

RoboticHelpTool

Manual

1. Handeingabe

- **Reset** → Resetten der Eingabemaske
- **Umrechnen** → Umrechnen der eingegebenen Location in die anderen Formate
- **RelBase XYZ** → Verschiebung relativ zur Base.
Die auf der rechten Seite eingegebene Location wird relativ zur benutzten Base um die Parameter X/Y/Z verschoben.
Es öffnet sich ein neues Fenster in dem die Parameter X/Y/Z eingegeben werden müssen. Es müssen alle Parameter benannt werden. Mit dem OK-Button werden die Parameter übernommen.
Es darf nur ein Location Format komplett ausgefüllt sein!
Die anderen Location Formate werden automatisch gefüllt.
- **RelTool XYZ** → Verschiebung relativ zum Tool
Die auf der rechten Seite eingegebene Location wird relativ zum benutzten Tool um die Parameter X/Y/Z verschoben.
Es öffnet sich ein neues Fenster in dem die Parameter X/Y/Z eingegeben werden müssen. Es müssen alle Parameter benannt werden. Mit dem OK-Button werden die Parameter übernommen.
Es darf nur ein Location Format komplett ausgefüllt sein!
Die anderen Location Formate werden automatisch gefüllt.
- **Mirror X/Z** → Spiegeln über die X/Z Achsen
Die auf der rechten Seite eingegebene Location wird über die Achsen X und Z gespiegelt.
Es darf nur ein Location Format komplett ausgefüllt sein!
Die anderen Location Formate werden automatisch gefüllt.

- **Transform** → Transformieren der eingegeben Location
 - Die auf der rechten Seite eingegeben Location wird mit der anderen Location multipliziert. (Transformation)
 - Wie mit dem Doppelpunkt-Operator im KUKA.
 - Es öffnet sich ein neues Fenster in der eine andere Location in den Formaten KUKA, ABB oder Transformation Matrix eingegeben werden können. Mit dem „Daten einlesen“-Button werden die Parameter übernommen.
 - Es darf in beiden Fenstern nur ein Location Format komplett ausgefüllt sein!
 - Die anderen Location Formate werden automatisch gefüllt.

- **TCP Calc** → Berechnen von TCP's
 - **KUKA OnePos / ABB OnePos**
 - Der Roboter muss mit einem Werkzeug, mit dem gewünschten TCP auf eine Messspitze oder ähnliches gefahren werden, auf der eine Base eingemessen wurde.
 - Die Werte dieser Position müssen genutzt werden.
 - In den obersten Feldern muss die benutzte Base eingegeben werden.
 - In den mittleren Feldern muss die Flanschposition in dieser Position eingegeben werden. Es wird eine höhere Genauigkeit erreicht, wenn ein Tool mit Flansch-TCP aber mit richtigen Gewichtsdaten benutzt wird.
 - Nach drücken auf „Daten einlesen“ wird der TCP in den unteren Feldern angezeigt.
 - Es ist eventuell sinnvoll die Orientierung von anderen Quellen zu benutzen.

2. KUKA Dateieingaben

Über Drag'n Drop oder manuelle Eingabe muss eine .dat Datei in das Textfeld beschrieben werden.

- **.dat einlesen** → die im Textfeld angegeben .dat wird eingelesen
- **ausgeben** → Von der eingelesenen Datei werden die Location mit dem im System festgelegten Editor geöffnet.
- **RelBase XYZ** → Verschiebung relativ zur Base.
Die Locations der eingelesenen Datei werden relativ zur zur benutzten Base um die Parameter X/Y/Z verschoben.
Es öffnet sich ein neues Fenster in dem die Parameter X/Y/Z eingegeben werden müssen. Es müssen alle Parameter benannt werden. Mit dem OK-Button werden die Parameter übernommen.
Die Locations werden mit dem im System festgelegten Editor geöffnet.
Wenn gewünscht, können diese Locations in die .dat eingesetzt werden, die zum einlesen benutzt wurde.
- **RelTool XYZ** → Verschiebung relativ zum Tool
Die Locations der eingelesenen Datei werden relativ zum benutzten Tool um die Parameter X/Y/Z verschoben.
Es öffnet sich ein neues Fenster in dem die Parameter X/Y/Z eingegeben werden müssen. Es müssen alle Parameter benannt werden. Mit dem OK-Button werden die Parameter übernommen.
Die Locations werden mit dem im System festgelegten Editor geöffnet.
Wenn gewünscht, können diese Locations in die .dat eingesetzt werden, die zum einlesen benutzt wurde.
- **Mirror X/Z** → Spiegeln über die X/Z Achsen
Die Locations der eingelesenen Datei werden über die Achsen X und Z gespiegelt.
Die Locations werden mit dem im System festgelegten Editor geöffnet.
Wenn gewünscht, können diese Locations in die .dat eingesetzt werden, die zum einlesen benutzt wurde.
- **Transform** → Transformieren der eingelesenen Locations
Die Locations der eingelesenen Datei werden mit der anderen Location multipliziert. (Transformation)
Wie mit dem Doppelpunkt-Operator im KUKA.
Es öffnet sich ein neues Fenster in der eine andere Location in den Formaten KUKA, ABB oder Transformation Matrix eingegeben werden können. Mit dem „Daten einlesen“-Button werden die Parameter übernommen.
Die Locations werden mit dem im System festgelegten Editor geöffnet.
Wenn gewünscht, können diese Locations in die .dat eingesetzt werden, die zum einlesen benutzt wurde.

