



OrangeApps

OrangeEdit 2.0 HowTo

Anwender Dokumentation

Stand: 23.07.2020, Version 0.3

© Copyright 2020

OrangeApps GmbH
Arnikaweg 1
87471 Durach
Deutschland
www.orangeapps.de

Diese Dokumentation darf –auch auszugsweise– vervielfältigt und Dritten zugänglich gemacht werden. Bei der auszugsweisen Vervielfältigung muss jedoch ein Verweis auf den Copyright Inhaber sowie dieses Dokument vermerkt werden.

Der Inhalt der Druckschrift wurde mit der beschriebenen Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass für die vollständige Übereinstimmung keine Gewähr übernommen werden kann.

Documentation validity

Version Dokumentation	Softwarestand		Freigabe	Datum
	Von	Bis		
0.2	2.0		Mayer	04.11.2018
0.3	2.0		Mayer	23.07.2021

Historie der Dokumentenversionen

Version	Datum	Autor	Änderungsgrund / Bemerkung
0.1	30.01.2015	Daniel Schmidt	Ersterstellung
0.2	06.05.2020	Christian Mayer	Allgemeine Überarbeitung
0.3	23.07.2021	Christian Mayer	Kap. 8 und 9 eingefügt

Inhalt

1	Einleitung	5
	1.1 Zielgruppe.....	5
	1.2 Darstellung von Hinweisen.....	5
	1.3 Verwendete Begriffe	5
2	Allgemein	6
	2.1 OrangeEdit Oberfläche (Einteilung).....	6
3	Funktionen „Extras“	7
	3.1 Positionen umbenennen	7
	3.2 Pdat und Ldat umbenennen	8
	3.3 Reihenfolge umkehren	9
	3.4 Block editieren	10
	3.5 Datenliste reinigen.....	12
	3.6 Positions-Transformation	13
	3.7 Positionen spiegeln.....	15
	3.8 Status/Turn anpassen.....	17
	3.9 Datenliste sortieren	19
	3.10 Array Assistent	20
	3.11 myHMI Designer.....	21
	3.12 Rechner (geometrische Operationen)	23
	3.13 Werkzeug und Basis Definitionen.....	26
	3.14 Langtext Editor.....	27
	3.15 Aktive Umgebung	28

4	Objectbrowser	30
5	Umgang mit Archiven	31
6	Umgang mit WorkVisual Projekten	32
7	Aktive Umgebung.....	33
8	Inlineformulare und KRL-Befehle.....	35
8.1	Inlineformulare	35
8.1.1	Beispiel: Inlineformular PTP – Bewegung	35
8.2	KUKA Standard Inlineformulare für KSS 8.5 oder höher.....	37
8.3	KRL-Befehle	37
9	Syntaxüberprüfung	39

1 Einleitung

1.1 Zielgruppe

Diese Dokumentation richtet sich an Anwender mit folgenden Kenntnissen:

- Anwender-Kenntnisse KUKA Roboterprogrammierung
- Anwender-Kenntnisse OrangeEdit 2.0

1.2 Darstellung von Hinweisen



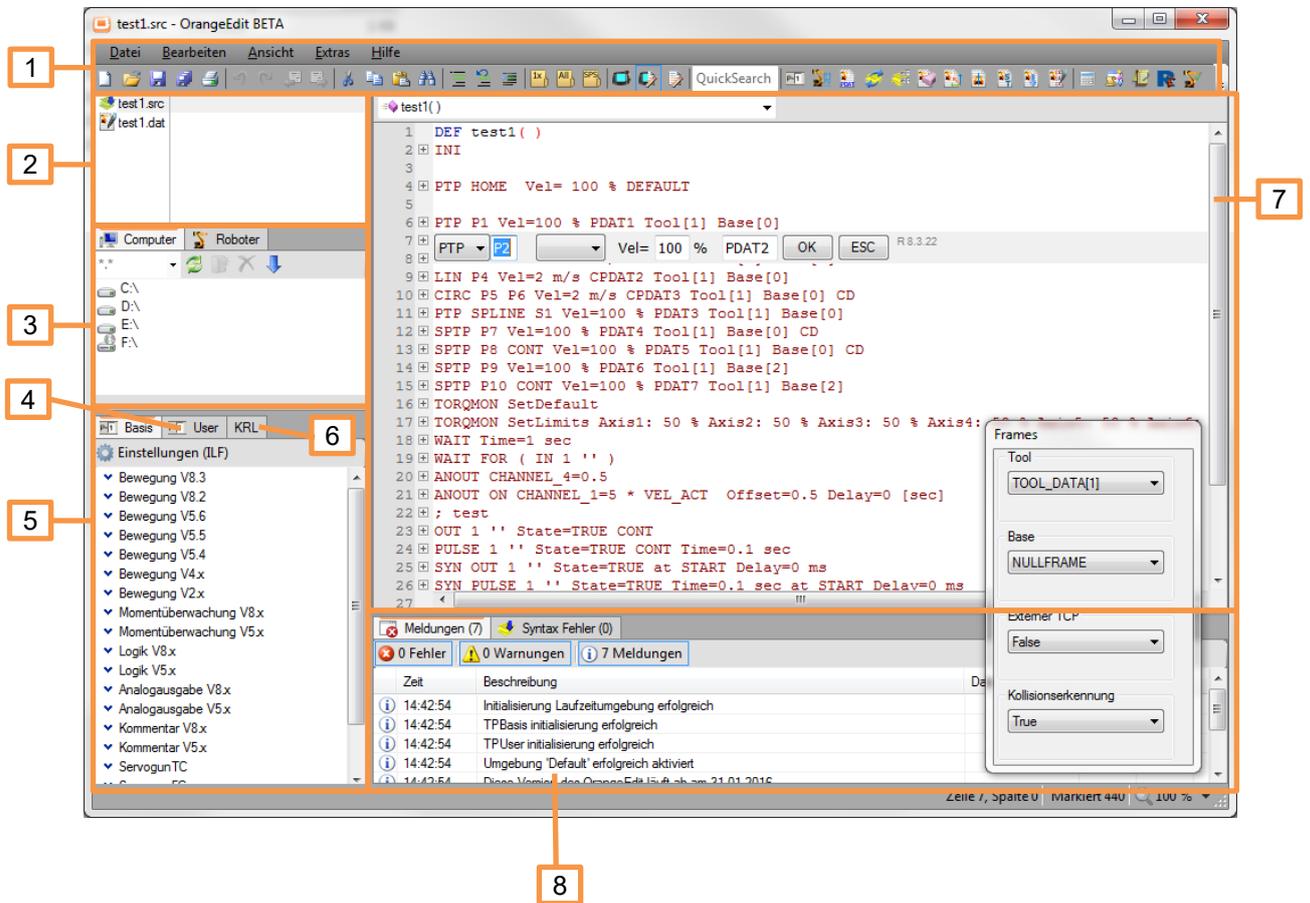
Diese Hinweise enthalten nützliche Tipps oder besondere Informationen für das aktuelle Thema.

1.3 Verwendete Begriffe

Begriff	Beschreibung
HMI	Die Human-Machine Interface (HMI) ist eine Schnittstelle, über die ein Mensch mit einer Maschine kommuniziert.
KSS	KUKA Systemsoftware
smartPad	Roboter Bediengerät
KRL	KUKA Roboter Language (KUKA Roboter Programmier-Sprache)
KFD	KUKA Form Description (Inlineformular Programmier-Sprache)
TPBasis	Standard Inlineformular-Befehle der Roboter-Oberfläche
TPUser	Inlineformulare die auf Basis UserTech in KFD entwickelt wurden

2 Allgemein

2.1 OrangeEdit Oberfläche (Einteilung)



1. Menü- und Werkzeug-Leiste
2. Liste der geöffneten Dateien
3. Explorer/Laufwerk-Fenster, unter der Registerkarte „Roboter“ befinden sich die geöffneten Archive
4. Registerkarte „User“ enthält UserTech Inlineformulare
5. Registerkarte „Basis“ enthält Basis (Standard) Inlineformulare
6. Registerkarte „KRL“ enthält KRL-Code-Vorlagen zum einfachen Einfügen
7. Programm-/Editorfenster
8. Meldungenfenster / Syntaxfehler-Anzeige

3 Funktionen „Extras“

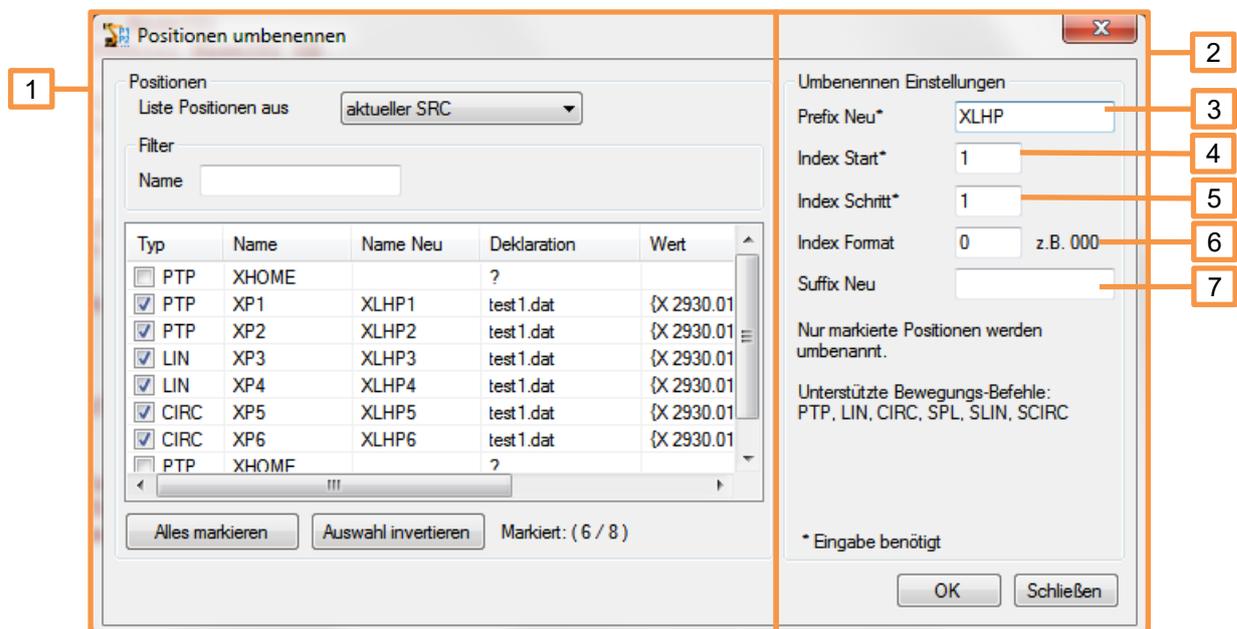
3.1 Positionen umbenennen

Diese Funktion erlaubt es, als Inlineformular angelegte Positionen, umzubenennen, bzw. neu zu nummerieren. Dies kann erforderlich sein, wenn in einem bestehenden Programm Positionen an beliebiger Stelle ergänzt wurden, oder die Benamung nicht dem Standard entspricht.

Aufruf

- Über Menü Extras -> Positionen umbenennen
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung



1. In diesem Teil des Fensters, werden die Positionen ausgewählt, welche umbenannt werden sollen. Verschiedene Filter helfen bei der Auswahl der gewünschten Punkte.
2. In diesem Teil des Fensters, werden alle Einstellungen die sich auf die Umbenennung auswirken getätigt.
3. Präfix Neu: Mit diesen Zeichen beginnt der neue Punktname.
4. Index Start: Für die Durchnummerierung der Punkte, wird ab dieser Eingabe begonnen.
5. Index Schritt: Um den Wert dieser Angabe wir für jeden Punkt hochgezählt.
6. Index Format: Soll der Nummerierung eine oder mehrere „0“ vorangestellt sein, kann über diese Eingabe das gewünschte Format festgelegt werden.
7. Suffix Neu: hier kann optional der neue Punktname mit einer fixen Zeichenfolge ergänzt werden.

Verwendung

- Roboter-Programm im OrangeEdit öffnen
- Funktion „Positionen umbenennen“ öffnen
- Gewünschte Positionen, die umbenannt werden sollen, selektieren

- Einstellungen für die Umbenennung vornehmen
- Mit „Ok“ den Vorrang starten



Der Vorgang der Umbenennung kann nicht rückgängig gemacht werden. Ein Backup des Programms sollte vorher erstellt werden.

