

# **Motion Tool Schunk (MTS)**

## **for electrical 'Pick and Place' unit**

### **PPU-E**

### **Software Manual**



**Imprint:****Copyright:**

This manual remains the copyrighted property of SCHUNK GmbH & Co. KG. It is solely supplied to our customers and operators of our products and forms part of the unit. This documentation may not be duplicated or made accessible to third parties, in particular competitive companies, without our prior permission.

**Technical changes:**

We reserve the right to make alterations for the purpose of technical improvement.

**Edition:** 09.01 / 02.05.2012 / en

© SCHUNK GmbH & Co. KG, Lauffen/Neckar

All rights reserved

Dear Customer,

Congratulations on choosing a SCHUNK product. By choosing SCHUNK, you have opted for the highest precision, top quality and best service.

**You are going to increase the process reliability of your production and achieve the best machining results – to your complete satisfaction.**

**SCHUNK products are inspiring.**

Our detailed assembly and operation manual will support you.

Do you have further questions? You may contact us at any time – even after purchase. You can reach us directly at the addresses listed in the final chapter of this manual.

Kindest regards,

SCHUNK GmbH & Co. KG  
Precision Workholding Systems  
Bahnhofstr. 106 – 134  
D-74348 Lauffen/Neckar

Tel. +49-7133-103-2503  
Fax +49-7133-103-2189  
automation@de.schunk.com  
www.schunk.com



## Table of contents

<b>1</b>	<b>About this manual.....</b>	<b>5</b>
1.1	Purpose/validity .....	5
1.2	Target groups.....	5
1.3	Symbols in this manual .....	5
<b>2</b>	<b>OPC server and gateway requirements .....</b>	<b>6</b>
<b>3</b>	<b>Motion Tool Schunk (MTS) installation.....</b>	<b>7</b>
<b>4</b>	<b>Before first use – connection settings.....</b>	<b>8</b>
4.1	User’s PC IP settings .....	8
4.2	Connection cable PC <-> Master controller.....	8
4.3	OPC server and gateway settings.....	9
4.4	Set correct project name .....	10
<b>5</b>	<b>Motion Tool Schunk (MTS) for PPU-E application description.....</b>	<b>14</b>
5.1	Main window .....	14
5.1.1	Menu .....	15
5.1.2	Symbol list .....	16
5.1.3	Logger .....	16
5.1.4	Connection status bar.....	16
5.2	Preferences window .....	17
5.3	“Module: PPU-E” window .....	18
5.3.1	“Command” buttons.....	19
5.3.2	Movement tabs.....	20
5.4	Axis window .....	22
5.4.1	“Command” buttons.....	23
5.4.2	Sub-tab “Jog” .....	24
5.4.3	Tab “Settings” .....	24
5.5	Program editor window .....	26
5.5.1	Tab “Program” .....	26
5.5.2	Tab “Code” .....	27
5.6	Application users.....	31
5.7	Application languages .....	32
5.8	External logger .....	32

---

<b>6</b>	<b>Appendix .....</b>	<b>33</b>
6.1	Programming commands .....	33
6.1.1	Motion Commands .....	33
6.1.2	Parameter Commands .....	36
6.1.3	I/O Commands .....	38
6.1.4	Program Organization Commands .....	38

## Applicable documents (on CD-ROM)

- Manual and references for MTS Software for PPU-E
- Assembly and operating manuals for linear motor drives
- General terms of business (AGB)

The specified data can alternative downloaded on the following page: [www.schunk.com](http://www.schunk.com)

# 1 About this manual

## 1.1 Purpose/validity

This manual is part of the MTS Software for PPU-E and describes how to safely and properly start up an electrically driven PPU-E30 unit.

This manual is only valid for the product specified on the front page.

## 1.2 Target groups

Target group	Task
Manufacturer, operator	<ul style="list-style-type: none"> <li>➔ Keep this manual accessible for the relevant personnel at all times.</li> <li>➔ Require personnel to read and observe this manual and the applicable documents, especially the safety notes and warnings.</li> </ul>
Skilled personnel, fitters	<ul style="list-style-type: none"> <li>➔ Read, observe and follow this manual and the applicable documents, especially the safety notes and warnings.</li> </ul>

Table 1

## 1.3 Symbols in this manual

To give you quick access to information, the following symbols will be used in this guide:


Symbol	Meaning
 <b>WARNING</b>	Danger for persons. Non-observance can cause death or serious injuries.
➔	Instruction for action, also measures in a warning or note.
1. 2. 3. ...	Step-by-step instruction for action. ➔ Observe the order.

Table 2

## **2 OPC server and gateway requirements**

MTS for PPU-E serves as test software. With the help of this software, the user can easily prepare the motion sequence for the PPU-E module.

User's PC requirements:

- Windows XP (with service pack 3)
- Network connection
- 30 MB free space on hard disk
- 256 MB RAM or more.

### 3 Motion Tool Schunk (MTS) installation

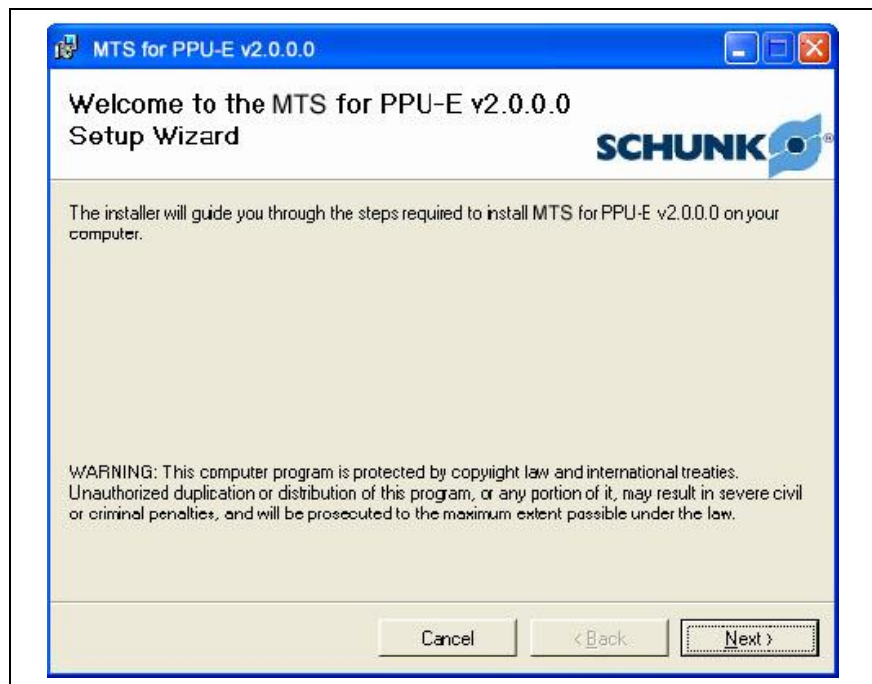


Fig. 1 MTS Installer

To install MTS, the user must run the supplied installation package:

- MTS for PPU-E Install EN vx.x.x.x (English version)  
**or**
- MTS für PPU-E Install DE vx.x.x.x (German version)

➔ Installation routine will install MTS for PPU-E vx.x.x.x and CoDeSys OPC Server v.2.3.13.4

➔ After installation, the PC should be **restarted**.

#### Note

If the installation cannot be performed, the user's PC should be connected to the network (internet).

The installer checks which *.net* version is installed on the user's PC. If the appropriate *.net* version (v2.0.50727 or higher) cannot be found, *.net* will be downloaded and installed directly from the network.

## 4 Before first use – connection settings

### 4.1 User's PC IP settings

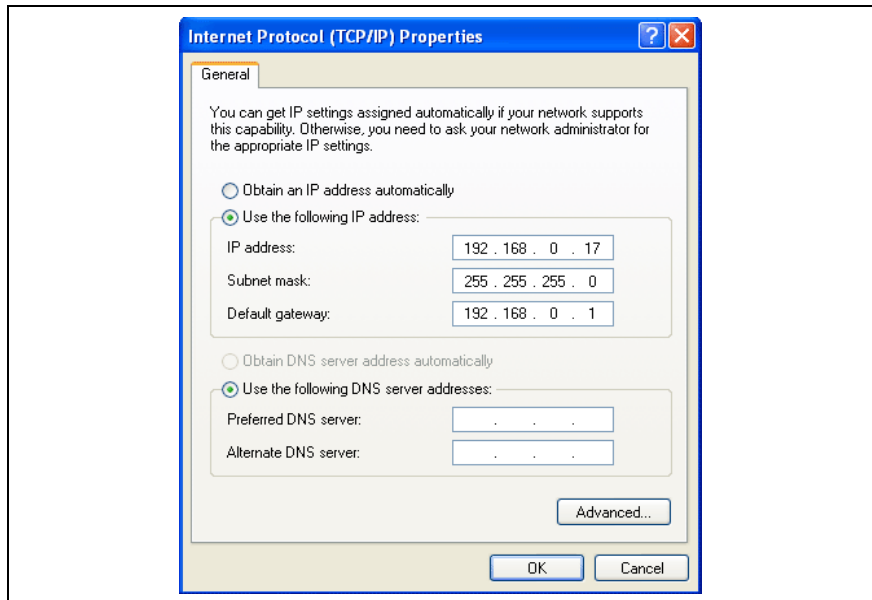


Fig. 2 User's IP settings

Following installation of the software, the user's PC IP address must be set to the proper value.

