

G-BEAM: implemented on ROS

Leonardo Cecchin

May 20, 2021

Abstract

This document contains an overview of the packages used in the ROS simulation and experimental setup for testing G-BEAM controller, as well as other approaches.

The document is structured in the following way: chapter 1 gives an overview of the setup, showing how different packages interface with each other, while chapter 2, chapter 3, chapter 3 and chapter 5 give a detailed explanation of the various packages, listing all nodes, topics and messages provided and used.

Contents

1	Packages setup	2
1.1	Scheme	2
1.2	Launchers	2
1.2.1	Experimental setup, DJI S1000	2
1.2.2	Simulation setup, Turtlebot	2
2	GBEAM Controller Package	5
2.1	Scheme	5
2.2	Nodes	5
2.2.1	Polytope generation node	5
2.2.2	Graph update node	6
2.2.3	Exploration node	6
2.3	Launch files	6
2.3.1	Gbeam controller	6
2.3.2	Gbeam s1000 parameters	7
3	GBEAM Library Package	8
3.1	Messages	8
3.1.1	Vertex	8
3.1.2	FreePolygon	8
3.1.3	FreePolygonStamped	8
3.1.4	GraphEdge	8
3.1.5	ReachabilityGraph	9
3.1.6	PolyArea	9
3.2	Polytope functions	9
3.2.1	fn	9
3.3	Graph functions	9
3.4	Exploration functions	9
4	GBEAM Ground Package	10
4.1	Scheme	10
4.2	Nodes	10
4.2.1	Polytope drawer node	10
4.2.2	Graph drawer node	10
4.3	Launch files	11
4.3.1	Gbeam ground	11
5	S1000 Interface Package	12
5.1	Scheme	12
5.2	Nodes	13
5.2.1	Lidar Merger	13
5.2.2	Tf2 Publisher	13
5.2.3	Bag Recorder	13
5.2.4	Position Controller	13
5.2.5	Obstacle Avoidance	13
5.3	Launch files	14
5.3.1	S1000 Interface	14

Chapter 1

Packages setup

This chapter gives an overview of the system architecture.

1.1 Scheme

In Figure 1.1 is shown a diagram representing how different packages of this setup interact with each other.

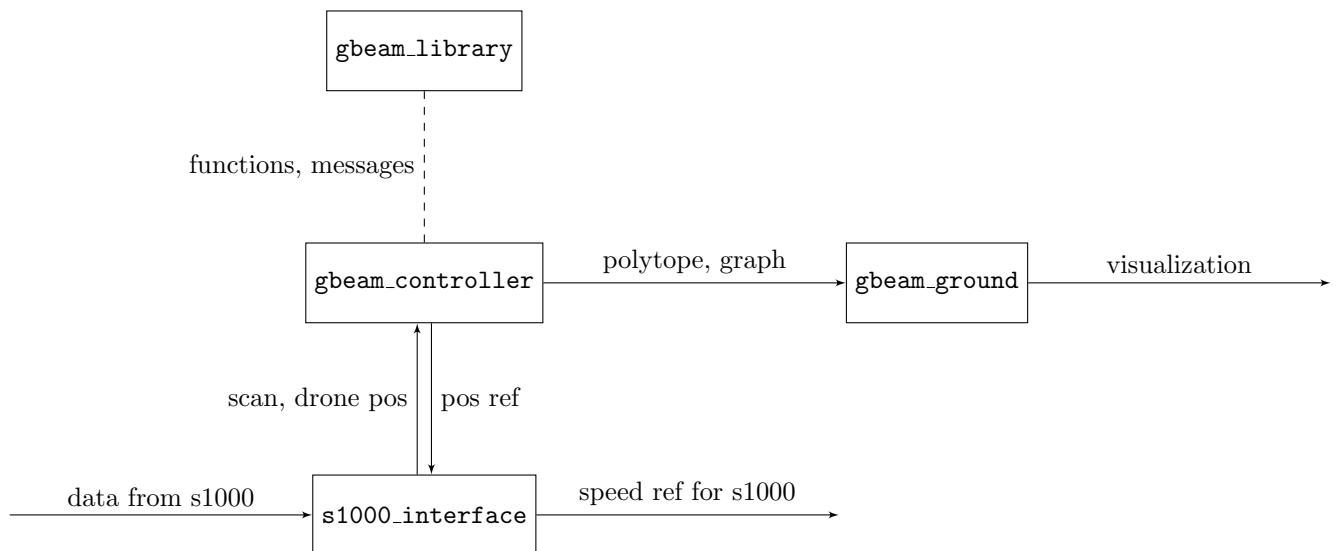


Figure 1.1: Diagram of interactions between packages

1.2 Launchers

1.2.1 Experimental setup, DJI S1000

The automatic start is done by the service (`systemctl`) `ros.s1000`.
This launches the launch file `root/autostart/s1000startup.launch`.

