

Motion Tool Schunk (MTS) für Elektrische 'Pick and Place' Einheit PPU-E

Software Anleitung



Impressum:**Urheberrecht:**

Die vorliegende Anleitung bleibt urheberrechtlich Eigentum der SCHUNK GmbH & Co. KG. Sie wird nur unseren Kunden und den Betreibern unserer Produkte mitgeliefert und ist Bestandteil des Moduls. Ohne unsere ausdrückliche Genehmigung dürfen diese Unterlagen weder vervielfältigt noch dritten Personen, insbesondere Wettbewerbsfirmen, zugänglich gemacht werden.

Technische Änderungen:

Änderungen im Sinne technischer Verbesserungen sind uns vorbehalten.

Auflage: 09.01 / 02.05.2012 / de

© SCHUNK GmbH & Co. KG, Lauffen/Neckar

Alle Rechte vorbehalten

Sehr geehrter Kunde,

wir gratulieren zu Ihrer Entscheidung für SCHUNK. Damit haben Sie sich für höchste Präzision, hervorragende Qualität und besten Service entschieden.

Sie erhöhen die Prozesssicherheit in Ihrer Fertigung und erzielen beste Bearbeitungsergebnisse – für die Zufriedenheit Ihrer Kunden.

SCHUNK-Produkte werden Sie begeistern.

Unsere detaillierte Montage- und Betriebsanleitung unterstützt Sie dabei.

Sie haben Fragen? Wir sind auch nach Ihrem Kauf jederzeit für Sie da. Sie erreichen uns unter den aufgeführten Kontaktadressen im letzten Kapitel dieser Anleitung.

Mit freundlichen Grüßen

Ihre SCHUNK GmbH & Co. KG
Spann- und Greiftechnik
Bahnhofstr. 106 – 134
D-74348 Lauffen/Neckar

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



Inhaltsverzeichnis

1	Zu dieser Anleitung	5
1.1	Zweck/Gültigkeit	5
1.2	Zielgruppen	5
1.3	Symbole in diesem Handbuch.....	5
2	Anforderungen für OPC-Server und -Gateway.....	6
3	Installation von MTS	7
4	Vor der Erstverwendung – Verbindungseinstellungen	8
4.1	IP-Einstellungen des PCs des Benutzers.....	8
4.2	Anschlusskabel PC <-> Master-Controller	8
4.3	Einstellungen für OPC-Server und -Gateway.....	9
4.4	korrekten Projektnamen festlegen.....	10
5	Beschreibung der Anwendung MTS für PPU-E.....	14
5.1	Hauptfenster.....	14
5.1.1	Menü	15
5.1.2	Symbolleiste	16
5.1.3	Protokollfunktion	16
5.1.4	Statusleiste für Verbindung.....	16
5.2	Vorgabenfenster.....	17
5.3	Fenster „Module: PPU-E“	18
5.3.1	Befehlsschaltflächen.....	19
5.3.2	Registerkarten für Bewegung	20
5.4	Achsenfenster	22
5.4.1	Befehlsschaltflächen.....	23
5.4.2	Unterregisterkarte „Jog“ (Manuelle Steuerung)	24
5.4.3	Registerkarte „Settings“ (Einstellungen)	24
5.5	Fenster „Program editor“ (Programmeditor)	26
5.5.1	Registerkarte „Program“	26
5.5.2	Registerkarte „Code“	27
5.6	Anwendungsbenutzer	31
5.7	Anwendungssprachen.....	32
5.8	Externe Protokollfunktion	32

6	Anhang	33
6.1	Programmierbefehle.....	33
6.1.1	Bewegungsbefehle.....	33
6.1.2	Parameterbefehle	36
6.1.3	E/A-Befehle	38
6.1.4	Programmorganisationsbefehle.....	38

Mitgeltende Unterlagen

- Handbuch und Hinweise für MTS Software für PPU-E
- Montage- und Betriebsanleitungen für Lineardirektantriebe
- Allgemeine Geschäftsbedingungen (AGB)

Die oben genannten Unterlagen können unter:
www.de.schunk.com heruntergeladen werden.

1 Zu dieser Anleitung

1.1 Zweck/Gültigkeit

Diese Anleitung ist Bestandteil der MTS Software für PPU-E und beschreibt, wie eine elektrisch angetriebene PPU-E30-Einheit sicher und ordnungsgemäß gestartet wird.

Diese Anleitung ist nur für das auf dem Deckblatt angegebene Produkt gültig.

1.2 Zielgruppen

Zielgruppe	Aufgabe
Hersteller, Bediener	<ul style="list-style-type: none"> ➔ Dieses Handbuch muss für das Personal jederzeit zugänglich sein. ➔ Personal zum Lesen und Beachten dieser Anleitung und der mitgeltenden Unterlagen anhalten, insbesondere der Sicherheitshinweise und Warnhinweise.
Fachkundiges Personal, Monteure	<ul style="list-style-type: none"> ➔ Hinweise in diesem Handbuch und in den entsprechenden Dokumenten lesen und beachten, insbesondere Sicherheitshinweise und Warnungen.

Tabelle 1

1.3 Symbole in diesem Handbuch

Um einen schnellen Zugriff auf Informationen zu ermöglichen, werden folgende Symbole in diesem Handbuch verwendet:


Symbol	Bedeutung
 WARNUNG	Gefahren für Personen. Nichtbeachtung führt zum Tod oder schweren Verletzungen.
➔	Handlungsanleitung, auch Maßnahmen in einem Warnhinweis oder Hinweis.
1. 2. 3. ...	Schrittweise Handlungsanweisung. ➔ Reihenfolge beachten.

Tabelle 2

2 Anforderungen für OPC-Server und -Gateway

MTS für PPU-E dient als Testsoftware. Diese Software ermöglicht dem Benutzer die einfache Vorbereitung des Bewegungsablaufs für das PPU-E-Modul.

Anforderungen an den PC des Benutzers:

- Windows XP (mit Service-Pack 3)
- Netzwerkverbindung
- 30 MB freier Speicherplatz auf der Festplatte
- Mindestens 256 MB Arbeitsspeicher.

3 Installation von MTS

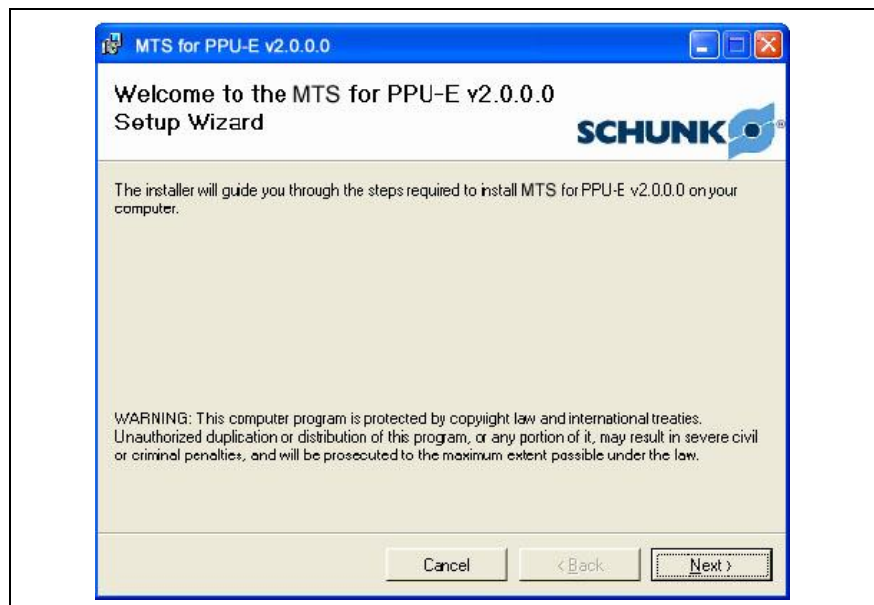


Abb. 1 Installationsprogramm für MTS

Der Benutzer muss zur Installation von MTS das mitgelieferte Installationspaket verwenden:

- MTS for PPU-E Install EN vx.x.x.x (englische Version)
oder
- MTS für PPU-E Install DE vx.x.x.x (deutsche Version)

➔ Das Installationsprogramm installiert MTS für PPU-E Version x.x.x.x und CoDeSys OPC Server Version 2.3.13.4

➔ Nach der Installation muss der PC **neu gestartet werden!**

Hinweis

Wenn die Installation nicht ausgeführt werden kann, muss der PC des Benutzers mit dem Netzwerk (Internet) verbunden werden.

Das Installationsprogramm prüft die *.net*-Version, die auf dem PC des Benutzers installiert ist. Wenn keine geeignete *.net*-Version (Version 2.0.50727 oder höher) gefunden wird, wird *.net* heruntergeladen und direkt vom Netzwerk installiert!

4 Vor der Erstverwendung – Verbindungseinstellungen

4.1 IP-Einstellungen des PCs des Benutzers

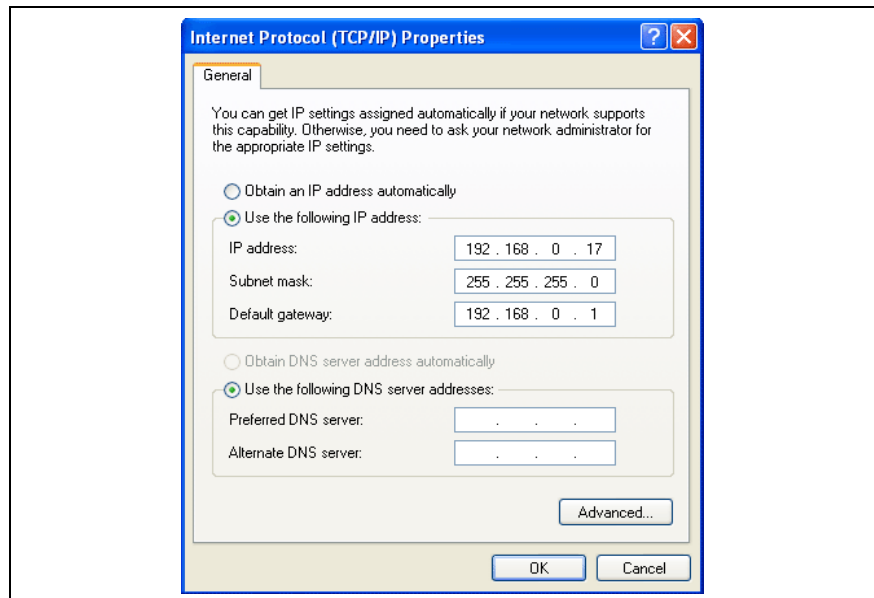


Abb. 2 IP-Einstellungen des Benutzers

Nach der Softwareinstallation muss die IP-Adresse des PCs des Benutzers auf den ordnungsgemäßen Wert gesetzt werden.

