



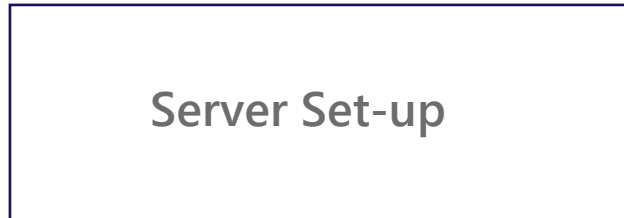
SDK USER GUIDE



LIPSAMR VSLAM SDK

VSLAM SDK Structure

Step-1



step-1

Open a terminal, and enter:

```
ros2 launch amr_api 4cam_AMRapp_startup.launch.py
```

→ [open camera](#)

step-2

Open a new terminal, and enter:

```
ros2 launch amr_api amr_api_component.launch.py
```

→ [open server](#)

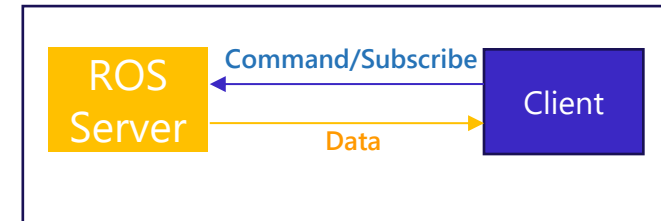
Step-3

Open a new terminal, and enter:

```
rviz2 -d /opt/ros/lips_amr_ws/slam.rviz
```

→ [open Rviz](#)

Step-2 ROS Service Client



Execute the function of the LIPSAMR VSLAM SDK via commands/ subscribe.

Access the LIPSAMR VSLAM SDK through the ROS service client template.

→ [Writing a simple service and client \(C++\) — ROS 2 Documentation: Humble documentation](#)

→ [Topic Subscribe Tutorial \(C++\) — ROS 2 Humble](#)





Config Setting



Config Setting_Camera & Obstacle Height (1/2)

Before executing the SDK, please adjust the the config setting first.

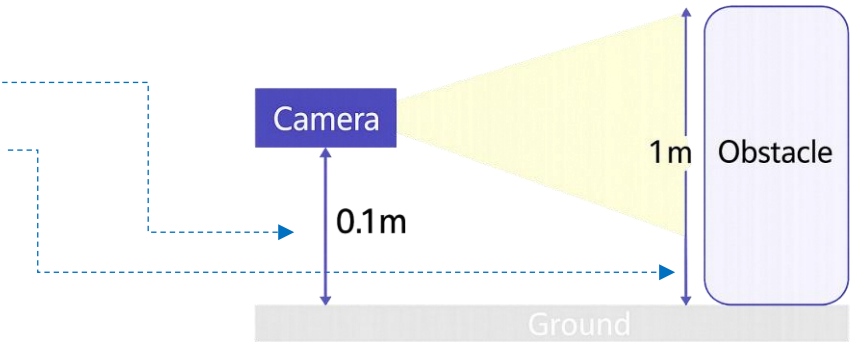
Go to the /opt/ros/lips_amr_ws/config.ini

Find the CameraHeight & ObstacleHeight in the list, and adjust them based on the following rules.

```
config.ini
/opt/ros/lips_amr_ws

14 CameraHeight = 0.1
15 ObstacleHeight = 1.0
```

Function	Description	Parameter Name	Remark
Explore	Set the camera Height	CameraHeight	Example CameraHeight: 0.1m ObstacleHeight: 1m
	Set the obstacle height	ObstacleHeight	





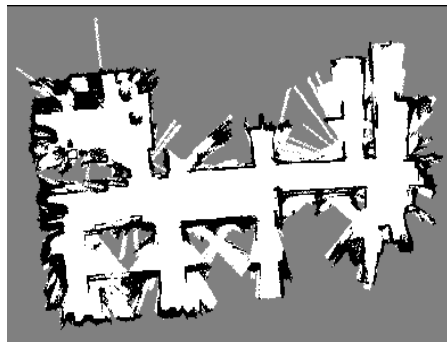
Config Setting_RangeMax (2/2)

Before executing the SDK, please adjust the the config setting first.
Go to the /opt/ros/lips_amr_ws/config.ini
Find the RangeMax in the list, and adjust it based on the following rules.

