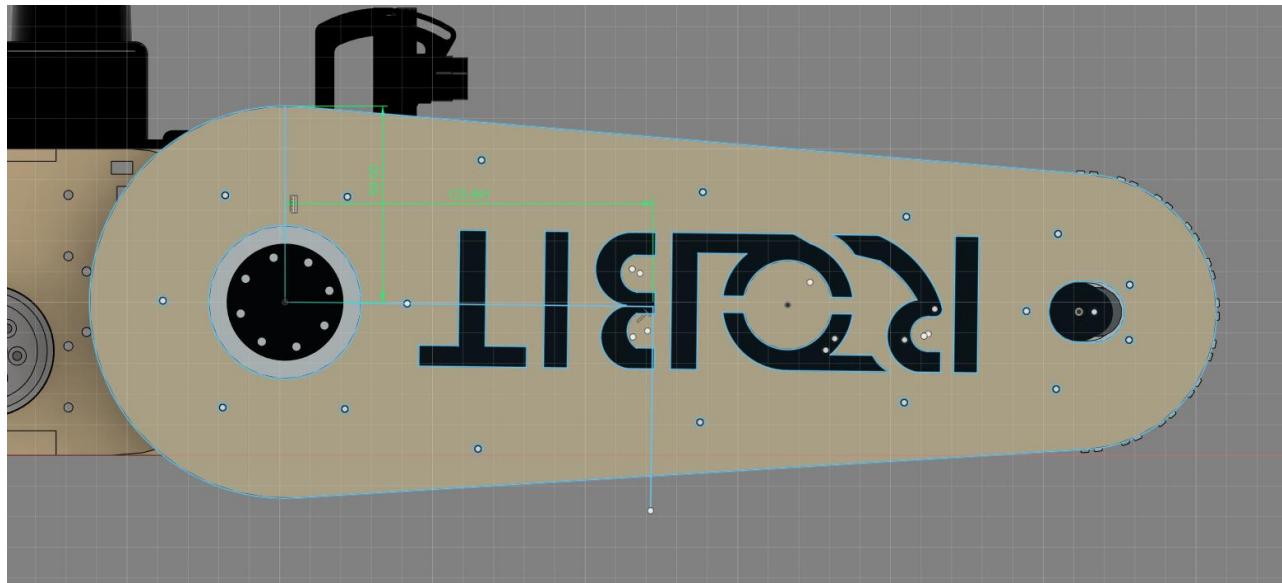
Track Urdf

➤ Get Collision model



1. 단순화 작업 전 원하는 프레임의 크기를 측정($L \times W \times H$)
Ex_(369 x 128 x 40)