IP-Standardadresse des Master-Controllers: **192.168.0.1**

Die IP-Adressklasse des PCs des Benutzers muss der des Master-Controllers entsprechen (in diesem Fall Adressklasse **C**). Beispiel: 192.168.0.17

4.2 Anschlusskabel PC <-> Master-Controller

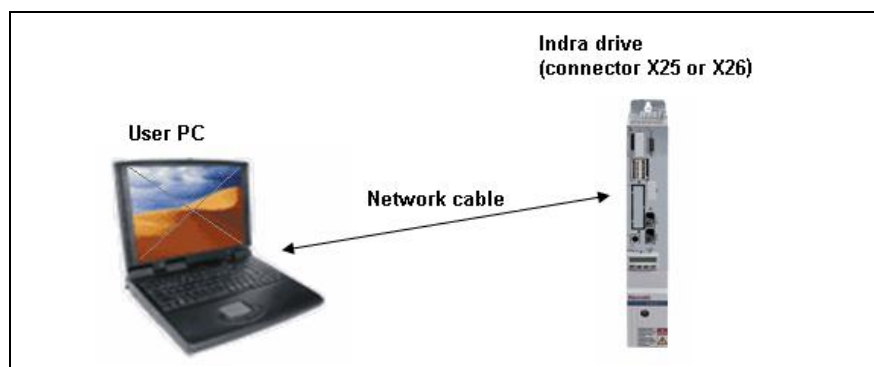


Abb. 3 PC des Benutzers <-> Anschluss an Master-Controller

Der PC des Benutzers muss mit einem einfachen Netzwerkkabel an den Master-Controller angeschlossen werden.

4.3 Einstellungen für OPC-Server und -Gateway

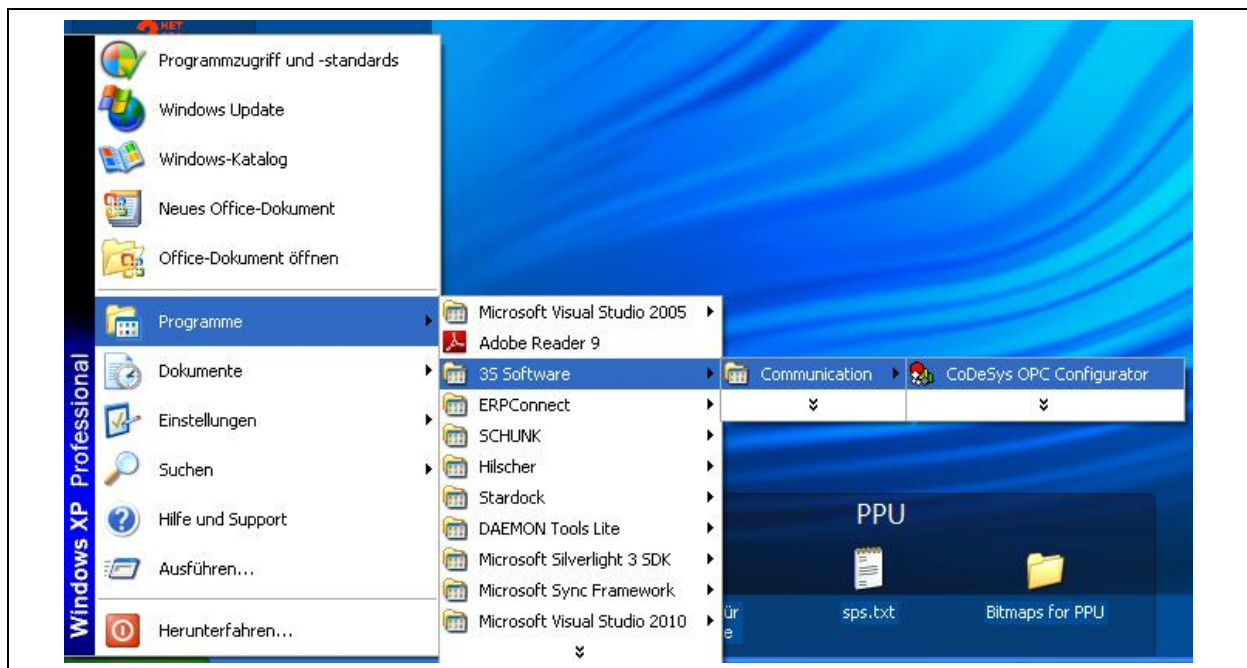


Abb. 4 OPC-Konfigurationsssoftware

Zur Herstellung des Anschlusses zwischen dem Master-Controller (PLC) und dem PC des Benutzers müssen der OPC-Server und das OPC-Gateway (einmalig) konfiguriert werden.

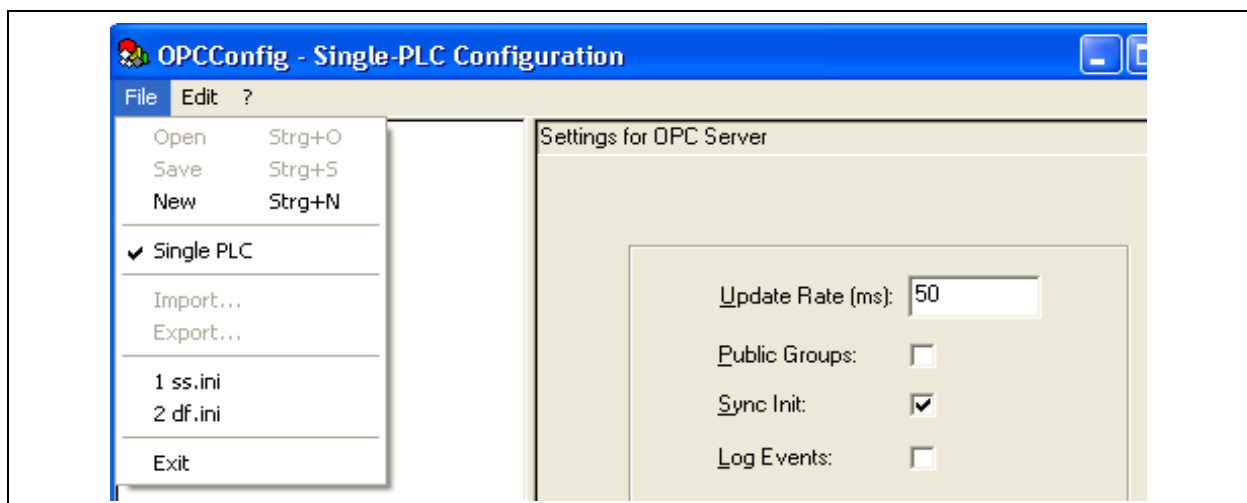


Abb. 5 OPC-Serverkonfiguration

1. „Single PLC“ festlegen
2. „Update rate“ (Aktualisierungsrate) auf 50 ms für OPC-Server setzen.
3. „Sync init“ für OPC-Server

4.4 korrekten Projektnamen festlegen

Standardwert:

PPU_E30_Koffersteuerung_v01_MLD001.pro

Hinweis

Dieser Name entspricht auch dem Namen des IndraWorks-Projekts im Master-Controller!

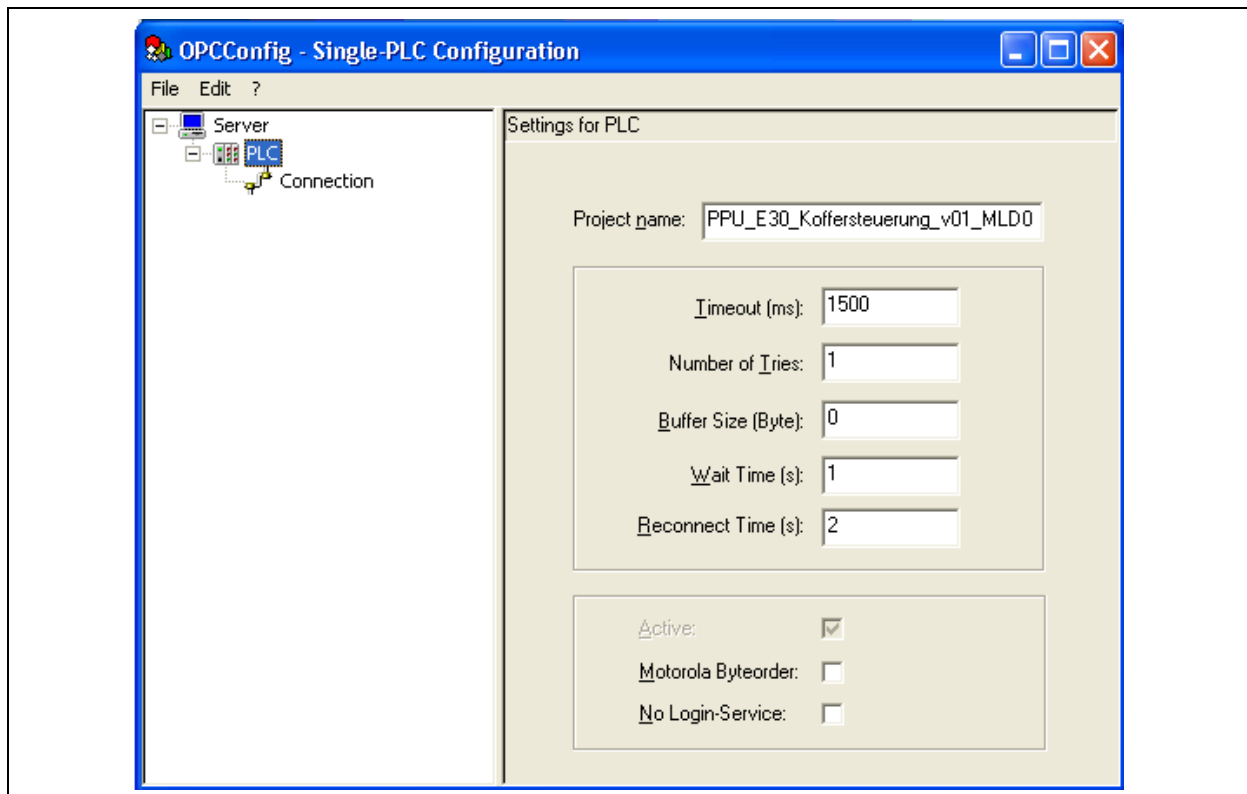


Abb. 6 PLC-Konfiguration

1. „Timeout“ (Zeitüberschreitung) auf 1500 ms setzen.
2. „Number of tries“ (Anzahl Versuche) auf 1 Mal setzen.
3. „Buffer size“ (Puffergröße) auf 0 Byte setzen.
4. „Wait time“ (Wartezeit) auf 1 s setzen
5. „Reconnect time“ (Wiederverbindungszeit) auf 2 s setzen.

