# Woosh Robot ROS Interface

| Version | Edit Time | Editor | Edit content |
| --- | --- | --- | --- |
| v0.0.1-beta | 2025-1-2 | HuiMin | First draft |

## Introduction

Slight

## Development and Operating Environment

1. Ubuntu 24.04 AMD64 ROS 2 **Jazzy**
2. Ubuntu 22.04 AMD64 ROS 2 **Humble**
3. Ubuntu 20.04 AMD64 ROS 2 **Foxy**
4. Ubuntu 20.04 **ARM64** ROS 2 **Foxy**

## Deploy and Run

Get the corresponding version of `ros-xxx-woosh-robot-agent_xxx_xxx.run` Installation package.

```
chmod +x ros-xxx-woosh-robot-agent_xxx_xxx.run
./ros-xxx-woosh-robot-agent_xxx_xxx.run
```

Run `agent`:

```
ros2 run woosh_robot_agent agent --ros-args -r __ns:=/woosh_robot -p
ip:="172.20.8.74"
```

Namespace `ns` Recommended to set as `/woosh_robot`, documentation and provided `Demo` are all based on this.

`ip` for the robot chassis `IP`, default is `169.254.128.2` .

# Interface Description

Currently, three types of interfaces are provided, namely `service`、`topic` and `action`

## Robot Information Related

### Get all robot information

- Interface Type: `service`
- Service Name: `robot/RobotInfo`
- Message Type: `woosh_robot_msgs/srv/RobotInfo`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/RobotInfo woosh_robot_msgs/srv/RobotInfo
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/RobotInfo --all-comments`

## Get General Information

- Interface Type: `service`
- Service Name: `robot/General`
- Message Type: `woosh_robot_msgs/srv/General`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/General woosh_robot_msgs/srv/General
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/General --all-comments`

```
# General Robot Information

# Robot Type
woosh_robot_msgs/Type type
    # Undefined
    int32 K_TYPE_UNDEFINED=0
    # General Chassis
    int32 K_BASE_ROBOT_200=1
    # Pallet Platform Lift
    int32 K_PALLET_LIFT_ROBOT_500=11
    # Mobile Cart Lift
    int32 K_SHELF_LIFT_ROBOT_500=21
    # Towing Robot
    int32 K_TRACTOR_ROBOT_500=31
    # Roller Robot
    int32 K_ROLLER_ROBOT_500=41
    # Composite Robot General Term
    int32 K_COMPLEX_ROBOT=50
    # Composite Manipulator
    int32 K_ARM_ROBOT_14=61

    int32 value
# Robot Dimensions + Self Weight + Load Capacity
woosh_robot_msgs/GeneralModelData model_data
    # Length
    uint32 length
    # Width
    uint32 width
    # Height
    uint32 height
    # Self Weight
    uint32 weight
    # Load Capacity
    uint32 load
# Model Name
string urdf_name
# Display Name
string display_model
```

```
# Robot number
uint32 serial_number
# Robot service number
string service_id
# Drive mode, 0: two-wheel differential, 1: four-wheel steering
uint32 driver_method
woosh_robot_msgs/GeneralVersion version
    # Robot system version number
    string system
    # Application module version number
    string rc
```

## Get Configuration Information

- Interface Type: `service` | `topic`
- Service Name: `robot/Setting`
- Topic Name: `robot/Setting`
- Message Type: `woosh_robot_msgs/msg/Setting`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/Setting woosh_robot_msgs/srv/Setting
ros2 topic echo /woosh_robot/robot/Setting
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/Setting --all-comments`

```
# Basic robot configuration information

# Robot identifier
woosh_robot_msgs/Identity identity
    # Robot nickname
    string name
# Connection server information
woosh_robot_msgs/Server server
    # Server IP
    string ip
    # Server port
    uint32 port
# Battery configuration information
woosh_robot_msgs/Power power
    # Warning battery value
    uint32 alarm
    # Low battery value
    uint32 low
    # Idle battery value
    uint32 idle
    # Full battery value
    uint32 full
# Sound configuration information
woosh_robot_msgs/Sound sound
    # Mute
    bool mute
```

```
    # Volume
    uint32 volume
woosh_robot_msgs/SettingAllow allow
    # Enable autonomous recharging when battery is low
    bool auto_charge
    # Enable autonomous parking when idle
    bool auto_park
    # Enable cargo detection
    bool goods_check
    # Enable mechanical detection
    bool mechanism_check
```

## Get Robot Status

- Interface Type: `service` | `topic`
- Service Name: `robot/RobotState`
- Topic Name: `robot/RobotState`
- Message Type: `woosh_robot_msgs/msg/RobotState`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/RobotState woosh_robot_msgs/srv/RobotState
ros2 topic echo /woosh_robot/robot/RobotState
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/RobotState --all-comments`

```
# Robot status
woosh_robot_msgs/State state
    # Undefined
    int32 K_STATE_UNDEFINED=0
    # Uninitialized
    int32 K_UNINIT=1
    # Idle
    int32 K_IDLE=2
    # Parking
    int32 K_PARKING=3
    # In task
    int32 K_TASK=4
    # Warning
    int32 K_WARNING=5
    # Exception
    int32 K_FAULT=6
    # Following
    int32 K_FOLLOWING=7
    # Charging
    int32 K_CHARGING=8
    # Mapping
    int32 K_MAPPING=9

    int32 value
```

## Get Mode Information

- Interface Type: `service` | `topic`
- Service Name: `robot/Mode`
- Topic Name: `robot/Mode`
- Message Type: `woosh_robot_msgs/msg/Mode`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/Mode woosh_robot_msgs/srv/Mode
ros2 topic echo /woosh_robot/robot/Mode
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/Mode --all-comments`

```
# Robot control mode information

# Control mode
woosh_robot_msgs/ControlMode ctrl
    # Undefined
    int32 K_CONTROL_MODE_UNDEFINED=0
    # Automatic
    int32 K_AUTO=1
    # Manual
    int32 K_MANUAL=2
    # Maintenance
    int32 K_MAINTAIN=3
    int32 value
# Working mode, control mode is effective when in automatic
woosh_robot_msgs/WorkMode work
    # Undefined
    int32 K_WORK_MODE_UNDEFINED=0
    # Deployment mode
    int32 K_DEPLOY_MODE=1
    # Task mode
    int32 K_TASK_MODE=2
    # Scheduling mode
    int32 K_SCHEDULE_MODE=3

    int32 value
```

## Get Pose Velocity

- Interface Type: `service` | `topic`
- Service Name: `robot/PoseSpeed`
- Topic Name: `robot/PoseSpeed`
- Message Type: `woosh_robot_msgs/msg/PoseSpeed`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/PoseSpeed woosh_robot_msgs/srv/PoseSpeed
ros2 topic echo /woosh_robot/robot/PoseSpeed
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/PoseSpeed --all-comments`

```
# Robot Pose Velocity

# Velocity
woosh_common_msgs/Twist twist
    # Linear Velocity
    float32 linear
    # Angular Velocity
    float32 angular
# Pose
woosh_common_msgs/Pose2D pose
    # x
    float32 x
    # y
    float32 y
    # Orientation
    float32 theta
# Map ID
uint32 map_id
# Cumulative Mileage, Unit m
uint32 mileage
```

