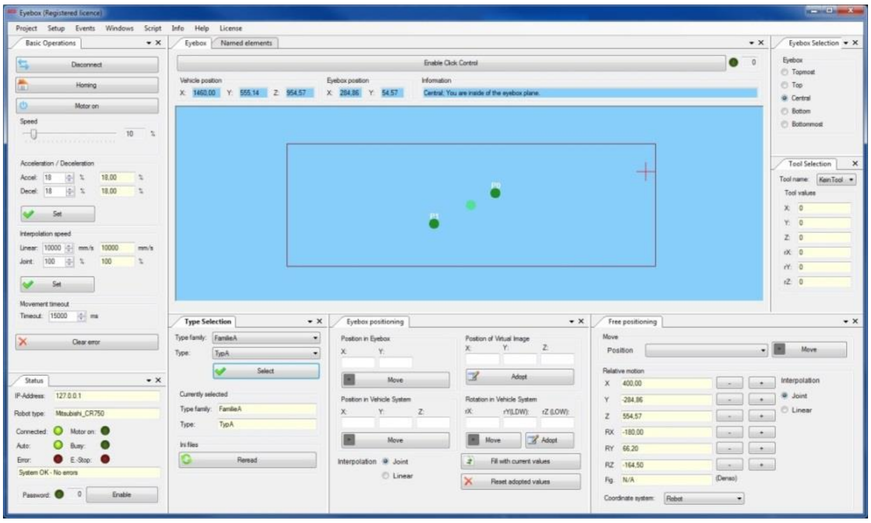
Manual

Toolmonitor Eyebox



Speed: 10 %

Acceleration / Deceleration
Accel: 18 % 18,00 %
Decel: 18 % 18,00 %

Interpolation speed
Linear: 10000 mm/s 10000 mm/s
Joint: 100 % 100 %

Movement timeout
Timeout: 15000 ms

Vehicle position X: 1460,00 Y: 555.14 Z: 954.57
Eyebox position X: 284,86 Y: 54.57
Infomation: Central: You are inside of the eyebox plane.

Eyebox: ○ Topmost ○ Top ● Central ○ Bottom ○ Bottommost

Tool name: KeinTool
Tool values
X: 0
Y: 0
Z: 0
rX: 0
rY: 0
rZ: 0

Type family: FamilieA
Type: TypA

Currently selected
Type family: FamilieA
Type: TypA

Ini files

Position in Eyebox X: Y:
Position in Vehicle System X: Y: Z:
Position of Virtual Image X: Y: Z:
Rotation in Vehicle System rX: rY(LDW): rZ (LOW):
Interpolation ● Joint ○ Linear

Move Position

Relative motion
X 400,00
Y -284,86
Z 554,57
RX -180,00
RY 66,20
RZ -164,50
Fig. N/A (Denso)
Interpolation ● Joint ○ Linear
Coordinate system: Robot

IP-Address: 127.0.0.1
Robot type: Mitsubishi_CR750
Connected: ● Motor on: ●
Auto: ● Busy: ●
Error: ● E-Stop: ●
System OK - No errors
Password: ● 0

Softline

Modline

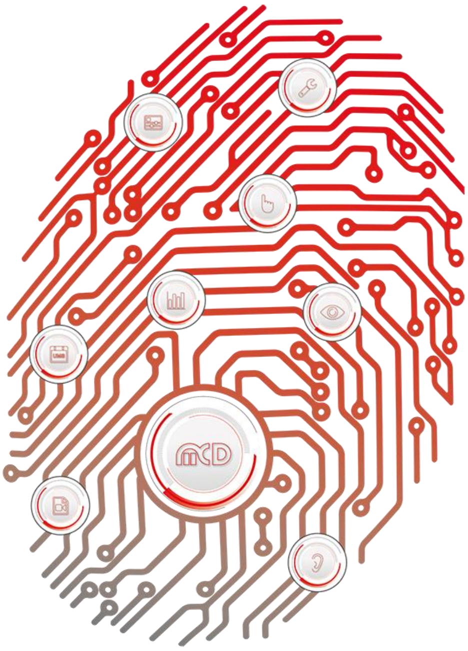
Conline

Boardline

Avidline

Pixline

Application



MCD Elektronik GmbH
Hoheneichstr. 52
75217 Birkenfeld
Tel. +49 (0) 72 31/78 405-0
Fax +49 (0) 72 31/78 405-10
info@mcd-elektronik.de
www.mcd-elektronik.com
HQ: Birkenfeld
Managing CEO: Bruno Hörter
Register Court Mannheim
HRB 505692

Template version: 5.4 / 2018-04-09
V1.7 2018- 09- 17 BM (JJ)

Table of Contents

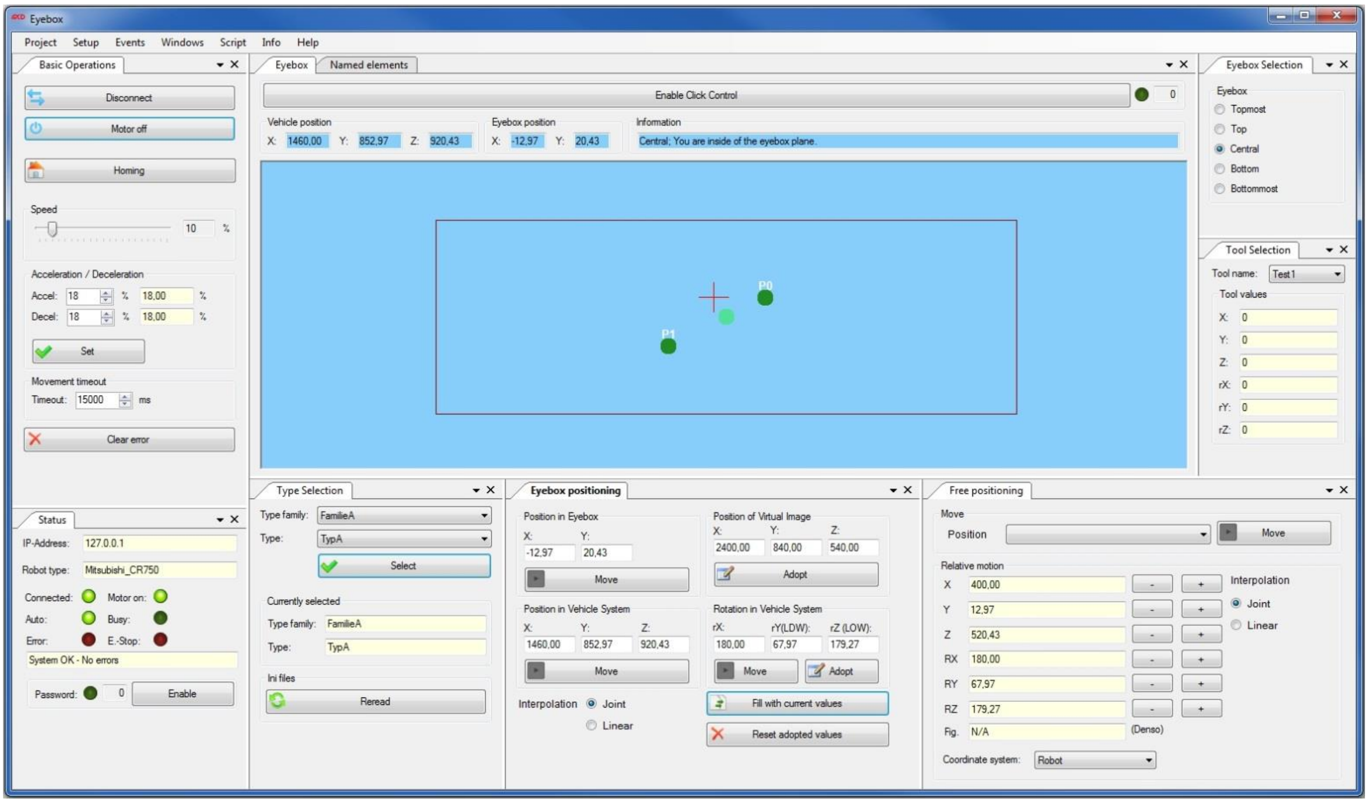
1. General

The primarily purpose of the Toolmonitor Eyebox is to position robots equipped with cameras in so - called eyeboxes.

The procedure can be carried out in the eyebox by clicking the mouse. The view of the camera at the end of the positioning process is pointed to a previously defined virtual image point.

The Toolmonitor also offers the option of controlling the robot outside the eyebox as well. Both relative and absolute movements are possible.

Mitsubishi robots in the CR750 controller series and Denso robots in the RC7 and RC8 controller series are currently supported. We can consider the integration of additional robot manufacturers and controllers upon request.



Speed: 10 %

Acceleration / Deceleration
Accel: 18 % 18,00 %
Decel: 18 % 18,00 %

Movement timeout
Timeout: 15000 ms

Vehicle position X: 1460,00 Y: 852,97 Z: 920,43

Eyebox position X: -12,97 Y: 20,43

Information: Central; You are inside of the eyebox plane.