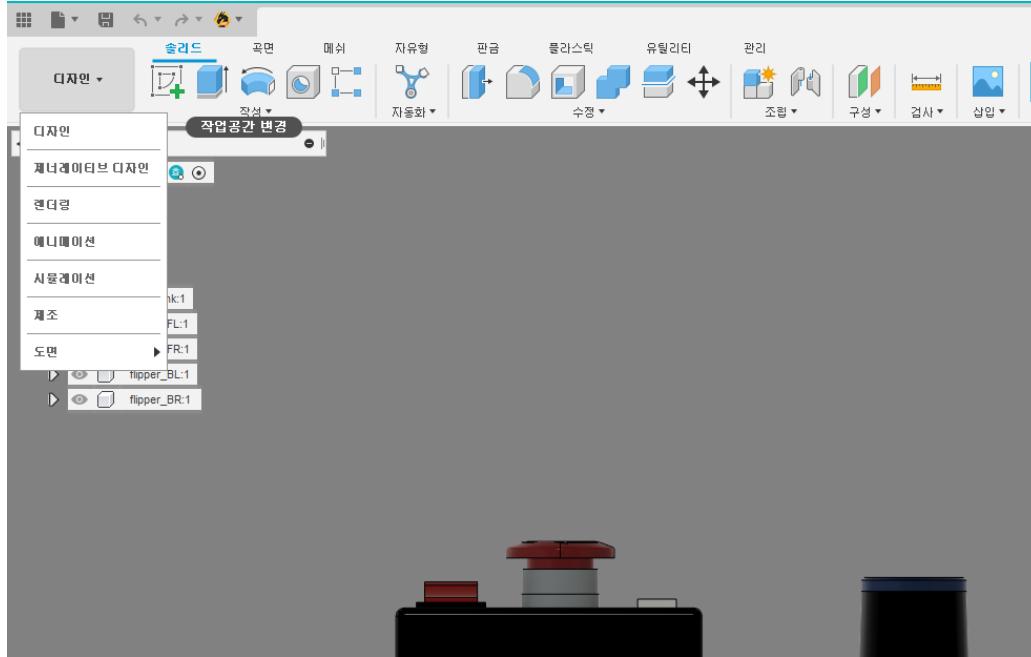
- Get Collision model



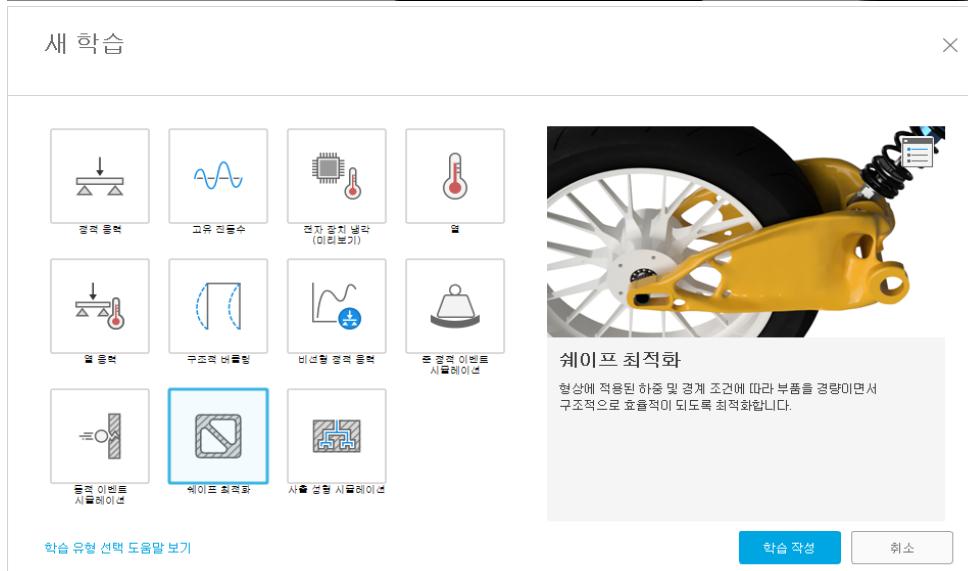
2. 단순화 작업의 박스는 대상 프레임의 중앙에 생성되므로 프레임 중앙에서 Joint까지의 거리 측정

Ex_ (120.5 x 64)

➤ Get Collision model

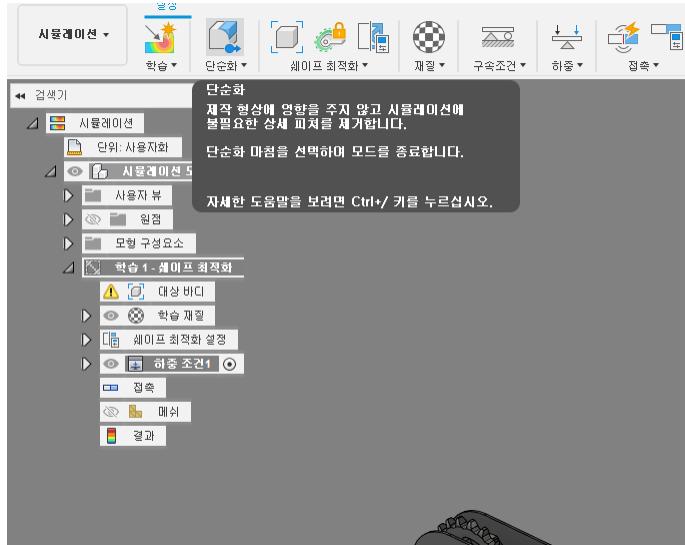


3. 디자인 탭 클릭 후, 시뮬레이션 클릭

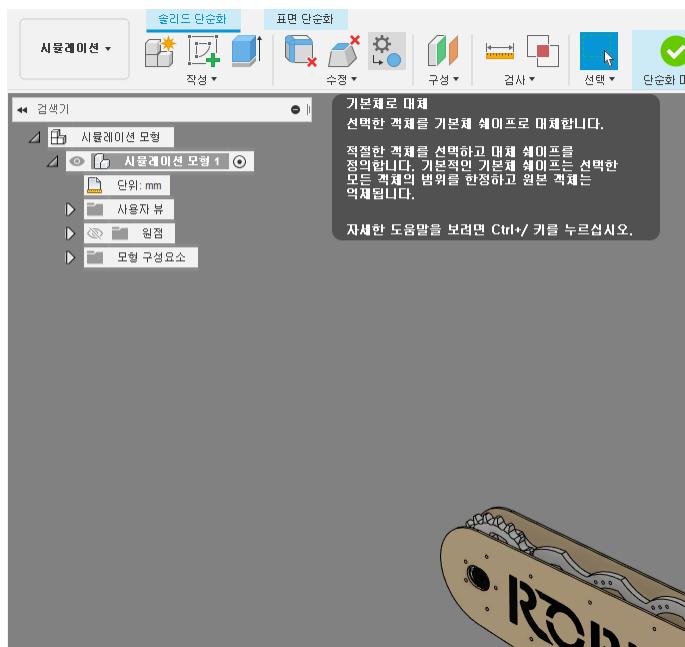


4. 아무거나 선택(처음에 크래딧 결제 관련 내용이 나올 수 있는데 무시하고 진행)