## Get Battery Information

- Interface Type: `service` | `topic`
- Service Name: `robot/Battery`
- Topic Name: `robot/Battery`
- Message Type: `woosh_robot_msgs/msg/Battery`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/Battery woosh_robot_msgs/srv/Battery
ros2 topic echo /woosh_robot/robot/Battery
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/Battery --all-comments`

```
# Robot Battery Information

# Charging Status
woosh_robot_msgs/BatteryChargeState charge_state
    # Undefined
    int32 K_CHARGE_STATE_UNDEFINED=0
    # 0: Not Charging
    int32 K_NOT=1
    # 1: Manually Charging
    int32 K_MANUAL=2
```

```
    # 2: Automatically Charging
    int32 K_AUTO=3

    int32 value
# Battery percentage, values from 0 to 100, where 100 indicates fully charged and
0 indicates no power
uint32 power
# Battery health (full capacity/design capacity)
uint32 health
# Number of iterations
uint32 charge_cycle
# Battery Life
uint32 battery_cycle
# Battery Temperature (Maximum Temperature)
uint32 temp_max
```

## Get Network Information

- Interface Type: `service` | `topic`
- Service Name: `robot/Network`
- Topic Name: `robot/Network`
- Message Type: `woosh_robot_msgs/msg/Network`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/Network woosh_robot_msgs/srv/Network
ros2 topic echo /woosh_robot/robot/Network
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/Network --all-comments`

```
# Robot Network Information

# Network Connection Status
bool is_connected
# Robot IP
string robot_ip
# Scheduling IP
string sch_ip
# Robot WiFi Information
woosh_robot_msgs/NetworkWiFi wifi
    # Current Connected WiFi Name
    string name
    # Network Connection Status Code
    uint64 code
    # WiFi List in JSON Format
    uint8[] list_json
    # WiFi Signal Strength
    uint32 strength
    # WiFi Mode
    woosh_robot_msgs/NetworkWiFiMode mode
        # Undefined
        int32 K_WI_FI_MODE_UNDEFINED=0
```

```
        # AP Mode
        int32 K_AP=1
        # Switching to AP mode
        int32 K_TO_AP=2
        # Client mode
        int32 K_CLIENT=3
        # Switching to client mode
        int32 K_TO_CLIENT=4

        int32 value
```

## Get scene information

- Interface Type: `service` | `topic`
- Service Name: `robot/Scene`
- Topic Name: `robot/Scene`
- Message Type: `woosh_robot_msgs/msg/Scene`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/Scene woosh_robot_msgs/srv/Scene
ros2 topic echo /woosh_robot/robot/Scene
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/Scene --all-comments`

```
# Robot scene information

# Scene name
string scene_name
# Current map ID
uint32 map_id
# Map name
string map_name
# Map data version number
int64 version
```

## Get task progress

- Interface Type: `service` | `topic`
- Service Name: `robot/TaskProc`
- Topic Name: `robot/TaskProc`
- Message Type: `woosh_robot_msgs/msg/TaskProc`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/TaskProc woosh_robot_msgs/srv/TaskProc
ros2 topic echo /woosh_robot/robot/TaskProc
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/TaskProc --all-comments`

```
# Robot task execution information

# Robot task ID
int64 robot_task_id
# Task type
woosh_task_msgs/Type type
    # Undefined
    int32 K_TYPE_UNDEFINED=0
    # Picking
    int32 K_PICK=1
    # Parking
    int32 K_PARKING=2
    # Charging
    int32 K_CHARGE=3
    # Transporting
    int32 K_CARRY=4

    int32 value
# Task status
woosh_task_msgs/State state
    # Undefined
    int32 K_STATE_UNDEFINED=0
    # Initializing
    int32 K_INIT=1
    # Prepared
    int32 K_READY=2
    # Executing
    int32 K_EXECUTING=3
    # Paused
    int32 K_PAUSED=4
    # Action Waiting
    int32 K_ACTION_WAIT=5
    # Task Waiting
    int32 K_TASK_WAIT=6
    # Completed
    int32 K_COMPLETED=7
    # Canceled
    int32 K_CANCELED=8
    # Failed
    int32 K_FAILED=9

    int32 value
# Action Information
woosh_robot_msgs/TaskProcAction action
    # Action Type
    woosh_action_msgs/Type type
        # Undefined
        int32 K_TYPE_UNDEFINED=0
        # Navigation
        int32 K_NAV=1
        # Step Control
        int32 K_STEP_CTRL=2
        # Secondary Positioning Enter
        int32 K_SECONDPOS_ENTER=3
        # Secondary Positioning Exit
```

```
        int32 K_SECONDPOS_QUIT=4
        # Movement Actions
        int32 K_CARRY=5
        # Wait
        int32 K_WAIT=6
        # Charging
        int32 K_CHARGE=7

        int32 value
    # Action Status
    woosh_action_msgs/State state
        # Undefined
        int32 K_STATE_UNDEFINED=0
        # Executing
        int32 K_ROS_EXECUTING=1
        # Warning
        int32 K_ROS_WARNING=2
        # Cancel
        int32 K_ROS_CANCEL=3
        # Complete
        int32 K_ROS_SUCCESS=4
        # Failure
        int32 K_ROS_FAILURE=5
        # Pause
        int32 K_SUSPEND=10
        # Control
        int32 K_TRAFFI_CTRL=11

        int32 value
    # Action Wait ID
    int32 wait_id
# Destination
string dest
# Message
string msg
# Last Update Time (s)
int32 time
```

## Get the last 50 historical tasks

- Interface Type: `service`
- Service Name: `robot/TaskHistory`
- Message Type: `woosh_robot_msgs/msg/TaskHistory`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/TaskHistory woosh_robot_msgs/srv/TaskHistory
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/TaskHistory --all-comments`

## Get device status

- Interface Type: `service` | `topic`
- Service Name: `robot/DeviceState`
- Topic Name: `robot/DeviceState`
- Message Type: `woosh_robot_msgs/msg/DeviceState`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/DeviceState woosh_robot_msgs/srv/DeviceState
ros2 topic echo /woosh_robot/robot/DeviceState
```

For parameter details, see:

```
ros2 interface show woosh_robot_msgs/msg/DeviceState --all-comments
ros2 interface show woosh_robot_msgs/msg/DeviceStateHardwareBit --all-comments
ros2 interface show woosh_robot_msgs/msg/DeviceStateSoftwareBit --all-comments
```

```
# Robot Device Status

# DeviceState.HardwareBit, each bit represents a state
uint32 hardware
# DeviceState.SoftwareBit, each bit represents a state
uint32 software
```

```
# Robot Hardware Device Bit Information

# Undefined
int32 K_HARDWARE_BIT_UNDEFINED=0
# Button 1 (Pause/Continue/Next)
int32 K_BTN1=1
# Button 2 (Reset)
int32 K_BTN2=2
# Button 3
int32 K_BTN3=4
# Button 4
int32 K_BTN4=8
# Button 5
int32 K_BTN5=16
# Button 6
int32 K_BTN6=32
# Button 7
int32 K_BTN7=64
# Button 8
int32 K_BTN8=128
# Servo Release Button
int32 K_SERVO_BTN=256
# Lift Button
int32 K_LIFT_BTN=512
# Emergency stop triggered
int32 K_EMG_BTN=1024
```