Eyebox: ○ Topmost ○ Top ● Central ○ Bottom ○ Bottommost

Tool name: Test1

Tool values
X: 0
Y: 0
Z: 0
rX: 0
rY: 0
rZ: 0

Status
IP-Address: 127.0.0.1
Robot type: Mitsubishi_CR750
Connected: ● Motor on: ●
Auto: ● Busy: ●
Error: ● E.-Stop: ●
System OK - No errors
Password: ● 0

Type family: FamilieA
Type: TypA

Currently selected
Type family: FamilieA
Type: TypA

Ini files

Position in Eyebox
X: -12,97 Y: 20,43

Position of Virtual Image
X: 2400,00 Y: 840,00 Z: 540,00

Position in Vehicle System
X: 1460,00 Y: 852,97 Z: 920,43

Rotation in Vehicle System
rX: 180,00 rY(LDW): 67,97 rZ (LOW): 179,27

Interpolation ● Joint ○ Linear

Move
Position

Relative motion
X 400,00
Y 12,97
Z 520,43
RX 180,00
RY 67,97
RZ 179,27
Fig. N/A (Denso)

Interpolation ● Joint ○ Linear

Coordinate system: Robot

Figure 1: User Interface of the Toolmonitor Eyebox

Order number: # 150669

2. Safety Instructions

When using the Toolmonitor, follow the safety requirements for the robot. Before each use, be sure that no people can be harmed, for example by the use of suitable light barriers and blocked - off areas. Also be sure that the user always has an emergency stop switch within reach.

Move only at low speed during teach - in and testing. Check travel paths thoroughly before enabling automated processes. Please note that different interpolation modes can cause the arm to twist.

3. Introduction to Operation

3.1. Quick Guide

The following is a brief description of how the Toolmonitor Eyebox can be used.

3.1.1. Termination

To ensure communications between the Toolmonitor and the robot controller, a termination character must be set for some types of robot.

In the controller for the Mitsubishi robot, this can be done by setting the parameter “*NETTERM”* to 1. Then the termination character must be set in the socket options of the Toolmonitor.

NOTE: Not needed for Denso RC7 / RC8

3.1.2. Transfering Robslave

In order to control the Denso RC7 controller via the Toolmonitor, a robslave - task must be installed there. This can be transferred with the software tools of Denso RobMaster or WINCAPS III. Further information can be found in the documentation of the RC7 controller.

NOTE: Not needed for Denso RC7.

3.1.3. Creating Setting.ini

Before starting the application for the first time, you should be sure that the *Settings.ini file* is present. It must be stored in the same folder as the application itself. The content of this file is described in the *Ini files* chapter. Among other things, it defines the path to the file that will be created in the next step.

3.1.4. Creating Types and Types Families

The records for the Toolmonitor are divided into type families and types. All type families are located in a single folder, the path of which is defined in the *Settings.ini file*. Any number of type families can be created, and each type family consists of several types.

A type consists of the following files:

- Robot.ini,
- Coordinates.ini and
- 5 Eyebox files

If one of these files is missing, the type will not be loaded.

The content of these files and how they must be arranged is described in the *Ini files* chapter.

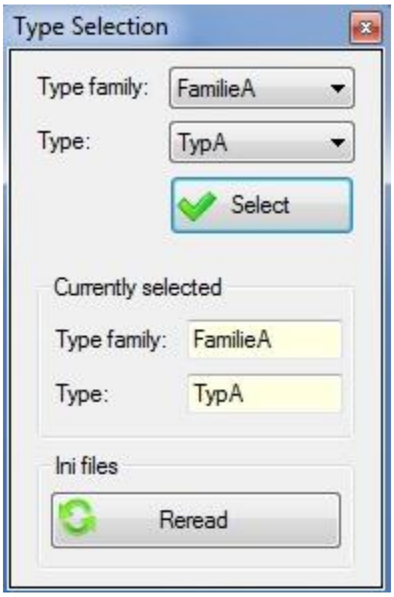
3.1.5. Starting the Toolmonitor

The Toolmonitor can now be started. If the *Settings.ini file* is missing, this is indicated by an error message. If no types can be found, the list for type selection will be blank.

By default, the window should now look like the illustration in chapter 1. If not, the default preset can be loaded from *Project → Presets*.

3.1.6. Select Type

The “*Type Selection”* interface can now be used to select the type. First select the type family, then the type itself. Via the button “*Select”* the files will be loaded.


Type Selection

Type family: FamilieA

Type: TypA

Currently selected

Type family: FamilieA

Type: TypA

Figure 2: Type Selection

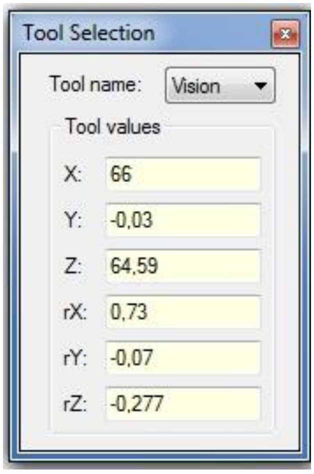
3.1.7. Set Speed

Once the type has loaded successfully, it is best to first set the speed using in the “*Basic Operations”* interface. During testing and teach - in, a speed of **less than 10%** is recommended.

WARNING: Risk of injury!

3.1.8. Select Tool

The *"Tool Selection"* interface can be used to select a tool the robot should use. This is important so that the correct coordinates can be displayed and moved to. How the tool has to be configured to permit the tool mechanism to work together with the Toolmonitor is described in the section *Configuring tool rotation*.



Tool name: Vision

Tool values

X: 66

Y: -0,03

Z: 64,59

rX: 0,73

rY: -0,07

rZ: -0,277

Figure 3: Tool Selection

3.1.9. Controlling the Robot

If the robot is not yet connected, a connection can be established using the interface *"Basic Operations"*. The status of the robot can be seen in the interface *"Status"*. If the motor is still turned off, turn it on now.

The robot can be moved using the interface *"Free Positioning"* and *"Eyebox Positioning"*. To make it leave the defined area shown in blue, you will have to enter a password. It is also possible to use the Eyebox interface to move the robot by clicking the mouse. To do this, it is always necessary to enter a password first. You can do this by clicking the button "*Enable Control".*

WARNING: Before moving the robot, be sure that no people or objects are within reach of the robot and could be harmed or damaged.

3.2. Configuring Tool Rotation

The following section describes the calculation and adjustment of the tool rotation so that the automatic alignment of the associated camera to the virtual image point can be carried out successfully. The prerequisite for this operation is that the base transformation (if necessary) must already have been carried out.

The base transformation is required when the robot is twisted relative to the vehicle coordinate system. This is achieved with a corresponding entry in the Robot.ini file.

Note: These instructions do not take any offset of the tool in the X, Y, and Z directions. These values must also be adjusted if necessary to guarantee the correct positioning of the robot.

The following procedure is only required once for a new tool.

3.2.1. Reset the Tool

First, set the offset and rotation for the tool in the corresponding Robot.ini to "0, 0, 0, 0, 0, 0".

```
Robot.ini:

...

[Tools]

; X Y Z rX rY rZ

Camera = 0, 0, 0, 0, 0, 0

...
```

3.2.2. Set Home Position

Set the home position to the desired center of the (central) eyebox.

NOTE: After setting up the tool, this can be changed at any time.

```
Robot.ini

...

; Position X Y Z rX rY rZ

HomePosition = 30, -350, 760, 0, 0, 0

...
```

Move to the home position.

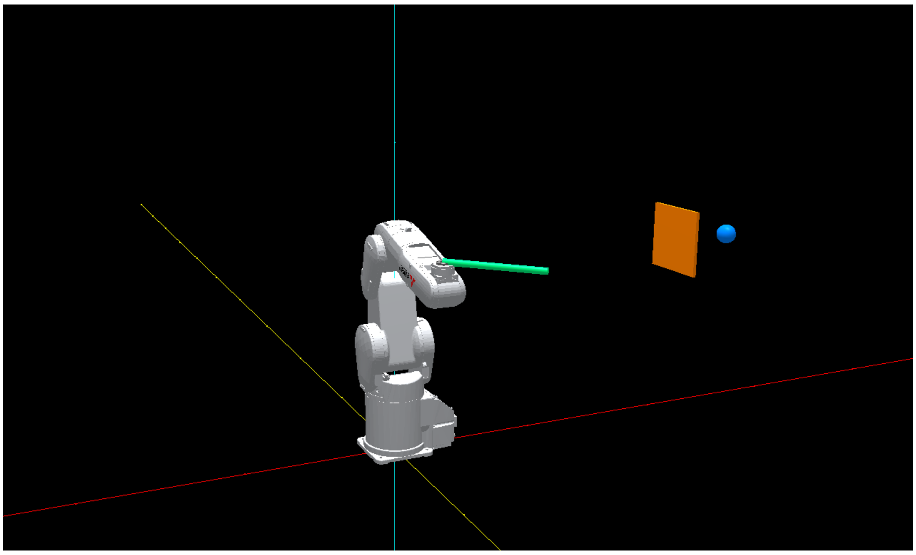


Figure 4: Target Home Position

(This illustration is only an example and does not reflect the concrete values.)

3.2.3. Set Y Rotation (rY)

Set the Y rotation for the home position so that the head of the robot (not yet the camera) faces in the direction of the projection surface and is aligned parallel with the X axis.

So either:

```
Robot.ini

...

; Position X Y Z rX rY rZ

HomePosition = 30, -350, 760, 0, -90, 0

...
```

or

```
Robot.ini

...

; Position X Y Z rX rY rZ

HomePosition = 30, -350, 760, 0, 90, 0

...
```

Move to the new home position.