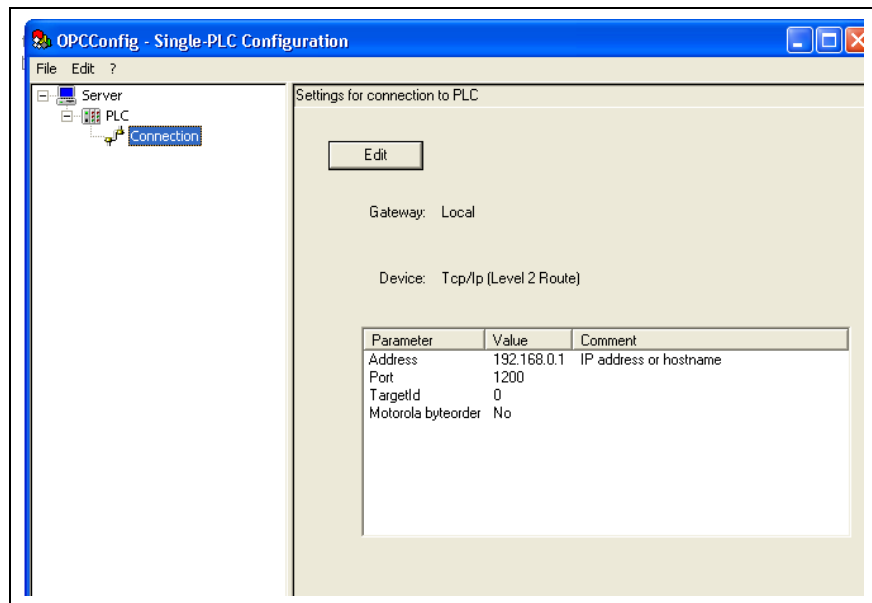


Abb. 7 Verbindung zur PLC-Konfiguration

Spezifische Einstellungen für OPC-Gateway:

6. Einstellungen für „Connection“ (Verbindung) auswählen.
7. Auf die Schaltfläche „Edit“ (Bearbeiten) klicken.

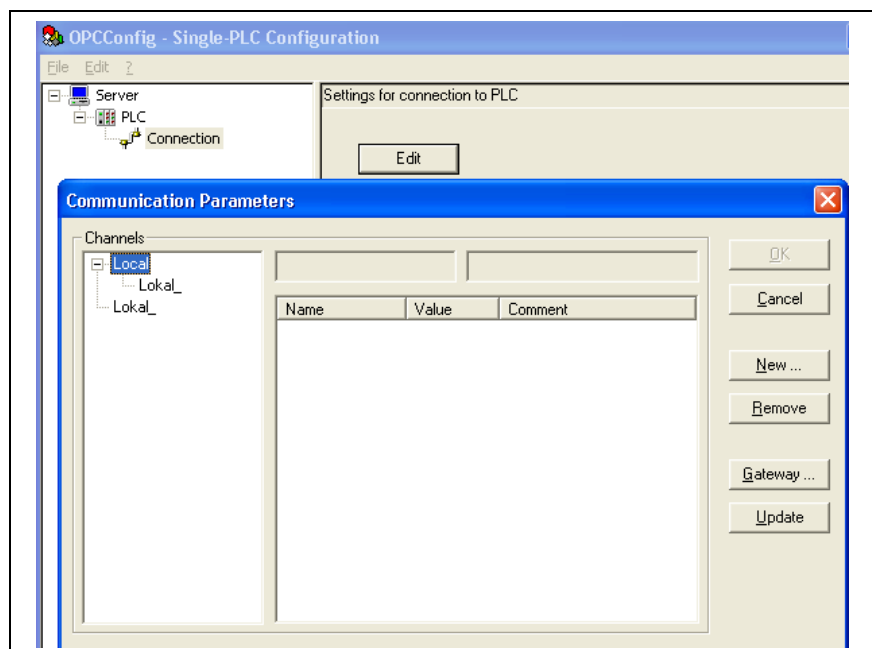


Abb. 8 Konfiguration des OPC-Gateways

8. Auf die Schaltfläche „New“ (Neu) klicken.

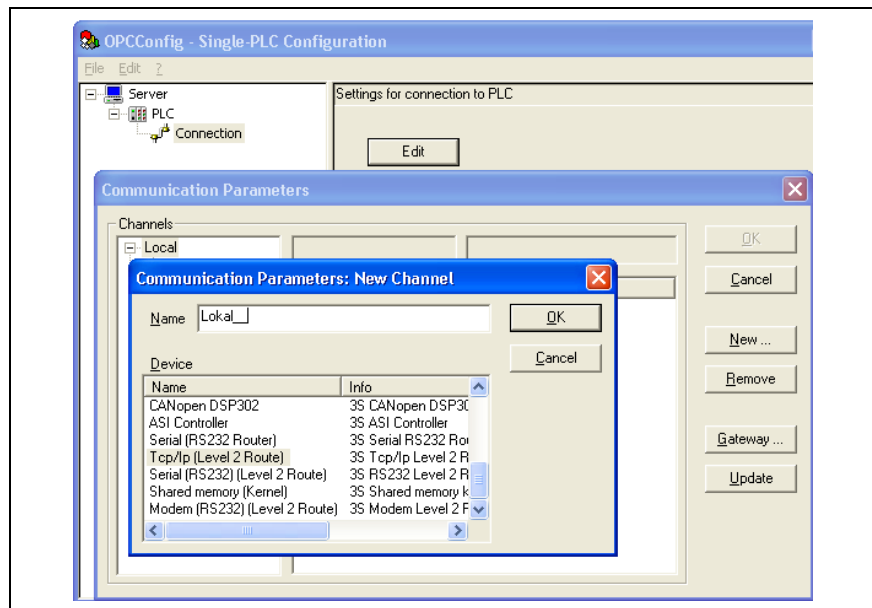


Abb. 9 Konfiguration des OPC-Gatewaykanals

9. Kommunikationskanal „Tcp/Ip (Level 2 Route)“ auswählen.
10. Auf die Schaltfläche „OK“ klicken.

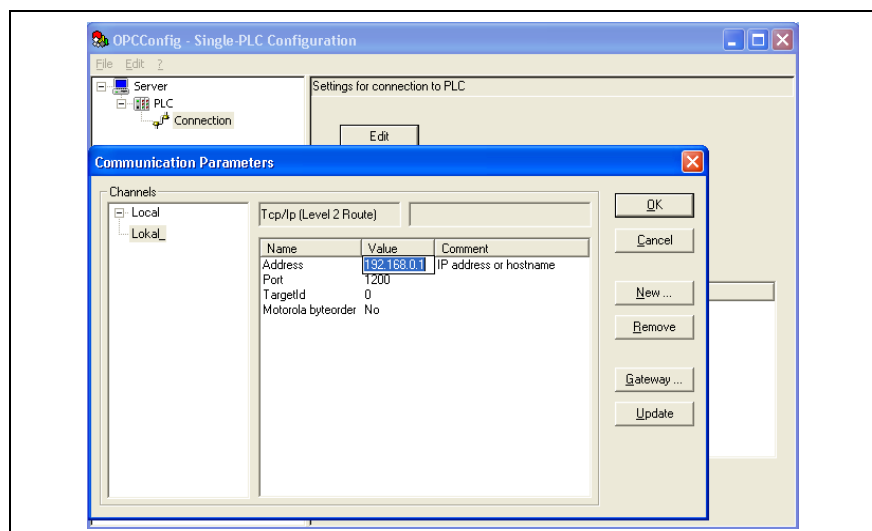


Abb. 10 Verbindung zur PLC-IP-Konfiguration

11. Doppelklick auf den Adresswert!

Hinweis

1. 192.168.0.1 ist eine IP-Adresse für den Master-Controller!
2. IP-Adresse festlegen, Standardwert: 192.168.0.1.
3. Einstellungen mit der Schaltfläche „OK“ bestätigen.

Die Einstellungen werden im Hauptfenster angezeigt:

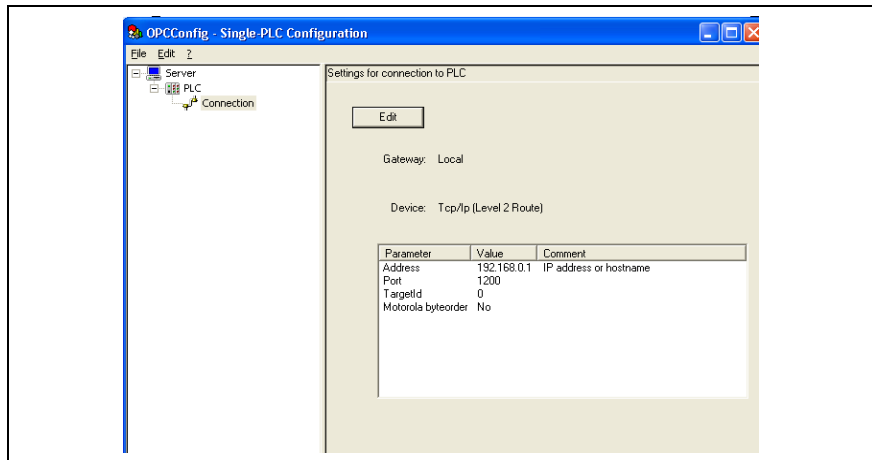


Abb. 11 Übersicht PLC-Einstellungen

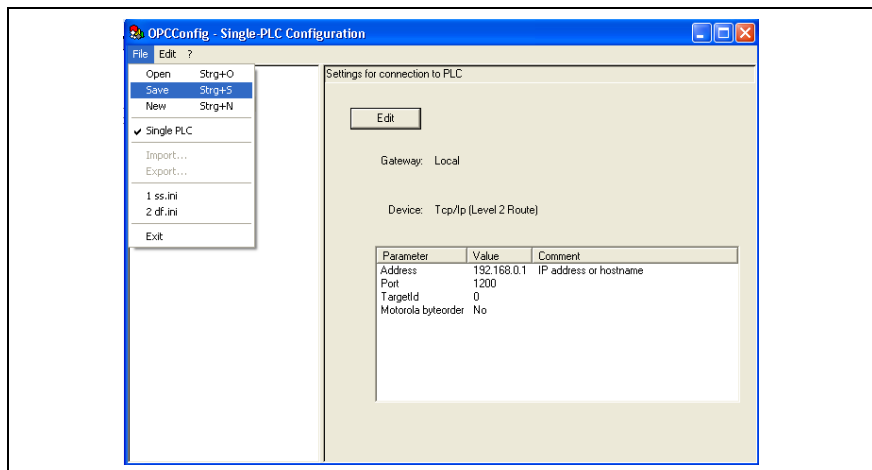


Abb. 12 Speichern der OPC-Konfigurationen

4. Als letzter Schritt muss die Konfiguration im Hauptmenü gespeichert werden.
5. OPC-Konfigurationsssoftware schließen.
6. Wenn alle Einstellungen vorgenommen wurden, kann die Anwendung MTS für PPU-E gestartet werden.

Wenn alle Vorgänge korrekt ausgeführt wurden, sollte es keine Probleme mit der Verbindung zwischen der PPU-E-Software und der PPU-Einheit geben!

Hinweis

Wenn die Anwendung nach dem Start von MTS für PPU-E keine Verbindung mit der Hardware herstellen kann, muss der Benutzer das Kabel, die IP-Adresse und alle Verbindungseinstellungen erneut überprüfen.

5 Beschreibung der Anwendung MTS für PPU-E

5.1 Hauptfenster

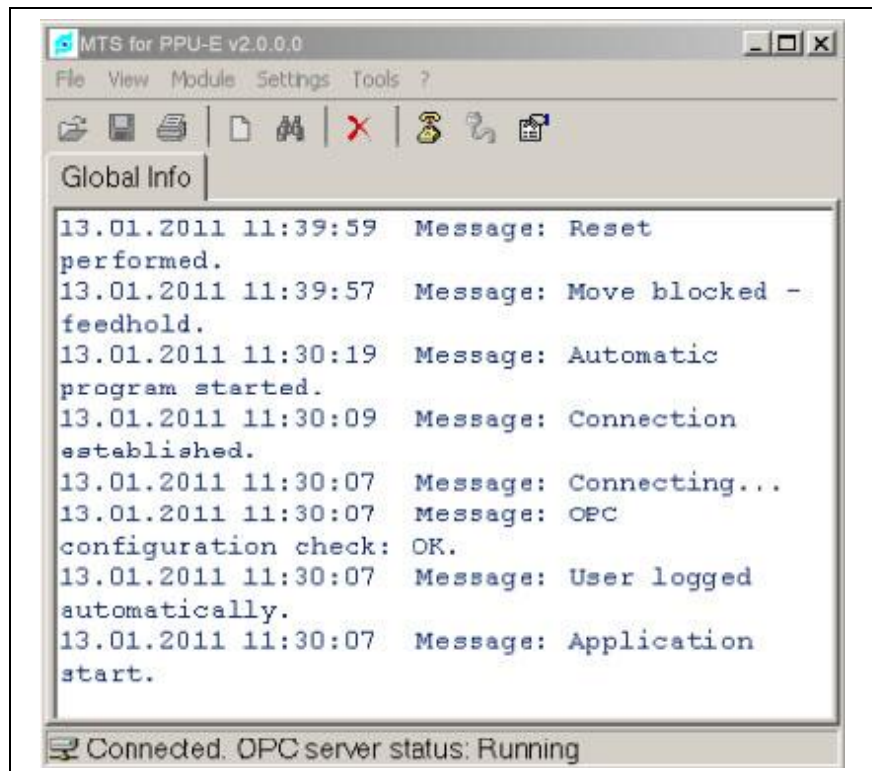


Abb. 13 Hauptfenster MTS für PPU-E

Im Hauptfenster der Anwendung kann der Benutzer auf die Anwendungseinstellungen zugreifen.