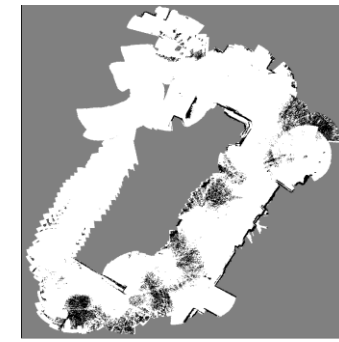
```
config.ini
/opt/ros/lips_amr_ws
64 Grid\RangeMax = 2.5
53 Grid\MapFrameProjection = true
54 Grid\MaxGroundAngle = 45
```

Function	Description	Parameter Name	Remark
Explore	Camera Scan Range	RangeMax	Narrow Space: Please adjust the “RangeMax” to 2.5 Wider Space: Please adjust the “RangeMax” to 4 *unit is meter.

Narrow Space(Such as: Office)



Wider Space(Such as: parking lot)





ROS Service Client Guide





LIPSAMR VSLAM SDK – Exploration

ROS Service Client Guide

Function	Description	Service Name	Service Type	Parameter Type
Explore	Start Mapping	/amr_api/map_service /start_explore	amr_api/srv/LoadMap	Parameter: path Type: String Description: db save path Example: '/home/orin/AMRapi_vslam/map/102225.db'
	Stop Mapping	/amr_api/map_service /stop_explore	amr_api/srv/Empty	
Load Map	Load Map	/amr_api/map_service /load_map	amr_api/srv/LoadMap	Parameter: path Type: String Description: db file path Example: '/home/orin/AMRapi_vslam/map/102225.db'
Recover VO (Visual Odometry)	Recover and align the VO to other VO.	/ <camera_ns> /recove r/reset_to_ekf	std_srvs/srv/Empty	





Service Client Quick Start - Python & C++

Python & C++ Examples

Function	Description	Service Name	Service Type
Load Map	Load database and switch to localization	/amr_api/map_service/load_map	amr_api/srv/LoadMap

Python Example

```
import rclpy
from amr_api.srv import LoadMap

rclpy.init()
node = rclpy.create_node("load_map_client")
client = node.create_client(LoadMap,
    "/amr_api/map_service/load_map")
client.wait_for_service(timeout_sec=10.0)

req = LoadMap.Request()
req.path = "/home/orin/AMRapi_vslam/map/102225.db"

future = client.call_async(req)
rclpy.spin_until_future_complete(node, future)
res = future.result()
node.get_logger().info(
    f"success={res.success}, message={res.message}")
rclpy.shutdown()
```

C++ Example

```
#include <chrono>
#include <memory>
#include <rclcpp/rclcpp.hpp>
#include <amr_api/srv/load_map.hpp>
using namespace std::chrono_literals;

int main(int argc, char ** argv) {
    rclcpp::init(argc, argv);
    auto node = rclcpp::Node::make_shared("load_map_client");
    auto client = node->create_client<amr_api::srv::LoadMap>(
        "/amr_api/map_service/load_map");
    client->wait_for_service(10s);

    auto req = std::make_shared<amr_api::srv::LoadMap::Request>();
    req->path = "/home/orin/AMRapi_vslam/map/102225.db";

    auto future = client->async_send_request(req);
    rclcpp::spin_until_future_complete(node, future);
    auto res = future.get();
    RCLCPP_INFO(node->get_logger(), "success=%d, message=%s",
        res->success, res->message.c_str());
    rclcpp::shutdown();
}
```



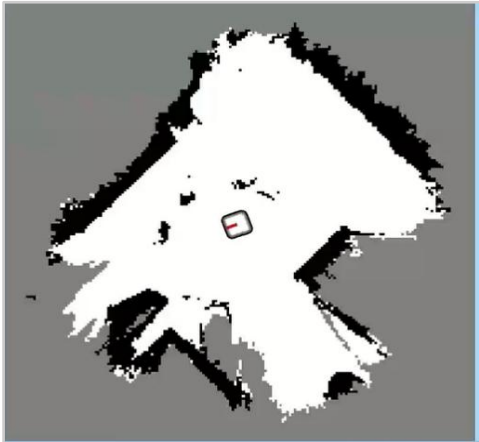


LIPSAMR VSLAM SDK – Exploration

Example Command Guide

Function	Description	Command	Remark
Explore	Start Mapping	ros2 service call /amr_api/map_service/start_explore amr_api/srv/LoadMap "{path: '/home/orin/AMRapi_vslam/map/102225.db'}"	Saving path is an example. User can specify it.
	Stop Mapping	ros2 service call /amr_api/map_service/stop_explore amr_api/srv/Empty	Once the user stops mapping, the map will be saved automatically.
Load Map	Load Map	ros2 service call /amr_api/map_service/load_map amr_api/srv/LoadMap "{path: '/home/orin/AMRapi_vslam/map/102225.db'}"	File path is an example. User can specify it.