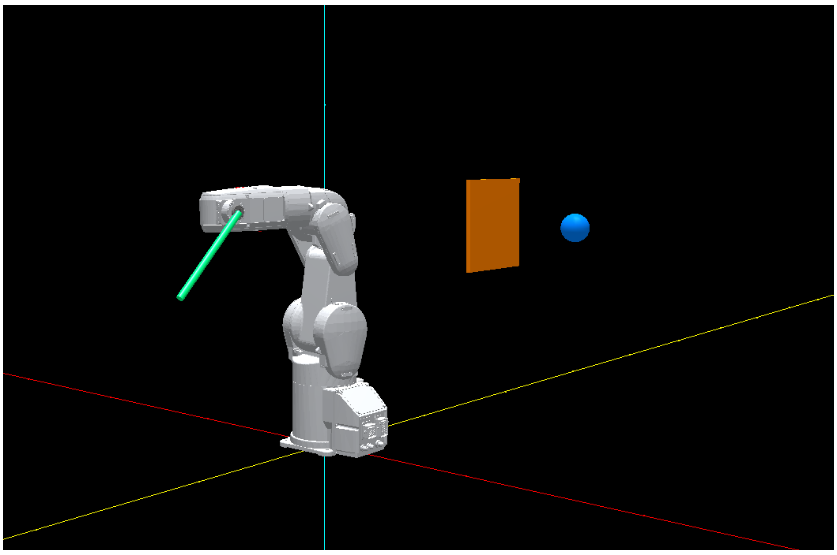


Figure 5: Set Y Rotation

(This illustration is only an example and does not reflect the concrete values.)

3.2.4. Adjust Step by Step

Now adjust the rotation of the tool (the camera) step by step so that it faces towards the projection surface and is aligned in parallel with the X axis. It is best to start by adjusting the rotation about X, then the rotation about Y, and finally the rotation about Z.

```
Robot.ini
...
; Position X Y Z rX rY rZ
Camera = 0, 0, 0, -90, 0, 135
...
```

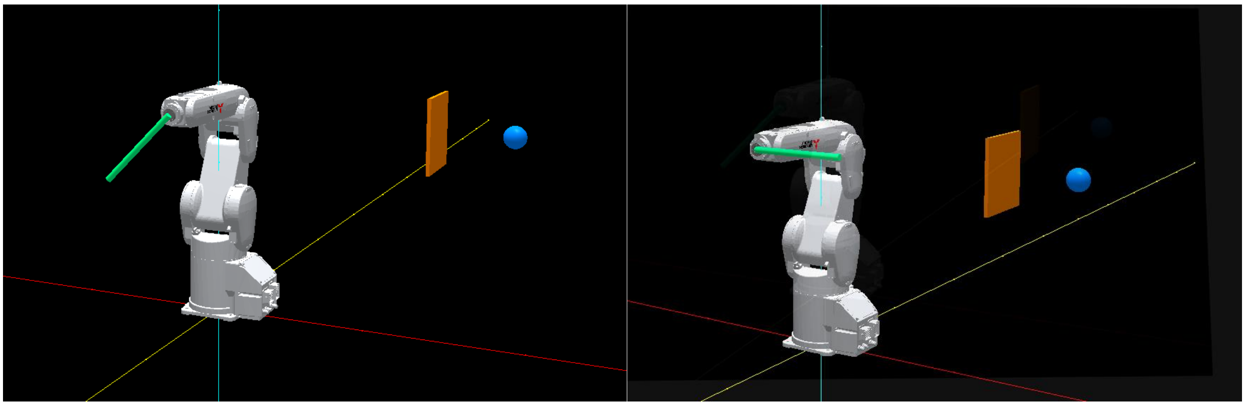


Figure 6: Adjust Settings Step by Step

(This illustration is only an example and does not reflect the concrete values.)

3.2.5. Correction of the Postion of Coordinates.ini

If the tool rotation has been successfully set, it is optimized for operation with the Eyebox interface. When operating with position data from the Coordinates.ini file, a correction offset is required.

This can be set in the Robot.ini by setting the flag ToolWasTaught = 1. If the view of the camera is oriented in the negative X-direction (taking into account the base translation), i.e. behind the robot, when moving to the temporarily set home position, the negative coordinate correction should be used.

If the value is not specified, ToolWasTaught = 1 applies for backward compatibility. Negative coordinate correction (UseNegativeToolCorrection) is used when the points of interest are behind (x <0) the robot. If the value is not specified, UseNegativeToolCorrection = 1 applies for backward compatibility.

[DEFAULT]

ToolWasTaught = 1

UseNegativeToolCorrection = 1

When using the Toolmonitor by setting Tool, Base, and desired Positioning, the so called “*gimbal lock”* can occur. This means that rotations around the X-axis have the same effect as rotations around the Z-axis. If this is the case, the Toolmonitor can be configured by setting the flag on ToolWasTaught = 2 to correct the behavior accordingly.

3.2.6. Result of the Tool Rotation Configuration

After the tool rotation has been set, the robot directs the view of the camera to the virtual image point of the loaded eyebox plane during the procedure in the eyebox. This is exemplified in the following pictures. In the diagram, the camera was extended for better comprehensibleness. It also shows that the robot remains in the eyebox plane.

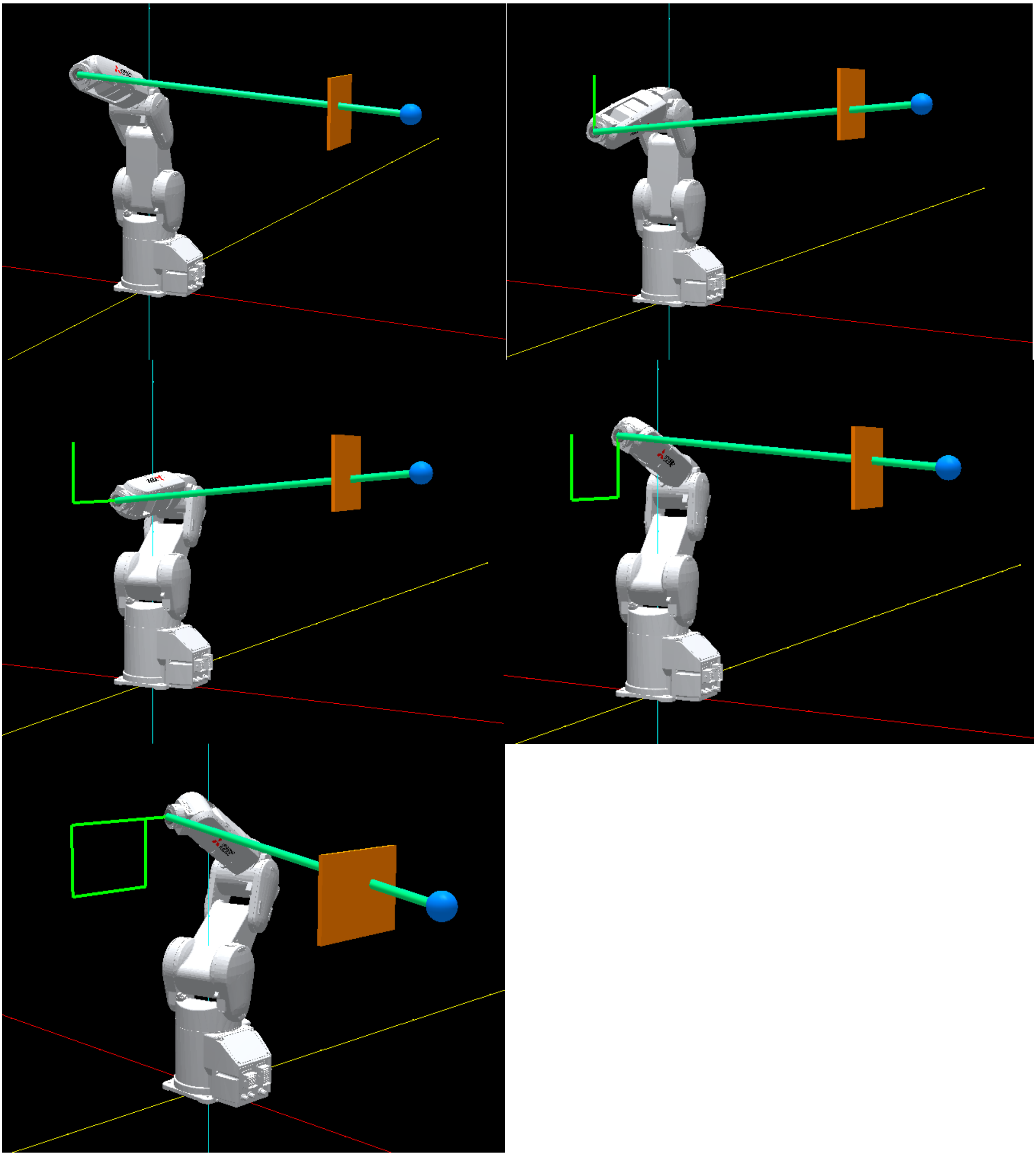


Figure 7: Settings Result

(The iIllustration is just an example and does not reflect the concrete values. In this example, the corners of an exemplary eyebox were approached with linear interpolation.)

4. Ini. Files

The data processed in the monitor is divided into type families. These type families consist of different types.

Each type consists of the following files:

- Robot.ini or DensoMidServer.ini
- Coordinates.ini
- 5 eyebox files: Bottommost, Bottom, Central, Top, Topmost

The data is structured as follows:

```
TypeFolder
|          FamilyA
|          |        TypeA
|          |        |       Robot.ini
|          |        |       Coordinates.ini
|          |        |       Eyebox
|          |        |       |              Bottommost_AnyName.ini
|          |        |       |              Bottom_AnyName.ini
|          |        |       |              Central_AnyName.ini
|          |        |       |              Top_AnyName.ini
|          |        |       |              Topmost_AnyName.ini
|          |        TypeB
|          |        |       Robot.ini
|          |        |       Coordinates.ini
|          |        |       Eyebox
|          |        |       |              ...
|          |        ...
|          FamilyB
|          |        ...
|          ...
```

The path to the type folder must be specified in the *Settings.ini* file. Information about how the content of this file must be structured can be found in the section on the *Settings.ini* file.

