Robotics map user manual

Introduction

The purpose of the robotic map is to provide an interface that is:
- displaying and updating relative and global position of robots and other objects,
- allowing to issue commands to robots
- providing situational awareness to the operator
- allowing globally and locally aware robots to share localization frames.
- resenting information in customizable layers.

This manual will cover functionalities of the robotic map and how they can be accessed using GUI and control devices (mouse, keyboard).

The interface is intended to work with touchscreen devices with no mouse or keyboard present, even with devices that have low touchscreen resolution (and with which, precise pointing on the map is not possible).

Using the interface: connecting to a robot

To make proper use of the robotic map, a robot must be nearby, broadcasting ROS beacons (see fakeRobot example on how this should be done). Once your robot is available, it will automatically show on the map. If the view isn't currently centered on your robot, click the menu item representing the robot on the left middle side of the map.

To issue commands to a robot, you need to connect to it. This is intended to support multi-robot scenarios as a way to indicate to which robot the command is given.

To connect to the robot, click it's icon on the left middle list. The icon will change to indicate that this robot is now selected – a small green "ok" icon will appear in the bottom right corner of the robot icon, both on the list and on the map.

Adding map layers

To add a layer with satellite imagery, just place the Tileserver format directory in your local installation data (i.e. ./local/share/marble/maps/earth/). It should follow the structure of the openstreetmap directory that is already there.

Two types of layers are automatically loaded from this location. One is the default aerial images layer – the folder must be named "aerial". All other layers can be named "overlay_x" where x can be a number or something else.

Other than that, local map is not considered a layer (it uses the overlay mechanism so it can be positioned and rotated freely). See installation guide on how to deploy a local map. There can be only one local map at a given moment currently.

Navigating around with the map
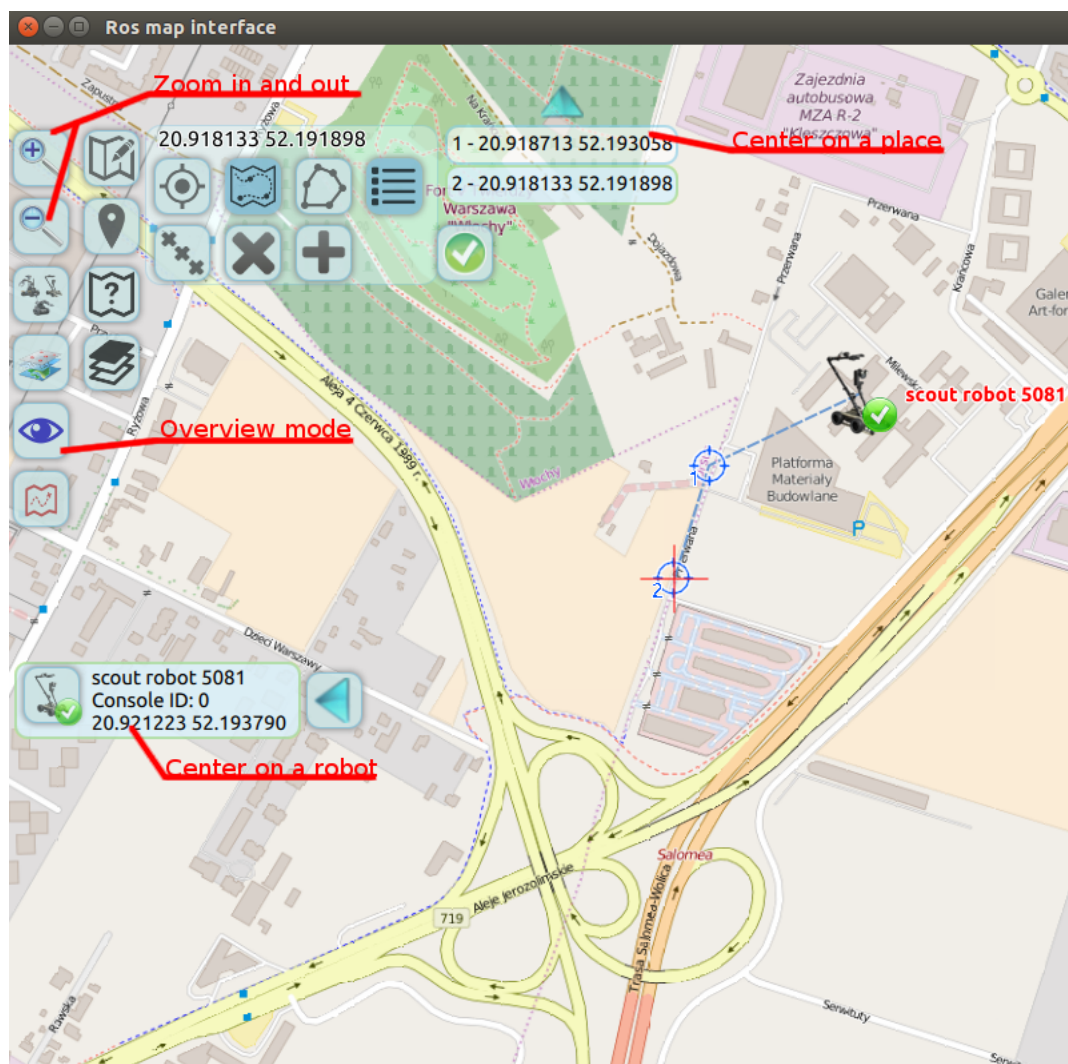
Picture 1: navigating the map