Master controller's default IP address: **192.168.0.1**

The user's PC should have an IP address from the same IP class (in this case address class **C**) e.g.: 192.168.0.17

### 4.2 Connection cable PC <-> Master controller

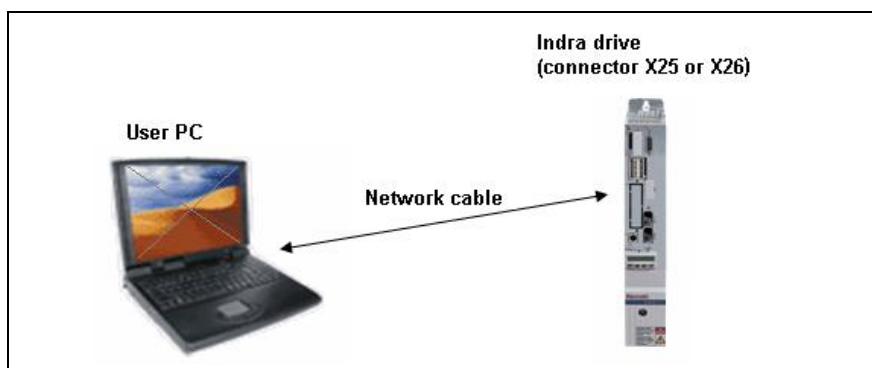


Fig. 3 User's PC <-> Master controller connection

The user's PC must be connected to the master controller with a simple network cable.



### 4.3 OPC server and gateway settings

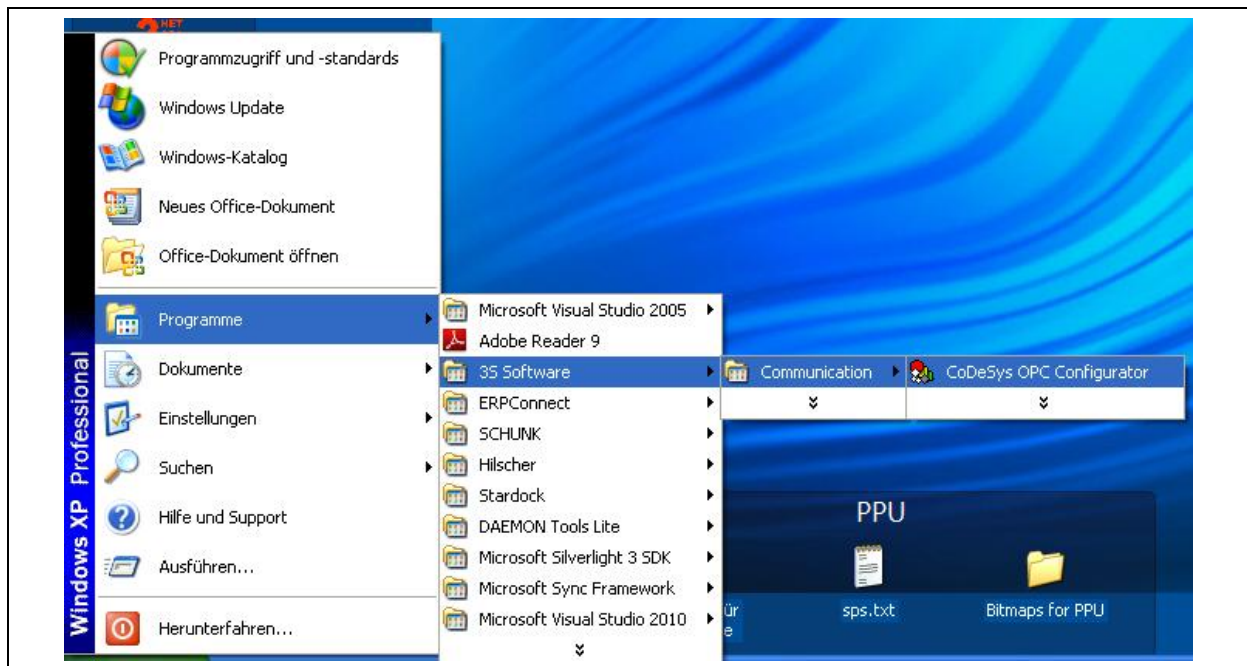


Fig. 4 OPC configuration software

To establish a connection between the master controller (PLC) and the user's PC, the OPC server and gateway should be configured (once only).

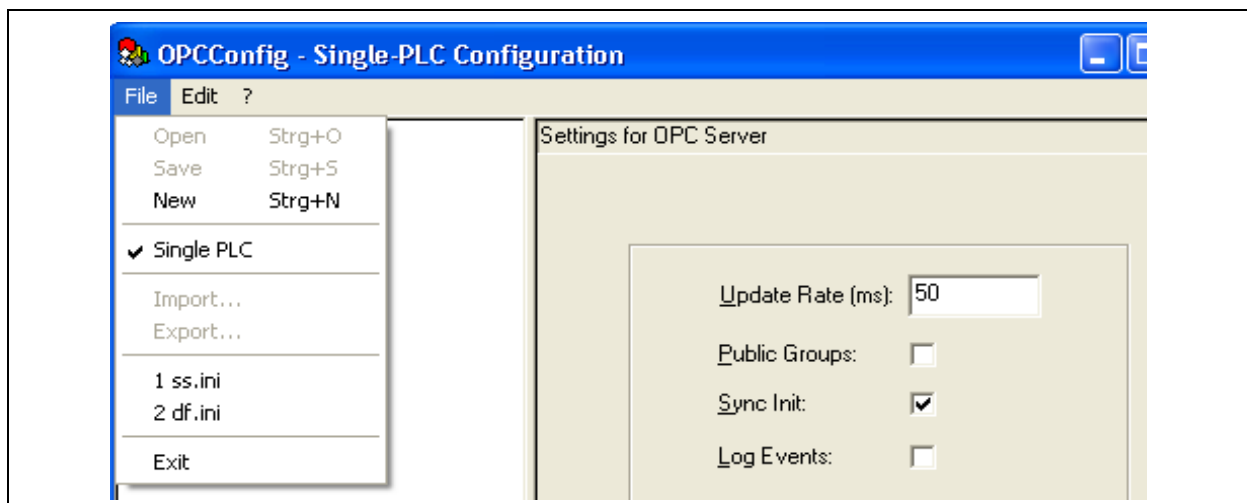


Fig. 5 OPC server configuration

1. Activate the “Single PLC” option
2. Set “Update Rate” to 50 ms for the OPC server
3. Activate the “Sync Init” checkbox for the OPC server

## 4.4 Set correct project name

Default: PPU\_E30\_Koffersteuerung\_v01\_MLD001.pro

### Note

This name is also the name of the IndraWorks project in the master controller!

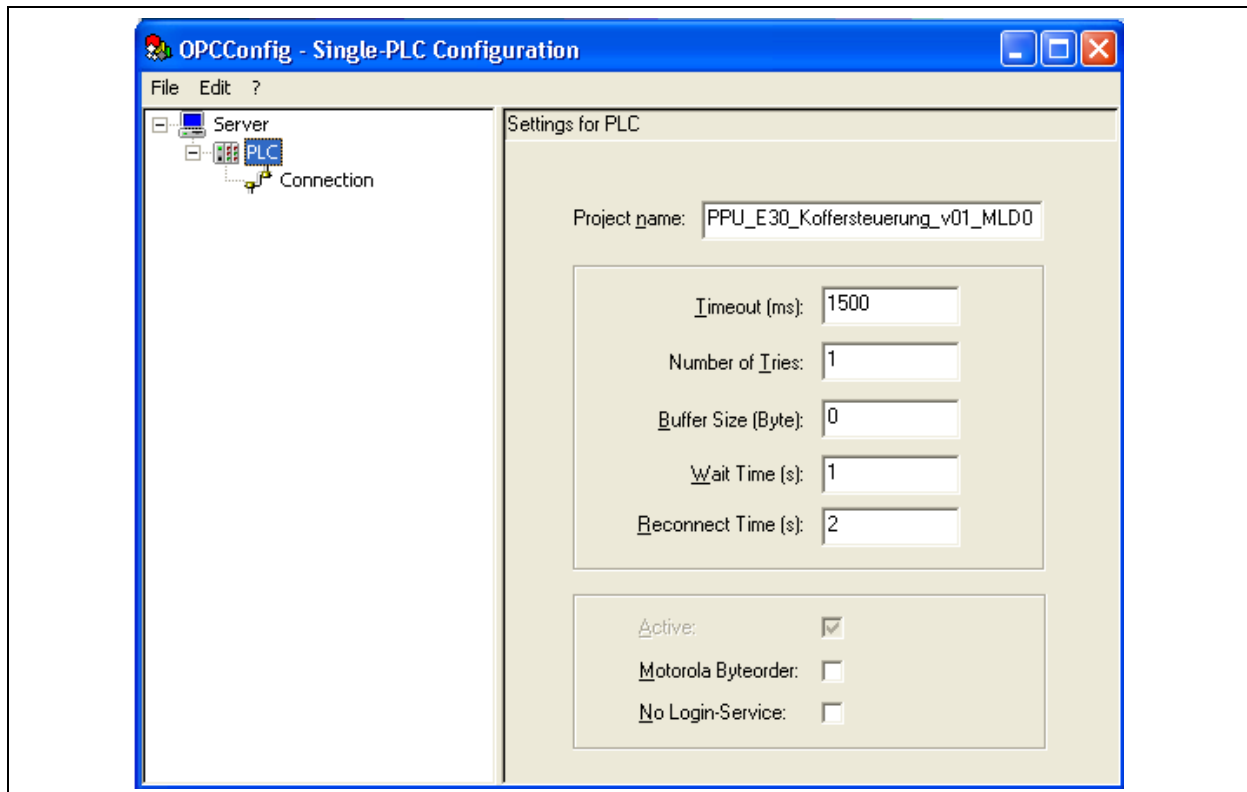


Fig. 6 PLC configuration

1. Set "Timeout" to 1500 ms
2. Set "Number of Tries" to 1
3. Set "Buffer Size" to 0 bytes
4. Set "Wait Time" to 1 s
5. Set "Reconnect Time" to 2 s

