В этом руководстве рассматривается интеграция URDF модели робота OpenMANIPULATOR-P в среду Unity для визуализации поведения цифрового двойника, способного в режиме реального времени отображать конфигурацию робота.

Для моделирования движения робота используется инструмент RViz с поддержкой MoveIt.

Требования к программному обеспечению:

* Docker Engine
* Unity 2021.3.5f.

**Установка**

Шаг 0.

Скопируйте репозиторий на вашу локальную машину.

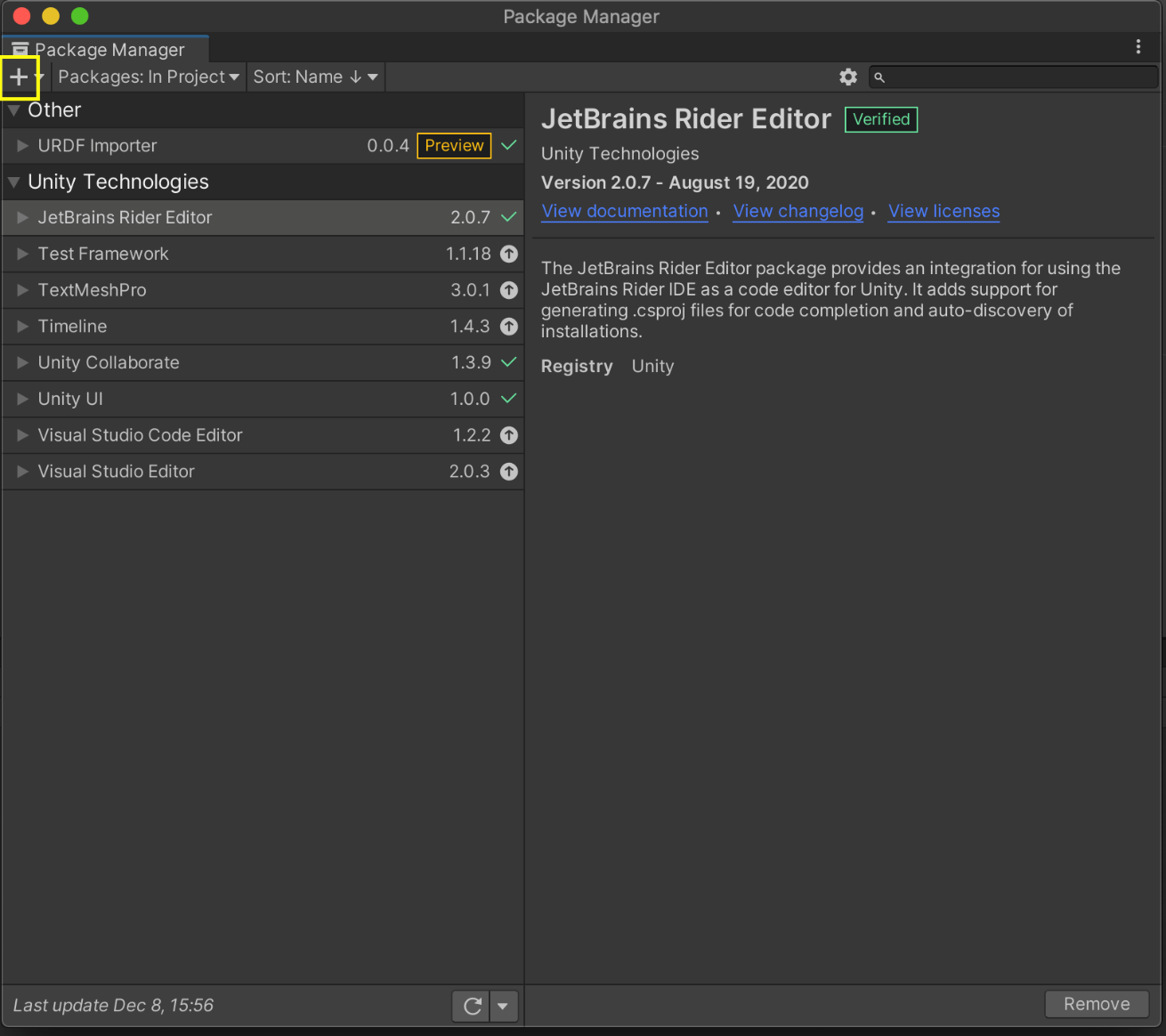
git clone --recurse-submodules https://github.com/FICompany/OpenMP.git

Шаг 1.

