

2 System konfigurieren Experte

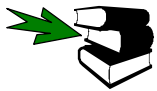
2.1 Benutzergruppe

Zur Erhöhung der Systemsicherheit können Funktionen der Robotersteuerung bzw. deren Programmierung für bestimmte Benutzergruppen gesperrt werden.

Zu diesem Zweck kann der Zugang zu Funktionen derart eingeschränkt werden, daß er nur innerhalb bestimmter "Benutzerebenen" möglich ist, deren Zugang durch ein Paßwort geschützt wird.

In der Grundeinstellung sind in der Steuerung zwei Benutzergruppen, "Experte" und "Anwender" eingerichtet. Nach dem Systemstart befinden Sie sich normalerweise in der Benutzergruppe "Anwender". Höhere "Benutzerebenen" sind nur nach Eingabe eines ebenenabhängigen Paßwortes zu erreichen.

2.1.1 Benutzergruppe wechseln



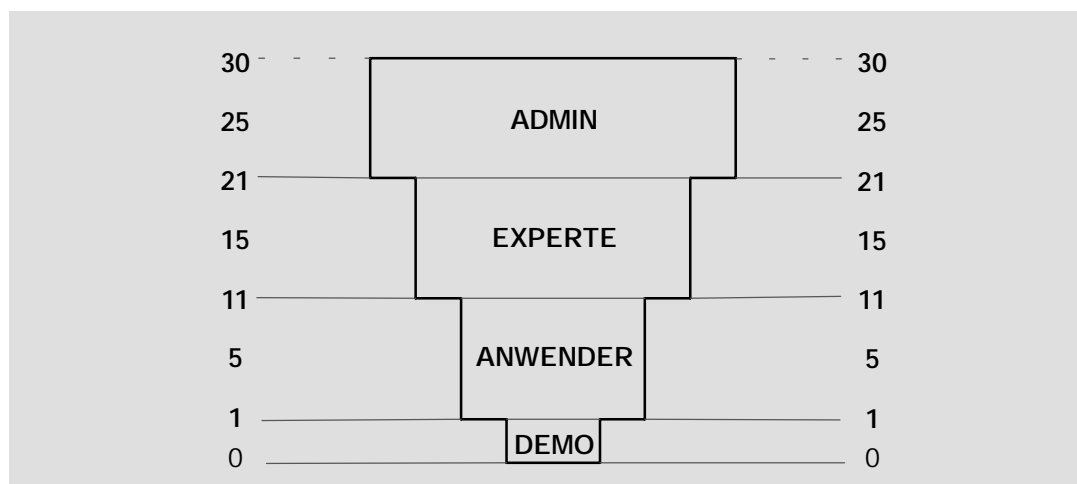
Informationen zum Wechseln der Benutzergruppe finden Sie im Kapitel **[System konfigurieren]**, Abschnitt **[Benutzergruppe]**.

2.1.2 Benutzergruppen einrichten / löschen

Alle verfügbaren Benutzergruppen sind in der Datei `SOFTKEYKUKA.INI` eingetragen. Sie finden diese Datei im Verzeichnis `C:\Programme\KRC\LIB\` (Siehe hierzu auch Abschnitt 2.1.4).

2.1.3 Funktionen sperren

Alle Funktionen können unterhalb einer bestimmten "Benutzerebene" gesperrt werden. Dazu wird die Menüoption bzw. der Softkey, mit dem die Funktion aufgerufen wird, ausgeschaltet. Dies geschieht automatisch in Abhängigkeit mit der Kennzahl der aktuellen Benutzergruppe.



Die Betätigung bzw. Auswahl eines Softkeys oder einer Menüoption ist nur möglich, wenn die Kennzahl der Benutzergruppe größer oder gleich dem Wert ist, der dem Softkey bzw. der Menüoption in der Datei `softkeyKUKA.ini` als siebtes Feldelement angehängt wurde.

**Beispiel:**

TOUCHUP_INSERT = TPBaseTouch, 7014, EDITOR, , , 10

Dieser Softkey ist ab der Benutzergruppe mit der Kennzahl 10 (Anwender) verfügbar.



Achten Sie darauf, daß zum Betrieb notwendige Funktionen in jeder Ebene verfügbar bleiben.