4.1. Settings.ini

The *Settings.ini* file must always be in the same folder as *EyeboxMonitor.exe*. The content of the file has the following structure:

```
[Settings]

RobotTYPE = Mitsubishi_CR750

Path_To_Types = %CURRENT%/TypeFolder
```

Here, *RobotType* specifies the type of the robot / controller. The possible robot types are currently: Mitsubishi_CR750 and Denso_RC8.

Path_To_Types specifies the path to the type folder in which the type families with the associated data are stored. Here, *CURRENT* is replaced with the current application path. It is also possible to work with absolute paths.

4.2. Robot.ini

Robot.ini stores basic information and settings for the robot. The Robot.ini file can also be named DensoMidServer.ini. The content of the file has the following structure (example records):

```
[CONNECTION]

IPAddress=192.168.0.1

Port=

[DEFAULT]

ExtSpeed=

Speed=

Accel=

Decel=

AllowedDistance=

; Position X Y Z rX rY rZ

HomePosition = 485.0, 200.0, 420.0, 90.0, 5.5, 180.0


[Tools]

; X Y Z rX rY rZ

Vision = 66.0, -0.03, 64.59, 0.73, -0.07, -0.277

LMK = 70.0, 0.0, 136.0, -0.4, -0.17, -0.18

Teachen = 25.0, -63.0, 186.0, 0.0, 0.0, 0.0

KeinTool = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0

[WORKS]

; X Y Z rX rY rZ

HUD = -766.7, 968.91, -469.4, 0, 0, 0

Teachen = -19.6, 818.59, 28.6, 0.0, 0.0, 0.0
```

```
KeinWork = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
[ALLOWEDAREA_VEHICLE]
Xmin=
Xmax=
Ymin=
Ymax=
Zmin=
Zmax=
rXmin=
rXmax=
rYmin=
rYmax=
rZmin=
rZmax=
[RobotTranslation]
; X Y Z rX rY rZ
Base = 0, 0, 0, 0, 0, 0
[TCPOffset]
;X, Y, Z, A, B, C
TCPOFFSET = 0.0,3.1,-0.1,3.8,1.5,3.8
[Offset]
; X, Y, Z
LHD = 1060.00, -840.00, 400.00
RHD = 1060.00, -840.00, 400.00
```

NOTE: Only the first value in the section *“RobotTranslation*” is loaded. The base translation is transmitted to the controller of the robot from the type selection. Before configuring the tool ration, be sure that the base translation has already been completed.

“AllowedDistance” configures the permitted distance in millimeter away from the eyebox plane while moving in the eyebox, before it is necessary to enter a password.

The area of movement of the robot can be limited by the Toolmonitor via the “*Allowed Area”*. If the name of the section is *“AllowedArea_Vehicle”*, the information is interpreted in the vehicle coordinate system. If the name is *“AllowedArea_Robot”*, the values are interpreted in the robot coordinate system. As of Eyebox V2.9, the section *“AllowedArea”* is no longer supported to prevent possible confusion. If both allowed areas for vehicle and robot coordinate system are defined, or if the old name *“AllowedArea”* has been used, an error message appears.

WARNING: The robot can interpret and, if necessary, optimize angle positions that have previously been checked by the *AllowedArea*. Changing one angle value can affect all three angle values, which cannot be checked by the Toolmonitor. The *AllowedArea* does not replace any defined motion limits that must be set in the controller of the robot. It is intended as additional assistance for the user, but cannot completely prevent a breaking out of the robot from the borders. For such functionality, refer to the instruction manual of the respective controller type.

The “*ToolWasTaught”* flag indicates that the tool has been set with the specified procedure described in this manual. If the value is not specified, *“ToolWasTaught = 1”* applies for backward compatibility. Negative Coordinate Correction (UseNegativeToolCorrection) is used when the points of interest are behind (x < 0) the robot. If the value is not specified, *“UseNegativeToolCorrection = 1”* applies for backward compatibility.

When using the Toolmonitor by setting tool, base, and desired positioning, the so called “*gimbal lock”* can occur. This means that rotations around the X - axis have the same effect as rotations around the Z - axis. If this is the case, the Toolmonitor can be configured by setting the flag on *“ToolWasTaught = 2”* to correct the behavior accordingly.

4.3. Coordinates.ini

The *Coordinates.ini* contains vehicle coordinates in the vehicle coordinate system which can be moved to using the program. The file can also include information (for example for appearance), but it will not be used in the Toolmonitor. The vehicle coordinates are represented as follows (example records):

```
[Fahrzeug_Koordinaten] // alternatively: [Denso_Fahrzeug_Koordinaten]
; X, Y, Z, A, B, C
1-C = 1668.000, -375.000, 845.000, -90.0, -4.59, 0.0,
2-C = 1668.000, -407.500, 845.000 -90.0, -4.59, -1.1,
3-C = 1668.000, -342.500, 845.000, -90.0, -4.59, 1.1,
```

4.4. Eyebox Files

A total of five eyebox files are required:

- Bottommost_AnyName.ini
- Bottom_AnyName.ini
- Central_AnyName.ini
- Top_AnyName.ini
- Topmost_AnyName.ini

NOTE: The original names Lowermost, Lowest, and Top are still supported.

The files contain the position and rotation of the eyebox in the vehicle coordinate system as well as the position of the virtual image the camera should face. The dimensions of the eyebox and the measurement points are also stored here. The content of the files has the following structure (example records):

```
 [COORDINATE_SYSTEMS]
; X Y Z
VEHICLE_POS = 1460, 840, 900
; A B C
VEHICLE_ROT = 0, 0, 0
; X Y Z
IMG_POS = 2000, 840, 900
; Top Bottom Left Right
EYEBOX = 100, 100, 300, 300
[MEASURING_POINTS]
; X Y Z
P0 = 1460, 835, 905
P1 = 1460, 845, 895
```

5. Operating of the Interface

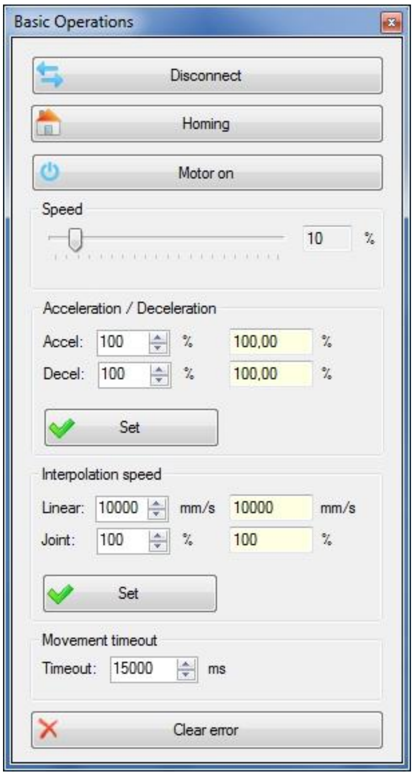
The individual elements of the user interface of the Toolmonitor are described below.

5.1. Basic Options

The user interface “*Basic Operations”* can be used to carry out basic functions.

The general basic operations are:

- Connecting and disconnecting from the robot / controller
- Turning the motor on and off
- Homing
- Adjusting the speed
- Adjusting the acceleration and braking
- Adjusting the interpolation speed
- Determining the positioning timeout
- Resetting error states



Speed: 10 %

Acceleration / Deceleration

Accel: 100 % 100,00 %

Decel: 100 % 100,00 %

Interpolation speed

Linear: 10000 mm/s 10000 mm/s

Joint: 100 % 100 %

Movement timeout

Timeout: 15000 ms

Figure 8: Basic Options

The positioning timeout can be used to determine how long a robot has to carry out a movement. If the robot does not reach the transmitted position in this period, it stops. Each of these actions can be carried out either by clicking the button on the user interface or by using the *virtual interface*.

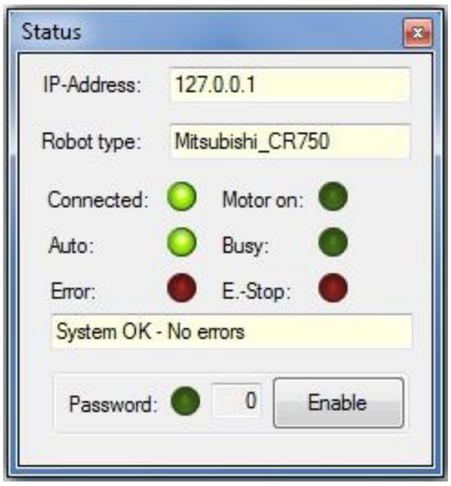
It is also possible to use the *virtual interface* to set and read variables on the controller of the robot as well as directly executing commands.

5.2. Status

The status interface can be used to view different information about the connection to the robot. This includes a view of the connection status, the state of the motor, and information about the emergency stop switch.

The connection status also shows the configured IP address of the robot as well as the robot type which must be defined in the *Settings.ini* file.

The text field on the other edge shows the error messages on the robot controller. Many of these functions can also be called from the v*irtual interface*.



