



# RESEARCH AUTONOMY KIT

All-in-one Autonomous Navigation

## ROS API DOCUMENTATION

VERSION 1.2

# CONTENTS

<b>1</b>	<b>Necessary Subscriptions</b>	<b>7</b>
1.1	tf . . . . .	7
1.2	platform_control_odom . . . . .	7
1.3	platform_odometry . . . . .	7
1.4	clock . . . . .	7
1.5	rear_scan . . . . .	7
1.6	front_scan . . . . .	8
<b>2</b>	<b>Subscribed Topics</b>	<b>9</b>
2.1	tf_static . . . . .	9
2.2	cancel_goal . . . . .	9
2.3	send_goal . . . . .	9
2.4	stereo_camera_0 . . . . .	9
<b>3</b>	<b>Published Topics</b>	<b>10</b>
3.1	platform_control_cmd_vel . . . . .	10
3.2	clock_echo . . . . .	10
3.3	map_data_features_markers_waypoint . . . . .	10
3.4	map_data_features_zones_exclusion . . . . .	10
3.5	control_selection_current_mode . . . . .	10
3.6	slam_map . . . . .	10
3.7	slam_current_mode . . . . .	11
3.8	slam_localization_status . . . . .	11
3.9	slam_estimated_pose . . . . .	11
3.10	path_2d . . . . .	11
3.11	path_planner_status . . . . .	11
3.12	path_planner_feedback . . . . .	11
3.13	path_planner_result . . . . .	11
3.14	path_planner_current_job . . . . .	11

3.15 path_planner_distance_to_goal . . . . .	12
3.16 robot_state_aggregator_state . . . . .	12
3.17 map_data_loaded_map_filename . . . . .	12
3.18 map_data_md5sum . . . . .	13
3.19 control_selection_current_mode . . . . .	13
3.20 control_selection_autonomy_disable_sources . . . . .	13
3.21 control_selection_autonomy_pause_sources . . . . .	13
3.22 control_selection_control_state . . . . .	13
3.23 control_selection_core_disable_sources . . . . .	13
3.24 control_selection_core_pause_sources . . . . .	13
3.25 ark_diagnostics . . . . .	14
<b>4 Services</b>	<b>15</b>
4.1 configure_ark . . . . .	15
4.2 start_autonomy . . . . .	15
4.3 stop_autonomy . . . . .	15
4.4 control_selection_set_mode . . . . .	15
4.5 slam_set_mode . . . . .	16
4.6 slam_set_initial_pose . . . . .	16
4.7 control_selection_autonomy_clear_all_disabled . . . . .	16
4.8 control_selection_autonomy_clear_all_paused . . . . .	16
4.9 control_selection_autonomy_disable . . . . .	16
4.10 control_selection_autonomy_enable . . . . .	16
4.11 control_selection_autonomy_pause . . . . .	16
4.12 control_selection_autonomy_resume . . . . .	17
4.13 control_selection_core_clear_all_disabled . . . . .	17
4.14 control_selection_core_clear_all_paused . . . . .	17
4.15 control_selection_core_disable . . . . .	17
4.16 control_selection_core_enable . . . . .	17
4.17 control_selection_core_pause . . . . .	17

4.18 control_selection_core_resume . . . . .	17
4.19 map_data_clear . . . . .	18
4.20 map_data_delete_map . . . . .	18
4.21 map_data_list_maps . . . . .	18
4.22 map_data_load_map_from_disk . . . . .	18
4.23 map_data_save_map_to_disk . . . . .	18
<b>5 Examples</b>	<b>19</b>
5.1 Commanding Positions - Python . . . . .	19
<b>6 Message Definitions</b>	<b>22</b>
6.1 ArkConfigSettingsCall . . . . .	22
6.2 ArkConfigSettingsResponse . . . . .	22
6.3 Attribute . . . . .	22
6.4 AutonomyMode . . . . .	22
6.5 Bool . . . . .	22
6.6 Clock . . . . .	23
6.7 CompressedMapData . . . . .	23
6.8 ControlSelectionState . . . . .	23
6.9 ControlState . . . . .	23
6.10 DiagnosticArray . . . . .	23
6.11 DiagnosticStatus . . . . .	24
6.12 DistanceToGoal . . . . .	24
6.13 duration . . . . .	24
6.14 Empty . . . . .	24
6.15 Float64 . . . . .	24
6.16 GeometryArray . . . . .	24
6.17 GeometryFeature . . . . .	25
6.18 GetCurrentMapResponse . . . . .	25
6.19 GoalID . . . . .	25
6.20 GoalStatus . . . . .	25

