

NeuroCheck RoboDirector



Copyright

Copyright © NeuroCheck GmbH
All rights reserved.
Version 2.6.0
Neckarstraße 76-1, 71686 Remseck, Germany

Phone: +49 (0) 7146 - 89 56-0
Fax: +49 (0) 7146 - 89 56-29
E-Mail: info@neurocheck.com
Web: www.neurocheck.com

Table of Contents

General Information	3
Introduction	3
Installation	4
Configuration	5
RoboDirector	6
Positions	6
Routes	7
Groups	8
NeuroCheck Remote Control	9
Sanity Check	13
Settings	14
Support Contact	15
Support Services	15

Introduction

RoboDirector allows **NeuroCheck** to communicate with the collaborative robots from Universal Robots. It allows teaching positions and routes and provides various other ways of controlling the robot. To setup an automated inspection system, the tool can be controlled from **NeuroCheck** with the Inter-Process Communication Driver.

The following topics describe the set up and basic usage of **RoboDirector**.

Warning

This software is used to send commands to a robot. Please ensure that safety standards for the robot are followed so it can not harm users during operation.

Installation

Copy Files

Copy the contents of the `Binaries\` folder to the target installation directory. It is recommended not to use `C:\Program Files\` as Windows has special permissions for that directory and will prevent the creation of files there (e.g. logs).

Configure

The default configuration is sufficient for basic use as is. For the full description of the `appsettings.json` configuration file please refer to [configuration](#).

NeuroCheck Communication

Install the inter-process communication Driver `InterComm` for **NeuroCheck**. See its help for an installation guide. The **RoboDirector** side settings are described in [settings](#).

Robot Communication

The communication to the robot works via TCP/IP. Make sure the robot controller is connected to your PC, your network configuration allows communication and you know its IP or hostname.

Configuration

The `appsettings.json` file contained in the installation directory contains the information needed to configure the **RoboDirector** server. It is written in *json* syntax, but already formatted to contain all relevant sections.

- [ConnectionStrings](#)
- [urls](#)
- [AuthenticationSettings](#)
- [Logging](#)

Note

The *json* format expects backslashes (\) to be escaped (\\). This can be relevant for paths or connection strings.

ConnectionStrings

```
"ConnectionStrings": {
  "DefaultConnection": "DataSource=users.db",
  "Provider": "sqlite"
}
```

This section contains the `DefaultConnection` key. This only needs to be changed when setup together with an **Inspector** system. Please refer to the **Inspector** help for more information.

urls

```
"urls": "http://localhost:6200/"
```

This section contains the urls on which the server will listen to incoming requests. It defaults to `http://localhost:6200/`. Currently only the *http* scheme is supported.

AuthenticationSettings

```
"AuthenticationSettings": {
  "WhitelistIPs": [
    "127.0.0.1",
    ":::1"
  ],
  "LoginExpiration" : 30
}
```

This section contains settings specific to authentication. It is defined by the following keys:

- `WhitelistIPs` list the IP's that are allowed to access the UI in case an unlock signal is received. An unlock signal can be send by **NeuroCheck** intercom to override the **RoboDirector** authentication. This can be used to unlock with authentication hardware communicating to **NeuroCheck**. It defaults to the IP V4 and IP V6 loopback IP's.
- `LoginExpiration` sets the time in minutes until a logged in used is logged out.

Logging

```
"Logging": {
  "File": {
    "LogLevel": {
      "Default": "Error",
      "NcRoboInspector.Server": "Error",
      "NcRoboInspector.UI": "Error"
    }
  }
}
```

This section contains logging related configuration. The most relevant setting here is the `File` section, that contains the logging level for log events that are written to the log files in the *Logs* directory. The `Default` category log level specifies the log level of all components of the server. Possible log levels are *Critical*, *Error*, *Warning*, *Information*, *Debug*, *Trace* and *None*.

Positions

Positions are locations in space a robot can move to. All regular positions must reference a [safety position](#). Positions can be categorized in [groups](#) and added to a [route](#).

Initial position

The initial position is a special position of each robot. All routes start and end at the initial position.

Safety position

Safety positions are special positions which are meant to enable collision free movement of the robot. Each regular position must reference a safety position. The safety should be chosen such that the robot can move freely from a regular position to its safety, as well as from the safety position to the initial position. The referenced safety positions are automatically inserted into a [path](#) when adding a position to a route. They are also used in several "go to initial" [remote control](#) commands, to ensure the robot can safely return to its initial position from any position in a route.

Routes

A route is a collection of [positions](#) and their connecting paths which a robot moves over. A route always starts and ends at the [initial position](#).

Paths

A path connects two positions in a route. It can contain one or more intermediate positions which the robot will move over before reaching the next position. The speed and type of movement can be set for each individual path:

Parameter	Description
Movement type	Fast Linear
Speed	
Acceleration	
Blend radius	

If a new position is inserted into a route new paths connecting the position to the previous and following one are automatically created. The default movement parameters for these paths can be configured in the [settings](#).

