xamla
robotic solutions

ROSVITA

robot programming software

Fast and intuitive robot programming based on ROS!

Dr. Andre Lemme
Xamla Robotics Team

ROS ximea IDS: YASKAWA ROBOTIQ

Table of contents

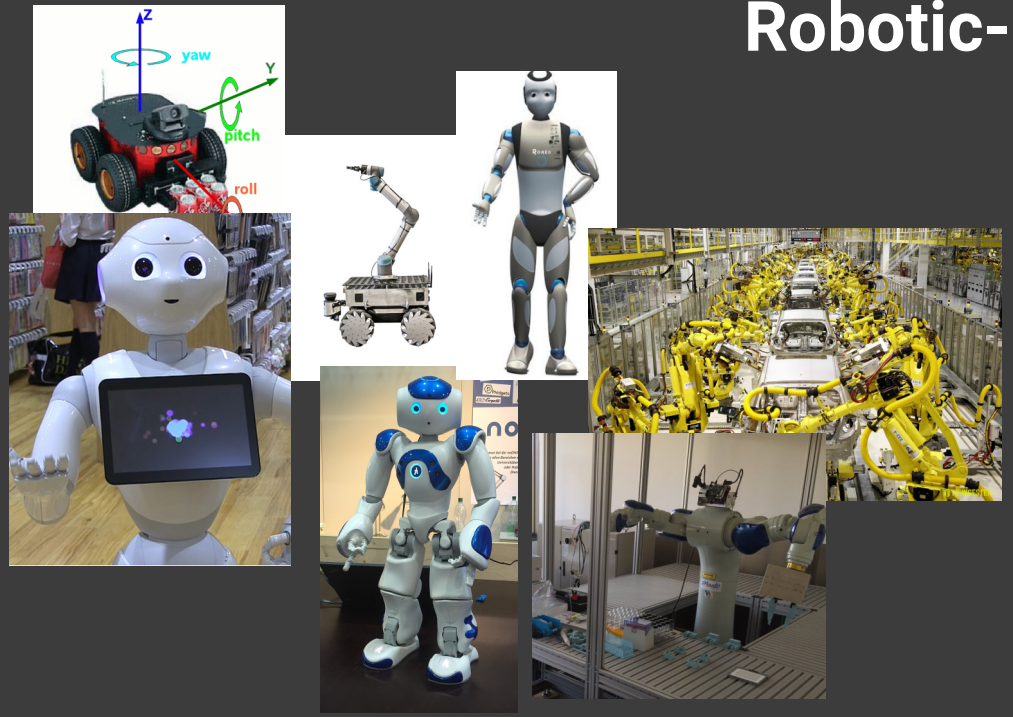
1.1 Overview of Robotics

- Mobile Robots
- Humanoid Robots
- Service Robots
- Industrial Robots

Robotics is a very diverse field where restrictions and requirements come in different combinations!

How can we handle this? → Robotic-Framework

1.2 Requirements and Tasks of a Robotic-Framework



- Integration of sensors and actuators
- Software interfaces

High-Level functionality can be used on different hardware platforms!
ROS represents such a Robotic-Framework.

2.1 Introduction of ROS

- Robot Operating System (ROS)
- Development started in context of the Standford-AI-Robot-Projects in 2007
- Since 2009, the research institute Willow Garage took over
- Since 2013, the Open Source Robotics Foundation (OSRF) is in charge
- Mobile Robots
- Humanoid Robots
- Service Robots
- Acknowledged standard in research and development

2.1 Introduction of ROS

- Robot Operating System (ROS)
  - Linux, MacOS (experimental)
- Open-Source Robotics-Framework
  - Core parts of ROS → BSD License
  - Community packages → Apache 2.0, GPL, MIT License
  - e.g. MoveIt: path-planning with collision avoidance

- Mobile Robots
- Humanoid Robots
- Service Robots
- Industrial Robots

What about industrial applications?

2.2 ROS-Industrial

1. started with the collaboration of
   - Yaskawa Motoman Robotics,
   - Southwest Research Institute
   - Willow Garage
2. founded by Shaun Edwards (SwRI) in January 2012

- Industrial Robots
- Shows promising applications
  1. Welding
  2. Blending process
  3. Pick and Place
  4. Etc...

Community size?

2.3 ROS-Industrial

Consortium Americas
(led by SwRI and located in San Antonio)

Consortium Europe
(led by Fraunhofer IPA and located in Stuttgart, Germany)

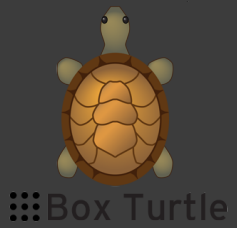
Consortium Asia Pacific
(led by Advanced Remanufacturing and Technology Center (ARTC) and Nanyang Technological Univertiy (NTU) located in Singapore)

Release cycle of ROS?

2.4 ROS Distribution

ROS Box Turtle

March 2, 2010



ROS C Turtle

August 2, 2010

ROS Diamondback

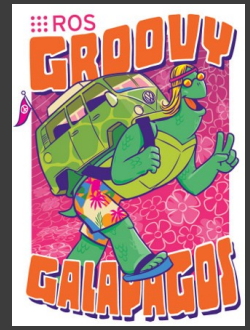
March 2, 2011

ROS Fuerte Turtle

April 23, 2012

ROS Groovy Galapagos

December 31, 2012



2.4 ROS Distribution

ROS Hydro Medusa

September 4, 2013

ROS Indigo Igloo

July 22, 2014

ROS Jade Turtle

May 23, 2015

ROS Kinetic Kame

May 23, 2016

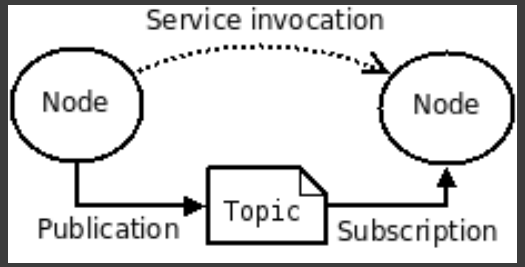
ROS Lunar Loggerhead

May 23, 2017

Synced with ubuntu releases
Every two years new LTS release

2.5 ROS Concepts

1. Modularity
   - Packages for organization
   - Basic build and release item (catkin workspace)
2. Robot programs
   - Nodes and launch-files
3. Communication
   - Topics
   - Services
   - Actions



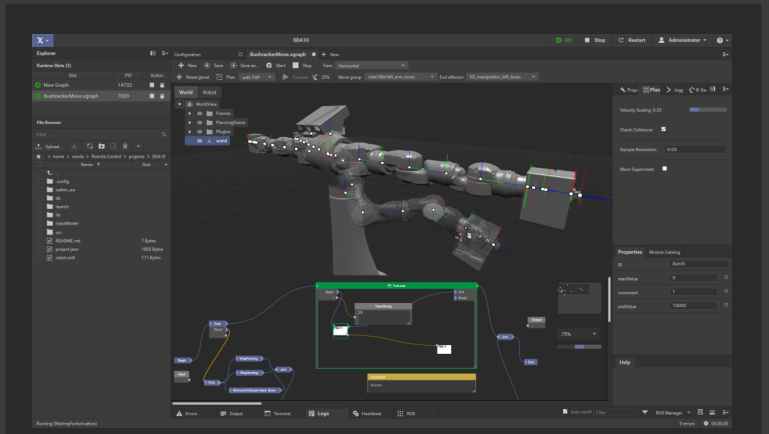
4. Tools
   - rostopic, rosservice, rosparam, roscd, rviz
5. Processes
   - Represented in graph architecture

Goals:

Scale-ability
Flexibility
?Robustness?

From ROS to ROSVITA

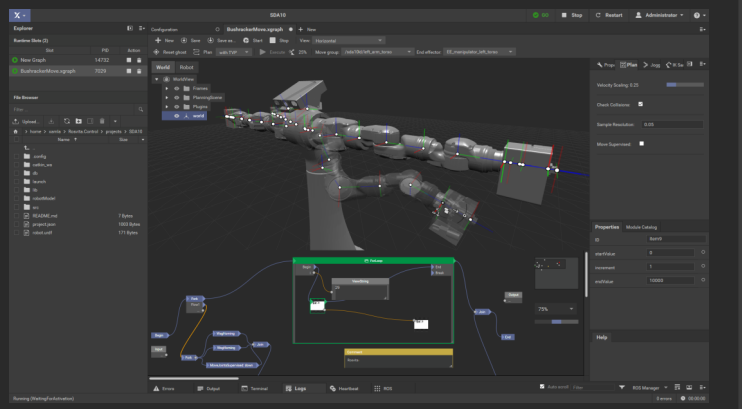
3.1 ROS vs. ROSVITA concepts



Rosvita

- Rosvita is a browser based programming IDE on top of ROS and therefore also using the ROS concepts!
  - Web-Server
  - RosGardener
  - RosServices
  - Motion planning with xamlamoveit

3.2 ROSVITA



Rosvita

- Browser based IDE on top of ROS and therefore also using the ROS concepts!
- High quality 2D and 3D visualization
- Ready-for-Industry robot drivers and other reliable ROS components
- Visual workflow editor with an expandable module system
- All features bundled in docker container for easy deployment

4.1 Getting started with ROSVITA

Getting Started

To use ROSVITA, no complex installation process is necessary, because the ROSVITA Docker image already contains all the necessary packages and dependencies. All you need is an up-to-date Ubuntu operating system (preferentially **Ubuntu 16.04 or higher**) and an up-to-date internet browser (preferentially Google Chrome). Moreover, please read the ROSVITA license agreement.

To **get ROSVITA**, simply download the shell script get_rosvita, then change permissions to be able to execute the script and finally run the script, i.e. in a terminal type:

```
wget -q https://raw.githubusercontent.com/Xamla/docs.xamla.com/gh-pages/rosvita/downloads/get_rosvita
chmod u+x get_rosvita
./get_rosvita
```

Here, we use **wget** to download the script. If not already installed, simply install **wget** via `sudo apt install wget`.

The script will:

1. Install Docker if not already installed, and add user to group docker.
2. Download the **ROSVITA Docker image** from Docker Hub.
3. Download the **ROSVITA start script**: rosvita_start.
4. Download the **ROSVITA stop script**: rosvita_stop.
5. Create a **desktop icon** for starting ROSVITA.
6. If not already in group docker, **restart the computer** to activate group changes.

Now you can **start ROSVITA by double clicking the desktop icon**, or by opening a terminal and running the start script by simply typing:

```
rosvita_start
```

(To run ROSVITA in **debug mode** type `rosvita_start --debug`. Afterwards enter **rosvita** into the terminal, open a web browser and enter `localhost:5000` into the address bar.)

VIDEO

1. Install Docker if not already installed, and add user to group docker.
2. Download the **ROSVITA Docker image** from Docker Hub.
3. Download the **ROSVITA start script**: rosvita_start.
4. Download the **ROSVITA stop script**: rosvita_stop.
5. Create a **desktop icon** for starting ROSVITA.
6. If not already in group docker, **restart the computer** to activate group changes.

Now you can **start ROSVITA by double clicking the desktop icon**, or by opening a terminal and running the start script by simply typing:

```
rosvita_start
```

(To run ROSVITA in **debug mode** type `rosvita_start --debug`. Afterwards enter **rosvita** into the terminal, open a web browser and enter `localhost:5000` into the address bar.)

As a result, the ROSVITA login screen appears in your default web browser. After successful login with username and password (default: **admin** and **r2d2c3po**), the ROSVITA main development environment opens (see next chapter).

To **stop ROSVITA** simply run the ROSVITA stop script:

```
rosvita_stop
```

To **update ROSVITA** simply run the **get_rosvita** script again, or use the following command:

```
docker pull xamla/rosvita:latest
```

With the terminal command `docker images` you will see all downloaded ROSVITA versions with corresponding image ids. Use `docker rmi <image-id>` to remove old versions and save disk space.

If you have any trouble or would like to give us feedback, feel free to visit our ROSVITA forum. Moreover, check out our ROSVITA tutorial videos, especially the ROSVITA quick start video.

Note

We strongly recommend to run ROSVITA in **Google Chrome**, because in other internet browsers like Firefox the user experience will be limited due to the significantly slower browser performance.