```
int32 value
```

```
# Robot software device position information

# Undefined
int32 K_SOFTWARE_BIT_UNDEFINED=0
# Positioning Status
int32 K_LOCATION=1
# Scheduling Connection
int32 K_SCHEDULE=2
# Cargo Status
int32 K_GOODS_STATE=4
# Occupancy Status
int32 K_OCCUPANCY=8
# Mute Call
int32 K_MUTE_CALL=16
# Mute the program
int32 K_PROGRAM_MUTE=32

int32 value
```

## Get hardware status

- Interface Type: `service` | `topic`
- Service Name: `robot/HardwareState`
- Topic Name: `robot/HardwareState`
- Message Type: `woosh_robot_msgs/msg/HardwareState`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/HardwareState
woosh_robot_msgs/srv/HardwareState
ros2 topic echo /woosh_robot/robot/HardwareState
```

For parameter details, see:

```
ros2 interface show woosh_robot_msgs/msg/HardwareState --all-comments
ros2 interface show woosh_robot_msgs/msg/HardwareStateState --all-comments
```

## Get operational status

- Interface Type: `service` | `topic`
- Service Name: `robot/OperationState`
- Topic Name: `robot/OperationState`
- Message Type: `woosh_robot_msgs/msg/OperationState`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/OperationState
woosh_robot_msgs/srv/OperationState
ros2 topic echo /woosh_robot/robot/OperationState
```

For parameter details, see:

```
ros2 interface show woosh_robot_msgs/msg/OperationState --all-comments
ros2 interface show woosh_robot_msgs/msg/OperationStateNavBit --all-comments
ros2 interface show woosh_robot_msgs/msg/OperationStateRobotBit --all-comments
```

```
# Robot Operating Status

# OperationState.NavBit Each bit represents a state
uint32 nav
# OperationState.RobotBit Each bit represents a state
uint32 robot
```

```
# Robot Navigation Related Position Information

# Undefined
int32 K_NAV_BIT_UNDEFINED=0
# Narrow Passage
int32 K_NARROW=1
# Guide to Arrival
int32 K_GUIDE=2
# In Elevator
int32 K_INA_LIFT=4
# Obstacle
int32 K_IMPEDE=8
# QR Code
int32 K_QR_CODE=16
# Segmented Arrival
int32 K_STAGE=32

int32 value
```

```
# Robot Position Information

# Undefined
int32 K_ROBOT_BIT_UNDEFINED=0
# Available Tasks
int32 K_TASKABLE=1

int32 value
```

## Get model information

- Interface Type: `service` | `topic`
- Service Name: `robot/Model`
- Topic Name: `robot/Model`
- Message Type: `woosh_robot_msgs/msg/Model`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/Model woosh_robot_msgs/srv/Model
ros2 topic echo /woosh_robot/robot/Model
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/Model --all-comments`

```
# Robot Model

woosh_common_msgs/Point[] model
    # x
    float32 x
    # y
    float32 y
    # z
    float32 z
# Model Type
woosh_robot_msgs/FootPrint type
    # Original
    int32 K_ORIGINAL=0
    # Expansion (Carrying Cargo)
    int32 K_EXPAND=1
    # Backup
    int32 K_SPARE=2
    # Docking
    int32 K_DOCK=3

    int32 value
```

## Get exception code

- Interface Type: `service` | `topic`
- Service Name: `robot/AbnormalCodes`
- Topic Name: `robot/AbnormalCodes`
- Message Type: `woosh_robot_msgs/msg/AbnormalCodes`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/AbnormalCodes
woosh_robot_msgs/srv/AbnormalCodes
ros2 topic echo /woosh_robot/robot/AbnormalCodes
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/AbnormalCodes --all-comments`

## Subscribe to status code

- Interface Type: `topic`
- Topic Name: `robot/StatusCode`
- Message Type: `woosh_robot_msgs/msg/StatusCode`

**ros cli command**:

```
ros2 topic echo /woosh_robot/robot/StatusCode
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/StatusCode --all-comments`

## Get the last 50 status codes

- Interface Type: `service`
- Service Name: `robot/StatusCodes`
- Message Type: `woosh_robot_msgs/msg/StatusCodes`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/StatusCodes woosh_robot_msgs/srv/StatusCodes
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/StatusCodes --all-comments`

## Get navigation path

- Interface Type: `service` | `topic`
- Service Name: `robot/NavPath`
- Topic Name: `robot/NavPath`
- Message Type: `woosh_robot_msgs/msg/NavPath`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/NavPath woosh_robot_msgs/srv/NavPath
ros2 topic echo /woosh_robot/robot/NavPath
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/msg/NavPath --all-comments`

```
# Robot Navigation Path

# Navigation Path
woosh_nav_msgs/Path path
    # Path
    woosh_common_msgs/Pose2D[] poses
        # x
        float32 x
        # y
        float32 y
        # Orientation
        float32 theta
```

# Robot request related

## Initialize robot

- Interface Type: `service`
- Service Name: `robot/InitRobot`
- Message Type: `woosh_robot_msgs/srv/InitRobot`

**ros cli command**:

```
# Reset to original position
ros2 service call /woosh_robot/robot/InitRobot woosh_robot_msgs/srv/InitRobot "
{arg:{is_record: true}}"
# Reset to specified coordinates
ros2 service call /woosh_robot/robot/InitRobot woosh_robot_msgs/srv/InitRobot "
{arg:{pose:{x: 1.23, y: 2.34, theta: 1.57}}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/InitRobot --all-comments`

```
# Initialize robot

woosh_robot_msgs/InitRobot arg
    # Should record point reset
    bool is_record
    # Set robot to new coordinates
    woosh_common_msgs/Pose2D pose
        # x
        float32 x
        # y
        float32 y
        # Orientation
        float32 theta
---
# Request success or failure
bool ok
# Request status message
string msg
```

## Robot position calibration

- Interface Type: `service`
- Service Name: `robot/SetRobotPose`
- Message Type: `woosh_robot_msgs/srv/SetRobotPose`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/SetRobotPose
woosh_robot_msgs/srv/SetRobotPose "{arg:{pose:{x: 1.23, y: 2.34, theta: 1.57}}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SetRobotPose --all-comments`

```
# Set robot pose

woosh_robot_msgs/SetRobotPose arg
    woosh_common_msgs/Pose2D pose
        float32 x
        float32 y
        float32 theta
---
# Request success or failure
bool ok
# Request status message
string msg
```

## Set robot occupancy

- Interface Type: `service`
- Service Name: `robot/SetOccupancy`
- Message Type: `woosh_robot_msgs/srv/SetOccupancy`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/SetOccupancy
woosh_robot_msgs/srv/SetOccupancy "{arg:{occupy: true}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SetOccupancy --all-comments`

```
# Set robot occupancy

woosh_robot_msgs/SetOccupancy arg
    bool occupy
---
# Request success or failure
bool ok
# Request status message
string msg
```

## Set call blocking

- Interface Type: `service`
- Service Name: `robot/SetMuteCall`
- Message Type: `woosh_robot_msgs/srv/SetMuteCall`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/SetMuteCall woosh_robot_msgs/srv/SetMuteCall
"{arg:{mute: true}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SetMuteCall --all-comments`

```
# Set call masking

woosh_robot_msgs/SetMuteCall arg
    bool mute
---
# Request success or failure
bool ok
# Request status message
string msg
```

## Set program mute

- Interface Type: `service`
- Service Name: `robot/SetProgramMute`
- Message Type: `woosh_robot_msgs/srv/SetProgramMute`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/SetProgramMute
woosh_robot_msgs/srv/SetProgramMute "{arg:{mute: true}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SetProgramMute --all-comments`