6.21 GoalStatusArray . . . . .	26
6.22 Header . . . . .	26
6.23 KeyValue . . . . .	26
6.24 LaserScan . . . . .	26
6.25 LocalizationStatus . . . . .	27
6.26 MapInfoResponse . . . . .	27
6.27 MapMetaData . . . . .	27
6.28 Mode . . . . .	27
6.29 OccupancyGrid . . . . .	27
6.30 Odometry . . . . .	28
6.31 Path . . . . .	29
6.32 PauseControlCall . . . . .	29
6.33 PauseControlResponse . . . . .	29
6.34 PauseStatus . . . . .	29
6.35 Point . . . . .	30
6.36 PointCloud2 . . . . .	30
6.37 PointField . . . . .	30
6.38 Pose . . . . .	30
6.39 Pose2D . . . . .	31
6.40 Pose2DArray . . . . .	31
6.41 PoseArray . . . . .	31
6.42 PoseStamped . . . . .	32
6.43 PoseWithCovariance . . . . .	32
6.44 PoseWithCovarianceStamped . . . . .	32
6.45 Quaternion . . . . .	33
6.46 Result . . . . .	33
6.47 RobotState . . . . .	33
6.48 SaveMapToDiskCall . . . . .	33
6.49 SendGoal . . . . .	33
6.50 SlamMode . . . . .	34

6.51 Status . . . . .	34
6.52 Step . . . . .	34
6.53 StepSequenceActionFeedback . . . . .	34
6.54 StepSequenceActionResult . . . . .	35
6.55 StepSequenceFeedback . . . . .	36
6.56 StepSequenceResult . . . . .	37
6.57 String . . . . .	37
6.58 StringArray . . . . .	38
6.59 tf2Message . . . . .	38
6.60 tfMessage . . . . .	38
6.61 time . . . . .	39
6.62 Transform . . . . .	39
6.63 TransformStamped . . . . .	39
6.64 Twist . . . . .	39
6.65 TwistWithCovariance . . . . .	40
6.66 Vector3 . . . . .	40
<b>7 Service Definitions</b>	<b>41</b>
7.1 ArkConfigSettings_service . . . . .	41
7.2 DeleteMap_service . . . . .	41
7.3 Empty_service . . . . .	41
7.4 ListMaps_service . . . . .	41
7.5 LoadMapFromDisk_service . . . . .	42
7.6 SaveMapToDisk_service . . . . .	42
7.7 SetAutonomyMode_service . . . . .	42
7.8 SetPose_service . . . . .	42
7.9 SetSlamMode_service . . . . .	43
7.10 String_service . . . . .	43

## NECESSARY SUBSCRIPTIONS

**Messages in this section are necessary for ARK to function properly. These must be provided by the robotic platform**

### tf

Message Type: tfMessage

Usage: The tf topic represents the relationship between coordinate frames in a tree structure buffered in time and must be maintained by the robotic platform. See <http://wiki.ros.org/tf> for more information

### platform\_control\_odom

Message Type: Odometry

Usage: One of two sources of odometry required from the robotic platform. This is the odometry provided by the platform control system, usually a velocity controller.

### platform\_odometry

Message Type: Odometry

Usage: One of two sources of odometry required from the robotic platform. This is the odometry provided by a separate source than the platform control odometry, such as an IMU, or fused odometry from multiple sources.

### clock

Message Type: Clock

Usage: The current time of the ROS system running on the platform. This must be of sufficient frequency, but excessive publishing frequency (greater than 100 Hz) may have an adverse effect on ARK performance.

### rear\_scan

Message Type: LaserScan

Usage: This is one of two necessary laser sources that must be provided to the ARK. The transform frame in the URDF that this laser is relative to must be called "rear\_laser".

## front\_scan

Message Type: LaserScan

Usage: This is one of two necessary laser sources that must be provided to the ARK. The transform frame in the URDF that this laser is relative to must be called "front\_laser".



## SUBSCRIBED TOPICS

### tf\_static

Message Type: tf2Message

Usage: This is similar to the necessary tf topic previously described, but it only needs to be published and latched once for transforms that are static and do not change. See <http://wiki.ros.org/tf> for more information

### cancel\_goal

Message Type: Empty

Usage: If the ARK is tracking a goal, the action can be canceled by publishing to this message.

### send\_goal

Message Type: SendGoal

Usage: ARK can be tasked with tracking a new goal using this message. Position and orientation are defined relative to coordinates in the map that the ARK is currently using. Position and Orientation tolerances can be defined.

### stereo\_camera\_0

Message Type: PointCloud2

Usage: This is an extra point cloud of the external environment. It can come from a single stereo/3D camera, or be fused from multiple sources. This cloud is used for obstacle avoidance only, and is not used in mapping

## PUBLISHED TOPICS

### platform\_control\_cmd\_vel

Message Type: Twist

Usage: This message is produced by the ARK to control the velocity of the platform. It is commonly relayed directly into the robot's velocity controller.