Откройте ваш Unity проект или создайте Новый 3D проект с помощью Unity Hub.

Шаг 2.

Откройте Package Manager в Unity Menu. Нажмите Window -> Package Manager.



Убедитесь, что в вашем проекте установлены пакеты:

* URDF-Importer
* ROS-TCP-Connector

или установите их, нажав на + и выбрав Add Package from disk.

В появившемся окне выберете файл:

PATH\TO\OpenMP\Packages\com.unity.robotics.ros-tcp-connector\package.json

Повторите предыдущие пункты для установки URDF-Importer.

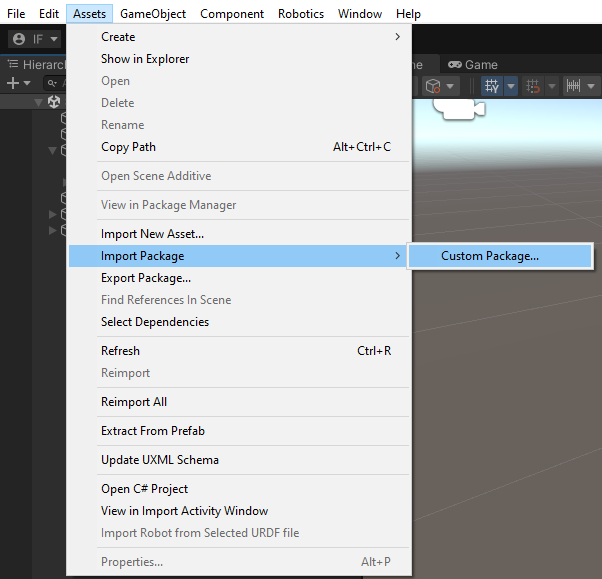
Возможна установка пакетов с помощью Git URL.

<https://github.com/Unity-Technologies/URDF-Importer.git?path=/com.unity.robotics.urdf-importer#v0.5.2>

<https://github.com/Unity-Technologies/ROS-TCP-Connector.git?path=/com.unity.robotics.ros-tcp-connector#v0.7.0>

Шаг 3.

Импортируйте assets package в ваше проект.



Выберете файл:

PATH\TO\OpenMP\UnityPackage\OpenMP.unitypackage

Шаг 4.

В окне Project среды Unity Editor найдите файл <open\_mp> и перенесите его на сцену.

Шаг 5.

Установите XLaunch (например, по ссылке <https://sourceforge.net/projects/xming/>)

1. Задайте следующие настройки:
2. 2.1. Display number 0
3. 2.2. Set Disable access control
4. 2.2. Unset Native opengl
5. 2.2. To additional parameters add: -xkblayout us,ru -xkbvariant winkeys -xkboptions grp:alt\_shift\_toggle
6. Либо запустите файл конфигурации config.xlaunch

Шаг 6.

Убедитесь, что Docker Engine запущен.

Откройте консоль и введите последовательно команды.

cd PATH\TO\OpenMP\

docker build -t open\_mp -f docker/Dockerfile .

Выполнение команды занимает значительное время.

После успешной сборки выполните:

docker run -it --rm -p 10000:10000 open\_mp /bin/bash

roslaunch openmp\_moveit digital\_clone.launch

После успешного выполнения команды откроется окно RViz с моделью робота.