- Zooming in and out can be achieved with:
  - magnifying glass icons
  - pressing + or – on a keyboard
  - using mouse wheel, both rolling and pressing.
  - mouse double click (for zoom in)
  - touchscreen pinch.
- Moving around the map (scrolling):
  - dragging with left mouse button pressed
  - touching and dragging on the touchscreen
  - arrow keys
- Moving to a robot, place, view:
  - can center view on a robot, place, or an overview, by clicking particular interface elements – see *Picture 1*.
  - keys 1-9 will also center view on robots.

Overview of map controls:
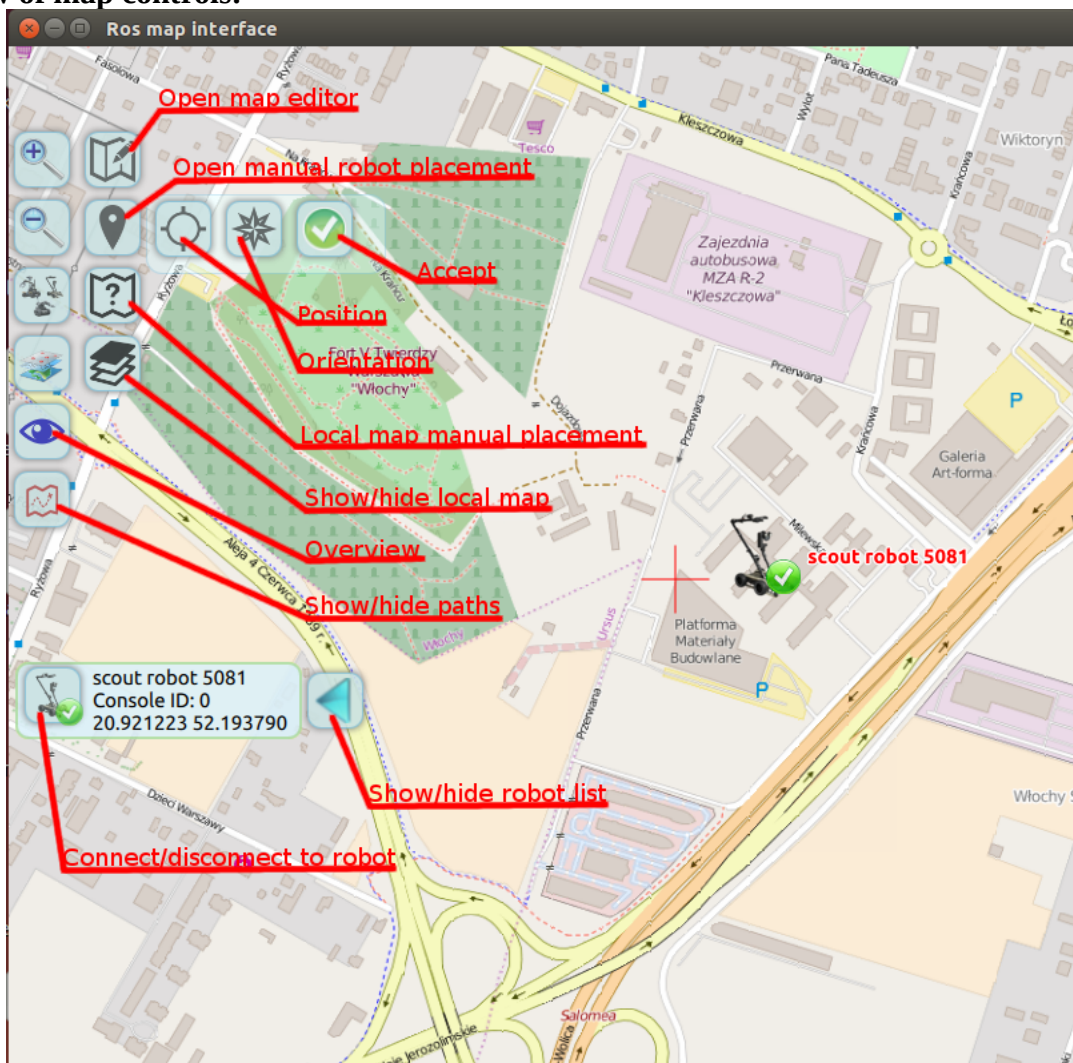
Picture 2: overview of controls