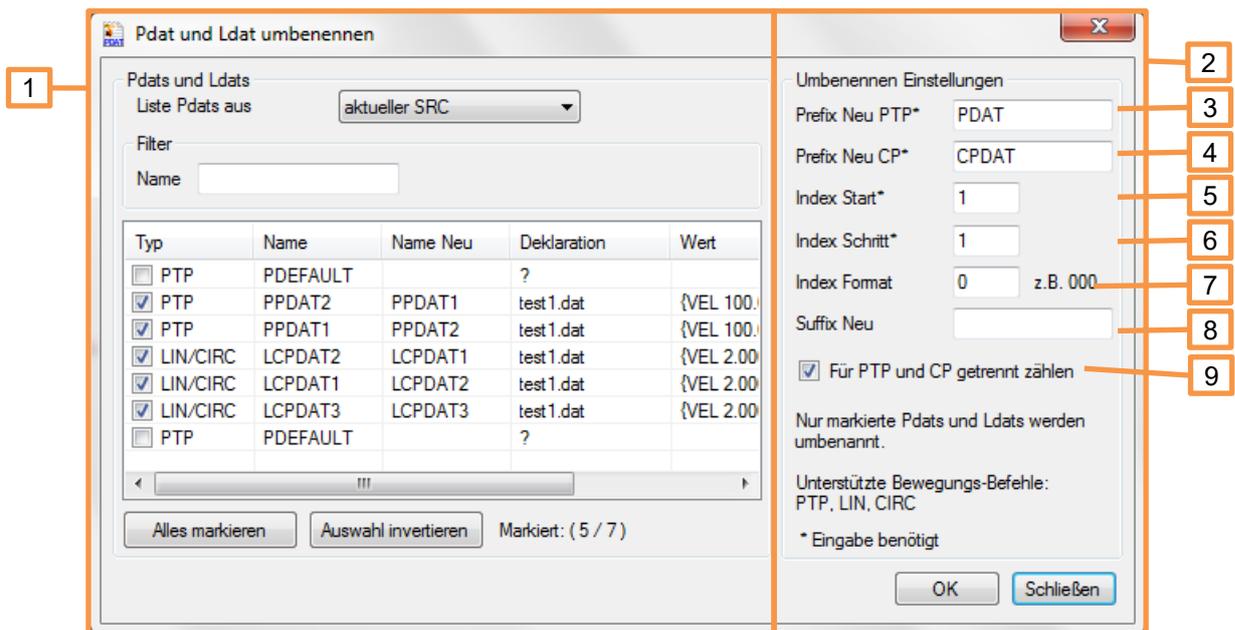
3.2 Pdat und Ldat umbenennen

Die Funktion „Pdat und Ldat umbenennen“ ermöglicht die Pdat und Ldat in entsprechenden Bewegungs-Inlineformularen umzubenenen. Dies kann erforderlich sein, wenn die Benennung nicht dem Standard entspricht, oder um einem Programm mehr „Ordnung“ zu geben.

Aufruf

- Über Menü Extras -> Pdat und Ldat umbenennen
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung



1. In diesem Teil des Fensters, werden die Positionen ausgewählt, welche umbenannt werden sollen. Verschiedene Filter helfen bei der Auswahl der gewünschten Punkte.
2. In diesem Teil des Fensters, werden alle Einstellungen die sich auf die Umbenennung auswirken getätigt.
3. Präfix Neu PTP: Mit dieser Zeichenfolge beginnt der neue Name der Pdat einer PTP-Bewegung.
4. Präfix Neu CP: Mit dieser Zeichenfolge beginnt der neue Name der Ldat einer LIN-, oder CIRC-Bewegung.
5. Index Start: Für die Durchnummerierung der Pdots und Ldots, wird ab dieser Eingabe begonnen.
6. Index Schritt: Um den Wert dieser Angabe wir für jede Pdat und Ldat hochgezählt.

7. Index Format: Soll der Nummerierung eine oder mehrere „0“ vorangestellt sein, kann über diese Eingabe das gewünschte Format festgelegt werden.
8. Suffix Neu: hier kann optional der neue Pdat-, oder Ldat-Name mit einer fixen Zeichenfolge ergänzt werden.
9. Ist diese Haken gesetzt, wird für Pdats, sowie Ldats, getrennt gezählt.

Verwendung

- Roboter-Programm im OrangeEdit öffnen
- Funktion „Pdat und Ldat umbenennen“ öffnen
- Gewünschte Positionen/Pdat und Ldat, die umbenannt werden sollen, selektieren
- Einstellungen für die Umbenennung vornehmen
- Mit „Ok“ den Vorrang starten

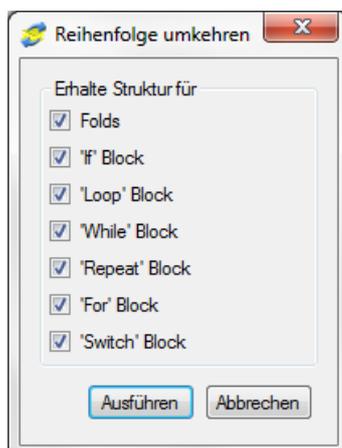
3.3 Reihenfolge umkehren

Die Funktion „Reihenfolge umkehren“ ermöglicht es, komplette Programme, oder Ausschnitte daraus, in der Reihenfolge der Abarbeitung umzukehren. Dies kann erforderlich sein, um z.B. einen programmierten Pfad rückwärts abzufahren (z.B. Roboterfalzen, Kleben).

Aufruf

- Über Menü Extras -> Reihenfolge umkehren
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung



- „Folds“, Foldstrukturen und deren Inhalte bleiben erhalten und werden nicht umgekehrt.
- „If“ Block, Kontrollstruktur für „IF, THEN, ELSE, ENDIF“ bleibt erhalten, Inhalte werden nicht umgekehrt.
- „Loop“ Block, Kontrollstruktur für „Loop, Endloop“ bleibt erhalten, Inhalte werden nicht umgekehrt.
- „While“ Block, Kontrollstruktur für „While, Endwhile“ bleibt erhalten, Inhalte werden nicht umgekehrt.

- „Repeat“ Block, Kontrollstruktur für „Repeat, Until“ bleibt erhalten, Inhalte werden nicht umgekehrt.
- „For“ Block, Kontrollstruktur für „For, Endfor“ bleibt erhalten, Inhalte werden nicht umgekehrt.
- „Switch“ Block, Kontrollstruktur für „Switch, Endswitch“ bleibt erhalten, Inhalte werden nicht umgekehrt.
- „Ausführen“, der Vorgang wird gestartet und in das Roboter-Programm übernommen.

Verwendung

- Roboter-Programm im OrangeEdit öffnen
- Bereich im Roboter-Programm markieren (z.B. mehrere Zeilen mit Bewegungspunkten)
- Funktion „Reihenfolge umkehren“ öffnen
- Einstellungen vornehmen
- Mit „Ausführen“ den Vorrang starten



Es empfiehlt sich, alle Haken gesetzt zu lassen, da ansonsten das Roboter-Programm, beim Hochladen auf die Roboter-Steuerung, Fehler aufweisen wird.



Wird zu Beginn kein Bereich im Roboter-Programm markiert, fragt die Funktion ab, ob das ganze Programm „umgekehrt“ werden soll.



Bei Wechseln von PTP zu LIN, bzw. bei CIRC-Bewegungen, wird nach dem Umkehren nicht die exakte Bahn rückwärts abgefahren. Hier muss von Hand am Roboter nachgearbeitet werden.

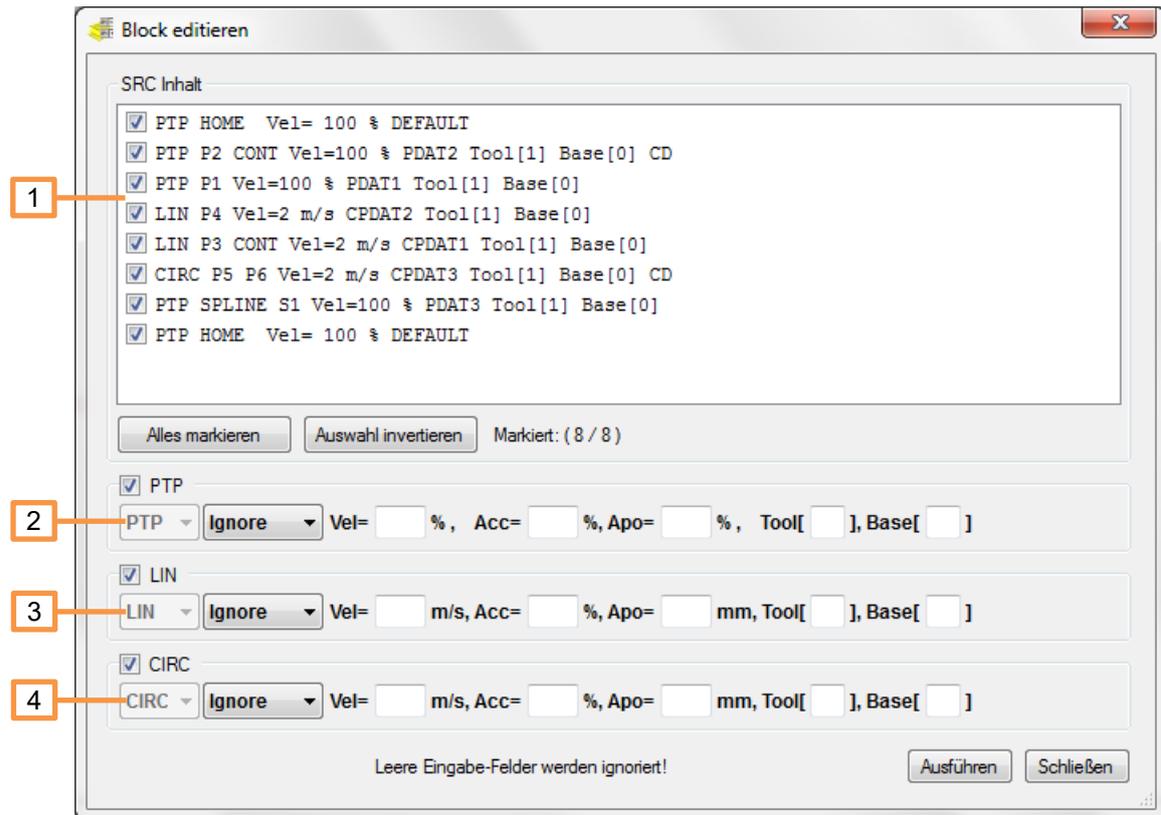
3.4 Block editieren

Die Funktion „Block editieren“ ermöglicht es, von mehreren Bewegungs-Inlineformularen, verschiedene Parameter, auf einmal zu ändern. Dies kann nützlich sein, wenn z.B. die Geschwindigkeit für eine ganze Reihe an Punkten, angepasst werden muss.

Aufruf

- Über Menü Extras -> Block editieren
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung



1. In dieser Liste, werden die Positionen ausgewählt, welche bearbeitet werden sollen.
2. Parameter für PTP Inlineformulare. Leere Felder in dieser Zeile werden ignoriert, die betreffenden Parameter in den Inlineformularen werden nicht geändert.
3. Parameter für LIN Inlineformulare. Leere Felder in dieser Zeile werden ignoriert, die betreffenden Parameter in den Inlineformularen werden nicht geändert.
4. Parameter für CIRC Inlineformulare. Leere Felder in dieser Zeile werden ignoriert, die betreffenden Parameter in den Inlineformularen werden nicht geändert.

Verwendung

- Roboter-Programm im OrangeEdit öffnen
- Bereich im Roboter-Programm markieren (z.B. mehrere Zeilen mit Bewegungspunkten)
- Funktion „Block editieren“ öffnen
- Einstellungen vornehmen
- Mit „Ausführen“ den Vorrang starten



Sollen alle Bewegungen in einem Programm angepasst werden, setzen Sie den Cursor an eine beliebige Stelle, ohne etwas zu markieren. Die Funktion „Block editieren“ listet nun alle Bewegungspunkte des aktuellen Programms auf.