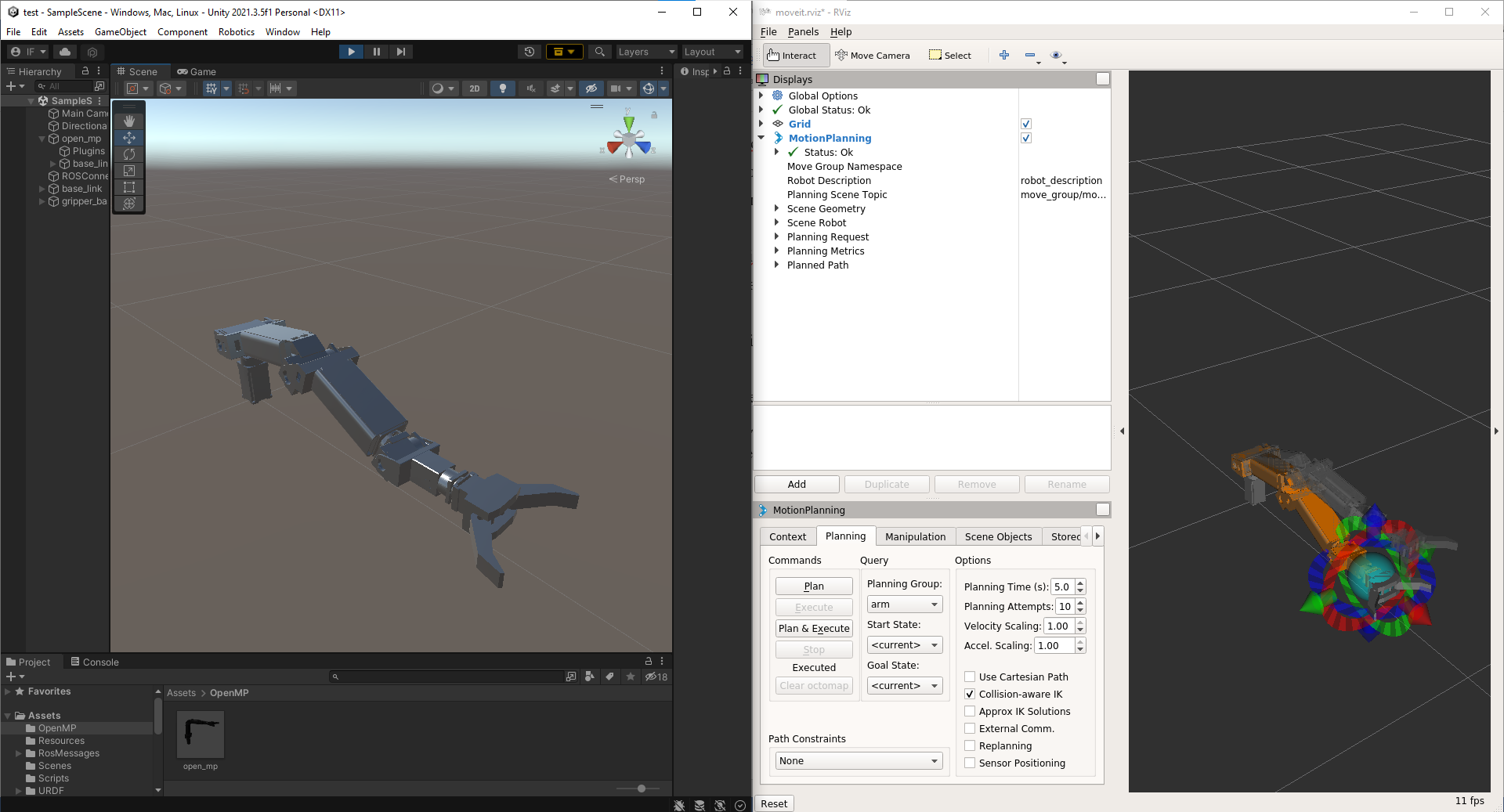
Шаг 7.

Нажмите кнопку Play в Unity Editor.

Шаг 8.

Управление роботом осуществляется в интерактивном окне.

В окне RViz выберете вкладку Planning, нажмите кнопку Plan&Execute для отправки конфигурации робота в среду Unity.



**Принцип работы проекта**

Cкрипт **JointsSubscriber** после запуска периодически принимает данные из ROS Topic "/joint\_states" и визуализирует принятую конфигурацию на сцене Unity.

На стороне ROS cреда RViz выбрала как источник данных – joints – для построения цифрового двойника.

Topic /joint\_states запускается вместе с MoveIt из пакета Sensor.