Note

By default, a newly created path will include the [safety positions](#) of the previous and target position as intermediates. This is to ensure that the robot can move between the positions safely without any potential collisions. If this behaviour is undesired, the automatically created intermediates can be deleted or the *Shift* key can be held when inserting a new position.

Regions

Regions are only relevant if multiple robots are controlled by **RoboDirector**. They can be used to mark regions of exclusive access for robots working in close proximity. Regions are unique across all robots and can be defined over one or more positions in the routes of multiple robots. If one robots enters a region, **RoboDirector** will lock this region and prevent all other robots from entering the same region until the locking robot has left it.

Groups

Positions can be added to groups. They are a way to manage and categorize larger numbers of positions.

NeuroCheck Remote Control

RoboDirector can communicate with and be controlled by a **NeuroCheck** instance to implement a robot based inspection system. The protocol follows a similar schema as the **NeuroCheck** remote control itself.

Configuration

RoboDirector can be remote controlled by **NeuroCheck** using the inter-process communication driver. To setup the communication:

1. Go to the **NeuroCheck** device manager and add a intercom client fieldbus device.
2. Connect this client device to **RoboDirector** using the server name and port configured in the [settings](#). If the NeuroCheck instance does not run on the same machine, set the connection type to network and enter the IP of the computer **RoboDirector** runs on. In that case also ensure that your network setup allows the two devices to communicate.
3. Add a **NeuroCheck** data format converter for the created fieldbus device and import the `pinConfiguration.csv` which is part of the **RoboDirector** installation package.

Refer to the following section for details on the communication protocol.

Reading RoboDirector Status

Various status information can be read by **NeuroCheck** using the configured input pins and the usual check functions:

Pin name	Bit	Data Type	Description
RoboDirector running	0	Bool	RoboDirector is currently running and the connection between NeuroCheck and the tool could be established.
Robot connected	1	Bool	The connection with the robot was successful.
Robot ready	2	Bool	The robot is ready to receive commands. The status is false if the robot is not initialized, the robot is moving or the safety is triggered.
Robot active	3	Bool	The robot is currently moving.
Robot initialized	4	Bool	The robot went through the initialization routine and can be moved.
Robot at initial position	5	Bool	The robot is at the defined initial position and can start moving along the route.
Position reached	6	Bool	If the robot arrived at the next position in the route, this signal is set. The robot will wait till it gets the command to continue to the next position.
Safety OK	7	Bool	Operation is normal, no safety was triggered.
Safety reduced mode	8	Bool	Signals if the robot is in reduced mode (meaning it will move at a much lower speed).
Safety Error	9	Bool	If any of the following safety signals are triggered, this pin will be set. The execution on the robot will stop: • Protective Stop • Recovery • System Emergency Stop • Robot Emergency Stop • Violation • Fault
Safety Safeguard	10	Bool	Signals if the safeguard on the robot is triggered. The robot will resume operation if the safeguard signal is not set anymore.

Safe movement possible	13	Bool	Signals if RoboDirector knows the current position or traced the path to the last known route position and can initiate a safe return to the initial position. See command code 101, with parameters 2 or 4.																				
Route finished OK	14	Bool	True if the route was successfully finished and the robot arrived at the last position.																				
Route finished NOK	15	Bool	True if there was an error during processing of the route.																				
ID current position	16-31	Int	ID of the current position the robot is at.																				
ID next position	32-47	Int	ID of the next position the robot will move to.																				
ID current route	48-55	Int	ID of the current route that the robot moves on.																				
Mirrored command code	56-63	Int	For synchronization purposes, this pin will show the command code that the RoboDirector received, if the code was executed successfully.																				
Mirrored command parameter	64-95	Float	For synchronization purposes, this pin will show the command parameter that the RoboDirector received, if the code was executed successfully.																				
Safety mode code	96-103	Int	Read the current safety mode. Possible values are: <table><thead><tr><th>Value</th><th>Safety Mode</th></tr></thead><tbody><tr><td>1</td><td>Normal</td></tr><tr><td>2</td><td>Reduced</td></tr><tr><td>3</td><td>Protective Stop</td></tr><tr><td>4</td><td>Recovery</td></tr><tr><td>5</td><td>Safeguard Stop</td></tr><tr><td>6</td><td>System Emergency Stop</td></tr><tr><td>7</td><td>Robot Emergency Stop</td></tr><tr><td>8</td><td>Violation</td></tr><tr><td>9</td><td>Fault</td></tr></tbody></table>	Value	Safety Mode	1	Normal	2	Reduced	3	Protective Stop	4	Recovery	5	Safeguard Stop	6	System Emergency Stop	7	Robot Emergency Stop	8	Violation	9	Fault
Value	Safety Mode																						
1	Normal																						
2	Reduced																						
3	Protective Stop																						
4	Recovery																						
5	Safeguard Stop																						
6	System Emergency Stop																						
7	Robot Emergency Stop																						
8	Violation																						
9	Fault																						

Controlling RoboDirector

To execute a command, a integer command code is set on bits 8-15, optionally accompanied by a float command parameter on bits 16-47, depending on the command code. The command is executed by signalling on bit 0. Note that pins for these signals are also created when importing the `pinConfiguration.csv` mentioned above.