Previous Next

4.2 Docker as deployment tool

“Docker – Build, Ship, and Run Any App, Anywhere. ”

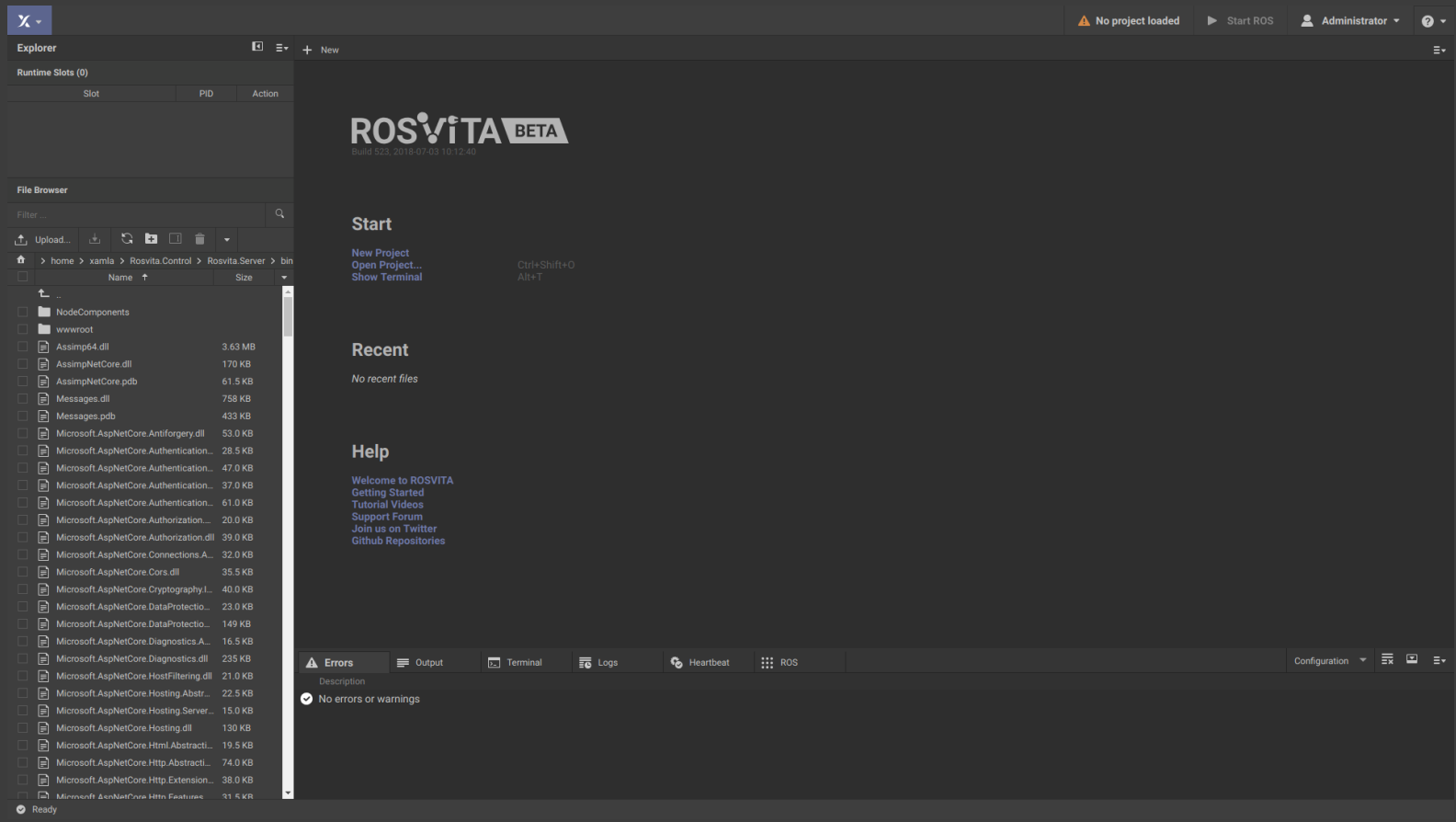
- Docker image
- Persistent data located in host system
  - ~/Rosvita/data
  - ~/Rosvita/projects
  - ~/Rosvita/robot_parts
- Advantages
  - Available on Linux/MacOs/Windows
  - Robust system → image restart
  - Dependencies bundeled

4.3 What is docker and why is it useful?

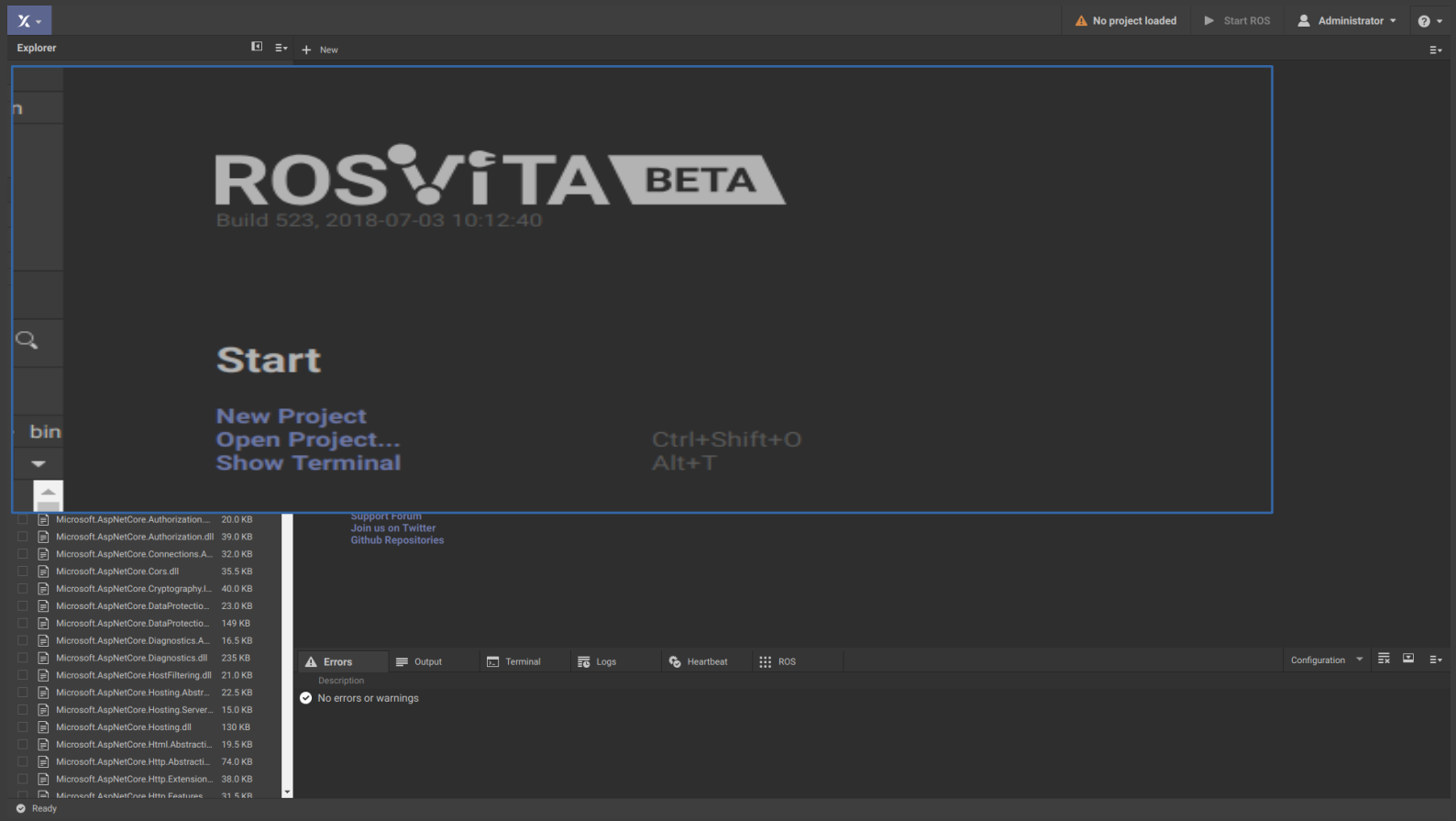
“Docker – Build, Ship, and Run Any App, Anywhere. ”

- Docker image
- Persistent data located in host system
- Advantages
  - Available on Linux/MacOs/Windows
  - Robust system → image restart
  - Dependencies bundeled
- Disadvantages
  - Difficult to connect to hardware