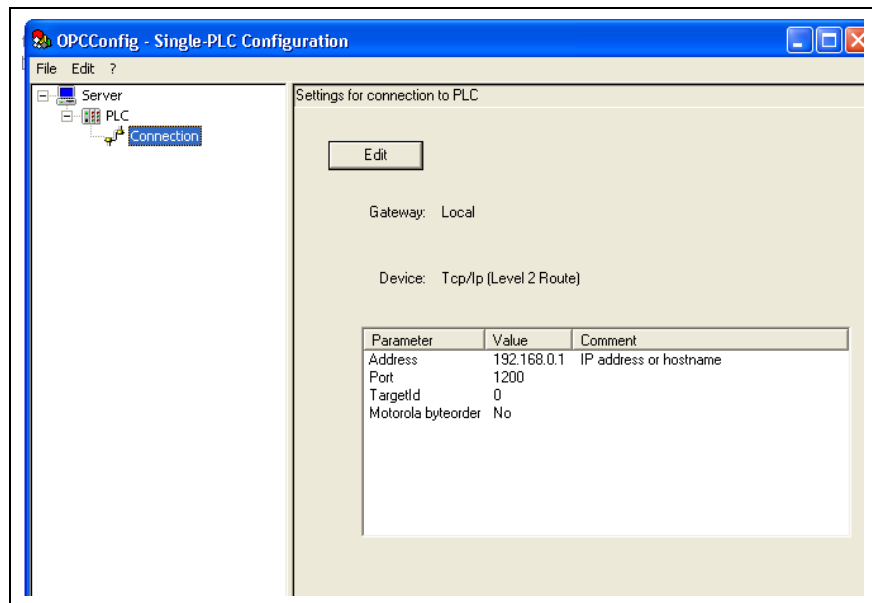


Fig. 7 Connection to PLC configuration

Specific settings for OPC gateway:

6. Select “Connection” settings
7. Click “Edit” button

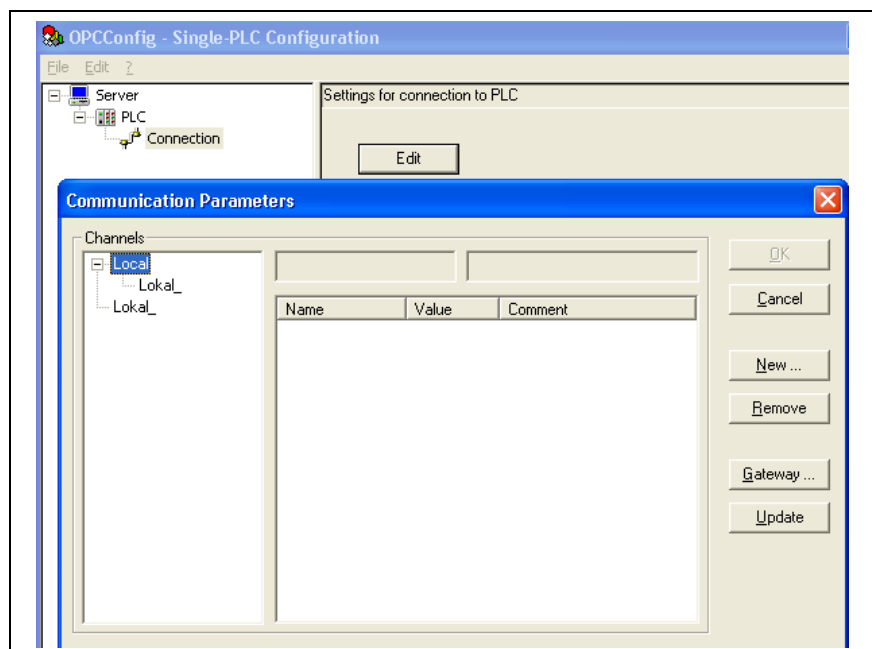


Fig. 8 OPC Gateway configuration

8. Click “New” button

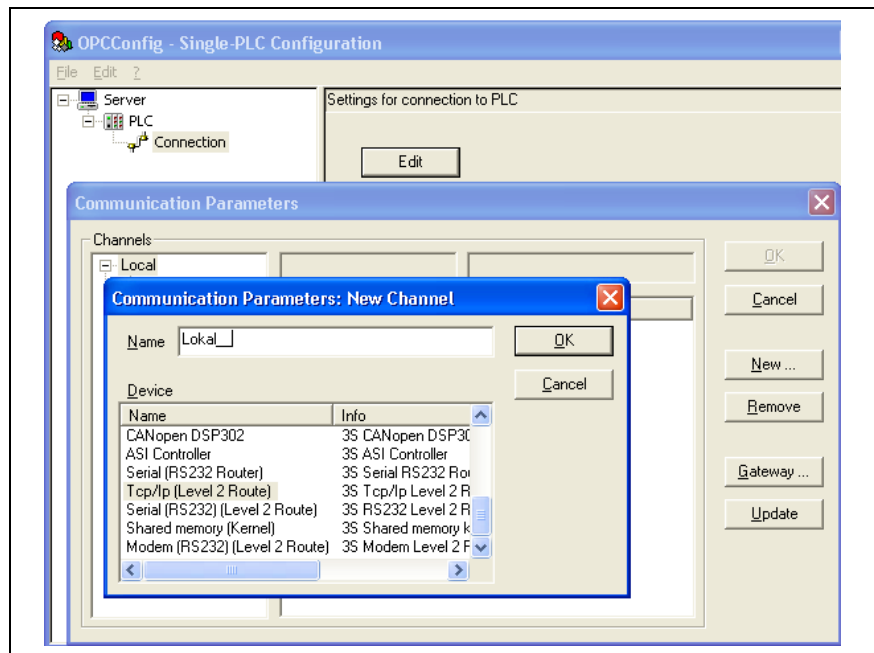


Fig. 9 OPC gateway channel configuration

9. Select "Tcp/Ip (Level 2 Route)" communication channel
10. Click "OK" button

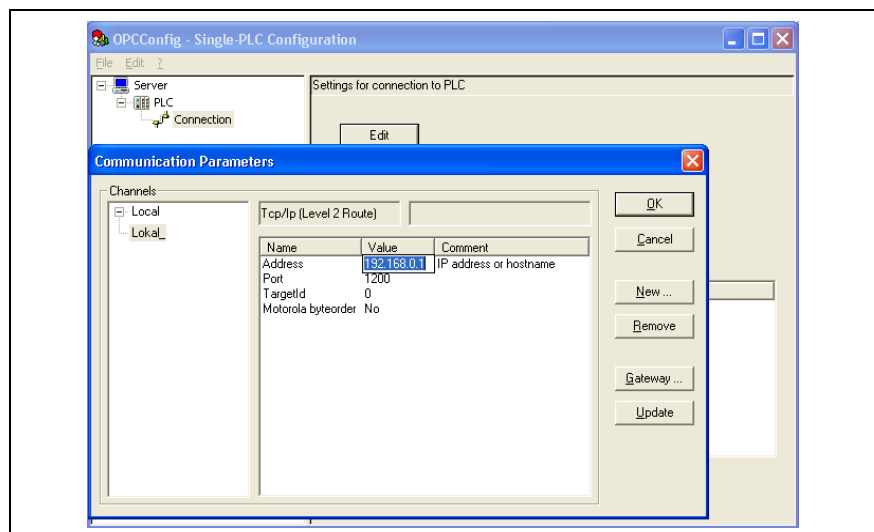


Fig. 10 Connection to PLC IP configuration

11. Double click on the IP address displayed in the "Value" column next to "Address".

**Note**

192.168.0.1 is a master controller IP address!