- Map editor is described in more detail in the next section.
- Manual robot placement can currently be only used for robots that are locally positioned:
  - Choose the robot position by aiming with the crosshair.
  - Choose orientation by clicking to a position that the robot should be facing. For more precise aiming, you can also press and hold the orientation control to enable another mode, where free movement and visual indication of orientation are available.
  - New placement must be confirmed with the green "ok" icon or it will be reverted.
- Local map can be manually placed, too. It is necessary to deploy the local map correctly first (see installation guide), so it is available displayed. More details in the local map section.
- Paths followed by robots can be shown and hidden, also by using "P" key. Paths are still in alpha version.
- 3rd icon from the top switches view mode of the robots. Available modes are "full icons" - the default one; "small icons" - triangles or circles are shown instead of full robot icons, depending on whether the orientation is available, "hidden" - robots are not shown.
- 4th icon from the top manages layers. It will toggle between available layers (open street maps, satellite imagery, custom layers).

**Map editor**
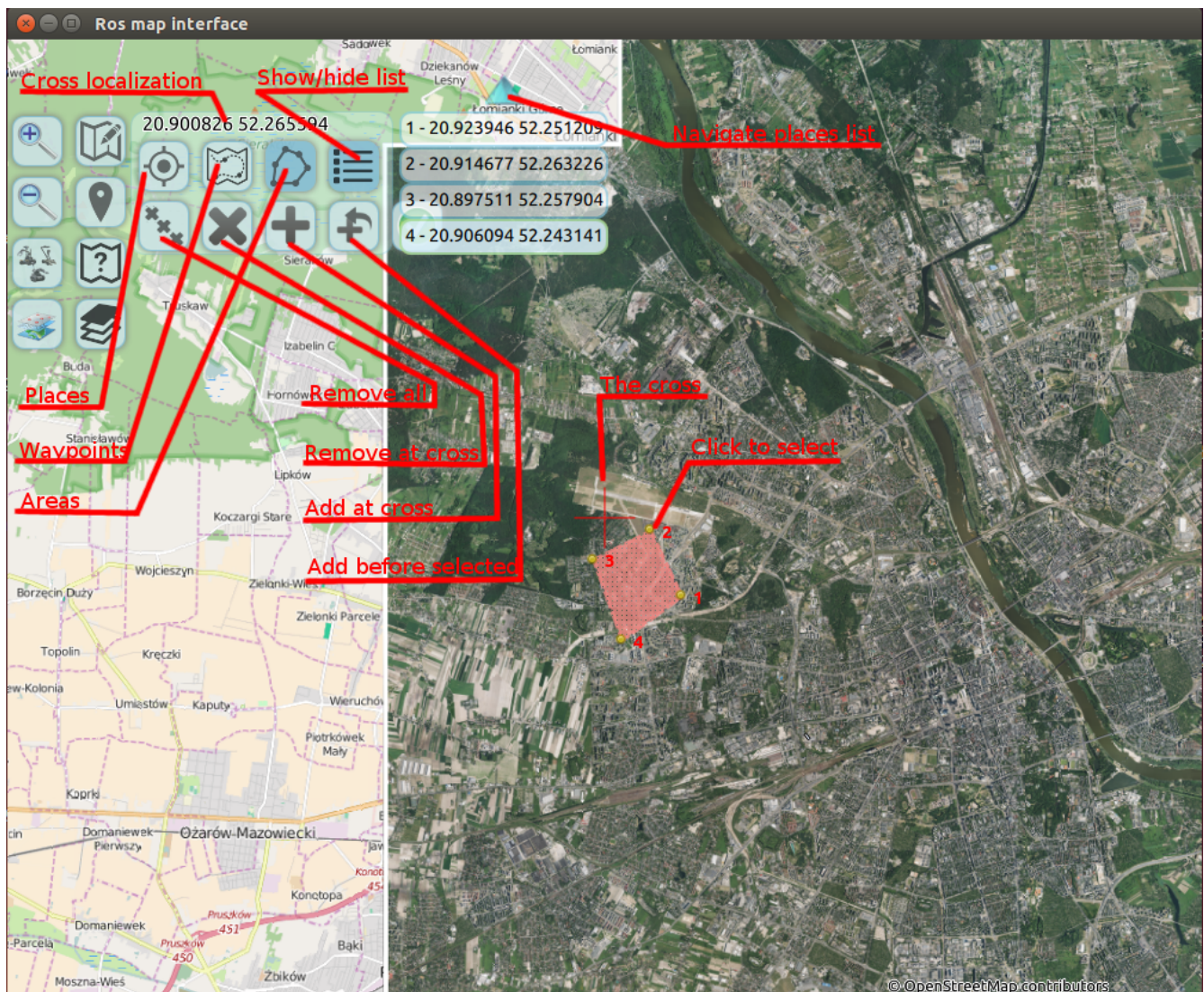
Picture 3: map editor

Map editor is a way to place or remove objects, waypoints and areas on the map. For precise input of such objects, the crosshair appears in the middle of the map while the edit mode is active. The edit mode can be turned on or off by clicking the map edit icon, just right from the zoom + icon.

The crosshair indicates the place for the edit action and is always in the middle of the map. On devices with poor touchscreen resolution, it's much easier to aim such crosshair instead of trying to point to a place where one would like to add a feature. Precise crosshair position is displayed above the edit panel. Icons for adding and deleting objects are always refering to what's under the crosshair (*Picture 3*).

The changes are not yet persistent and the map editor is to be further developed. Areas or places are for operator's information only. However, waypoints can be sent to the robot as a navigate command, by placing them for a connected robot and pressing the green "ok" button (see Picture 1)

Places, waypoints and area nodes can be selected, by clicking on them directly or by selecting the entry on places list. This enables the control of "adding before" for ordered features (waypoints, area nodes), so that a point can be added in a middle of the sequence.