Bei Tool- oder Base-Änderung, findet derzeit keine Transformation der Punktkoordinaten statt. Die geteachte Position bleibt unverändert.

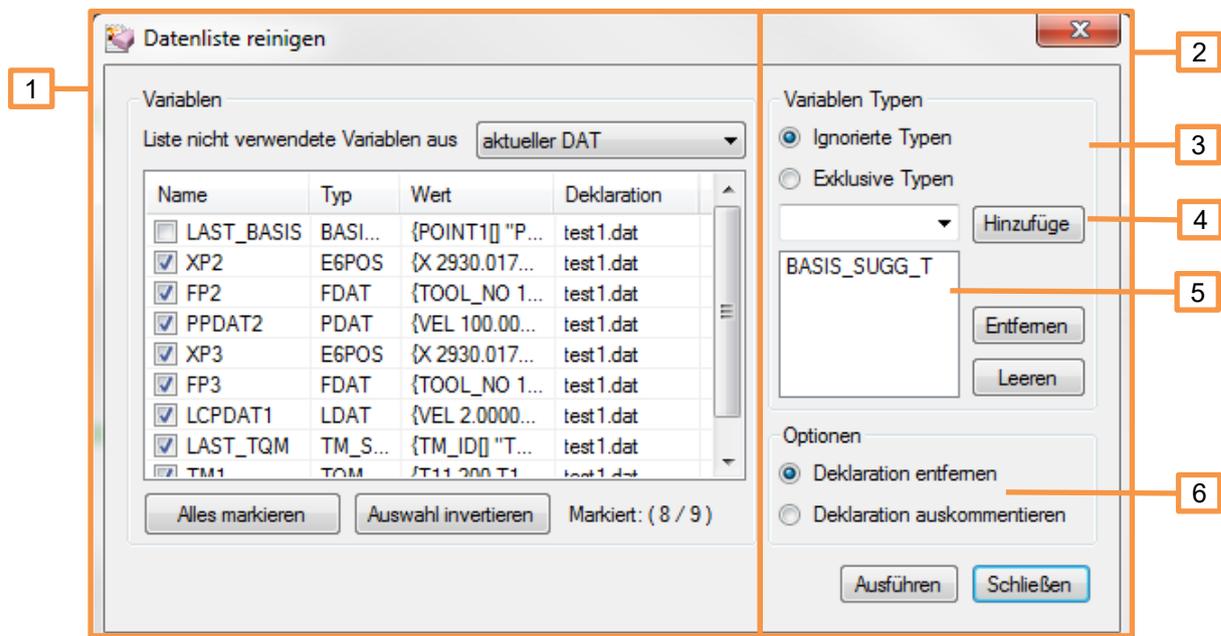
3.5 Datenliste reinigen

Die Funktion „Datenliste reinigen“ ermöglicht, nicht verwendete Variablen aus der Datenliste automatisch zu entfernen, oder zu kommentieren. Durch das Entfernen von nicht verwendeten Variablen, wird weniger Speicher für die jeweilige Datenliste benötigt.

Aufruf

- Über Menü Extras -> Datenliste reinigen
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung



1. In diesem Teil des Fensters, werden die Variablen ausgewählt, welche entfernt werden sollen. In der Liste werden nur **nichtverwendete Variablen angezeigt**.
2. In diesem Teil des Fensters, werden Filter, sowie Einstellungen für die Reinigung gemacht.
3. Hier wird festgelegt, ob die Variablen-Typen aus der Liste weiter unten, ignoriert werden sollen, oder exklusiv, nur diese Variablen-Typen entfernt werden sollen.
4. Hier können weitere Variablen-Typen, zur Filter-Liste unterhalb, hinzugefügt werden.
5. Liste mit Variablen-Typen die, je nach Einstellung, ignoriert werden sollen. Mit der Schaltfläche „Entfernen“ kann ein markierter Eintrag entfernt werden. Mit der Schaltfläche „Leeren“ werden alle Einträge der Liste entfernt.
6. Hier kann gewählt werden, ob die ausgewählten Variablen gelöscht, oder nur auskommentiert werden.

Verwendung

- Roboter-Programm im OrangeEdit öffnen
- Funktion „Datenliste reinigen“ öffnen
- Nicht verwendete Variablen werden in der Liste angezeigt
- Auswahl und Einstellungen vornehmen
- Mit „Ausführen“ den Vorrang starten

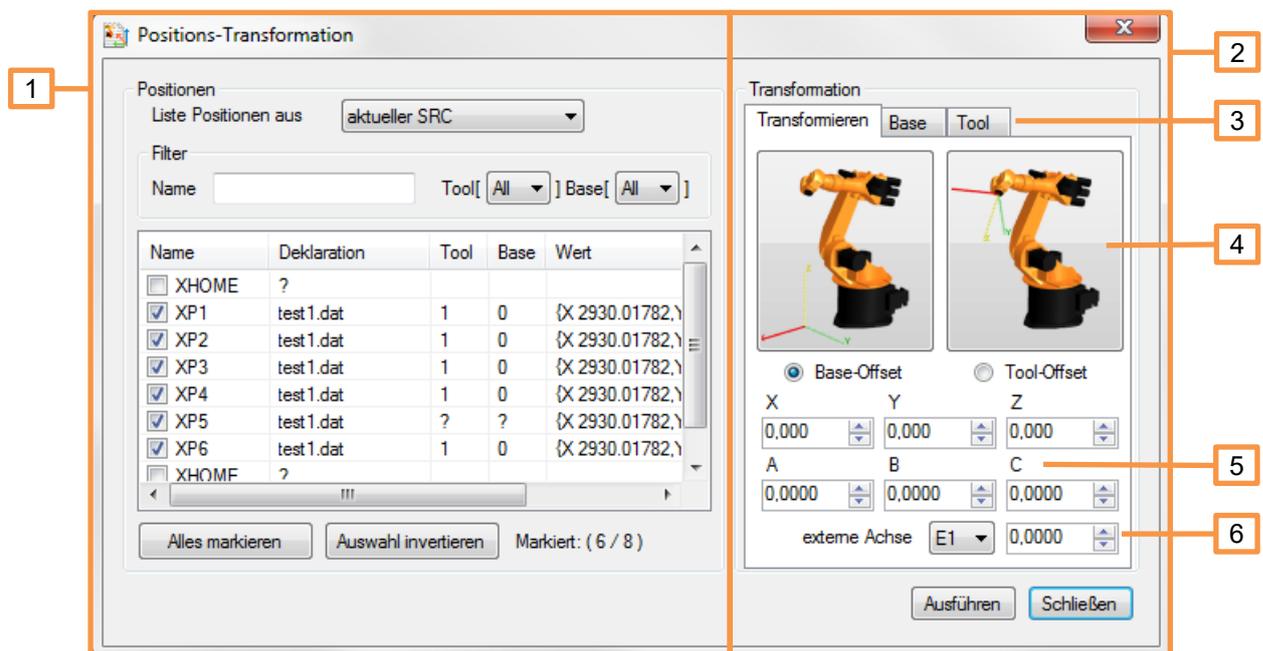
3.6 Positions-Transformation

Die „Positions-Transformation“ ermöglicht es, Punkte geometrisch zu transformieren (verschieben). Hierbei kann in Tool- oder Base-Richtung transformiert werden, weiter besteht die Möglichkeit, durch die Eingabe einer alten und einer neuen Base, bzw. eines alten oder neuen Tools, eine Transformation zwischen verschiedenen Koordinatensystem durchzuführen.

Aufruf

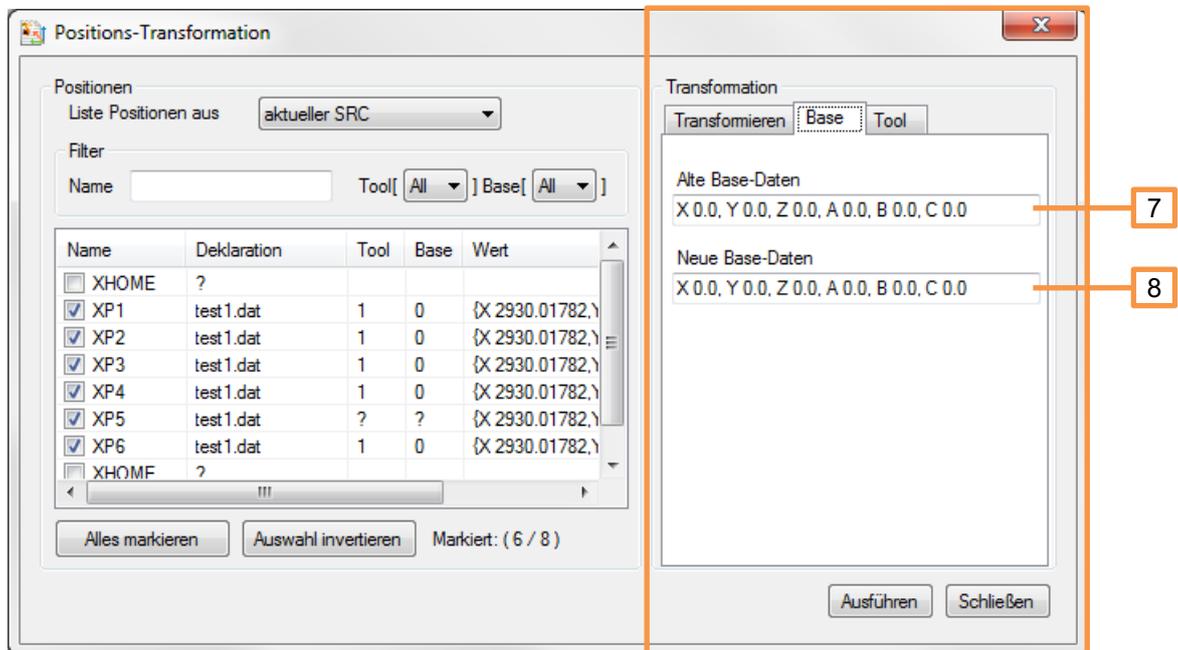
- Über Menü Extras -> Positions-Transformation
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung



1. In diesem Teil des Fensters, werden die Positionen ausgewählt, welche transformiert werden sollen. Verschiedene Filter helfen bei der Auswahl der gewünschten Punkte.
2. In diesem Teil des Fensters, werden alle Einstellungen die sich auf die Transformation auswirken getätigt.
3. Hier kann zwischen folgenden Transformations-Arten gewählt werden:
 - Transformieren: Positionen werden mit dem eingegebenen Wert transformiert.
 - Base: Hier werden die Positionen, um die Verschiebung von alter zu neuer Base, transformiert.

- Tool: Hier werden die Positionen, um die Verschiebung von altem zu neuem Tool, transformiert.
4. Richtung in welcher sich die Transformation auswirkt.
 5. Transformations-Werte in X, Y, Z, A, B, C.
 6. Optionaler Verschiebewert von E1 (externe Achse). Diese Verschiebung wird „nur“ zum E1-Wert der jeweiligen Position addiert.



7. Eingabefeld für die alten Base-Daten, mit welchen die Positionen geteacht wurden.
8. Eingabefeld für die neuen Base-Daten, in welches die Positionen transformiert werden sollen.

Verwendung

- Roboter-Programm im OrangeEdit öffnen
- Funktion „Positionen-Transformation“ öffnen
- Gewünschte Positionen, die transformiert werden sollen, selektieren
- Einstellungen für die Transformation vornehmen
- Mit „Ausführen“ den Vorrang starten



Sind die Positionen mit externem TCP geteacht und man möchte den externen TCP (z.B. Klebedüse) ändern, so muss die Transformation als Tool-Offset durchgeführt werden, bzw. die Transformations-Art „Tool“ gewählt werden. Soll bei externem TCP der Greifer-TCP verändert werden, so muss die Transformation als Base-Offset durchgeführt werden, bzw. die Transformations-Art „Base“ gewählt werden.