**Beispiel**

Sperren der Funktion "Benutzergruppe wechseln" in der Benutzergruppe "Anwender":

Nach einem Systemstart wird automatisch in die Benutzergruppe "Anwender" geschaltet (Grundeinstellung). Ist die Funktion "Benutzergruppe wechseln" nicht verfügbar, so kann die Benutzergruppe nicht mehr verlassen werden.

2.1.4 Konfigurationsbeispiel



Es soll eine neue Benutzergruppe eingerichtet werden, die alle Funktionen der Gruppe "Anwender" außer der Funktion "Touch Up" beinhaltet. Die programmierte Bahn des Roboters kann dann in dieser Benutzergruppe nicht mehr verändert werden. Dementsprechend soll die neue Ebene "Bestücker" genannt werden.

1.Schritt

Laden Sie die Datei "... \programme\krc\lib\softkeyKUKA.ini" in einen Texteditor und suchen Sie den Abschnitt:

```
;***** User Mode - O C X *****
CLOSE_UMODE    = Close, 0, USERMODE
OK_UMODE       = Ok, 1, USERMODE
USER_UMODE     = UserModeUser, 20, USERMODE, 10
EXPERT_UMODE   = UserModeExpert, 20, USERMODE, 20
```

Fügen Sie diesem Abschnitt die Zeile

```
BESTUECKER_UMODE = Bestuecker, 20, USERMODE, 9
```

hinzu.



Die fettgedruckten Passagen sind frei wählbar, alles andere muß unverändert eingegeben werden.
Ändern Sie auf keinen Fall die Einträge "USER_UMODE" und "EXPERT_UMODE" !

2.Schritt

Suchen Sie den Abschnitt

```
[USERMODE]
```

und fügen die nachfolgend fettgedruckt dargestellte Passage ein.

```
UserGroup = USER_UMODE , EXPERT_UMODE , BESTUECKER_UMODE , ,
, , CLOSE_UMODE
PassWord  = , , , , , OK_UMODE, CLOSE_UMODE
```



Die Anzahl der Feldelemente (7) ist wichtig. Achten Sie sorgfältig darauf, daß Sie kein Komma löschen oder überschreiben !

3.Schritt

Suchen Sie den Eintrag

`TOUCHUP_INSERT` = TPBaseTouch, 7014, EDITOR, , , ,

und hängen die nachfolgend fettgedruckt dargestellte Passage an.

`TOUCHUP_INSERT` = TPBaseTouch, 7014, EDITOR, , , , **10**



Die Anzahl der Feldelemente ist wichtig. Achten Sie sorgfältig darauf, daß die Kennzahl der Benutzerebene für den Softkey als siebtes Feldelement in der Aufzählung steht !

Speichern Sie die geänderte Datei. Nun müssen Sie noch

6 ein Paßwort für die neue Benutzergruppe vergeben, sowie

6 die Benutzergruppe festlegen, die nach einem Systemstart aktiv sein soll.

Diese Themen werden aus Gründen der Systemsicherheit in der Dokumentation [Reference book - Sicherheit und Erstinbetriebnahme von KUKA Industrierobotern] beschrieben.

2.2 Konfigurationsdateien

2.2.1 Projektierbare Anzeige - Datei <ConfigMon. INI>

Diese Datei liegt standardmäßig im Verzeichnis "C:\Programme\KRC\Init". Jede Gruppe anzuzeigender Variablen wird in einer eigenen Sektion zusammengefaßt, die mit [Groupn] beginnt und erst durch den Beginn einer darauf folgenden Sektion oder dem Dateiende (EOF) abgeschlossen wird.



Beispiel

| Projektierbare Anzeige | |
|-------------------------------------|---------------|
| Gruppe 4 : ManuellerSicherheitstest | |
| SystemVariable | aktuellerWert |
| \$STOPMESS | FALSE |
| \$PERI_RDY | TRUE |
| \$USER_SAF | TRUE |
| \$ALARM_STOP | TRUE |
| \$PRO_STATE1 | #P_FREE |
| \$T1 | TRUE |
| \$T2 | FALSE |
| \$AUT | FALSE |
| \$EXT | FALSE |
| \$EXTSTARTTYP | FALSE |
| Refresh | |