➤ Get Collision model



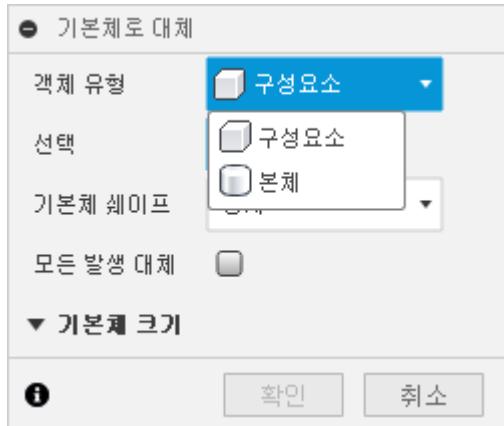
5. 단순화 클릭



6. 기본체로 대체 클릭

Track Urdf

➤ Get Collision model

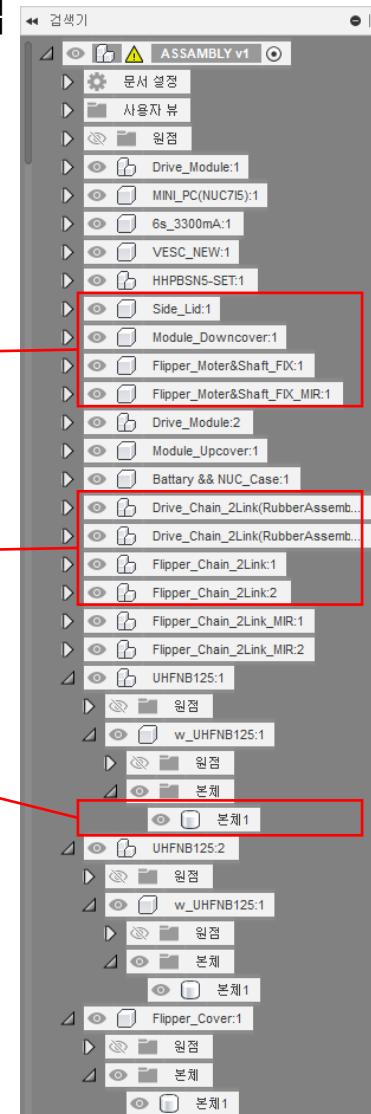


7. 다음과 같은 화면에서 본체, 구성요소 선택
 구성요소 – 여러 개의 프레임을 한번에 선택
 본체 – 단일 프레임 하나만 선택

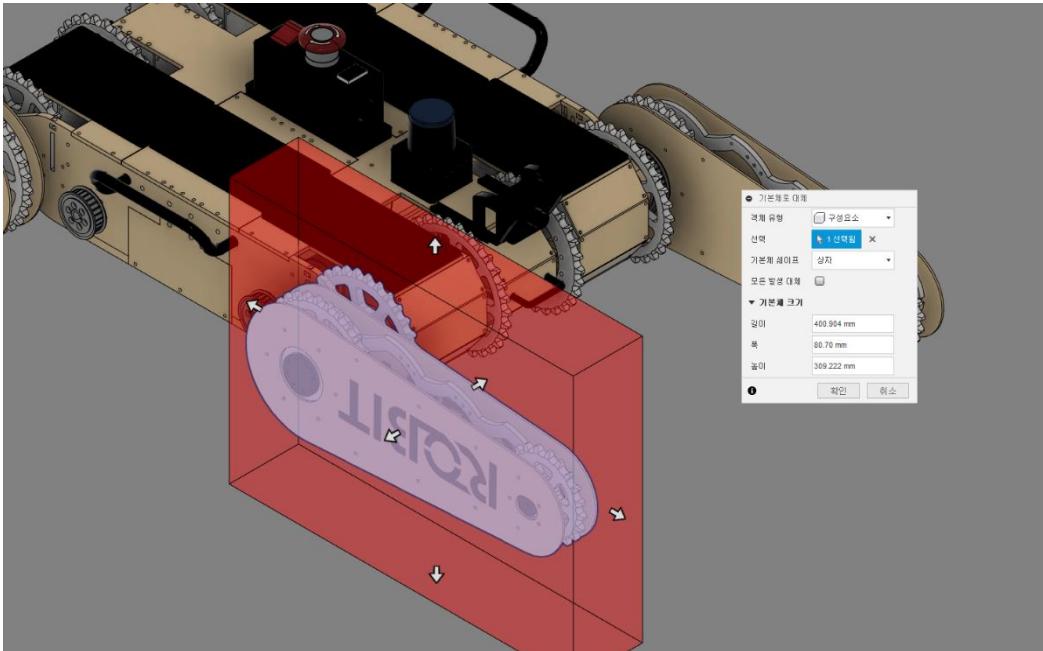
단일 구성요소(본체 1개)

구성요소 안에 1개 이상의 구성요소
 (본체 1개 이상)