### clock\_echo

Message Type: Clock

Usage: This message is a status check to make sure the ARK is connected properly. As you are publishing a clock to the ARK, the ARK will send back its current time approximately every second to verify that it is still connected and how much delay there is in the networking between the robot and the ARK.

### map\_data\_features\_markers\_waypoint

Message Type: GeometryArray

Usage: The ARK will publish the markers saved in the currently loaded map.

### map\_data\_features\_zones\_exclusion

Message Type: GeometryArray

Usage: The ARK will publish the no-go/exclusion zones saved in the currently loaded map.

### control\_selection\_current\_mode

Message Type: AutonomyMode

Usage: The ARK will publish the current state of its Autonomy mode, which is whether the ARK will try to control the platform autonomously or not.

### slam\_map

Message Type: OccupancyGrid

Usage: The ARK will publish the currently loaded map. It can be saved (see [http://wiki.ros.org/map\\_server](http://wiki.ros.org/map_server)) or viewed in RVIZ.

## slam\_current\_mode

Message Type: SlamMode

Usage: Whether the ARK is currently updating its map, or just using it.

## slam\_localization\_status

Message Type: LocalizationStatus

Usage: The ark will publish if it is confident in its localization or if it is lost using this topic.

## slam\_estimated\_pose

Message Type: PoseWithCovarianceStamped

Usage: The ark will publish its estimated position within the map on this topic.

## path\_2d

Message Type: Pose2DArray

Usage: The ark will publish its current motion plan using this topic.

## path\_planner\_status

Message Type: GoalStatusArray

Usage: The ark will maintain a list of current objectives and their status using this topic.

## path\_planner\_feedback

Message Type: StepSequenceActionFeedback

## path\_planner\_result

Message Type: StepSequenceActionResult

## path\_planner\_current\_job

Message Type: String

Usage: The ID of the currently executing job

## path\_planner\_distance\_to\_goal

Message Type: DistanceToGoal

Usage: Direct distance between the current location and the intended goal

## robot\_state\_aggregator\_state

Message Type: RobotState

Usage: Used to indicate the current status of the ARK and individual components. Each bit of this binary number is used for a different status:

Status Bit	Information
E_STOP	Emergency stop - user intervention required.
SAFETY_STOP	Safety stop.
SAFE_TO_LOAD	Robot is safe to approach and work may be done on payload.
NARROW_HALLWAY	Robot is operating with reduced LIDAR fields and speed.
DOCKING	Not used by ARK
CHARGING	Not used by ARK
PARKED	Robot is parked and has no intent of moving.
ABOUT_TO_MOVE	Movement is indicated.
PAUSED	Robot is paused and cannot move. If in autonomous, this means robot autonomy or robot core is paused, if in manual, robot core is paused.
MANUAL	Robot is being controlled by teleoperation - cannot be driven autonomously.
AUTONOMY	Robot is autonomous.
PENDANT	Not used by ARK
APPLIANCE_ACTIVE	Not used by ARK
TRAINING	Robot is creating a new map. Or retraining an existing map.
LOCALIZING	Robot is attempting to localize.
LOCALIZED	Robot has localized.
LOST	Robot has lost localization and may need user intervention.
EXECUTING_JOB	Robot is actively executing a job.
PLANNING	Not used by ARK
TRACKING	Not used by ARK
BLOCKED	Robot is blocked from moving and can't replan.
NIMBUS_LOST	Not used by ARK

## map\_data\_loaded\_map\_filename

Message Type: String

Usage: The filename of the currently loaded map.

## map\_data\_md5sum

Message Type: String

Usage: The MD5 checksum of the currently loaded map.

## control\_selection\_current\_mode

Message Type: AutonomyMode

Usage: The current control mode of the ARK.

## control\_selection\_autonomy\_disable\_sources

Message Type: StringArray

Usage: The list of sources currently disabling the Autonomy of the ARK.

## control\_selection\_autonomy\_pause\_sources

Message Type: StringArray

Usage: The list of sources currently pausing the Autonomy of the ARK.

## control\_selection\_control\_state

Message Type: ControlSelectionState

Usage: The ARK will publish the current state of its control mode, which is whether the ARK will try to control the platform or not.

## control\_selection\_core\_disable\_sources

Message Type: StringArray

Usage: The list of sources currently disabling the control of the ARK.

## control\_selection\_core\_pause\_sources

Message Type: StringArray

Usage: The list of sources currently pausing the control of the ARK.

## ark\_diagnostics

Message Type: DiagnosticArray

Usage: Diagnostics from the ARK providing more information relating to issues and the status of the ARK and its systems.

## SERVICES

All ARK services have value of "ark\_service\_timeout" that is used to ensure the ARK successfully received the service call. If this is set to False, it means there was no timeout and the call was received successfully. If this is set to true however, it means that the ARK either didn't receive the call, or didn't respond to the call. If this is the case, check your User Manual to verify that all setup steps have been followed correctly and verify any troubleshooting steps.

