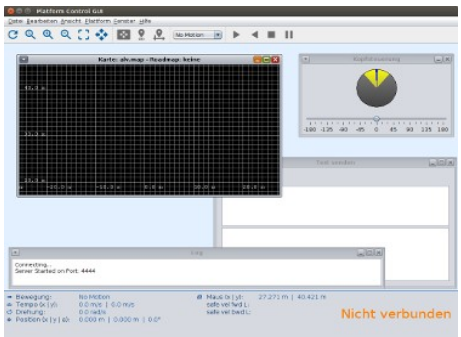
NEOBOTIX

PlatformCtrlGUI

Operating Manual



23 August 2016

Contents

1 Introduction

1.1 Contents of this document

This operating manual explains how to set up and use the Neobotix platform control software to get an idea of its functionality and performance.

There are two possibilities to use this software: real-world and simulation mode. In simulation mode no platform hardware is necessary. The original platform controller software is running on the host-computer and simulates all sensor signals.

In detail you will learn how to:

- Start and initialise the Java user interface application PlatformCtrlGUI and the robot control software PlatformCtrl
- Establish a socket connection between these two programs
- Create a map
- Create a roadmap
- Move the robot within a simulated or its real-world environment

All these actions and most of the GUI's features are described in detail. So for most customers and applications this document should provide all the information necessary to set up the mobile robot.

If any further information is needed, for example when doing research or installing very specialised applications, please refer to the Neobotix “PlatformCtrl – Programmer's Guide” or contact Neobotix.

1.2 Format of this document

Once the hardware has been set up according to the operating manual, the robot platform will operate very safely. Changing any settings in the GUI will not cause danger to the platform hardware or affect the safety system.

Nevertheless, highest safety can only be achieved if all parameters are set correctly! To help setting up a safe and convenient application the following icons mark important paragraphs:

The warning triangle marks paragraphs that concern the danger of injuries, damages or both. Please make sure to read these paragraphs very carefully!

The prohibition sign marks actions or applications that are explicitly prohibited and which might result in damages, reduced life expectancy and loss of warranty.

The light bulb marks paragraphs which deal with common problems, misunderstandings and errors and is meant to be a help in troubleshooting.

2 Installing the host computer

A common PC system is sufficient to run the Neobotix graphical user interface.

2.1 Windows operating system

Installing the Java Runtime Environment

The Neobotix GUI is programmed in Java and thus platform independent. This also means that a Java Runtime Environment and in some cases a library for 3D-calculations must be installed before using the GUI.

If there is none or an older version on the host computer please download the latest Java version from www.java.com and install it manually. In case any additional software is required, please check the content of the CD or DVD that was included in delivery.

Installing the GUI

The graphical user interface does not need to be installed. Simply copy the folder “NeoPltfGUI_<version>” from the disk to your hard drive.

To run the GUI just double-click on *start.bat* or create a shortcut on your desktop by right-dragging the *start.bat*-icon onto the desktop and selecting *Create shortcut here* from the pop-up-menu.

2.2 Linux operating system

Installing the Java Runtime Environment

Most common Linux distributions already come with an installed Java Runtime Environment. Please make sure that a Java 7 Runtime Environment or later is available. For Debian/Ubuntu distributions installation starts after entering

```
apt-get install openjdk-7-jre
```

on root command line. For other distributions please use your package manager.

Installing the GUI

The graphical user interface does not need to be installed. Instead simply copy the folder “PltfGUI” from the CD to your hard drive.

To run the GUI open a terminal, change to the location of “PltfGUI” and enter

```
java -jar PlatformCtrlGUI.jar
```

2.3 Setting up the network

If ordered, a wireless LAN-device was delivered together with the platform to allow easy connecting to the robot. In case the settings of this device need to be changed, please use the software on the according driver disc and work directly on the platform’s on-board computer as described in chapter “Maintenance”.

Make sure that both platform and host PC are in the same subnet. Please refer to the system administrator of the local computer network.

3 Getting started

3.1 Starting the required programs

To control the robot, two components are needed:

- The platform control software PlatformCtrl on the platform itself
- The Java based graphical user interface PlatformCtrlGUI

PlatformCtrl

The control software on the platform itself is started automatically. If for any reason the software is not running, please restart the robot.

If the platform has a small LC-Display the start of the control software will be indicated there.

The operating system is a Debian based Linux. The operating system on the robot platform may be write protected using fsprotect to avoid corruption. In this case any changes (e. g. wireless LAN settings) are discarded after reboot.

To see which partition is protected open a terminal and enter:

```
mount | grep fsprotect
```

If you see something like in the following picture the partition is protected.

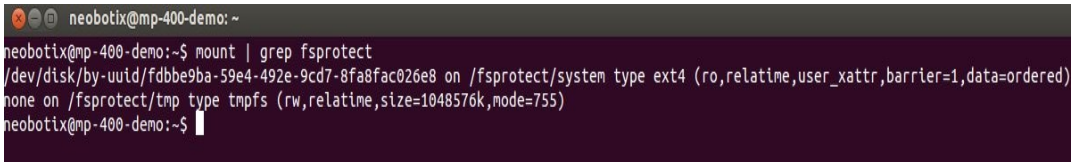


Figure 1: Result of a write protection check

To temporarily remove the protection please press key *e* as soon as the GRUB boot manager menu appears.

Figure 2: The GRUB boot manager menu

Search for the entry **fsprotect=1G** or **fsprotect=auto.** Remove this kernel boot option (remove only the text that is marked in the picture) and reboot by pressing F10.

Figure 3: The boot options

To permanently remove the protection (not recommended) please edit the grub boot manager settings in “/etc/default/grub”:

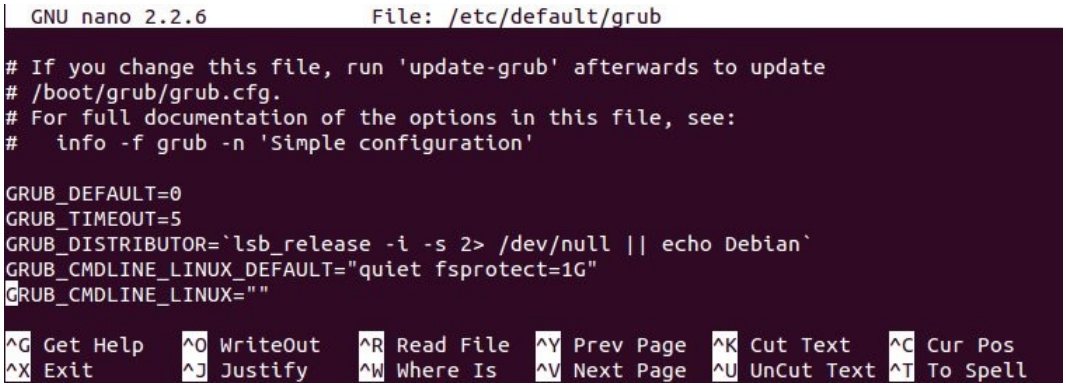


Figure 4:The grub boot manager

To make the setting permanent enter:

```
sudo update-grub
```

PlatformCtrlGUI

To start the PlatformCtrlGUI please follow the instructions in chapter “Installing the host computer” depending on your choice of operating system.

3.2 Connecting to the platform

A connection must be established between the GUI and the platform. Use the main menu to open the connection dialogue (*Platform → Connect Platform*). If the platform is not listed in the *Saved host* area, its IP-address must be entered under *New Host*. If the option *Remember this host* has been ticked, a new dialogue will appear after clicking *Connect*. The name entered in this dialogue will then be added to the list of known platforms.

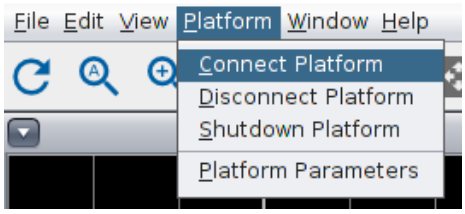


Figure 5: Connection menu

In the status field at the bottom of the main window, the status of the connection will change from an orange *Disconnected* to a green *Ready*. The left hand side of the window shows the status of the platform (i.e. Mode Motion, Speed, Rotation, Position).

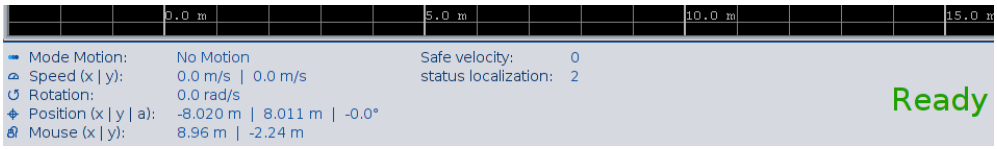
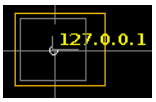


Figure 6: Status field of the platform

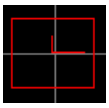
3.3 The map window

The map window gives a view over the area around the platform. As shown in the figure below, the legends indicate as follow:

- **Green lines/points:** Visualize data received from the platform's scanner.
- **Blue lines:** Represent straight obstacles such as boxes or walls.