▼ Map Output (Explore / Load Map)





LIPSAMR VSLAM SDK – Exploration

Example Command Guide

Function	Description	Command	Remark
Recover VO (Visual Odometry)	Recover and align camera's VO to other VO.	ros2 service call /camera0/recover/reset_to_ekf std_srvs/srv/Empty	We use camera0 for example here. The user can trigger this function to recover the camera's VO once the camera is lost.

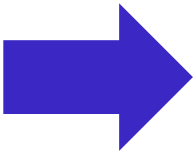


We use camera0 as an example.
Please refer to the camera name as follows:
camera0
camera1
camera2
camera3

EX:
The following UI is from LIPSAMR APP.
We use camera0 as an example.



Red color means the camera is lost.



If we trigger this function, the camera0 will be recovered.





Get VSLAM Data



Mapping - Subscribe & Parse the Map

ROS2 Library API

Function	Description	Topic Name	Message Type
Map	Subscribe & parse map data	/rtabmap/map	nav_msgs/msg/OccupancyGrid

Python Example

```
import rclpy
from rclpy.node import Node
from nav_msgs.msg import OccupancyGrid

class MapSubscriber(Node):
    def __init__(self):
        super().__init__("map_subscriber")
        self.create_subscription(
            OccupancyGrid, "/rtabmap/map", self.on_map, 10)

    def on_map(self, msg):
        w, h = msg.info.width, msg.info.height
        row, col = h // 2, w // 2
        cell = msg.data[row * w + col]
        self.get_logger().info(f"Map {w}x{h}, center={cell}")

rclpy.init()
rclpy.spin(MapSubscriber())
rclpy.shutdown()
```

C++ Example

```
#include <memory>
#include <rclcpp/rclcpp.hpp>
#include <nav_msgs/msg/occupancy_grid.hpp>

class MapSubscriber : public rclcpp::Node {
public:
    MapSubscriber() : Node("map_subscriber") {
        sub_ = create_subscription<nav_msgs::msg::OccupancyGrid>(
            "/rtabmap/map", 10,
            [this](nav_msgs::msg::OccupancyGrid::SharedPtr msg) {
                auto w = msg->info.width;
                auto h = msg->info.height;
                auto cell = msg->data[(h / 2) * w + (w / 2)];
                RCLCPP_INFO(get_logger(), "Map %ux%u, center=%d", w, h, cell);
            });
    }
private:
    rclcpp::Subscription<nav_msgs::msg::OccupancyGrid>::SharedPtr sub_;
};

int main(int argc, char ** argv) {
    rclcpp::init(argc, argv);
    rclcpp::spin(std::make_shared<MapSubscriber>());
    rclcpp::shutdown();
}
```





Mapping – Reset GLC (Global Loop Closure)

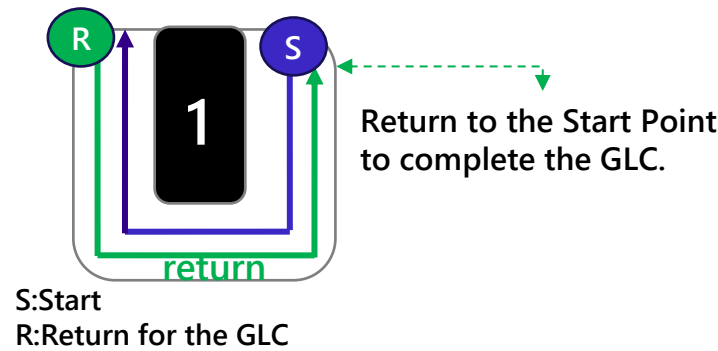
ROS2 Library API

Function	Description	Service Name	Message Type	Field Name	Message	Action
Reset GLC	Restart counting the movement distance for GLC.	/vslam_health_monitor/reset	std_srvs/srv/Trigger	success	' VSLAM monitor reset.'	When you complete the first GLC on the map and want to resume the session, please move the AMR to the next start position and activate the Reset GLC function.