Zusätzlich werden alle ausgeführten Vorgänge protokolliert. Die Verbindung kann an- und abgeschaltet werden.

Das Hauptfenster besteht aus folgenden Elementen:

- Menü (siehe Kapitel 5.1.1)
- Symbolleiste (siehe Kapitel 5.1.2)
- Protokollfunktion (siehe Kapitel 5.1.3)
- Statusleiste für Verbindung (siehe Kapitel 5.1.4)

5.1.1 Menü*Abb. 14 Menü*

- | | |
|----------------|--|
| Datei | <ul style="list-style-type: none">• Speichern: Nicht belegt.• Drucken: Nicht belegt.• Beenden: Anwendung wird geschlossen. |
| Ansicht | <ul style="list-style-type: none">• Busdetails: Nicht belegt.• Lesezeichen setzen: Nicht belegt.• Protokolle löschen: Inhalte der Protokollfunktion werden gelöscht. |
| Modul | <ul style="list-style-type: none">• Öffnen: PPU-E-Modul-Fenster wird geöffnet.• Schließen: PPU-E-Modul-Fenster wird geschlossen.• Stoppen: Der Schnellstopfbefehl wird an alle Module gesendet.• Einstellungen• Verbindung herstellen/Verbindung trennen: Verbindung an-/abschalten.• Kommunikation öffnen: Fenster mit den IP-Einstellungen des PCs wird geöffnet. Benutzer kann die IP-Adresse des PCs festlegen.• Sprache: Anwendungssprache auswählen.• Vorgaben: Fenster mit den Vorgaben für die Anwendung wird geöffnet. |
| Tools | <ul style="list-style-type: none">• CRC-Rechner: Nicht belegt.• Numerische Umrechnungsfunktion: Nicht belegt.
• Programmeditor: Programmeditor-Fenster wird geöffnet. In diesem Fenster kann der Benutzer den Bewegungsablauf für die PPU-E-Einheit vorbereiten.• ? Informationen zur Anwendung MTS für PPU-E: Version, Händler etc. |

5.1.2 Symbolleiste



Abb. 15 Symbolleiste

Über diese Symbolleiste kann der Anwendung auf einige Anwendungsfunktionen und -optionen einfach zugreifen










Symbol	Beschreibung
	Nicht belegt
	Nicht belegt
	Nicht belegt
	Nicht belegt
	Nicht belegt
	Der Schnellstoppbefehl wird an alle Module gesendet.
	Verbindung herstellen/Verbindung trennen: Verbindung an-/abschalten
	Nicht belegt
	Fenster mit den Vorgaben für die Anwendung wird geöffnet

Tabelle 3

5.1.3 Protokollfunktion

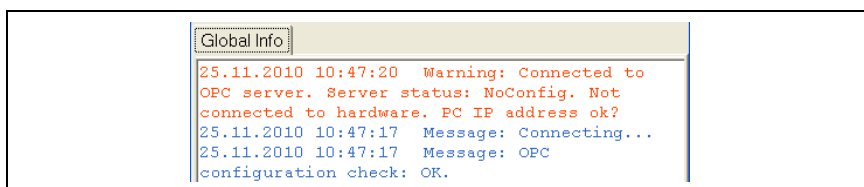


Abb. 16 Protokollfunktion

Alle Aktionen, Änderungen der Vorgaben und Anwendungsnachrichten werden von der Protokollfunktion protokolliert.

5.1.4 Statusleiste für Verbindung

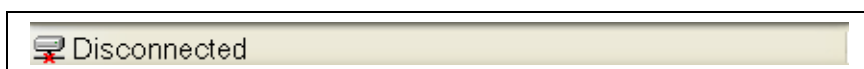


Abb. 17 Statusleiste für Verbindung

Der aktuelle Verbindungsstatus zwischen der Anwendung und der Hardware wird immer auf der Statusleiste angezeigt.

5.2 Vorgabenfenster

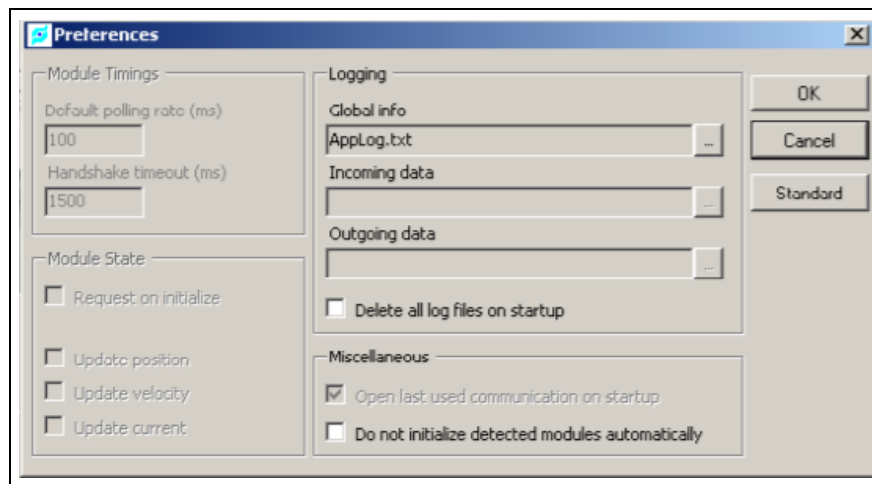


Abb. 18 Vorgaben

Im Vorgabenfenster kann der Benutzer bestimmte Anwendungseinstellungen festlegen. Wenn Einstellungen geändert werden, müssen die Änderungen immer mit der Schaltfläche „OK“ bestätigt werden.

- Modulablaufsteuerung – nicht editierbar.
- Modulstatus – nicht editierbar.
- Protokollierung
 - Globale Info: externe Datei, in der alle Aktionen und Anwendungsnachrichten protokolliert werden.
 - Eingehende Daten – nicht belegt.
 - Abgehende Daten – nicht belegt.
 - Alle Protokolldateien beim Start löschen: wenn das Kontrollkästchen markiert ist, wird die externe Datei (Protokolldatei) bei jedem Start der Anwendung initialisiert.
- Sonstiges
 - Erkannte Module nicht automatisch initialisieren: wenn dieses Kontrollkästchen markiert ist, wird die erkannte Hardware beim Start der Anwendung nicht initialisiert.

Hinweis

Alle Anwendungsvorgaben können einfach auf Standardwerte gesetzt werden.

hierzu kann die Schaltfläche „Standard“ verwendet werden.

5.3 Fenster „Module: PPU-E“

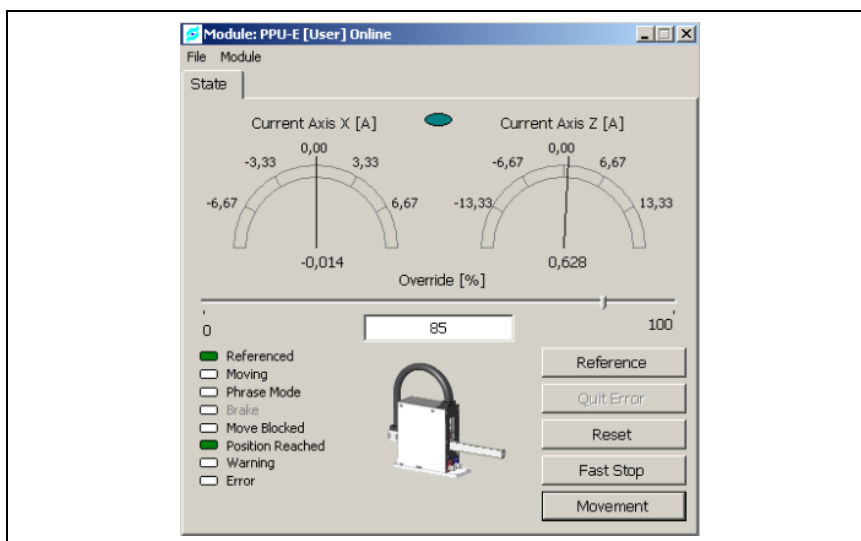


Abb. 19 Fenster „Module: PPU-E“

Im Fenster „Module: PPU-E“ wird der aktuelle Status der Einheit (PPU-E) angezeigt. Zusätzlich können einige Hauptbefehle an die Einheit gesendet werden. Der Status der Einheit wird alle 100 ms aktualisiert.

Der Name des Fensters beinhaltet die folgenden Informationen: Fenster „Module: PPU-E [<Benutzer>] <Status (Online/Offline)>“. Wenn keine Verbindung zwischen der Anwendung und der Hardware hergestellt werden kann, wird der Status „Offline“ angezeigt und das Fenster „Module: PPU-E“ ist deaktiviert!

Mit dem Schieberegler „Override“ (Überschreiben) kann der Benutzer die Achsenverfahrgeschwindigkeit beeinflussen (Anwender kann den empfohlenen Korrekturwert auch direkt eingeben). Beispiel: Die Achsengeschwindigkeit ist auf 3000 mm/s gesetzt, der Korrekturwert auf 50 %.

Die errechnete Zielgeschwindigkeit beträgt 1500 mm/s. Wenn der Korrekturwert auf 0 % gesetzt wird, werden alle Achsen gestoppt.

Mit den Schaltflächen rechts im Fenster können die wichtigsten Befehle an die Einheit gesendet werden. Mit der Schaltfläche „Movement“ (Bewegung) wird das Fenster vergrößert, und es werden neue wichtige Optionen und Bewegungsbefehle angezeigt.

Hinweis

Die Schaltfläche „Movement“ ist nur aktiviert, wenn die Einheit referenziert ist!

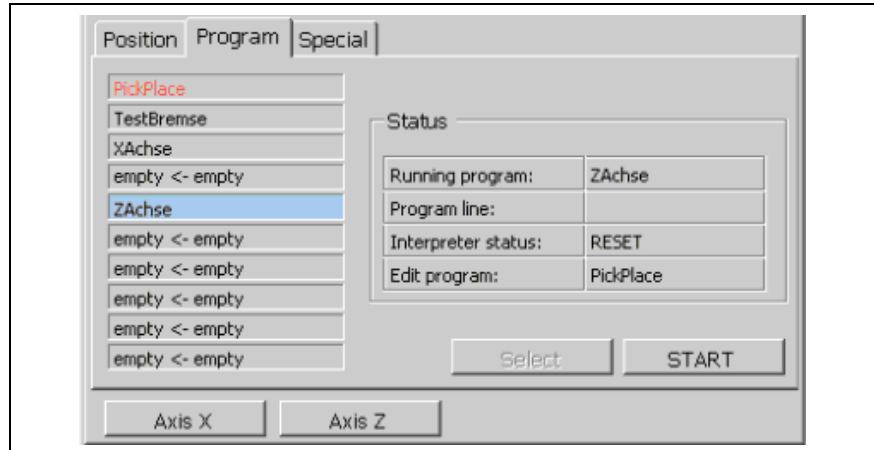


Abb. 20 Vergrößertes Fenster Schaltfläche „Movement“

- Registerkarte „Position“,
- Registerkarte „Programm“ (Programm),
- Registerkarte „Gripper“ (Greifer) – nur aktiv, wenn die Einheit einen Greifer umfasst,
- Registerkarte „Special“ (Extras),
- Schaltflächen, über die auf Achsenfenster zugegriffen werden kann.