1.2.2 Simulation setup, Turtlebot

This simulation environment uses Gazebo to simulate a turtlebot differential drive robot, equipped with a 360° LiDAR.

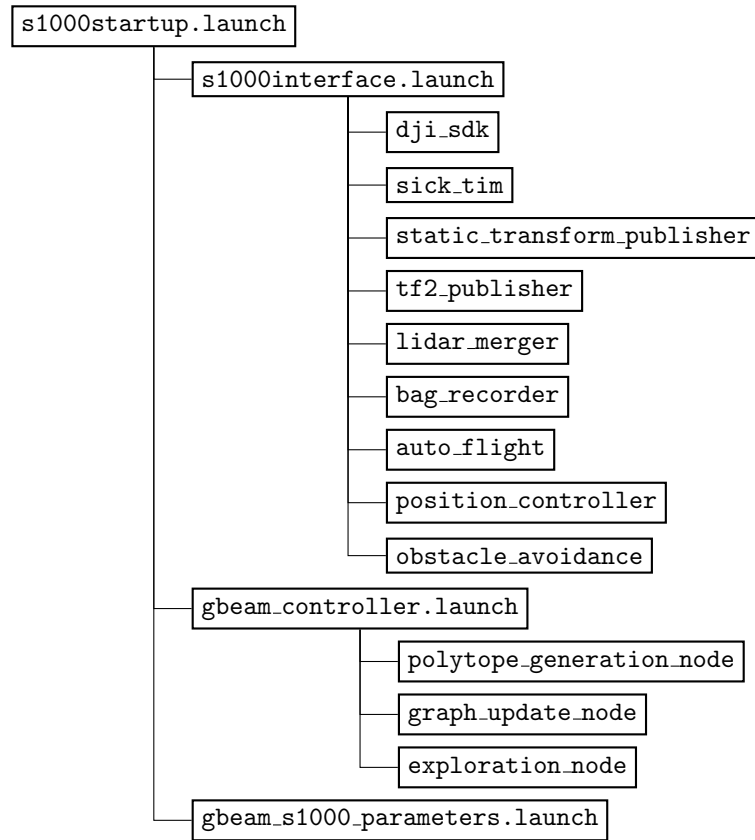


Figure 1.2: Diagram of onboard launchers for DJI S1000 experimental setup

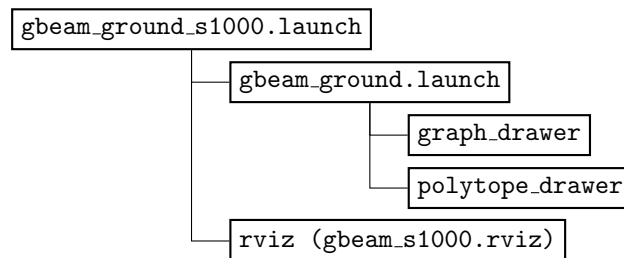


Figure 1.3: Diagram of ground station launchers for DJI S1000 experimental setup

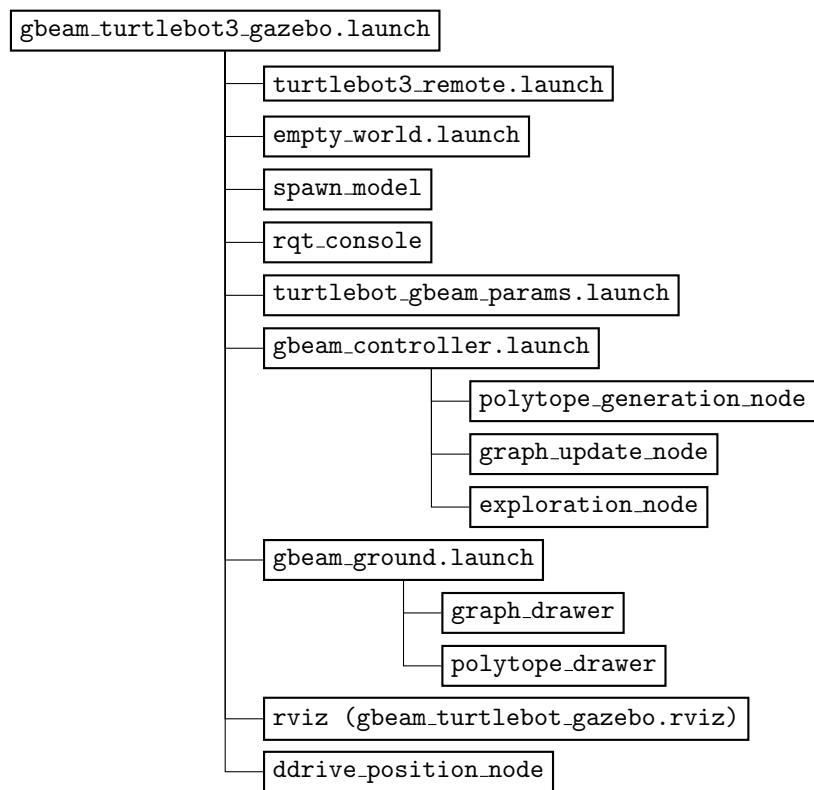


Figure 1.4: Diagram of launchers for Turtlebot simulation setup

Chapter 2

GBEAM Controller Package

This chapter describes the components of the “gbeam_controller” ROS package.

2.1 Scheme

Here an overview of the control scheme

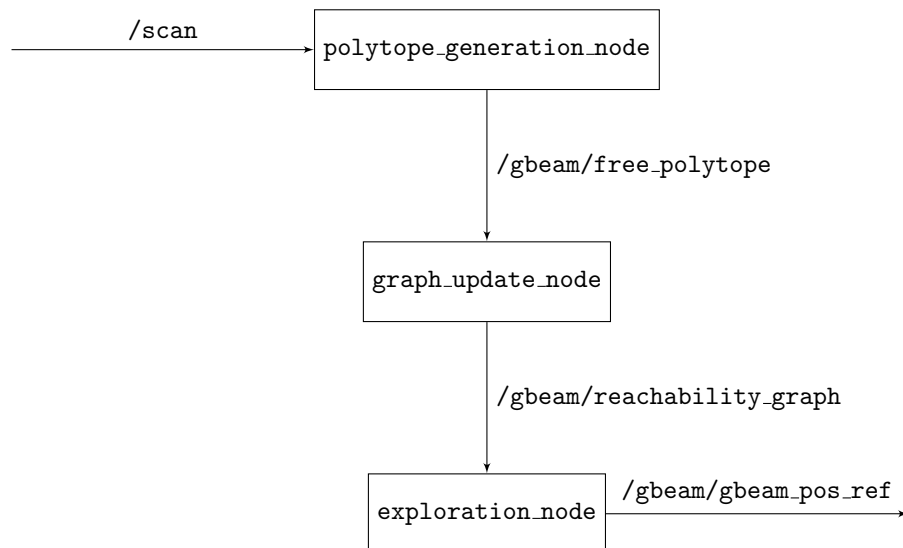


Figure 2.1: Diagram of gbeam controller package

2.2 Nodes

2.2.1 Polytope generation node

Subscribed topics

Topic	Datatype
/scan	sensor_msgs::LaserScan

Published topics

Topic	Datatype	Rate
/gbeam/free_polytope	gbeam_library::FreePolygonStamped	Same as /scan

Parameters

Parameter	Datatype
/gbeam_controller/polytope_generation_param/num_vertices	int
/gbeam_controller/polytope_generation_param/distance_step	double
/gbeam_controller/polytope_generation_param/start_distance	double
/gbeam_controller/polytope_generation_param/polytope_convex	bool
/gbeam_controller/polytope_generation_param/vertex_ostacle_dist	double
/gbeam_controller/robot_param/safe_dist	double

2.2.2 Graph update node

Subscribed topics

Topic	Datatype
/gbeam/free_polytope	gbeam_library::FreePolygonStamped

The node also looks up the transform from odom (i.e. the fixed local map frame) to **base_scan** (i.e. the frame fixed to the LiDAR sensor)

Published topics

Topic	Datatype	Rate
/gbeam/reachability_graph	gbeam_library::ReachabilityGraph	Same as polytope

Parameters

Parameter	Datatype
/gbeam_controller/graph_update_param/node_dist_min	double
/gbeam_controller/graph_update_param/node_dist_open	double
/gbeam_controller/graph_update_param/node_bound_dist	double
/gbeam_controller/robot_param/safe_dist	double

