SERO Robotersteuerung

Generated by Doxygen 1.8.17

Chapter 1

Robotermodellierung MoveIt! Path Planning mit Gazebo

1.1 1. Clone git repository

```
git clone https://github.com/nils93/Robotermodellierung.git sero_ws && cd sero_ws
```

1.2 2. Start the setup.sh

```
./setup.sh
```

1.3 3. Enjoy the ultimate sero experience!

Chapter 2

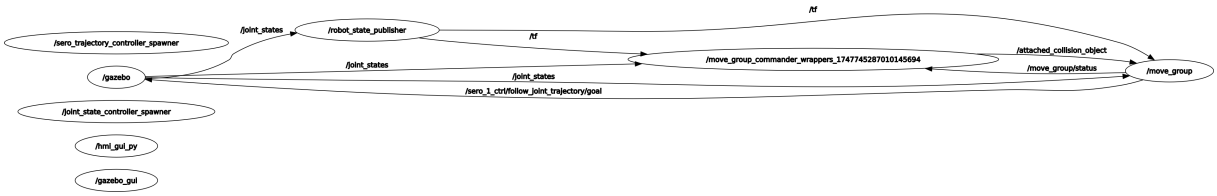
ROS Computation Graph

2.1 Description

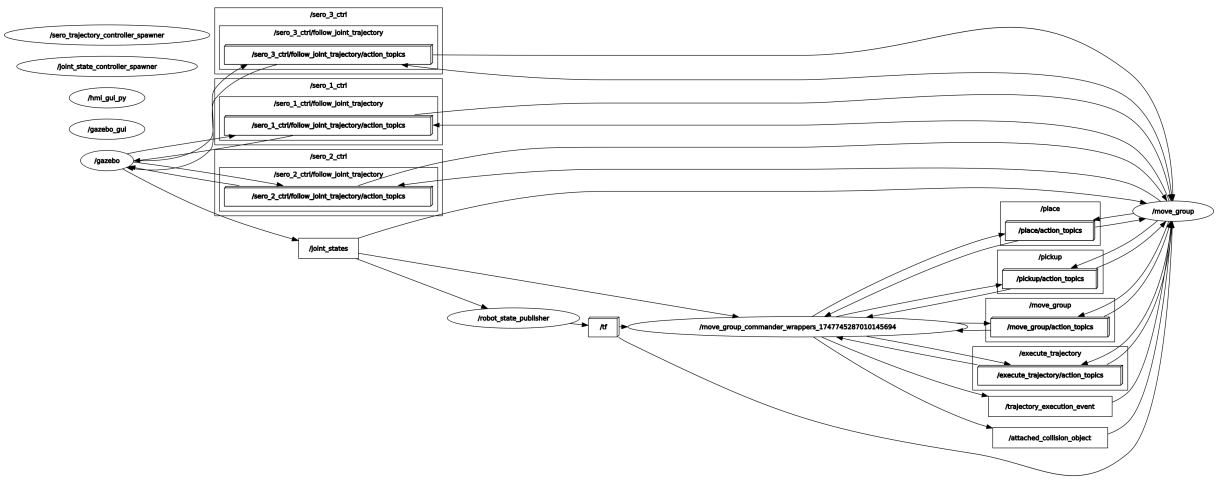
This overview shows how the main ROS components interact in the SERO robotic workcell:

- MoveIt planners
- HMI ImGui interface
- Controllers and robot descriptions
- Gazebo simulation