5.3.1 Befehlsschaltflächen

Folgende Befehlsschaltflächen sind verfügbar:

- **Reference** (Referenz) – Referenz-Sequenz wird gestartet. Wenn der Referenzpunkt für die Einheit bereits festgelegt ist, werden die Achsen lediglich auf die Referenzpunkte verfahren.

Hinweis

Die Schaltfläche „Reference“ (Referenz) ist nur aktiv, wenn sich die Einheit im folgenden Betriebsstatus befindet – kein Fehler, Einheit eingeschaltet!

Wenn eine vollständige Referenz-Sequenz erforderlich ist, kann der Benutzer folgende Option im Modulmenü verwenden: „Force Reference“ (Referenz erzwingen). Referenz-Sequenz für Einheit wird ausgeführt.

- **Quit Error** (Fehler quittieren) – alle Fehler, Warnungen und Nachrichten von allen Achsen werden bestätigt.

- **Reset** (Zurücksetzen) – Fehler, Warnungen und Nachrichten von allen Achsen werden bestätigt. Zusätzlich wird der Motion-Interpreter zurückgesetzt.
- **Fast stop** (Schnellstopp) – alle Achsen werden sofort gestoppt. Alle Bewegungen der Einheit werden blockiert. Um die Blockierung aufzuheben, muss ein Befehl zum Zurücksetzen an die Einheit gesendet werden.
- **Movement** (Bewegung) – das Fenster wird vergrößert, zusätzliche Optionen werden angezeigt.

5.3.2 Registerkarten für Bewegung

- Registerkarte „**Position**“ – hier können Benutzer alle Achsen an die gewünschte Position bewegen. Der Benutzer muss lediglich die gewünschte Position in die Eingabefelder eingeben und die Schaltfläche „Move“ (Bewegen) betätigen.

Hinweis

Die Funktion zum Bewegen ist nur aktiv, wenn das automatische Programm nicht gestartet wurde. Dies entspricht folgendem Status des Motion-Interpreters: Reset (Zurücksetzen).

- Registerkarte „**Program**“ (Programm) – im Speicher der Einheit enthaltene Bewegungsprogramme werden hier aufgelistet. Insgesamt können zehn Bewegungsprogramme vorbereitet und gespeichert werden. Darüber hinaus wird auf der rechten Seite der Registerkarte der Status des Motion-Interpreters angezeigt.

Hinweis

Der Hintergrund des aktuell ausgeführten (ausgewählten) Programms ist blau. Das zu bearbeitende Programm wird in Orange angezeigt.

WARNUNG

PPU-E kann schwerwiegende Unfälle und Verletzungen verursachen

- ➔ Vor der Verwendung der Schaltfläche „START“ muss sichergestellt sein, dass sich nichts in Reichweite der PPU-E-Einheit befindet!

Benutzer können hier das ausgewählte Bewegungsprogramm starten (Schaltfläche „START“), ein anderes Bewegungsprogramm zum Bearbeiten auswählen (durch Klicken auf den Programmnamen) oder ein anderes Bewegungsprogramm für den automatischen Bewegungsablauf auswählen, durch klicken auf den Programmnamen und anschließend auf die Schaltfläche „Select“ (Auswählen). Die Schaltfläche „Select“ (Auswählen) ist für „einfache“ Benutzer (siehe Kapitel „Anwendungsbenutzer“) deaktiviert.

- **Registerkarte „Gripper“** (Greifer) – Handhabung des Greifers: öffnen, schließen, lösen. Das Bild zeigt den aktuellen Status des Greifers an.

Hinweis

Die Registerkarte „Gripper“ (Greifer) ist nur aktiv, wenn die Einheit über einen Greifer verfügt. Die Einheitenkonfiguration wird von der Anwendung automatisch erkannt.

- ➔ **Registerkarte „Special“** (Extras) – kann standardmäßig nur vom Administrator verwendet werden! Dennoch kann die Hardwarekonfiguration hier geändert werden.

Im unteren Fensterbereich befinden sich Achsenschnittflächen. Das ausgewählte Fenster zu Achsenhandhabung kann hier geöffnet werden.

5.4 Achsenfenster

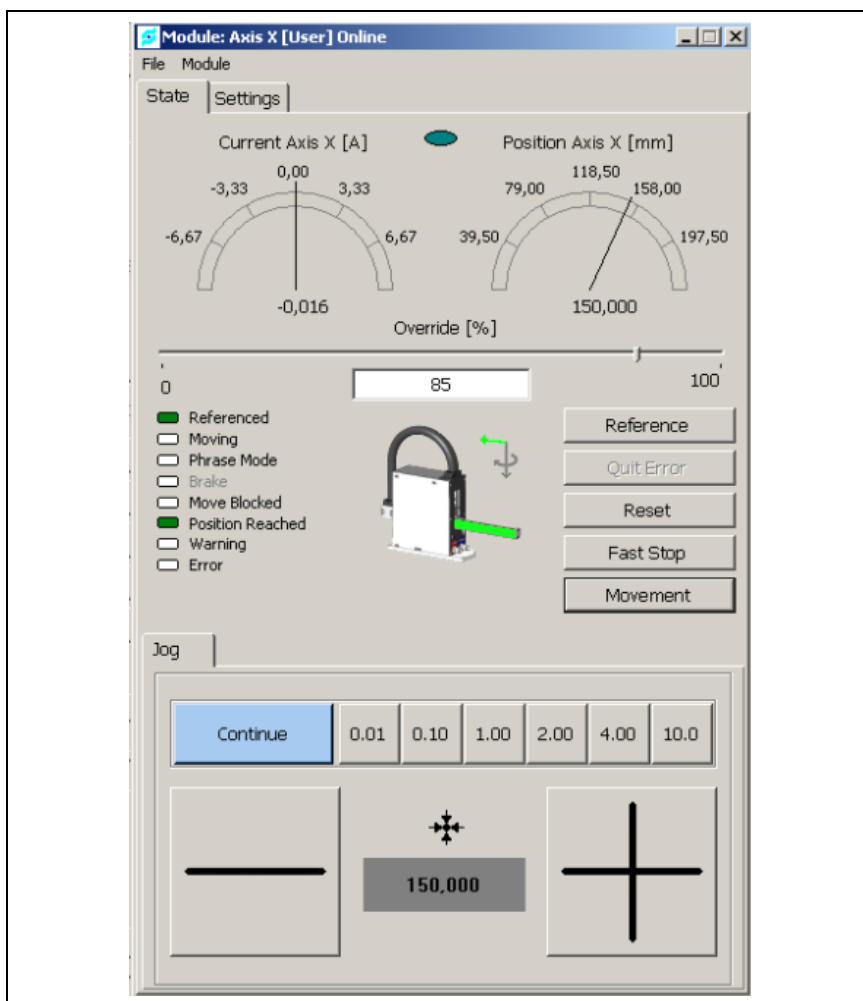


Abb. 21 Achsenfenster

Im Achsenfenster wird eine einzelne Achse der Einheit beschrieben. Es ist ähnlich wie das Fenster „Module: PPU-E“ aufgebaut.

Sämtliche Anzeigen in diesem Fenster beziehen sich auf eine einzelne Achse mit Ausnahme der Befehlsschaltfläche „Override“ (Überschreiben) und „Reset“ (Zurücksetzen).

„Override“ (Überschreiben) in diesem Fenster und in allen anderen Anwendungsfenstern, in denen „Override“ auf den gewünschten Wert gesetzt werden kann,

ist global. Das bedeutet, dass sich das Ändern von „Override“ in einem beliebigen Fenster auf die gesamte Einheit und alle Achsen auswirkt.

Das Achsenfenster beinhaltet zwei Registerkarten:

- **Registerkarte „State“** (Status) – Achsenstatus gesamt, mit Unterregisterkarte „Jog“ (Manuelle Steuerung).
- **Registerkarte „Settings“** (Einstellungen) – Achsenspezifische Einstellungen.

Auf der Registerkarte „State“ (Status) findet der Benutzer Informationen zum Achsenstatus, zur tatsächlichen Position und zum anliegenden Strom.

Zusätzlich beinhaltet die Registerkarte – ähnlich wie andere Anwendungsfenster – einige Befehlsschaltflächen.

5.4.1 Befehlsschaltflächen

Folgende Befehlsschaltflächen sind verfügbar:

- **Reference** (Reference) – Achsenreferenz-Sequenz wird gestartet. Wenn der Referenzpunkt für die Einheit bereits festgelegt ist, wird die Achse lediglich auf den Referenzpunkt verfahren.

Hinweis

Die Schaltfläche „Reference“ (Referenz) ist nur aktiv, wenn sich die Einheit im folgenden Betriebsstatus befindet – kein Fehler, Einheit eingeschaltet!

- **Quit Error** (Fehler quittieren) – alle Achsenfehler, Warnungen, Nachrichten werden bestätigt.
- **Reset** (Zurücksetzen) – Fehler, Warnungen und Nachrichten von allen Achsen werden bestätigt. Zusätzlich wird der Motion-Interpreter zurückgesetzt.
- **Fast stop** (Schnellstopp) – alle Achsen werden sofort gestoppt. Alle Bewegungen der Einheit werden blockiert. Um die Blockierung aufzuheben, muss ein Befehl zum Zurücksetzen an die Einheit gesendet werden.
- **Movement** (Bewegung) – das Fenster wird vergrößert, die Registerkarte „Jog“ (Manuelle Steuerung) wird angezeigt.

5.4.2 Unterregisterkarte „Jog“ (Manuelle Steuerung)

“Die Registerkarte „Jog“ (Manuelle Steuerung) unterstützt den Benutzer bei der Achspositionierung.

Es sind mehrere Bewegungsschrittlängen verfügbar:

- 0,01 mm
- 0,10 mm
- 1,00 mm
- 2,00 mm
- 4,00 mm
- 10,00 mm
- „Continue“ (Fortfahren) – die Bewegung wird fortgesetzt.

Mit den Schaltflächen „-“ und „+“ und der ausgewählte Schrittlänge kann die Achse einfach auf die gewünschte Position gesetzt werden.

Hinweis

Der manuelle Steuerungsmodus ist nur aktiv, wenn das automatische Programm nicht gestartet wurde.

Dies entspricht dem folgenden Status des Motion-Interpreters: Reset (Zurücksetzen).