2.2.3 Exploration node

Subscribed topics

Topic	Datatype
/gbeam/reachability_graph	gbeam_library::ReachabilityGraph

Published topics

Topic	Datatype	Rate
/gbeam/gbeam_pos_ref	geometry_msgs::PoseStamped	Previous target reached

Parameters

Parameter	Datatype
/gbeam_controller/exploration_param/reached_tol	double
/gbeam_controller/exploration_param/limit_xi	double
/gbeam_controller/exploration_param/limit_xs	double
/gbeam_controller/exploration_param/limit_yi	double
/gbeam_controller/exploration_param/limit_ys	double

2.3 Launch files

2.3.1 Gbeam controller

Nodes

Package	Type	Name
gbeam_controller	polytope_generation_node	polytope_gen
gbeam_controller	graph_update_node	graph_update
gbeam_controller	exploration_node	exploration

2.3.2 Gbeam s1000 parameters

Parameters

Parameter	Datatype
/gbeam.controller/robot_param/safe_dist	double
/gbeam.controller/polytope_generation_param/num_vertices	int
/gbeam.controller/polytope_generation_param/start_distance	double
/gbeam.controller/polytope_generation_param/distance_step	double
/gbeam.controller/polytope_generation_param/vertex_obstacle_dist	double
/gbeam.controller/graph_update_param/node_dist_min	double
/gbeam.controller/graph_update_param/node_dist_open	double
/gbeam.controller/graph_update_param/node_bound_dist	double

Chapter 3

GBEAM Library Package

This chapter describes the components of the “`gbeam_library`” ROS package. This package does not contain any nodes, instead it contains messages and functions used in the `gbeam_controller` package.

3.1 Messages

3.1.1 Vertex

Attributes

Type	Name
int64	id
float32	x
float32	y
float32	z
float32	gain
geometry_msgs/Vector3	obstacle_normal
bool	is_obstacle
bool	is_visited
bool	is_reachable
bool	is_completely_connected

3.1.2 FreePolygon

Attributes

Type	Name
<code>gbeam_library/Vertex[]</code>	vertices_obstacle
<code>gbeam_library/Vertex[]</code>	vertices_reachable

3.1.3 FreePolygonStamped

Attributes

Type	Name
std_msgs/Header	header
<code>gbeam_library/FreePolygon</code>	polygon

3.1.4 GraphEdge

Attributes

Type	Name
int64	id
int64	v1
int64	v2
float32	length
geometry_msgs/Vector3	direction
bool	is_boundary
bool	is_walkable

Note: direction is defined from v1 to v2.

3.1.5 ReachabilityGraph

Attributes

Type	Name
<code>gbeam_library/Vertex[]</code>	<code>nodes</code>
<code>gbeam_library/GraphEdge[]</code>	<code>edges</code>
<code>gbeam_library/PolyArea[]</code>	<code>area</code>

3.1.6 PolyArea

Attributes

Type	Name
<code>geometry_msgs/Polygon[]</code>	<code>polygons</code>

3.2 Polytope functions

3.2.1 fn

Input

Name	Type	Description
<code>in1</code>	<code>float</code>	input argument

Output

Name	Type	Description
<code>out1</code>	<code>float</code>	output argument

Description

3.3 Graph functions

3.4 Exploration functions

Chapter 4

GBEAM Ground Package

This chapter describes the components of the “gbeam_ground” ROS package.