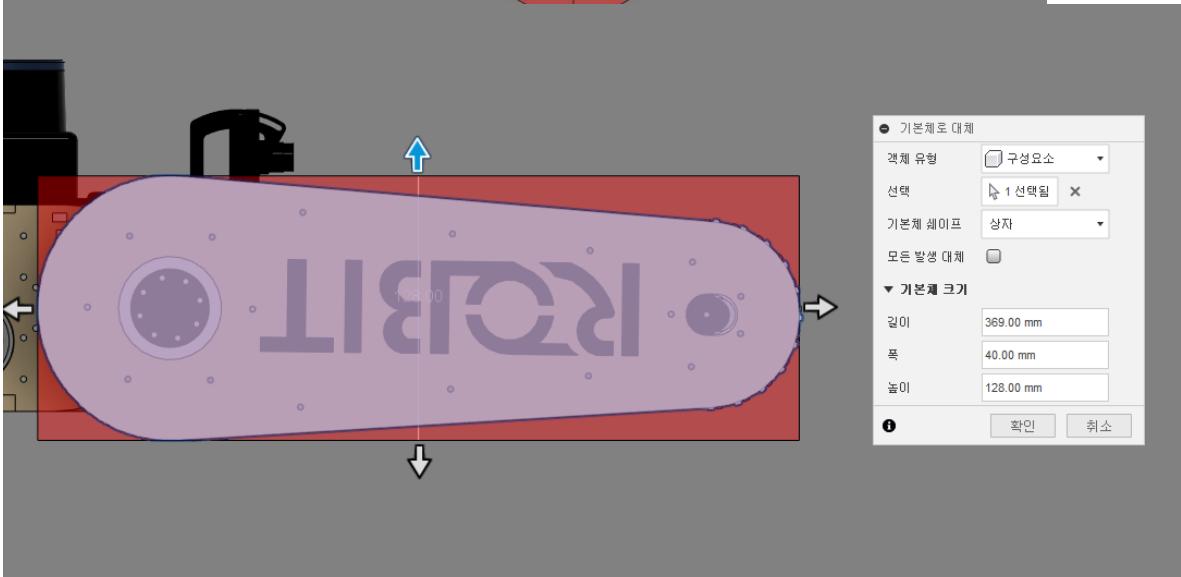
본체



➤ Get Collision model

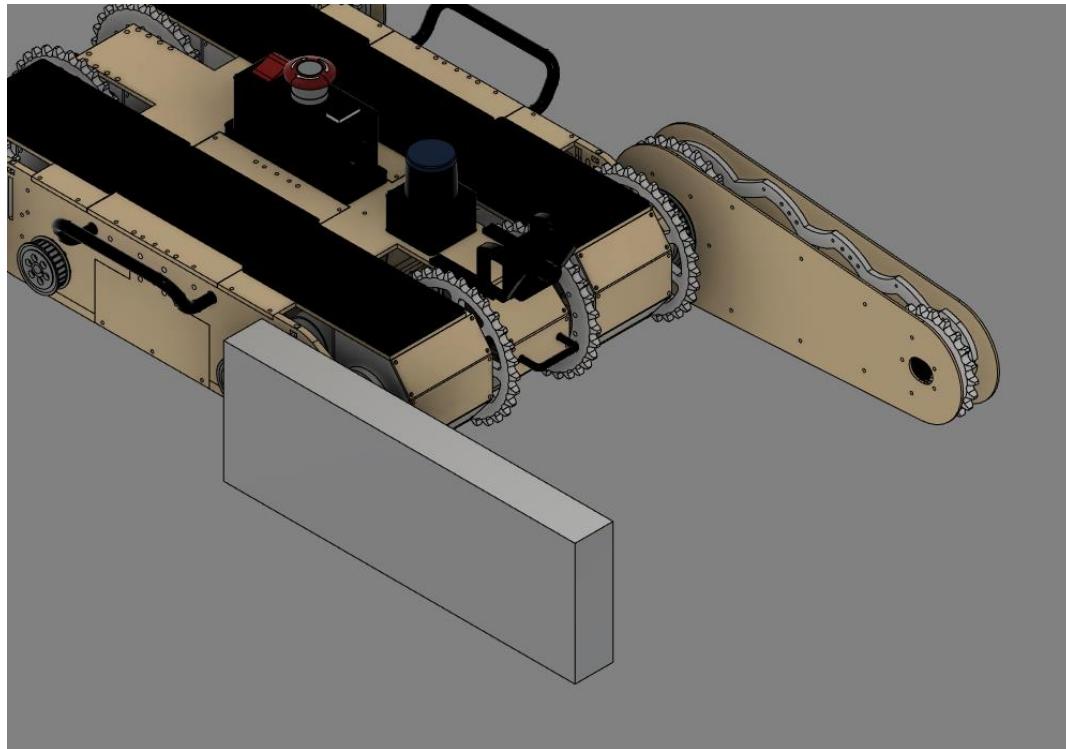


8. Ex) 구성요소 flipper_FR 선택 후
길이, 폭, 높이 조절



9. 조절 후 확인 클릭

- Get Collision model



10. 369 * 40 * 128

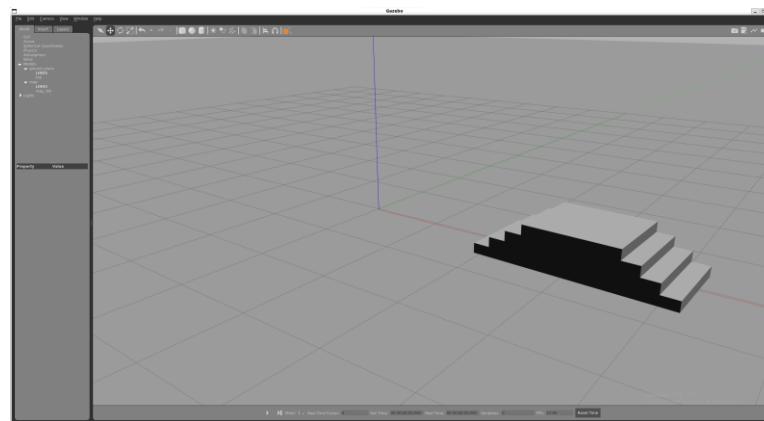
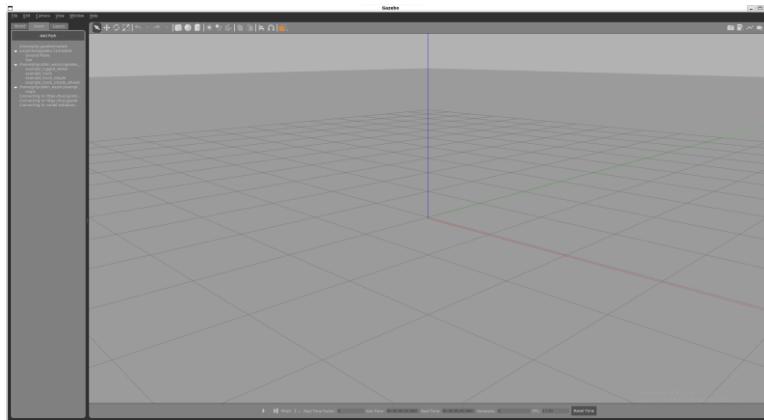
- Set Collision model (urdf/robit.xacro) -> x y z pos (p.29)
 - Change spawn pos (launch/gazebo.launch)

```
28 <link name="flipper_FL">
29   <inertial>
30     <origin xyz="-0.08395375283162221 0.051302889963027565 -0.0010078517627294392" rpy="0 0 0"/>
31     <mass value="1.2818842189231356"/>
32     <inertia ixx=".001659" iyy=".015451" izz=".015081" ixz="-0.000706" iyz="9e-06" ixz=".000161"/>
33   </inertial>
34   <visual>
35     <origin xyz="-0.201729 -0.14315 -0.05" rpy="0 0 0"/>
36     <geometry>
37       <mesh filename="package://robit_description/meshes/flipper_FL.stl" scale="0.001 0.001 0.001"/>
38     </geometry>
39     <material name="silver"/>
40   </visual>
41   <collision>
42     <origin xyz="0.120 0.064 0" rpy="0 0 0"/>
43     <geometry>
44       <box size="0.369 0.040 0.128"/>
45     </geometry>
46   </collision>
47 </link>
48
```

```
1 <launch>
2   <arg name="x" default="0.0"/>
3   <arg name="y" default="0.0"/>
4   <arg name="z" default="0.2"/>
5   <arg name="yaw" default="0.0"/>
6   <!-- command-line arguments -->