2.2 Simplified Graph (Nodes only)



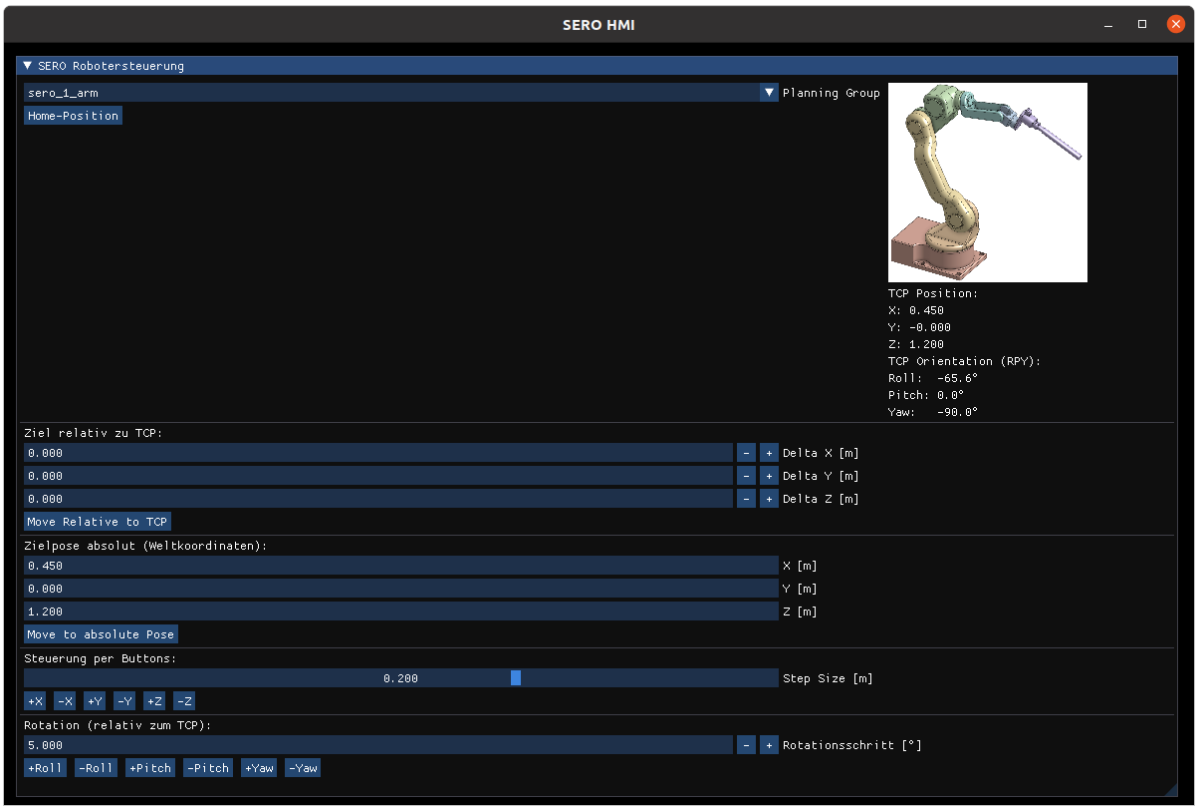
2.3 Full Computation Graph



Chapter 3

SERO HMI Interface

3.1 ImGui-Based GUI for Manual Robot Control



This image shows the graphical Human-Machine Interface (HMI) used to control the SERO robot arms. The GUI is implemented in Python using the ImGui library (pyimgui + OpenGL) and communicates with MoveIt via ROS.

Key elements:

- On the left: selection of the active planning group and a button to move to the predefined home pose.
- In the center: fields to define relative and absolute Cartesian motions (TCP-based).
- On the right: a live image of the selected robot, current TCP position, and movement confirmation.
- Below: button-based incremental movement in XYZ and RPY space.

The GUI is designed for real-time feedback and fast manual positioning during development and testing.