IP-Address: 127.0.0.1

Robot type: Mitsubishi_CR750

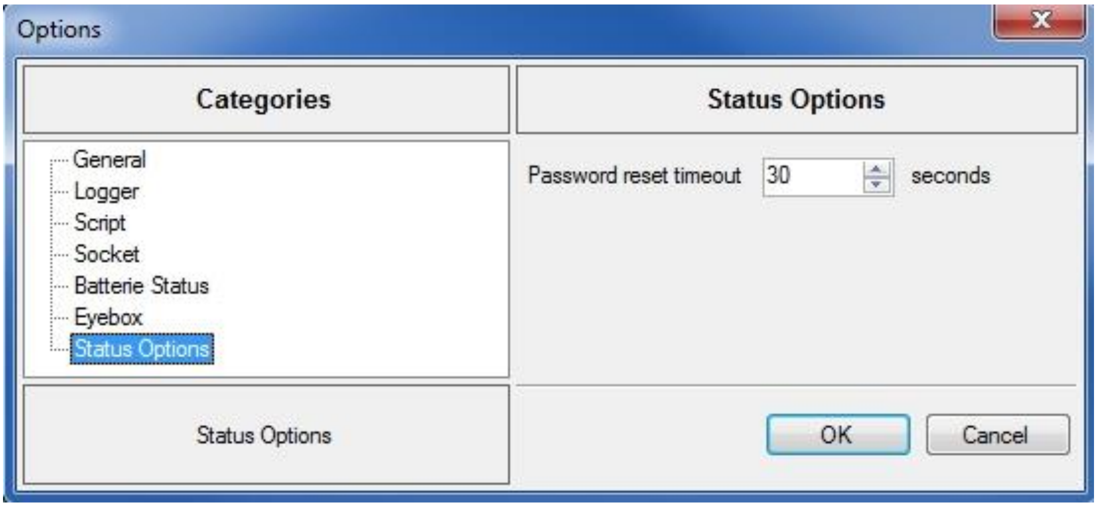
System OK - No errors

Password: 0

Figure 9: Status Interface

The countdown to password validity is also displayed. Password confirmation is required for any attempt to leave the blue safe zone or change the speed beyond a limit.

It is also possible to use the *"enable"* button to stop the controller or disable it before the password entry case applies. The time the password will remain valid before it has to be entered again can be changed in the options.



Password reset timeout 30 seconds

Figure 10: Status Options

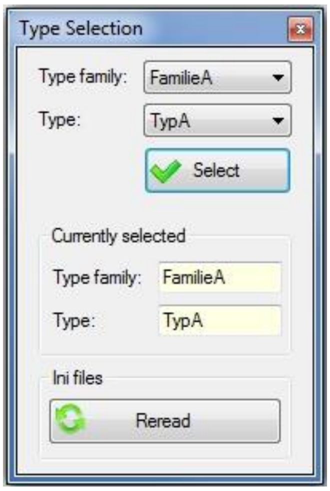
5.3. Type Selection

The interface *“type selection”* is selected to define the test type. This is used to decide which .ini files apply in the individual areas of the Toolmonitor. The selection is made by first reading in the corresponding type family, then selecting the family type on it.

The button *“Reread”* can be used to read the .ini files back in again. If the family and type names remain the same, the last type selected is read in again.

WARNING: Any existing connection to the robot will be disconnected.

The functions in the user interface can also be called through the *virtual interface*.



Type Selection

Type family: FamilieA

Type: TypA

Select

Currently selected

Type family: FamilieA

Type: TypA

Ini files

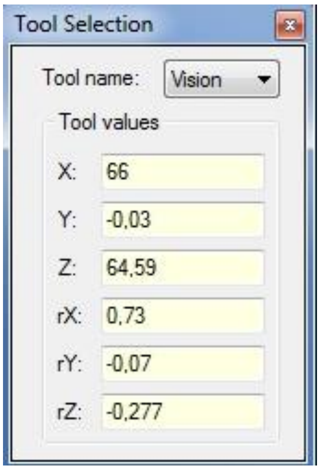
Reread

Figure 11: Type Selection

5.4. Tool Selection

The interface *“Tool Selection”* can be used to select the current tool for the robot. This transmits the tool coordinates to the controller of the robot.

It is also possible to change the tool by name and to query the current tool through the *virtual interface*.



Tool Selection

Tool name: Vision

Tool values

X: 66

Y: -0,03

Z: 64,59

rX: 0,73

rY: -0,07

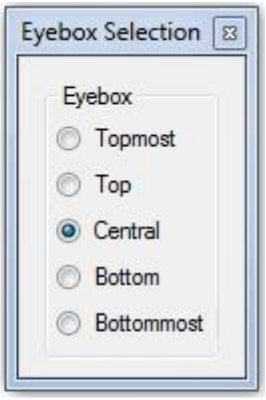
rZ: -0,277

Figure 12: Tool Selection

5.5. Eyebox Selection

The interface *"Eyebox Selection"* can be used to read in five different eyeboxes. If an eyebox is selected, its display in the eyebox interface is updated and the corresponding measurement points are loaded.

It is also possible to select the eyebox through the *virtual interface*.



Eyebox Selection

Eyebox

○ Topmost

○ Top

● Central

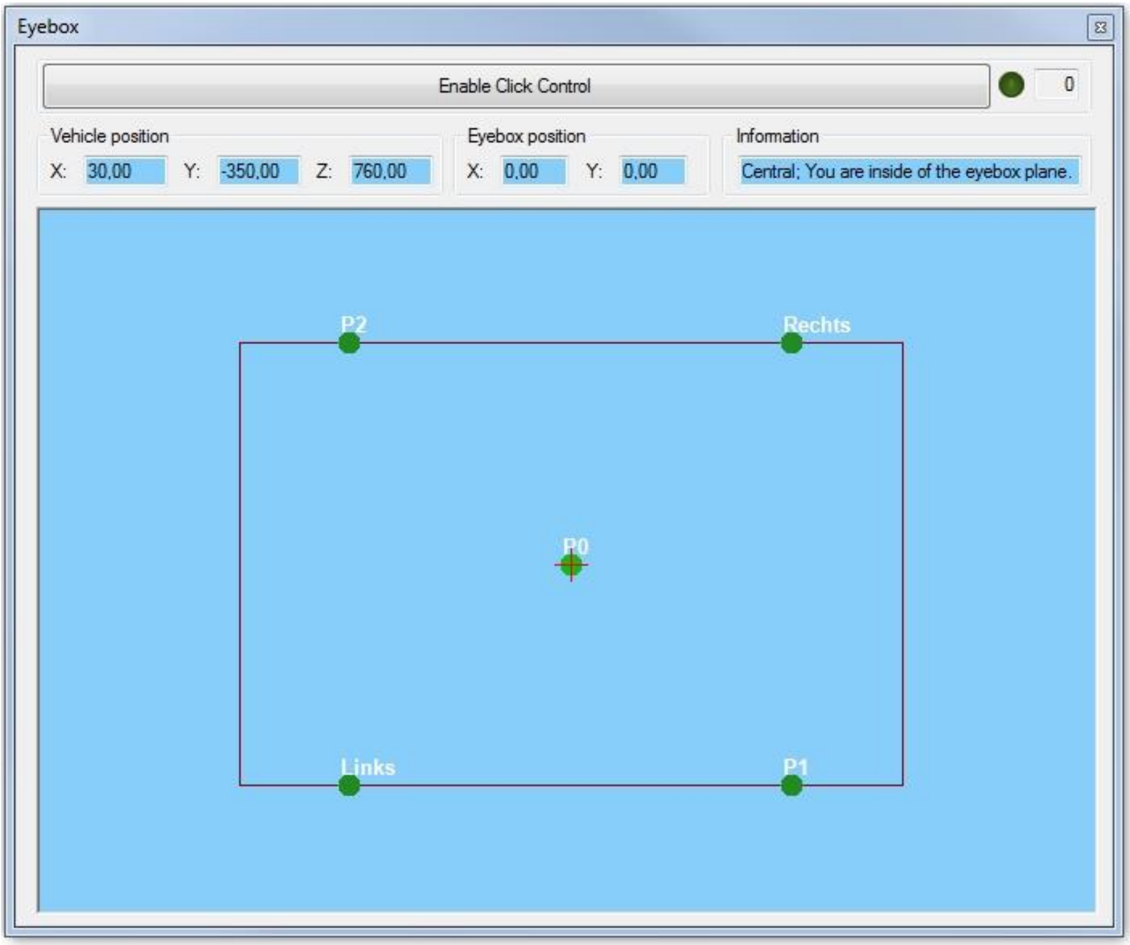
○ Bottom

○ Bottommost

Figure 13: Eyebox Selection

5.6. Eyebox Display

The Eyebox interface shows the currently selected eyebox with its associated measurement points. The position of the camera is also shown.