Local map
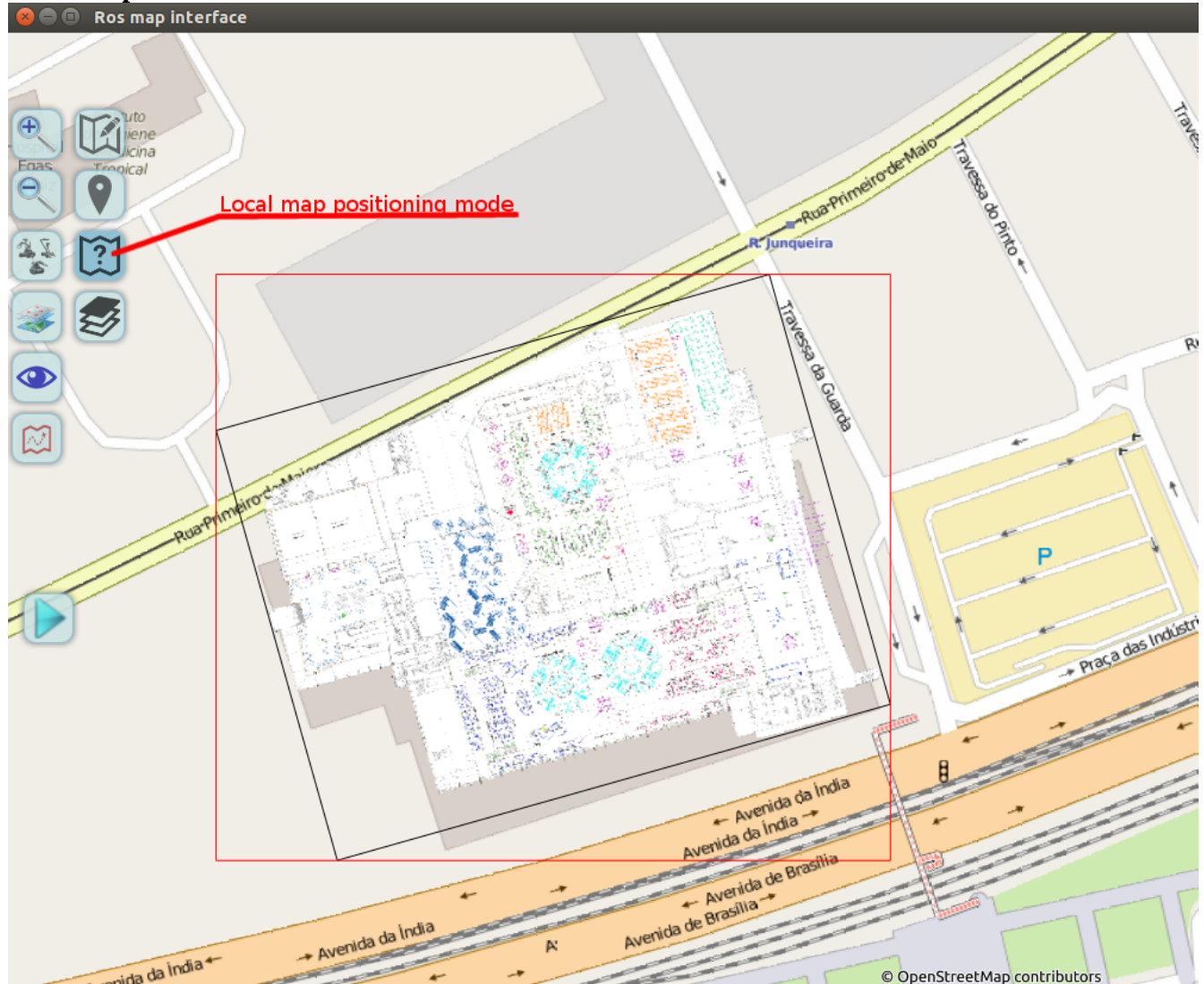
Picture 4: Local map overlay

Picture 4 shows the toggleable icon that allows for local map positioning. The map can be moved around by dragging it's central part. It can also be rotated:

- By using right mouse button.
- By grabbing the outer parts of the local map frame (this is important for touchscreen devices).

The final position is currently not persisted between application runs.

All robots that are locally positioned will appear on the local map (or close to it). (0, 0) position is in the lower left corner of the map. The position is in meters. All local robots will retain their position relative to local map, even when the map is rotated and positioned. Their global position will also be correct as long as their local position is correct and the map is correctly positioned.