12. Set IP address, default: 192.168.0.1.

13. Confirm the settings by clicking on the “OK” button.  
The settings appear in the main window:

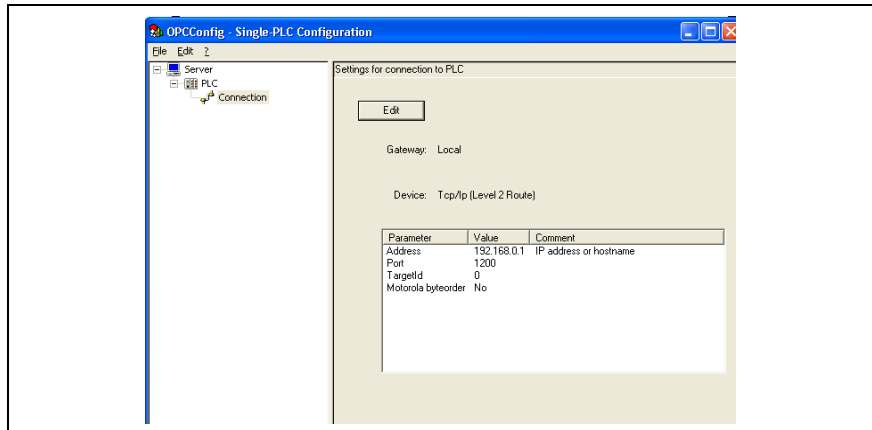


Fig. 11 PLC settings overview

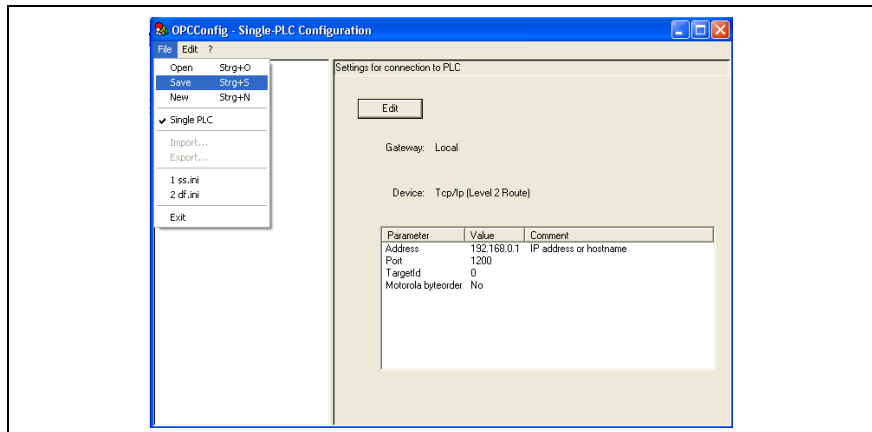


Fig. 12 Saving the OPC configuration

14. Upon completion, the configuration must be saved using the save function in the main menu.

15. Close the OPC configuration software.

16. Once all the settings have been made, the MTS for PPU-E application can be started.

If everything has been done correctly, you should not experience any problems when using the PPU-E software in connection with the PPU device.

**Note**

In the event that the MTS for PPU-E application cannot establish a connection with the hardware following start-up, the user should check the cable, IP address and all connection settings once again.

## 5 Motion Tool Schunk (MTS) for PPU-E application description

### 5.1 Main window

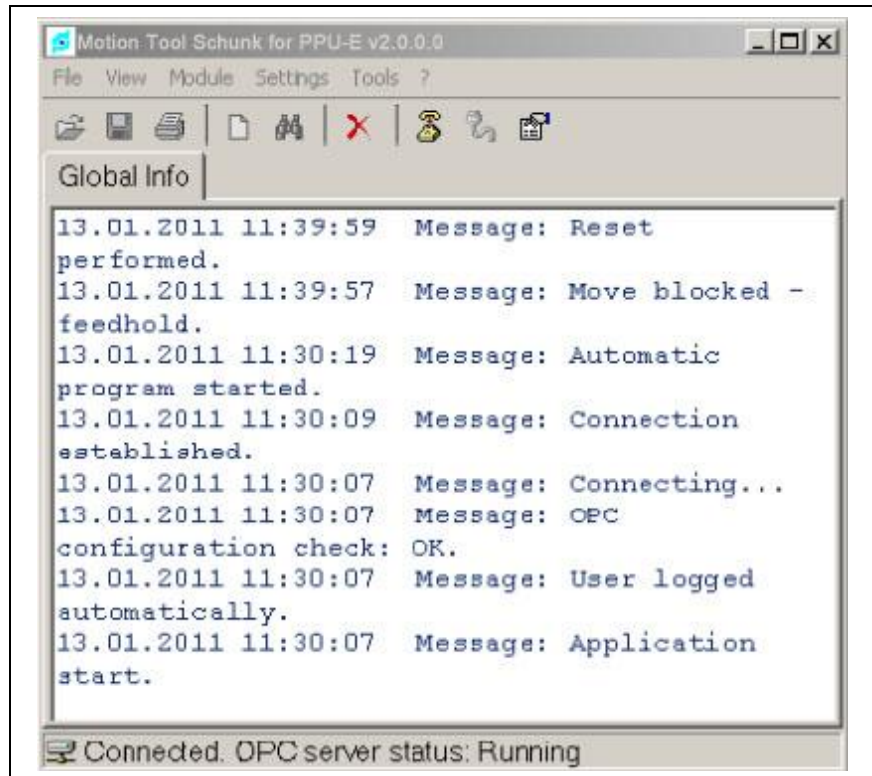


Fig. 13 Motion Tool Schunk for PPU-E main window

From the application's main window, the user has access to the application settings.

Additionally, all performed actions will be logged.

Connection can be switched on/off.

The main window comprises:

- Menu (see Chapter 5.1.1)
- Symbol list (see Chapter 5.1.2)
- Logger (see Chapter 5.1.3)
- Connection status bar (see Chapter 5.1.4)

### 5.1.1 Menu



*Fig. 14* Menu

- File**
- Save: not used.
  - Print: not used.
  - Exit: application will be closed.

- View**
- Bus Details: not used.
  - Set Bookmark: not used.
  - Clear Logs: logger contents will be cleared.

- Module**
- Open: PPU-E module window will be opened.
  - Close: PPU-E module window will be closed.
  - Stop: fast stop command will be sent to all modules.
  - Settings
  - Connect/Disconnect: switch connection on/off.
  - Open Communication: PC IP settings window will be opened. User can set PC IP address.
  - Language: select application language.
  - Preferences: application preferences window will be opened.

- Tools**
- CRC Calculator: not used.
  - Numeric Converter: not used.
  
  - Program Editor: program editor window will be opened. User can prepare the motion sequence for PPU-E unit.
  - ? Information about MTS for PPU-E application: version, vendor etc.

### 5.1.2 Symbol list



Fig. 15 Symbol list

With this symbol list, the user can easily access some application functions and options:










Symbol	Description
	Not used
	Not used
	Not used
	Not used
	Not used
	Fast stop command will be send to all modules
	Connect/Disconnect: switch connection on/off
	Not used
	Application preferences window will be opened

Table 3

### 5.1.3 Logger

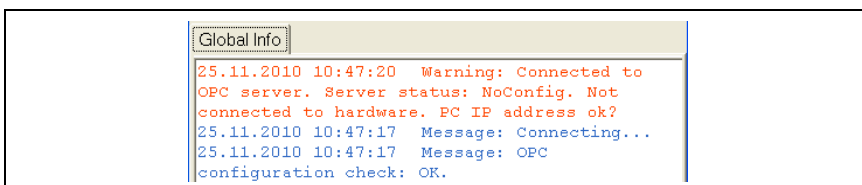


Fig. 16 Logger

All actions, preference changes and application messages are logged in the logger.

### 5.1.4 Connection status bar



Fig. 17 Connection status bar

The actual connection status between the application and hardware is always shown in the status bar.



## 5.2 Preferences window

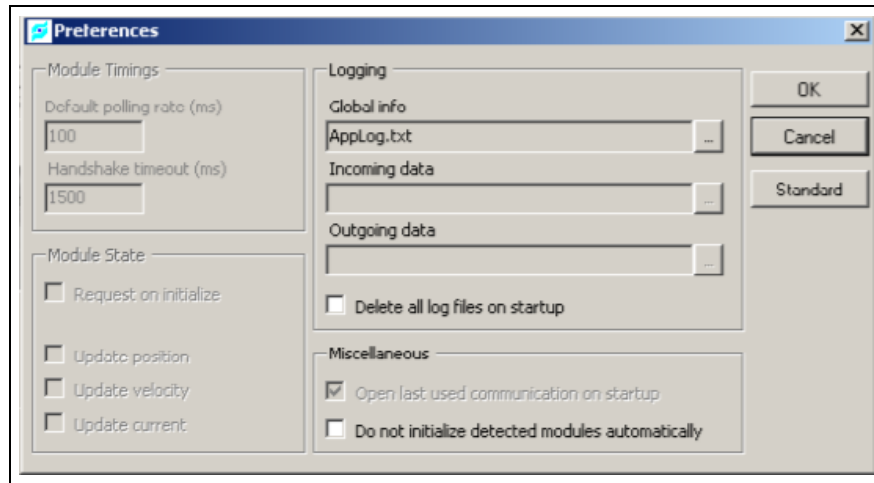


Fig. 18 Preferences

In the “Preferences” window, the user can set specific application settings. Whenever changes are made to these settings, the changes must be confirmed by clicking on the “OK” button.