```
# Set program mute

woosh_robot_msgs/SetProgramMute arg
    bool mute
---
# Request success or failure
bool ok
# Request status message
string msg
```

## Switch control mode

- Interface Type: `service`
- Service Name: `robot/SwitchControlMode`
- Message Type: `woosh_robot_msgs/srv/SwitchControlMode`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/SwitchControlMode
woosh_robot_msgs/srv/SwitchControlMode "{arg:{mode:{value: 1}}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SwitchControlMode --all-comments`

```
# Switch control mode

woosh_robot_msgs/SwitchControlMode arg
```

```
    # Robot control mode
    woosh_robot_msgs/ControlMode mode
        # Undefined
        int32 K_CONTROL_MODE_UNDEFINED=0
        # Automatic
        int32 K_AUTO=1
        # Manual
        int32 K_MANUAL=2
        # Maintenance
        int32 K_MAINTAIN=3

        int32 value
---
# Robot mode
woosh_robot_msgs/Mode ret
    # Control mode
    woosh_robot_msgs/ControlMode ctrl
        # Undefined
        int32 K_CONTROL_MODE_UNDEFINED=0
        # Automatic
        int32 K_AUTO=1
        # Manual
        int32 K_MANUAL=2
        # Maintenance
        int32 K_MAINTAIN=3

        int32 value
    # Working mode, effective when control mode is automatic
    woosh_robot_msgs/WorkMode work
        # Undefined
        int32 K_WORK_MODE_UNDEFINED=0
        # Deployment mode
        int32 K_DEPLOY_MODE=1
        # Task mode
        int32 K_TASK_MODE=2
        # Scheduling mode
        int32 K_SCHEDULE_MODE=3

        int32 value
# Request success or failure
bool ok
# Request status message
string msg
```

## Switch working mode

- Interface Type: `service`
- Service Name: `robot/SwitchWorkMode`
- Message Type: `woosh_robot_msgs/srv/SwitchWorkMode`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/SwitchWorkMode
woosh_robot_msgs/srv/SwitchWorkMode "{arg:{mode:{value: 2}}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SwitchWorkMode --all-comments`

```
# Switch working mode

woosh_robot_msgs/SwitchWorkMode arg
    # Working mode
    woosh_robot_msgs/WorkMode mode
        # Undefined
        int32 K_WORK_MODE_UNDEFINED=0
        # Deployment mode
        int32 K_DEPLOY_MODE=1
        # Task mode
        int32 K_TASK_MODE=2
        # Scheduling mode
        int32 K_SCHEDULE_MODE=3

        int32 value
---
# Robot mode
woosh_robot_msgs/Mode ret
    # Control mode
    woosh_robot_msgs/ControlMode ctrl
        # Undefined
        int32 K_CONTROL_MODE_UNDEFINED=0
        # Automatic
        int32 K_AUTO=1
        # Manual
        int32 K_MANUAL=2
        # Maintenance
        int32 K_MAINTAIN=3

        int32 value
    # Working mode, effective when control mode is automatic
    woosh_robot_msgs/WorkMode work
        # Undefined
        int32 K_WORK_MODE_UNDEFINED=0
        # Deployment mode
        int32 K_DEPLOY_MODE=1
        # Task mode
        int32 K_TASK_MODE=2
        # Scheduling mode
        int32 K_SCHEDULE_MODE=3

        int32 value
# Request success or failure
bool ok
# Request status message
string msg
```

## Switch model type

- Interface Type: `service`
- Service Name: `robot/SwitchFootPrint`
- Message Type: `woosh_robot_msgs/srv/SwitchFootPrint`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/SwitchFootPrint
woosh_robot_msgs/srv/SwitchFootPrint "{arg:{type:{value: 1}}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SwitchFootPrint --all-comments`

```
# Switch model type

woosh_robot_msgs/SwitchFootPrint arg
    # Model type
    woosh_robot_msgs/FootPrint type
        # Original
        int32 K_ORIGINAL=0
        # Expansion (Carrying Cargo)
        int32 K_EXPAND=1
        # Spare
        int32 K_SPARE=2
        # Docking
        int32 K_DOCK=3

        int32 value
---
# Model Type
woosh_robot_msgs/SwitchFootPrint ret
    # Model type
    woosh_robot_msgs/FootPrint type
        # Original
        int32 K_ORIGINAL=0
        # Expansion (Carrying Cargo)
        int32 K_EXPAND=1
        # Spare
        int32 K_SPARE=2
        # Docking
        int32 K_DOCK=3

        int32 value
# Request success or failure
bool ok
# Request status message
string msg
```

## Switch map

- Interface Type: `service`
- Service Name: `robot/SwitchMap`
- Message Type: `woosh_robot_msgs/srv/SwitchMap`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/SwitchMap woosh_robot_msgs/srv/SwitchMap "
{arg:{scene_name: "scenex"}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SwitchMap --all-comments`

```
# Switch Map

woosh_robot_msgs/SwitchMap arg
    # Scene Name
    string scene_name
    # Map Name
    string map_name
    # If empty, only switch; otherwise, update together
    woosh_common_msgs/FileData[] file_datas
        # File Name
        string name
        # File Data
        uint8[] data
---
# Request success or failure
bool ok
# Request status message
string msg
```

## Radar point cloud data

- Interface Type: `service`
- Service Name: `robot/ScannerData`
- Message Type: `woosh_robot_msgs/srv/ScannerData`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/ScannerData woosh_robot_msgs/srv/ScannerData
"{arg:{}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/ScannerData --all-comments`

```
# Radar Data Request

---
# Radar Point Cloud Data
woosh_robot_msgs/ScannerData ret
```

```
    # Start Angle of scan [radians]
    float32 angle_min
    # End Angle of scan [radians]
    float32 angle_max
    # Distance between measured angles [radians]
    float32 angle_increment
    # Time between measurements [seconds]
    float32 time_increment
    # Time between scans [seconds]
    float32 scan_time
    # Minimum measurement distance [meters]
    float32 range_min
    # Maximum measurement distance [meters]
    float32 range_max
    # Measured distance data [meters] (Note: Values < range_min or > range_max
should be discarded)
    float32[] ranges
    # Pose
    woosh_common_msgs/Pose2D pose
        # x
        float32 x
        # y
        float32 y
        # Orientation
        float32 theta
# Request success or failure
bool ok
# Request status message
string msg
```

## Execute predefined task

- Interface Type: `service`
- Service Name: `robot/ExecPreTask`
- Message Type: `woosh_robot_msgs/srv/ExecPreTask`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/ExecPreTask woosh_robot_msgs/srv/ExecPreTask
"{arg:{task_set_id: 666}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/ExecPreTask --all-comments`

```
# Execute predefined task

woosh_robot_msgs/ExecPreTask arg
    # Predefined task set ID
    int32 task_set_id
---
# Request success or failure
bool ok
# Request status message
string msg
```

## Execute task request

- Interface Type: `service`
- Service Name: `robot/ExecTask`
- Message Type: `woosh_robot_msgs/srv/ExecTask`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/ExecTask woosh_robot_msgs/srv/ExecTask "
{arg:{type:{value: 1}, mark_no: A23}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/ExecTask --all-comments`

## Action command request

- Interface Type: `service`
- Service Name: `robot/ActionOrder`
- Message Type: `woosh_robot_msgs/srv/ActionOrder`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/ActionOrder woosh_robot_msgs/srv/ActionOrder
"{arg:{order:{value: 2}}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/ActionOrder --all-comments`

```
# Action command request

woosh_robot_msgs/ActionOrder arg
    # Action command
    woosh_action_msgs/Order order
        # Undefined
        int32 K_ORDER_UNDEFINED=0
        # Start (deprecated)
        int32 K_START=1
        # Pause
        int32 K_PAUSE=2
        # Continue
        int32 K_CONTINUE=3
```