Gray robot-icon: Indicates the position of the robot on the map. Its orientation is indicated by the longer line in the middle. When a map has been created and loaded, the platform continues updating its position. The yellow rectangle shows the robot's collision radius.



Red robot-icon: Shows the position where the robot should be during automatic motion (following a roadmap, driving to a specified target frame). In this case the grey robot-icon usually follows the red robot-icon. If the robot is not performing automatic movement, the red icon can be ignored.

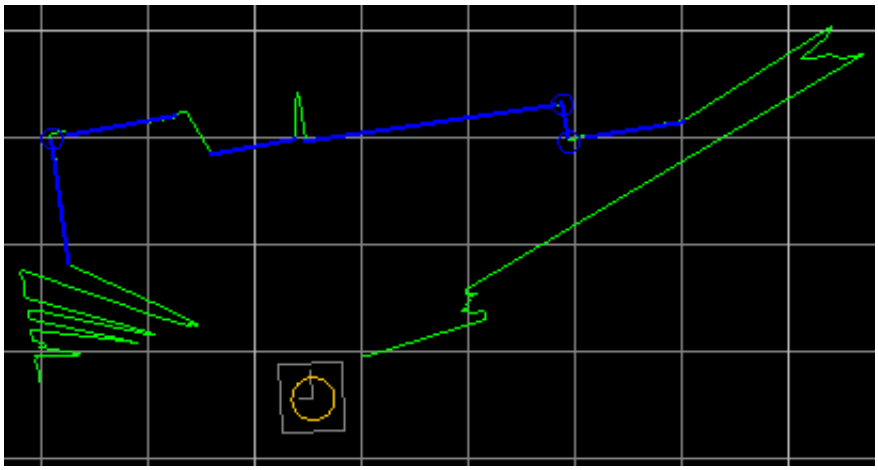


Figure 7: Initial scan as displayed in the map window

3.4 The map toolbar

The map tool bar can be found below the menu of the main window. It contains the most important controls for handling the map.

Figure 8: The map toolbar

Refresh map: Reload the map from the mobile robot.

Zoom all: Use this button to automatically zoom the map window to fit all objects that are currently visible.

Zoom in: Decrease the zoom factor to show more details.

Zoom out: Increase the zoom factor to get a better overview of the robots surrounding.

Zoom window: After clicking this button, drag a rectangle over the desired area in the map window to automatically fit it to the map window. Right click to deselect this tool.

Pan view: Click the button to drag the current view of the map without zooming. Right click to deselect this tool.

Move platform: Drag the centre of the robot-icon to rotate, drag outside the icon to move it. Right click to deselect this tool.

Set target frame: Specify a location on the map as new target position.
See chapter “Moving the platform” for further information on this function.

Set target station: Specify a station of a roadmap as new target position.
See the chapter “Moving the platform” for further information on this function.

3.5 Using background pictures

Technical drawings of the workspace can be used as background picture of the map view. This allows to create maps and roadmaps in advance, without a robot being placed in the

workspace.

It is not possible to convert drawings or CAD data to maps or roadmaps automatically.

Open the configuration dialogue via the main menu points *View* → *View options*. Now select *Map* in the box at the left side of the dialogue and use the button *Open picture* to select and load a picture file from the hard drive.

You can adjust resolution as well as X and Y offsets by using the edit fields below. Click the button *Apply* to apply your changes to the map view.

4 Mapping

For the robot to move autonomously, a map of the environment has to be created.

This map contains distinctive features of the robot's surrounding as detected by the laser scanner. The robot compares this information to the features that are extracted from the scan at runtime and thus is able to identify its current position and orientation relative to the map's origin.

Additionally multiple areas with different effects on the robot's behaviour and actions can be defined in the map. Prohibited zones can be integrated as well as workspaces or user defined information for more convenient handling of the robot.

4.1 The map editor

In order to create and later edit a map manually, you have to start the map editor. Therefore select *Edit* from the menu bar and then *Edit Map*.

On the left hand side, you should now see the map editor toolbar. It contains tools for saving and loading map files to and from hard disk, uploading to as well as downloading from the platform and also for adding and editing obstacles.

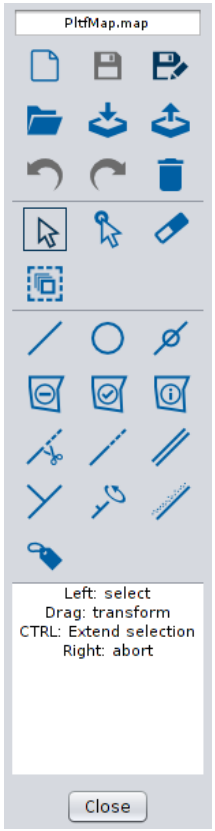


Figure 9: Map Editor

If you want to know more about each tool, simply move the mouse over to one of the buttons and a small tool tip will appear. After selecting a tool more information on its usage is shown at the bottom of the editor.

Click *Close* at the bottom of the tool bar to close the map editor. You will then be asked to save changes to the hard disk and/or upload them to the platform.


Before beginning to create a new map you should orient the platform to fit your environment. Select the *Move platform* tool then click and drag in the map until the station points into the right direction. The origin of the map should have an appropriate counterpart in the real world, e.g. the corner of a big hall or the main door to your building.

4.2 Creating objects in a map

Creating a map is very much a recursive process. One marks what the robot “sees” at its current position, moves the platform a small distance and then marks what it “sees” at its new position. For marking the obstacles which the robot recognizes, use the tools described below.

The hardware joystick is the best way to move the robot at this stage.

You should periodically upload your edited map to the platform. As the robot uses its current map to track its position, it might loose its position if there is no map present for the current area.

Do not use too many landmarks. Best results are achieved when using a small number of large and easy to see landmarks. Using temporary landmarks, like movable crates or other non-static objects, should be avoided.

Adding lines to a map

Straight landmarks can be added either manually or semi-automatically. Adding them manually is the preferable method since it allows adding additional knowledge to the map. The robot can only extract and add line segments from the current scan. But the operator knows that for example the two wall segments currently scanned actually belong to one single wall which stretches five more metres beyond the scanned area and should be parallel to the map's X-axis.

Select the *Line* tool. Now point at the beginning of a blue line in the map window, click and drag to the end of the blue line. Repeat this process for any blue line, you see in the scan. Right click to abort.

Another way of adding lines to the map is by using the *Convert line from scan* tool. Select this tool and click on a blue line in the scan. A new line will be created, that fits the scanned straight obstacle. Right click to abort.

Every line has one visible and one invisible side. The visible side is marked by a small line segment in the middle of each landmark. Use the tool *invert line direction* to change the orientation of a line.

Adding circles to a map

Using only straight landmarks might not be enough to create a usable map. Round objects in the environment, e.g. columns, can be represented as circles in the map.

Select the *Circle* tool to create a circle. Left click where you want its centre to be and drag the mouse to define the size of the circle. Do not worry if you do not accurately hit the centre. You can still edit objects after creating them as described later. Right click to abort.

Adding reflectors to a map

In areas with only a very small number of usable features and landmarks the localisation and navigation can be improved by using reflectors. These highly reflective markers, which can be ordered from Neobotix as well, are used as additional landmarks and are displayed as magenta coloured dots in the GUI.

Select *Reflector*, click in the map, drag the mouse and release it to create a reflector. The centre of the reflector mark is where the scanner recognizes some highly reflective material. If you missed the right location you can change the reflector's position as described later on. Right click to abort.

You can also add reflectors by just clicking on them in the scan. Select the *Convert line from scan* tool and then click on a reflector in the scan. As reflectors are usually applied to walls you have to point exactly at the recognized reflector when you create it. Right click to abort.

Adding prohibited zones to a map

There might be locations where you do not want your platform to move through such as an area which contains small obstacles that are not visible to the robot or an area which is likely to have dropping or fragile objects.

To create a prohibited zone select the *Prohibited zone* tool and define the starting point of the area by clicking in the map. Move the mouse where you want to create the second point of the prohibited zone. When you create the last point, double click and the polygon is closed automatically.
You can also create rectangular areas by pressing the *Ctrl* key while dragging.

The prohibited zones are no safety feature. It is possible for a robot to enter prohibited areas due to loss of localisation or other errors. In order to ensure a safe operation in areas with physical dangers additional measures must be taken.

Never rely on prohibited zones only to keep a robot from falling down stairs or edges. Additional safety measures must be taken to avoid serious injuries and damages.

Adding a workspace to a map

A workspace reflects the area you do not want the platform to leave. If there is more than one platform operating it might be easier to separate their workspaces.