Eyebox

Vehicle position X: 30,00 Y: -350,00 Z: 760,00

Eyebox position X: 0,00 Y: 0,00

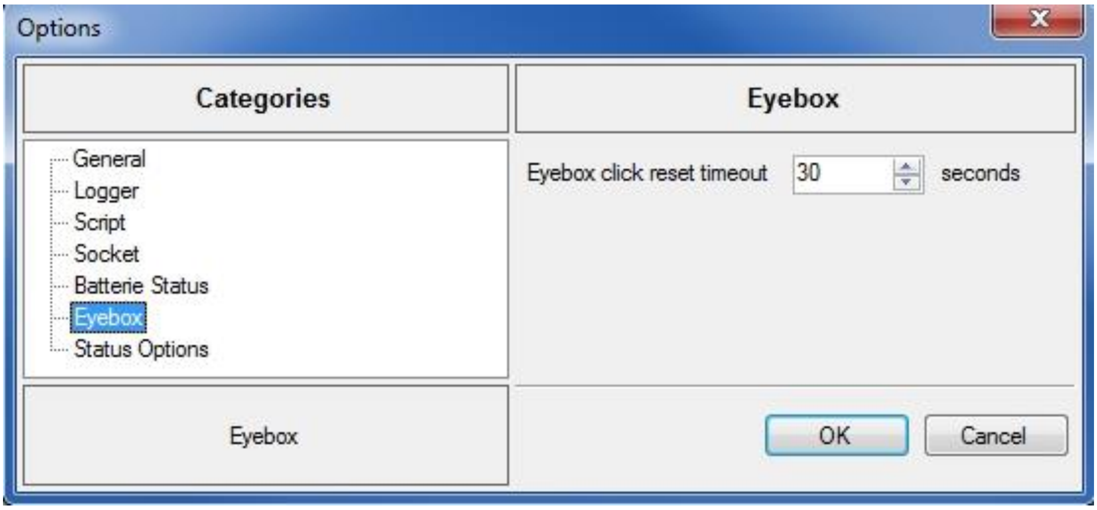
Information: Central; You are inside of the eyebox plane.

Figure 14: Eyebox Display

5.6.1. Moving the Camera

In order to click the eyebox to change the position of the camera, first the controller has to be enabled. This is done by clicking the button “*Enable Click Control”*. The purpose of this keyword is primarily to focus the attention of the user.

If no action takes place on the user interface for a certain period, the controller is disabled again. This also takes place if the button is clicked again. The length of this interval can be configured in the options.



Eyebox click reset timeout 30 seconds

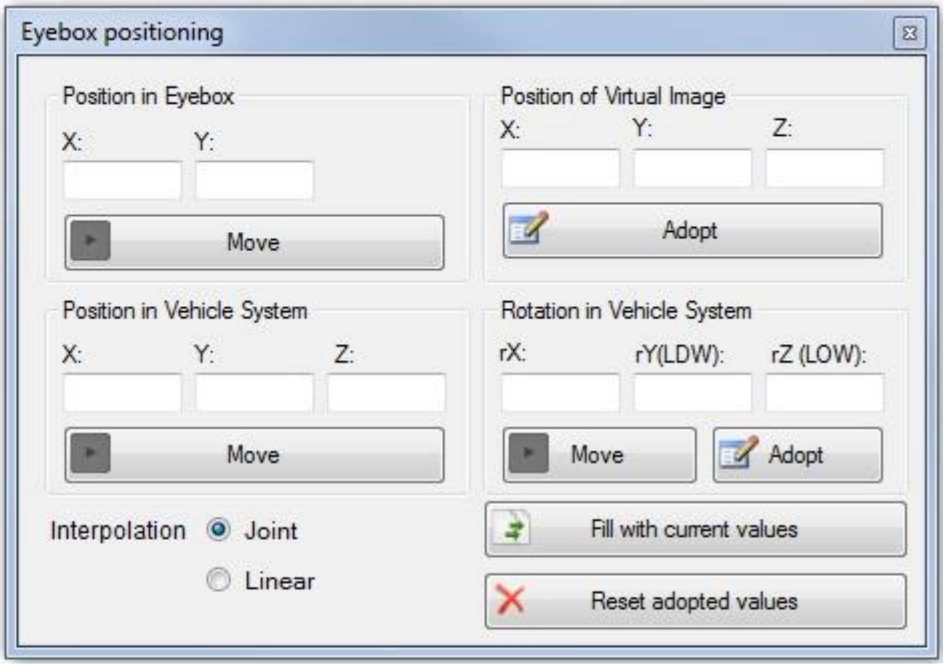
Figure 15: Eyebox Options

Since it is possible to enter a password and then use other user interfaces to move the robot back out of the eyebox plane, the text boxes showing the current position turn red if the robot leaves the eyebox. This is especially relevant when the robot leaves the plane orthogonally so that it is no longer obvious from the display whether the robot is still in the plane or not.

The *virtual interface* can be used to retrieve information about the eyebox.

5.7. Eyebox Positioning

The interface *“Eyebox Positioning”* can be used to move directly to positions in the vehicle coordinate system. It is also possible to change the rotation of the camera and the position of the virtual image here, and to move to positions in the PictureBox.



Interpolation ● Joint
○ Linear

Figure 16: Eyebox Positioning

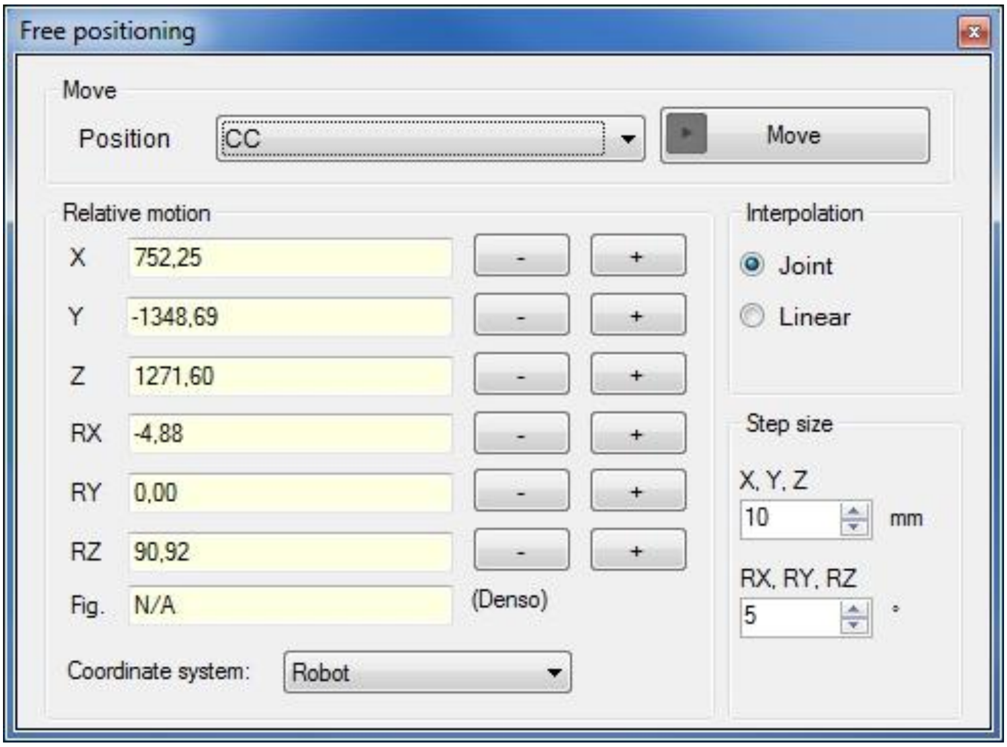
The commands can also be executed through the *virtual interface*.

Values set by the *“Adopt buttons”* stay set until the eyebox changes or the reset button is clicked. The "*Fill with Current Values"* button fills the text boxes with the current values.

5.8. Free Positioning

The interface for free positioning can be used to move the robot manually. It is also possible to move to defined "named" positions which can be viewed in the interface “*Named Positions”*.

The interpolation type (joint or linear interpolation) can be selected here, as well as the coordinate system in which the coordinates should be displayed.



Free positioning

Move

Position: CC

Relative motion

X	752,25	-	+
Y	-1348,69	-	+
Z	1271,60	-	+
RX	-4,88	-	+
RY	0,00	-	+
RZ	90,92	-	+
Fig.	N/A	(Denso)	

Coordinate system: Robot

Interpolation

● Joint

○ Linear

Step size

X, Y, Z: 10 mm

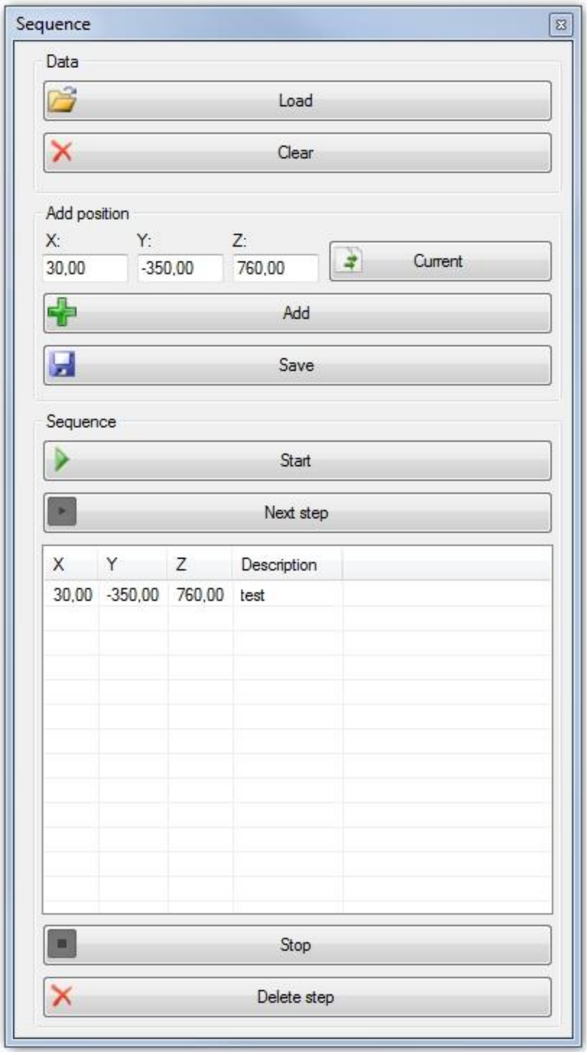
RX, RY, RZ: 5 °

Figure 17: Free Positioning

The *virtual interface* can be used to read out the current position in the robot and vehicle coordinate system and to move to positions.