### configure\_ark

Type: ArkConfigSettings\_service

Usage: This service is used to configure general settings that the ARK will use when planning, navigating, and using the messages it receives from the platform. Example configurations are provided at <https://github.com/autonomyresearchkit>. It is **strongly** recommended not to set the vehicle width and length parameters to smaller than they are set in the example configurations or the ARK will not know how large your platform is and may collide with objects or people. This is very dangerous!

### start\_autonomy

Type: Empty\_service

Usage: Calling this service will start the mapping, planning, and localization features of the ARK using the current configuration. If the ARK does not start, verify that all necessary topics are being published, and a proper configuration is set in the ARK. The ARK's diagnostics topic will also explain start up issues.

### stop\_autonomy

Type: Empty\_service

Usage: Calling this service will stop the autonomy and completely restart the ARK.

### control\_selection\_set\_mode

Type: SetAutonomyMode\_service

Usage: This service will set the control mode

Mode Number	Control Type	Information
0	Neutral	ARK will not command movement
1	Manual	ARK may command movement using the joystick in the web interface
2	Autonomous	ARK may command autonomous movement

## slam\_set\_mode

Type: SetSlamMode\_service

Usage: Enable or disable updating the map using SLAM

## slam\_set\_initial\_pose

Type: SetPose\_service

Usage: If the robot has just started up or has become "lost" because of extreme map changes or incorrect odometry, the current position of robot can be "hinted" using this service

## control\_selection\_autonomy\_clear\_all\_disabled

Type: Empty\_service

Usage: Clears all sources that are disabling autonomy

## control\_selection\_autonomy\_clear\_all\_paused

Type: Empty\_service

Usage: Clears all sources that are pausing autonomy

## control\_selection\_autonomy\_disable

Type: String\_service

Usage: Sets a new source to pause autonomy. A name must be provided for this source so it can be selectively be enabled later

## control\_selection\_autonomy\_enable

Type: String\_service

Usage: Removes a source that is disabling autonomy by name.

## control\_selection\_autonomy\_pause

Type: String\_service

Usage: Sets a new source to pause autonomy. A name must be provided for this source so it can be selectively be unpaused later



## control\_selection\_autonomy\_resume

Type: String\_service

Usage: Removes a source that is pausing autonomy by name.

## control\_selection\_core\_clear\_all\_disabled

Type: Empty\_service

Usage: Clears all sources that are disabling core functionality

## control\_selection\_core\_clear\_all\_paused

Type: Empty\_service

Usage: Clears all sources that are pausing core functionality

## control\_selection\_core\_disable

Type: String\_service

Usage: Sets a new source to disable core functionality. A name must be provided for this source so it can be selectively be enabled later

## control\_selection\_core\_enable

Type: String\_service

Usage: Removes a source that is disabling autonomy by name.

## control\_selection\_core\_pause

Type: String\_service

Usage: Sets a new source to pause core functionality. A name must be provided for this source so it can be selectively be unpaused later

## control\_selection\_core\_resume

Type: String\_service

Usage: Removes a source that is pausing autonomy by name.

## map\_data\_clear

Type: Empty\_service

Usage: Completely clears the current map that the ARK is using and prepares the ARK to begin creating a new map.

## map\_data\_delete\_map

Type: DeleteMap\_service

Usage: Deletes a map file that is currently saved on the ARK

## map\_data\_list\_maps

Type: ListMaps\_service

Usage: Lists all map files that are currently saved on the ARK

## map\_data\_load\_map\_from\_disk

Type: LoadMapFromDisk\_service

Usage: Loads a map file by name into the ARK's memory to be used for navigation. The location should be '/var/tmp/ros/<filename>'.

## map\_data\_save\_map\_to\_disk

Type: SaveMapToDisk\_service

Usage: Saves the current map to the ARK. The map\_topic must always be '/slam/map' and the location should be '/var/tmp/ros/<filename>'.