Select the *Workspace* tool, click to define the first point, then click for each additional point and at the last point double click to close the workspace.
You can also create rectangular areas by pressing the *Ctrl* key while dragging.
Right click to abort.

You can only define one workspace on a map. Defining a new workspace deletes the existing one.

Adding polygons and labels on a map

Sometimes it is hard to recognize where certain rooms are if the map does not contain every single wall. The following tools can be used to make the map look more comprehensible while not affecting the robot's behaviour.

To create an area, select the *Hidden polygon* tool. Click to define the first point, then click for each additional point and double click to close the polygon.
You can also create rectangular areas by pressing the *Ctrl* key while dragging.

To create a label, select the *Hidden label* tool. Click on the position in the map that you want to label. An *Edit label* dialogue will appear. Type the name of the label under *New text*, and click *OK* to confirm. Click on the label again to change the name.

4.3 Editing objects

Selecting, moving, rotating and scaling objects

Activate the *Select object and transform* tool. Click the object you want to edit. Hold the *Ctrl* button to add other objects to the current selection. Click and drag the mouse to select all objects within a rectangular area.
A white frame with solid squares on the vertices and circles on the sides will appear, enveloping the objects. These are handles that allow you to scale (squares) or rotate (circles) all selected objects.

To select all objects on a map at the same time, use the *Select all* tool.

- Lines: To move the entire line, click inside the white rectangle, hold the mouse-button and drag the line. By dragging one of the square boxes on the edges of the white rectangle you can change the line's length. By dragging the circles on the sides, you change its orientation.
- Circles: To move the circle, click inside the white rectangle and drag. By dragging one of the square boxes on the edges of the white rectangle you can change the radius of the circle.
- Reflectors and polygons can be modified like lines.

You can use the arrow keys on your keyboard to achieve the most accurate positioning of selected objects.

Change the step size under *Edit* → *Options* → *Grid* → *Cursor step*.

Changing the vertices of lines and polygons

To move points on a line or a polygon, select the *Select vertices and edit* tool. Click on any object to highlight its vertices. Select the vertex you want to edit and drag it to the new position.
Left double click adds a vertex, right double click or pressing *Del* removes a vertex.

Point coordinates can also be manipulated numerically. Activate the *Select vertices and edit* tool and press the *Ctrl* key while clicking a point. A dialogue will appear in which the X and Y coordinates can be adjusted in millimetre steps.

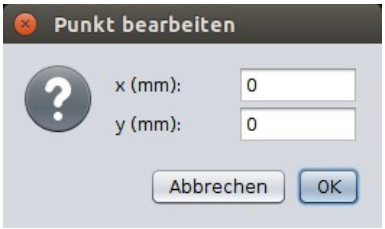


Figure 10: Dialoge "Edit point"

Deleting objects

Deleting an object is rather simple. Select the *Delete object* tool and simply click on any object to delete it.

Trimming lines

Select the *Trim line* tool to cut a line with respect to a second one. Click the line you want to trim and then select another line as a reference. The first line will be cut at the intersection of both lines, leaving the longer part.

Extending lines

Select the *Extend line* tool to extend a line until it reaches another one. Click the line you want to extend, then click on a reference line. The first line will be extended to the intersection of both lines.

Making lines parallel to each other

If you want to make lines parallel to each other select the *Make line parallel* tool. Click on the line you want to be parallel to the reference line and then click the reference line.

Making lines orthogonal to each other

Lines can also be made orthogonal to each other. Select the *Make line orthogonal* tool and then click on the line you want to adjust. Click on another line as a reference line. The first line is made orthogonal to reference line.

Clearing the map

The *Clear all* tool can be used to remove all elements from the map if you want to start anew.

Undo and redo changes

Select “*Undo*” to go back to the previous step.

Select “*Redo*” to recover changes.

Completing the map creation process

Once a map has been finished, it should look similar to the example below.

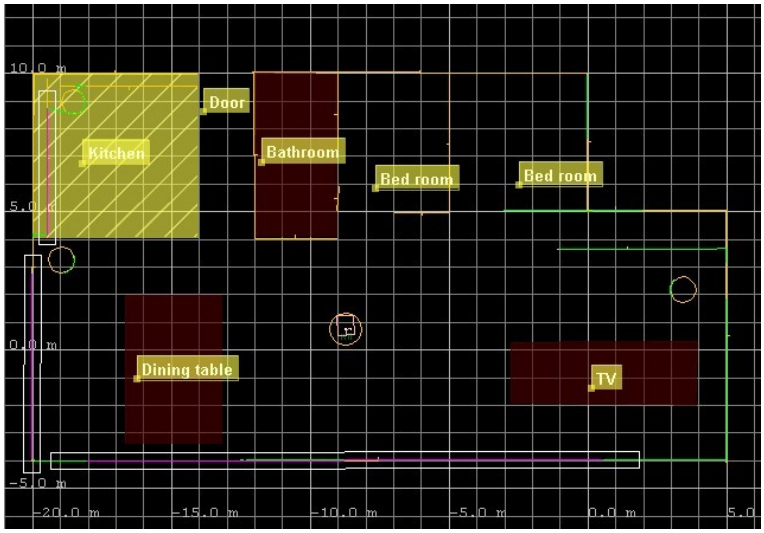


Figure 11: A map of an apartment

After the map has been created it needs to be saved and uploaded to the platform in order to make the changes available to the control software.

First the map should be saved to the local hard disk by clicking the *Save* button. The first time a map is saved a dialogue will appear, asking for a file name and the location you want to save the map to.

Select the *Save as* button if you want to save your work under another file name and continue working on this new file.

Do not forget to upload the map to the platform before exiting the map editor. Click the *Save on platform* button and the map will be sent to the platform.

Click the *Close* button at the bottom of the toolbar to close the map editor. If you cannot see this button, you may have to scroll down a bit. If you have not yet saved the latest changes to the map you will be asked whether you want to save the map to a file and/or upload it to the platform.

5 Creating a roadmap

When a map has been created, a roadmap should be used to tell the robot on which path it may move to reach any predefined target station. This significantly decreases the time required for planning and also prevents the robot from planning a path through obstacles which are currently neither in its field of view nor in the map. That way both safety of operations as well as repeatability can be ensured. Once a map of the environment in which the robot is meant to move has been created, it is easy to add a roadmap.

By default the robot loads the last used map after rebooting. If no or the wrong map has been loaded, select the appropriate map and upload it to the platform (*Edit → Edit Map, Load , Save on platform*). Close the map editor with the button on the bottom of the editor window.

In some cases the robot might not be able to find its position in the map immediately. In that case use the *Move Platform* button (drag the centre of robot-icon to rotate, drag outside the icon to move).

5.1 The roadmap editor

In order to create and later edit a roadmap, you have to start the roadmap editor. Therefore select *Edit* from the menu bar and then *Edit Roadmap*.

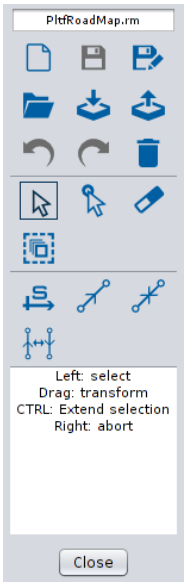


Figure 12: The Roadmap Editor

On the left hand side the Roadmap toolbar will appear. It contains tools for saving and loading roadmap files to and from hard disk, uploading them to as well as downloading them from the platform and also for adding and editing roadmap elements.

If you want to know more about one of the tools simply move the mouse over the button. After a second a small tool tip appears, after clicking on the button the usage of the selected tool will be explained at the bottom of the toolbar.

The *Close* button at the bottom of the toolbar closes the roadmap editor. You will then be asked to save changes to hard disk and / or upload them to the platform.

5.2 Creating stations and paths on a roadmap

Adding stations to the roadmap

Stations are basically the positions and poses the mobile robot is meant to reach when commanded to. They are always defined by their X- and Y- coordinates in the map and by the direction in which the robot will be facing.

To add a station to the roadmap, select the *Station* tool. Click anywhere on the map to create a station. For a roadmap you have to define at least two stations. Drag an existing station with the right mouse button to rotate it. Left click a station to open a parameter dialogue for fine adjustments.

Adding and editing paths

Paths are the connections between stations. The platform reaches the target position by following paths from its current position.

To define a path, select the *Path (unidirectional)* tool and click on the starting station. Create defining points of the path by clicking on the map. Finally click on second station to create the path. Repeat the procedure until all desired stations are connected.

Click *Invert path direction* and left-click on any segment of the path you want to change the direction of.

Every path has only one direction for the robot to move. This can be used to define one-way paths in areas where turning around might be difficult or unsafe or where the robot should follow the main traffic direction.

In order to easily create normal routes, two paths can be created at the same time. Use the *Path (bidirectional)* tool to create two paths at once with antipodal directions.