7   <rde name="spawn_urdf" pkg="gazebo_ros" type="spawn_model" args="-x $(arg x) -y $(arg y) -z $(arg z) -Y $(arg yaw) -param robot_description -urdf -model robit"/>
8   <!-- gazebo configuration -->
9   <arg name="paused" value="true"/>
10  <arg name="use_sim_time" value="true"/>
11  <arg name="gui" value="true"/>
12  <arg name="headless" value="false"/>
13  <arg name="debug" value="false"/>
14  </include>
15 </launch>
16
```

Custom map

- git clone https://github.com/phj5024/example_gazebo_map.git
- roslaunch gazebo_ros empty_world.launch
- Insert map1 & Save World as -> example_gazebo_map/worlds/map.world



- Add another model ->
- Copy example folder in models ->
- Change folder name ->
- Change model name, mesh in model.sdf & model.config

Custom map

- Make spawn_robot.launch in example_gazebo_map/launch

```

1  <?xml version="1.0"?>
2
3  <launch>
4      <arg name="x" default="0.0"/>
5      <arg name="y" default="0.0"/>
6      <arg name="z" default="0.2"/>
7      <arg name="yaw" default="0.0"/>
8      <param name="robot_description" command="$(find xacro)/xacro $(find robit_description)/urdf/robit.xacro"/>
9      <node name="spawn_urdf" pkg="gazebo_ros" type="spawn_model" args="-x $(arg x) -y $(arg y) -z $(arg z) -Y $(arg yaw) -param robot_description -urdf -model robit"/>
10
11     <include file="$(find robit_description)/launch/controller.launch" />
12 </launch>