- Module Timings – not editable.
- Module State – not editable.
- Logging
  - Global info: external file where all actions and application messages are logged.
  - Incoming data – not used.
  - Outgoing data – not used.
  - Delete all log files on start-up: when checked, external file (log file) will be initialized at every application start.
- Miscellaneous
  - Do not initialize detected modules automatically: when checked, detected hardware will not be initialized at application start.

### Note

All application preferences can be easily set to default values.

For this purpose, the “Standard” button can be used.

### 5.3 “Module: PPU-E” window

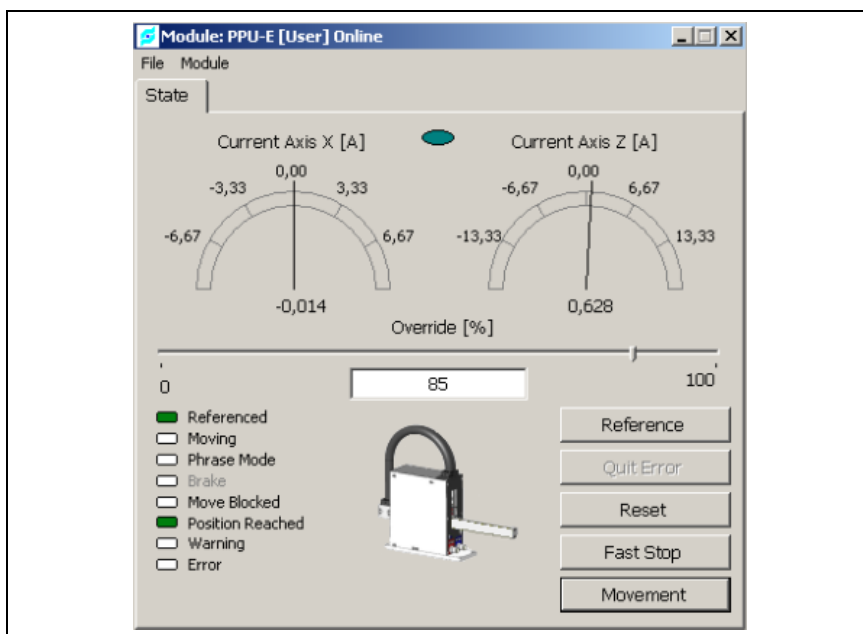


Fig. 19 Module: PPU-E window

The “Module: PPU-E” window shows the actual device (PPU-E) status. Additionally, main couple commands can be sent to the device. The device status is refreshed every 100 ms.

The window name contains the following information: “Module: PPU-E [ <User> ] <Status (Online/Offline)>”. If a connection between the application and hardware cannot be established, the status “Offline” is shown and the “Module: PPU-E” window is disabled.

By using the “Override” track bar (user can also directly enter an advisable override), the user can influence the travelling velocity of the axes. For example, the axis velocity is set to 3000 mm/s and the override is set to 50%.

The calculated target axis velocity is 1500 mm/s. In the event that the override is set to 0%, all axes are stopped.

Using the buttons located on the right side of the window, the most important commands can be sent to the device. The “Movement” button enlarges the window, enabling new important options and movement commands to be shown.

**Note**

“Movement” button is only enabled when the device is referenced!

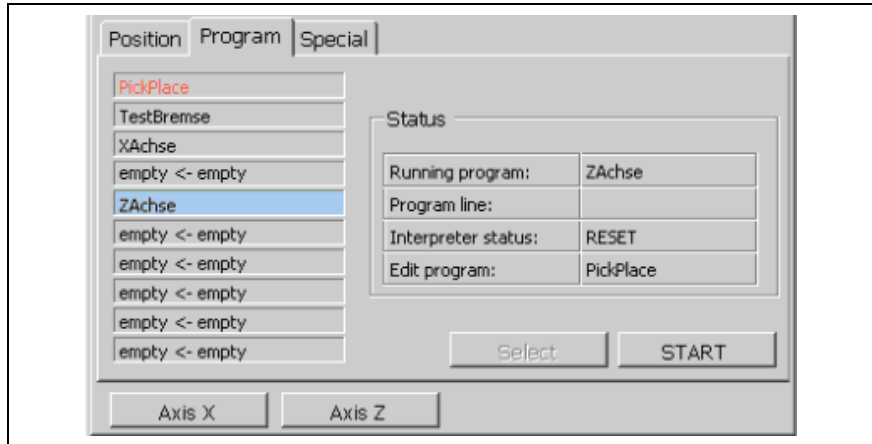


Fig. 20 "Movement" button enlarged window

- Tab “Position”,
- Tab “Program”,
- Tab “Gripper” – only active if device features a gripper,
- Tab “Special”,
- Buttons which give access to “Axis” windows.

**5.3.1 “Command” buttons**

The following “command” buttons are available:

- **Reference** – the reference sequence will be started. If the device was referenced earlier, every axis is only sent to the home position.

**Note**

The “Reference” button is only enabled in the device operational state – no error, power switched on. When a “whole” reference sequence is required, the user can select the "Force Reference" option from the “Module” menu. The device reference sequence will be performed.

- **Quit Error** – all errors, warnings and messages from all axes are acknowledged.
- **Reset** – errors, warnings and messages from all axes are acknowledged. Additionally, motion interpreter reset is performed.

- **Fast stop** – all axes are stopped immediately. Whole device movement is blocked. In order to unlock movement, a reset command must be sent to the device.
- **Movement** – window will be enlarged, additional options are shown.

### 5.3.2 Movement tabs

- Tab **“Position”** – here, the user has the possibility to move all axes to the selected position. All that is required is to type the desired position into the input fields and click on the “Move” button.

**Note**

The "Move" function is only active when the automatic program is not started.

This is equivalent to the motion interpreter status "Reset".

- Tab **“Program”** – the motion programs saved in the device memory are listed here. All in all, ten motion programs can be prepared and stored. Additionally, on the right side of the tab, the motion interpreter status is shown.

**Note**

The background of the program currently running (or selected) is blue.

The edit program font color is orange.

**⚠ WARNING**

**The PPU-E can cause serious bodily damage and personal injury!**

➔ Before actuating the “START” button, ensure that nothing is standing in the way of the PPU-E device.

Here, the user can start a selected motion program (“START” button), select another motion program for editing (by clicking on the program name), or select another motion program for automatic motion sequence (by clicking on the program name and then the “Select”

button). By default, the “Select” button is disabled for all “simple” users (see chapter “Application users”).

- **Tab “Gripper”** – gripper handling: open, close, release. Picture shows current status of gripper.

**Note**

The “Gripper” tab is only active if the device features a gripper.

The application automatically recognizes device configuration.

- ➔ **Tab “Special”** – by default, this tab can only be used by an administrator. Nevertheless, the hardware configuration can be changed here.

Axes buttons can be found in the lower section of the window. A window for the selected axis can be opened here.

## 5.4 Axis window

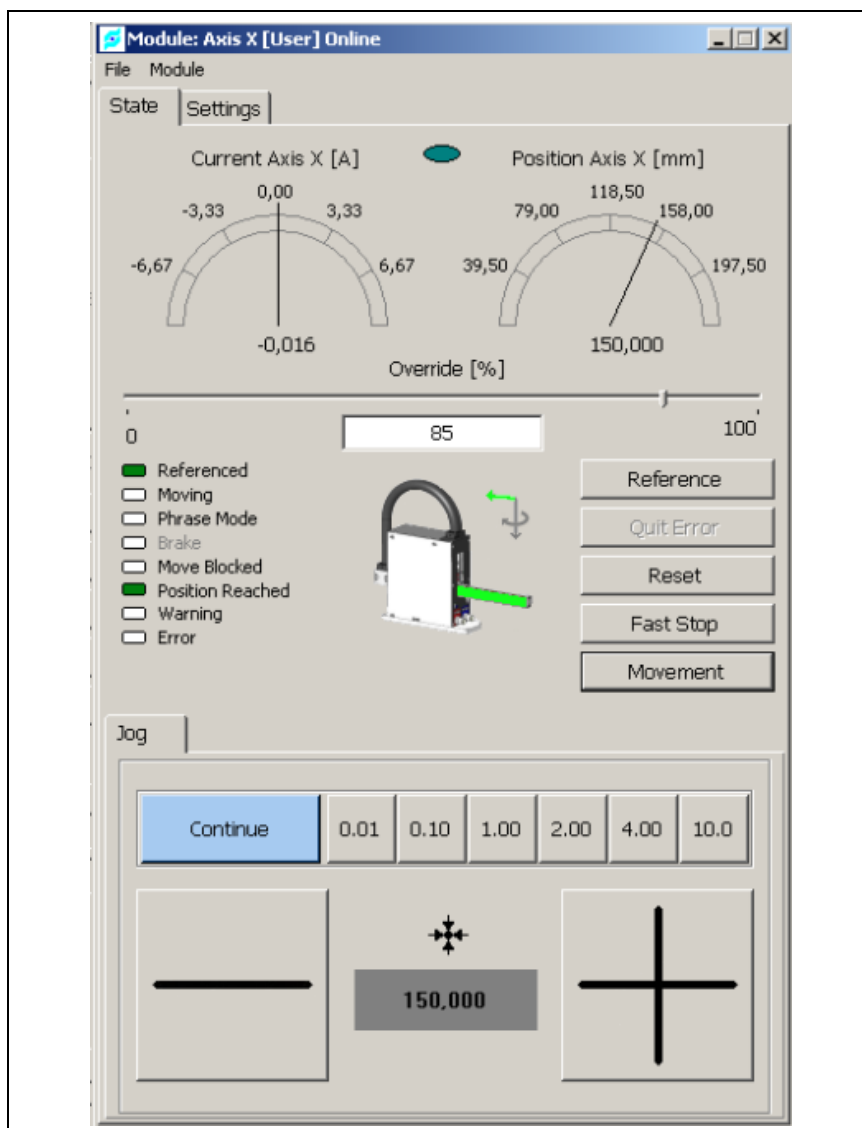


Fig. 21 Axis window

The Axis window describes a single axis of the device. It is similar in structure to the “Module: PPU-E” window.