5.9. Sequences

The interface *“Sequence”* can be used to load, edit, and store sequences. The individual points can then be moved to by starting the sequence. A sequence is started by clicking the “*Start”* button. The robot moves to the next point by clicking the “*Next”* button.



X: 30,00 Y: -350,00 Z: 760,00

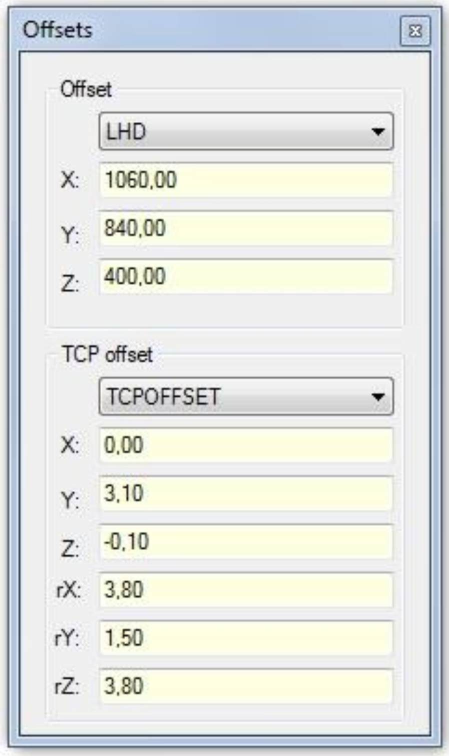
X	Y	Z	Description
30,00	-350,00	760,00	test

Figure 18: Sequences

Most functions, however, can also be executed through the *virtual interface*.

5.10. Offset Interface

The interface *“Offsets”* can be used to set both the general offset (e.g. for right - hand or left - hand steering) and the TCP offset.



Offsets

Offset

LHD

X: 1060,00

Y: 840,00

Z: 400,00

TCP offset

TCPOFFSET

X: 0,00

Y: 3,10

Z: -0,10

rX: 3,80

rY: 1,50

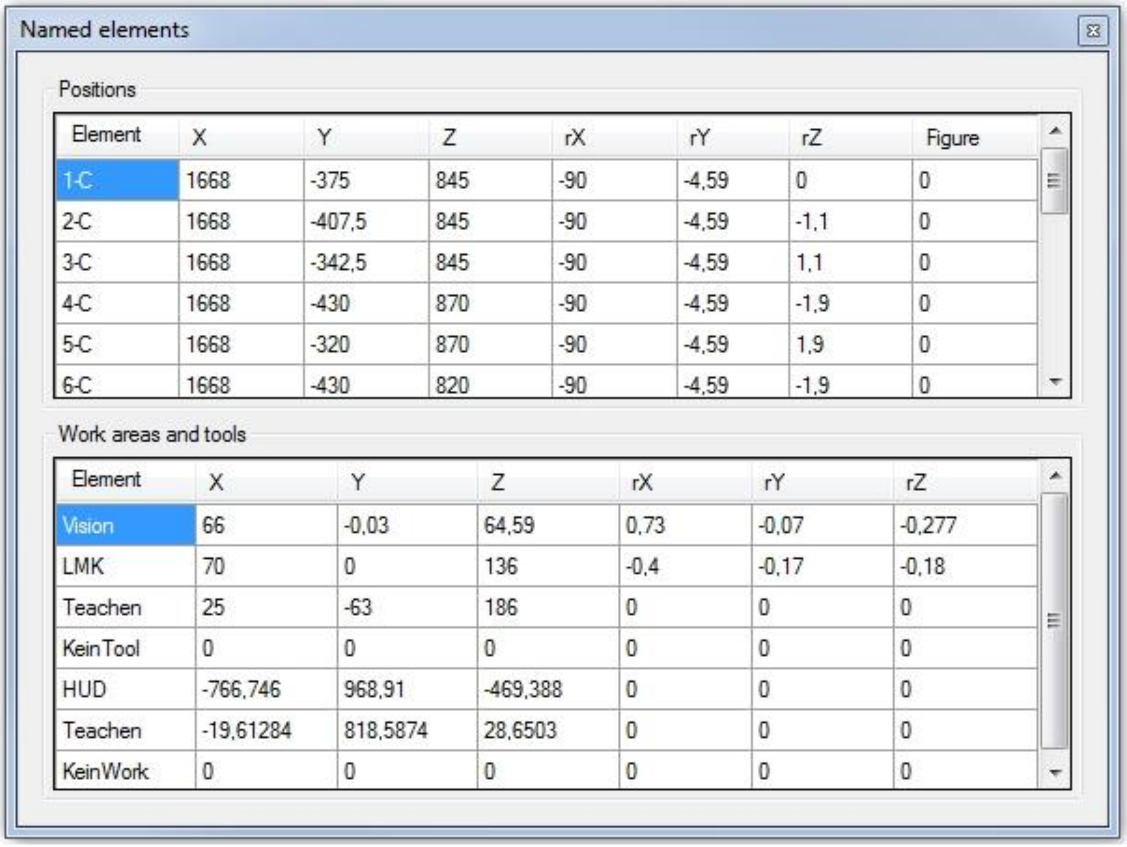
rZ: 3,80

Figure 19: Offset Settings

This can also be done through the *virtual interface*. In this case, the name of the offset to which the application should be set must be specified.

5.11. Named Elements

The interface “*Named Elements”* is used to display the vehicle coordinates defined in the .ini files of the loaded family type. The tools and works are also displayed here.



Positions

Element	X	Y	Z	rX	rY	rZ	Figure
1-C	1668	-375	845	-90	-4,59	0	0
2-C	1668	-407,5	845	-90	-4,59	-1,1	0
3-C	1668	-342,5	845	-90	-4,59	1,1	0
4-C	1668	-430	870	-90	-4,59	-1,9	0
5-C	1668	-320	870	-90	-4,59	1,9	0
6-C	1668	-430	820	-90	-4,59	-1,9	0

Work areas and tools

Element	X	Y	Z	rX	rY	rZ
Vision	66	-0,03	64,59	0,73	-0,07	-0,277
LMK	70	0	136	-0,4	-0,17	-0,18
Teachen	25	-63	186	0	0	0
Kein Tool	0	0	0	0	0	0
HUD	-766,746	968,91	-469,388	0	0	0
Teachen	-19,61284	818,5874	28,6503	0	0	0
Kein Work	0	0	0	0	0	0

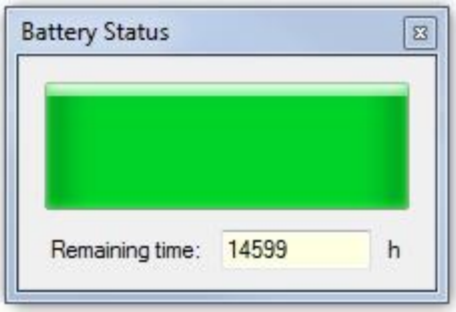
Figure 20: Named Elements

The *virtual interface* can be used to read out the positions.

The interface “*Free Positioning”* allows movement to the vehicle coordinates. Tools can be changed using the interface „*Tool selection”*.

5.12. Battery Status

The interface for the battery status can be used to read out the number of hours remaining before it will be necessary to replace the battery. The charge status is also displayed using a progress bar.



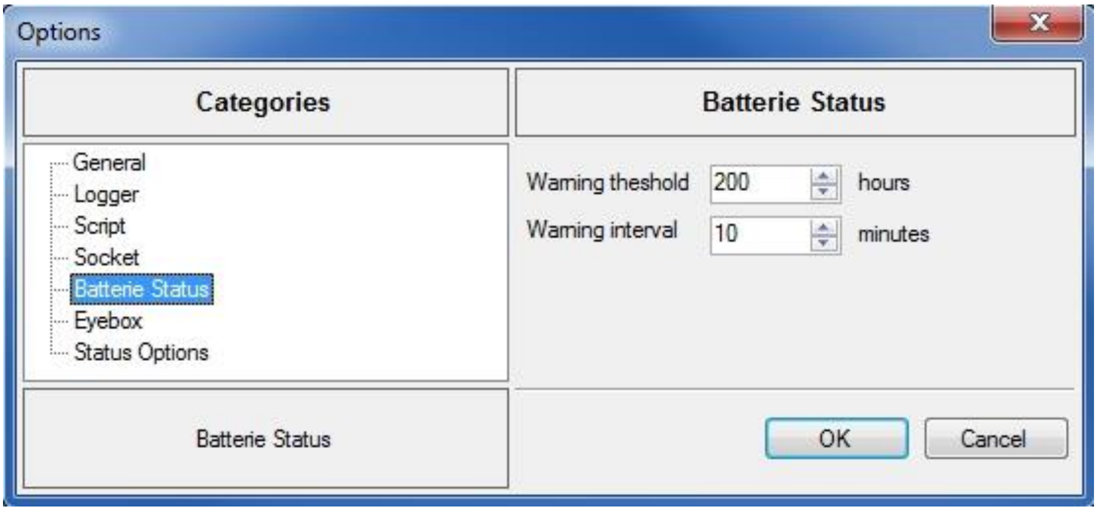
Remaining time: 14599 h

Figure 21: Display of Remaining Hours

This information can also be queried through the *virtual interface*.