5.3 Editing roadmap objects

Selecting, moving, rotating and scaling objects

Activate the *Select object and transform* tool. Click the object you want to edit. Hold the *Ctrl* button to add other objects to the current selection. Click and drag the mouse to select all objects within a rectangular area.
A white frame with solid squares on the vertices and circles on the sides will appear, enveloping the objects. These are handles that allow you to scale (squares) or rotate (circles) all selected objects.

To select all objects on a map at the same time, use the *Select all* tool.

Inserting path-points into edges

You can insert path-points into edges to manipulate the way the platform should move without inserting more stations.

Select the *Select vertices and edit* tool. Select the path segment you want to insert a point into by clicking it. Then double click to insert a new point. Click on any point and drag to redefine the path. Double right-click or press *Del* to delete a point.

Point coordinates can also be manipulated numerically. Activate the *Select vertices and edit* tool and press the *Ctrl* key while clicking a point. A dialogue will appear in which the X

and Y coordinates can be adjusted in millimetre steps.

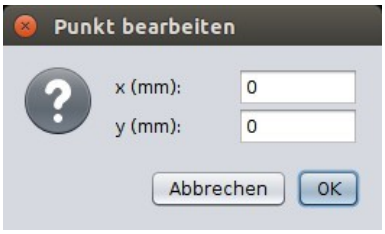


Figure 13: Dialoge "Edit point"

Deleting objects

Objects (stations, nodes, paths) can be deleted by selecting the *Delete objects* tool and then left clicking the object.

Clearing the roadmap

Select the *Clear all* tool to delete all the objects of the roadmap.

Undoing and recovering changes

Select *Undo* to go back to the previous step.

Select *Redo* to recover changes.

Completing the roadmap creation process

After the roadmap has been created it should look similar to the example below.

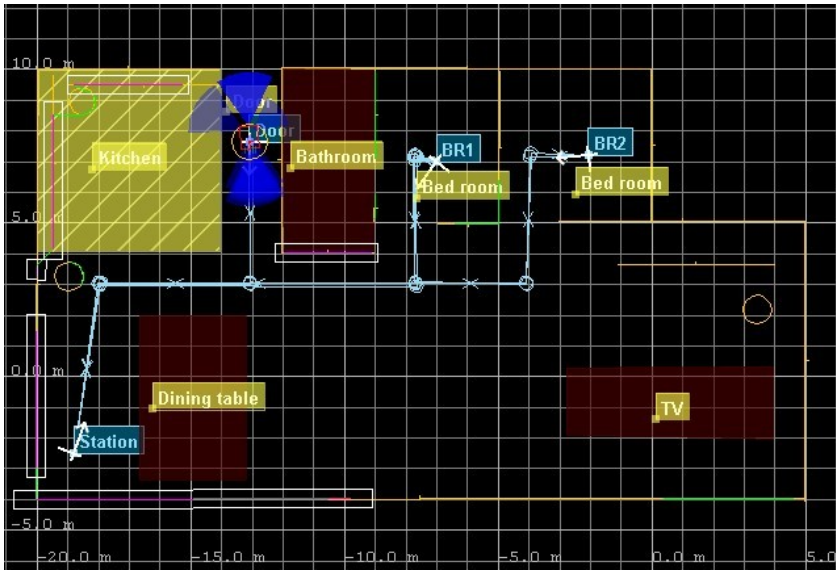


Figure 14: An example of a roadmap in an apartment.

The roadmap should be saved to the local hard disk by using the *Save* button. The first time the roadmap is saved a dialogue appears, asking for a file name and the location you want to save the map to.

Use the *Save as* button if you want to save your work in another file and continue working on that file.

Do not forget to upload the roadmap to the platform before exiting the roadmap editor. Use the *Save on platform* button and the roadmap will be sent to the platform.

Use the *Close* button at the bottom of the toolbar to close the roadmap editor. If you can not see it, scroll down using the scrollbar at the right side of the toolbar to make the button visible. If you have not saved the latest changes to the map, you will be asked whether you want to save the map to a file and / or upload it to the platform.

6 Driving the mobile platform

The platform's movement can be controlled in different modes which are explained in the following sections. To select or change the motion mode, use the drop-down menu in the main toolbar.

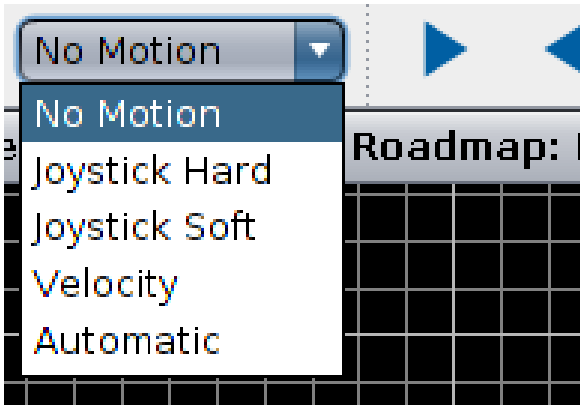


Figure 15: Mode control toolbar

6.1 The hardware joystick

To move the platform with a hardware joystick, connect a joystick to the platform. To connect a wireless joystick put the batteries in and press any button on the joystick. Joysticks with a cable connection should be plugged into a USB port of the platform before the robot is turned on. They might not be properly detected by the operating system and the control software if they are plugged in too late.

Select *Joystick (Hardware)* from the drop-down menu in the main toolbar. Press *Start* ▶ to activate this mode. With the *Stop* ■ button, this mode is aborted.

You can also press button *1* or *X* on the joystick to activate and deactivate the hardware joystick mode.

6.2 The software joystick

To move the platform with a software joystick, select the mode *Joystick (Software)* from the list. After pressing the *Start* button ▶ a small window will appear in the bottom right corner of the main window.

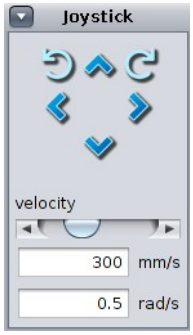


Figure 16: The software joystick window

The dark blue buttons make the platform move forward, backward and sideways (for omnidirectional robots). The slider and the two text fields allow to enter the transversal

velocity of the platform in mm/s and the radial speed in rad/s. Click *Stop* ■ on the main toolbar to abort the mode.

Due to the transmission delay between the GUI on a remote control PC and the mobile robot a delay between any command given in the GUI and its execution will occur. This might be critical when moving the robot in very narrow areas or when stopping a motion close to an obstacle.

6.3 The automatic mode

By setting a target location and using the *Automatic* mode, the robot will drive to the target autonomously. In this mode, there are two alternatives for a robot to reach the target location: by taking the shortest route or by using a roadmap defined by the user.

Free roaming

The platform can be told to drive to a specific point in the map. This can be done by selecting the *Set target frame* tool from the main toolbar.

Select *Set target frame* and click at the location where you want the platform to drive to. Hold the right mouse button and drag around the target frame to change its orientation. Select *Automatic* from the list and click *Start* ▶ to make the platform drive to the specified location.

Moving along a roadmap

You can also specify a station that is part of a roadmap as target position. The platform will use the current roadmap to find a way to the target.

Select *Set target station* and click on a station of the roadmap to mark it as destination. A red flag will appear on the station. Select *Automatic* from the drop-down list and press the *Start* ▶ button. The platform will find a way to the target station by following the roadmap.

If the distance between obstacles and the destination is smaller than the platform's collision radius the robot may report *Collision on target*.

6.4 Programming the robot

You can also program the robot to execute various predefined actions, depending on the system’s hardware and the environment it is intended to be used in.

To control the robot by a program or script please refer to the “PlatformCtrl-Programmer's Guide” or contact Neobotix.

7 Creating application with the Taskhandler

7.1 Introduction

Direct access to the robot control software PlatformCtrl is possible at any time. This allows the user to create very complex or time critical applications. The PlatformCtrl API is described in detail in the “PlatformCtrl – Programmer's Guide”. Please contact Neobotix if you have any questions.

For most applications, though, the Neobotix Taskhandler will be a powerful and convenient alternative to using the API. The Taskhandler is a separate program on the mobile robot that enables it to execute complete assignments autonomously.