The following command codes (and potentially parameters) can be sent to **RoboDirector** to control the connected robot:

Command code	Command parameter	Description
10		Start the robot. This will execute the power on and brake release commands on the robot
20	0 = unlock 1 = lock	(Un)lock the teach panel of the robot
30	0 = unlock 1 = lock	(Un)lock the RoboDirector UI

40	1-100 meaning 1% to 100% of possible speed	Set relative Speed. The robot speed can be globally set to between 1% and 100% (also available on the teach panel) and this command lets you control this value. This relative speed will automatically be used when you start a movement with the robot, e.g. going directly to a position or when starting the movement along a route.
50	0-6 as the index of the digital output line	Set the general purpose digital output of the robot to active (high). The last output line is used for internal communication between the robot and RoboDirector , so it can't be set manually.
51	0-6 as the index of the digital output line	Set the general purpose digital output of the robot to inactive (low). The last output line is used for internal communication between the robot and RoboDirector , so it can't be set manually.
52	0-7 as the index of the digital output line	Set the configurable digital output of the robot to active (high).
53	0-7 as the index of the digital output line	Set the configurable digital output of the robot to inactive (low).
60	Offset X axis in mm	Sets the value of the offset in the X axis in mm. The offset will only be applied if you activate the offset in the settings in NcRoboDirector.
62	Offset Y axis in mm	Sets the value of the offset in the Y axis in mm. The offset will only be applied if you activate the offset in the settings in NcRoboDirector.
64	Offset Z axis in mm	Sets the value of the offset in the Z axis in mm. The offset will only be applied if you activate the offset in the settings in NcRoboDirector.
100	ID of the route	Start processing the route and move to the first position. The robot needs to be at the initial position and in ready state to successfully execute the command.

101	0 = The robot will directly move from its current position to the initial position 2 = The robot will move to the safety position of the last position it was at and then go to the initial position, alternatively the robot will use the traced return path to move back stepwise to the last known route position and then go over its safety to the initial position, if the movement between two route positions has been interrupted. This parameter works only if the last route position is known and the robot can use the traced path to return to it.. 4 = same as parameter 2, but if there is neither any information about the last position nor traced path data available, the robot will directly go from its current position to the initial position. You can think of this as a combination of executing with parameter 2, and if it didn't work, then parameter 0 is executed. If the last known robot position already was the initial position and the robot is still very close to it, then it will move directly, even if parameter 2 is used. The allowed proximity can only be modified via the parameter tolerance_joint_rad in the XML configuration file.	Move to the initial position. If a route is currently processed, it will be aborted.
102		Move to the next position in the route.
103		Abort processing of the route.
104	ID of the route position	Move to a specific route position. The robot will move in the route without stops till the target position. The target position must lie after the current position in the route. If the target position ID matches the current position ID, the robot stays at the position.
105	ID of the position	Move to a specific position safely. The robot moves outside of the route and follows the safe path from the current position to the target position. If the target position ID matches the current position ID, the robot stays at the position. If the target position is present in the currently processed route, the route processing can be continued from the target position.
106	ID of the route	Start processing the route. The robot needs to be at the initial position and in ready state to successfully execute the command. In contrast to command 100 the robot stays at the initial position and waits for next commands.

Sanity Check

If positions taught with one robot are used by a different one, i.e. when importing a configuration or exchanging the robot of a system, some positions may not be reachable with the new robot. This can happen especially with positions which require extreme joint angles to reach the desired tool center point coordinates. Slight variations in the joint limits or calibration of the robots can render those positions unreachable. To prevent interruptions during route execution **RoboDirector** provides the Sanity Check feature: It will determine if the desired tool space coordinates of all positions in the Sanity Check are reachable using the current joint calibration by calculating the inverse kinematic for each position.

The Sanity Check can be started for all positions, all positions of a group or all positions of a route from the respective views.

Note

Even when a position is reachable with a new robot, this does not necessarily mean that the actual physical location and orientation of the tool center point in space are exactly the same. Small mechanical variations in the installation of a new robot can result in noticeably different absolute poses of the tool. If your inspection relies on precise positioning, e.g. the location of certain image sections, make sure to check the correct behavior of your check routines after changing robots.

Settings

These are settings per robot that can be accessed via the dashboard.

Intercom

The intercom server name and port which are needed for the **NeuroCheck** connection can be configured here.

Movement

The defaults for acceleration and speed can be configured here. These are used when creating a path in the route editor

Movement in tool space coordinates can be enabled, otherwise movements are specified in joint space coordinates.

Connection

The robot connection settings together with a robot description can be set on the overview page of the user interface, outside of the robot dashboard.

Support Services

For technical support, please contact your local NeuroCheck partner or NeuroCheck GmbH:

Phone: +49 (0) 7146 - 89 56-40

E-Mail: support@neurocheck.com

Web: www.neurocheck.com