```
        # Cancel
        int32 K_CANCEL=4
        # Resume (valid for standalone tasks)
        int32 K_RECOVER=5
        # Wait for interruption
        int32 K_WAIT_BREAK=6
        # Traffic control
        int32 K_TM_CTRL=7
        # Lift control
        int32 K_RELEASE_CTRL=8

        int32 value
---
# Request success or failure
bool ok
# Request status message
string msg
```

## Change navigation path

- Interface Type: `service`
- Service Name: `robot/ChangeNavPath`
- Message Type: `woosh_robot_msgs/srv/ChangeNavPath`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/ChangeNavPath
woosh_robot_msgs/srv/ChangeNavPath "{arg:{paths:{plan_path:[{target:{x: 1.23, y:
2.34, theta: 1.57}, path:[{x: 0.0, y: 0.0, theta: 0.0}, {x: 1.23, y: 2.34, theta:
1.57}]}]}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/ChangeNavPath --all-comments`

```
# Change navigation path request

woosh_robot_msgs/ChangeNavPath arg
    # Navigation path set
    woosh_robot_msgs/PlanPath paths
        # Global planning path
        woosh_nav_msgs/PlanPath[] plan_path
            # Navigation path, cannot be empty; a single value indicates a path
planned autonomously by the standalone system
            woosh_nav_msgs/Path path
                woosh_common_msgs/Pose2D[] poses
                    # x
                    float32 x
                    # y
                    float32 y
                    # Orientation
                    float32 theta
            # Map ID of the path
            uint32 map_id
```

```
            # Destination wormhole ID; a wormhole ID of 0 indicates that the path
does not go through a wormhole
            uint32 wormhole_id
            # Map ID reached by the wormhole
            uint32 dest_map_id
            # Segment target points
            woosh_common_msgs/Pose2D target
                # x
                float32 x
                # y
                float32 y
                # Orientation
                float32 theta
            # Path optimization
            woosh_nav_msgs/PlanPathOptimal optimal
                # Undefined
                int32 K_OPTIMAL_UNDEFINED=0
                # Optimization
                int32 K_OPTIMAL=1
                # Target point optimization
                int32 K_DEST_OPTIMAL=2
                # Strict (disable optimization)
                int32 K_STRICT=9

                int32 value
---
# Request success or failure
bool ok
# Request status message
string msg
```

## Change navigation mode

- Interface Type: `service`
- Service Name: `robot/ChangeNavMode`
- Message Type: `woosh_robot_msgs/srv/ChangeNavMode`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/ChangeNavMode
woosh_robot_msgs/srv/ChangeNavMode "{arg:{nav_mode:{type:{value: 2}, mode:{value:
1}}}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/ChangeNavMode --all-comments`

```
# Change navigation mode request

woosh_robot_msgs/ChangeNavMode arg
    # Navigation mode settings
    woosh_nav_msgs/ModeSetting nav_mode
        # Navigation arrival type
        woosh_nav_msgs/ArrType type
            # Undefined
```

```
            int32 K_ARR_TYPE_UNDEFINED=0
            # Fuzzy arrival
            int32 K_VAGUE=1
            # Precise arrival
            int32 K_ACCURATE=2

            int32 value
        # Navigation mode
        woosh_nav_msgs/Mode mode
            # Undefined
            int32 K_MODE_UNDEFINED=0
            # Navigation Obstacle Avoidance
            int32 K_AVOID=1
            # Waiting...
            int32 K_NAV_WAIT=2
            # Waiting. Timeout. Replanning
            int32 K_TIMEOUT=3
            # Waiting. Timeout. Navigation Failure
            int32 K_OVERTIME=4
            # Narrow Passage
            int32 K_NARROW=10
            # Magnetic Stripe Navigation
            int32 K_MAGNETIC=11
            # QR Code Navigation
            int32 K_QRCODE=12

            int32 value
        # Effective when nav_mode is kTimeout, this parameter specifies the
timeout duration (seconds)
        uint32 wait_timeout
        # Maximum speed for navigation, defaults to the default speed when set to
0
        float32 max_speed
        # Whether passage is allowed
        bool permitted_passage
        # Passage Vehicle Count
        int32 capacity
    # Domain Entry Point
    woosh_common_msgs/Pose2D in_point
        # x
        float32 x
        # y
        float32 y
        # Orientation
        float32 theta
    # Domain Exit Point
    woosh_common_msgs/Pose2D out_point
        # x
        float32 x
        # y
        float32 y
        # Orientation
        float32 theta
---
# Request success or failure
bool ok
```

```
# Request status message
string msg
```

## Voice broadcast

- Interface Type: `service`
- Service Name: `robot/Speak`
- Message Type: `woosh_robot_msgs/srv/Speak`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/Speak woosh_robot_msgs/srv/Speak "{arg:
{text: "Hello\ World"}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/Speak --all-comments`

```
# Voice Broadcast Request

woosh_robot_msgs/Speak arg
    # Content for voice synthesis, stops broadcasting if empty
    string text
---
# Request success or failure
bool ok
# Request status message
string msg
```

## Speed control (remote control)

- Interface Type: `service`
- Service Name: `robot/Twist`
- Message Type: `woosh_robot_msgs/srv/Twist`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/Twist woosh_robot_msgs/srv/Twist "{arg:
{linear: 0.2, angular: 0.785}}"
```

**Description**: This interface requires continuous requests. After stopping the request, the robot will smoothly decelerate to a speed of 0. To stop the robot immediately, a speed of 0 must be actively sent, i.e.: `{arg:{linear: 0.0, angular: 0.0}}`

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/Twist --all-comments`

```
# Speed Control Request

woosh_robot_msgs/Twist arg
    # Linear velocity, unit is m/s, positive value moves forward
    float32 linear
```

```
    # Angular velocity, unit is radians/s, positive value rotates
counterclockwise
    float32 angular
    # Linear velocity y, unit is m/s, positive value moves forward
    float32 linear_y
---
# Request success or failure
bool ok
# Request status message
string msg
```

### Follow Request

- Interface Type: `service`
- Service Name: `robot/Follow`
- Message Type: `woosh_robot_msgs/srv/Follow`

**ros cli command**:

```
ros2 service call /woosh_robot/robot/Follow woosh_robot_msgs/srv/Follow "{arg:
{type: true}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/Follow --all-comments`

```
# Follow request

woosh_robot_msgs/Follow arg
    # 1: Enable automatic follow, 0: Disable automatic follow
    bool type
---
# Request success or failure
bool ok
# Request status message
string msg
```

## Robot Settings Related

### Set Robot Identifier

- Interface Type: `service`
- Service Name: `setting/SetIdentity`
- Message Type: `woosh_robot_msgs/srv/SetIdentity`

**ros cli command**:

```
ros2 service call /woosh_robot/setting/Identity woosh_robot_msgs/srv/SetIdentity
"{arg:{name: "woow"}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SetIdentity --all-comments`

```
# Set identifier

woosh_robot_msgs/Identity arg
    # Robot nickname
    string name
---
# Robot identifier
woosh_robot_msgs/Identity ret
    # Robot nickname
    string name
# Request success or failure
bool ok
# Request status message
string msg
```

## Set Server Connection

- Interface Type: `service`
- Service Name: `setting/Server`
- Message Type: `woosh_robot_msgs/srv/SetServer`