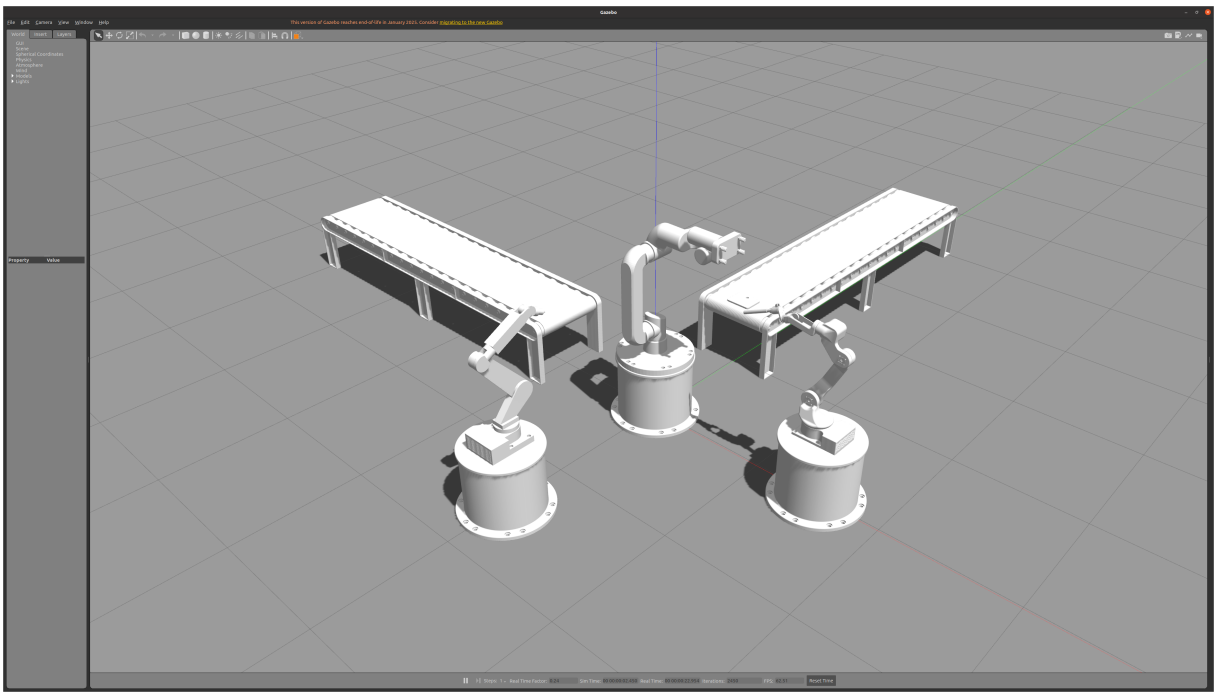
Chapter 4

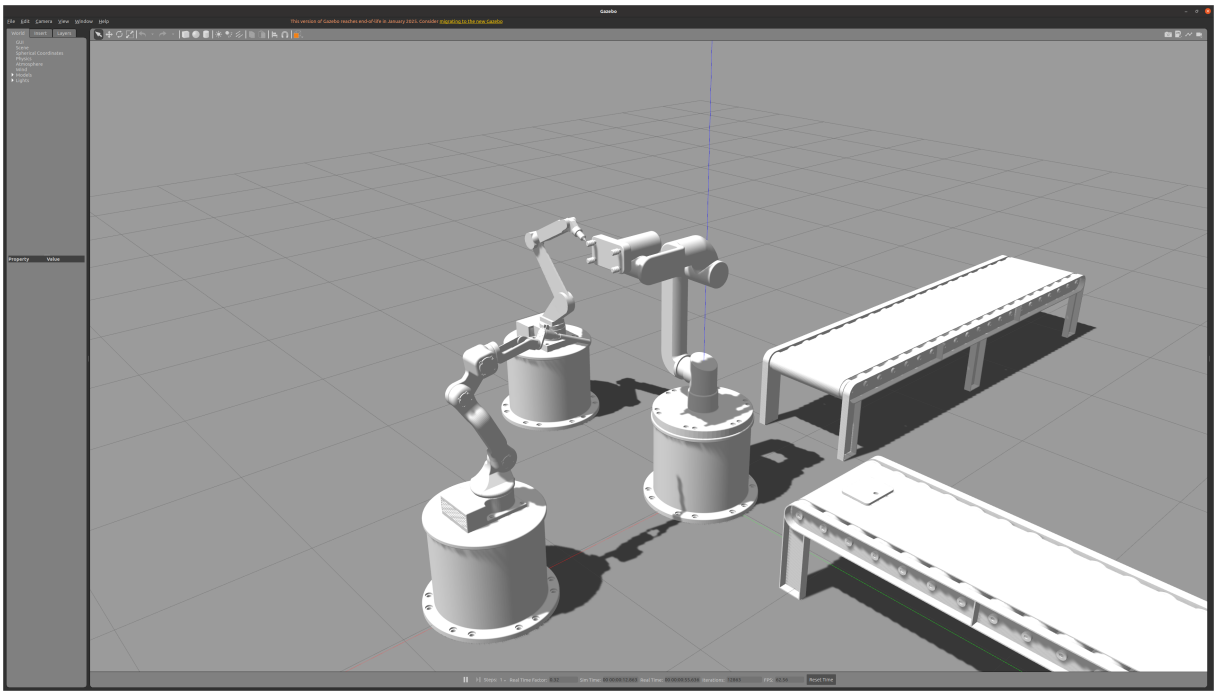
The Factory Station

4.1 Description

This overview shows our robotic station:

- Sero_1
- Sero_2
- Sero_3
- workobject





Chapter 5

Namespace Index

5.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

Chapter 7

Namespace Documentation

7.1 hmi_gui Namespace Reference

Functions

- def move_to_home (group_name)
- def move_relative_rpy (group, droll_deg, dpitch_deg, dyaw_deg)
- def move_relative (group, dx, dy, dz)
- def move_to_absolute_pose (group, pose)
- def load_texture_from_png (path)

Variables

- list planning_groups = ["sero_1_arm", "sero_2_arm", "sero_3_arm"]
- list tcp_links = ["sero_1_tcp", "sero_2_tcp", "sero_3_tcp"]
- int current_index = 0
- group = moveit_commander.MoveGroupCommander(planning_groups[current_index])
- float relative_x = 0.0
- float relative_y = 0.0
- float relative_z = 0.0
- float step_size = 0.2
- window = glfw.create_window(1400, 800, "SERO HMI", None, None)
- impl = GlfwRenderer(window)
- pkg_dir = os.path.dirname(os.path.abspath(__file__))
- dictionary image_paths
- dictionary textures = {}
- tex_id
- width
- height
- changed
- list group_name = planning_groups[current_index]
- list current_group_name = planning_groups[current_index]
- w
- h
- current_pose = group.get_current_pose(tcp_links[current_index]).pose
- q = current_pose.orientation
- roll

- pitch
- yaw
- int eps = 1e-2
- step
- abs_pose = Pose()
- x
- y
- z
- orientation
- list move = [0.0, 0.0, 0.0]
- base_pose = group.get_current_pose(tcp_links[current_index]).pose
- target = Pose()
- success = group.plan()
- wait
- float rot_step = 5.0
- list rpy_move = [0.0, 0.0, 0.0]

7.1.1 Function Documentation

7.1.1.1 load_texture_from_png()

```
def hmi_gui.load_texture_from_png (
              path )
```

```
@brief Loads a PNG image as an OpenGL texture for ImGui.

@param path Absolute path to the PNG image file.
@return (texture_id, width, height) tuple.
```

7.1.1.2 move_relative()

```
def hmi_gui.move_relative (
              group,
              dx,
              dy,
              dz )
```

```
@brief Moves the robot TCP relatively in Cartesian space.

@param group MoveGroupCommander instance.
@param dx Relative X offset in meters.
@param dy Relative Y offset in meters.
@param dz Relative Z offset in meters.
```

7.1.1.3 move_relative_rpy()

```
def hmi_gui.move_relative_rpy (
              group,
              droll_deg,
              dpitch_deg,
              dyaw_deg )
```

```
@brief Rotates the robot TCP relative to its current orientation.

@param group MoveGroupCommander instance.
@param droll_deg Roll offset in degrees.
@param dpitch_deg Pitch offset in degrees.
@param dyaw_deg Yaw offset in degrees.
```

7.1.1.4 move_to_absolute_pose()

```
def hmi_gui.move_to_absolute_pose (
              group,
              pose )
```

```
@brief Moves the robot TCP to a given absolute target pose.

@param group MoveGroupCommander instance.
@param pose Target geometry_msgs/Pose object in world coordinates.
```

7.1.1.5 move_to_home()

```
def hmi_gui.move_to_home (
              group_name )
```

```
@brief Moves the selected robot to its predefined home pose.

@param group_name Name of the MoveIt planning group (e.g. "sero_1_arm").
```

7.1.2 Variable Documentation

7.1.2.1 abs_pose

```
hmi_gui.abs_pose = Pose()
```

7.1.2.2 base_pose

```
hmi_gui.base_pose = group.get_current_pose(tcp_links[current_index]).pose
```

7.1.2.3 changed

```
hmi_gui.changed
```

7.1.2.4 current_group_name

```
list hmi_gui.current_group_name = planning_groups[current_index]
```

7.1.2.5 current_index

```
hmi_gui.current_index = 0
```

7.1.2.6 current_pose

```
hmi_gui.current_pose = group.get_current_pose(tcp_links[current_index]).pose
```

7.1.2.7 eps

```
int hmi_gui.eps = 1e-2
```

7.1.2.8 group

```
hmi_gui.group = moveit_commander.MoveGroupCommander(planning_groups[current_index])
```

7.1.2.9 group_name

```
list hmi_gui.group_name = planning_groups[current_index]
```