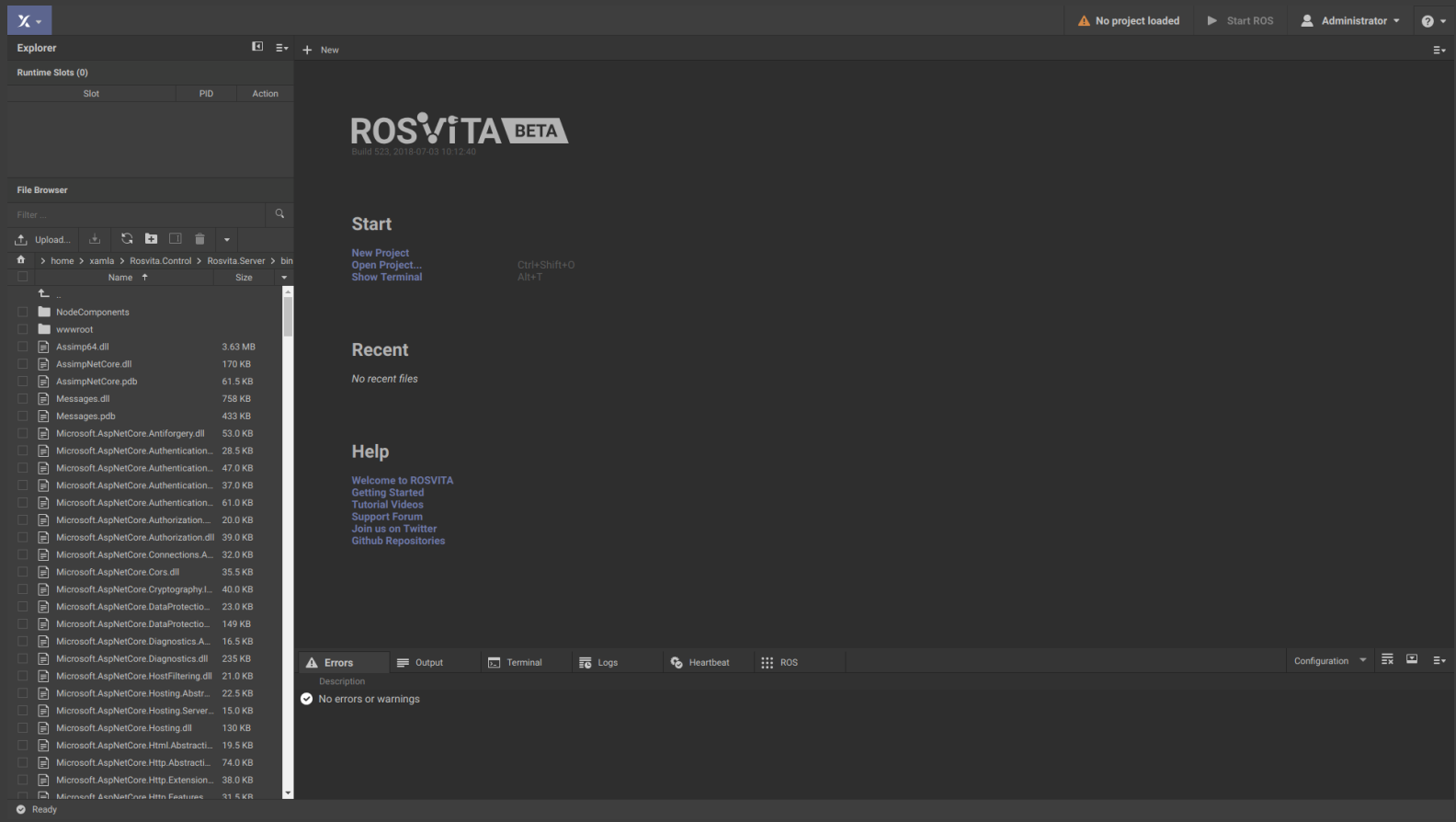
5.1 Welcome to ROSVITA



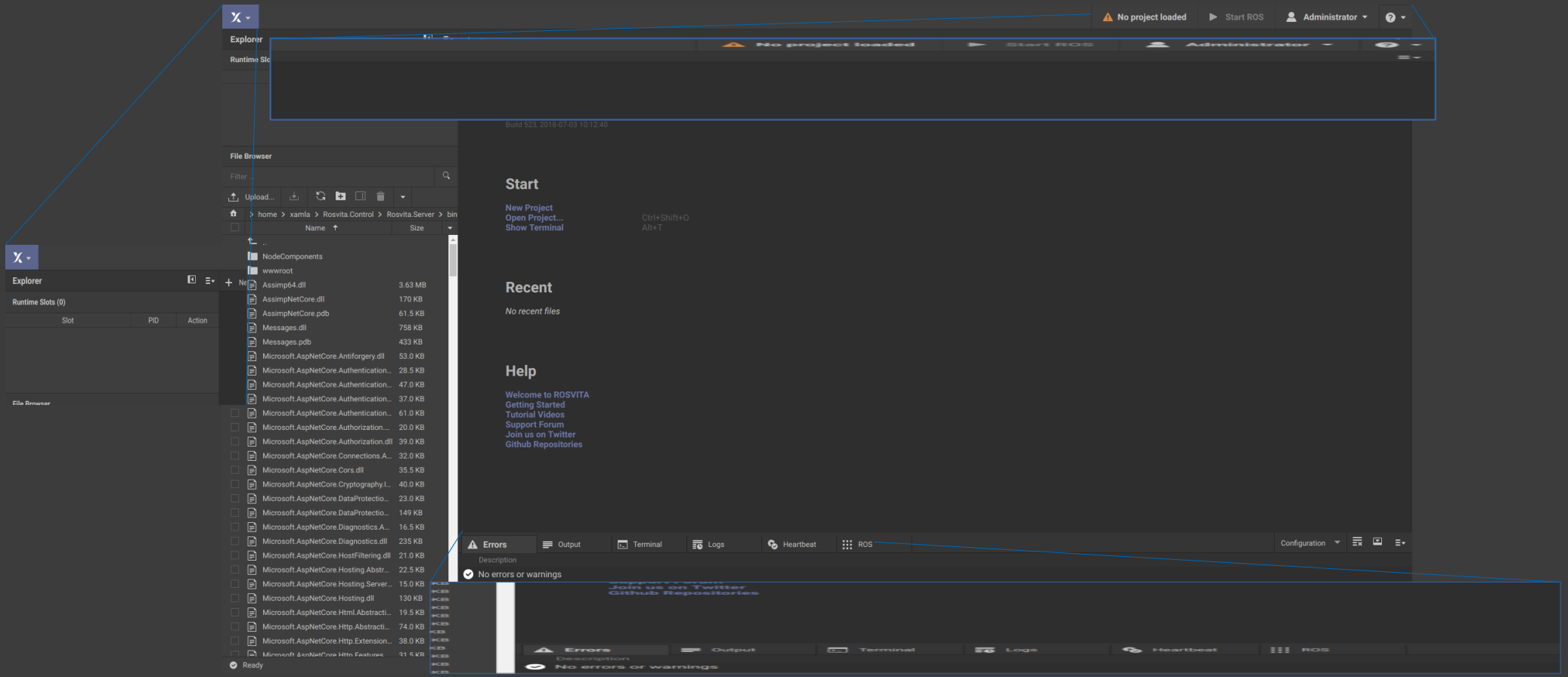
5.2 Welcome to ROSVITA



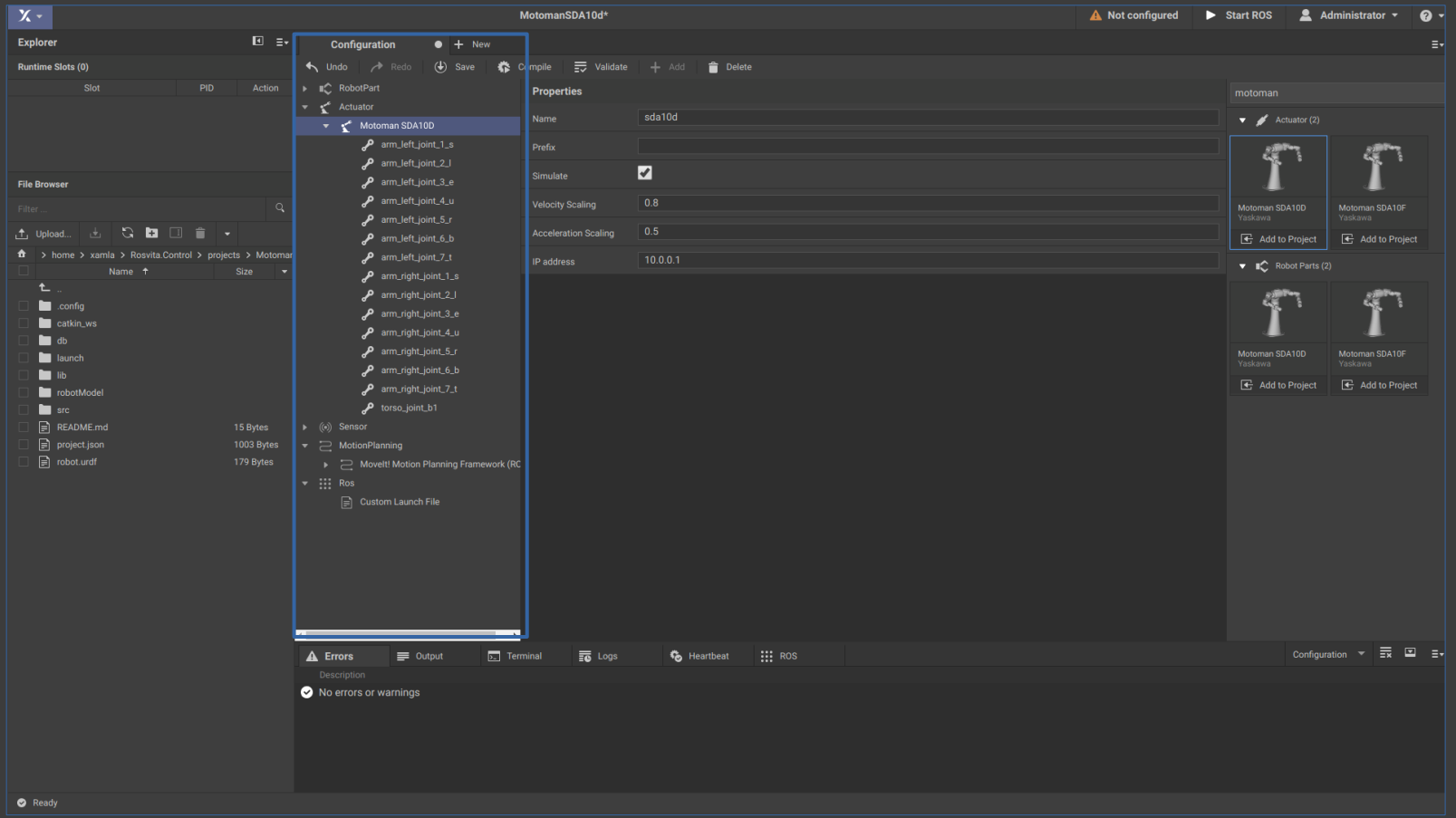
5.2 Welcome to ROSVITA

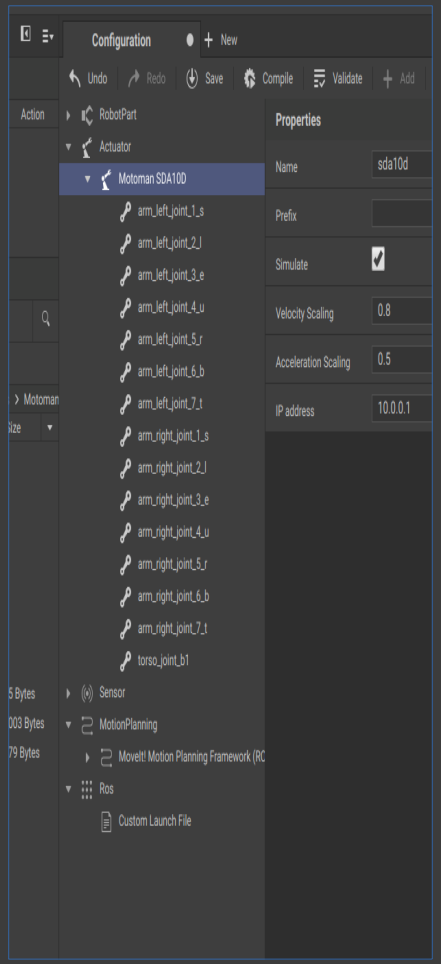


5.2 Welcome to ROSVITA



6.1 Robot configuration

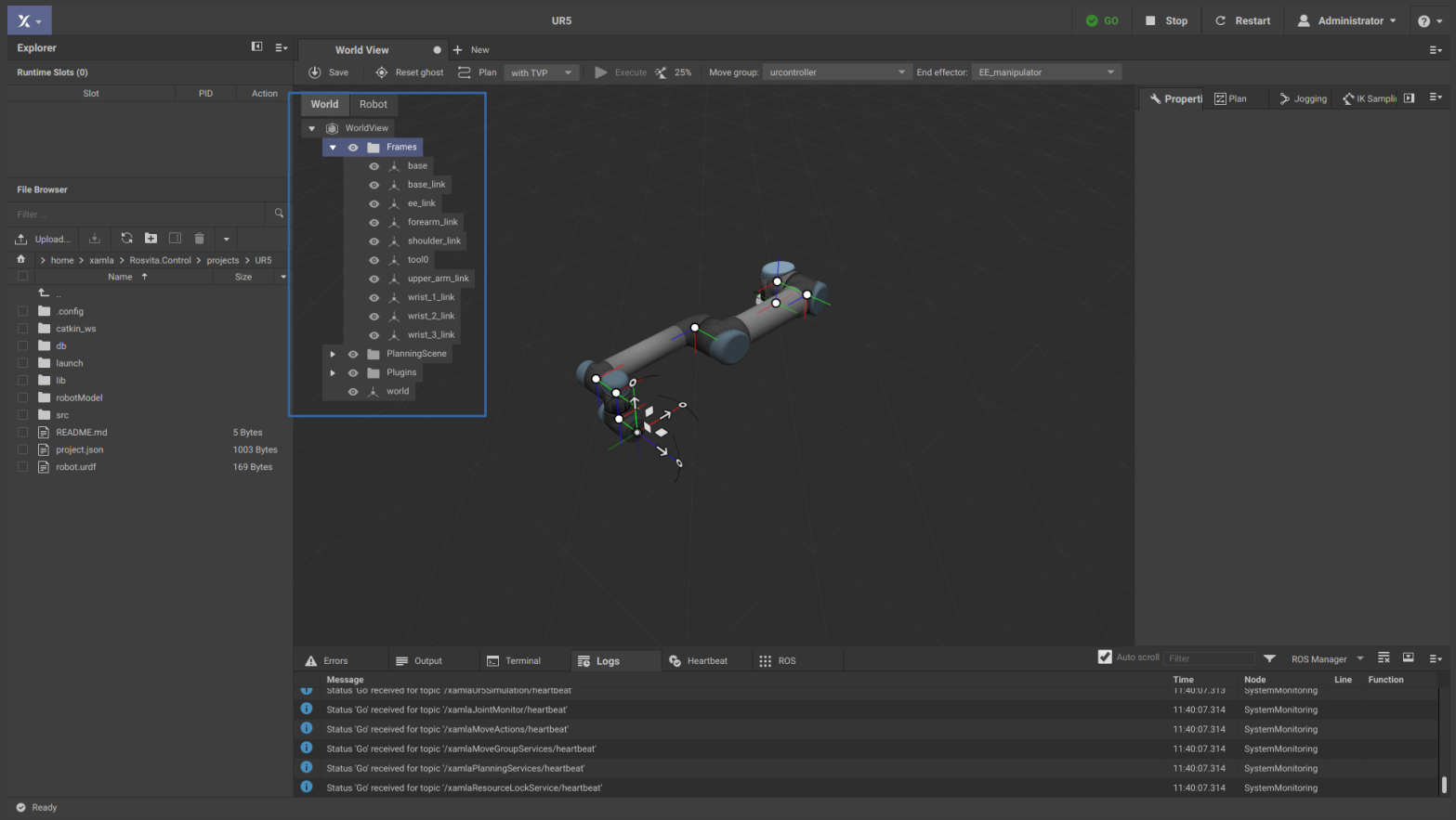




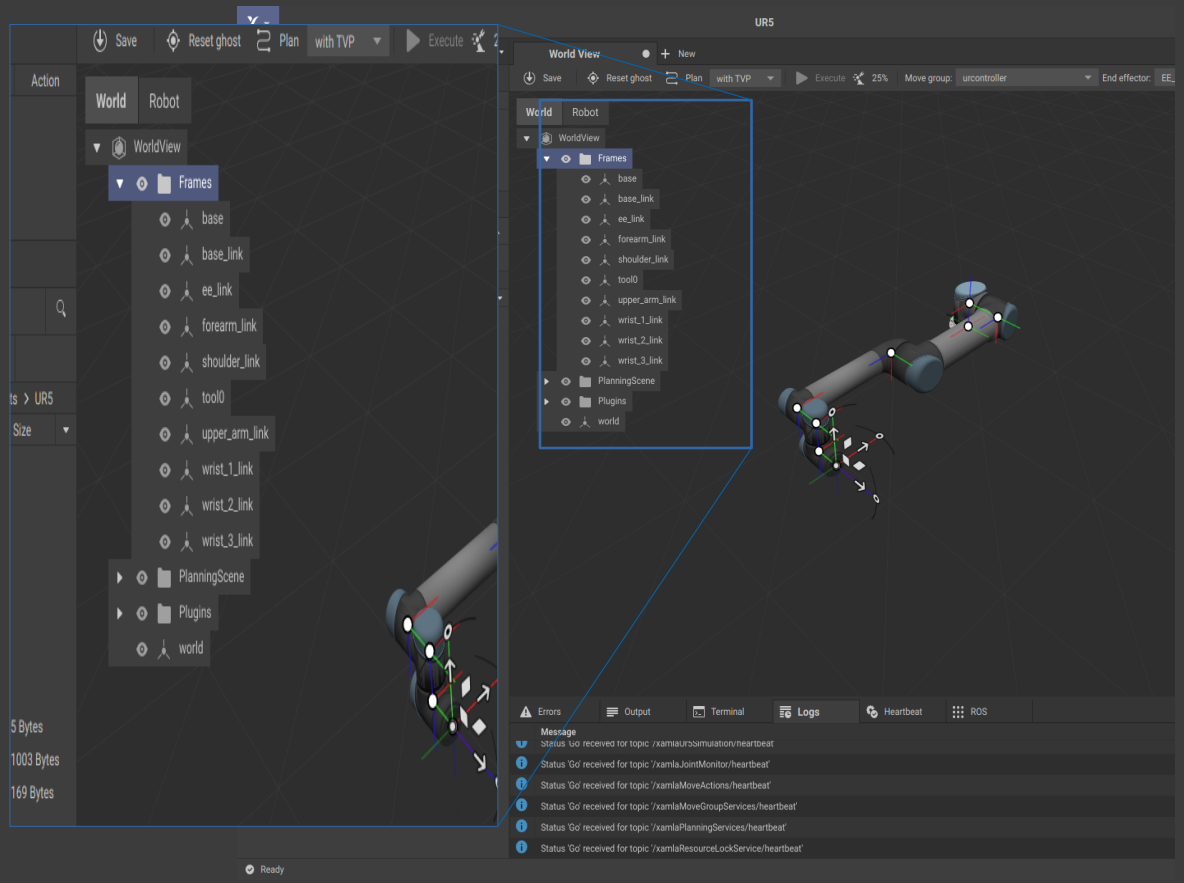
6.2 Robot configuration

- RobotPart
  - URDF, Xacro, standard interface
- Actuator
  - Driver, corresponding to urdf
- Sensor
  - Cameras etc...
- MotionPlanning
  - MoveIt configuration and extras
- ROS Launch File