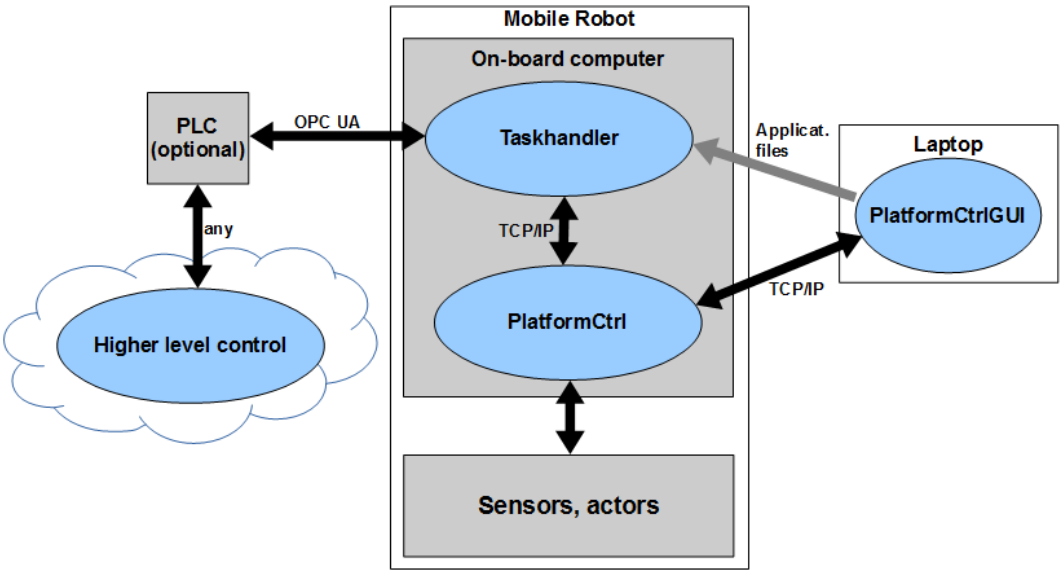


Figure 17: System diagram of the Taskhandler

Subtasks

Subtasks are the generic basic functions that the robot is capable of. They include, among others, moving to destinations autonomously, switching relays and digital outputs, waiting for an input and waiting for a certain time.

Tasks

Any number of subtasks can be combined to create a task. The task “Fetch parts from machine 1” could, for example, be made up of the following subtasks:

1. Subtask: Move to station “Aisle1”
2. Subtask: Activate warning light
3. Subtask: Move to station “Machine1”
4. Subtask: Pick up load with the load handling device
5. Subtask: Move to station “Aisle1”
6. Subtask: Deactivate warning light

Sequences

A set of tasks can then be combined to create complete assignments. This way tasks can be used in different sequences and do not need to be programmed over and over again for every sequence. The task described above could be part of a sequence “Store output

of machine 1” as follows:

1. Task: “Drive from storage to production hall”
2. Task: “Fetch parts from machine 1”
3. Task: “Drive from production hall to storage”
4. Task: “Hand load to storage worker”

Operating modes

The Taskhandler supports two modes of operation to meet different applications and requirements.

- In mode **Automatic** one of the predefined sequences will be executed and automatically restarted when finished. This allows the user to automate applications without having a higher level control structure like an ERP system and also without complex communication between the mobile robot and other machinery. This mode is the default setting at delivery.
  After each task of the sequence the robot can check for commands that where received via the OPC UA interface. If a command is pending the robot will switch to remote mode and execute the command.
- In **Remote** mode the robot will only wait for and execute commands received via the OPC UA interface. After executing all commands the robot can be switched back to automatic mode if necessary.

7.2 Configuring the Taskhandler

Use the GUI's main menu items *Window* → *Taskhandler* to open the configuration window of the Taskhandler.

Figure 18: Taskhandler configuration window

1: Taskhandler main menu

2: Switching between task and sequence configuration

3: List of existing tasks respectively sequences

4: List of the elements in the currently selected task or sequence

Use the *File* menu (1) to create new application files and to save and load them. These files contain all information necessary to operate the Taskhandler.

Parameters of the Taskhandler

The menu item *Edit* → *Parameters* opens a dialogue in which the application's properties can be specified. The tab *Taskhandler* contains the properties of the Taskhandler itself.

On Error: When set to *Abort* the robot will immediately stop and cancel the execution of its current command if an error occurs. *Retry* will make the robot execute the subtask that caused the error once more after the error has been reset. If the subtask can be finished successfully the execution will continue as normal. Further information on error handling

can be found in the corresponding chapter in this manual.

In case *Expert mode* is activated further configuration is available:

Operating mode: Changes the robot's behaviour from continuous execution of the default sequence (*Automatic*) to waiting for and executing only commands received via OPC UA (*Remote*). The operating mode can also be changed at runtime via OPC UA.

OPC UA visible: Activates the OPC UA interface.

PlatformCtrl IP: The IP address that the Taskhandler uses to connect to PlatformCtrl.

Taskhandler IP: The IP address that any OPC UA client may use to connect to the Taskhandler.

Taskhandler port: The port for connecting to the Taskhandler.

Alternative parameter sets

The tabs *Parameter set X* can be used to create and configure several alternative parameter sets for certain situations.

The robot's default parameters, which can be configured via the GUI's main menu item *Window → Platform Parameters*, are meant to ensure a safe operation of the mobile robot under normal conditions. This means that the default settings of the motion parameters and the collision avoidance will keep the robot from moving too close to people or obstacles.

In order to, for example, pick up cargo, though, the robot has to move much closer to a machine or conveyor than would normally be possible with the default settings. To allow this, alternative parameter sets can be designed which will only be activated right before the final approach to a workstation (subtask type 12).

Faulty collision avoidance parameters may lead to dangerous situations causing damages or injuries.

Only the operator will be responsible for the safe operation of the mobile robot. Neobotix cannot be held responsible in any way for injuries or damages which are caused by unsafe platform parameters.

Manipulating the platform parameters heavily affects the control software and requires solid understanding of both the mobile robot and the application. Please do not hesitate to contact Neobotix if you have any questions or the robot moves strangely.

Use the buttons “+” and “-” to add or delete parameter sets. Each set contains the parameters that are relevant for collision avoidance. You must enter a value for every parameter before the application file can be saved and used.

After each reboot PlatformCtrl will set the parameters to the default values that are saved in several ini-files on the robot. Changing the parameters via the GUI will not change the default values in the ini-files. Please contact Neobotix if permanent changes to the platform parameters are necessary.

7.3 Creating tasks

Adding, deleting and renaming tasks

Select the *Tasks* tab (2 in figure18) in order to manage the tasks of your application. The left side of the window will list all existing tasks.

Use the “+” button to add a new, empty task to the list. Left click on any task of the list to highlight it and show its subtask structure in the right side of the window.

Use the “-” button to delete the currently highlighted task.

Double-click a task to open a dialogue where the task's name and ID can be changed. The task IDs have to be unique within the application to allow the Taskhandler to identify

and activate them correctly.

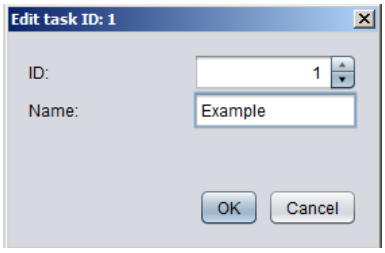
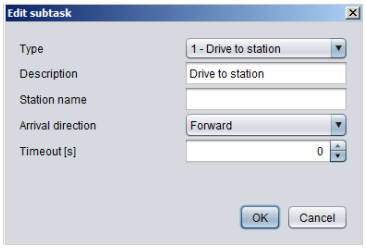


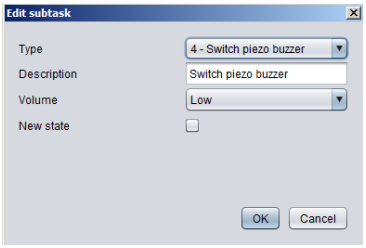
Figure 19: Task configuration dialogue

Adding subtasks

Select a task to show its current subtask structure in the right side of the window. The button “+” in that area adds a new subtask with default values. Left-clicking a subtask selects it for modification. It can then be moved to a new position within the structure by clicking the buttons “/\” and “V”. Click the “-” button to delete the selected subtask.

Double-clicking a subtask opens another dialogue for modifying its properties.





Figures 20 and 21: Different subtask properties

A subtask's type specifies which basic function of the mobile robot will be used. Different types require different parameters and the dialogue will change accordingly. The *Description* field is not read by the Taskhandler and can be used for any comment or description.

The following subtask types are available:

Type 1 – Move to station

The robot autonomously moves to a station saved in its current roadmap. The default motion parameters and collision avoidance settings will be used. Please use Type 12 (Move to work station) if another set of parameters has to be applied.

Planning time specifies the time that the robot may spend with searching for a route before it actually starts moving. If the robot starts from a populated area and might often be blocked by passers-by, this time should be increased.

If the time is set to 0 the robot will search for a route only once.

Parameter	Description	Range
1	Station name	Any station name
2	Direction (forward / backwards)	0 / 1
3	Maximum planning time in seconds	0 .. 10000

Type 2 – Software joystick

The robot can turn with a hard coded speed or move in X or Y direction. This can be helpful when docking to a charging station or approaching a work station.

In software joystick mode the collision avoidance settings are disabled!

Parameter	Description	Range
1	Direction (X = 1, Y = 2, α = 3)	1 / 2 / 3
2	Speed in mm/s or rad/s	1 - 150 0.01 – 2
3	Distance in mm or angle in radiant	10 – 10000 0.1 – 6.5

Type 3 – Switch relay on RelayBoard

This subtask type changes the state of one of the auxiliary relays on the RelayBoard.