**ros cli command**:

```
ros2 service call /woosh_robot/setting/Server woosh_robot_msgs/srv/SetServer "
{arg:{name: "woow"}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SetServer --all-comments`

```
# Set connection server address

woosh_robot_msgs/Server arg
    # Server IP
    string ip
    # Server port
    uint32 port
---
# Connection server address
woosh_robot_msgs/Server ret
    # Server IP
    string ip
    # Server port
    uint32 port
# Request success or failure
bool ok
# Request status message
string msg
```

## Switch Autonomous Charging

- Interface Type: `service`
- Service Name: `setting/AutoCharge`
- Message Type: `woosh_robot_msgs/srv/SetAutoCharge`

**ros cli command**:

```
ros2 service call /woosh_robot/setting/AutoCharge
woosh_robot_msgs/srv/SetAutoCharge "{arg:{allow: true}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SetAutoCharge --all-comments`

```
# Switch autonomous recharging

woosh_robot_msgs/AutoCharge arg
    # Allow or not
    bool allow
---
# Autonomous recharging
woosh_robot_msgs/AutoCharge ret
    # Allow or not
    bool allow
# Request success or failure
bool ok
# Request status message
string msg
```

## Switch Autonomous Parking

- Interface Type: `service`
- Service Name: `setting/AutoPark`
- Message Type: `woosh_robot_msgs/srv/SetAutoPark`

**ros cli command**:

```
ros2 service call /woosh_robot/setting/AutoPark woosh_robot_msgs/srv/SetAutoPark
"{arg:{allow: true}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SetAutoPark --all-comments`

```
# Switch autonomous parking

woosh_robot_msgs/AutoPark arg
    # Allow or not
    bool allow
---
# Autonomous parking
woosh_robot_msgs/AutoPark ret
    # Allow or not
```

```
    bool allow
# Request success or failure
bool ok
# Request status message
string msg
```

## Switch Cargo Detection

- Interface Type: `service`
- Service Name: `setting/GoodsCheck`
- Message Type: `woosh_robot_msgs/srv/SetGoodsCheck`

**ros cli command**:

```
ros2 service call /woosh_robot/setting/GoodsCheck
woosh_robot_msgs/srv/SetGoodsCheck "{arg:{allow: true}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SetGoodsCheck --all-comments`

```
# Switch cargo inspection

woosh_robot_msgs/GoodsCheck arg
    # Allow or not
    bool allow
---
# Cargo inspection
woosh_robot_msgs/GoodsCheck ret
    # Allow or not
    bool allow
# Request success or failure
bool ok
# Request status message
string msg
```

## Charging Power Configuration

- Interface Type: `service`
- Service Name: `setting/Power`
- Message Type: `woosh_robot_msgs/srv/SetPower`

**ros cli command**:

```
ros2 service call /woosh_robot/setting/Power woosh_robot_msgs/srv/SetPower "{arg:
{alarm: 5, low: 20, idle: 80, full: 100}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SetPower --all-comments`

```
# Power configuration

woosh_robot_msgs/Power arg
```

```
    # Warning battery value
    uint32 alarm
    # Low battery value
    uint32 low
    # Idle battery value
    uint32 idle
    # Full battery value
    uint32 full
---
# Power configuration
woosh_robot_msgs/Power ret
    # Warning battery value
    uint32 alarm
    # Low battery value
    uint32 low
    # Idle battery value
    uint32 idle
    # Full battery value
    uint32 full
# Request success or failure
bool ok
# Request status message
string msg
```

## System Sound Settings

- Interface Type: `service`
- Service Name: `setting/Sound`
- Message Type: `woosh_robot_msgs/srv/SetSound`

**ros cli command**:

```
ros2 service call /woosh_robot/setting/Sound woosh_robot_msgs/srv/SetSound "{arg:
{mute: false, volume: 50}}"
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/srv/SetSound --all-comments`

```
# Sound settings

woosh_robot_msgs/Sound arg
    # Mute
    bool mute
    # Volume
    uint32 volume
---
# Sound settings
woosh_robot_msgs/Sound ret
    # Robot sound settings

    # Mute
    bool mute
    # Volume
    uint32 volume
```

```
# Request success or failure
bool ok
# Request status message
string msg
```

# Map Related

## Obtain Scene Data

- Interface Type: `service`
- Service Name: `map/SceneDataEasy`
- Message Type: `woosh_map_msgs/srv/SceneDataEasy`

**ros cli command**:

```
ros2 service call /woosh_robot/map/SceneDataEasy woosh_map_msgs/srv/SceneDataEasy
"{}"
```

Parameter description can be found in `ros2 interface show woosh_map_msgs/srv/SceneDataEasy --all-comments`

```
# Get Scene Data (Easy)

---
woosh_map_msgs/SceneDataEasy info
    # Simple Version of Scene Data

    # Scene Name
    string name
    # Map Information Data
    woosh_map_msgs/SceneDataEasyMap[] maps
        # Map ID
        uint32 id
        # Map Name
        string name
        # Floor Name
        string floor
        # Map Version
        int64 version
        # Storage Location Set
        woosh_map_msgs/Storages storages
            # Storage Location Collection
            woosh_map_msgs/StoragesBase[] bases
                # Identifier
                woosh_map_msgs/Identity identity
                    # ID (Unique)
                    uint32 id
                    # Number (Unique)
                    string no
                    # Description
                    string desc
                # Pose
                woosh_map_msgs/Pose pose
```

```
                    # Docking Point Coordinates
                    woosh_common_msgs/Pose2D dock
                        # x
                        float32 x
                        # y
                        float32 y
                        # Orientation
                        float32 theta
                    # Actual Coordinates
                    woosh_common_msgs/Pose2D real
                        # x
                        float32 x
                        # y
                        float32 y
                        # Orientation
                        float32 theta
                # Custom Field
                uint8[] custom
```

# Task Related

## Obtain Predefined Task List

- Interface Type: `service`
- Service Name: `task/RepeatTasks`
- Message Type: `woosh_task_msgs/srv/RepeatTasks`

**ros cli command**:

```
ros2 service call /woosh_robot/task/RepeatTasks woosh_task_msgs/srv/RepeatTasks "{}"
```

Parameter description can be found in `ros2 interface show woosh_task_msgs/srv/RepeatTasks --all-comments`

## Obtain Call Task List

- Interface Type: `service`
- Service Name: `task/CallTasks`
- Message Type: `woosh_task_msgs/srv/CallTasks`

**ros cli command**:

```
ros2 service call /woosh_robot/task/CallTasks woosh_task_msgs/srv/CallTasks "{}"
```

Parameter description can be found in `ros2 interface show woosh_task_msgs/srv/CallTasks --all-comments`

# Action

## Task Execution

- Interface Type: `action`
- Service Name: `robot/ExecTask`
- Message Type: `woosh_robot_msgs/action/ExecTask`

**ros cli command**:

```
ros2 action send_goal /woosh_robot/robot/ExecTask
woosh_robot_msgs/action/ExecTask "{arg:{type:{value: 1}, mark_no: A2}}" --
feedback
```