4.1 Scheme

Here an overview of the control scheme

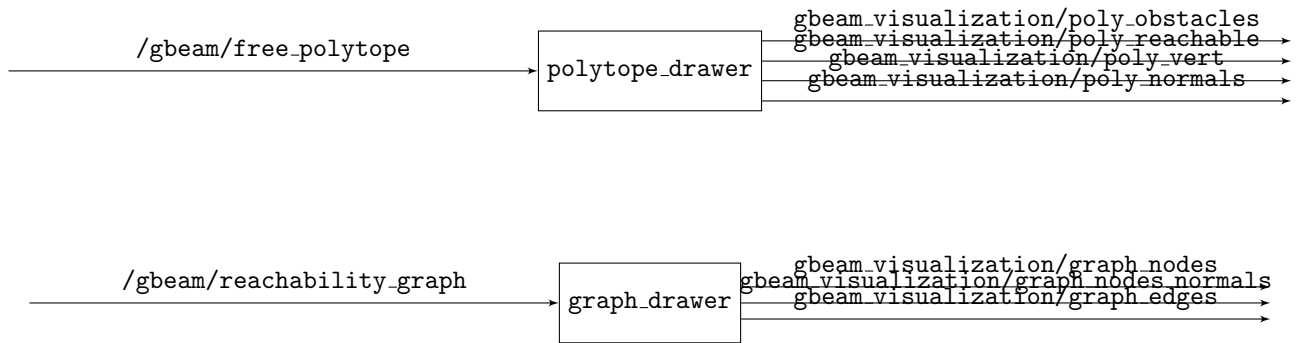


Figure 4.1: Diagram of gbeam ground package

4.2 Nodes

4.2.1 Polytope drawer node

Subscribed topics

Topic	Datatype
/gbeam/free_polytope	gbeam_library::FreePolygonStamped

Published topics

Topic	Datatype	Rate
/gbeam_visualization/poly_obstacles	geometry_msgs::PolygonStamped	As /gbeam/free_polytope
/gbeam_visualization/poly_reachable	geometry_msgs::PolygonStamped	As /gbeam/free_polytope
/gbeam_visualization/poly_vert	sensor_msgs::PointCloud	As /gbeam/free_polytope
/gbeam_visualization/poly_normals	visualization_msgs::Marker	As /gbeam/free_polytope

4.2.2 Graph drawer node

Subscribed topics

Topic	Datatype
/gbeam/reachability_graph	gbeam_library::ReachabilityGraph

Published topics

Topic	Datatype	Rate
/gbeam.visualization/graph_nodes	sensor_msgs::PointCloud	As /gbeam/reachability_graph
/gbeam.visualization/graph_nodes_normals	visualization_msgs::Marker	As /gbeam/reachability_graph
/gbeam.visualization/graph_edges	visualization_msgs::Marker	As /gbeam/reachability_graph

4.3 Launch files

4.3.1 Gbeam ground

Nodes

Package	Type	Name
gbeam_ground	graph_drawer	graph_drawer
gbeam_ground	polytope_drawer	polytope_drawer

Chapter 5

S1000 Interface Package

This chapter describes the components of the “s1000_interface” ROS package.