```

- Make gazebo.launch in example_gazebo_map/launch

```

1  <?xml version="1.0"?>
2
3  <launch>
4      <include file="$(find example_gazebo_map)/launch/map.launch" />
5      <include file="$(find example_gazebo_map)/launch/spawn_robot.launch" />
6  </launch>
7

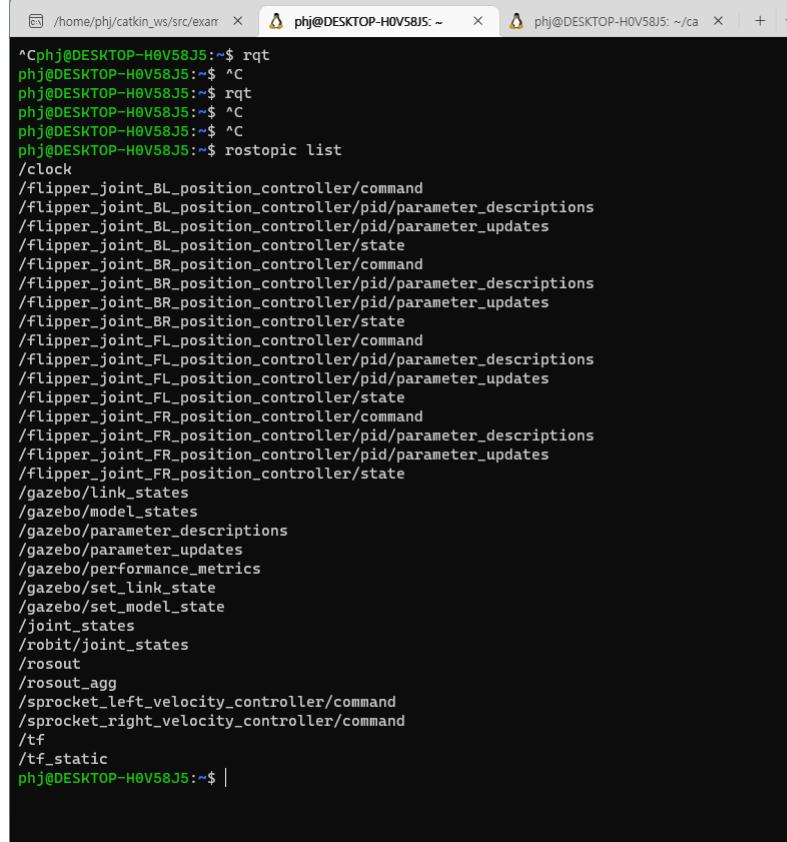
```

- rosrun example_gazebo_map gazebo.launch

Simple Joy

- controller.yaml -> too many variables.
- Same controller type creates a group

```
#robit_controller:  
# Publish all joint states -----  
joint_state_controller:  
  type: joint_state_controller/JointStateController  
  publish_rate: 50  
  
# Position Controllers -----  
flipper_joint_FL_position_controller:  
  type: effort_controllers/JointPositionController  
  joint: flipper_joint_FL  
  pid: {p: 100.0, i: 0.01, d: 10.0}  
flipper_joint_FR_position_controller:  
  type: effort_controllers/JointPositionController  
  joint: flipper_joint_FR  
  pid: {p: 100.0, i: 0.01, d: 10.0}  
flipper_joint_BL_position_controller:  
  type: effort_controllers/JointPositionController  
  joint: flipper_joint_BL  
  pid: {p: 100.0, i: 0.01, d: 10.0}  
flipper_joint_BR_position_controller:  
  type: effort_controllers/JointPositionController  
  joint: flipper_joint_BR  
  pid: {p: 100.0, i: 0.01, d: 10.0}  
  
sprocket_left_velocity_controller:  
  type: velocity_controllers/JointVelocityController  
  joint: sprocket_left_axle  
sprocket_right_velocity_controller:  
  type: velocity_controllers/JointVelocityController  
  joint: sprocket_right_axle
```



A terminal window showing the output of the command `rostopic list`. The output lists numerous topics, primarily related to joint controllers and gazebo states. The topics include: /clock, /flipper_joint_BL_position_controller/command, /flipper_joint_BL_position_controller/pid/parameter_descriptions, /flipper_joint_BL_position_controller/pid/parameter_updates, /flipper_joint_BL_position_controller/state, /flipper_joint_BR_position_controller/command, /flipper_joint_BR_position_controller/pid/parameter_descriptions, /flipper_joint_BR_position_controller/pid/parameter_updates, /flipper_joint_BR_position_controller/state, /flipper_joint_FL_position_controller/command, /flipper_joint_FL_position_controller/pid/parameter_descriptions, /flipper_joint_FL_position_controller/pid/parameter_updates, /flipper_joint_FL_position_controller/state, /flipper_joint_FR_position_controller/command, /flipper_joint_FR_position_controller/pid/parameter_descriptions, /flipper_joint_FR_position_controller/pid/parameter_updates, /flipper_joint_FR_position_controller/state, /gazebo/link_states, /gazebo/model_states, /gazebo/parameter_descriptions, /gazebo/parameter_updates, /gazebo/performance_metrics, /gazebo/set_link_state, /gazebo/set_model_state, /joint_states, /robit/joint_states, /rosout, /rosout_agg, /sprocket_left_velocity_controller/command, /sprocket_right_velocity_controller/command, /tf, /tf_static.

```
#robit_controller:

# Publish all joint states -----
joint_state_controller:
  type: joint_state_controller/JointStateController
  publish_rate: 50

# Position Controllers -----
flipper_position_controller:
  type: effort_controllers/JointGroupPositionController
  joints:
    - flipper_joint_FL
    - flipper_joint_FR
    - flipper_joint_BL
    - flipper_joint_BR
  flipper_joint_FL:
    pid: {p: 100.0, i: 0.01, d: 10.0}
  flipper_joint_FR:
    pid: {p: 100.0, i: 0.01, d: 10.0}
  flipper_joint_BL:
    pid: {p: 100.0, i: 0.01, d: 10.0}
  flipper_joint_BR:
    pid: {p: 100.0, i: 0.01, d: 10.0}

# Velocity Controllers -----
sprocket_velocity_controller:
  type: velocity_controllers/JointGroupVelocityController
  joints:
    - sprocket_left_axle
    - sprocket_right_axle
  sprocket_left_axle:
    pid: {p: 100.0, i: 0.01, d: 10.0}
  sprocket_right_axle:
    pid: {p: 100.0, i: 0.01, d: 10.0}
```