## EXAMPLES

### Commanding Positions - Python

```
#!/usr/bin/env python
from ark_bridge.msg import SendGoal
from ark_bridge.msg import GoalStatusArray
import actionlib_msgs
import time

import rospy

tracking_goal = True

#####
# This is just simple storage for the individual points by name #
#####
class locations:
    def __init__(self):
        self.location_list = {}

        self.location_list["ORIGIN"] = pose_data("ORIGIN", 0.0, 0.0, 0.0, 0.0, 0.0, 1.0)
        self.location_list["WORKBENCH"] = pose_data("WORKBENCH", 1.42520270935, -2.07659318067, 0.0, 0.0,
            0.0, 1.0)
        self.location_list["LOCATION 3"] = pose_data("LOCATION 3", 0.387670869059, -3.0498080698, 0.0, 0.0,
            0.0, 1.0)
        self.location_list["LOCATION 4"] = pose_data("LOCATION4", 3.13896623959, -1.65007802521, 0.0, 0.0,
            0.0, -0.755254064534, 0.655432145996)

#####
# This is a simple class for making the end code cleaner #
#####
class pose_data:
    def __init__(self, name, x, y, z, q1, q2, q3, q4):
        self.name = name
        self.x = x
        self.y = y
        self.z = z
        self.q1 = q1
        self.q2 = q2
        self.q3 = q3
        self.q4 = q4

    # Returns a ROS message to send to the ark #
    def arkPose(self):
        msg = SendGoal()
        msg.pose.position.x = self.x
        msg.pose.position.y = self.y
        msg.pose.position.z = self.z
        msg.pose.orientation.x = self.q1
```

```

    msg.pose.orientation.y = self.q2
    msg.pose.orientation.z = self.q3
    msg.pose.orientation.w = self.q4

    msg.position_tolerance = 0.1
    msg.orientation_tolerance = 0.1

    return msg

#####
# This keeps track of the ARK's status and if it is tracking a goal or awaiting a new goal #
#####
def statusCallback(msg):
    global tracking_goal
    running_goal = False
    for s in msg.status_list:
        if s.status == 0 or s.status == 1:
            running_goal = True

    tracking_goal = running_goal

if __name__ == '__main__':
    try:
        # Start a ROS node called "PositionCommander" #
        rospy.init_node("PositionCommander", anonymous=True)

        # Get our list of locations #
        positionList = locations().location_list

        # Pull out the locations we want to use and put them in a list so we can run them in a specific order
        #
        positions_to_follow = []
        positions_to_follow.append(positionList["ORIGIN"])
        positions_to_follow.append(positionList["WORKBENCH"])
        positions_to_follow.append(positionList["LOCATION 4"])

        # Get a handle to the topic the ARK uses to accept goal #
        pub = rospy.Publisher('/ark_bridge/send_goal', SendGoal, queue_size=1)
        # Get a handle to the topic the ARK uses to return its status #
        rospy.Subscriber("/ark_bridge/path_planner_status", GoalStatusArray, statusCallback)

        # Loop through the positions in our list one at a time #
        for p in positions_to_follow:
            rospy.loginfo("Moving to " + p.name)
            # Get a message for the ark and send it to the new goal topic #
            pub.publish(p.arkPose())
            # Wait for a second to make sure the ARK has time to accept the goal #
            time.sleep(1)

            # Loop until the ARK says it is done running a goal #
            while(tracking_goal) and not rospy.is_shutdown():
                time.sleep(1)

```

```
rospy.loginfo("Done")
```

## MESSAGE DEFINITIONS

### ArkConfigSettingsCall

```
float64 max_fwd_linear_speed
float64 max_rev_linear_speed
float64 min_linear_speed
float64 max_linear_acceleration
float64 max_linear_deceleration
float64 max_ang_speed
float64 max_ang_accel
float64 vehicle_length
float64 vehicle_width
float64 stopping_distance_1M
float64 lidar_spacing
float64 laser_fov
float64 camera_3d_memory
int8 drive_direction
```

### ArkConfigSettingsResponse

```
bool result
string information
```

### Attribute

```
string name
int64 offset
int16 datatype
int64 length
```

### AutonomyMode

```
int8 mode
```

### Bool

```
bool data
```

## Clock

```
time clock
```

## CompressedMapData

```
int32 ndata  
int8[] data
```

## ControlSelectionState

```
ark_bridge/ControlState core  
  bool enabled  
  bool paused  
ark_bridge/ControlState autonomy  
  bool enabled  
  bool paused  
ark_bridge/AutonomyMode mode  
  int8 mode
```

## ControlState

```
bool enabled  
bool paused
```

## DiagnosticArray

```
std_msgs/Header header  
  uint32 seq  
  time stamp  
  string frame_id  
int32 nstatus  
ark_bridge/DiagnosticStatus[] status  
  byte level  
  string name  
  string message  
  string hardware_id  
  int32 nvalues  
  ark_bridge/KeyValue[] values  
    string key  
    string value
```

## DiagnosticStatus

```
byte level
string name
string message
string hardware_id
int32 nvalues
ark_bridge/KeyValue[] values
  string key
  string value
```

## DistanceToGoal

```
string job_id
float32 distance_to_goal
```

## duration

```
int32 secs
int32 nsecs
```

## Empty

## Float64

```
float64 data
```

## GeometryArray

```
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
int32 nfeatures
ark_bridge/GeometryFeature[] features
  string id
  int16 type
  int32 npoints
  int32 nattributes
```



```
int32 ndata
ark_bridge/Point[] points
  float64 x
  float64 y
  float64 z
ark_bridge/Attribute[] attributes
  string name
  int64 offset
  int16 datatype
  int64 length
  int16[] data
```