Mehr Informationen zum Thema „Transformation“ erhalten Sie beim Punkt „Rechner“ (geometrische Operationen)

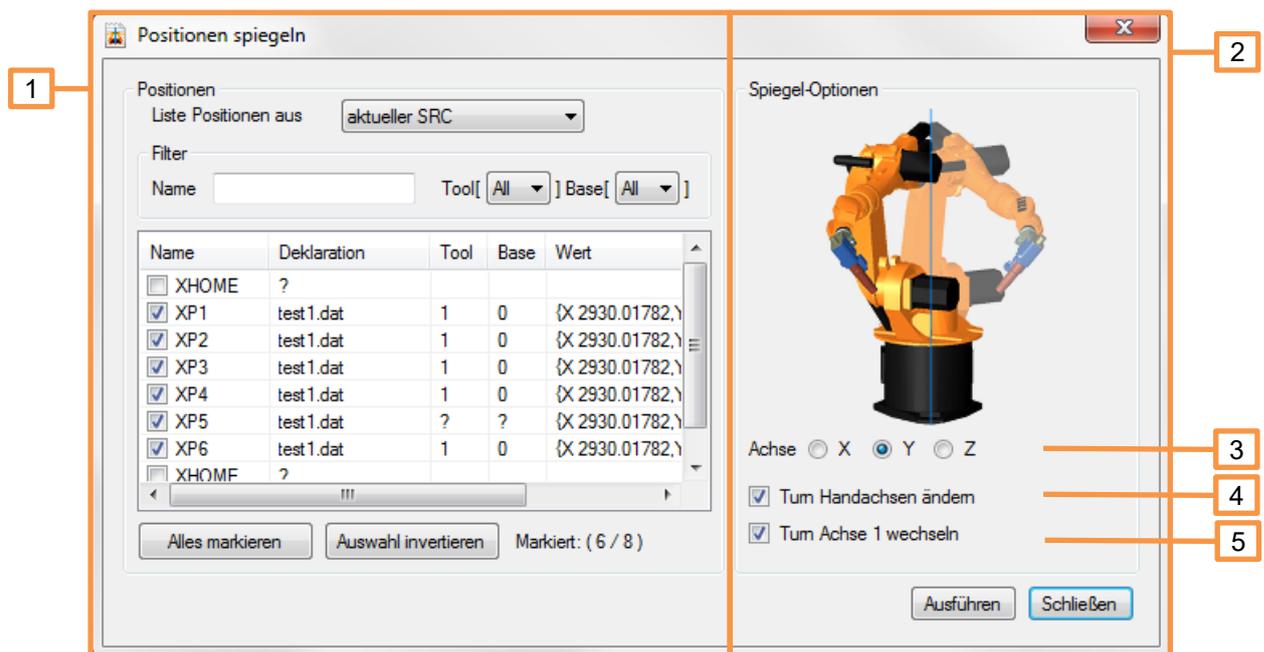
3.7 Positionen spiegeln

Die Funktion „Positionen spiegeln“ ermöglicht es, ausgewählte Positionen in einer wählbaren Ebene zu spiegeln. Dies kann erforderlich sein, um z.B. ein Programm für ein gespiegeltes Bauteil, unter Verwendung der Positionen des originalen Bauteils, zu erstellen.

Aufruf

- Über Menü Extras -> Positionen spiegeln
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung



1. In diesem Teil des Fensters, werden die Positionen ausgewählt, welche gespiegelt werden sollen. Verschiedene Filter helfen bei der Auswahl der gewünschten Punkte.
2. In diesem Teil des Fensters, werden alle Einstellungen die sich auf die Spiegel-Transformation auswirken getätigt.
3. Spiegel-Achse:
 - X → Positionen werden auf der Y-Z Ebene gespiegelt. Die X-Koordinate wechselt das Vorzeichen.
 - Y → Positionen werden auf der X-Z Ebene gespiegelt. Die Y-Koordinate wechselt das Vorzeichen. (Standard)
 - Z → Positionen werden auf der X-Y Ebene gespiegelt. Die Z-Koordinate wechselt das Vorzeichen.
4. Mit dieser Option werden die Turn-Bits der Handachsen invertiert.
5. Mit dieser Option wird das Turn-Bit der Achse 1 invertiert.



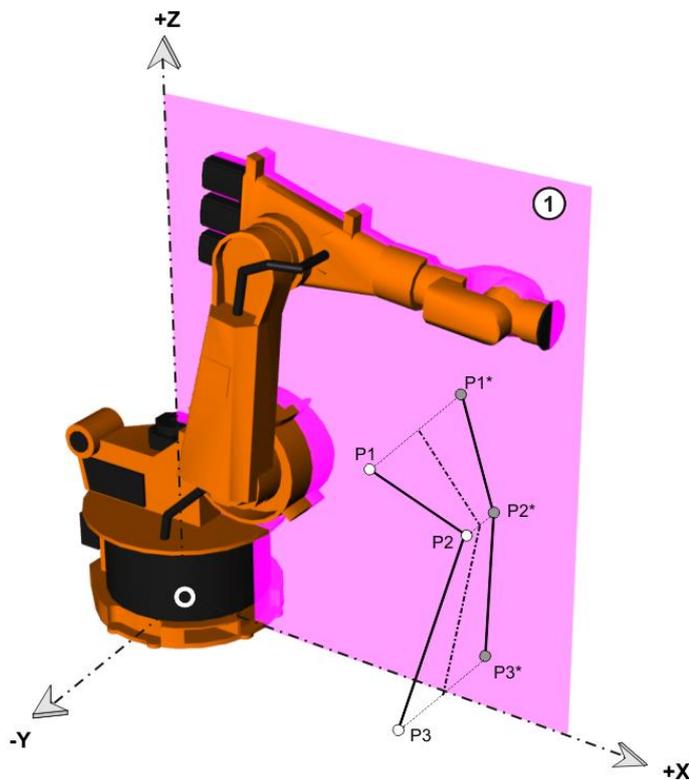
Mehr Informationen zum Thema „Turn-Bit“ erhalten Sie beim Punkt „Status/Turn anpassen“.

Verwendung

- Roboter-Programm im OrangeEdit öffnen
- Funktion „Positionen spiegeln“ öffnen
- Gewünschte Positionen, die gespiegelt werden sollen, selektieren
- Einstellungen für die Spiegelung vornehmen
- Mit „Ausführen“ den Vorrang starten

Beispiel

Die Punkte P1, P2, und P3 werden in „Y“ gespiegelt (X-Z Ebene). Die neuen Positionen der Punkte sind P1*, P2* und P3*.



Nach dem Achsspiegeln muss das verwendete Werkzeug ebenfalls in der X-Z-Ebene gespiegelt werden.



Da OrangeEdit das Roboter-Kinematik-Model nicht berücksichtigen kann, ist die Wahrscheinlichkeit gegeben, dass gespiegelte Punkte, mit der jeweiligen Achskonfiguration, nicht erreichbar sind. Es wird empfohlen, solche Punkte mit LIN anzufahren, da so die Status/Turn-Informationen nicht beachtet werden.

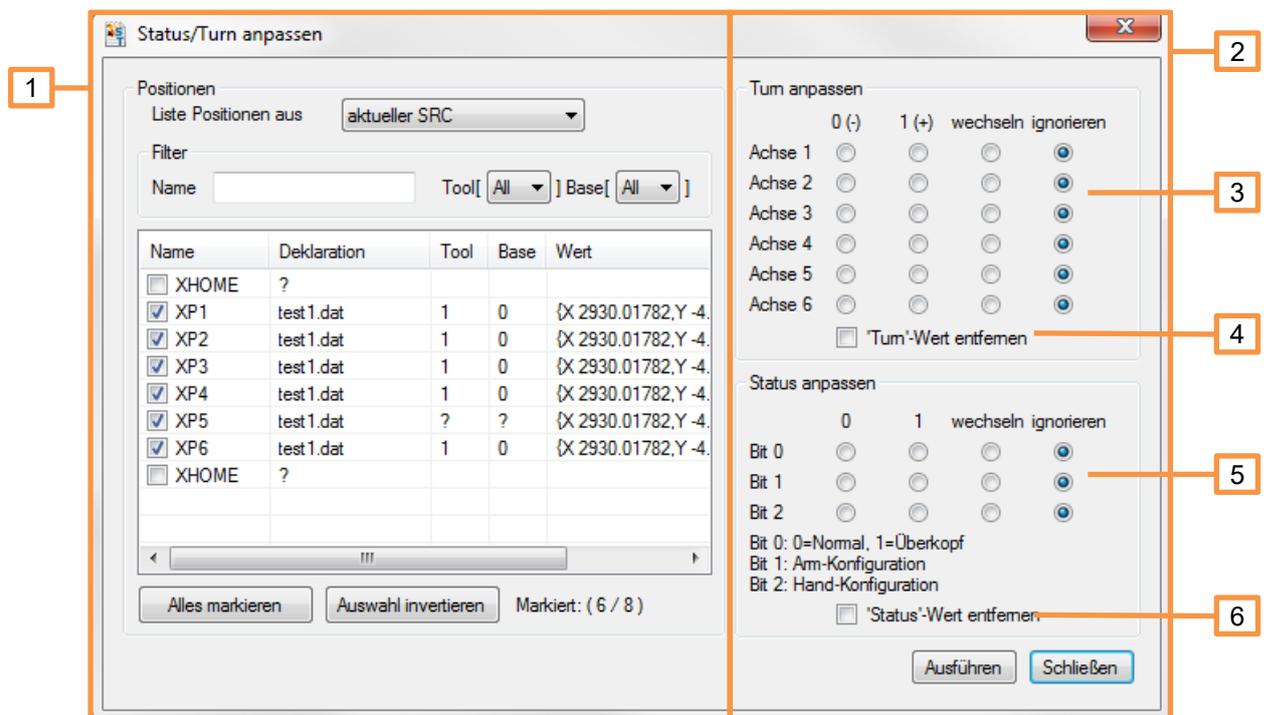
3.8 Status/Turn anpassen

Die Funktion „Status/Turn anpassen“ ermöglicht es auf einfache weiße die Status-, sowie Turn-Information in einer Position (E6POS Datentyp) anzupassen. Dies kann nach einer Transformation von Positionen erforderlich sein. Wird z.B. ein Roboter auf dem Sockel gedreht, kann es sein, dass selbst nach neuer Base-Vermessung, Positionen nicht erreichbar sind. Oftmals liegt dies am Vorzeichenwechsel der Achse 1 Position.

Aufruf

- Über Menü Extras -> Status/Turn anpassen
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung



1. In diesem Teil des Fensters, werden die Positionen ausgewählt, deren Status/Turn-Informationen angepasst werden sollen. Verschiedene Filter helfen bei der Auswahl der gewünschten Punkte.
2. In diesem Teil des Fensters, werden alle Einstellungen die sich auf Status und Turn beziehen getätigt.
3. Hier kann für jede Roboter-Achse eine Vorgabe für die Turn-Information gemacht werden.
 - 0(-), Turn-Information der Achse auf Minus festlegen (Achse-Wert ist < 0).
 - 1(+), Turn-Information der Achse auf Plus festlegen (Achse-Wert ist >= 0).
 - Wechseln, die Turn-Information wird invertiert. Aus 1 wird 0, aus 0 wird 1.
 - Ignorieren, die Turn-Information wird für diese Achse nicht verändert.
4. Die Turn-Information wird komplett aus der E6POS Deklaration entfernt (alle Achsen). Der Roboter wählt beim Anfahren nun den kürzesten Achsenwert (Achtung! Bewegungen des Roboters prüfen).
5. Hier kann das jeweilige Status-Bit angepasst werden (siehe Bilder weiter unten).
 - 0, die Status-Information wird auf den Wert 0 gesetzt.

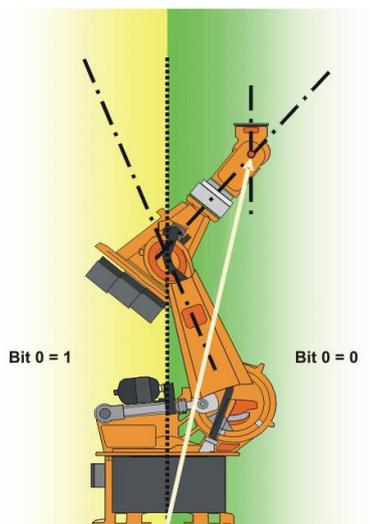
- 1, die Status-Information wird auf den Wert 1 gesetzt.
 - Wechseln, die Status-Information wird invertiert. Aus 1 wird 0, aus 0 wird 1.
 - Ignorieren, die Status-Information bleibt unverändert.
6. Die Status-Information wird komplett aus der E6POS Deklaration entfernt. Der Roboter wählt beim Anfahren nun den kürzesten Achsenwert (Achtung! Bewegungen des Roboters prüfen).