7.1 World view

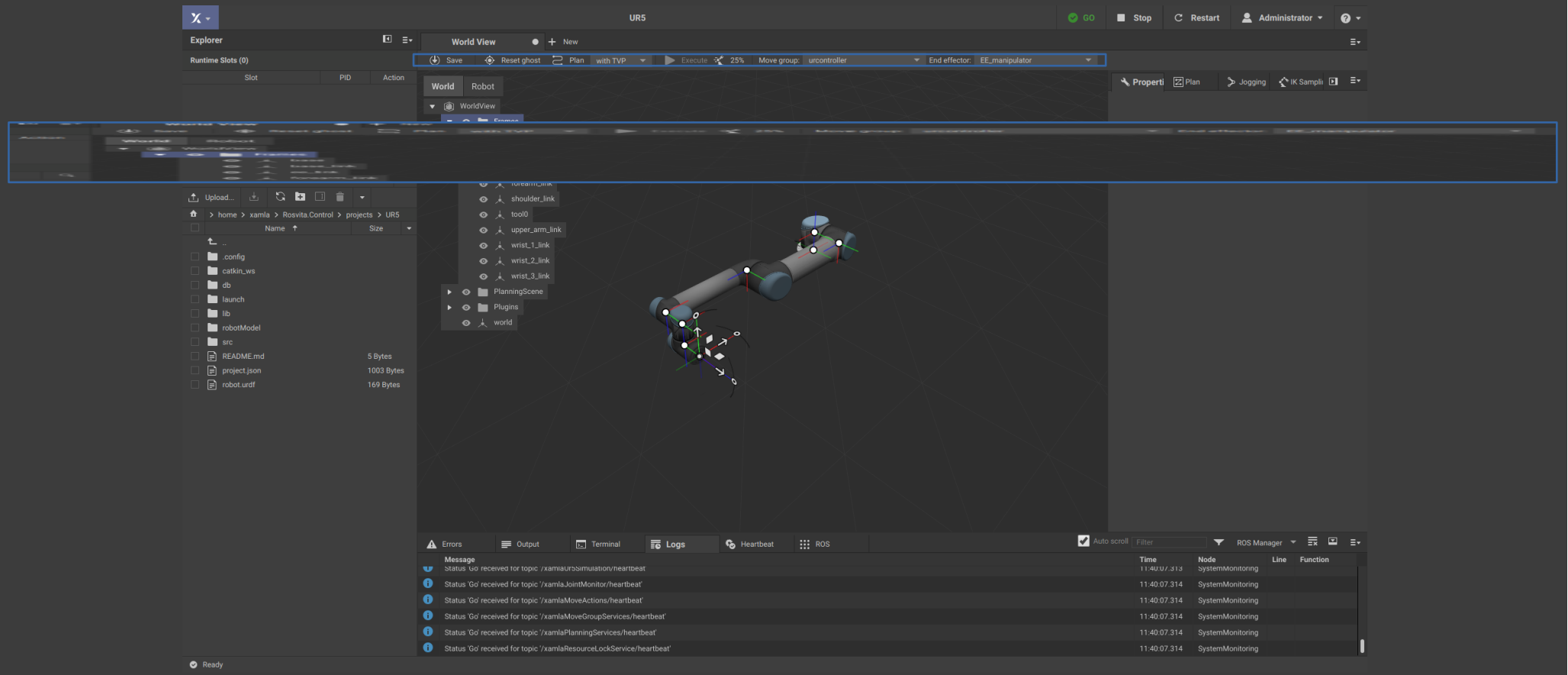


7.1 World view

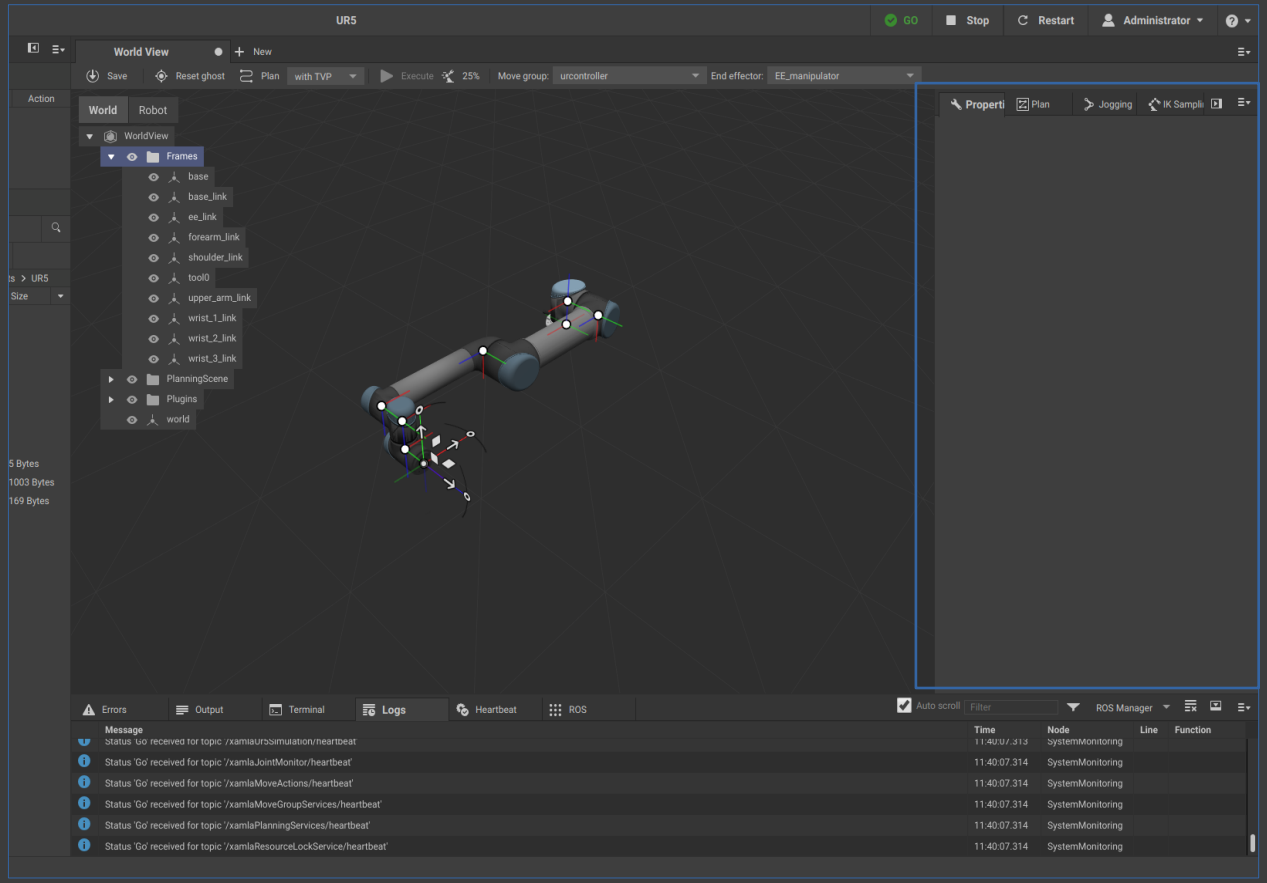


- Frames
- CollsionObjects
- Plugins
- Poses
- JointValues

7.1 World view

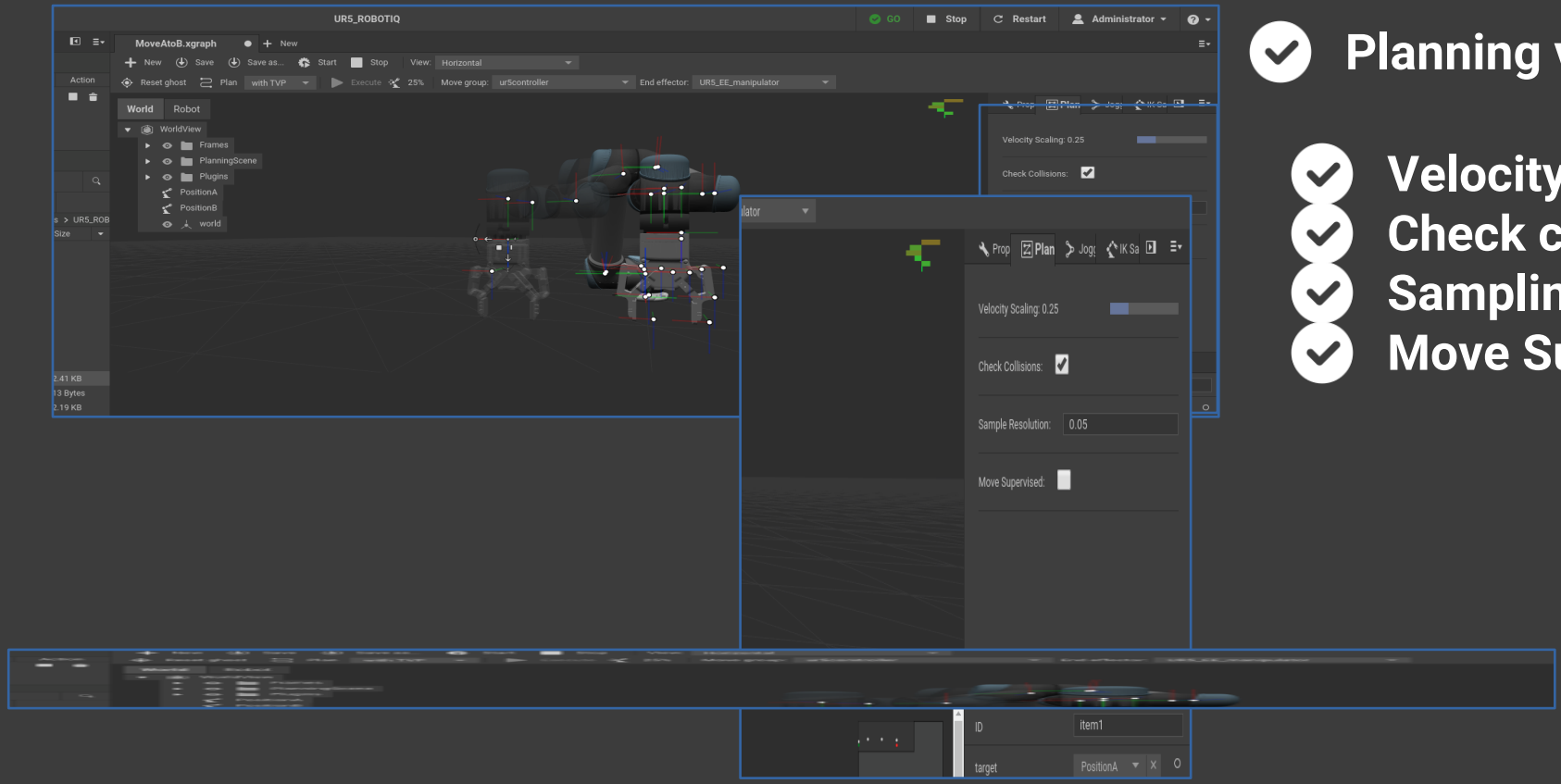


7.2 World view



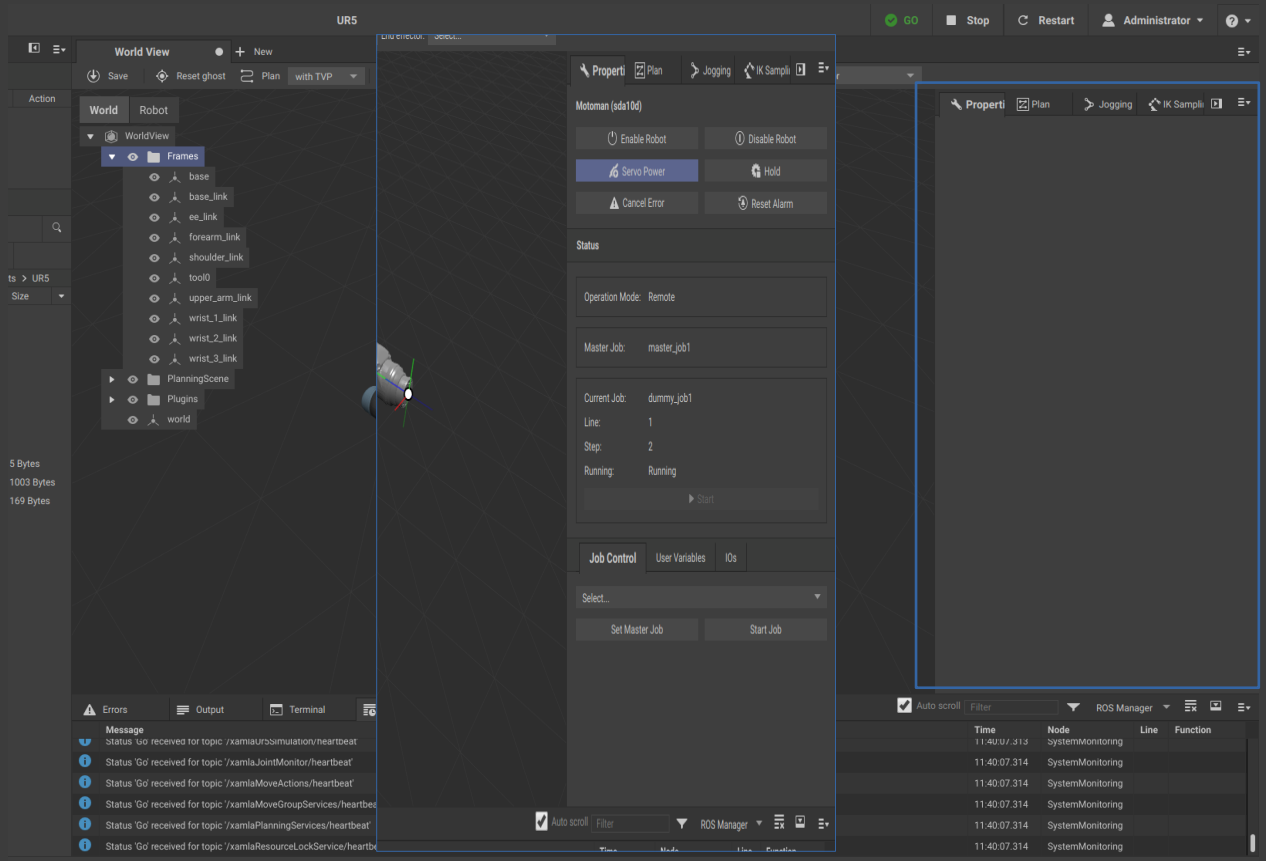
- **Property view**
  - **Detailed information of all datatypes in the tree view.**