- **.dat reinigen** → Bereinigen der Deklarationen direkt in der .dat
 Es wird eine Meldung Angezeigt, in der Bestätigt werden soll, dass die Datei bereinigt werden soll.
 Bei der Bereinigung wird die zugehörige .src mit der der .dat verglichen und alle nicht genutzten Deklarationen In der Datei werden entfernt.
 Globale Deklarationen bleiben vorhanden.
 Vor dem Bereinigen wird im Ursprungsverzeichnis Einen Gleichnamige .bak erstellt, die als Backup der nicht Bereinigten Datei dienen soll.

- **.dat sortieren** → Sortieren der Deklarationen direkt in der .dat
 Es wird eine Meldung angezeigt, in der Bestätigt werden soll, dass die Datei sortiert werden soll.
 Bei der Sortierung werden die in der gemachten Deklarationen Nach bekannten Typen sortiert.
 E6POS, PDAT, FDAT usw.
 Unbekannte Typen werden unter Misc gespeichert.
 Die sortierten Deklarationen werden mit den vorherigen unsortierten Deklarationen verglichen und nur wenn diese Identisch sind, wird die .dat geschrieben.
 Es wird keine Backup erstellt.

3. ABB Dateieingaben

Über Drag'n Drop oder manuelle Eingabe muss eine .dat Datei in das Textfeld beschrieben werden.

- **.dat einlesen** → die im Textfeld angegebenen .dat wird eingelesen
- **Euler Quaternien** → Von der eingelesenen Datei werden die Locations abwechselnd von Euler-Winkel auf Quaternien umgeschaltet
- **ausgeben** → Von der eingelesenen Datei werden die Location mit dem im System festgelegten Editor geöffnet.
- **RelBase XYZ** → Verschiebung relativ zur Base.
Die Locations der eingelesenen Datei werden relativ zur zur benutzten Base um die Parameter X/Y/Z verschoben.
Es öffnet sich ein neues Fenster in dem die Parameter X/Y/Z eingegeben werden müssen. Es müssen alle Parameter benannt werden. Mit dem OK-Button werden die Parameter übernommen.
Die Locations werden mit dem im System festgelegten Editor geöffnet.
Wenn gewünscht, können diese Locations in die .dat eingesetzt werden, die zum einlesen benutzt wurde.
- **RelTool XYZ** → Verschiebung relativ zum Tool
Die Locations der eingelesenen Datei werden relativ zum benutzten Tool um die Parameter X/Y/Z verschoben.
Es öffnet sich ein neues Fenster in dem die Parameter X/Y/Z eingegeben werden müssen. Es müssen alle Parameter benannt werden. Mit dem OK-Button werden die Parameter übernommen.
Die Locations werden mit dem im System festgelegten Editor geöffnet.
Wenn gewünscht, können diese Locations in die .dat eingesetzt werden, die zum einlesen benutzt wurde.
- **Mirror X/Z** → Spiegeln über die X/Z Achsen
Die Locations der eingelesenen Datei werden über die Achsen X und Z gespiegelt.
Die Locations werden mit dem im System festgelegten Editor geöffnet.
Wenn gewünscht, können diese Locations in die .dat eingesetzt werden, die zum einlesen benutzt wurde.

- **Transform** → Transformieren der eingelesenen Locations
 Die Locations der eingelesenen Datei werden mit der anderen Location multipliziert. (Transformation)
 Wie mit dem Doppelpunkt-Operator im KUKA.
 Es öffnet sich ein neues Fenster in der eine andere Location in den Formaten KUKA, ABB oder Transformation Matrix eingegeben werden können. Mit dem „Daten einlesen“-Button werden die Parameter übernommen.
 Die Locations werden mit dem im System festgelegten Editor geöffnet.
 Wenn gewünscht, können diese Locations in die .dat eingesetzt werden, die zum einlesen benutzt wurde.

- **.dat reinigen**→ Bereinigen der Deklarationen direkt in der .mod
 Es wird eine Meldung angezeigt, in der Bestätigt werden soll, dass die Datei bereinigt werden soll.
 Bei der Bereinigung werden die zugehörigen Locations mit der der Deklarationen verglichen und alle nicht genutzten Deklarationen In der Datei werden entfernt.
 Globale Deklarationen bleiben vorhanden.
 Vor dem Bereinigen wird im Ursprungsverzeichnis Einen Gleichnamige .bak erstellt, die als Backup der nicht Bereinigten Datei dienen soll.