Parameter	Description	Range
1	Relay number (charge relay = 0)	0 / 1 / 2 / 3 / 4
2	New state	0 / 1

Type 4 – Switch piezo buzzer

This subtask type activates a piezo buzzer that can be connected to the RelayBoard. It can be active for a maximum time of ten seconds. If the new state is to be 0 (off), parameter 1 will be ignored and the buzzer is always switched off.

Parameter	Description	Range
1	Volume (normal = 0, high = 1)	0 / 1
2	New state	0 / 1

Type 5 – Switch output of IOBoard

If the robot features an IOBoard, this subtask can be used to switch the board's outputs.

Further information on the IOBoard can be found in the board's operating manual.

Parameter	Description	Range
1	Output ID	0 - 16
2	New state	0 / 1

Type 6 – Wait for IOBoard input

If the robot features an IOBoard, this subtask can be used to wait until one of its inputs changes to the desired state.

Further information on the IOBoard can be found in the board's operating manual.

Parameter	Description	Range
1	Input ID	0 - 16
2	Desired state	0 / 1
3	Maximum waiting time	0 - 10000

Type 7 – Use load handling device

The optional load handling device can be used by sending commands to its integrated LAMBoard. Please mind that there is no check whether the LAMBoard is actually available or if the command ID is valid.

Further information on the LAMBoard can be found in the board's operating manual.

Parameter	Description	Range
1	ID of the LAMBoard sequence to execute	0 - 20

Type 8 – Wait

The robot waits for the specified time.

Parameter	Description	Range
1	Waiting time in seconds	0 - 10000

Type 9 – Wait for button

The robot waits for the specified button to be pressed. The available triggers are the four default buttons Info, Start, Stop and Home as well as the brake release button. The 24V auxiliary input on the RelayBoard can also be used as trigger.

Parameter	Description	Range
1	Button	Info = 5 / Start = 2 / Stop = 3 / Home = 1 / Brake = 4 / Input = 6
2	Maximum wait time in seconds	0 - 100

Type 10 – Configure parameter

This type allows the user to change a single platform parameter. The new value will be used until the next restart of the robot.

Parameter	Description	Range
1	Parameter name (see PlatformCtrl – Programmer's Guide)	String
2	New value	String / Integer / Bool

Type 11 – Play audio file

This subtask type allows the robot to play any audio file (WAV or OGG) on the robot's hard drive. The file will be searched for in folder *home/neobotix/bin/Platform/sound*.

The wait time parameter specifies the time until the Taskhandler switches to the next subtask. If the playback time is longer, it will be stopped when the time has run out. If it is shorter, the robot will continue waiting after playback has finished.

Setting the time to 0 will play the file in a separate thread in full length while the robot immediately steps to the next subtask.

Parameter	Description	Range
1	File name including file extension	String
2	Maximum time in seconds	0 - 10000

Type 12 – Move to work station

This type works similar to type 1 (move to station) but will activate one of the alternative parameter sets first. This allows the robot to move closer to obstacles than would be possible with the default motion and collision avoidance parameters.

Parameter	Description	Range
1	Station name	Any station name
2	Direction (forward / backwards)	0 / 1
3	Maximum planning time in seconds	0 - 10000
4	Parameter set	1 - 100

7.4 Creating sequences

The *Sequences* tab is very similar to the *Tasks* tab.

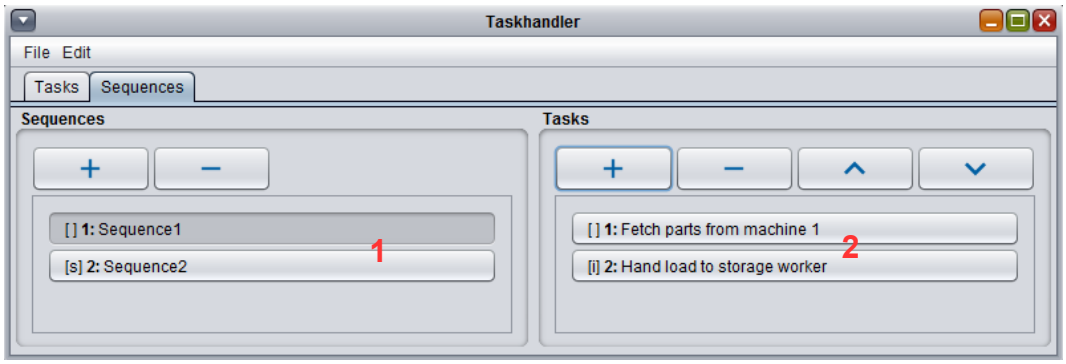


Figure 22: The Sequences tab

1: List of existing sequences *2: Build-up of the selected sequence*

The left side (1) lists all existing sequences while the right side (2) shows the build-up of the currently selected sequence.

New sequences can be added with the “+” button on the left side and can selected sequences can be deleted with the “-” button. A double-click on any sequence opens a dialogue for changing the ID and description of the sequence. The sequence IDs must be unique within each application but may overlap with the task IDs.

The sequence configuration dialogue also offer the possibility to set the sequence as default. In the sequence list (1) the default sequence is marked with an *[s]* in front of its name.

When a sequence is selected in the left side of the window it can be filled with tasks by clicking the “+” button on the right side. Surplus tasks can be deleted with the “-” button. The buttons “/\” and “V” move a task to a new position within the sequence.

After adding a new task to a sequence a configuration dialogue appears. Use the combo box to select the task that shall be executed at this step of the sequence. The check box *Interruptible* specifies, whether the robot should check for pending OPC UA commands after finishing this task. If a command has been received the Taskhandler will switch to remote mode and immediately execute the command.

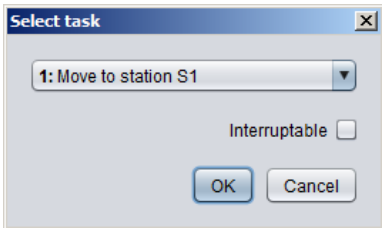


Figure 23: Sequence element configuration dialogue

Tasks after which the sequence can be interrupted by OPC UA commands are indicated by a [I].

7.5 Activating the Taskhandler

The robot can work autonomously once a map, a roadmap and an application file have been created and saved. Maps and roadmaps can be downloaded to the robot by using the buttons in the map and roadmap editor toolboxes. Application files need to be copied to the robot's hard drive by hand.

Please copy the application files that was created with the GUI into folder *<home>/neobotix/bin/Platform/Applications* and change the file name to *Application.xml*. The Taskhandler will use this file automatically after the next restart of the robot.

7.6 Error handling

The Taskhandler does not support decisions or selecting different actions based on input states or other conditions. Every deviation from the planned process will interrupt the execution of the current task and make the robot wait for user input. Only the error code indicated via OPC UA will change to reflect the cause of the error.

Different recovery procedures are available, depending on the operating mode and the selected error handling.

Automatic – Retry

The easiest way to use the mobile robot is in mode *Automatic* with setting *Retry* in case of an error. If the robot stops, please identify the cause of the problem and remove it.

The robot can immediately continue working once the cause of the error has been cleared. After pressing the *Start* button on the robot (please refer to the operating manual of your robot) or the equivalent OPC UA signal the robot will repeat the subtask that caused the error. It will, for example, continue moving to its destination or try to hand over cargo that was stuck before.

The operator has to make sure that this repetition of the interrupted subtask will not lead to another error. The load on the load handling device must therefore be moved back to its former position and should not be moved on by hand. Otherwise the subtask will fail again because the load that is to be handled is no longer available.

The robot will then continue its default sequence indefinitely.

Automatic – Abort

The setting *Abort* will make the robot stop working on the default sequence altogether.

Pressing the **Start button** will then start the default sequence from the beginning again. The robot and all the machines it interacts with will have to first be moved and / or reset to their starting positions and states.

If the error is acknowledged via **OPC UA** the robot will switch to remote operation and wait for commands. It can then be sent to its starting position via OPC UA, for example. Commanding it to change back to automatic mode will start the default sequence from the beginning.

Remote – Retry

This combination is equivalent to *Automatic – Retry* with the one difference that the robot will only finish its current task or sequence and not restart automatically. Instead it will wait for commands via OPC UA.

As before it is important to make sure that the repeated execution of the interrupted subtask will not cause an another error.

Remote – Abort

In this configuration the robot will simply stop the current task or sequence and wait for new commands.

Fatal errors

In case of fatal errors, like a permanent obstacle on a route or damage to a machine, it will be necessary to modify the robot's data (map, roadmap, application file). These modifications should only be done by a qualified technician.

8 Hardware monitoring

The GUI provides access to most of the mobile robot's electrical and electronic components. This includes status indicators for diagnosis as well as the possibility to change outputs manually.