5.4.3 Registerkarte „Settings“ (Einstellungen)

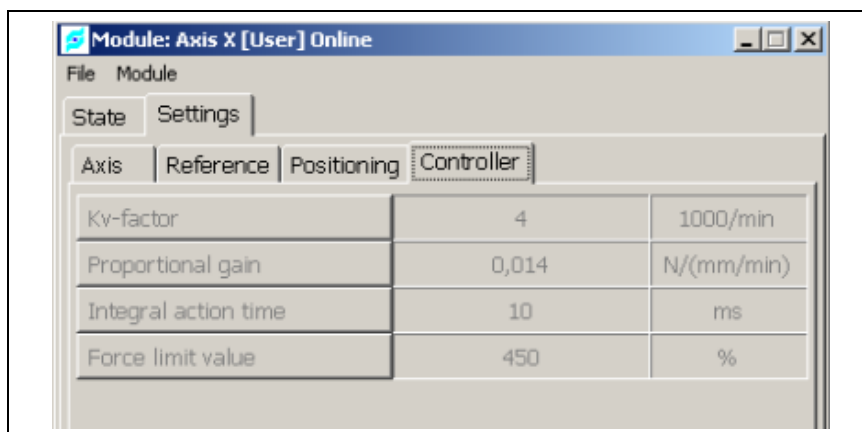


Abb. 22 Achseneinstellungen

Die Achseneinstellungen sind in vier Gruppen unterteilt:

- „Axis“ (Achse)
 - „Min. Position“ – negatives Limit [mm] Achsenpositionierung
 - „Max. Position“ – positives Limit [mm] Achsenpositionierung

- **„Reference“ (Referenz)**
 - „Axis Home Position“ – gewünschter Achsenreferenzpunkt [mm]
 - „Direction of motion“ – Bewegungsrichtung durch Festlegung von Referenzpunkten
- **„Positioning“ (Positionierung):**
 - „Jog Velocity Normal“ – erste (normale) Steuerungsgeschwindigkeit [mm/s]
 - „Jog Velocity High“ – zweite (hohe) Steuerungsgeschwindigkeit [mm/s]
 - „Jog Acceleration Normal“ – erste (normale) Beschleunigung manuelle Steuerung [mm/s²]
 - „Jog Acceleration High“ – zweite (hohe) Beschleunigung manuelle Steuerung [mm/s²]
 - „Jog Deceleration“ – Abbremsung manueller Steuerungsmodus [mm/s²]
 - „Jog Time Velo. Change“ – manuelle Steuerung Wechsel von erster zu zweiter Geschwindigkeit/ Beschleunigung [ms]
 - „Max. Velocity“ – maximale Geschwindigkeit der Achse [mm/s]
 - „Max. Acceleration“ – maximale Beschleunigung der Achse [mm/s²]
- **„Controller“**
 - „Kv-factor“ – Achsen-Controller Positionsschleife Faktor [1000/min],
 - „Proportional gain“ – Achsen-Controller Geschwindigkeitsschleife proportionale Zunahme [N/(mm/min)]
 - „Integral action time“ – Achsen-Controller Geschwindigkeitsschleife integrale Aktionszeit [ms].

Hinweise

Alle Änderungen an den Achseneinstellungen sollten durch Klicken auf die Schaltfläche „Änderungen speichern“ bestätigt werden.

Alle Änderungen an den Einstellungen, die auf der Registerkarte „Controller“ aufgeführt sind, werden sofort an den Controller gesendet. In diesem Fall ist kein Speichervorgang erforderlich.

Nicht alle Registerkarten für Achseneinstellungen sind für den „einfachen“ Benutzer aktiviert. Der „Supervisor“ und der „Administrator“ haben Zugriff auf alle Achseneinstellungen.

5.5 Fenster „Program editor“ (Programmeditor)

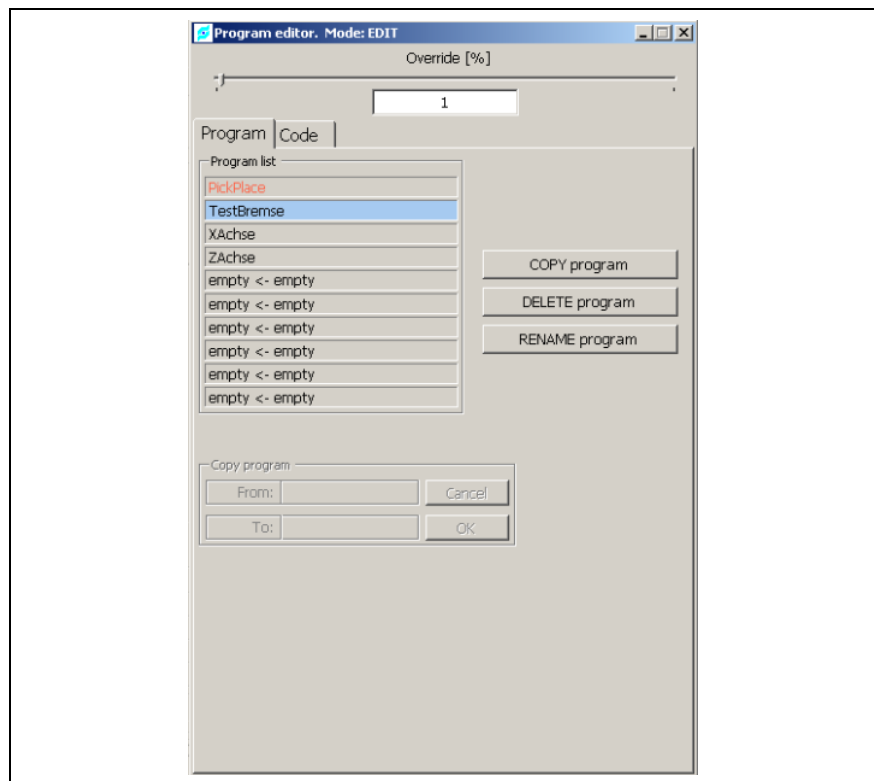


Abb. 23 Automatischer Programmeditor

Mit dem automatischen Programmeditor können Benutzer ihre eigene „Bestückungs“-Sequenz vorbereiten.

Der Programmeditor umfasst zwei Registerkarten:

- Registerkarte „Program“
- Registerkarte „Code“.

5.5.1 Registerkarte „Program“

Momentan gespeicherte Sequenzen sind in dieser Registerkarte aufgelistet. Maximal können 10 Kundensequenzen gleichzeitig gespeichert werden. Das zum Bearbeiten ausgewählte Programm wird in Orange angezeigt.

Das ausgewählte Programm kann:

- kopiert werden – mit der Schaltfläche zum Kopieren wird die Objektgruppe „Copy program“ aktiviert. Zielposition im Speicher
- muss ausgewählt werden. Mit der Schaltfläche „OK“ wird der Kopiervorgang bestätigt.

- gelöscht werden – Schaltfläche „Delete program“ (Programm löschen) verwenden.

Hinweis

Das für die automatische Sequenz ausgewählte Programm

(Hintergrund: blau) kann nicht gelöscht werden!

Der Speicherbereich für die Programmspeicherung des Master-Controllers ist gesperrt.

- umbenannt werden – mit der Schaltfläche „Rename program“ (Programm umbenennen). Der neue Programmname kann nicht mit einem bereits gespeicherten Programmnamen identisch sein.

5.5.2 Registerkarte „Code“

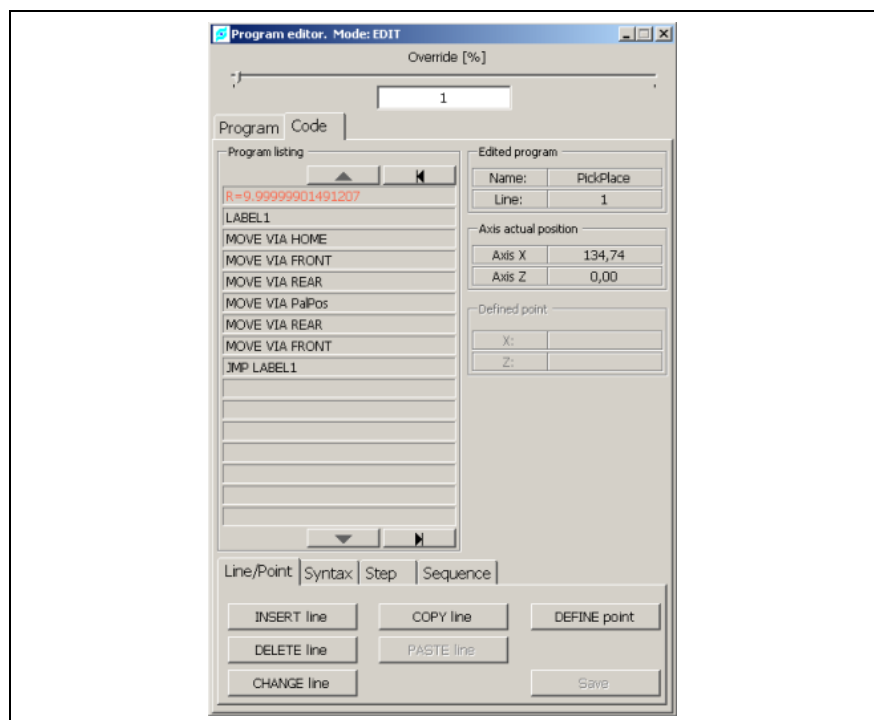






Abb. 24 Programmeditor, Registerkarte „Code“

Die Registerkarte „Code“ beinhaltet folgende Elemente:

- Liste der ausgewählten Programme, wobei die Zeile/der Befehl für das ausgewählte Programm in Orange angezeigt wird,
- Informationen zum Programm, das bearbeitet werden soll,
- derzeitige Achsenpositionen der Einheit,

- defined point – aktiviert, wenn ausgewählte Programmzeile definierte Punkte enthält,
- Registerkarten zum Bearbeiten/Prüfen von Programmen: „Line/Point“, „Syntax“, „Step“, „Sequence“ (Zeile/Einfügen, Syntax, Schritt, Sequenz),
- Schaltflächen für Navigation im Programm:
 -   : ausgewählte Zeile nach oben/nach unten,
 -   : zur ersten/letzten Programmzeile/Befehl gehen.

4.5.2.1 Registerkarten zum Bearbeiten/Prüfen von Programmen

Registerkarte
„Line/Point“
(Zeile/Punkt)

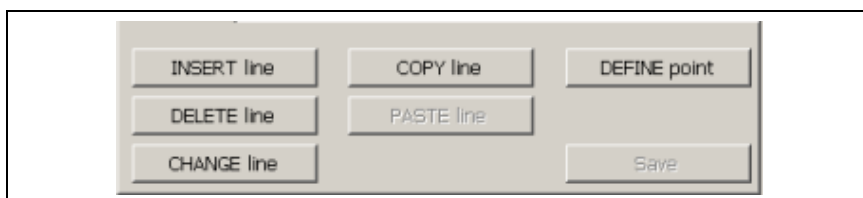


Abb. 25 Line/Point

Über diese Registerkarte können Benutzer verschiedene Vorgänge für das ausgewählte Programm ausführen. Zunächst kann eine neue Programmzeile eingefügt werden (Schaltfläche: INSERT line). Eine ausgewählte Programmzeile kann auch gelöscht werden (Schaltfläche: DELETE line). Außerdem kann eine ausgewählte Programmzeile auch geändert werden (Schaltfläche: CHANGE line). Die Änderung bezieht sich nur auf Parameter, Zielpunkte oder Zielpositionen – Befehle bleiben unverändert. Eine Programmzeile kann auch kopiert (Schaltfläche: COPY line) und mit der Schaltfläche „PASTE line“ an der gewünschten Position eingefügt werden. Zusätzlich können Programmpunkte definiert werden. Für jedes Programm kann der Benutzer maximal 50 Punkte festlegen.

Bei einem Punkt handelt es sich um die Einheitenposition (der Punkt besteht aus den jeweiligen Positionen der Einheitenachsen).