7.3 World view



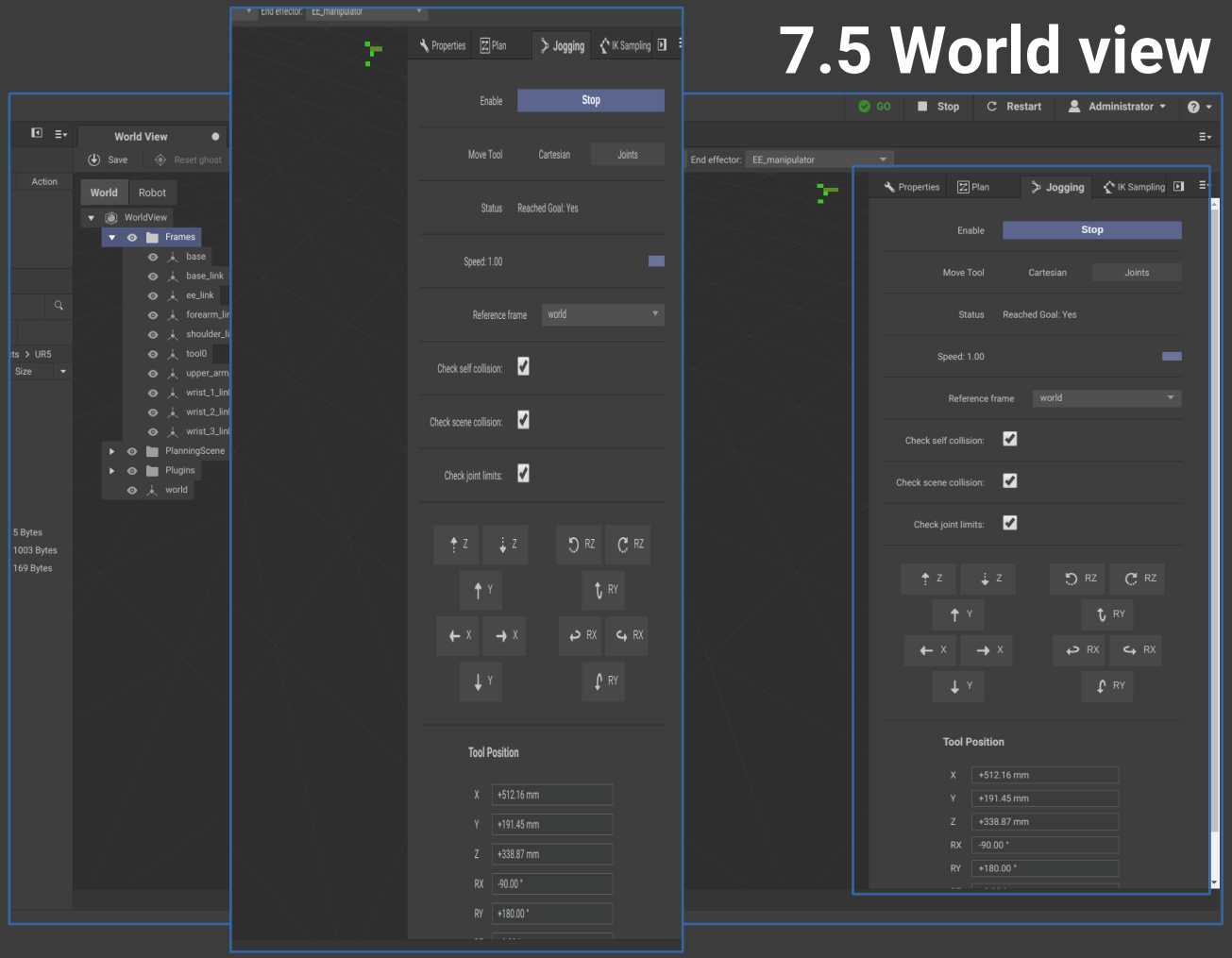
- Planning view
- Velocity scaling
- Check collisions
- Sampling Resolution
- Move Supervised

7.4 World view



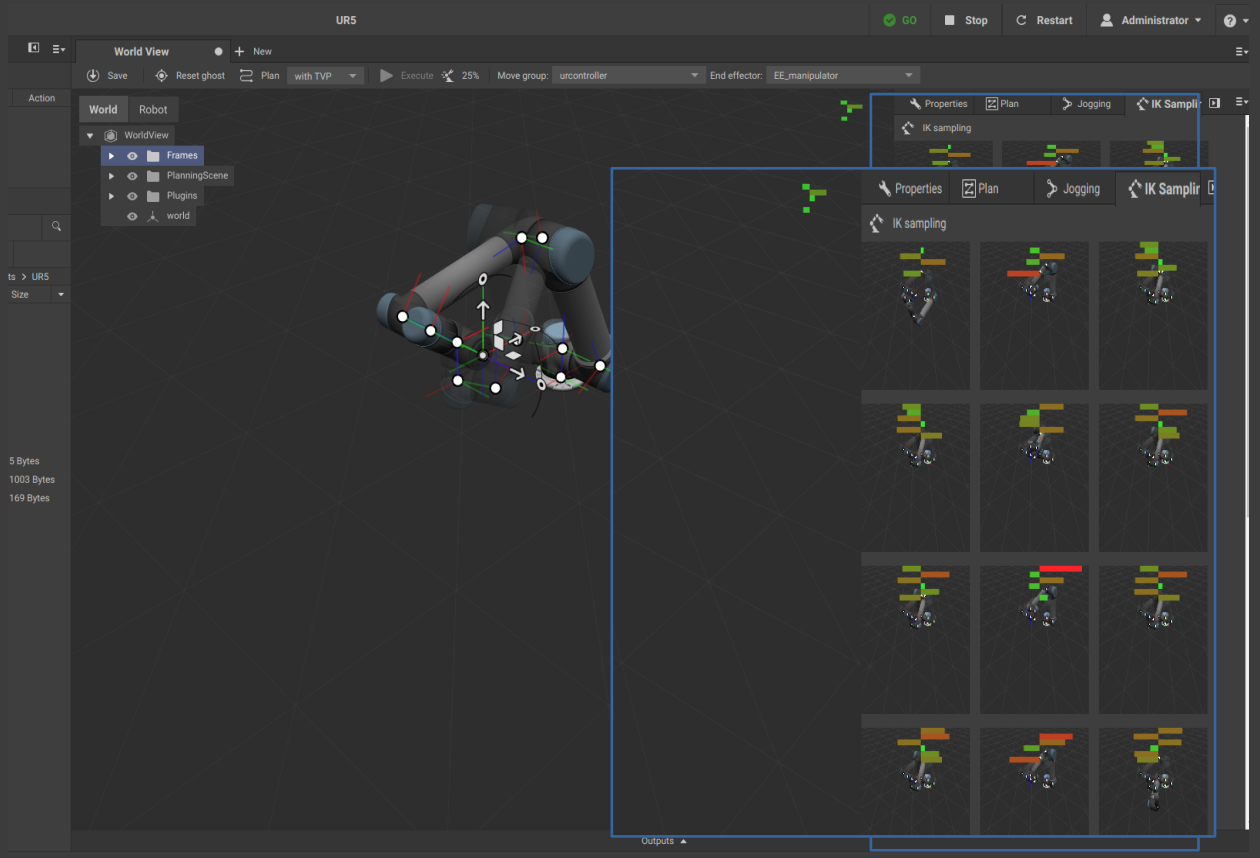
- Property view
  - Detailed information of all datatypes in the tree view.
  - Plugins

7.5 World view



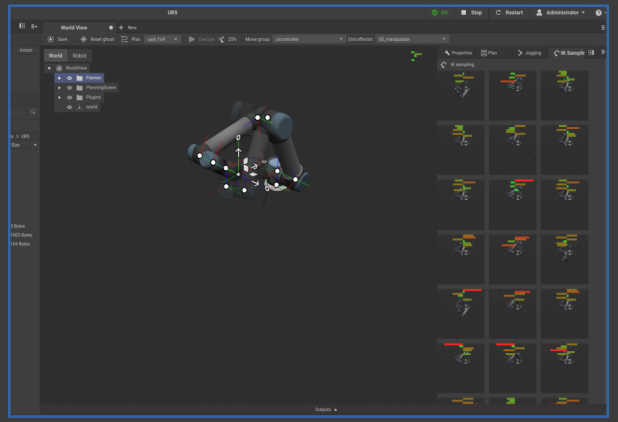
- Property view
  - Detailed information of all datatypes in the tree view.
  - Plugins
- Jogging view
  - Joint Space
  - Cartesian Space

7.6 World view



- Property view
  - Detailed information of all datatypes in the tree view.
  - Plugins
- Jogging view
  - Joint Space
  - Cartesian Space
- IK-Sampling
  - Find better joint values for Pose

7.8 World view



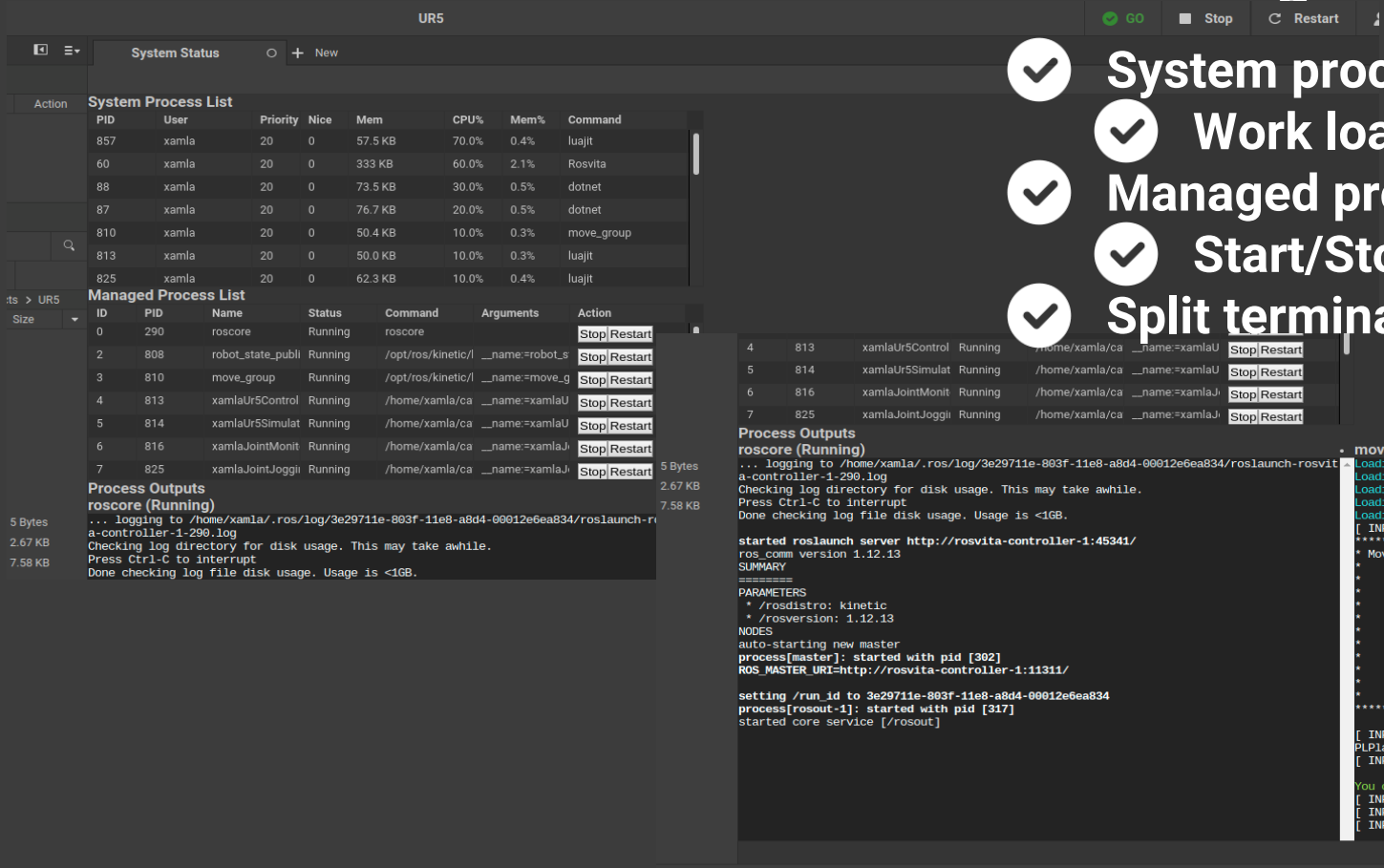
Conclusion:

- Positioning the robot
- Store data
- Try different things

- Jogging view
  - Joint Space
  - Cartesian Space

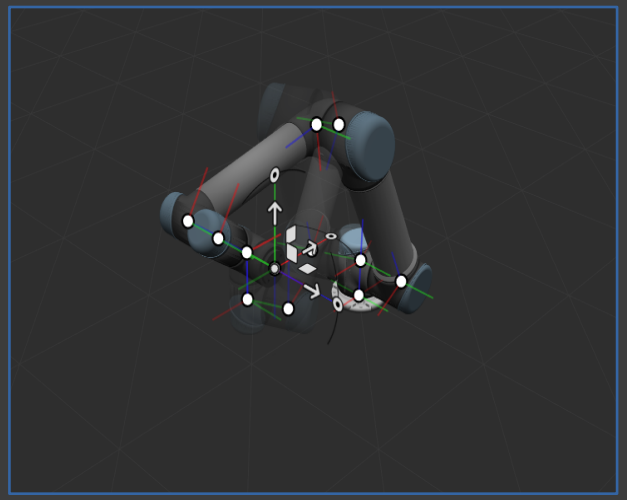
- IK-Sampling
  - Find better joint values for Pose