## GeometryFeature

```
string id
int16 type
int32 npoints
int32 nattributes
int32 ndata
ark_bridge/Point[] points
  float64 x
  float64 y
  float64 z
ark_bridge/Attribute[] attributes
  string name
  int64 offset
  int16 datatype
  int64 length
  int16[] data
```

## GetCurrentMapResponse

```
ark_bridge/CompressedMapData map
  int32 ndata
  int8[] data
```

## GoalID

```
time stamp
string id
```

## GoalStatus

```
ark_bridge/GoalID goal_id
  time stamp
  string id
int16 status
string text
```

## GoalStatusArray

```
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
int32 nstatus_list
ark_bridge/GoalStatus[] status_list
  ark_bridge/GoalID goal_id
    time stamp
    string id
  int16 status
  string text
```

## Header

```
int64 seq
time stamp
string frame_id
```

## KeyValue

```
string key
string value
```

## LaserScan

```
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
float32 angle_min
float32 angle_max
float32 angle_increment
float32 time_increment
float32 scan_time
```

```
float32 range_min  
float32 range_max  
int32 nranges  
int32 nintensities  
float32[] ranges  
float32[] intensities
```

## LocalizationStatus

```
int16 status
```

## MapInfoResponse

```
string name  
string md5sum  
int32 size
```

## MapMetaData

```
time map_load_time  
float32 resolution  
int64 width  
int64 height  
ark_bridge/Pose origin  
  ark_bridge/Point position  
    float64 x  
    float64 y  
    float64 z  
ark_bridge/Quaternion orientation  
  float64 x  
  float64 y  
  float64 z  
  float64 w
```

## Mode

```
int8 mode
```

## OccupancyGrid

```
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
ark_bridge/MapMetaData info
  time map_load_time
  float32 resolution
  int64 width
  int64 height
ark_bridge/Pose origin
  ark_bridge/Point position
    float64 x
    float64 y
    float64 z
  ark_bridge/Quaternion orientation
    float64 x
    float64 y
    float64 z
    float64 w
int32 ndata
int8[] data
```

## Odometry

```
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
string child_frame_id
ark_bridge/PoseWithCovariance pose
  ark_bridge/Pose pose
    ark_bridge/Point position
      float64 x
      float64 y
      float64 z
    ark_bridge/Quaternion orientation
      float64 x
      float64 y
      float64 z
      float64 w
  float64[36] covariance
ark_bridge/TwistWithCovariance twist
  ark_bridge/Twist twist
    ark_bridge/Vector3 linear
      float64 x
      float64 y
      float64 z
    ark_bridge/Vector3 angular
      float64 x
```

```
float64 y
float64 z
float64[36] covariance
```

## Path

```
std_msgs/Header header
uint32 seq
time stamp
string frame_id
int32 nposes
ark_bridge/PoseStamped[] poses
std_msgs/Header header
uint32 seq
time stamp
string frame_id
ark_bridge/Pose pose
ark_bridge/Point position
float64 x
float64 y
float64 z
ark_bridge/Quaternion orientation
float64 x
float64 y
float64 z
float64 w
```

## PauseControlCall

```
string job_id
string source
```

## PauseControlResponse

```
ark_bridge/PauseStatus status
int16 status
string job_id
int32 nsources
string[] sources
int16 result
string result_text
```

## PauseStatus

```
int16 status
string job_id
int32 nsources
string[] sources
```

## Point

```
float64 x
float64 y
float64 z
```

## PointCloud2

```
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
int64 height
int64 width
int32 nfields
ark_bridge/PointField[] fields
  string name
  int64 offset
  int16 datatype
  int64 count
bool is_bigendian
int64 point_step
int64 row_step
int32 ndata
int16[] data
bool is_dense
```

## PointField

```
string name
int64 offset
int16 datatype
int64 count
```

## Pose

```
ark_bridge/Point position
```

```
float64 x
float64 y
float64 z
ark_bridge/Quaternion orientation
float64 x
float64 y
float64 z
float64 w
```

## Pose2D

```
float64 x
float64 y
float64 theta
```

## Pose2DArray

```
int32 nposes
std_msgs/Header header
uint32 seq
time stamp
string frame_id
ark_bridge/Pose2D[] poses
float64 x
float64 y
float64 theta
```

## PoseArray

```
std_msgs/Header header
uint32 seq
time stamp
string frame_id
int32 nposes
ark_bridge/Pose[] poses
ark_bridge/Point position
float64 x
float64 y
float64 z
ark_bridge/Quaternion orientation
float64 x
float64 y
float64 z
float64 w
```