Verwendung

- Roboter-Programm im OrangeEdit öffnen
- Funktion „Status/Turn anpassen“ öffnen
- Gewünschte Positionen, die angepasst werden sollen, selektieren
- Einstellungen für die Anpassung vornehmen
- Mit „Ausführen“ den Vorrang starten

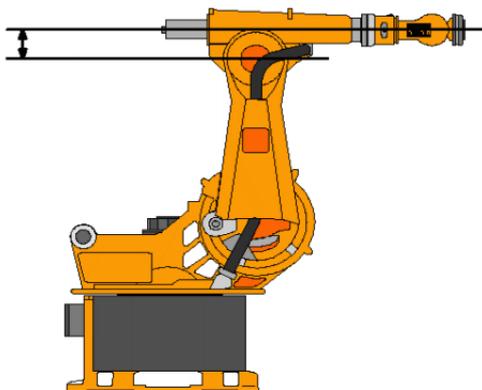
Status Bit 0

Die Status-Angabe verhindert Mehrdeutigkeiten bei der Achsstellung. Bit 0 gibt die Position des Schnittpunkts der Handachsen (A4, A5, A6) an.



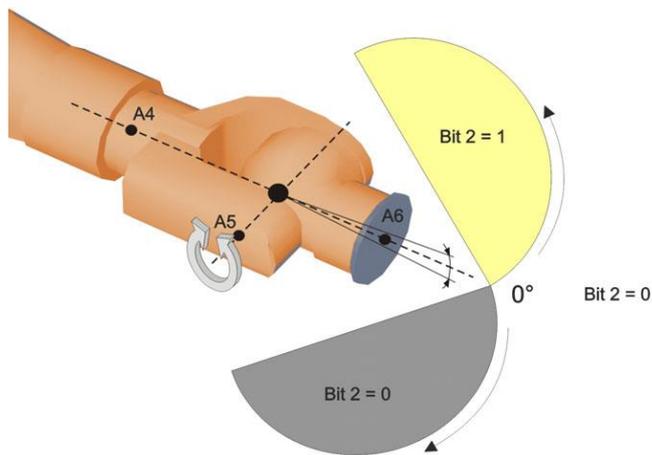
Status Bit 1

Bit 1 gibt die Position von Achse 3 an. Der Winkel, bei dem sich der Wert von Bit 1 ändert, ist abhängig vom Robotertyp.



Status Bit 2

Bit 2 gibt die Position von Achse 5 an. Das Vorzeichen des Winkels von A5 ist nicht ausschlaggebend! Ausschlaggebend ist die Richtung, in die die A5 gekippt ist, also nach oben oder unten. Die Richtungsangabe wiederum bezieht sich auf die Nullstellung der A4.



Mehr Informationen zum Thema „Status/Turn“, erhalten Sie in der KUKA Dokumentation:

KUKA System Software 8.3 Bedien- und Programmieranleitung für Systemintegratoren >> Kapitel 9.10

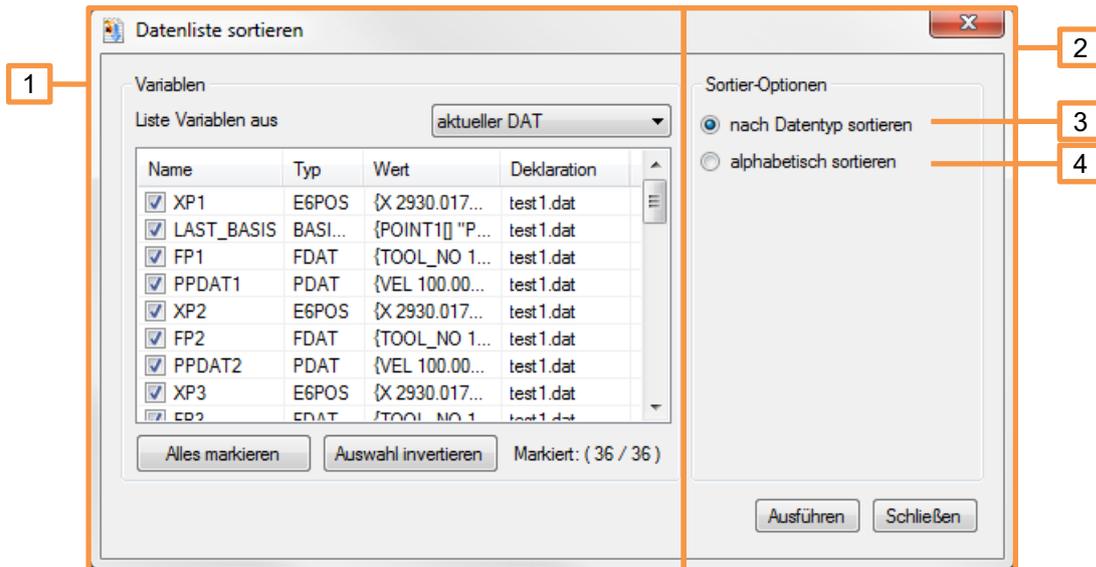
3.9 Datenliste sortieren

Mit der Funktion „Datenliste sortieren“, ist es möglich, die Deklarationen in einer Datenliste zu sortieren. Dies bringt etwas mehr Ordnung und Lesbarkeit in eine Datenliste. Die Funktion des Roboters bleibt davon unberührt.

Aufruf

- Über Menü Extras -> Datenliste sortieren
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung



1. In diesem Teil des Fensters, werden die Variablen ausgewählt, welche sortiert werden sollen.
2. In diesem Teil des Fensters, werden alle Einstellungen die sich auf die Sortierung auswirken getätigt.
3. „nach Datentyp sortieren“, alle ausgewählten Variablen werden nach Datentyp gruppiert und sortiert.
4. „alphabetisch sortieren“, alle ausgewählten Variablen werden alphabetisch geordnet und sortiert.

Verwendung

- Roboter-Programm im OrangeEdit öffnen
- Funktion „Datenliste sortieren“ öffnen
- Gewünschte Variablen, die sortiert werden sollen, selektieren
- Einstellungen für die Sortierung vornehmen
- Mit „Ausführen“ den Vorrang starten

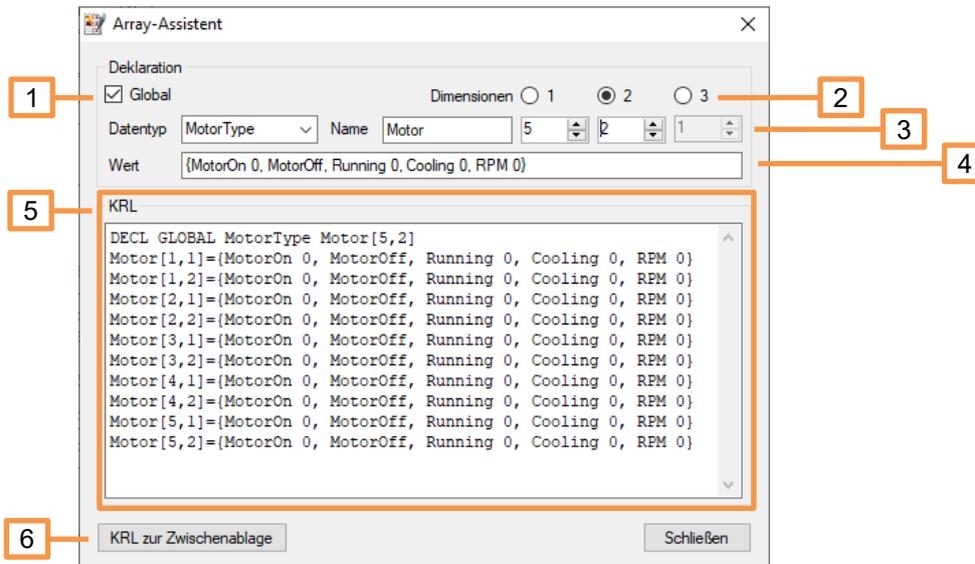
3.10 Array Assistent

Der Array Assistent erzeugt die Deklaration von ein- bis dreidimensionalen Arrays und erspart dadurch das mühevoll händische Tippen.

Aufruf

- Über Menü Extras -> Array Assistent
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung



1. Deklaration als globale oder lokale Variable.
2. Dimension des Arrays
3. Datentyp und Benennung der Array-Variablen
4. Werte der Elemente des Array
5. Darstellung des Array in KRL
6. Kopiert das Array in die Zwischenablage

3.11 myHMI Designer

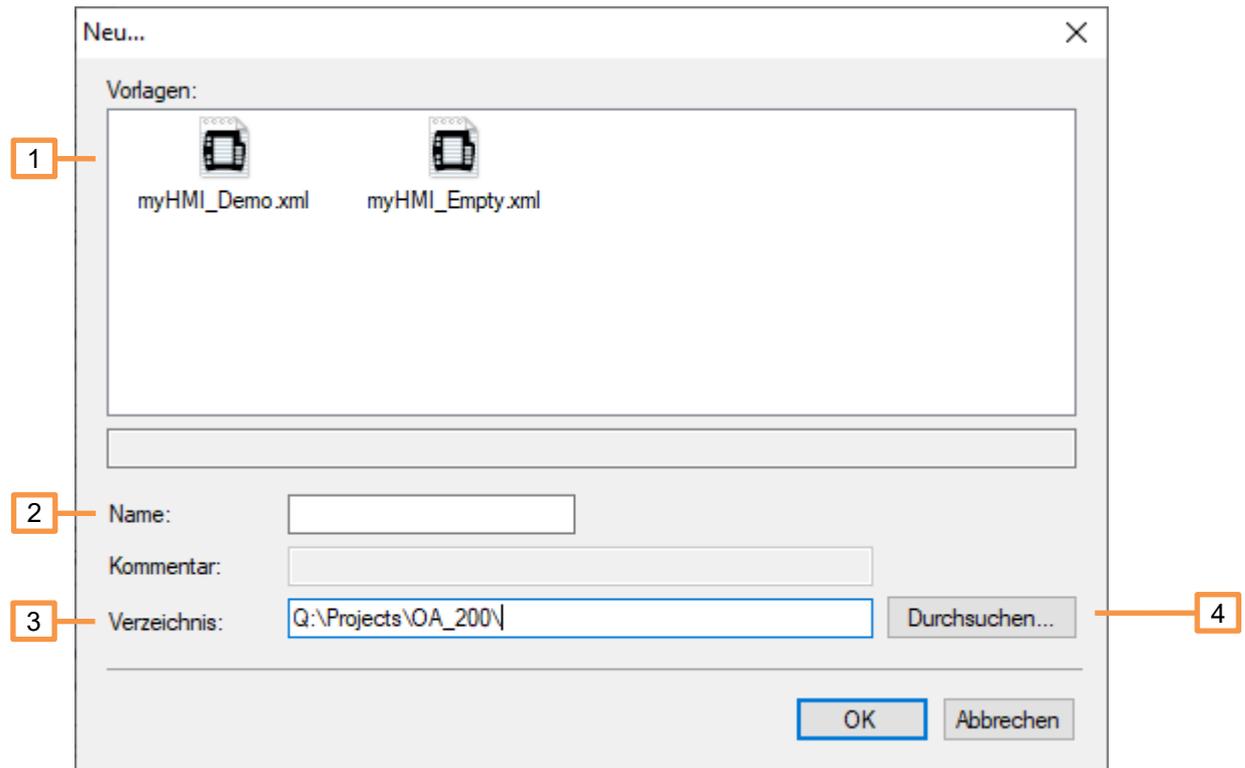
Designer zum Erstellen von Benutzeroberflächen für die Software „myHMI“.

Aufruf

- Über Menü Extras -> myHMI Designer
- Über das entsprechende Symbol in der Werkzeugleiste

Nach der Auswahl dieses Befehls öffnet sich ein Dateidialog zur Auswahl einer bereits bestehenden Oberfläche oder eines der beiden implementierten Templates.