Parameter description can be found in `ros2 interface show woosh_robot_msgs/action/ExecTask --all-comments`

```
# Task Execution

woosh_robot_msgs/ExecTask arg
    # Task ID
    int64 task_id
    # Task Type
    woosh_task_msgs/Type type
        # Task Type

        # Undefined
        int32 K_TYPE_UNDEFINED=0
        # Picking
        int32 K_PICK=1
        # Parking
        int32 K_PARKING=2
        # Charging
        int32 K_CHARGE=3
        # Transport
        int32 K_CARRY=4

        int32 value
    # Action Direction
    woosh_task_msgs/Direction direction
        # Direction

        # Undefined
        int32 K_DIRECTION_UNDEFINED=0
        # Loading
        int32 K_FEEDING=1
        # Unloading
        int32 K_CUTTING=2

        int32 value
    # Type Combination
    uint32 task_type_no
    # Target Point Number (Choose one of three)
    string mark_no
```

```
    # Navigation Path Set (Choose one of three)
    woosh_robot_msgs/PlanPath plan_path
        # Robot Global Planning Path

        # Global planning path
        woosh_nav_msgs/PlanPath[] plan_path
            # Planned Path

            uint8 PATH_FIELD_SET=1
            uint8 TARGET_FIELD_SET=16

            # Navigation path, cannot be empty; a single value indicates a path
planned autonomously by the standalone system
            woosh_nav_msgs/Path path
                # Path (...)
    # Pose (Choose one of three)
    woosh_common_msgs/Pose2D pose
        # x
        float32 x
        # y
        float32 y
        # Orientation
        float32 theta
    # Custom field, varies by project
    uint8[] custom
---
woosh_robot_msgs/TaskProc ret
    # Robot task ID
    int64 robot_task_id
    # Task Type
    woosh_task_msgs/Type type
        # Undefined
        int32 K_TYPE_UNDEFINED=0
        # Picking
        int32 K_PICK=1
        # Parking
        int32 K_PARKING=2
        # Charging
        int32 K_CHARGE=3
        # Transport
        int32 K_CARRY=4

        int32 value
    # Task status
    woosh_task_msgs/State state
        # Undefined
        int32 K_STATE_UNDEFINED=0
        # Initialization
        int32 K_INIT=1
        # Prepared
        int32 K_READY=2
        # Executing
        int32 K_EXECUTING=3
        # Paused
        int32 K_PAUSED=4
        # Action waiting
```

```
        int32 K_ACTION_WAIT=5
        # Task waiting
        int32 K_TASK_WAIT=6
        # Completed
        int32 K_COMPLETED=7
        # Canceled
        int32 K_CANCELED=8
        # Failed
        int32 K_FAILED=9

        int32 value
    # Action information
    woosh_robot_msgs/TaskProcAction action
        # Action type
        woosh_action_msgs/Type type
            # Undefined
            int32 K_TYPE_UNDEFINED=0
            # Navigation
            int32 K_NAV=1
            # Single step control
            int32 K_STEP_CTRL=2
            # Secondary positioning entry
            int32 K_SECONDPOS_ENTER=3
            # Secondary positioning exit
            int32 K_SECONDPOS_QUIT=4
            # Transport action
            int32 K_CARRY=5
            # Waiting
            int32 K_WAIT=6
            # Charging
            int32 K_CHARGE=7

            int32 value
        # Action status
        woosh_action_msgs/State state
            # Undefined
            int32 K_STATE_UNDEFINED=0
            # Executing
            int32 K_ROS_EXECUTING=1
            # Warning
            int32 K_ROS_WARNING=2
            # Cancel
            int32 K_ROS_CANCEL=3
            # Completion
            int32 K_ROS_SUCCESS=4
            # Failure
            int32 K_ROS_FAILURE=5
            # Pause
            int32 K_SUSPEND=10
            # Control
            int32 K_TRAFFI_CTRL=11

            int32 value
        # Action waiting ID
        int32 wait_id
    # Destination
```

```
    string dest
    # Message
    string msg
    # Last update time(s)
    int32 time
---
woosh_robot_msgs/TaskProc fb
    # Robot task ID
    int64 robot_task_id
    # Task Type
    woosh_task_msgs/Type type
        # Undefined
        int32 K_TYPE_UNDEFINED=0
        # Picking
        int32 K_PICK=1
        # Parking
        int32 K_PARKING=2
        # Charging
        int32 K_CHARGE=3
        # Transport
        int32 K_CARRY=4

        int32 value
    # Task status
    woosh_task_msgs/State state
        # Undefined
        int32 K_STATE_UNDEFINED=0
        # Initialization
        int32 K_INIT=1
        # Prepared
        int32 K_READY=2
        # Executing
        int32 K_EXECUTING=3
        # Paused
        int32 K_PAUSED=4
        # Action waiting
        int32 K_ACTION_WAIT=5
        # Task waiting
        int32 K_TASK_WAIT=6
        # Completed
        int32 K_COMPLETED=7
        # Canceled
        int32 K_CANCELED=8
        # Failed
        int32 K_FAILED=9

        int32 value
    # Action information
    woosh_robot_msgs/TaskProcAction action
        # Action type
        woosh_action_msgs/Type type
            # Undefined
            int32 K_TYPE_UNDEFINED=0
            # Navigation
            int32 K_NAV=1
            # Single step control
```

```
            int32 K_STEP_CTRL=2
            # Secondary positioning entry
            int32 K_SECONDPOS_ENTER=3
            # Secondary positioning exit
            int32 K_SECONDPOS_QUIT=4
            # Transport action
            int32 K_CARRY=5
            # Waiting
            int32 K_WAIT=6
            # Charging
            int32 K_CHARGE=7

            int32 value
        # Action status
        woosh_action_msgs/State state
            # Undefined
            int32 K_STATE_UNDEFINED=0
            # Executing
            int32 K_ROS_EXECUTING=1
            # Warning
            int32 K_ROS_WARNING=2
            # Cancel
            int32 K_ROS_CANCEL=3
            # Completion
            int32 K_ROS_SUCCESS=4
            # Failure
            int32 K_ROS_FAILURE=5
            # Pause
            int32 K_SUSPEND=10
            # Control
            int32 K_TRAFFI_CTRL=11

            int32 value
        # Action waiting ID
        int32 wait_id
    # Destination
    string dest
    # Message
    string msg
    # Last update time(s)
    int32 time
```

## Stepping Control

- Interface Type: `action`
- Service Name: `ros/StepControl`
- Message Type: `woosh_ros_msgs/action/StepControl`

**ros cli command**:

```
# Step forward
ros2 action send_goal /woosh_robot/ros/StepControl
woosh_ros_msgs/action/StepControl "{arg:{action:{value: 1}, steps:[{mode:{value:
1}, speed: 0.5, value: 2}]}}" --feedback
# Step rotation
ros2 action send_goal /woosh_robot/ros/StepControl
woosh_ros_msgs/action/StepControl "{arg:{action:{value: 1}, steps:[{mode:{value:
2}, speed: 0.78, value: 3.14}]}}" --feedback
```