The main menu item *Windows → Hardware* monitor opens a new window with several tabs. Each tab shows the state of one component. Depending on the robot's features some tabs may not be available.

Almost all features and information that are accessible via the GUI are also available vie the socket interface to PlatformCtrl and the Python modules. You can find further information in the "PlatformCtrl – Programmer's Guide" and the "Python – Programmer's Guide".

8.1 RelayBoard

The RelayBoard is the main control board of all mobile robots from Neobotix. It handles all basic features and also provides some auxiliary inputs and outputs as well as status information and measurements.

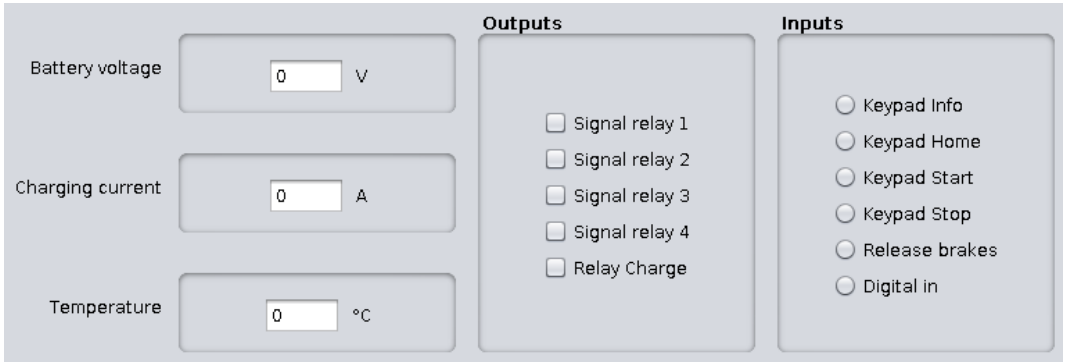


Figure 24: The hardware monitor for the RelayBoard

Battery voltage: The current battery voltage is shown here. More information on the voltage ranges of the different models can be found in the according operating manuals.

Charging current: This indicator shows the charging current from the automatic charging station.

If the robot was connected to an external battery charger by hand, then the charging current is not monitored by the RelayBoard and will not be shown. In this case the only indication of the charging process will be the increased battery voltage.

Temperature: This is the current temperature at the RelayBoard. Other parts of the robots, especially the on-board computer, may be warmer.

Outputs: Beside the relays for the robot's basic features (main power relay, emergency stop circuits etc.) the RelayBoard also offers some auxiliary relays which can be switched manually.
If the robot was set up for automatic charging then the charging process can be started by activating *Relay Charge* after the robot has docked to the charging station.

8.2 IOBoard

The Neobotix IOBoard offers a number of digital inputs and outputs as well as analogue inputs. These can be used to connect additional devices, sensors and actors to the robot.

Please see the corresponding operating manual for further information.

Digital outputs: The digital outputs of the IOBoard, both optocouplers and relays, can be switched here.

Digital inputs: These indicators reflect the state of all digital inputs of the IOBoard.

Analogue inputs: The voltages at the IOBoard's analogue inputs are shown as percentage of the maximum admissible input voltage.

8.3 USBoard

The USBoard can read the measurements of up to 16 ultrasonic sensors. These values can be sent to PlatformCtrl for collision avoidance. The board also offers four analogue inputs for customised use.

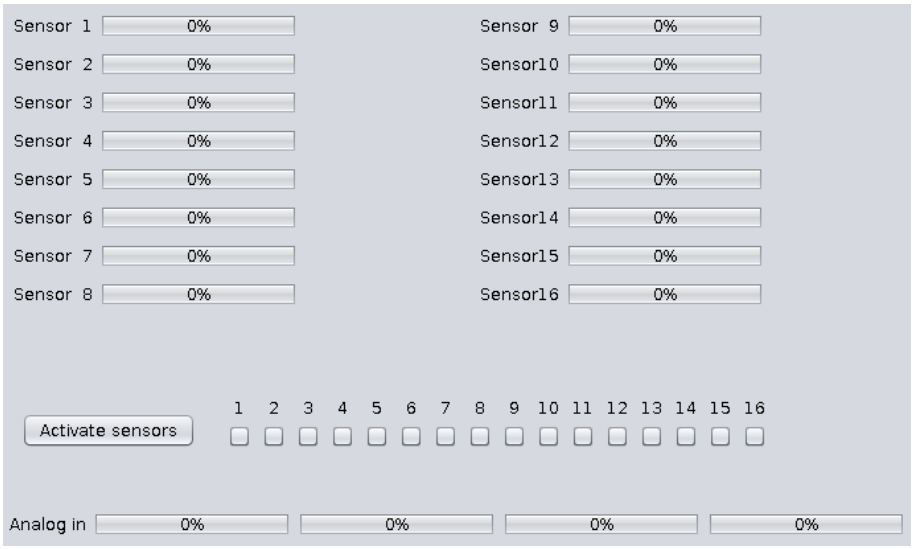


Figure 25: The hardware monitor for the USBoard

Sensor 1 .. Sensor 16: These bars show the current distance measurements of the available ultrasonic sensors.

Activate sensors: To select which sensors are to be active set the control boxes accordingly and click this button.

Analog in: The voltages at the USBoard's analogue inputs are shown as percentage of the maximum admissible input voltage.

8.4 FlexiSoft

This tab is only available if the robot is equipped with a FlexiSoft safety controller. Depending on the robot's safety features some elements may be unavailable or without function.

The FlexiSoft's safety functions must be set up by the responsible operator according to the results of the assessment of risks of each individual application. The functions and features described in this document represent the default configuration which may not fulfil the safety requirements of your application.

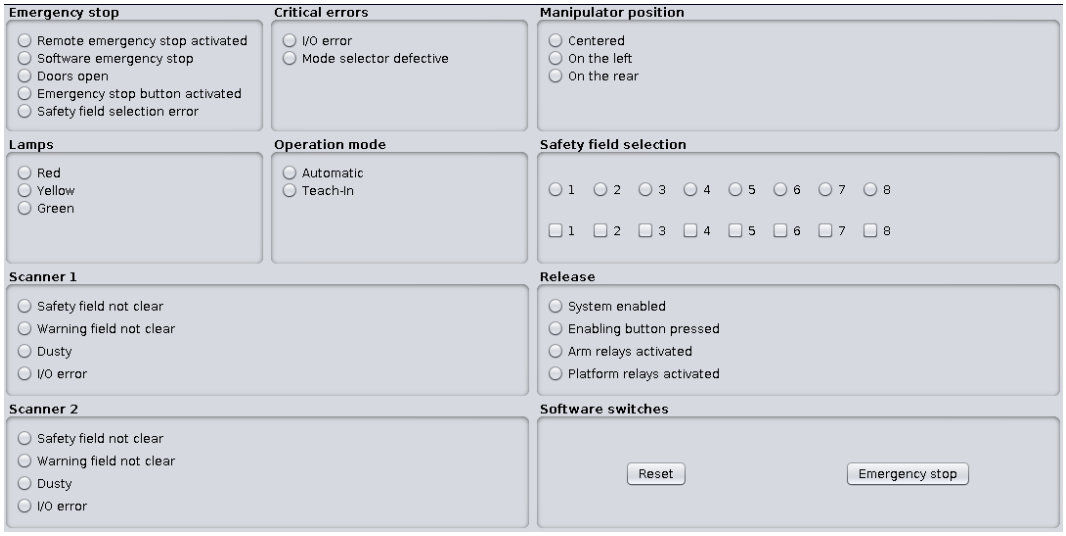


Figure 26: The hardware monitor for the FlexiSoft

Emergency stop: This field indicates non-critical errors and conditions which can cause an emergency stop. After these errors have been resolved the safety system can be reset.

Remote emergency stop activated: The radio connection to the transmitter was lost and has to be re-established.

Software emergency stop: The safety controller received an emergency stop command from the GUI, a Python script or some other source.

The software emergency stop is not an approved safety system and only intended as addition for situations when a normal emergency stop device is not in reach. In case of danger ALWAYS ALSO engage an emergency stop by pushing a mechanical emergency stop button!

Doors open: If the robot has door or hatches that are monitored by safety switches their current state is indicated here.

Emergency stop button activated: As soon as any of the robot's emergency stop buttons is pressed the robot will go to emergency stop and the cause will be reported here.

Safety field selection error: In some robots the safety fields of the laser scanners can be activated by the control software. This error indicates that the field selection was technically faulty, for example, no or a wrong code was transmitted to the FlexiSoft.

Neither the FlexiSoft nor PlatformCtrlGUI check whether the selected safety field actually guarantees the required level of safety in the current situation. Only the higher-level control software respectively the operator are responsible for the correct operation of the safety system.

Critical errors: This group contains errors that are not expected to occur in normal operation and must immediately be analysed and fixed by a technician.