Everything displayed in this window concerns a single axis – except for the “Override” and “Reset” command buttons.

“Override” in this case, and in all other application windows where “Override” can be set to desired value, is global.

This means that, whenever changes are made to the “Override”, these changes will be applied to the device as a whole and to all axes.

The Axis window contains two tabs:

- **Tab “State”** – overall axis status, also contains sub-tab “Jog”.
- **Tab “Settings”** – axis-specific settings.

In the “State” tab, the user can find the axis status, actual position and actual current.

Additionally – similar to other application windows – the tab contains couple command buttons.

#### 5.4.1 "Command" buttons

The following “command” buttons are available:

- **Reference** – the axis reference sequence will be started. If the device was referenced earlier, the axis is only sent to the home position.

##### **Note**

The “Reference” button is only enabled in the device operational state – no error, power switched on.

- **Quit Error** – all axis errors, warnings and messages are acknowledged.
- **Reset** – errors, warnings and messages from all axes are acknowledged. Additionally, motion interpreter reset is performed.
- **Fast stop** – all axes are stopped immediately. Whole device movement is blocked. In order to unlock movement, a reset command must be sent to the device.
- **Movement** – window will be enlarged, “jog” tab appears.

### 5.4.2 Sub-tab “Jog”

The “Jog” sub-tab helps the user with axis positioning.

Couple move step lengths are available:

- 0.01 mm
- 0.10 mm
- 1.00 mm
- 2.00 mm
- 4.00 mm
- 10.00 mm
- “Continue” – continues movement.

By using the “-” and “+” buttons and selecting a step length, the axis can be easily shifted to the requested position.

**Note**

Jog mode is only active when the automatic program is not started.

This is equivalent to the motion interpreter status "Reset".

### 5.4.3 Tab “Settings”

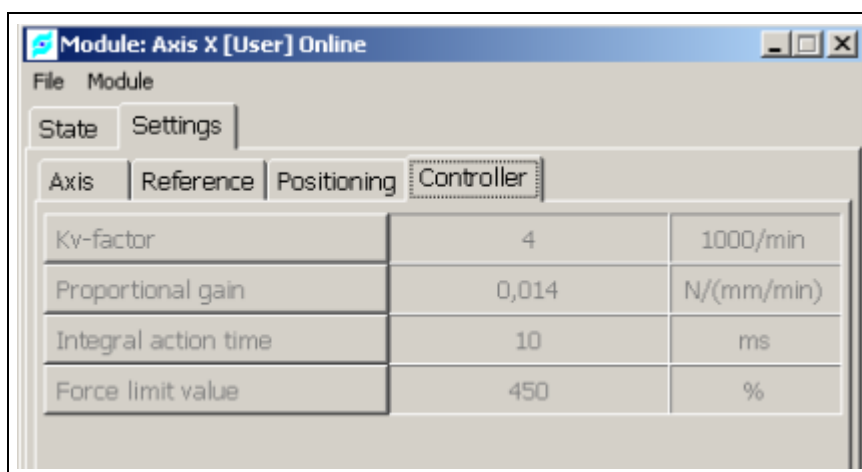


Fig. 22 Axis settings

Axis settings are divided into four groups:

- “Axis”
  - “Min. Position” – axis positioning negative limit [mm]
  - “Max. Position” – axis positioning positive limit [mm]
  -



- **“Reference”**
  - “Axis Home Position” – desired axis home position [mm]
  - “Direction of motion” – motion direction by referencing
  
- **“Positioning”**
  - “Jog Velocity Normal” – first (normal) jog velocity [mm/s]
  - “Jog Velocity High” – second (high) jog velocity [mm/s]
  - “Jog Acceleration Normal” – first (normal) jog acceleration [mm/s<sup>2</sup>]
  - “Jog Acceleration High” – second (high) jog acceleration [mm/s<sup>2</sup>]
  - “Jog Deceleration” – jog mode deceleration [mm/s<sup>2</sup>]
  - “Jog Time Velo. Change” – jog mode change time from first to second velocity/acceleration [ms]
  - “Max. Velocity” – axis highest velocity [mm/s]
  - “Max. Acceleration” – axis highest acceleration [mm/s<sup>2</sup>]
  
- **“Controller”**
  - “Kv-factor” – axis controller position loop factor [1000/min]
  - “Proportional gain” – axis controller velocity loop proportional gain [N/(mm/min)]
  - “Integral action time” – axis controller velocity loop integral action time [ms]

**Note**

Any axis setting change should be confirmed by clicking on the “Save Changes” button.

However, any setting change listed in the “Controller” tab is immediately sent to the controller – in this case, it is not necessary to save the changes.

Not all axis setting tabs are enabled for “simple” users. “Supervisor” and “Administrator” users have access to all axis settings.

## 5.5 Program editor window

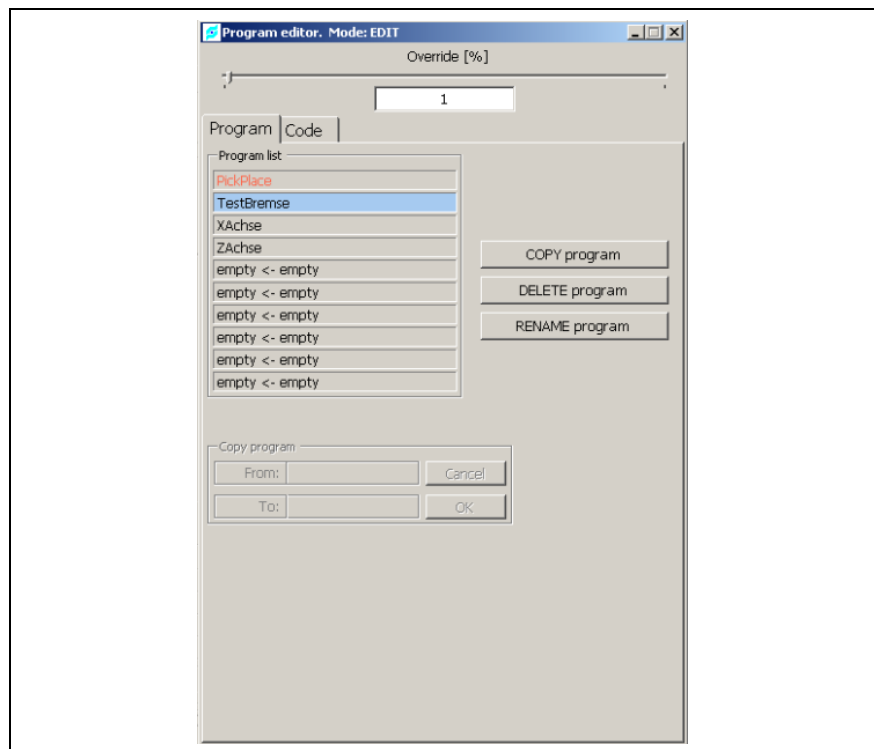


Fig. 23 Automatic program editor

The automatic program editor enables the user to prepare his own “pick and place” sequence.

The program editor comprises two tabs:

- Tab “Program”
- Tab “Code”.

### 5.5.1 Tab “Program”

Currently memorized sequences are listed in this tab. A maximum of 10 sequences can be memorized simultaneously. The selected edit program font color is orange.

The selected program can be:

- copied – copy button enables the “Copy program” object group. Target memory position should be selected. The “OK” button confirms the copy operation.
- deleted – use the “Delete program” button.