7.1.2.10 h

```
hmi_gui.h
```

7.1.2.11 height

```
hmi_gui.height
```

7.1.2.12 image_paths

```
dictionary hmi_gui.image_paths
```

Initial value:

```
1 =  {
2     "sero_1_arm": os.path.join(pkg_dir, "../resources/sero_1_arm.png"),
3     "sero_2_arm": os.path.join(pkg_dir, "../resources/sero_2_arm.png"),
4     "sero_3_arm": os.path.join(pkg_dir, "../resources/sero_3_arm.png")
5 }
```

7.1.2.13 impl

```
hmi_gui.impl = GlfwRenderer(window)
```

7.1.2.14 move

```
list hmi_gui.move = [0.0, 0.0, 0.0]
```

7.1.2.15 orientation

```
hmi_gui.orientation
```

7.1.2.16 pitch

```
hmi_gui.pitch
```

7.1.2.17 pkg_dir

```
hmi_gui.pkg_dir = os.path.dirname(os.path.abspath(__file__))
```

7.1.2.18 planning_groups

```
list hmi_gui.planning_groups = ["sero_1_arm", "sero_2_arm", "sero_3_arm"]
```

7.1.2.19 q

```
hmi_gui.q = current_pose.orientation
```

7.1.2.20 relative_x

```
hmi_gui.relative_x = 0.0
```

7.1.2.21 relative_y

```
hmi_gui.relative_y = 0.0
```

7.1.2.22 relative_z

```
hmi_gui.relative_z = 0.0
```

7.1.2.23 roll

```
hmi_gui.roll
```

7.1.2.24 rot_step

```
hmi_gui.rot_step = 5.0
```

7.1.2.25 rpy_move

```
list hmi_gui.rpy_move = [0.0, 0.0, 0.0]
```

7.1.2.26 step

```
hmi_gui.step
```

7.1.2.27 step_size

```
hmi_gui.step_size = 0.2
```

7.1.2.28 success

```
hmi_gui.success = group.plan()
```

7.1.2.29 target

```
hmi_gui.target = Pose()
```

7.1.2.30 tcp_links

```
list hmi_gui.tcp_links = ["sero_1_tcp", "sero_2_tcp", "sero_3_tcp"]
```

7.1.2.31 tex_id

```
hmi_gui.tex_id
```

7.1.2.32 textures

```
dictionary hmi_gui.textures = {}
```

7.1.2.33 w

```
hmi_gui.w
```

7.1.2.34 wait

```
hmi_gui.wait
```

7.1.2.35 width

```
hmi_gui.width
```

7.1.2.36 window

```
hmi_gui.window = glfw.create_window(1400, 800, "SERO HMI", None, None)
```

7.1.2.37 x

```
hmi_gui.x
```

7.1.2.38 y

```
hmi_gui.y
```

7.1.2.39 yaw

```
hmi_gui.yaw
```

7.1.2.40 z

```
hmi_gui.z
```

7.2 pathplanning Namespace Reference

Functions

- def move_to_named_target (group_name, target_name)
- def move_to_pose (group_name, pose)
- def move_to_position (group_name, x, y, z)
- def create_pose (name, x, y, z, roll_deg, pitch_deg, yaw_deg)
- def move_to_joint_positions_deg (group_name, joint_values_deg)

Variables

- anonymous

7.2.1 Function Documentation

7.2.1.1 create_pose()

```
def pathplanning.create_pose (
              name,
              x,
              y,
              z,
              roll_deg,
              pitch_deg,
              yaw_deg )
```

```
Creates and returns a geometry_msgs Pose from position and orientation (RPY in degrees).
@param name Optional label for logging
@param x,y,z Cartesian coordinates
@param roll_deg, pitch_deg, yaw_deg Orientation in degrees
@return geometry_msgs.msg.Pose
```

7.2.1.2 move_to_joint_positions_deg()

```
def pathplanning.move_to_joint_positions_deg (
              group_name,
              joint_values_deg )
```

```
Moves the robot arm to specific joint positions given in degrees.
@param group_name MoveIt planning group
@param joint_values_deg List of joint angles in degrees

Args:
    group_name (str): MoveIt planning group (e.g., "sero_1_arm")
    joint_values_deg (list of float): Joint angles in degrees
```

7.2.1.3 move_to_named_target()

```
def pathplanning.move_to_named_target (
              group_name,
              target_name )
```

```
Moves the robot to a predefined named target.
@param group_name MoveIt group name (e.g. "sero_1_arm")
@param target_name Name of the predefined target
```