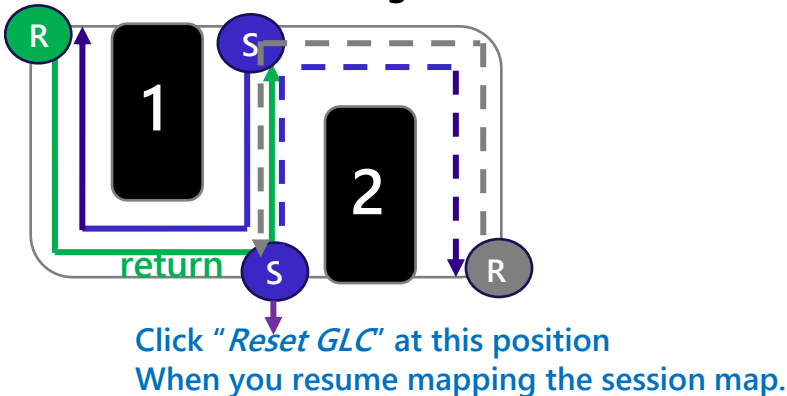
*How to perform the GLC(Global Loop Closure)

Please follow the guide below to execute the global loop closure before saving the map.

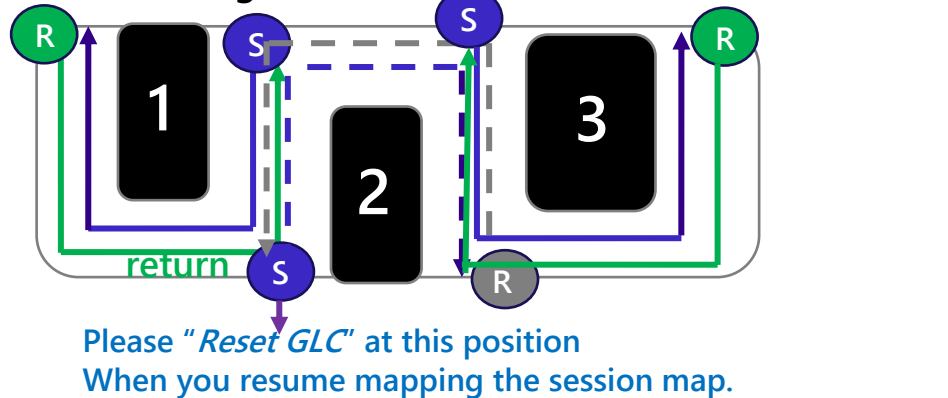
Scenario-1 : Small Area



Scenario-2 : Medium/Large Area



Scenario-3 : Large Area





Mapping & Localization - Obstacle

ROS2 Library API

Function	Description	Topic	Message / Type	Field / Parameter
2D PointCloud	2D obstacle cloud.	/<camera_ns>/obstacle_cloud	sensor_msgs/msg/PointCloud2	points
3D PointCloud	3D obstacle cloud.	/<camera_ns>/obstacle_cloud	sensor_msgs/msg/PointCloud2	points

Function	Description	Config	Message / Type	Field / Parameter
3D Enable	Enable per-camera 3D cloud output in the camera YAML.	amr_tools/config/camera_configs/*.yaml	measurement_selector	enabled: true, mode: '3d'



Mapping & Localization - Pose

ROS2 Library API

Function	Description	Topic Name	Message Type	Field name
Pose Topic	odometry for navigation / planner.	/multi_odom	nav_msgs/msg/Odometry	header.frame_id, child_frame_id, pose.pose

Function	Description	Frame Id	Message Type	Field name
TF Lookup	Use TF when the consumer needs a transform lookup.	map -> base_link	geometry_msgs/msg/TransformStamped	transform



Mapping & Localization - Pose

ROS2 Library API

Function	Description	Frame Id	Message Type	Field name
TF Lookup	Use TF when the consumer needs a transform lookup.	map -> base_link	geometry_msgs/msg/TransformStamped	transform

Python Example

```
import rclpy
from rclpy.node import Node
from rclpy.time import Time
from tf2_ros import Buffer, TransformListener

rclpy.init()
node = Node("tf_lookup")
tf_buffer = Buffer()
listener = TransformListener(tf_buffer, node)
rclpy.spin_once(node, timeout_sec=1.0)

tf = tf_buffer.lookup_transform(
    "map", "base_link", Time())
t = tf.transform.translation
node.get_logger().info(
    f"base_link in map: x={t.x:.3f}, y={t.y:.3f}")
rclpy.shutdown()
```