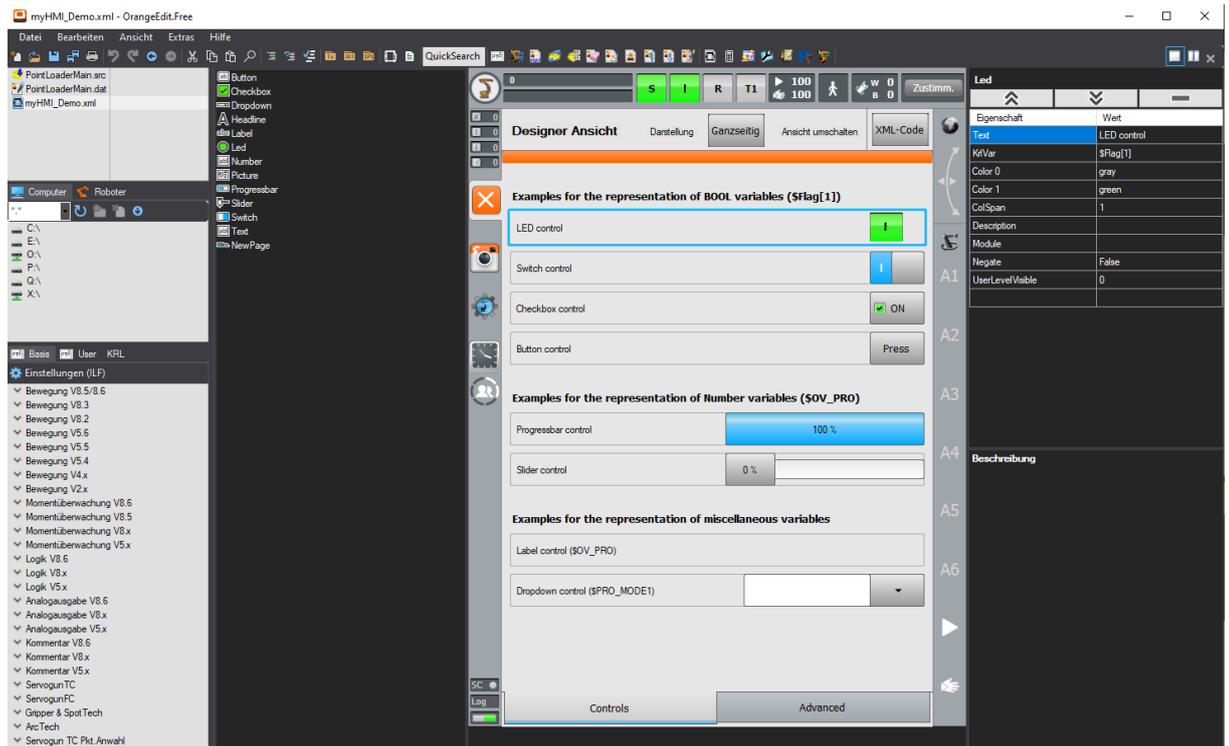
Darstellung Dateiauswahldialog



1. Template auswählen.
2. Namen der HMI eintragen (keine Leerzeichen erlaubt)
3. Dateipfad für bestehende Oberfläche
4. Ordnerauswahldialog für bestehende Oberfläche

Nach der Wahl eines Template oder eine bestehender Oberfläche öffnet sich das Designer-Fenster.

Darstellung



Eine Beschreibung der Bedienung befindet sich in der Dokumentation zur Software „myHMI“.

Link zur Software und Anleitung: <https://orangeapps.de/?lng=de&page=apps%2Fmyhmi>

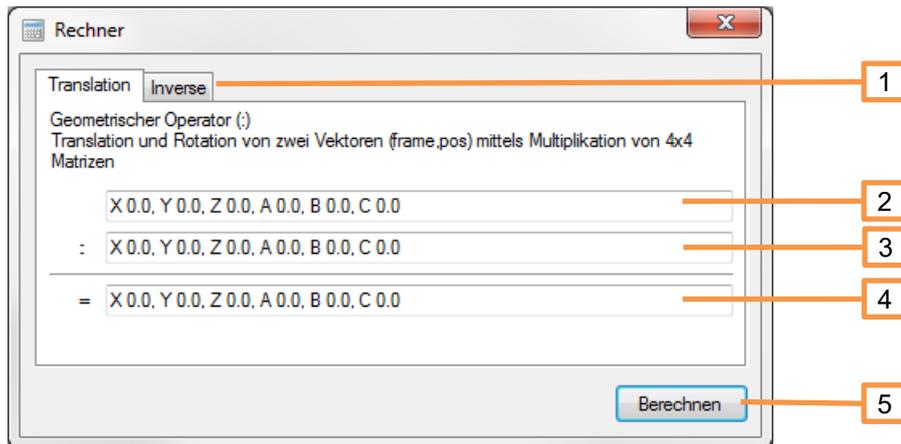
3.12 Rechner (geometrische Operationen)

Mit dem „Rechner“ im OrangeEdit, können geometrische Operationen durchgeführt werden. So kann z.B. ein Offset auf einen TCP gerechnet werden, eine Base-Verschiebung und vieles mehr.

Aufruf

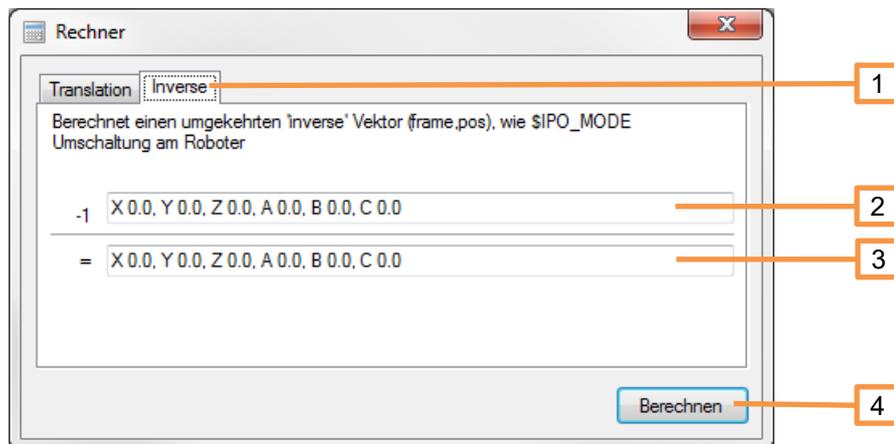
- Über Menü Extras -> Rechner
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung (Transformation)



1. Umschalten zwischen Transformation und Inverse.
2. Eingabefeld für Vektor 1.
3. Eingabefeld für Vektor 2.
4. Ergebnis der Transformation.
5. „Berechnen“-Schaltfläche, löst die Berechnung aus.

Darstellung (Inverse)



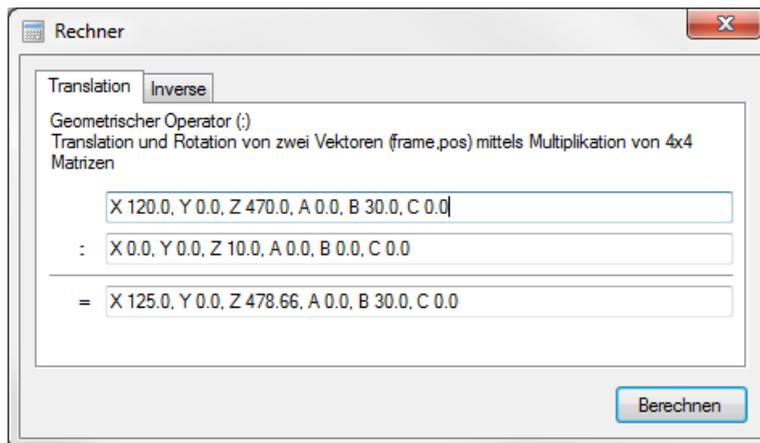
1. Umschalten zwischen Transformation und Inverse.
2. Eingabefeld für den Vektor.
3. Ergebnis (Inverse) des eingegebenen Vektors .
4. „Berechnen“-Schaltfläche, löst die Berechnung aus.

Beispiel 1 (Offset auf TCP addieren)

Sie möchten auf einen TCP eines Werkzeugs in Z-Richtung des Werkzeug 10mm addieren. Das Werkzeug ist 30° gewinkelt am Roboter-Flansch angebracht.

Tool = X 120.0, Y 0.0, Z 470.0, A 0.0, B 30.0, C 0.0

Geben Sie die Daten das erste Feld ein und die gewünschte Verschiebung in das Zweite.



Als Ergebnis erhalten Sie neue Tool-Daten für Ihr Werkzeug, welchem in Richtung Z 10 mm addiert wurden.

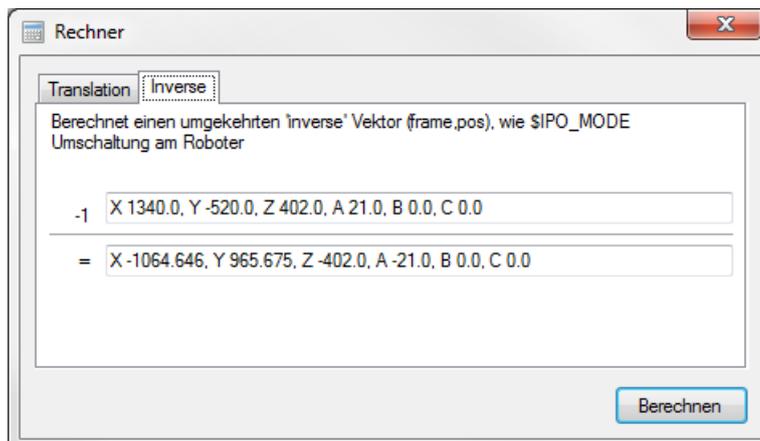
Beispiel 2 (Differenz zwischen zwei Base-Daten ermitteln)

Sie haben ein Base neu vermessen, da sich eine Bearbeitungs-Station verschoben hat. Nun möchten Sie wissen, wie groß die Differenz in Bezug zur alten Base ist.

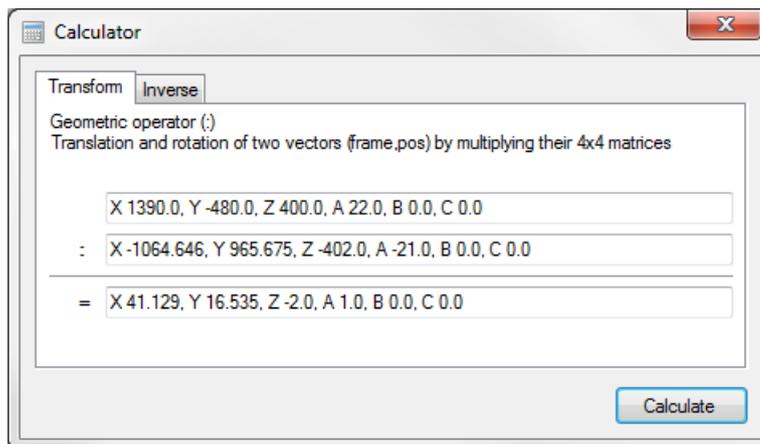
Alte Base: X 1390.0, Y -480.0, Z 400.0, A 22.0, B 0.0, C 0.0

Neue Base: X 1340.0, Y -520.0, Z 402.0, A 21.0, B 0.0, C 0.0

1. Berechnen Sie die Inverse der neuen Base.



2. Tragen Sie die alte Base in das erste Feld bei Translation ein, das Ergebnis der Inverse in das zweite Feld und klicken Sie „Berechnen“.



Als Ergebnis erhalten Sie die Differenz des alten Base zur Neuen im Koordinaten-Netz des alten Base.

Diesen Wert kann nun als Transformations-Vektor zum Verschieben von Positionen verwenden.