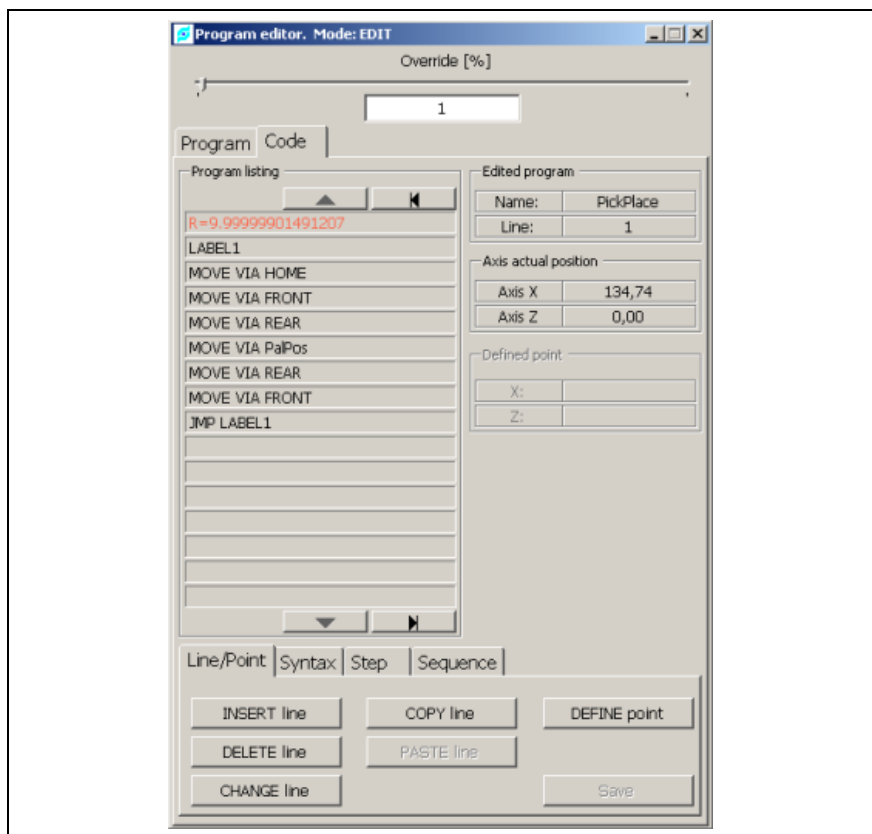
**Note**

The program selected for the automatic sequence (background: blue) cannot be deleted.

The master controller’s program memory storage area is locked.

- renamed - the “Rename program” button is used for this purpose. The new (renamed) program name cannot be identical to any other program name currently stored in the memory.





**5.5.2 Tab “Code”**



*Fig. 24* Program editor, “Code” tab

The “Code” tab contains the following elements:

- selected program listing, where selected program line/command font color is orange,
- edited program information,
- actual positions of the device axes,
- defined point – enabled if the selected program line contains any,

- edit/check program tabs: “Line/Point”, “Syntax”, “Step”, “Sequence”,
- program navigation buttons:
  -   : selected line up/down,
  -   : go to first/last program line/command.

#### 4.5.2.1 Edit/check program tabs

##### Tab Line/Point

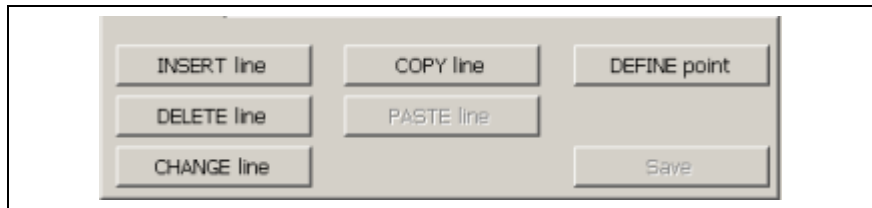


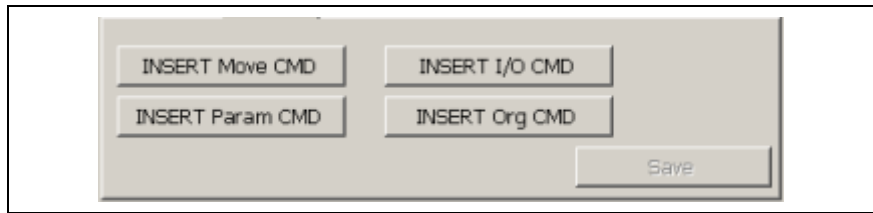
Fig. 25 Tab Line/Point

This tab allows the user to do a number of different things with the selected program. First of all, a new empty program line can be inserted (button: INSERT line). A selected program line can also be deleted (button: DELETE line) or changed (button: CHANGE line). Changes can only be made to parameters, destination points or destination positions – commands stay unchanged. A program line can also be copied (button: COPY line) and pasted (button: PASTE line) into a desired program position. Additionally, program “points” can be defined. Every program user has the possibility to determine a maximum of 50 points.

A point is nothing less than a device position (a point consists of the position of every device axis).

Points can then be used in a program in place of the axis direct positions.

Upon completion, all changes can be saved (button: Save).

**Tab Syntax**

*Fig. 26* Tab Syntax

Each of the four buttons shown in the figure above represents a command group:

- INSERT Move CMD button represents the move command family, e.g.: MOVE VIA,
- INSERT Param CMD button represents the parameter command family, e.g.: R,
- INSERT I/O CMD button represents the input/output command family, e.g.: WAIT INPUT,
- INSERT Org CMD button represents the organization command family, e.g.: LABEL.

It is possible to insert a new command by clicking on some of these command family buttons.

In order to insert a new command, the user should firstly click on the representative of the desired command family (command family button), and then act upon the subsequent instructions.

For example, if the command: MOVE should be inserted into the program:

- click on button: INSERT Move CMD,
- select the command: MOVE from the move command family,
- select the movement destination point from the list.

All available commands are described in detail at the end of this manual (see Chapter "Appendix: Programming commands").

**Tab Step**



*Fig. 27 Tab Step*

Button STEP+ enables the prepared sequence to be checked step by step. Such verification helps to avoid unintentional device behavior and minimizes the possibility of crashes.

Edit mode is automatically changed to “Step” mode. Every click of the “STEP+” button sends a trigger to the device and the next command is executed. The command currently being performed is highlighted – the font color is orange.

**Tab Sequence**



*Fig. 28 Tab Sequence*

The START button enables the prepared sequence to be checked immediately. Edit mode is automatically changed to “Sequence” mode. The command currently being performed is highlighted – the font color is orange.

**Note**

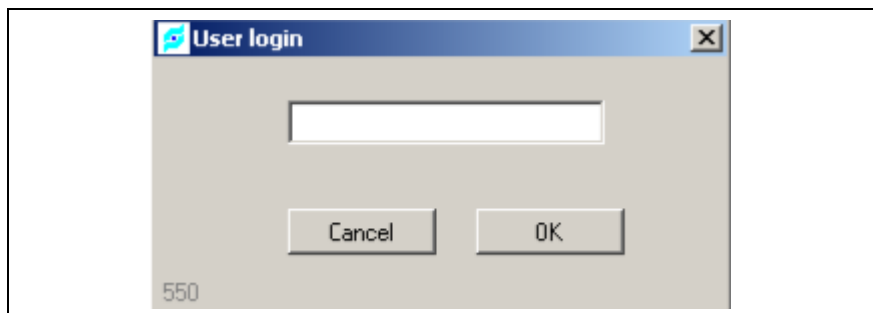
The automatic program cannot be started (application window: Module: PPU-E, START button) if the device is in the “Step” or “Sequence” edit mode. After testing, the user should always remember to reset the device (Reset button on the Step and Sequence tabs).

## 5.6 Application users

There are currently three user access levels available:

- **User** – no password required; logged in by default on start-up of the application; so-called "low rights user"; many application functions and options are not available for this type of user.
- **Supervisor** – password protected; almost all application functions and options are unlocked for this type of user; default supervisor password: "sv".
- **Administrator** – password protected, dynamic password created based on a random number, access to all application functions and options.

The application user can log in at any time. From almost every application window, the user can access the "Change User" login form, shortcut "F6".



*Fig. 29* User login form

The user logs in by typing the password in the box shown above:

User level – no password (empty),

Supervisor level – "sv".

## 5.7 Application languages

- The current MTS for PPU-E application supports two languages: German and English.
- The application language can be changed at any time. However, changes only take effect after the application has been restarted.

The user can change the language via the “Settings/Language” menu which can be accessed from the main application window.

## 5.8 External logger

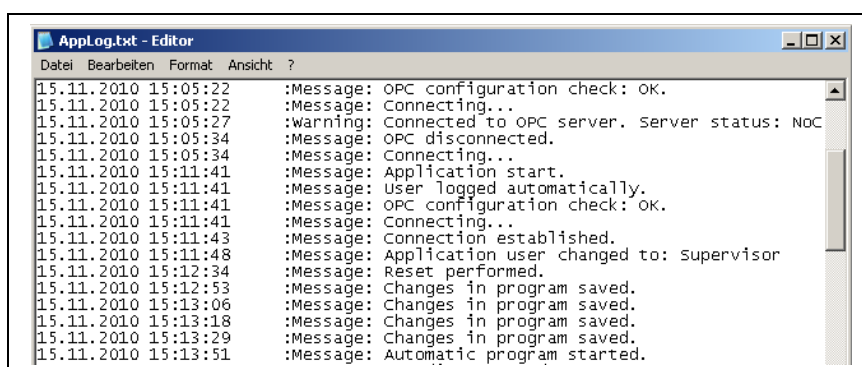


Fig. 30 External logger

All user activities are logged in the application's internal logger (Fig. 16, page 16). This information is also logged in the “external” logger in the form of a .txt file. The external logger file can be found in the application folder.