- JointGroupPositionController & JointGroupVelocityController

- Change args in controller.launch

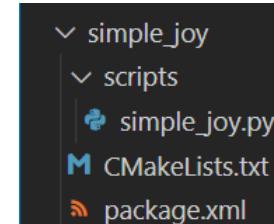
- roslaunch example_gazebo_map gazebo.launch

```
phj@DESKTOP-HOV5835:~$ rostopic list
/clock
/flipper_position_controller/command
/flipper_position_controller/flipper_joint_BL/pid/parameter_descriptions
/flipper_position_controller/flipper_joint_BL/pid/parameter_updates
/flipper_position_controller/flipper_joint_BR/pid/parameter_descriptions
/flipper_position_controller/flipper_joint_BR/pid/parameter_updates
/flipper_position_controller/flipper_joint_FL/pid/parameter_descriptions
/flipper_position_controller/flipper_joint_FL/pid/parameter_updates
/flipper_position_controller/flipper_joint_FR/pid/parameter_descriptions
/flipper_position_controller/flipper_joint_FR/pid/parameter_updates
/gazebo/link_states
/gazebo/model_states
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/performance_metrics
/gazebo/set_link_state
/gazebo/set_model_state
/joint_states
/robit/joint_states
/rosout
/rosout_agg
/sprocket_velocity_controller/command
/tf_static
```

```
_description> launch> --> controller.launch
<launch>

  <node name="controller_spawner" pkg="controller_manager" type="spawner" respawn="false" output="screen" args="flipper_position_controller sprocket_velocity_controller joint_state_controller "/>
  <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher" respawn="false" output="screen">
    <remap from="/joint_states" to="/robit/joint_states"/>
  </node>
</launch>
```

- cd ~/catkin_ws/src
- catkin_create_pkg simple_joy
- Create scripts folder & simple_joy.py



- sudo apt install ros-noetic-teleop-twist-keyboard
 - cd simple_joy/scripts
 - chmod a+x simple_joy.py
 - catkin build or catkin_make
-
- roslaunch example_gazebo_map gazebo.launch
 - rosrun teleop_twist_keyboard teleop_twist_keyboard.py

```
simple_joy > scripts > simple_joy.py
 1  #!/usr/bin/env python3
 2  import rospy
 3  from geometry_msgs.msg import Twist
 4  from std_msgs.msg import Float64MultiArray
 5
 6  def cmd_vel_callback(data, pub):
 7      # This function is called whenever a new message is received on the cmd_vel topic
 8      rospy.loginfo("Received cmd_vel command: linear_x = %f, angular_z = %f", data.linear.x, data.angular.z)
 9
10     # Create a Float64MultiArray message and fill it with some data
11     msg = Float64MultiArray()
12     msg.data = [data.linear.x, data.linear.x]
13
14     pub.publish(msg) # Publish the message
15
16 def simple_joy():
17     rospy.init_node('simple_joy') # Initialize the node
18     rospy.loginfo("Start simple_joy")
19     pub = rospy.Publisher('sprocket_velocity_controller/command', Float64MultiArray, queue_size=10) # Create a publisher for the float64_multi_array_topic
20     sub = rospy.Subscriber('cmd_vel', Twist, cmd_vel_callback, pub) # Subscribe to the cmd_vel topic, passing the publisher object as an argument
21     rospy.spin() # Keep the node running until it is shut down
22
23 if __name__ == '__main__':
24     simple_joy()
```

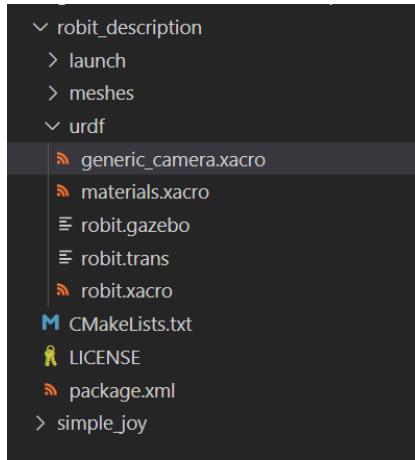
➤ rosrun simple_joy simple_joy.py

- If 'python3\r' : No such file or directory
- sudo apt install dos2unix
- Dos2unix simple_joy.py

➤ What should we do to make a rotation?

Add Camera

- Create generic_camera.xacro in robit_description/urdf



- Change <far>100</far>

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

  <xacro:property name="M_PI" value="3.1415926535897931" />

  <xacro:macro name="generic_camera" params="name parent *origin ros_topic cam_info_topic update_rate
res_x res_y image_format hfov">
    <joint name="${name}_joint" type="fixed">
      <xacro:insert_block name="origin" />
      <parent link="${parent}" />
      <child link="${name}_link"/>
    </joint>