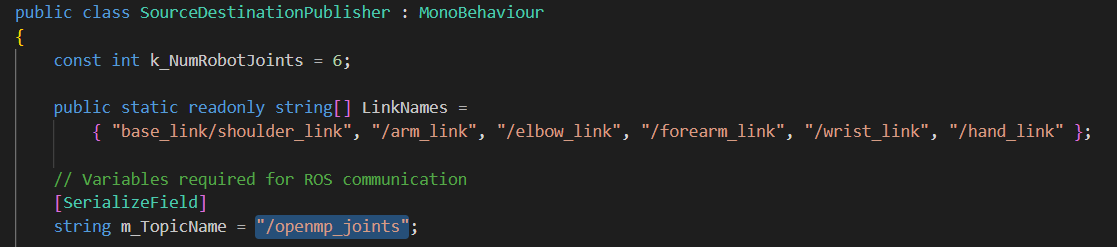
В окне RViz возможно переключить Planning Group, а также использовать вкладку Joints для точной настройки каждой кинематической пары.

**Встраивание собственной модели**

Выполнить настройку робота по URDF модели в MoveIt Setup Assistant.

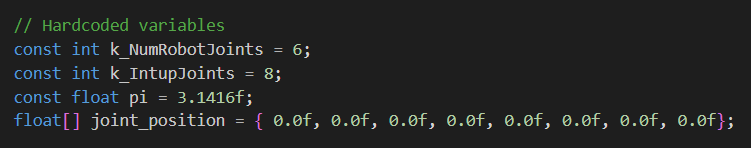
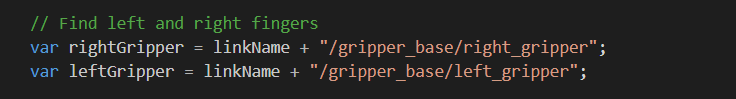
В **SourceDestinationPublisher** – изменить количество сочленений, их название, а также название Topic.

При изменении msg использовать новый класс взамен OpenmpMoveitJointsMsg.



Также в trajectory\_subscriber.py заменить OpenmpMoveitJoints на название нового msg и изменить название Topic.

В **JointsSubscriber:**

* Изменить количество сочленений до захвата и общее их количество  
  
* Изменить путь до захвата  
  
* Изменить название Topic и заменить JointStates\_Msg на новый msg при изменении источника joints

При изменении или добавлении файлов msg и srv необходимо повторно выполнить генерацию кода. Для некоторых пакетов ROS данные скрипты уже сформированы и генерировать их заново не нужно.

Кроме того, необходимо будет корректировать файлы CMakeLists и скрипты, если изменить название файлов msg и srv.

**Если осуществляется только замена модели робота на подобный по кинематике, изменять topic, msg и srv необязательно, но при этом нельзя менять их названия.**

Для конвертации данных ROS-Unity используются **RosMessages**, которые формируют классы на языке C# по файлам msg и srv из ROS.

**Для формирования кода RosMessages выполните:**

Robotics - > Generate ROS Message

В ROS message path выберете путь до папки с файлами ROS.

В built message path - папку RosMessage.

Выполните:

ROS -> moveit\_msgs -> msg -> Robot Trajectore.msg - > build msg

ROS -> openmp\_moveit - >msg -> OpenmpMoveitJoints.msg -> build msg

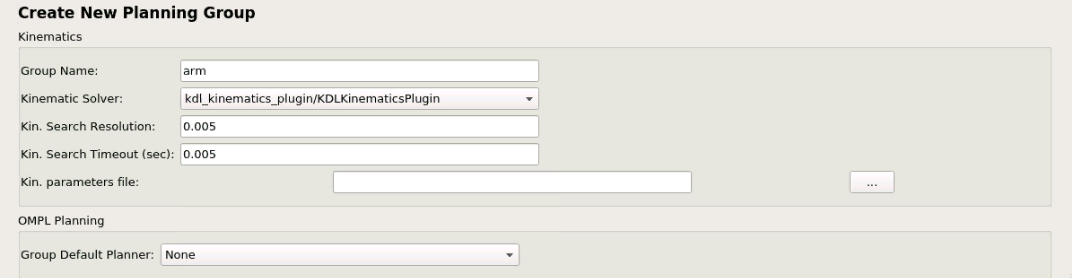
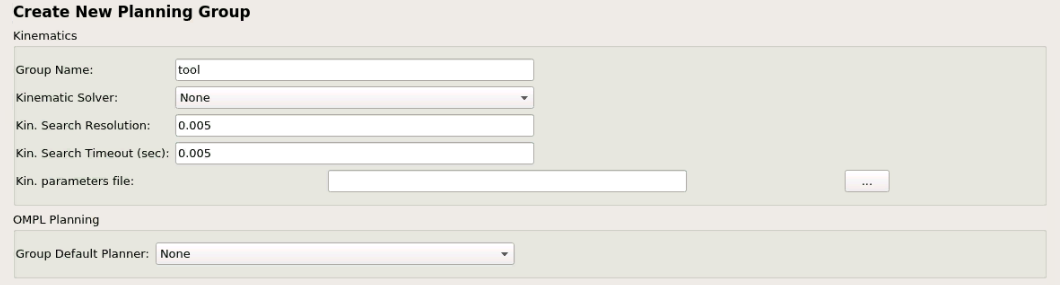
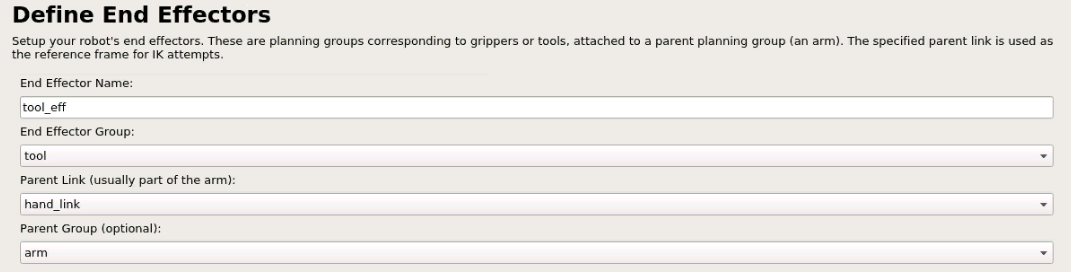
ROS -> openmp\_moveit - >msg -> OpenmpTrajectory.msg -> build msg

ROS -> openmp\_moveit - >srv ->MoverService.msg -> build msg

Чтобы **создать конфигурацию** робота по URDF модели в **MoveIt Setup Assistant** необходимо для удобства отладки иметь доступ к среде ROS с возможностью загрузки и скачивания файлов. Если выполнить настройку в контейнере Dockerпредлагаемого проекта, то после каждого удаления контейнера настройку нужно будет выполнять заново.

Как один из способов, воспользуемся **The ROS Development Studio by The Construct.**

В этом онлайн сервисе необходимо зарегистрироваться, создать новый Rosject на ROS Melodic. Пакет MoveIt в нем предустановлен.

* Загрузите в папку ROS/src ваши ROS файлы – в ROS Dev Studio через Code Editor
* При каждом изменении пакетов выполняйте команды:  
   cd /catkin\_ws  
   catkin\_make  
   source devel/setup.bash
* Для запуска MoveIt Setup Assistant выполните:   
   roslaunch moveit\_setup\_assistant setup\_assistant.launch
* Сгенерируйте матрицу Self-Collision
* Создайте Planning Group под именем **arm**Добавьте joints до захвата, так чтобы звенья образовывали кинематическую цепь  
  В данном проекте добавлены 6 joints – от joints\_1 до joints\_6
* Создайте Planning Group для описания конечного звена манипулятора  
    
  В проекте это сочленение «end\_link\_fixed»
* (Необязательно) Создайте еще две Planning Group для правого и левого захватов по аналогии с arm
* Настройте End Effector  
  
* Внесите информацию об авторе и сгенерируйте пакет (virtual joints необязательны)  
  Сгенерированные файлы вручную скопируйте в ROS/src/openmp\_moveit  
  Обратите внимание, что необходимо сохранить один launch файл и один yaml:  
  digital\_clone.launch  
  params.yaml