7.2.1.4 move_to_pose()

```
def pathplanning.move_to_pose (
              group_name,
              pose )
```

```
Plans and executes a motion to the specified absolute pose.
@param group_name MoveIt group name
@param pose Target pose as geometry_msgs/Pose
```

7.2.1.5 move_to_position()

```
def pathplanning.move_to_position (
              group_name,
              x,
              y,
              z )
```

```
Moves the robot to a target position (x, y, z) with no strict orientation constraint,
using set_approximate_joint_value_target() for better IK compatibility with KDL.
@param group_name MoveIt group
@param x X position [m]
@param y Y position [m]
@param z Z position [m]
Args:
    group_name (str): MoveIt planning group
    x, y, z (float): Target position in meters

Returns:
    bool: True if successful, False otherwise
```

7.2.2 Variable Documentation

7.2.2.1 anonymous

```
pathplanning.anonymous
```

7.3 pathplanning_cmd Namespace Reference

Functions

- def move_to_position (group_name, x, y, z)

  Moves the specified MoveIt group to a 3D target position (x, y, z).

Variables

- group_name = sys.argv[1]
- x = float(sys.argv[2])
- y = float(sys.argv[3])
- z = float(sys.argv[4])

7.3.1 Function Documentation

7.3.1.1 move_to_position()

```
def pathplanning_cmd.move_to_position (
              group_name,
              x,
              y,
              z )
```

Moves the specified MoveIt group to a 3D target position (x, y, z).

Parameters

group_name	Name of the MoveIt planning group (e.g., "`sero_3_arm`").
x	X-coordinate in meters
y	Y-coordinate in meters
z	Z-coordinate in meters

- A dummy quaternion (w = 1) is used for orientation
- Orientation tolerance is set to (any orientation allowed)
- Position tolerance is set to 1 cm
- Uses `set_pose_target()` and `go(wait=True)` to execute motion

Returns

Logs success/failure to ROS log output

7.3.2 Variable Documentation

7.3.2.1 group_name

```
pathplanning_cmd.group_name = sys.argv[1]
```

7.3.2.2 x

```
pathplanning_cmd.x = float(sys.argv[2])
```

7.3.2.3 y

```
pathplanning_cmd.y = float(sys.argv[3])
```

7.3.2.4 z

```
pathplanning_cmd.z = float(sys.argv[4])
```

7.4 sero_multi_station Namespace Reference

Moves a robot arm between predefined named targets and optionally to custom poses, positions, or joint configurations.

7.4.1 Detailed Description

Moves a robot arm between predefined named targets and optionally to custom poses, positions, or joint configurations.

Moves a robot arm to a 3D position with neutral orientation using MoveIt.

This script uses MoveIt to execute a typical sequence for a multi-robot cell. It includes named targets and helper functions for absolute poses, Cartesian positions, and joint values.

@requires rospy @requires moveit_commander @requires geometry_msgs.msg @requires tf.transformations

This script is intended for basic position-only control (ignoring orientation). It sets a pose goal with w=1.0 and disables orientation constraints.

@requires rospy @requires moveit_commander @requires geometry_msgs.msg

Chapter 8

File Documentation

8.1 doc/gazebo_station.dox File Reference

8.2 doc/hmi_interface.dox File Reference

8.3 doc/ros_overview.dox File Reference

8.4 README.md File Reference

8.5 setup.sh File Reference

8.6 src/sero_hmi/CMakeLists.txt File Reference

8.7 src/sero_multi_station/CMakeLists.txt File Reference

8.8 src/sero_multi_station_moveit_config/CMakeLists.txt File Reference

Functions

- cmake_minimum_required (VERSION 3.1.3) project(sero_multi_station_moveit_config) find_package(catkin REQUIRED) catkin_package() install(DIRECTORY launch DESTINATION $

8.8.1 Function Documentation

8.8.1.1 cmake_minimum_required()

```
cmake_minimum_required (
             VERSION 3.1.  3 )
```

8.9 src/station_peripherals/CMakeLists.txt File Reference

8.10 src/sero_hmi/package.xml File Reference

8.11 src/sero_multi_station/package.xml File Reference

8.12 src/sero_multi_station_moveit_config/package.xml File Reference

8.13 src/station_peripherals/package.xml File Reference

8.14 src/sero_hmi/scripts/hmi_gui.py File Reference

Namespaces

- hmi_gui

Functions

- def hmi_gui.move_to_home (group_name)
- def hmi_gui.move_relative_rpy (group, droll_deg, dpitch_deg, dyaw_deg)
- def hmi_gui.move_relative (group, dx, dy, dz)
- def hmi_gui.move_to_absolute_pose (group, pose)
- def hmi_gui.load_texture_from_png (path)