## 6 Appendix

### 6.1 Programming commands

The following commands description is a copy taken from:

*Rexroth IndraMotion for Handling 02VRS*

*Operating and Programming Instructions*

*R911318055*

*Edition 02*

#### 6.1.1 Motion Commands

The following definitions apply to all commands of the "move" group:

The Tool Center Point (called TCP below) moves from the current position to the target position in the form of a PTP motion (PTP -> point-to-point).

When the various axis velocities are calculated, the maximum velocities and accelerations of the individual axes as well as the velocity override value are taken into consideration.

The PTP motion can be optimized in the following two ways through the parameter command "INTERPOLATION":

- "PTP": time-optimized
- "LINEAR": path-optimized

The following applies for either case:

If the motion is a PTP motion, all axes involved in the motion process start at the same time and also reach their target position at the same time.

Related to the particular maximum values, the axis velocities and accelerations can be weighted by percent through V\_PTP and A.

---

<b>MOVE</b>	<p>Example: MOVE HOME</p> <p>The TCP moves from its current position to the addressed target position (HOME) in a PTP motion.</p>
<b>MOVE_REL</b>	<p>Example: MOVE_REL HOME</p> <p>The target position is calculated by a vector addition of the current position and the addressed point (e.g. HOME). The motion is a PTP motion.</p>
<b>MOVE TIL</b>	<p>Example: MOVE HOME TIL GR_CLOSE</p> <p>The TCP moves from its current position to the target position (e.g. HOME, PTP motion).</p> <p>As soon as the programmed input (e.g. GR_CLOSE) changes to High, the traversing movement is cancelled and a stepping to the next block is executed.</p> <p>If the stop condition is not fulfilled during the traversing motion, the block will be carried out as a normal absolute traversing motion with following block advance.</p> <p>If the stop condition has already been fulfilled before the motion is started, the next block advance will follow immediately.</p>
<b>MOVE_REL TIL</b>	<p>Example: MOVE_REL HOME TIL GR_CLOSE</p> <p>The target position (PTP motion) is calculated through a vector addition of the current position and the addressed point (HOME). As soon as the addressed input (e.g. GR_CLOSE) changes to High, the traversing movement is cancelled.</p> <p>A block advance follows as soon as all axes are stopped.</p> <p>If the stop condition is not fulfilled during the traversing motion, the block will be carried out as a normal relative traversing motion with following block advance.</p> <p>If the stop condition has already been fulfilled before the motion is started, the next block advance will follow immediately.</p>
<b>MOVE VIA</b>	<p>Example: MOVE VIA HOME</p>

The TCP moves from its current position to the target position (HOME) in a PTP motion.

The block advance follows as soon as the axes have approached the end position by the rounding distance.

**MOVE\_REL VIA**

Example: MOVE\_REL VIA HOME

The target position is calculated by a vector addition of the current position and the programmed point (e.g. HOME). The motion is a PTP motion. The block advance follows as soon as the axes have approached the end position by the rounding distance.

**MOVE Ax=**

Example: MOVE X=1000.0

The selected axis traverses to the entered position (e.g. X axis on 1000.0) in an absolute manner.

**MOVE VIA Ax=**

Example: MOVE VIA X=1000.0

The selected axis (X-axis) makes a rounding movement across the entered absolute position (1000.0). The block advance follows as soon as the axis has approached the end position by the rounding distance.

**MOVE Ax=Point**

Example: MOVE X=HOME

The selected axis makes an absolute movement to the entered point coordinate.

**MOVE VIA Ax=P**

Example: MOVE VIA X=HOME

The selected axis makes a rounding movement across the entered point coordinate.

The block advance follows as soon as the axis has approached the end position by the rounding distance.

### 6.1.2 Parameter Commands

Parameter accesses have a modal effect. That means that a parameter, once programmed, affects all following blocks until it is reprogrammed. Any program abortion, reset or program end resets all active parameter registers (except real and integer variables) of the interpreter.

**V\_PTP=** Example: V\_PTP=30

V\_PTP allows velocity weighting by percent in the sequence. This reduction is based on the maximum values defined during configuration.

**A=** Example: A=12

On principle, the maximum accelerations defined in machine configuration are used for all traversing blocks. This parameter command also allows a reduction by percent of all accelerations. This reduction is based on the maximum values defined during configuration.

**R=** Example: R=25

The rounding distance (in mm) controls the block advance for rounding movements.

As soon as the distance to be traversed is shorter than the rounding distance, the next motion command will be executed.

**REAL\_VAR\_** Example: REAL\_VAR\_12=58.2

This parameter can be used to describe the variables managed in a global field of real values. A real value is assigned to a real variable (e.g. 12), in this case 58.2. The real values are saved to a global field. That means that only one field is available for all programs.

**INT\_VAR\_**

Example: INT\_VAR\_12=12

This parameter can be used to describe the variables managed in a global field of integer values. An integer value is assigned to an integer variable (e.g. 12), in this case 12. The integer values are saved in a global field. That means that only one field is available for all programs.

**INC INT\_VAR\_**

Example: INC INT\_VAR\_12

The value of the integer variable is incremented by one.

**DEC INT\_VAR\_**

Example: DEC INT\_VAR\_12

The value of the integer variable is decremented by one.

**INTERPOLATION**

Example: INT= LINEAR or INT= PTP

This command involves the PTP motion.

PTP: Execution of the PTP motion is time-optimized.

LINEAR: That means that, although all axes involved start and complete the movement at the same time, they differ in their acceleration phases.

As a result, shortest point-to-point traversing times can be achieved.

Execution of the PTP motion is path-optimized.

That means that all axes involved start and complete the movement at the same time. And they have the same acceleration phases.

As a result, shortest point-to-point distances can be achieved.

**OVERRIDE**

Example: OVERRIDE ON or OVERRIDE OFF

This command activates and deactivates the velocity override set through the user interface.

### 6.1.3 I/O Commands

#### OUTPUT

Example: Gripper=0

The output (gripper) can assume a value of zero or one. This allows switching of peripheral equipment.

#### WAIT INPUT

Example: WAIT GR\_CLOSE=1

The input (GR\_CLOSE) is polled for a specific value (here: 1). The block is advanced as soon as the condition is fulfilled. This synchronizes the program with the peripheral equipment. The program enters the "WAIT" state until the input is high (1).

#### OUTPUT ON PATH

Example: X=12.58 Gripper=1

This command allows precise positions to be achieved during the movement. This command always affects the following traversing block only. Position-related switching requires consideration of the defined axis (here: X) only. Once this axis has reached the defined switching position (12.58), the defined output is switched (here: to 1). The output can, of course, also be set to zero (gripper = 0).

### 6.1.4 Program Organization Commands

#### WAIT

Example: WAIT 12.88

This command defines a waiting time in seconds. Program processing is stopped for that time.

#### COMMENT

Example: (\*this is a comment\*)

This command allows insertion of comment lines in the program. It does not affect processing of the program.

The comment text may comprise 17 characters. For identification purposes, the entered command is automatically supplemented by insertion characters (\*comment text\*).

**CALL SUBROUT**

Example: CALL\_SUB subGetTool

This command calls a local subroutine. After the subroutine has been processed, processing of the program is continued with the following program line.

**CALL SUBPRG**

Example: CALL\_PROG Sequence\_1

This command calls another program, which must be loaded to the main memory of the control (referred to as global subroutine below). After the global subroutine has been processed, processing of the program is continued with the following program line of the calling program.

A global subroutine must not contain any further global subroutine calls nor any local subroutines.

**END PROGRAM**

Example: END\_PROGRAM

This command stands for the end of the programs. When this command line is reached, processing of the program is completed.

If local subroutines are used, this command is mandatory, because local subroutines can only be programmed after an END\_RPROGRAM command.

**BEGIN SUB**

Example: BEGIN subGetTool

This command stands for the start of a local subroutine in a program.

Local subroutines can only be programmed after an END\_PROGRAM command.

A local subroutine must not contain any global subroutine calls nor any further local subroutines.

**END SUB**

Example: END\_SUB

This command stands for the end of a local subroutine in a program.

When this command line is reached, processing of the local subroutine is completed and processing of the program is continued after the subroutine is called.

---

<b>IF INPUT THEN</b>	Example: IF GR_CLOSE=1 JMP LABEL12  If the input specified (GR_CLOSE) is equal to the value (here: 1), processing of the program jumps to a defined label. If the condition fails to be fulfilled, processing is continued with the next command in the sequence.
<b>LABEL</b>	Example: LABEL12  This command defines a label serving as a branch location in the program.
<b>JMP LABEL</b>	Example: JMP LABEL12  Processing of the program jumps to the label specified.
<b>IF INT VAR =</b>	Example: IF INT12=23 LABEL13  If the programmed condition is fulfilled, processing jumps to the defined label.
<b>IF INT VAR &gt;</b>	Example: IF INT12>23 LABEL13  If the programmed condition is fulfilled, processing jumps to the defined label.
<b>IF INT VAR &lt;</b>	Example: IF INT12<23 LABEL13  If the programmed condition is fulfilled, processing jumps to the defined label.
<b>IF Ax_Pos &gt;</b>	Example: IF X>12.90 LABEL12  If the programmed condition is fulfilled, processing jumps to the defined label.
<b>IF Ax_Pos &lt;</b>	Example: IF X<12.90 Label12  If the programmed condition is fulfilled, processing jumps to the defined label.