Welcome to robot_sim's documentation!

Indices and tables

robot_sim documentation!

Here is the documentation of the robot_sim package!

goal_set Module

The ROS Node 1

scripts.goal_set.cancelGoal(*client*) [source]

Cancels the current goal sent to the action server.

Parameters **client** (*actionlib.SimpleActionClient*) – The action client.

scripts.goal_set.createClient() [source]

Creates and returns a SimpleActionClient for the 'reaching_goal' action server.

Returns The created action client.
Return type actionlib.SimpleActionClient

scripts.goal_set.error() [source]

Displays an error message for invalid input.

scripts.goal_set.sendGoal(*client*, *x*, *y*) [source]

Sends a goal request to the action server with the specified x and y coordinates.

Parameters
- **client** (*actionlib.SimpleActionClient*) – The action client.
- **x** (*float*) – The x-coordinate of the goal position.

- **y** (*float*) – The y-coordinate of the goal position.

scripts.goal_set.showWelcome() [source]

Clears the screen and displays the welcome message.

goal_service Module

The ROS Node 2

scripts.goal_service.sendGoalSummary(*res*) [source]

Service callback function for sending the goal summary.

Parameters	**res** (*robot_sim.srv.GoalSummaryRequest*) – The request object for the 'goalSummary' service.
Returns	A list containing the number of goals reached and goals cancelled.
Return type	list

scripts.goal_service.updateGoalSummary(*msg*) [source]

Callback function for updating the goal summary based on the received feedback message.

Parameters	**msg** (*robot_sim.msg.PlanningActionFeedback*) – The feedback message received from the 'reaching_goal' action server.

robot_monitoring Module

The ROS Node 3

scripts.robot_monitoring.movingAverage(*vx*, *vy*) [source]

Calculates the moving average of velocities for the given vx and vy values.

Parameters	• **vx** (*float*) – The current velocity in the x-direction. • **vy** (*float*) – The current velocity in the y-direction.
Returns	A list containing the average velocity in the x-direction and the average velocity in the y-direction.
Return type	list

scripts.robot_monitoring.updateKinematicInfo(*odoMsg*) [source]

Updates the kinematic information based on the received odometer message.

Parameters **odoMsg** (*robot_sim.msg.OdoSensor*) – The odometer message containing the current position and velocity information.

odometer Module

The ROS Node 4

scripts.odometer.setOdoMessage(*msg*) [source]

Callback function for setting the odometry message values.

Parameters **msg** (*nav_msgs.msg.Odometry*) – The received odometry message.