## PoseStamped

```
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
ark_bridge/Pose pose
  ark_bridge/Point position
    float64 x
    float64 y
    float64 z
  ark_bridge/Quaternion orientation
    float64 x
    float64 y
    float64 z
    float64 w
```

## PoseWithCovariance

```
ark_bridge/Pose pose
  ark_bridge/Point position
    float64 x
    float64 y
    float64 z
  ark_bridge/Quaternion orientation
    float64 x
    float64 y
    float64 z
    float64 w
float64[36] covariance
```

## PoseWithCovarianceStamped

```
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
ark_bridge/PoseWithCovariance pose
  ark_bridge/Pose pose
    ark_bridge/Point position
      float64 x
      float64 y
      float64 z
    ark_bridge/Quaternion orientation
      float64 x
      float64 y
      float64 z
```



```
float64 w
float64[36] covariance
```

## Quaternion

```
float64 x
float64 y
float64 z
float64 w
```

## Result

```
int16 result
string text
```

## RobotState

```
int64 state
```

## SaveMapToDiskCall

```
string map_topic
duration timeout
string filename
```

## SendGoal

```
string id
ark_bridge/Pose pose
  ark_bridge/Point position
    float64 x
    float64 y
    float64 z
  ark_bridge/Quaternion orientation
    float64 x
    float64 y
    float64 z
    float64 w
float32 position_tolerance
float32 orientation_tolerance
```

## SlamMode

```
int64 mode
```

## Status

```
int64 status
```

## Step

```
string id
string type
bool interruptible
int32 nattributes
int32 ndata
ark_bridge/Attribute[] attributes
  string name
  int64 offset
  int16 datatype
  int64 length
int16[] data
```

## StepSequenceActionFeedback

```
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
ark_bridge/GoalStatus status
  ark_bridge/GoalID goal_id
  time stamp
  string id
  int16 status
  string text
ark_bridge/StepSequenceFeedback feedback
  int32 nfinished_steps
  int32 nremaining_steps
  ark_bridge/Step[] finished_steps
    string id
    string type
    bool interruptible
    int32 nattributes
    int32 ndata
    ark_bridge/Attribute[] attributes
      string name
```

```

    int64 offset
    int16 datatype
    int64 length
    int16[] data
    ark_bridge/Step current_step
    string id
    string type
    bool interruptible
    int32 nattributes
    int32 ndata
    ark_bridge/Attribute[] attributes
    string name
    int64 offset
    int16 datatype
    int64 length
    int16[] data
    ark_bridge/Step[] remaining_steps
    string id
    string type
    bool interruptible
    int32 nattributes
    int32 ndata
    ark_bridge/Attribute[] attributes
    string name
    int64 offset
    int16 datatype
    int64 length
    int16[] data
    float32 current_step_progress
    float32 total_progress

```

## StepSequenceActionResult

```

std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
ark_bridge/GoalStatus status
  ark_bridge/GoalID goal_id
  time stamp
  string id
  int16 status
  string text
ark_bridge/StepSequenceResult result
  int32 nfinished_steps
  int32 nunfinished_steps
  ark_bridge/Step[] finished_steps
  string id
  string type
  bool interruptible

```

```

    int32 nattributes
    int32 ndata
    ark_bridge/Attribute[] attributes
        string name
        int64 offset
        int16 datatype
        int64 length
    int16[] data
    ark_bridge/Step[] unfinished_steps
        string id
        string type
        bool interruptible
        int32 nattributes
        int32 ndata
        ark_bridge/Attribute[] attributes
            string name
            int64 offset
            int16 datatype
            int64 length
        int16[] data

```

## StepSequenceFeedback

```

    int32 nfinished_steps
    int32 nremaining_steps
    ark_bridge/Step[] finished_steps
        string id
        string type
        bool interruptible
        int32 nattributes
        int32 ndata
        ark_bridge/Attribute[] attributes
            string name
            int64 offset
            int16 datatype
            int64 length
        int16[] data
    ark_bridge/Step current_step
        string id
        string type
        bool interruptible
        int32 nattributes
        int32 ndata
        ark_bridge/Attribute[] attributes
            string name
            int64 offset
            int16 datatype
            int64 length
        int16[] data
    ark_bridge/Step[] remaining_steps

```

```

string id
string type
bool interruptible
int32 nattributes
int32 ndata
ark_bridge/Attribute[] attributes
  string name
  int64 offset
  int16 datatype
  int64 length
int16[] data
float32 current_step_progress
float32 total_progress

```

## StepSequenceResult

```

int32 nfinished_steps
int32 nunfinished_steps
ark_bridge/Step[] finished_steps
  string id
  string type
  bool interruptible
  int32 nattributes
  int32 ndata
  ark_bridge/Attribute[] attributes
    string name
    int64 offset
    int16 datatype
    int64 length
  int16[] data
ark_bridge/Step[] unfinished_steps
  string id
  string type
  bool interruptible
  int32 nattributes
  int32 ndata
  ark_bridge/Attribute[] attributes
    string name
    int64 offset
    int16 datatype
    int64 length
  int16[] data