8.1 Status Monitoring



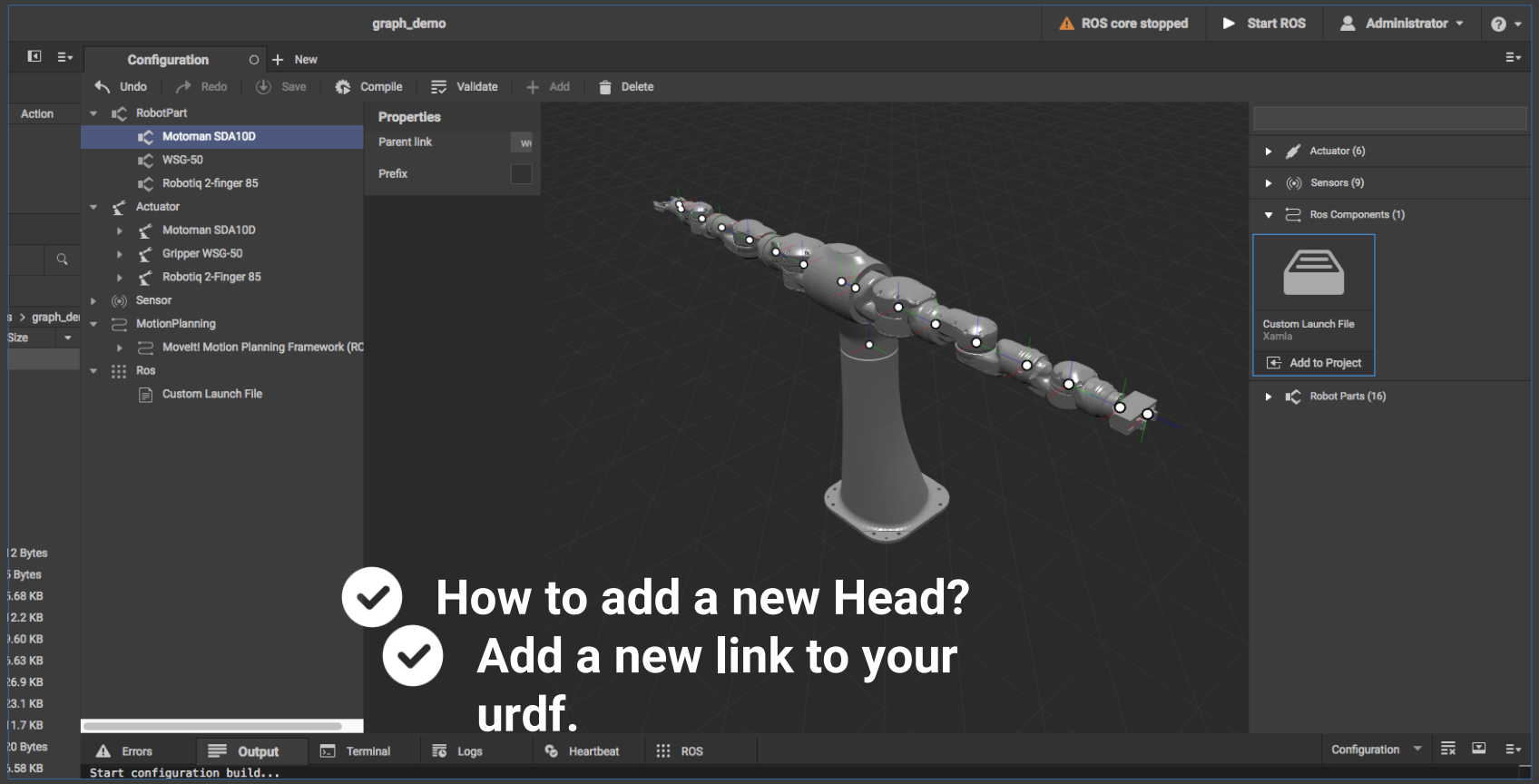
- System process list
- Work load on system
- Managed process list
- Start/Stop of individual nodes
- Split terminal output per node

9.1 Customizing

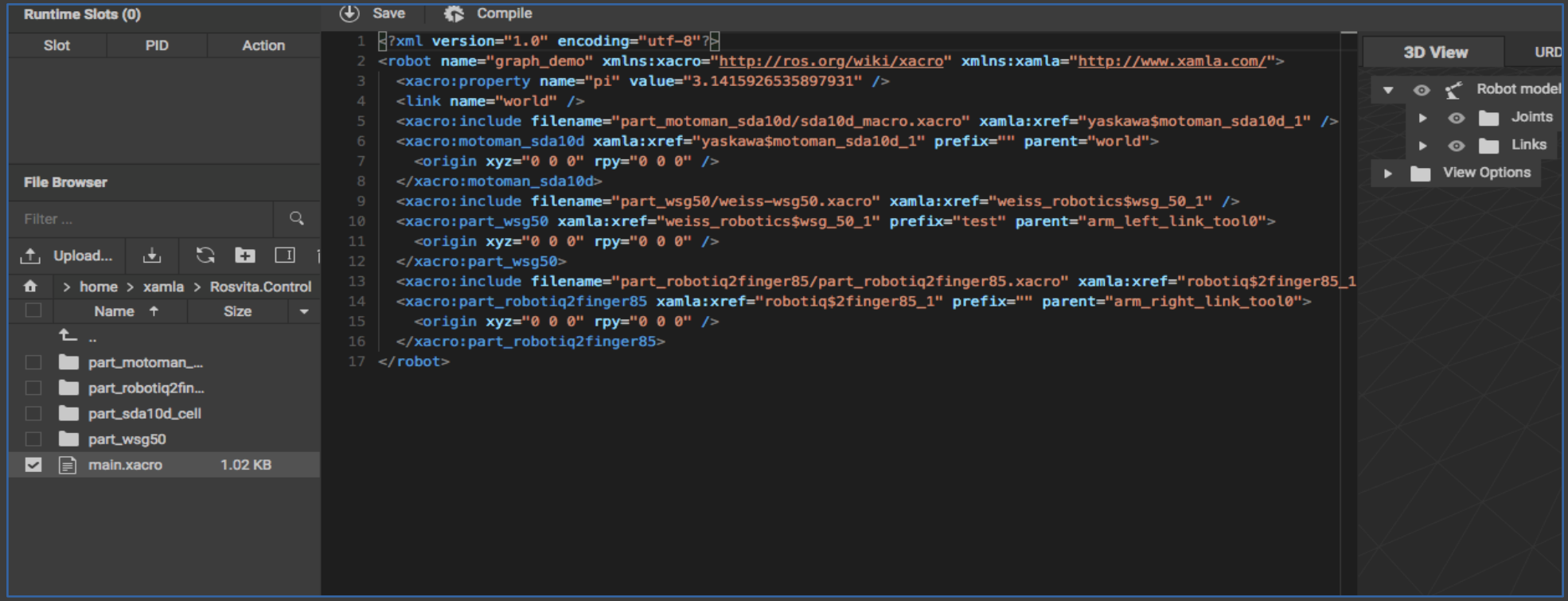


- Add your own link to your setup
- Add full robot part
- Add ros package
- Add your own hardware plugin

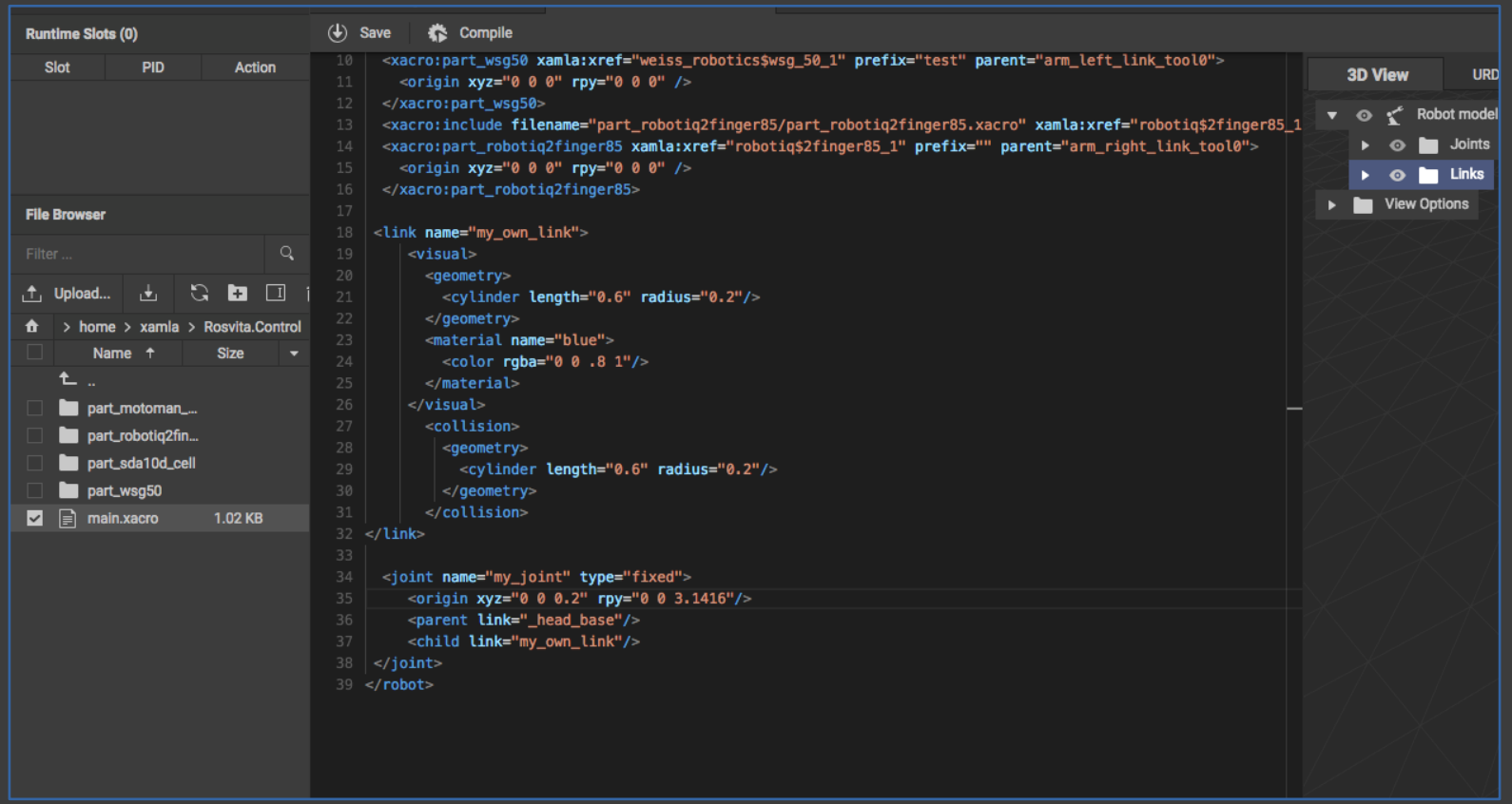
9.2 Costum robot part



9.2 Costum robot part



9.3 Costum robot part



9.3 Costum robot part

```
    <origin xyz="0 0 0" rpy="0 0 0" />
  </xacro:part_robotiq2finger85>

<link name="my_own_link">
    <visual>
      <geometry>
        <cylinder length="0.6" radius="0.2"/>
      </geometry>
      <material name="blue">
        <color rgba="0 0 .8 1"/>
      </material>
    </visual>
      <collision>
        <geometry>
          <cylinder length="0.6" radius="0.2"/>
        </geometry>
      </collision>
</link>

  <joint name="my_joint" type="fixed">
    <origin xyz="0 0 0.2" rpy="0 0 3.1416"/>
    <parent link="_head_base"/>
    <child link="my_own_link"/>
</joint>
```

- Simple primitives
- Mesh descriptions
- Visual (Collada files)
- Collision (step files)

How to connect to my model?

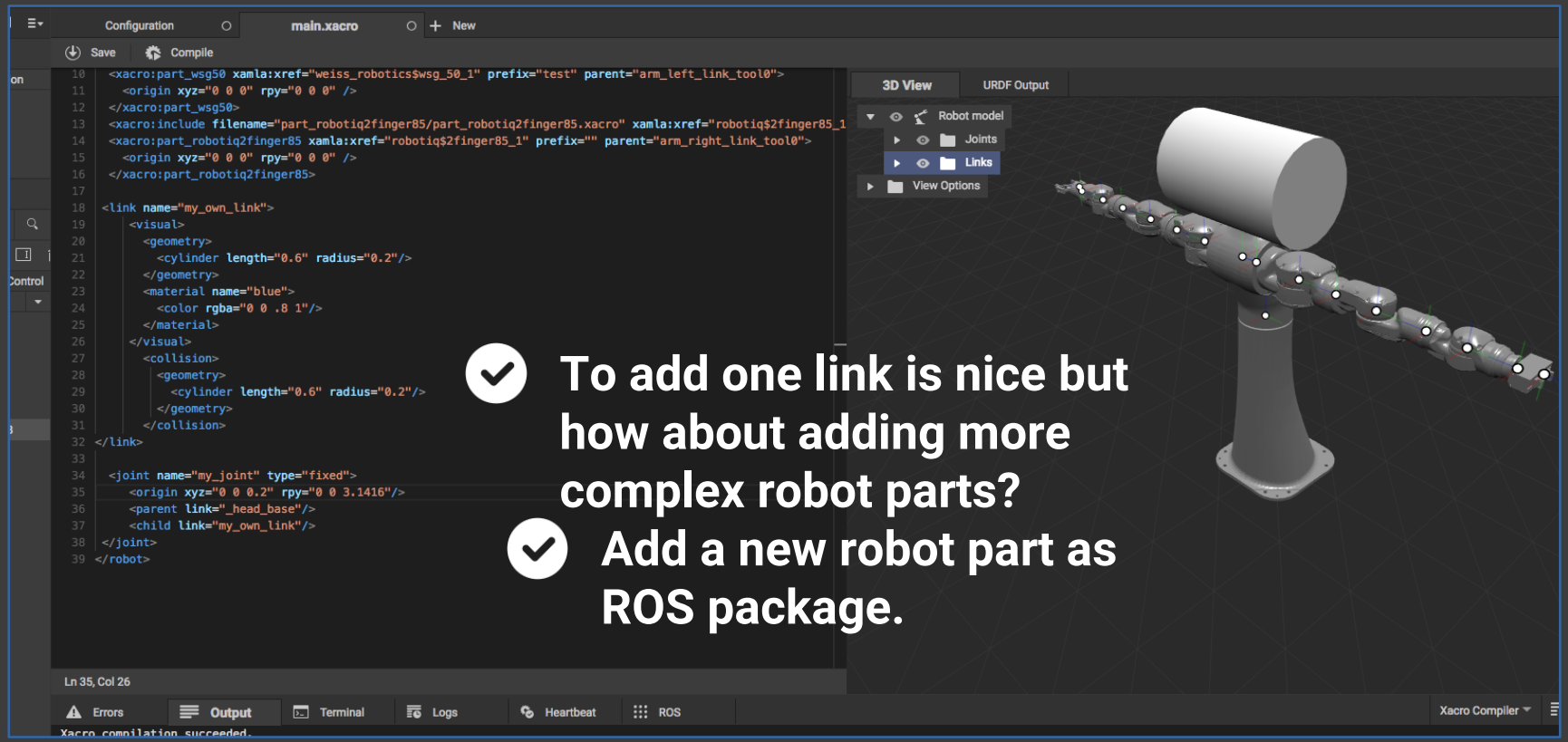
9.3 Costum robot part

```
    <origin xyz="0 0 0" rpy="0 0 0" />
  </xacro:part_robotiq2finger85>

<link name="my_own_link">
    <visual>
      <geometry>
        <cylinder length="0.6" radius="0.2"/>
      </geometry>
      <material name="blue">
        <color rgba="0 0 .8 1"/>
      </material>
    </visual>
      <collision>
        <geometry>
          <cylinder length="0.6" radius="0.2"/>
        </geometry>
      </collision>
</link>

  <joint name="my_joint" type="fixed">
    <origin xyz="0 0 0.2" rpy="0 0 3.1416"/>
    <parent link="_head_base"/>
    <child link="my_own_link"/>
  </joint>
</robot>
```