I/O error: The FlexiSoft controller monitors all its inputs and outputs to detect electrical problems. This error indicates that one of the devices connected to the FlexiSoft is wired incorrectly of defect. The FlexiSoft-Designer software from Sick can be used to check the condition of individual I/Os.

Mode selector defective: The mode selector switch must only create one of a given set of signals. In case of a defect an incorrect signal will be created, causing this error.

Arm position: With some robotic arms it may be necessary to monitor the arm's pose relative to the mobile platform. This area shows the states of all arm monitoring switches.

Lights: The current state of the optional signal column is shown here.

Operating mode: Shows the robot's current operating mode.

Safety field selection: Using a FlexiSoft controller allows to create several different safety and warning fields for the laser scanners. The control software can then dynamically select one field set at a time by sending the corresponding bit pattern to the FlexiSoft. These bit patterns can be chosen and assigned by the operator. By default the FlexiSoft expects patterns of eight bits size.

One possible type of bit pattern is 1-in-8. This mode is used at delivery and requires the control software to set exactly one of the bits to *1 / True* and all others to *0 / False*.

Neither the FlexiSoft nor PlatformCtrlGUI check whether the selected safety field actually guarantees the required level of safety in the current situation. Only the higher-level control software respectively the operator are responsible for the correct operation of the safety system.

Scanner: This group shows the status of up to four laser scanners.

Safety / Warning field not clear: An obstacle was detected in the currently activated safety or warning field. The robot will resume its normal operation after the obstacle was removed from the safety field, given that no other error occurs.

Dusty: The scanner's optics cover should be cleaned in order to ensure proper operation.

I/O error: An error was detected at either the scanner's electrical inputs or in the data input received from the FlexiSoft.

Operation: This area indicates the proper operation of the safety system.

System enabled: All safety functions are clear, the last emergency stop has been reset and the robot is operational. In Teach-In mode the enabling button must be pressed to move the robot.

Enabling button pressed: In order to move the robot in Teach-In mode the safety system must first be reset by either turning the key switch or by pressing the reset button of the remote emergency stop system. The motors will then only be powered as long as the enabling button at the manual control unit is pressed.

Arm relays activated: The motors of the robotic arm are powered and the arm can be moved.

Platform relays activated: The motors of the mobile platform are powered and the robot can drive.

Software switches: If included in the FlexiSoft program, these buttons allow the operator to remotely reset the safety system or to set the robot to emergency stop.

Reset: This button is equivalent to the key switch and may reset an emergency stop remotely as long as the safety system does not detect any error.

Emergency stop: In case of an emergency the robot can also be stopped by software. This signal will remain in the FlexiSoft even if the connection to the GUI is lost. It must be reset via the GUI, by some other software or by restarting the robot.

Please note that this emergency stop signal may be cleared by any other software module and not necessarily by the GUI that originally set it. After resetting the signal the robot can be enable as usual, as long as no other errors occur.

The software emergency stop is not an approved safety system and only intended as addition for situations when a normal emergency stop device is not in reach. In case of danger ALWAYS ALSO engage an emergency stop by pushing a mechanical emergency stop button!

9 Platform parameters

Most of the robot's parameters can be changed in order to adjust its features and behaviour.

In the main menu, select *Parameters → Platform Parameters* to open the panel. To configure a parameter, click on the parameter and a text field or a drop-down menu (for Boolean parameters) will appear at the bottom of the panel. Enter or choose the new value and click *set* to change it.

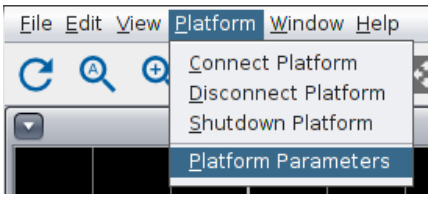


Figure 27: Platform Parameters drop-down menu.

Parameters are grouped into tabs. A brief explanation of each parameter in each tab is given in this chapter. For more in-depth information please refer to the “PlatformCtrl-Programmer's Guide” or contact Neobotix.

Surveillance

Parameter	Description	Value
FlashlightEnabled	Turning the lights on or off	true: on false: off
PIRSensorsEnabled	Enable PIR Sensor.	true: enable false: disable
RadarSensorsEnabled	Enable radar sensor.	true: enable false: disable
SirenEnabled	Enable sirens	true: enable false: disable
SurveillanceModeEnabled	Enable the surveillance mode	true: enable false: disable

Tracking & Detection

Parameter	Description	Value
ChargeStationDetectEnabled	Enable function to detect a charge station.	true: enable false: disable
PersonDetectionEnabled	Enable function to detect a person.	true: enable false: disable
TrackingEnabled	Enable tracking of moved objects with the scanner.	true: enable false: disable

Ultrasonic

Configuration of the ultrasonic sensors. These values are used for safety purposes and therefore should not be changed.

Each ultrasonic sensor has four parameters:

- X-coordinate with respect to the robot frame in [mm]

- Y-coordinate with respect to the robot frame in [mm]
- Z-coordinate with respect to the robot frame in [mm]
- Angle with respect to x-axis in [degree]

Mapping

Parameter	Description	Value
MinLineLenAux	Minimum length of auxiliary lines [mm]: lines extracted from scan.	-
MinLineLenMap	Minimum length of mapped lines [mm]	-

Motion Control

These parameters affect the robot's movements. Note that the parameters “PathFollowParam1” and “PathFollowParam2” are correction values and should not be changed.

Parameter	Description	Value
PathFollowParam1	First parameter for path following function (correction value)	-
PathFollowParam2	Second parameter for path following function (correction value)	-
PositionStabilizationEnabled	Enable posture stabilization at the end of a move command.	true: enable false: disable
OmniMoveEnabled	Enables sidewards movement for omnidirectional platforms. The platform keeps its orientation while moving. Parameter has no effect for differential platforms.	true: enable false: disable

Planning

Parameter	Description	Value
Planning Enabled	Enable planning in automatic mode	true: enable false: disable
SimuOnlineEnabledPostDis	Enable online simulation	true: enable false: disable
ViewPlanObstEnabled	View the obstacles used for planning (red lines in the platform GUI).	true: enable false: disable

Collision

Parameter	Description	Value
CollCheckEnabled	Enable the collision check with all sensors	true: enable false: disable
CollLaserMinDist	Minimum distance to the obstacle [mm]	-

Geometry

This tab displays the dimensions of the platform that are used for collision avoidance and path planning. These values are hardware dependent and should not be changed.

Parameter	Description	Value

DistPltfCenterToWheelAxis	Distance of the centre of the robot to its wheel axis. [mm]	-
DistWheels	Distance between wheels [mm]	-
PltfLength	Length of the robot [mm]	-
RadiusWheel	Wheel radius [mm]	-

Interpolation

Parameter	Description	Value
EndTurnEnabled	Allows the robot to turn on the target position to face the target direction. Otherwise the robot will drive a curve.	true: enable false: disable
IpoRotvel	Rotational velocity of the interpolated frame in [rad/s].	Default: 0.6 rad/s
IpoTransVel	Translational velocity of the interpolated frame [mm/s]. If there is no obstacle and a path is straight, the robot will accelerate up to the specified velocity.	Default: 1000 mm/s
StartTurnEnabled	In automatic mode with disabled planning the path is interpolated with a Bezier-curve. Enable this function to make the robot turn first into the direction of the path before driving to the target.	true: enable false: disable

Localization

Parameter	Description	Value
GlobalLocalization	Makes the robot use the whole map to localize its possible position depending on the scan.	true: enable false: disable
LineExtractMaxCov	Maximum uncertainty of extracted lines	
LocalLocalization	Enable the robot to correct its possible position using the laser scanner.	true: enable false: disable
MappingMode	Not used at the moment.	

10 Legal notes

Version information

This document has been translated and is not the original. Please refer to the German version in case of uncertainties or questions.

Liability

Every care has been taken in the preparation of this manual which represents the state of technology at the time of its composing. However, inaccuracies or omissions might occur. Please inform Neobotix in case you notice any.

The GPS GmbH, Neobotix, cannot be held responsible for any technical or typographical errors and reserves the right to make changes to the product and manual without prior notice.

Neobotix makes no warranty of any kind with regard to the material contained within this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Neither the GPS GmbH nor Neobotix shall be liable or responsible for incidental or consequential damages in connection with the improper use of one or more of the products described in this manual.

Declaration of conformity

This product fulfils all relevant directives of the European Union. For further information please contact Neobotix.

Downloads and further information

Additional information, data sheets and documentations, also for the other products of Neobotix, can be found on our homepage www.neobotix.de/downloads.

Imprint

Neobotix GmbH

Weipertstr. 8-10, D-74076 Heilbronn

www.neobotix-roboter.de

Contact: Dipl.-Ing. Till May

Tel.: (+49) 7131 / 7669 - 300

e-mail: may@neobotix.de

PltfCtrlGUI-OperatingManual.odm, edited 23. August 2016 in Heilbronn, Germany