    <link name="${name}_link">
      <inertial>
        <mass value="0.001" />
        <origin xyz="0 0 0" rpy="0 0 0" />
        <inertia ixx="0.0001" ixy="0" ixz="0" iyy="0.0001" iyz="0" izz="0.0001" />
      </inertial>
      <visual>
        <origin xyz="0 0 0" rpy="0 0 0" />
        <geometry>
          <box size="0.01 0.01 0.01" />
        </geometry>
        <material name="Blue">
          <color rgba="0.0 0.0 0.8 1"/>
        </material>
      </visual>
      <collision>
        <origin xyz="0 0 0" rpy="0 0 0" />
        <geometry>
          <box size="0.01 0.01 0.01" />
        </geometry>
      </collision>
    </link>

    <joint name="${name}_optical_joint" type="fixed">
      <origin xyz="0 0 0" rpy="${-M_PI/2} 0.0 ${-M_PI/2}" />
      <parent link="${name}_link" />
      <child link="${name}_optical_frame"/>
    </joint>

    <link name="${name}_optical_frame"/>

    <gazebo reference="${name}_link">
      <sensor type="camera" name="${name}_camera_sensor">
        <update_rate>$update_rate</update_rate>
        <camera>
          <horizontal_fov>$hfov * M_PI/180.0</horizontal_fov>
          <image>
            <format>$image_format</format>
            <width>$res_x</width>
            <height>$res_y</height>
          </image>
          <clip>
            <near>0.01</near>
            <far>100</far>
          </clip>
        </camera>
      <plugin name="${name}_camera_controller" filename="libgazebo_ros_camera.so">
        <cameraName>${name}</cameraName>
        <imageTopicName>${ros_topic}</imageTopicName>
        <cameraInfoTopicName>${cam_info_topic}</cameraInfoTopicName>
        <frameName>${name}_optical_frame</frameName>
      </plugin>
    </sensor>
  </gazebo>
</xacro:macro>

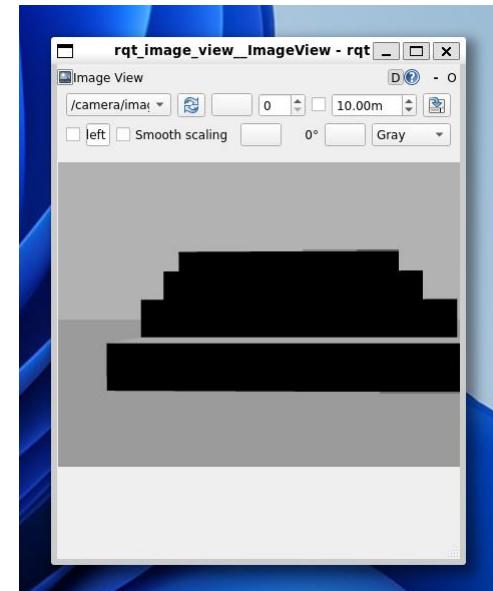
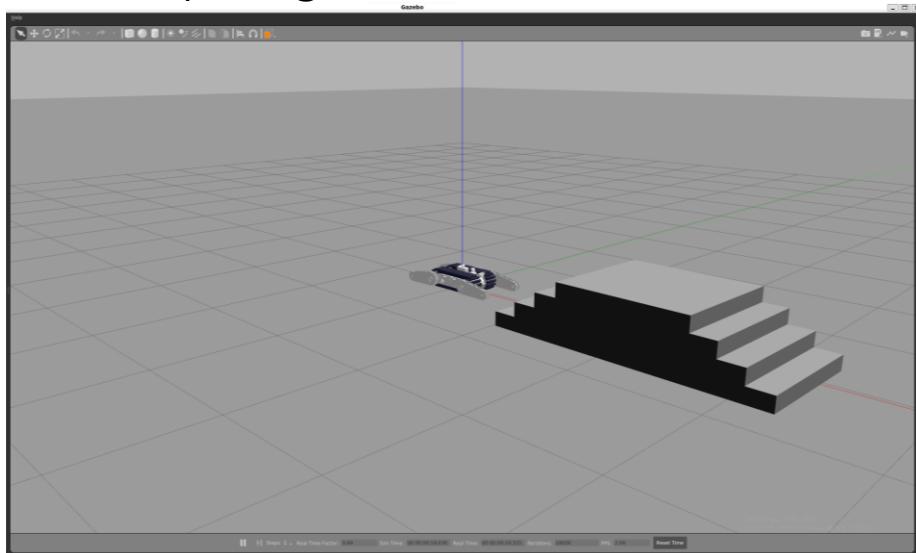
</robot>
```

Add Camera

- Edit robbit.xacro

```
5
4   <xacro:include filename="$(find robbit_description)/urdf/materials.xacro" />
5   <xacro:include filename="$(find robbit_description)/urdf/robbit.trans" />
6   <xacro:include filename="$(find robbit_description)/urdf/robbit.gazebo" />
7   <xacro:include filename="$(find robbit_description)/urdf/generic_camera.xacro" />
8
141
142   <xacro:generic_camera name="camera" parent="base_link" ros_topic="image_raw" cam_info_topic="camera_info" update_rate="30" res_x="320" res_y="240" image_format="R8G8B8" hfov="70">
143     <origin xyz="0.265 0.0 0.105" rpy="0.0 0 0"/> <!-- base_frame to camera mount transform -->
144   </xacro:generic_camera>
145
```

- roslaunch example_gazebo_map gazebo.launch
- rqt_image_view



- What should we do to add a lidar?

감사합니다.

Q&A