C++ Example

```
#include <chrono>
#include <memory>
#include <rclcpp/rclcpp.hpp>
#include <tf2_ros/buffer.h>
#include <tf2_ros/transform_listener.h>

int main(int argc, char ** argv) {
    rclcpp::init(argc, argv);
    auto node = rclcpp::Node::make_shared("tf_lookup");
    auto buffer = std::make_shared<tf2_ros::Buffer>(node->get_clock());
    auto listener = std::make_shared<tf2_ros::TransformListener>(*buffer);

    rclcpp::sleep_for(std::chrono::seconds(1));
    auto tf = buffer->lookupTransform(
        "map", "base_link", tf2::TimePointZero);
    auto t = tf.transform.translation;
    RCLCPP_INFO(node->get_logger(),
        "base_link in map: x=%.3f, y=%.3f", t.x, t.y);
    rclcpp::shutdown();
}
```





Mapping & Localization - Pose

Example Command Guide

Function	Description	Command	Remark
TF echo	Command-line transform check.	<code>ros2 run tf2_ros tf2_echo map base_link</code>	



AMR VSLAM ROS2 API Examples



AMR VSLAM ROS2 API Examples

- ROS 2 Topic, Service, and TF Quick Start

These examples show how to use the ROS2 APIs and can be run directly on ROS2.

Python Run Examples

```
# Build once
colcon build --symlink-install --packages-select amr_client_examples
source install/setup.bash

# Subscribe map
ros2 run amr_client_examples map_subscriber_py

# Load saved map
ros2 run amr_client_examples load_map_client_py /path/to/map.db

# Read robot pose from TF
ros2 run amr_client_examples tf_lookup_py
```

C++ Run Examples

```
# Build once
colcon build --symlink-install --packages-select amr_client_examples
source install/setup.bash

# Subscribe map
ros2 run amr_client_examples map_subscriber_cpp

# Load saved map
ros2 run amr_client_examples load_map_client_cpp /path/to/map.db

# Read robot pose from TF
ros2 run amr_client_examples tf_lookup_cpp
```





VSLAM Status



VSLAM Status Summary(1/2)

Insight	Description	Mode / Condition	Topic	Fields to watch
Health Gate	VSLAM status	All modes	/vslam_status	status_ok, status_label, reason
	VSLAM mode.	MAPPING / LOCALIZATION		mode, mode_label
	VSLAM odometry lost state.	All modes		odom_lost
	Accepted global loop closure happened during mapping.	MAPPING		has_map_match, last_loop_closure_id, last_match_id
Localization Match	Robot relocalized against the loaded database.	LOCALIZATION		has_map_match, matched_ref_id, last_match_stamp
	No accepted match yet: waiting for loop closure or relocalization.	Mode dependent		STATUS_WAITING_FOR_MATCH, reason
Loop Closure Status	Latest accepted match is older than the freshness threshold.	Mapping / Localization		STATUS_STALE_BY_TIME, match_elapsed_sec
	Robot traveled beyond allowed distance after mapping loop closure.			STATUS_STALE_BY_DISTANCE, travel_dist_since_match_m
	Backend info or map graph is missing or stale.			STATUS_BACKEND_STALE, map_graph_age_sec





VSLAM Status Summary(2/2)

Function	Description	Topic Name	Message Type	Field name
VO Lost State	Track Visual Odometry lost state of camera0	/camera0/odom_info	rtabmap_msgs/msg/OdomInfo	lost



We use camera0 as an example.
Please refer to the camera name as follows:
camera0
camera1
camera2
camera3

If user get the "false" information ➔ It means that the camera is not lost.
If user get the "true" information ➔ It means that the camera is **lost**.





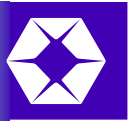
How to Get VSLAM Status Data

Example Command Guide

Method	Description	Command	Message Type	Notes / Fields
Command line	Reference for complete definitions.	ros2 interface show amr_tools/msg/VslamStatus	amr_tools/msg/VslamStatus	all fields
	Read live VSLAM state.	ros2 topic echo /vslam_status	amr_tools/msg/VslamStatus	all fields



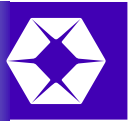
Camera Status



Camera Status Summary

Insight	Description	Topic	Fields to watch
All Camera State	React when any configured camera goes offline or recovers.	/camera_status	all_cameras_online
Per-camera State	Detailed per-camera state.		cameras[].online, last_seen_age_sec
	Discovered but no heartbeat yet.		seen=false
	Heartbeat age exceeded timeout.		online=false





How to Get Camera Status Data

Example Command Guide

Method	Description	Command	Message Type	Notes / Fields
Command line	Reference for complete definitions.	ros2 interface show <code>amr_tools/msg/CameraOnlineStatusArray</code>	CameraOnline StatusArray	array + per-camera fields
	Read live camera state.	ros2 topic echo <code>/camera_status</code>	CameraOnline StatusArray	all fields