Punkte können in einem Programm anstelle von direkten Achsenpositionen verwendet werden.

Schließlich können die finalen Änderungen gespeichert werden (Schaltfläche: Save).

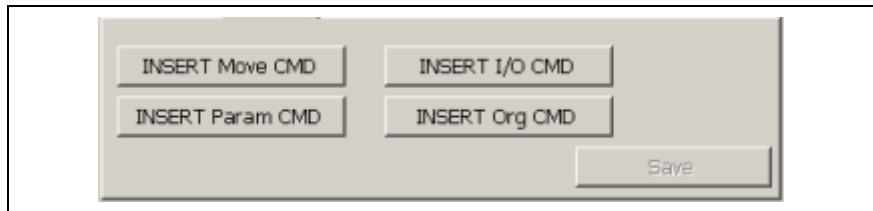
**Registerkarte
„Syntax“**

Abb. 26 Registerkarte „Syntax“

Jede der vier Schaltflächen repräsentiert eine Befehlsgruppe:

- Die Schaltfläche „INSERT Move CMD“ repräsentiert eine Gruppe mit Bewegungsbefehlen. Beispiel: MOVE VIA
- Die Schaltfläche „INSERT Param CMD“ repräsentiert eine Gruppe mit Parameterbefehlen. Beispiel: R
- Die Schaltfläche „INSERT I/O CMD“ repräsentiert eine Gruppe mit Eingabe-/Ausgabebefehlen. Beispiel: WAIT INPUT
- Die Schaltfläche „INSERT Org CMD“ repräsentiert eine Gruppe mit Organisationsbefehlen. Beispiel: LABEL.

Ein neuer Befehl kann eingefügt werden, indem die entsprechende Schaltfläche für die Befehlsgruppe betätigt wird.

Um einen neuen Befehl einzufügen, muss der Benutzer zunächst die entsprechende Schaltfläche für die Befehlsgruppe betätigen und dann die folgenden Anweisungen befolgen.

Beispiel: Hinzufügen des Befehls MOVE zum Programm:

- Schaltfläche INSERT Move CMD betätigen
- Befehl MOVE in der Gruppe der Bewegungsbefehlen auswählen,
- Bewegungszielpunkt aus der Liste auswählen.

Alle verfügbaren Befehle werden am Ende dieser Dokumentation präzise beschrieben

(siehe Kapitel: Anhang: Programmierbefehle).

**Registerkarte
„Step“ (Schritt)**

Abb. 27 Registerkarte „Step“ (Schritt)

Über die Schaltfläche „STEP+“ kann die vorbereitete Sequenz schrittweise überprüft werden. Mit dieser Überprüfung kann ungewolltes Verhalten der Einheit vermieden und die Gefahr eines Zusammenstoßes minimiert werden.

Der Bearbeitungsmodus wechselt automatisch in den Modus „Step“. Bei jedem Klick auf die Schaltfläche „STEP+“ wird ein Auslöser an die Einheit gesendet, und der nächste Befehl wird ausgeführt. Der momentan ausgeführte Befehl ist markiert – die Schriftfarbe ist orange.

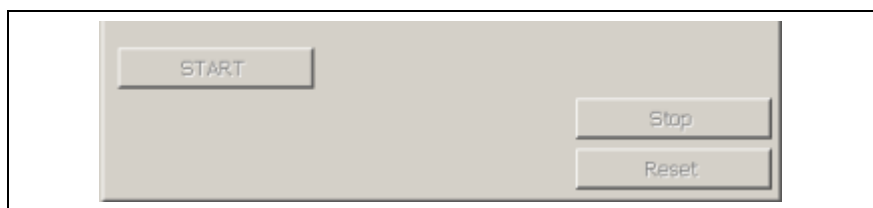
**Registerkarte
„Sequence“
(Sequenz)**

Abb. 28 Registerkarte „Sequence“ (Sequenz)

Mit der Schaltfläche „START“ kann die vorbereitete Sequenz sofort geprüft werden. Der Bearbeitungsmodus wechselt automatisch in den Modus „Sequence“. Der momentan ausgeführte Befehl ist markiert – die Schriftfarbe ist orange.

Hinweis

Das automatische Programm kann nicht gestartet werden (Anwendungsfenster: „Module: PPU-E“, Schaltfläche „START“), wenn der Bearbeitungsmodus der Einheit „Step“ oder „Sequence“ lautet. Nach dem Test muss der Benutzer die Einheit zurücksetzen (Schaltfläche: „Reset“ auf den Registerkarten: „STEP+“, „Sequence“).

5.6 Anwendungsbenutzer

Es sind momentan drei Benutzerzugriffsebenen implementiert:

- **Benutzer** – ohne Kennwort; beim Start der Anwendung standardmäßig angemeldet; sogenannter Benutzer mit beschränkten Berechtigungen; viele Anwendungsfunktionen und -optionen sind für diesen Benutzertyp nicht verfügbar.
- **Supervisor** – kennwortgeschützt; fast alle Anwendungsfunktionen und -optionen sind für diesen Benutzertyp verfügbar; Standardkennwort für Supervisor: „sv“.
- **Administrator** – kennwortgeschützt, dynamisches Kennwort wird auf der Basis einer Zufallszahl erstellt, Zugriff auf alle Anwendungsfunktionen und -optionen.

Der Anwendungsbenutzer kann sich zu jeder Zeit anmelden. Der Benutzer hat in fast allen Anwendungsfenstern Zugriff auf das Anmeldeformularmenü „Change User“ (Benutzer ändern), Direktaufruf „F6“.

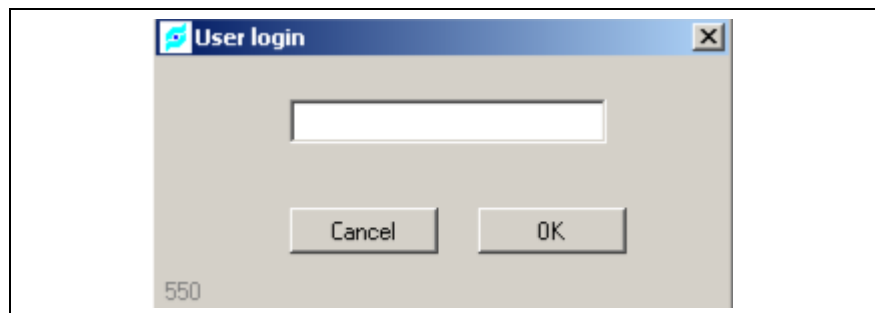


Abb. 29 Benutzeranmeldeformular

Benutzer können sich, durch Eingabe eines Kennwortes anmelden: Benutzerebene – kein Kennwort (leer),

Supervisor-Ebene – „sv“.

5.7 Anwendungssprachen

- In MTS für PPU-E werden derzeit zwei Sprachen unterstützt: Deutsch und Englisch.
- Die Anwendungssprache kann jederzeit geändert werden. Die Änderungen werden jedoch erst bei einem Neustart der Anwendung übernommen.

Benutzer können die Sprache im Menü „Settings/Language“ (Einstellungen/Sprache) im Hauptfenster der Anwendung ändern.

5.8 Externe Protokollfunktion

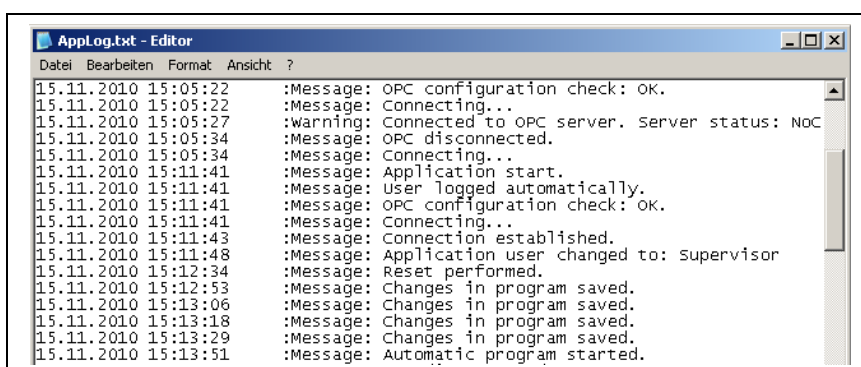


Abb. 30 Externe Protokollfunktion

Alle Benutzeraktivitäten werden in der internen Protokollfunktion der Anwendung protokolliert (Abb. 16, Seite 16). All diese Daten werden auch in einer externen Protokollfunktion in Form einer TXT-Datei gespeichert. Die externe Protokolldatei befindet sich im Anwendungsordner.

6 Anhang

6.1 Programmierbefehle

Die folgende Beschreibung von Befehlen wurde aus folgenden Dokumenten kopiert:

Rexroth IndraMotion for Handling 02VRS

Operating and Programming Instruction

R911318055

Edition 02

6.1.1 Bewegungsbefehle

Die folgenden Definitionen gelten für alle Befehle aus der Gruppe „move“:

Der TCP (Werkzeugmittelpunkt) bewegt sich von der derzeitigen Position zur Zielposition in Form einer PTP-Bewegung (PTP -> Punkt-zu-Punkt).

Bei der Berechnung der verschiedenen Achsengeschwindigkeiten werden die Maximalgeschwindigkeiten und -beschleunigungen der einzelnen Achsen sowie der Korrekturwert für die Geschwindigkeit berücksichtigt.

Die PTP-Bewegung kann auf die folgenden zwei Arten über den Parameterbefehl „INTERPOLATION“ optimiert werden:

- „PTP“: zeitoptimiert
- „LINEAR“: streckenoptimiert

Für beide Fälle gilt Folgendes:

Wenn die Bewegung eine Punkt-zu-Punkt-Bewegung ist, werden alle Achsen, die in den Bewegungsprozess einbezogen sind, zur selben Zeit gestartet und erreichen zur selben Zeit ihr Ziel.

Die Achsengeschwindigkeiten und -beschleunigungen können in Bezug auf die entsprechenden Maximalwerte durch V_PTP und A prozentual gewichtet werden.

MOVE	<p>Beispiel: MOVE HOME</p> <p>Der TCP bewegt sich von seiner derzeitigen Position zur angegebenen Zielposition (HOME) in einer Punkt-zu-Punkt-Bewegung.</p>
MOVE_REL	<p>Beispiel: MOVE_REL HOME</p> <p>Die Zielposition wird durch eine Vektoraddition der derzeitigen Position und des angesteuerten Punktes (z. B. HOME) errechnet. Die Bewegung ist eine Punkt-zu-Punkt-Bewegung.</p>
MOVE TIL	<p>Beispiel: MOVE HOME TIL GR_CLOSE</p> <p>Der TCP bewegt sich von seiner derzeitigen Position zur Zielposition (z. B. HOME, Punkt-zu-Punkt-Bewegung). Sobald sich die programmierte Eingabe (z. B. GR_CLOSE) in High ändert, wird die Verfahrbewegung abgebrochen und es wird eine Weberschaltung zum nächsten Satz ausgeführt. Wenn während der Verfahrbewegung keine Stoppbedingung erfüllt ist, wird der Satz als normale, absolute Verfahrbewegung mit Vorschub zum nachfolgenden Satz ausgeführt. Wenn die Stoppbedingung bereits vor dem Starten der Bewegung erfüllt ist, erfolgt sofort der Vorschub zum nächsten Satz.</p>
MOVE_REL TIL	<p>Beispiel: MOVE_REL HOME TIL GR_CLOSE</p> <p>Die Zielposition (Punkt-zu-Punkt-Bewegung) wird durch eine Vektoraddition der derzeitigen Position und des angesteuerten Punktes (HOME) errechnet. Sobald die angesteuerte Eingabe (z. B. GR_CLOSE) zu High wechselt, wird die Verfahrbewegung abgebrochen. Sobald alle Achsen gestoppt sind, erfolgt ein Vorschub zum nächsten Satz. Wenn während der Verfahrbewegung keine Stoppbedingung erfüllt ist, wird der Satz als normale, relative Verfahrbewegung mit Vorschub zum nächsten Satz ausgeführt. Wenn die Stoppbedingung bereits vor dem Starten der Bewegung erfüllt ist, erfolgt sofort der Vorschub zum nächsten Satz.</p>

MOVE VIA

Beispiel: MOVE VIA HOME

Der TCP bewegt sich von seiner derzeitigen Position zur Zielposition (HOME) in einer Punkt-zu-Punkt-Bewegung.