NOTE: Denso RC8 does not support the battery status.

If the battery charge falls below a certain threshold value, this is displayed in a pop - up window. The threshold and interval at which the reminder should appear can be configured in the options.



Options

Categories

- General
- Logger
- Script
- Socket
- Batterie Status
- Eyebox
- Status Options

Batterie Status

Warning theshold 200 hours

Warning interval 10 minutes

Figure 22: Options of the Battery Status

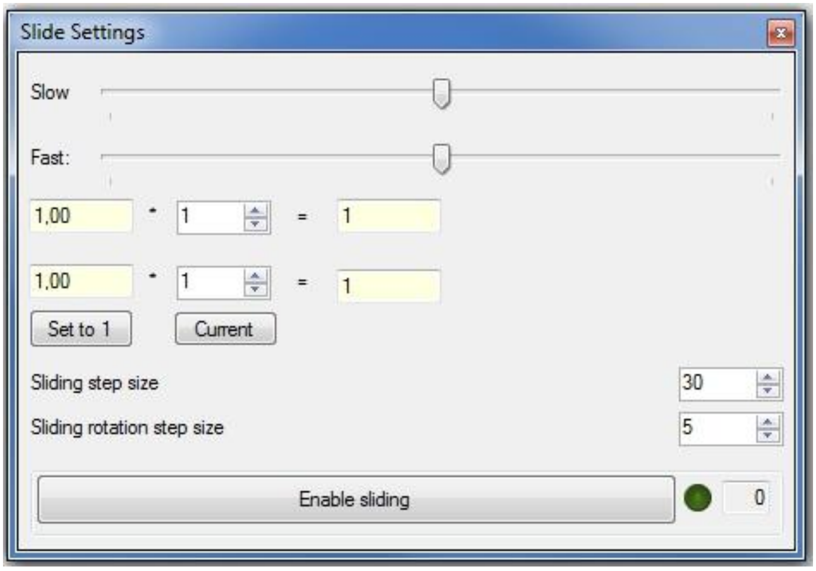
5.13. Sliding

Sliding permits movement in the eyebox plane (the area shown in blue). The focus must be on the eyebox interface. The arrow keys on the keyboard are used to control the movement. Only one coordinate can be changed at a time.

The following key combinations are supported:

- Arrow keys: Movement in the eyebox plane corresponding to the X and Y values on the eyebox display on the interface, at low speed
- Shift + arrow keys: Movement at high speed
- Ctrl + arrow keys: Slow rotation about the Y or Z axis
- Ctrl-Shift + arrow keys: Fast rotation

The speed and increments of the movement can be configured with the interface "*Slide Settings*". The speed is multiplied by the factor set. The track bars are scaled logarithmically.



1,00 * 1 = 1

1,00 * 1 = 1

Sliding step size 30

Sliding rotation step size 5

0

Figure 23: Settings of the Sliding Function

It is recommended that small increments should be used, since there is no influence on how accurately the controller of the robot will plan its path to this position. With larger increments, it could be that the robot would leave the eyebox on its way to the target position.

If the speed exceeds the speed limit or if the robot is not located in the eyebox plane, sliding is not permitted without entering a password.

To be able to use sliding, it must first be enabled by clicking the button "*Enable sliding"*. It will be disabled again after the countdown expires or if the button is clicked again.

5.14. Optics Mode

When using the *"Optics Mode"*, the necessary information required to define an eyebox plane can be specified without using a given virtual image position. For the Vehicle_ROT, the value LOOK_DOWN_ANGLE is used instead of the B value, and the LOOK_OVER_ANGLE instead of the C value.

For example:

VEHICLE_ROT = 20, 0, 0

LOOK_DOWN_ANGLE = 40

LOOK_OVER_ANGLE = 30

PROJECTION_DISTANCE = 2000

The used rotation then is (20, 40, 30). Instead of the IMG_POS the value PROJECTION_DISTANCE is used in this mode.

The virtual image position, which is described in normal mode with IMG_POS, is then calculated by the Toolmonitor. If you draw a line through the virtual image position and VEHICLE_POS, it would be orthogonal to the plane defined with VEHICLE_POS and (VEHICLE_ROT.A | LOOK_DOWN_ANGLE | LOOK_OVER_ANGLE). The orthogonal intersects the plane in point VEHICLE_POS. The distance between VEHICLE_POS and the calculated IMG_POS is the value PROJECTION_DISTANCE.

Measuring points are relatively defined to the VEHICLE_POS in *Optics Mode*. Absolut points are ignored. The section is called MEASURING_POINTS_REL.

```
[MEASURING_POINTS_REL]
;     X     Y     Z
PB=-0.75, -80, 0
```

The *Optics Mode* can be activated via the *virtual interface*.

```
SetValue("Eyebox.OpticsMode", 1); // Turn on optics mode
SetValue("Eyebox.OpticsMode", 0); // Turn off optics mode
```

The optics mode is disabled by default after program start. This is also the case when changes are made in the options of the Toolmonitor. When a reread of the ini files is performed, the Optics Mode remains activated.

6. Error Messages

This section describes different error messages which can occur during use of the Toolmonitor. Note that this is not a complete list.

Message	**Description**	**Solution**
The file 'settings.ini' could not be found	The Settings.ini file could not be found or loaded.	Be sure that the Settings.ini file is located in the same folder as the application and that it contains the data described in the *Settings.ini* section.
Connecting to the robot failed *Or:* Robot is not connected *Or:* Could not reconnect to robot after type selection.	No connection could be established to the robot.	If no type is loaded, the Toolmonitor will attempt to connect to the last robot in use. So be sure to load a type first. If the error continues to occur, check the connection settings in the Robot.ini file for the type loaded. Also be sure that the robot is connected to the computer and that the correct network settings have been configured.
Could not set speed from .ini file *Or* Setting override failed	The Toolmonitor has attempted to set the speed on the controller of the robot, but this action failed.	Be sure that the controller of the robot is in automatic mode and that the robot is connected.
No robot type was declared in settings.ini	There was no supported robot defined in the Settings.ini file.	Define the type of the controller / robot in the Robot.ini file. You can find more information in the chapter Settings.ini.
Turning on the motor failed *Or:* Turning off the motor failed	The servo / motor could not be turned on or off.	Be sure the controller is in automatic mode and repeat the action.
Starting RobSlave task failed *Or:* Stopping RobSlave task failed	The RobSlave task could not be started or stopped.	This function is available only for Denso robots. Be sure the controller is in automatic mode.
Could not clear error	The error messages could not be cleared on the controller.	Be sure that the controller is in automatic mode and there is no emergency stop situation. In the latter case, first acknowledge the emergency stop and repeat the action.
Homing was not successful	An error occurred during the homing process.	Be sure that the position specified in the Robot.ini file can be reached by the robot and that it can be reached during the positioning timeout period specified.

Please select a type *Or:* No type selected	For most actions, it is necessary to select a type first.	Select a type in the interface “*Type selection”* first.
Setting acceleration and deceleration failed	The acceleration and deceleration could not be set successfully.	Check that the robot is connected and is in automatic mode. Then execute the action again.
Moving the robot in the given time failed.	The robot could not carry out a requested movement successfully.	Be sure the robot can reach the position specified and that this is possible during the specified positioning timeout.
Position outside of allowed range	The position to be moved to is outside the limits specified in the Robots.ini file in the “*Allowed Area”* section.	If necessary, expand the movement range of the robot in the *“Allowed Area”* section of the *Robots.ini file*.
TypeFolder was not read successfully	The type folder with the families and types could not be read in successfully.	Be sure the specified path to the types is specified correctly in the Settings.ini file and that the Toolmonitor has read privileges to the folder specified. Be sure as well that the folder has the tree structure defined in the chapter *.Ini files*.
Could not move robot. Current position is not available.	For relative movements, the robot first queries the current position of the robot. If it cannot be read, the action is canceled.	Be sure the robot is connected and in automatic mode.
Could not set tool on robot controller or Changing the tool failed	The selected tool could not be transmitted to the robot during type selection or manual tool selection.	**WARNING:** Do not move the robot if the tool could not be set successfully. Be sure the robot is connected and in automatic mode. Then click the button *“*Reread” in the interface “*Type Selection”* to be sure that the tool is transmitted again, or select the tool again from the interface *“Tool Selection”* interface.
Servo is off	The servo / motor is turned off.	Turn the servo / motor on by clicking the Motor on button on the Basic Operations interface.
Emergency stop	The emergency stop button was pressed. Actions can then not be carried out.	The emergency stop must be acknowledged on the hardware first. The error status can then be reset by clicking the “*Clear”* button in the interface *“Basic Operations”*.
Syntax error in ini file + the name of the file. *Or:* Data in ... is faulty	The syntax or content of the specified file is incorrect.	Check the data in the .ini file and correct the syntax accordingly.

7. Virtual Interface

The *virtual interface* can be used to control the Toolmonitor remotely from other applications.

The program can be controlled in the higher - level external software using a COM / DCOM interface or a Net assembly. This means it can be integrated into a variety of applications such as Microsoft Visual Studio® (C#, C++, Visual Basic), Microsoft Office® (such as Excel®), Open Office®, LabView® and MCD TestManager CE.

The commands of the *virtual interface* can be found in the software help for the Toolmonitor. The help can be retrieved via the menu *Help* → *Eyebox Help*.