```

## String

```

string data

```

## StringArray

```
int32 nstrings
string[] data
```

## tf2Message

```
int32 ntransforms
ark_bridge/TransformStamped[] transforms
  string child_frame_id
  std_msgs/Header header
    uint32 seq
    time stamp
    string frame_id
  ark_bridge/Transform transform
    ark_bridge/Vector3 translation
      float64 x
      float64 y
      float64 z
    ark_bridge/Quaternion rotation
      float64 x
      float64 y
      float64 z
      float64 w
```

## tfMessage

```
int32 ntransforms
ark_bridge/TransformStamped[] transforms
  string child_frame_id
  std_msgs/Header header
    uint32 seq
    time stamp
    string frame_id
  ark_bridge/Transform transform
    ark_bridge/Vector3 translation
      float64 x
      float64 y
      float64 z
    ark_bridge/Quaternion rotation
      float64 x
      float64 y
      float64 z
      float64 w
```

## time

```
int32 secs
int32 nsecs
```

## Transform

```
ark_bridge/Vector3 translation
  float64 x
  float64 y
  float64 z
ark_bridge/Quaternion rotation
  float64 x
  float64 y
  float64 z
  float64 w
```

## TransformStamped

```
string child_frame_id
std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
ark_bridge/Transform transform
  ark_bridge/Vector3 translation
    float64 x
    float64 y
    float64 z
  ark_bridge/Quaternion rotation
    float64 x
    float64 y
    float64 z
    float64 w
```

## Twist

```
ark_bridge/Vector3 linear
  float64 x
  float64 y
  float64 z
ark_bridge/Vector3 angular
  float64 x
  float64 y
  float64 z
```

## TwistWithCovariance

```
ark_bridge/Twist twist
  ark_bridge/Vector3 linear
    float64 x
    float64 y
    float64 z
  ark_bridge/Vector3 angular
    float64 x
    float64 y
    float64 z
  float64[36] covariance
```

## Vector3

```
float64 x
float64 y
float64 z
```



## SERVICE DEFINITIONS

### ArkConfigSettings\_service

```
ark_bridge/ArkConfigSettingsCall req_data
  float64 max_fwd_linear_speed
  float64 max_rev_linear_speed
  float64 min_linear_speed
  float64 max_linear_acceleration
  float64 max_linear_deceleration
  float64 max_ang_speed
  float64 max_ang_accel
  float64 vehicle_length
  float64 vehicle_width
  float64 stopping_distance_1M
  float64 lidar_spacing
  float64 laser_fov
  float64 camera_3d_memory
  int8 drive_direction
---
ark_bridge/ArkConfigSettingsResponse res_data
  bool result
  string information
bool ark_service_timeout
```

### DeleteMap\_service

```
ark_bridge/String req_data
  string data
---
ark_bridge/Empty res_data
bool ark_service_timeout
```

### Empty\_service

```
ark_bridge/Empty req_data
---
ark_bridge/Empty res_data
bool ark_service_timeout
```

### ListMaps\_service

```
ark_bridge/Empty req_data
---
```

```
ark_bridge/StringArray res_data
  int32 nstrings
  string[] data
bool ark_service_timeout
```

## LoadMapFromDisk\_service

```
ark_bridge/String req_data
  string data
---
ark_bridge/Result res_data
  int16 result
  string text
bool ark_service_timeout
```

## SaveMapToDisk\_service

```
ark_bridge/SaveMapToDiskCall req_data
  string map_topic
  duration timeout
  string filename
---
ark_bridge/Empty res_data
bool ark_service_timeout
```

## SetAutonomyMode\_service

```
ark_bridge/AutonomyMode req_data
  int8 mode
---
ark_bridge/Empty res_data
bool ark_service_timeout
```

## SetPose\_service

```
ark_bridge/PoseWithCovarianceStamped req_data
  std_msgs/Header header
  uint32 seq
  time stamp
  string frame_id
ark_bridge/PoseWithCovariance pose
  ark_bridge/Pose pose
  ark_bridge/Point position
```

```
float64 x
float64 y
float64 z
ark_bridge/Quaternion orientation
float64 x
float64 y
float64 z
float64 w
float64[36] covariance
---
ark_bridge/Bool res_data
bool data
bool ark_service_timeout
```

## SetSlamMode\_service

```
ark_bridge/SlamMode req_data
int64 mode
---
ark_bridge/Status res_data
int64 status
bool ark_service_timeout
```

## String\_service

```
ark_bridge/String req_data
string data
---
ark_bridge/String res_data
string data
bool ark_service_timeout
```