Der Satzvorschub erfolgt, sobald sich die Achsen um die Überschleifdistanz an die Endposition angenähert haben.

MOVE_REL VIA

Beispiel: MOVE_REL VIA HOME

Die Zielposition wird durch eine Vektoraddition der derzeitigen Position und des programmierten Punktes (z. B. HOME) errechnet. Die Bewegung ist eine Punkt-zu-Punkt-Bewegung. Der Satzvorschub erfolgt, sobald sich die Achsen um den Überschleifabstand an die Endposition angenähert haben.

MOVE Ax=

Beispiel: MOVE X=1000.0

Die ausgewählte Achse verfährt zur eingegebenen absoluten Position (z. B. X-Achse auf 1000,0).

MOVE VIA Ax=

Beispiel: MOVE VIA X=1000.0

Die ausgewählte Achse (X-Achse) fährt verschleifend über die eingegebene absolute Position (1000,0) aus. Der Satzvorschub erfolgt, sobald sich die Achse um die Verschleifposition an die Endposition angenähert hat.

MOVE Ax=Point

Beispiel: MOVE X=HOME

Die ausgewählte Achse führt eine absolute Bewegung zur eingegebenen Punktcoordinate aus.

MOVE VIA Ax=P Beispiel: MOVE VIA X=HOME

Die ausgewählte Achse fährt verschleifend über die eingegebene Punktcoordinate.

Der Satzvorschub erfolgt, sobald sich die Achse um den Verschleifabstand an die Endposition angenähert hat.

6.1.2 Parameterbefehle

Parameterzugriffe haben einen modalen Effekt. Dies bedeutet, dass ein programmierter Parameter sich auf alle nachfolgenden Sätze auswirkt, bis er erneut programmiert wird. Bei jedem Programmabbruch, jeder Rücksetzung oder Programmbeendigung werden alle aktiven Parametereinträge (mit Ausnahme von Gleitkomma- und Ganzzahlvariablen) des Interpreters zurückgesetzt.

V_PTP= Beispiel: V_PTP=30

V_PTP ermöglicht die prozentuale Gewichtung der Geschwindigkeit in der Sequenz. Diese Reduktion basiert auf den bei der Konfiguration definierten Maximalwerten.

A= Beispiel: A=12

Grundsätzlich werden die in der Maschinenkonfiguration definierten Maximalbeschleunigungen für alle Verfahrssätze verwendet. Mit diesem Parameterbefehl können alle Beschleunigung prozentual reduziert werden. Diese Reduktion basiert auf den bei der Konfiguration definierten Maximalwerten.

R= Beispiel: R=25

Der Überschleifabstand (in mm) steuert den Satzvorschub für verschleifende Bewegungen.

Sobald die zu verfahrende Strecke kürzer ist als die Überschleifdistanz, wird der nächste Bewegungsbefehl ausgeführt.

REAL_VAR_ Beispiel: REAL_VAR_12=58.2

Mit diesem Parameter können die Variablen beschrieben werden, die in einem globalen Feld von Gleitkommawerten verwaltet werden. Ein Gleitkommawert ist einer realen Variablen (z. B. 12) zugeordnet, in diesem Fall 58,2. Die Gleitkommawerte werden in einem globalen Feld gespeichert. Dies bedeutet, dass nur ein Feld für alle Programme verfügbar ist.

INT_VAR_

Beispiel: INT_VAR_12=12

Mit diesem Parameter können die Variablen beschrieben werden, die in einem globalen Feld von Ganzzahlwerten verwaltet werden. Ein Ganzzahlwert ist einer Ganzzahlvariablen (z. B. 12) zugeordnet, in diesem Fall 12. Die Ganzzahlwerte werden in einem globalen Feld gespeichert. Dies bedeutet, dass nur ein Feld für alle Programme verfügbar ist.

INC INT_VAR_

Beispiel: INC INT_VAR_12

Der Wert der Ganzzahlvariablen wird um eins erhöht.

DEC INT_VAR_

Beispiel: DEC INT_VAR_12

Der Wert der Ganzzahlvariablen wird um eins verringert.

INTERPOLATION

Beispiel: INT= LINEAR oder INT= PTP

Dieser Befehl umfasst die Punkt-zu-Punkt-Bewegung.

PTP: Ausführung der Punkt-zu-Punkt-Bewegung ist zeitoptimiert.

LINEAR: Dies bedeutet, dass, obwohl alle Achsen die Bewegung zur selben Zeit starten und beenden, sich ihre Beschleunigungsphasen unterscheiden.

Folglich können die kürzesten Punkt-zu-Punkt-Verfahrzeiten erzielt werden.

Ausführung der Punkt-zu-Punkt-Bewegung ist streckenoptimiert.

Dies bedeutet, dass auf allen Achsen die Bewegung zur selben Zeit gestartet und gestoppt wird. Und sie verfügen über dieselben Beschleunigungsphasen.

Folglich können die kürzesten Punkt-zu-Punkt-Abstände erzielt werden.

OVERRIDE

Beispiel: OVERRIDE ON oder OVERRIDE OFF

Mit diesem Befehl wird die über die Benutzerschnittstellen festgelegte Geschwindigkeitsüberschreibung aktiviert und deaktiviert.

6.1.3 E/A-Befehle

OUTPUT

Beispiel: Gripper=0

Die Ausgabe (Greifer) kann den Wert Null oder Eins annehmen. Dies ermöglicht ein Umschalten von Peripheriegeräten.

WAIT INPUT,

Beispiel: WAIT GR_CLOSE=1

Die Eingabe (GR_CLOSE) wird für einen bestimmten Wert abgefragt (hier: 1). Der Satzvorschub erfolgt, sobald die Bedingung erfüllt ist. Damit wird das Programm mit den Peripheriegeräten synchronisiert. Das Programm wird in den Wartestatus versetzt, bis die Eingabe hoch (1) ist.

OUTPUT ON PATH

Beispiel: X=12.58 Gripper=1

Mit diesem Befehl können präzise Positionen während der Bewegung erreicht werden. Dieser Befehl wirkt sich immer nur auf den nachfolgenden Verfahrssatz aus. Das positionsbezogene Umschalten bezieht sich nur auf die definierte Achse (hier: X). Wenn die Achse die definierte Schalterposition (12,58) erreicht hat, erfolgt die Schaltung der definierten Ausgabe

(hier: auf 1). Die Ausgabe kann natürlich auch auf Null (grripper = 0) gesetzt werden.

6.1.4 Programmorganisationsbefehle

WAIT

Beispiel: WAIT 12.88

Mit diesem Befehl wird eine Wartezeit (in Sekunden) definiert. Die Programmverarbeitung wird für diese Zeit gestoppt.

COMMENT

Beispiel: (*Dies ist ein Kommentar*)

Mit diesem Befehl können Kommentarzeilen in das Programm eingefügt werden. Der Befehl wird sich nicht auf die Verarbeitung des Programms aus

Der Kommentartext kann 17 Zeichen enthalten. Zu Identifikationszwecken wird der eingegebene Kommentar automatisch durch Einfügezeichen ergänzt (*Kommentartext*).

CALL SUBROUT

Beispiel: CALL_SUB subGetTool

Mit diesem Befehl wird eine lokale Subroutine aufgerufen. Nach der Verarbeitung der Subroutine wird die Verarbeitung des Programms mit der nachfolgenden Programmzeile fortgesetzt.

CALL SUBPRG

Beispiel: CALL_PROG Sequence_1

Mit diesem Befehl wird ein anderes Programm aufgerufen, das aus dem Hauptspeicher der Steuerung geladen werden muss (wird im Folgenden als globale Subroutine bezeichnet). Nach der Verarbeitung der globalen Subroutine wird die Verarbeitung des Programms mit der nachfolgenden Programmzeile des aufrufenden Programms fortgesetzt. Eine globale Subroutine darf weder weitere globale Subroutinenaufrufe noch lokale Subroutinen beinhalten.

END PROGRAM

Beispiel: END_PROGRAM

Dieser Befehl steht für das Ende des Programms. Wenn diese Befehlszeile erreicht wird, ist die Verarbeitung des Programms abgeschlossen.

Wenn lokale Subroutinen verwendet werden, ist dieser Befehl obligatorisch, da lokale Subroutinen nur nach dem Befehl END_PROGRAM programmiert werden können.

BEGIN SUB

Beispiel: BEGIN subGetTool

Mit diesem Befehl wird eine lokale Subroutine in einem Programm gestartet.

Lokale Subroutinen können nur nach dem Befehl END_PROGRAM programmiert werden.

Eine lokale Subroutine darf weder globale Subroutinenaufrufe noch weitere lokale Subroutinen beinhalten.

END SUB

Beispiel: END_SUB

Mit diesem Befehl wird eine lokale Subroutine in einem Programm beendet.

Wenn diese Befehlszeile erreicht wird, wird die Verarbeitung der lokalen Subroutine abgeschlossen und die Verarbeitung des Programms wird nach Aufruf der Subroutine fortgesetzt.

IF INPUT THEN	Beispiel: IF GR_CLOSE=1 JMP LABEL12 Wenn die angegebene Eingabe (GR_CLOSE) dem Wert (hier: 1) entspricht, springt die Verarbeitung des Programms zu einem definierten Kennzeichen. Wenn die Bedingung nicht erfüllt wird, wird die Verarbeitung mit dem nächsten Befehl in der Sequenz fortgesetzt.
LABEL	Beispiel: LABEL12 Mit diesem Befehl wird ein Label definiert, das als Sprungziel im Programm dient.
JMP LABEL	Beispiel: JMP LABEL12 Die Verarbeitung des Programms springt zum angegebenen Label.
IF INT VAR =	Beispiel: IF INT12=23 LABEL13 Wenn die programmierte Bedingung erfüllt ist, springt die Verarbeitung zu dem definierten Label.
IF INT VAR >	Beispiel: IF INT12>23 LABEL13 Wenn die programmierte Bedingung erfüllt ist, springt die Verarbeitung zu dem definierten Label.
IF INT VAR <	Beispiel: IF INT12<23 LABEL13 Wenn die programmierte Bedingung erfüllt ist, springt die Verarbeitung zu dem definierten Label.
IF Ax_Pos >	Beispiel: IF X>12.90 LABEL12 Wenn die programmierte Bedingung erfüllt ist, springt die Verarbeitung zu dem definierten Label.
IF Ax_Pos <	Beispiel: IF X<12.90 Label12 Wenn die programmierte Bedingung erfüllt ist, springt die Verarbeitung zu dem definierten Label.