

# AI4DI: BUT-INTRA interfacing

This document describes interfacing between the robotic system developed by BUT, and high-level control and visualization system developed by INTRA.

## Tools

The robotic system is ROS2-based; it uses Foxy Fitzroy ROS version running on Ubuntu 20.04.2.0 LTS (Focal Fossa) operating system. For the interfacing, standard ROS communication tools are utilized; *topics*, *services* and *actions*, specifically.

## Interface overview

There are two communication directions:

1. **Robotic system (BUT) → Control and visualization (INTRA)**: robots publish various data into specific topics on a regular basis (robot pose, status etc.); INTRA subscribes to the topics to visualize the data.
2. **Control and visualization (INTRA) → Robotic system (BUT)**: INTRA calls services and actions provided by robots; robots then provide desired data or perform a desired task.

Available interfaces are summarized in the following table:

Name	Type	Interface file	Frequency
pose	topic	PoseStamped.msg (common)	10 Hz
status	topic	DiagnosticStatus.msg (common)	1 Hz
assigned_task	service	AssignedTask.srv (custom)	N/A
task_go_to	action	TaskGoTo.action (custom)	N/A
task_go_home	action	TaskGoHome.action (custom)	N/A
task_transport	action	TaskTransport.action (custom)	N/A

## Interface details

### pose [topic]

#### Description

Robots publish their pose (position and orientation) and timestamp into this topic.

#### Interface file

geometry\_msgs/msg/PoseStamped.msg (common)

[https://github.com/ros2/common\\_interfaces/blob/master/geometry\\_msgs/msg/PoseStamped.msg](https://github.com/ros2/common_interfaces/blob/master/geometry_msgs/msg/PoseStamped.msg)

## status [topic]

### Description

Robots publish their status into this topic. The *level* variable contains the overall robot status, the following values are possible:

State	Value	Description
OK	0	Everything is ok
WARN	1	Something is wrong, functionality may be limited, see <i>values</i>
ERROR	2	Critical problem, see <i>values</i>

Additional information is passed via key-value *values* variable. It can, for example, contain following data:

Key	Value	Description
battery	ok / low / critical	Battery status
busy	no / yes	Indicates whether the robot performs a task
assigned_task	<task ID> *	Task being currently performed

\* The ID of each task must be a unique string (a hash, for example).

### Interface file

diagnostic\_msgs/msg/DiagnosticStatus.msg (common)

[https://github.com/ros2/common\\_interfaces/blob/master/diagnostic\\_msgs/msg/DiagnosticStatus.msg](https://github.com/ros2/common_interfaces/blob/master/diagnostic_msgs/msg/DiagnosticStatus.msg)

## assigned\_task [service]

### Description

A service returning the ID of the task being currently performed. If no task is assigned, it returns an empty string.

### Interface file

AssignedTask.srv (custom)

## task\_go\_to [action]

### Description

An action provided by the robot to navigate the robot to specific coordinates.

### Interface file

TaskGoTo.action (custom)

## task\_go\_home [action]

### Description

An action provided by the robot to navigate the robot to home (parking) coordinates.

### Interface file

TaskGoHome.action (custom)

## task\_transport [action]

### Description

An action provided by the robot to transport an object. It requires the coordinates of the object and target. This action represents an experimental functionality, it may change in the future.

### Interface file

TaskTransport.action (custom)

## Namespacing

The system may consist of numerous robots, each of them offering the aforementioned interfaces. To distinguish between the robots, namespaces are utilized. The full interface name is assembled as follows:

```
/factory/robot_<ID>/<interface name>
```

where the <ID> stands for a unique robot identifier (positive integer), and the <interface name> for the interface name (see table above).

## Examples

/factory/robot\_1/pose - a topic containing robot 1 poses

/factory/robot\_24/task\_go\_to - an action for navigating robot 24 to certain coordinates

## Source files

All interfacing source files are available at BUT GitHub repository

<https://github.com/BUT-INTRA/interfaces>. Stable releases are available at *main* branch. The repository also contains a simple simulator for interfacing and visualization testing.

## Notes

- All units (robots, computers etc.) in the same network may access any ROS2 interface (topics, services, actions) by nature. The robotic system developed by BUT uses a large number of interfaces to ensure proper functionality, however, only the interfaces described in this document are intended for BUT-INTRA interfacing.