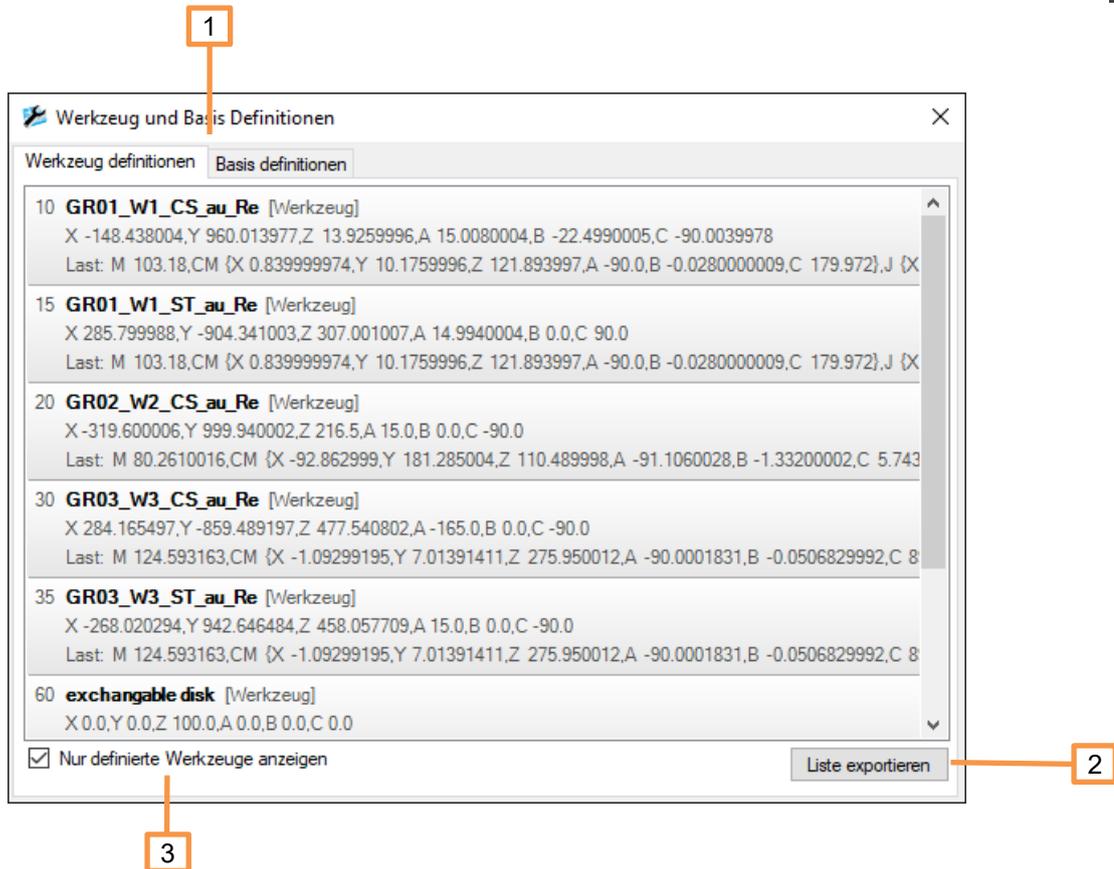
3.13 Werkzeug und Basis Definitionen

Zeigt die Werkzeug- und Basisdaten der aktiven Umgebung an

Aufruf

- Über Menü Extras -> Rechner
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung



1. Umschalten zwischen Werkzeug und Basis Definitionen
2. Exportiert die Liste in eine CSV-Datei
3. Blendet nur definierte Werkzeuge oder Basissysteme ein

3.14 Langtext Editor

Der Langtext Editor ermöglicht die Bearbeitung der Langtexte für Eingänge, Ausgänge, Analoge E/As, Timer, Zähler und Flags. Die Langtexte werden KRC1/2 kompatibel in einer Access-Datenbank gespeichert und können KRC2 und KRC4 (WorkVisual) kompatibel exportiert werden.

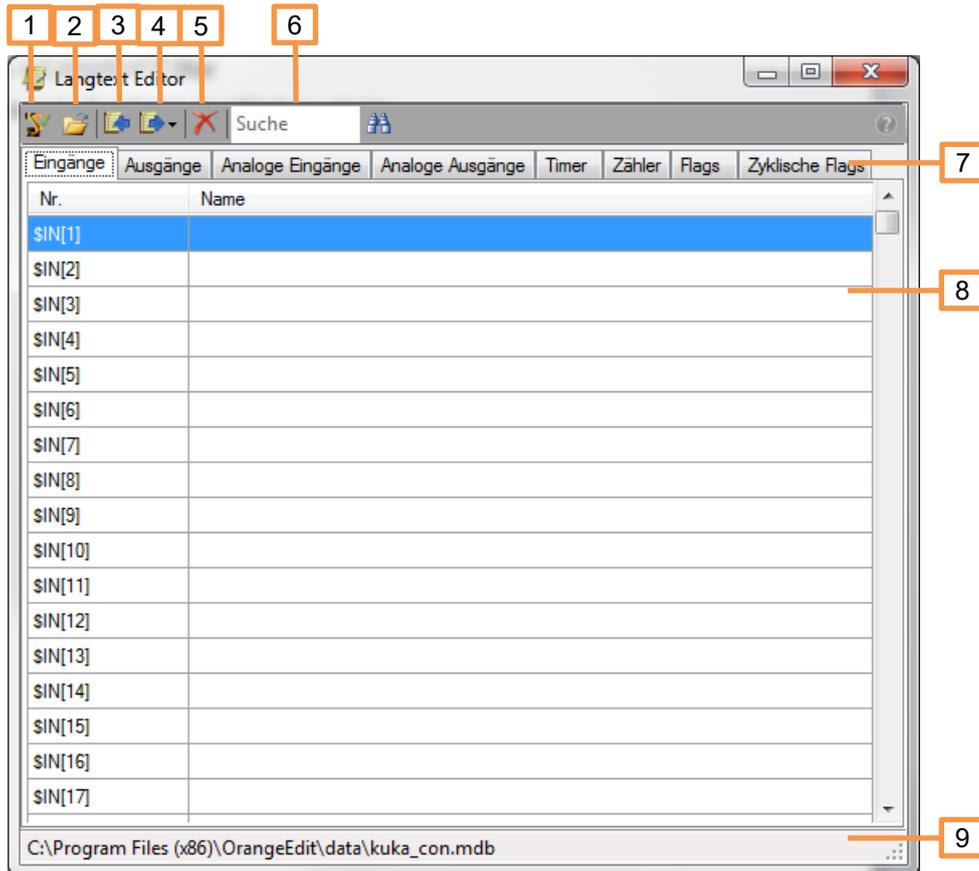
Merkmale

- Öffnen und Bearbeiten von Langtext-Datenbanken von einem KRC1/2/4 Roboter
- Importieren von Langtexten aus .csv oder .txt Datei
- Exportieren von Langtexten nach .csv oder .txt
- Leeren (alles löschen) einer Langtext-Datenbank
- Suche

Aufruf

- Über Menü Extras -> Langtext Editor
- Über das entsprechende Symbol in der Werkzeuggestreife

Darstellung



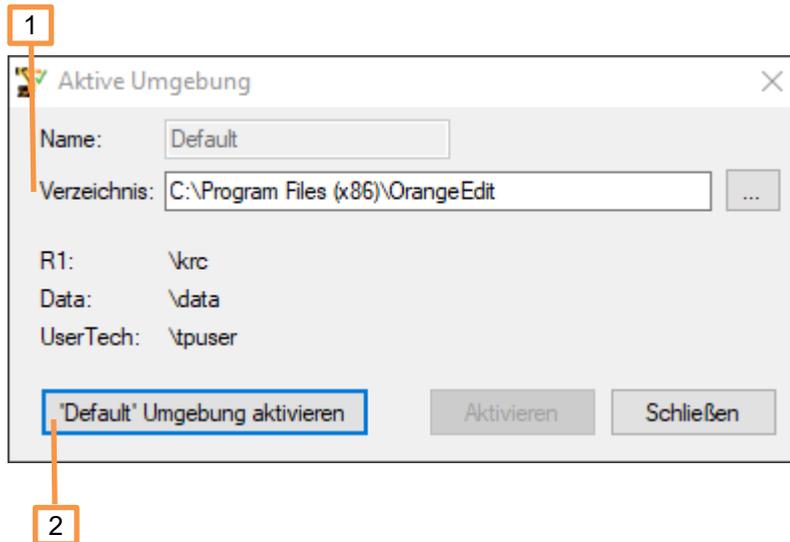
1. Langtexte aus aktiver Umgebung laden
2. Langtext-Datenbank laden (.mdb)
3. Langtexte aus .txt oder .csv importieren
4. Langtexte nach .txt oder .csv exportieren
5. Datenbank leeren (alle Langtexte aus Datenbank löschen)
6. Suche, ermöglicht die Suche nach einem Langtext
7. Registerkarten für die unterschiedlichen Langtexte
8. Ansicht- und Bearbeitungs-Liste der Langtexte
9. Pfad zur aktuell geladenen Langtext-Datenbank

3.15 Aktive Umgebung

Zeigt die aktuelle aktive Umgebung an oder setzt diese auf einen auswählbaren Ordner. In diesem Ordner wird nach dem Ordner "KRC" und dann nach den Ordnern „R1“, „DATA“ und „TPUSER“ gesucht.

Aufruf

- Über Menü Extras -> aktive Umgebung
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung

1. Zeigt das aktuelle Verzeichnis der aktiven Umgebung
2. Aktiviert die Default Umgebung C:\Program Files (x86)\OrangeEdit

Es ist nicht notwendig, dass sich im ausgewählten Ordner alle Unterordner befinden, d.h. wenn sich dort z.B. nur ein R1 Verzeichnis befindet, dann wird nur dieses Verzeichnis aktiviert.

4 Objectbrowser

Zeigt Funktionen, Variablen, Strukturen und Aufzählungstypen an. Für manche Variablen wird ein Beschreibungstext angezeigt. Über einen Filter ist es möglich nach Werten zu suchen.

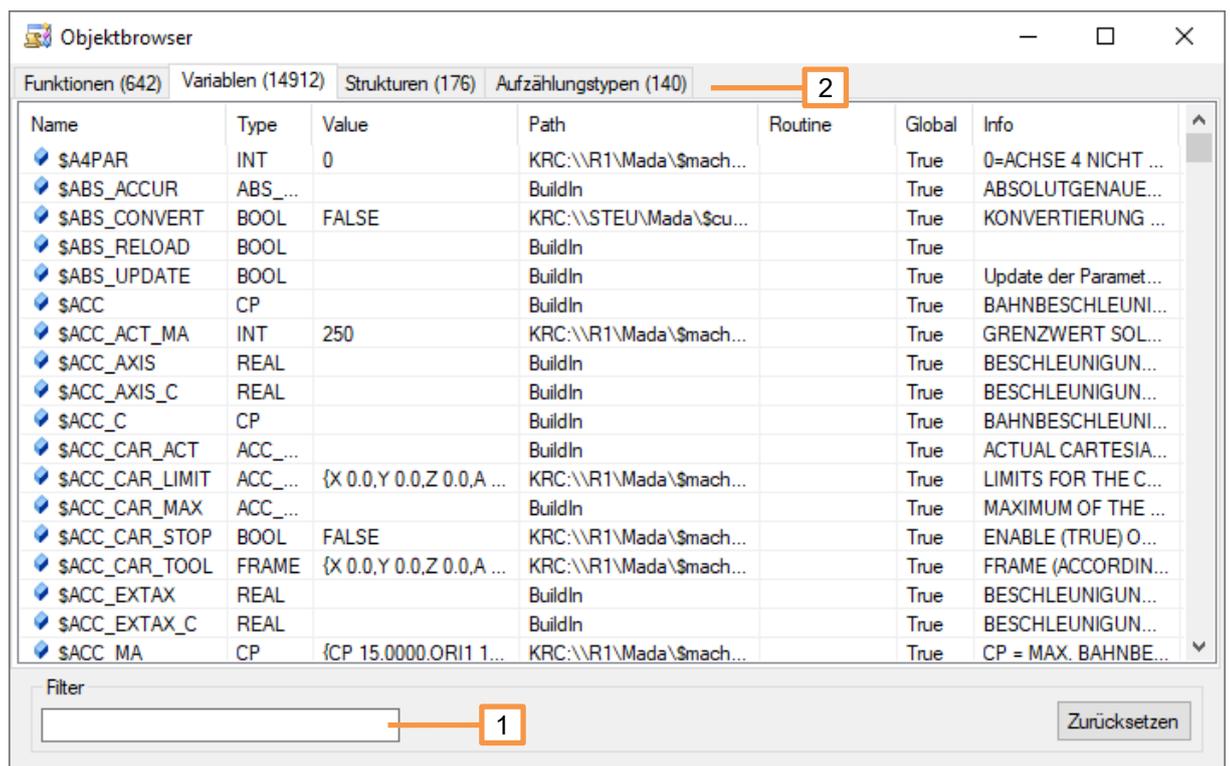
Merkmale

- Anzeige von System- und benutzerdefinierten Funktionen, Variablen, Strukturen und Aufzählungstypen
- Suche

Aufruf

- Über Menü Ansicht -> Objectbrowser
- Über das Symbol  in der Werkzeugleiste

Darstellung



1. Suche, ermöglicht die Suche nach einem Objekt
2. Registerkarten für die unterschiedlichen Objekte

5 Umgang mit Archiven

Der OrangeEdit unterstützt das Öffnen, Bearbeiten, sowie Speichern von Archiven. Über die Menüfolge Datei->Öffnen oder durch Drag&Drop auf das Fenster mit den geöffneten Dateien oder das Explorer Fenster, können ZIP Archive geladen werden.