- Connect with joints
  - Add to the chain
  - Parent
  - Child

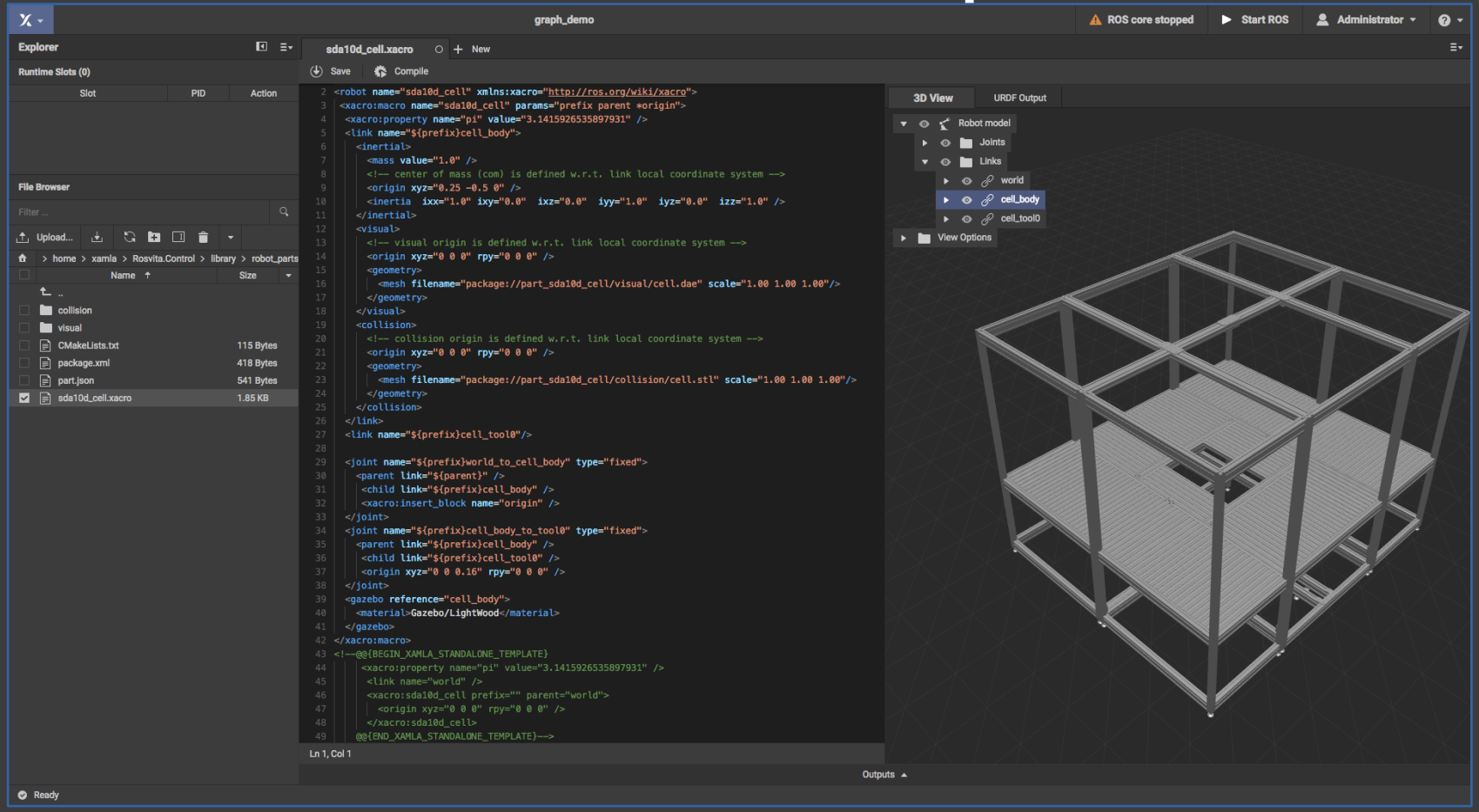
Press compile and you are done!

9.3 Costum robot part

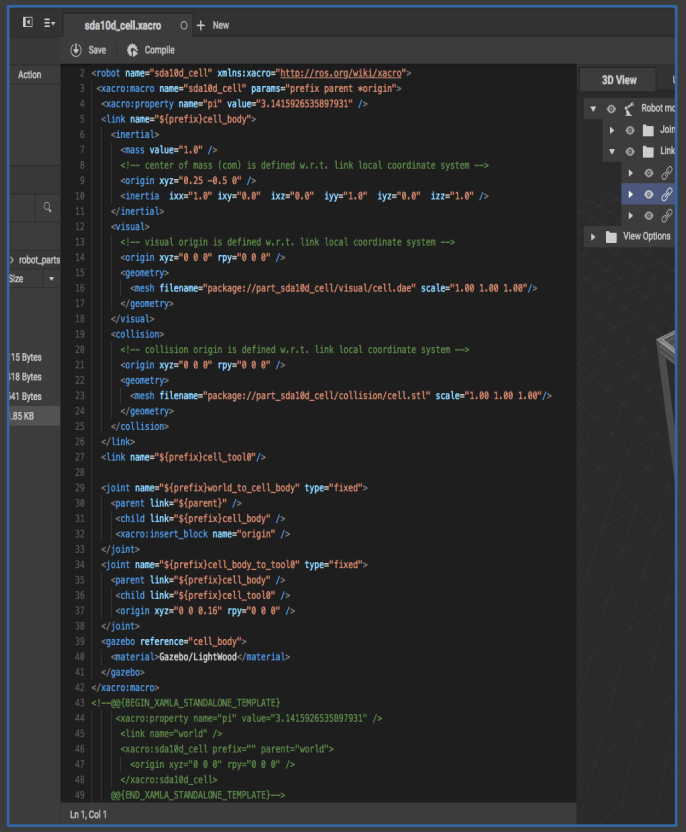


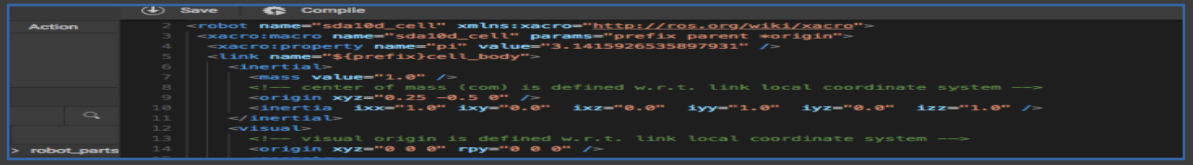
- To add one link is nice but how about adding more complex robot parts?
- Add a new robot part as ROS package.

9.4 Costum robot part



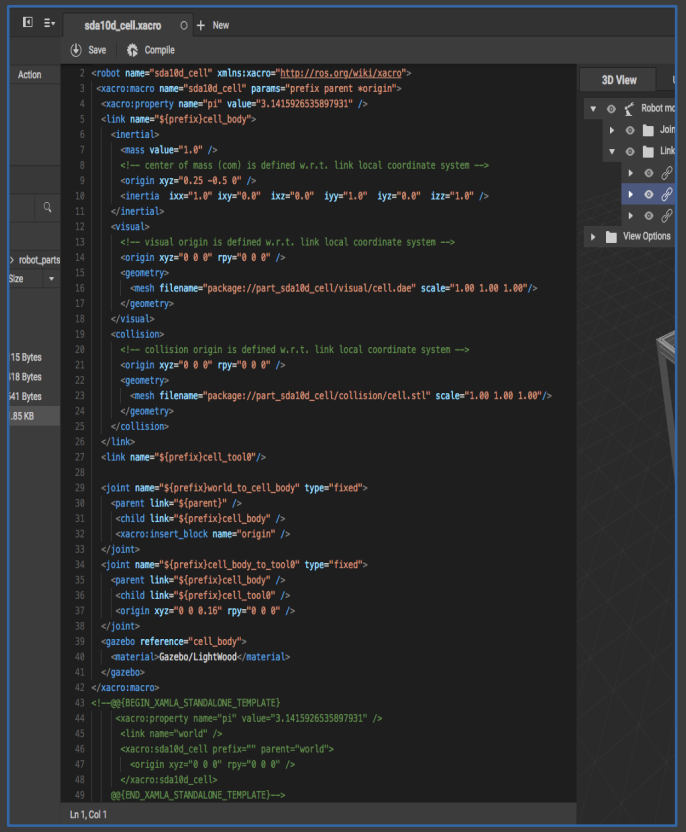
9.4 Costum robot part





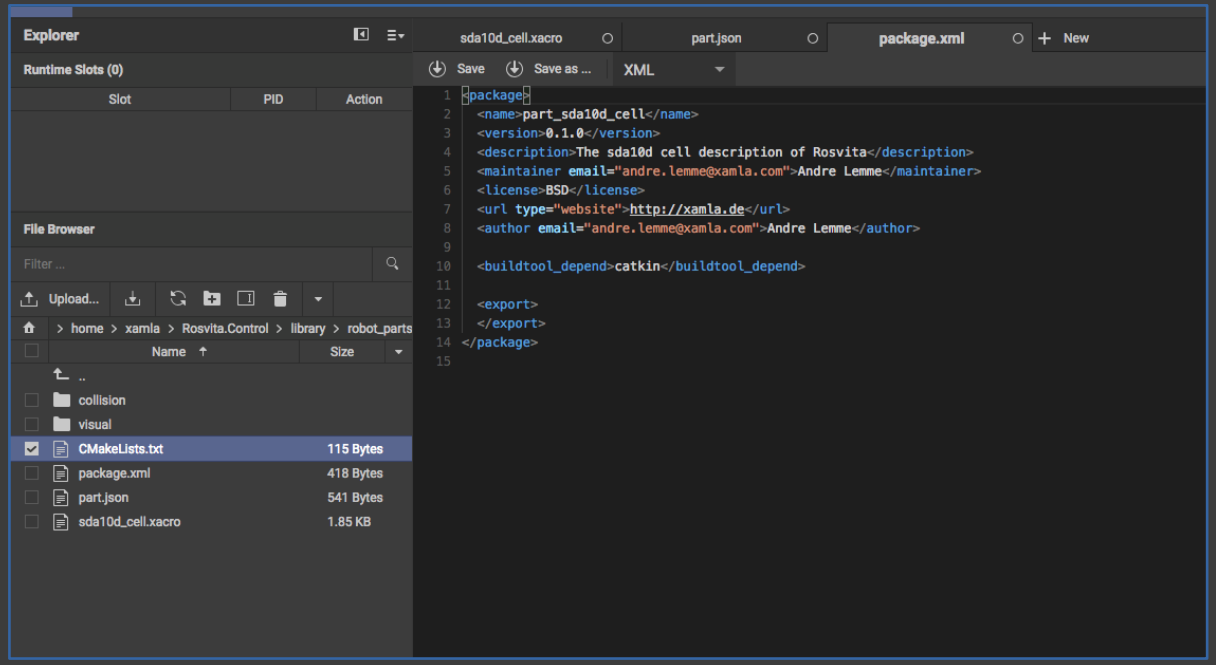
- Prefix
  - Make sure your joints have uniqe identifier
- Parent
  - Where to attache to
- Origin
  - Releative transformation to parent link