Variables

- list hmi_gui.planning_groups = ["sero_1_arm", "sero_2_arm", "sero_3_arm"]
- list hmi_gui.tcp_links = ["sero_1_tcp", "sero_2_tcp", "sero_3_tcp"]
- int hmi_gui.current_index = 0
- hmi_gui.group = moveit_commander.MoveGroupCommander(planning_groups[current_index])
- float hmi_gui.relative_x = 0.0
- float hmi_gui.relative_y = 0.0
- float hmi_gui.relative_z = 0.0
- float hmi_gui.step_size = 0.2
- hmi_gui.window = glfw.create_window(1400, 800, "SERO HMI", None, None)
- hmi_gui.impl = GlfwRenderer(window)
- hmi_gui.pkg_dir = os.path.dirname(os.path.abspath(__file__))
- dictionary hmi_gui.image_paths
- dictionary hmi_gui.textures = {}

- hmi_gui.tex_id
- hmi_gui.width
- hmi_gui.height
- hmi_gui.changed
- list hmi_gui.group_name = planning_groups[current_index]
- list hmi_gui.current_group_name = planning_groups[current_index]
- hmi_gui.w
- hmi_gui.h
- hmi_gui.current_pose = group.get_current_pose(tcp_links[current_index]).pose
- hmi_gui.q = current_pose.orientation
- hmi_gui.roll
- hmi_gui.pitch
- hmi_gui.yaw
- int hmi_gui.eps = 1e-2
- hmi_gui.step
- hmi_gui.abs_pose = Pose()
- hmi_gui.x
- hmi_gui.y
- hmi_gui.z
- hmi_gui.orientation
- list hmi_gui.move = [0.0, 0.0, 0.0]
- hmi_gui.base_pose = group.get_current_pose(tcp_links[current_index]).pose
- hmi_gui.target = Pose()
- hmi_gui.success = group.plan()
- hmi_gui.wait
- float hmi_gui.rot_step = 5.0
- list hmi_gui.rpy_move = [0.0, 0.0, 0.0]

8.15 src/sero_multi_station/config/joint_state_controller.yaml File Reference

8.16 src/sero_multi_station/config/trajectory_controller.yaml File Reference

8.17 src/sero_multi_station/launch/bringup_moveit.launch File Reference

8.18 src/sero_multi_station/launch/bringup_moveit_just_sim.launch File Reference

8.19 src/sero_multi_station/launch/control_utils.launch File Reference

8.20 src/sero_multi_station/launch/factory_station.launch File Reference

8.21 src/sero_multi_station/launch/sero_multi_station_empty_↩ world.launch File Reference

8.22 src/sero_multi_station/robot_description/sero_multi_station.urdf File Reference

8.23 src/sero_multi_station/scripts/pathplanning.py File Reference

Namespaces

- pathplanning
- sero_multi_station

  Moves a robot arm between predefined named targets and optionally to custom poses, positions, or joint configurations.

Functions

- def pathplanning.move_to_named_target (group_name, target_name)
- def pathplanning.move_to_pose (group_name, pose)
- def pathplanning.move_to_position (group_name, x, y, z)
- def pathplanning.create_pose (name, x, y, z, roll_deg, pitch_deg, yaw_deg)
- def pathplanning.move_to_joint_positions_deg (group_name, joint_values_deg)

Variables

- pathplanning.anonymous

8.24 src/sero_multi_station/scripts/pathplanning_cmd.py File Reference

Namespaces

- pathplanning_cmd
- sero_multi_station

  Moves a robot arm between predefined named targets and optionally to custom poses, positions, or joint configurations.

Functions

- def pathplanning_cmd.move_to_position (group_name, x, y, z)

  Moves the specified MoveIt group to a 3D target position (x, y, z).

Variables

- pathplanning_cmd.group_name = sys.argv[1]
- pathplanning_cmd.x = float(sys.argv[2])
- pathplanning_cmd.y = float(sys.argv[3])
- pathplanning_cmd.z = float(sys.argv[4])

Index