Parameter description can be found in `ros2 interface show woosh_ros_msgs/action/StepControl --all-comments`

```
# Step control

woosh_ros_msgs/StepControl arg
    # Step control set
    woosh_ros_msgs/StepControlStep[] steps
        # Control Mode
        woosh_ros_msgs/StepControlStepMode mode
            # Undefined
            int32 K_NONE=0
            # Move Forward
            int32 K_STRAIGHT=1
            # Rotate
            int32 K_ROTATE=2
            # Lateral Move
            int32 K_LATERAL=3
            # Diagonal Move
            int32 K_DIAGONALIZE=4

            int32 value
        # Rotation Angle/Travel Distance, Positive Forward Negative Backward,
Positive Counterclockwise Negative Clockwise, Positive Left Negative Right
        float32 value
        # Angular Velocity (radians/s)/Linear Velocity (m/s)
        float32 speed
        # Diagonal Movement Angle, Positive Left Negative Right
        float32 angle
    # 0: Enable Obstacle Avoidance, 1: Disable Obstacle Avoidance
    int32 avoid
    # Control Action
    woosh_ros_msgs/ControlAction action
        # Cancel
        int32 K_CANCEL=0
        # Execute
        int32 K_EXECUTE=1
        # Pause
        int32 K_PAUSE=2
        # Continue
        int32 K_RESUME=3

        int32 value
---
woosh_ros_msgs/Feedback ret
```

```
    # ros feedback

    # action name, e.g. woosh.ros.action.StepControl
    string action
    # Status
    woosh_ros_msgs/State state
        # Undefined
        int32 K_ROS_NONE=0
        # Cancel
        int32 K_ROS_CANCEL=-2
        # Failure
        int32 K_ROS_FAILURE=-1
        # Complete
        int32 K_ROS_SUCCESS=1
        # Executing
        int32 K_ROS_EXECUTING=2
        # Pause
        int32 K_ROS_PAUSE=3
        # Pause Failure
        int32 K_ROS_PAUSE_FAILED=4
        # Execution Failure
        int32 K_ROS_EXECUTE_FAILED=5
        # Exception Message
        int32 K_ROS_ERR_MSG=10
        # WiFi Request Status Code
        int32 K_ROS_WI_FI_CODE=100
        # WiFi Information JSON
        int32 K_ROS_WI_FI_JSON=101

        int32 value
    # Status Code
    uint64 code
    # Message
    string msg
---
woosh_ros_msgs/Feedback fb
    # action name, e.g. woosh.ros.action.StepControl
    string action
    # Status
    woosh_ros_msgs/State state
        # Undefined
        int32 K_ROS_NONE=0
        # Cancel
        int32 K_ROS_CANCEL=-2
        # Failure
        int32 K_ROS_FAILURE=-1
        # Complete
        int32 K_ROS_SUCCESS=1
        # Executing
        int32 K_ROS_EXECUTING=2
        # Pause
        int32 K_ROS_PAUSE=3
        # Pause Failure
        int32 K_ROS_PAUSE_FAILED=4
        # Execution Failure
        int32 K_ROS_EXECUTE_FAILED=5
```

```
        # Exception Message
        int32 K_ROS_ERR_MSG=10
        # WiFi Request Status Code
        int32 K_ROS_WI_FI_CODE=100
        # WiFi Information JSON
        int32 K_ROS_WI_FI_JSON=101

        int32 value
    # Status Code
    uint64 code
    # Message
    string msg
```

## Lifting Mechanism Control

- Interface Type: `action`
- Service Name: `ros/LiftControl`
- Message Type: `woosh_ros_msgs/action/LiftControl`

**ros cli command**:

```
# Lift Up
ros2 action send_goal /woosh_robot/ros/LiftControl
woosh_ros_msgs/action/LiftControl "{arg:{action:{value: 1}, execute_mode:{value:
1}}}" --feedback
# Lift Down
ros2 action send_goal /woosh_robot/ros/LiftControl
woosh_ros_msgs/action/LiftControl "{arg:{action:{value: 1}, execute_mode:{value:
2}}}" --feedback
```

Parameter description can be found in `ros2 interface show woosh_ros_msgs/action/LiftControl --all-comments`

```
# Lifting Mechanism Control

woosh_ros_msgs/LiftControl arg
    # Execution Mode
    woosh_ros_msgs/LiftControlExecuteMode execute_mode
        int32 K_NONE_EXECUTE_MODE=0
        # Ascend
        int32 K_UP=1
        # Descend
        int32 K_DOWN=2

        int32 value
    # Control Action
    woosh_ros_msgs/ControlAction action
        # Cancel
        int32 K_CANCEL=0
        # Execute
        int32 K_EXECUTE=1
        # Pause
        int32 K_PAUSE=2
        # Continue
```

```
        int32 K_RESUME=3

        int32 value
---
woosh_ros_msgs/Feedback ret
---
woosh_ros_msgs/Feedback fb
```

## Lifting Device Control

- Interface Type: `action`
- Service Name: `ros/LiftControl3`
- Message Type: `woosh_ros_msgs/action/LiftControl3`

**ros cli command**:

```
# Absolute Position
ros2 action send_goal /woosh_robot/ros/LiftControl3
woosh_ros_msgs/action/LiftControl3 "{arg:{action:{value: 1}, execute_mode:{value:
1}, speed: 0.2, height: 0.5}}" --feedback
# Relative Position
ros2 action send_goal /woosh_robot/ros/LiftControl3
woosh_ros_msgs/action/LiftControl3 "{arg:{action:{value: 1}, execute_mode:{value:
2}, speed: 0.2, height: 0.2}}" --feedback
```

Parameter description can be found in `ros2 interface show woosh_ros_msgs/action/LiftControl3 --all-comments`

```
# Lifting Mechanism Control 3

woosh_ros_msgs/LiftControl3 arg
    # Execution Mode
    woosh_ros_msgs/LiftControl3ExecuteMode execute_mode
        # Query Status
        int32 K_QUERY=0
        # Absolute Position
        int32 K_ABSOLUTE=1
        # Relative Position
        int32 K_RELATIVE=2
        # Position Calibration
        int32 K_CALIBRATION=3
        # Test Mode
        int32 K_TSET_MODE=4

        int32 value
    # Speed (m/s)
    float32 speed
    # Height (m), positive value goes up, negative value goes down
    float32 height
    uint32 flags
    # Control Action
    woosh_ros_msgs/ControlAction action
        # Cancel
        int32 K_CANCEL=0
```

```
        # Execute
        int32 K_EXECUTE=1
        # Pause
        int32 K_PAUSE=2
        # Continue
        int32 K_RESUME=3

        int32 value
---
woosh_ros_msgs/Feedback ret
---
woosh_ros_msgs/Feedback fb
```

## Basic Navigation

- Interface Type: `action`
- Service Name: `ros/MoveBase`
- Message Type: `woosh_ros_msgs/action/MoveBase`

**ros cli command**:

```
ros2 action send_goal /woosh_robot/ros/MoveBase woosh_ros_msgs/action/MoveBase "
{arg:{poses:[{x: 0.57, y: 2.54, theta: 1.57}], target_pose:{x: 0.57, y: 2.54,
theta: 1.57}}}" --feedback
```

Parameter description can be found in `ros2 interface show woosh_ros_msgs/action/MoveBase --all-comments`

```
# Basic Navigation

woosh_ros_msgs/MoveBase arg
    # Navigation Path
    woosh_common_msgs/Pose2D[] poses
        # x
        float32 x
        # y
        float32 y
        # Orientation
        float32 theta
    # Target Point
    woosh_common_msgs/Pose2D target_pose
        # x
        float32 x
        # y
        float32 y
        # Orientation
        float32 theta
    # Execution Mode
    woosh_ros_msgs/MoveBaseExecutionMode execution_mode
        # Free Execution
        int32 K_FREE=0
        # Point-by-Point Execution
        int32 K_ONE_BY_ONE=1
```

```
        int32 value
    # Control Action
    woosh_ros_msgs/ControlAction action
        # Cancel
        int32 K_CANCEL=0
        # Execute
        int32 K_EXECUTE=1
        # Pause
        int32 K_PAUSE=2
        # Continue
        int32 K_RESUME=3

        int32 value
---
woosh_ros_msgs/Feedback ret
---
woosh_ros_msgs/Feedback fb
```