Was passiert beim Öffnen

- Das Archiv wird vollständig in das Temporäre-Verzeichnis (%temp%) entpackt und im OrangeEdit angezeigt.

Was passiert beim Speichern eines Archivs

- Alle Dateien aus dem zuvor entpackten Archiv, werden zurück in eine ZIP-Datei gepackt.



Wurden Dateien aus dem Archiv im OrangeEdit geändert und gespeichert ist es zusätzlich notwendig das Archiv zu speichern, da ansonsten die geänderten Dateien nicht im Archiv enthalten sind.

Was passiert beim Schließen eines Archivs

- Nach dem Dialog, ob das Archiv gespeichert werden soll, werden alle Dateien und Ordner die beim Öffnen/Entpacken erstellt wurden, wieder gelöscht.

6 Umgang mit WorkVisual Projekten

Der OrangeEdit unterstützt das Öffnen und Bearbeiten von WorkVisual Projekten (*.wvs und *.asz). Über die Menüfolge Datei->Öffnen oder durch Drag&Drop auf das Fenster mit den geöffneten Dateien oder das Explorer Fenster, können die Dateien geladen werden.

Was passiert beim Öffnen

- Das Archiv wird vollständig in das Temporäre-Verzeichnis (%temp%) entpackt und im OrangeEdit angezeigt.

Was passiert beim Speichern eines Archivs

- Es werden keine Dateien aus dem zuvor entpackten Projekt in das Projekt zurückgespeichert.



Wurden Dateien aus dem Archiv im OrangeEdit geändert und gespeichert ist es zusätzlich notwendig die geänderten Dateien manuell in ein Verzeichnis Ihrer Wahl zu speichern.

Was passiert beim Schließen eines Archivs

- Es erscheint ein Dialog der zum Speichern geänderter Dateien auffordert.

7 Aktive Umgebung

Eine „aktive Umgebung“ bezeichnet ein Verzeichnis oder ein Archiv, welches vom OrangeEdit vollständig eingelesen und geladen wird. Alle darin enthaltenen globalen Variablen oder Funktionen, stehen Ihnen mit der Syntaxvervollständigung zur Auswahl zur Verfügung. Weiter werden die Base-, und Tool-Namen für die Inlineformulare geladen. Ebenso werden, sofern vorhanden, alle UserTech-Inlineformulare als Befehle unter „User“ zur Verfügung gestellt. Wenn es sich bei der aktiven Umgebung um ein KRC1/2 Archiv handelt, dann wird auch die Langtext-Datenbank geladen, dies ermöglicht das Erstellen von Logik-Befehlen als Inlineformular, inklusive der Langtexte.

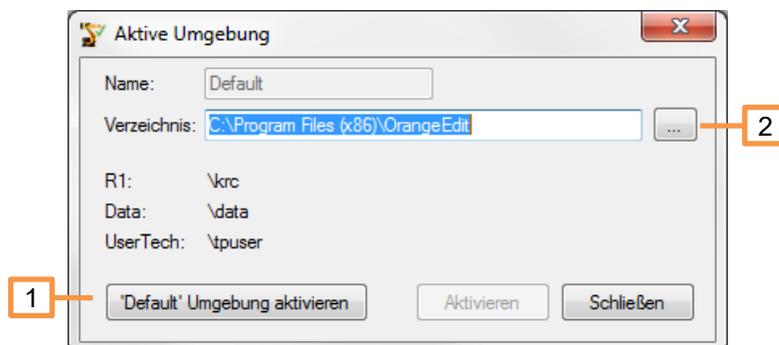
Eine Umgebung besteht aus folgenden Teilen:

- KRC-Verzeichnis, enthält alle KRL-Dateien (*.src, *.dat).
- Data-Verzeichnis, enthält bei KRC1/2 die Datei „KUKACON.MDB“, eine Datenbank mit den jeweiligen Langtexten.
- UserTech, enthält Inlineformulare (.kfd-Dateien) die im OrangeEdit unter der Seite „User“ aufgelistet werden.

Aufruf

- Über Menü Extras -> Aktive Umgebung
- Über das entsprechende Symbol in der Werkzeugleiste

Darstellung (Transformation)



1. „Default“ Umgebung laden. Lädt als aktive Umgebung, dass in der Installation mitgelieferte KRC Verzeichnis (z.B. C:\Programme\OrangeEdit\Krc)
2. Hier kann ein beliebiges Verzeichnis als aktive Umgebung geladen werden. Das Verzeichnis muss mindestens einen KRC Ordner enthalten, sonst erscheint eine Fehlermeldung.

Default Umgebung

Die Default Umgebung lädt folgende Verzeichnisse im Installations-Ordner von OrangeEdit:

- KRC: ..\OrangeEdit\krc\
- Data: ..\OrangeEdit\data\
- UserTech: ..\OrangeEdit\tpuser\



Die Ordner der Default-Umgebung können mit eigenen Inlineformulare, sowie KRL-Dateien ergänzt werden. Die Inlineformulare, sowie Variablen und Funktionen, stehen dann im OrangeEdit zur Verfügung.

Achtung: bei einem Update des OrangeEdit, werden vorhandene Dateien überschrieben. Hier sollte vorher eine Kopie gemacht werden.

Archiv als aktive Umgebung

Ein KUKA-Roboter Archiv kann ebenso als aktive Umgebung festgelegt werden.

Vorgehensweise (Archiv öffnen)

- Archiv im OrangeEdit öffnen.
- Den dann erscheinenden Dialog „Dieses Archiv als aktive Umgebung setzen?“ mit „Ja“ beantworten.

Vorgehensweise (bereits geöffnetes Archiv)

- Rechtsklick auf das Archiv (Roboter-Symbol).
- „als aktive Umgebung setzen“ klicken.

8 Inlineformulare und KRL-Befehle

8.1 Inlineformulare

Inlineformulare erleichtern das Programmieren des Roboters. Dazu stellt OrangeEdit zahlreiche Inlineformulare zur Verfügung. Durch Doppelklick auf das gewünschte Inlineformular wird dieses in das Programm eingefügt und, je nach Inlineformular, ein entsprechender Datensatz in der Datenliste angelegt, falls nicht bereits vorhanden.

In Programmen bestehende Inlineformulare werden durch Doppelklick geöffnet und können durch den Benutzer bearbeitet und gespeichert werden.

Zum Bearbeiten der Parameter in den Inlineformularen werden die Daten aus der aktiven Umgebung oder von geöffneten Datenlisten verwendet.

Fenster Inlineformulare



Aufruf des Fensters ILF

Im Fenster links unten auf „Basis“ klicken

8.1.1 Beispiel: Inlineformular „PTP – Bewegung“

Im Beispielprogramm wird ein PTP-Bewegungspunkt eingefügt.

1. Programm öffnen

```

1  DEF Modul ( )
2  ⊕ INI
3
4  ⊕ PTP HOME Vel= 100 % DEFAULT
5
6
7  ⊕ PTP HOME Vel= 100 % DEFAULT
8
9  END

```

2. Cursor auf Einfügestelle setzen und durch Doppelklick auf PTP einen PTP-Punkt einfügen

^ Bewegung V8.5/8.6	
<input type="checkbox"/> PTP	SPTP 8.5/8.6
<input type="checkbox"/> PTP	SLIN 8.5/8.6
<input type="checkbox"/> PTP	SCIRC 8.5/8.6
<input checked="" type="checkbox"/> PTP	PTP 8.5/8.6
<input type="checkbox"/> PTP	LIN 8.5/8.6
<input type="checkbox"/> PTP	CIRC 8.5/8.6

→ der PTP Punkt wird eingefügt

```

1  DEF Modul ( )
2  ⊕ INI
3
4  ⊕ PTP HOME Vel= 100 % DEFAULT
5  |
6  ⊕ PTP P1 Vel=100 % PDAT1 Tool[0] Base[0]
7
8  ⊕ PTP HOME Vel= 100 % DEFAULT
9
10 END

```

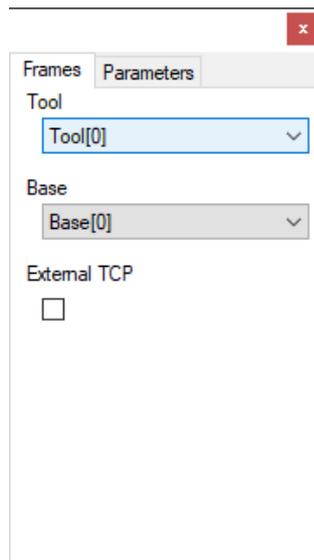
3. Der Punkt wird in der Datenliste des Programms mit allen benötigten Parameter angelegt. Diese können durch Öffnen des Inlineformulars bearbeitet werden. Das Inlineformular wird mit einem Doppelklick auf die Befehlszeile geöffnet.

```

1  DEF Modul ( )
2  ⊕ INI
3
4  ⊕ PTP HOME Vel= 100 % DEFAULT
5
6  ⊕     [%]    
7
8  ⊕ PTP HOME Vel= 100 % DEFAULT
9
10 END

```

4. Durch Anklicken der jeweiligen Parameterfelder können alle parameter geändert werden. Dazu öffnen sich ggf. weitere Fenster.



8.2 KUKA Standard Inlineformulare für KSS 8.5 oder höher

Ab der Steuerungsversion 8.5 haben sich die KUKA Standard Inlineformulare deutlich geändert. Damit diese im OrangeEdit benutzt werden können, ist ein kostenpflichtiges Plugin notwendig.

Zu den Standard Inlineformularen gehören:

- Bewegung: SPTP, SLIN, SCIRC, PTP, LIN, CIRC
- Momentenüberwachung: SetDefault, SetLimits, Off, SaveMax, UseDataSet
- Logik: Out, Pulse, SynOut, SynPulse, Wait, WaitFor
- Analogausgabe: Static, Dynamic
- Kommentar: Kommentar, Stempel

Auf unserer Webseite stehen weitergehende Informationen zur Verfügung:

<https://orangeapps.de/?lng=de&page=apps%2Forangeditilf>

8.3 KRL-Befehle

OrangeEdit stellt vordefinierte KRL-Befehle zur Verfügung. Durch Doppelklick werden diese in das Programm eingefügt.

Aufruf des Fensters

Im Fenster links unten auf „KRL“ klicken

Fenster KRL-Befehle

Routine
Function
FOR loop
GOTO
IF block
INTERRUPT
LOOP
REPEAT loop
SWITCH block
TRIGGER Distance
TRIGGER Path
WHILE loop
PTP XYZ
PTP Axis
PTP Rel
Wait Sec
Wait for FALSE
Signal IN
Signal OUT
Notify message (simple)
Notify message (advanced)
Quit message (simple)
Quit message (advanced)
State message (simple)
State message (advanced)
Dialog message (simple)
Dialog message (advanced)
Loop message
Loop message reset

9 Syntaxüberprüfung

OrangeEdit verfügt über eine eingeschränkte Syntaxprüfung. Gefundene Fehler werden durch einen roten Balken markiert und es wird eine entsprechende Meldung im Meldungsfenster erzeugt.

Geprüft werden:

- Vollständigkeit von Kontrollstrukturen
- Vollständigkeit von Klammern
- Anweisungen außerhalb des Anweisungsteils
- Deklaration von Übergabeparametern



Für die Richtigkeit der Überprüfung wird keinerlei Gewährleistung übernommen.