5.1 Scheme

Here an overview of the s1000 interface scheme

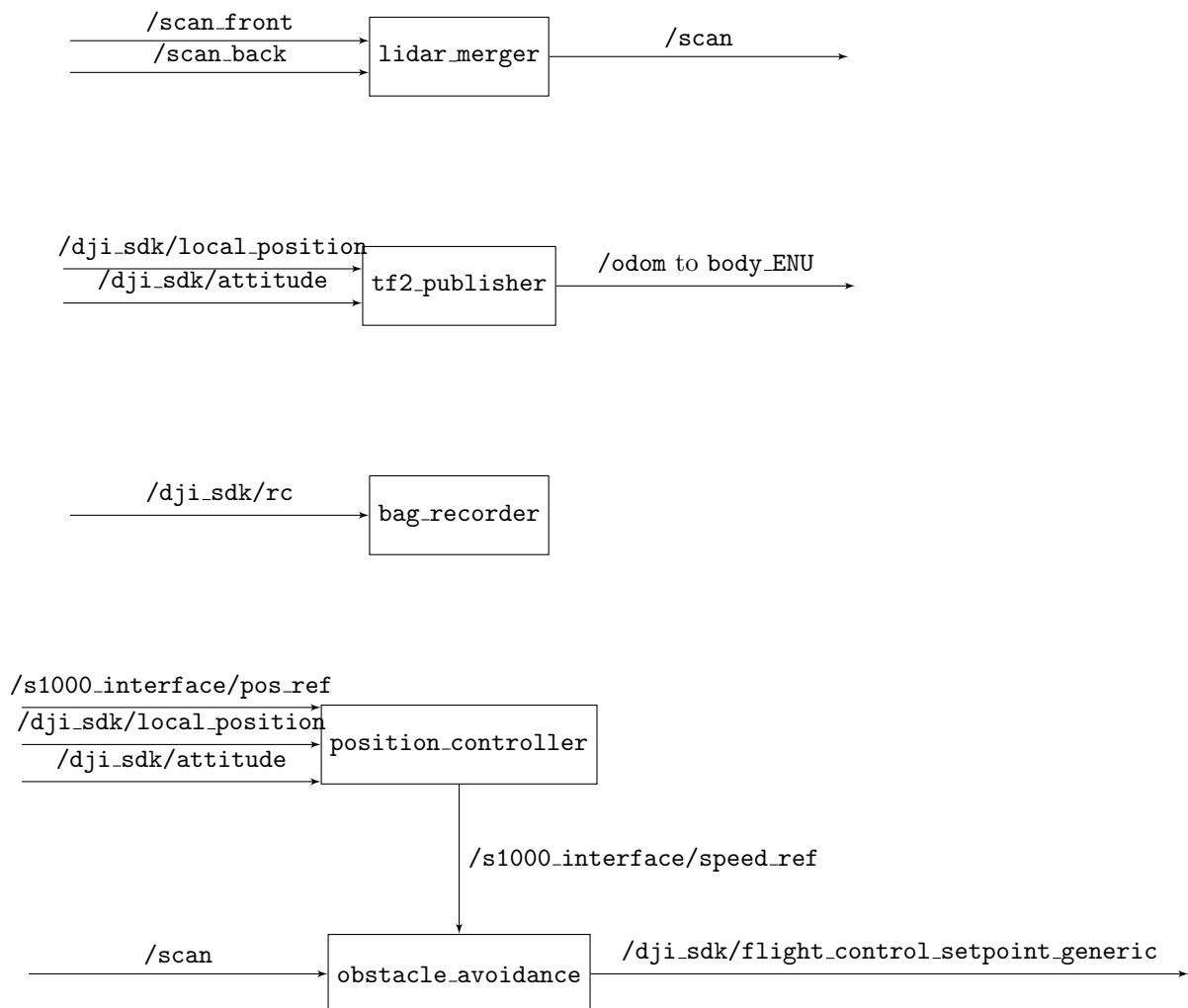


Figure 5.1: Diagram of s1000 interface package

5.2 Nodes

5.2.1 Lidar Merger

Subscribed topics

Topic	Datatype
/scan_front	sensor_msgs::LaserScan
/scan_back	sensor_msgs::LaserScan

Published topics

Topic	Datatype	Rate
/scan	sensor_msgs::LaserScan	Same as /scan_back

The output scan messages have reference frame: `base_scan`.

5.2.2 Tf2 Publisher

Subscribed topics

Topic	Datatype
/dji_sdk/local_position	geometry_msgs::PointStamped
/dji_sdk/attitude	geometry_msgs::QuaternionStamped

Published topics

This node publishes the transform from odom to body_FLU.

5.2.3 Bag Recorder

Subscribed topics

Topic	Datatype
/dji_sdk/rc	sensor_msgs::Joy

5.2.4 Position Controller

Subscribed topics

Topic	Datatype
/s1000_interface/pos_ref	geometry_msgs::PoseStamped
/dji_sdk/local_position	geometry_msgs::QuaternionStamped
/dji_sdk/attitude	geometry_msgs::PointStamped

Published topics

Topic	Datatype	Rate
/s1000_interface/speed_ref	sensor_msgs::Joy	10Hz

Parameters

Parameter	Datatype
/s1000_interface/position_controller/K_pos_xy	double
/s1000_interface/position_controller/K_pos_z	double
/s1000_interface/position_controller/K_yaw	double

5.2.5 Obstacle Avoidance

Subscribed topics

Topic	Datatype
/s1000_interface/speed_ref	sensor_msgs::Joy
/scan	sensor_msgs::LaserScan

Published topics

Topic	Datatype	Rate
/dji_sdk/flight_control_setpoint_generic	sensor_msgs::Joy	Same as /speed_ref

Parameters

Parameter	Datatype
/gbeam_controller/robot_param/safe_dist	double
/gbeam_controller/obstacle_avoidance/obstacle_spd_tol	double

5.3 Launch files

5.3.1 S1000 Interface

Parameters

Parameter	Datatype
/s1000_interface/position_controller/K_pos_xy	double
/s1000_interface/position_controller/K_pos_z	double
/s1000_interface/position_controller/K_yaw	double

Nodes

Package	Type	Name
dji_sdk	dji_sdk_node	dji_sdk
sick_tim	sick_tim551_2050001	sick_tim.back
sick_tim	sick_tim551_2050001	sick_tim.front
tf	static_transform_publisher	lidar_back_2_map_tf
tf	static_transform_publisher	lidar_front_2_map_tf
tf	static_transform_publisher	lidar_2_map_tf
s1000_interface	tf2_publisher	s1000_pos_publisher
s1000_interface	lidar_merger	front_back_lidar_merger
s1000_interface	bag_recorder	s1000_bag_recorder
s1000_interface	position_controller	position_controller
s1000_interface	auto_flight	auto_flight
s1000_interface	obstacle_avoidance	obstacle_avoidance

Bibliography