9.5 Costum robot part



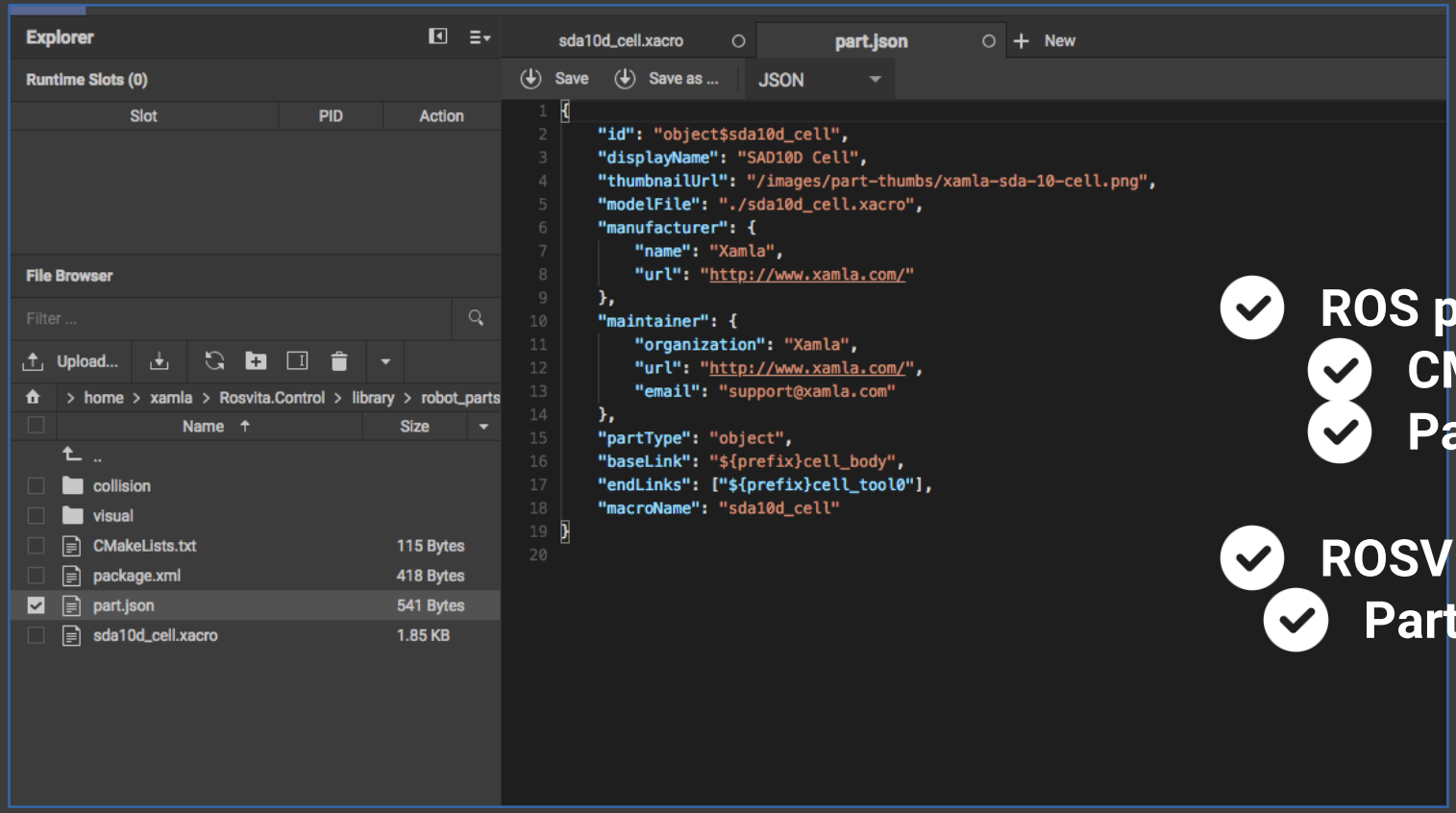

```
<robot name="sda10d_cell" xmlns:xacro="http://ros.org/wiki/xacro">
 <xacro:macro name="sda10d_cell" params="prefix parent *origin">
  <xacro:property name="pi" value="3.1415926535897931" />
  <link name="${prefix}cell_body">
    <inertial>
      <mass value="1.0" />
      <!-- center of mass (com) is defined w.r.t. link local coordinate system -->
      <origin xyz="0.25 -0.5 0" />
      <inertia  ixx="1.0" ixy="0.0"  ixz="0.0"  iyy="1.0"  iyz="0.0"  izz="1.0" />
    </inertial>
    <visual>
      <!-- visual origin is defined w.r.t. link local coordinate system -->
      <origin xyz="0 0 0" rpy="0 0 0" />
      <geometry>
        <mesh filename="package://part_sda10d_cell/visual/cell.dae" scale="1.00 1.00 1.00"/>
      </geometry>
    </visual>
    <collision>
      <!-- collision origin is defined w.r.t. link local coordinate system -->
      <origin xyz="0 0 0" rpy="0 0 0" />
      <geometry>
        <mesh filename="package://part_sda10d_cell/collision/cell.stl" scale="1.00 1.00 1.00"/>
      </geometry>
    </collision>
  </link>
  <link name="${prefix}cell_tool0"/>

  <joint name="${prefix}world_to_cell_body" type="fixed">
    <parent link="${parent}" />
    <child link="${prefix}cell_body" />
    <xacro:insert_block name="origin" />
  </joint>
  <joint name="${prefix}cell_body_to_tool0" type="fixed">
    <parent link="${prefix}cell_body" />
    <child link="${prefix}cell_tool0" />
    <origin xyz="0 0 0.16" rpy="0 0 0" />
  </joint>
  <gazebo reference="cell_body">
    <material>Gazebo/LightWood</material>
  </gazebo>
 </xacro:macro>
 <!--@@{BEGIN_XAMLA_STANDALONE_TEMPLATE}
    <xacro:property name="pi" value="3.1415926535897931" />
    <link name="world" />
    <xacro:sda10d_cell prefix="" parent="world">
      <origin xyz="0 0 0" rpy="0 0 0" />
    </xacro:sda10d_cell>
   @@{END_XAMLA_STANDALONE_TEMPLATE}-->
```

9.5 Costum robot part



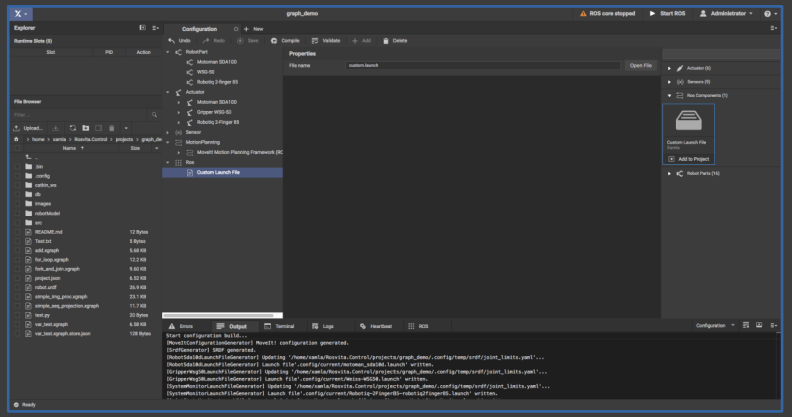
- ROS package
  - CMakeLists.txt
  - Package.xml

9.5 Costum robot_part

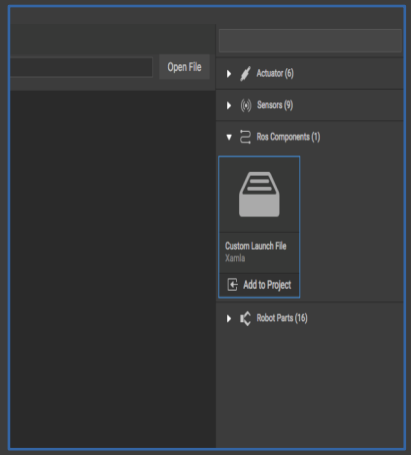


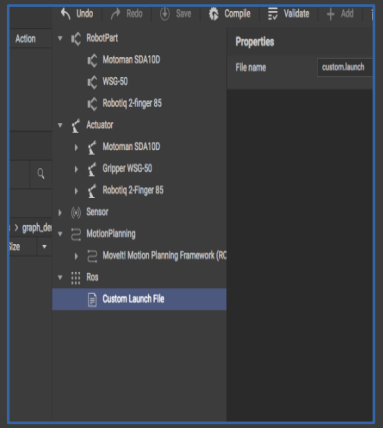
- ROS package
  - CMakeLists.txt
  - Package.xml
- ROSVITA additions
  - Part.json

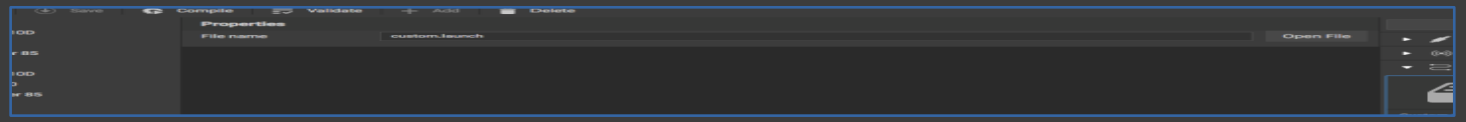
9.6 Costum launch files



- Select ROS Launch File in rosvita library
- Open launch file

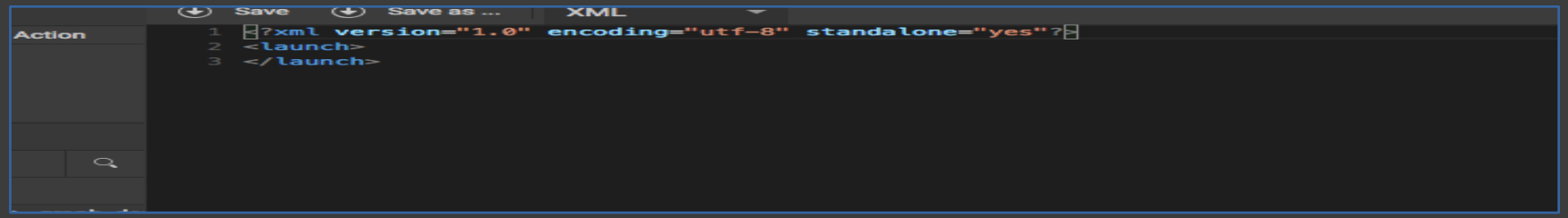






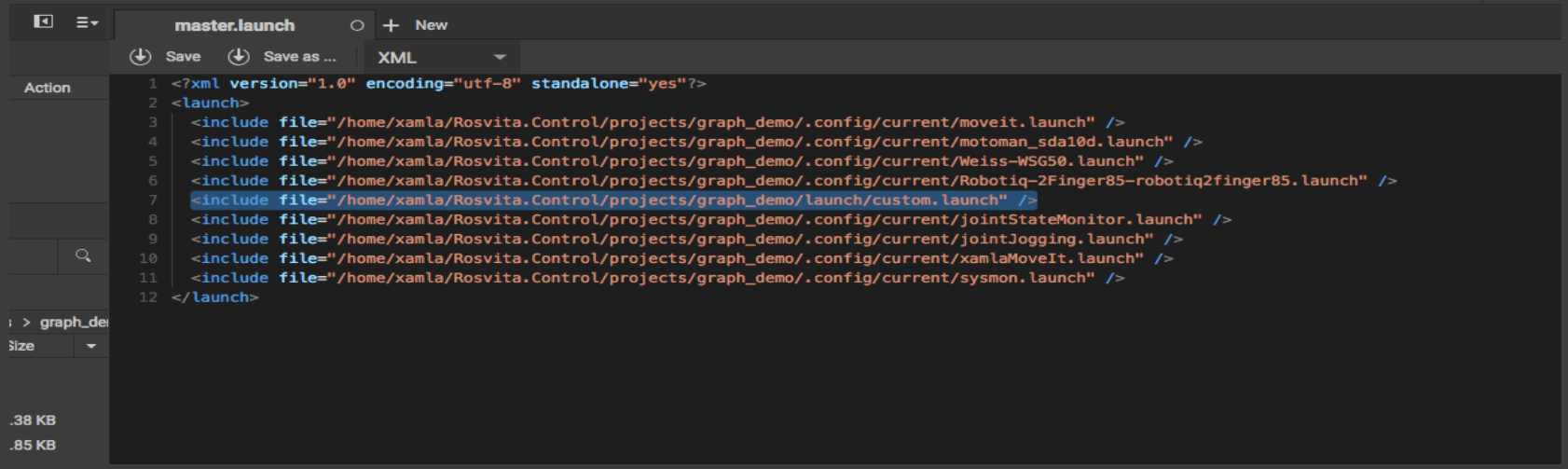
9.6 Costum launch files

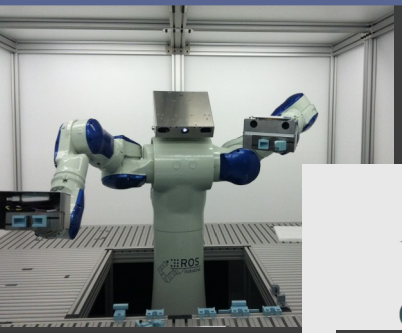
- Select ROS Launch File in rosvita library
- Open launch file
- Add any node you want rosvita to launch for you.

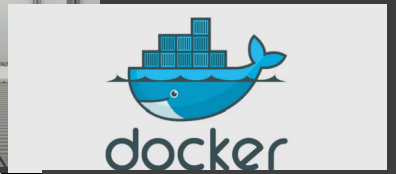


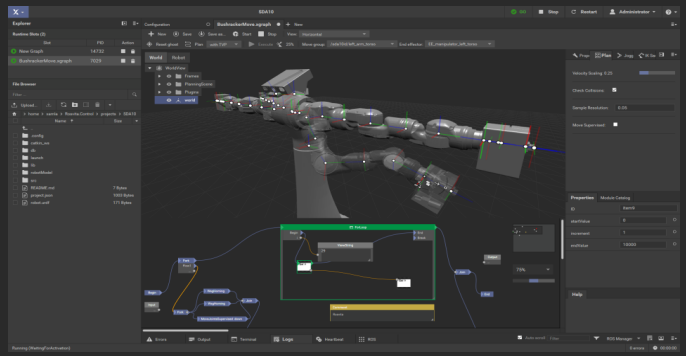
9.6 Costum launch files

- Select ROS Launch File in rosvita library
- Open launch file
- Add any node you want rosvita to launch for you.









Rosvita

10.1 ROSVITA

- Based on ROS
- Library of robot parts and drivers included
- MotionPlanning
  - MoveIt configuration and extras
- Custemizing on multiple levels
  - Robot parts
  - Launch files
  - Plugins

Quotes

“Technology is nothing. What's important is that you have a faith in people, that they're basically good and smart, and if you give them tools, they'll do wonderful things with them.”

Steve Jobs

xamla
robotic solutions

Thank you for your attention!