Die Nebenstehende Tabelle benötigt die folgenden Einträge in der Datei "ConfigMon.ini":

```
[Version]
Version=1.0

.
.
.

[Group4]
GroupTitle=ManuellerSicherheitstest
NbDataCols=4
User=10
Substitute==;==;==;==;
MaxColWidths=1800;1500;
ColTitles=SystemVar.;aktuellerWert;
Item_0=TMP;-T;$STOPMESS;$STOPMESS;
Item_1=TMP;-T;$PERI_RDY;$PERI_RDY;
Item_2=TMP;-T;$USER_SAF;$USER_SAF;
Item_3=TMP;-T;$ALARM_STOP;$ALARM_STOP;
Item_4=TMP;-T;$PRO_STATE1;$PRO_STATE1;
Item_5=TMP;-T;$T1;$T1;
Item_6=TMP;-T;$T2;$T2;
Item_7=TMP;-T;$AUT;$AUT;
Item_8=TMP;-T;$EXT;$EXT;
Item_9=TMP;-T;$EXTSTARTTYP;$EXTSTARTTYP;
```

Die einzelnen Zeilen haben folgende Bedeutung:

[Version]

Version=1.0

Versionsnummer für Upgradezwecke

[Group4]

Nummer der jeweiligen Anzeigegruppe von [Group1]...[Group10]

GroupTitle=Manueller Sicherheitstest

Überschrift im Zustandsfenster

NmbDataCols= 4

Anzahl der anzuzeigenden Spalten plus 2, wobei mindestens zwei Spalten erforderlich sind

User=10

Hier wird festgelegt, ab welcher Benutzergruppe Variablenwerte gesetzt werden dürfen. Werte zwischen "User=0"... "User=10" erlauben allen Benutzergruppen (standardmäßig Anwender und Experte) Variablenwerte zu setzen. Wird "User=20" gesetzt, dürfen nur Experten Änderungen vornehmen, während "User=30" keinerlei Modifikationen gestattet. Ist kein Wert angegeben, wird automatisch "User=20" angenommen.

Substitute=JA=TRUE; NEIN=FALSE; =; =; =;

So kann beispielsweise der logische Zustand "True" durch den Begriff "Ja" oder der Zustand "False" durch das Wort "Nein" ersetzt werden.

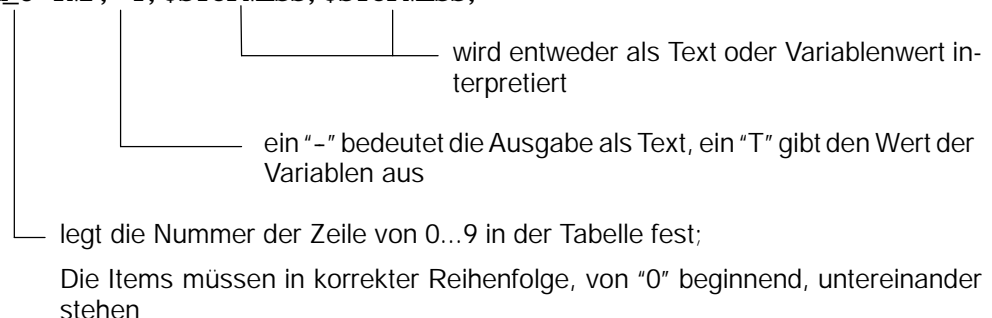
MaxCol Widths=1800; 1500;

Breite der Spalten in Twips, einer unter Windows gebräuchlichen Maßeinheit. Bei einer Auflösung von 800x600 Bildpunkten entsprechen 15 Twips einem Pixel auf dem Bildschirm.

ColTitles=Systemvariablen; aktueller Wert;

Spaltenüberschriften im Zustandsfenster

Item 0=TMP; - T; \$STOPMESS; \$STOPMESS;



2.2.2 Templates - Datei <Template.ini>

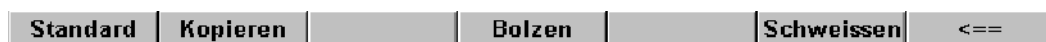
Neben der Möglichkeit ein neues Programm zu erstellen oder ein vorhandenes zu kopieren können Programme auch nach speziellen Vorgaben, sogenannten Templates, erstellt werden. Der Anwender bestimmt hierbei, welches Programm als Vorgabe in der Softkeyleiste angeboten wird. Dafür stehen dem Bediener maximal 4 Templates auf den Softkeys 3..6 zur Verfügung.



Diese 4 Softkeys können vom Anwender belegt werden



Beispiel



Anwenderspezifische Musterprogramme

Folgende Einträge sind für obiges Beispiel in der Datei "C:\Programme\KRC\Init\Template.ini" notwendig:

```

.
.
.
[Template1]
[Template2]
Modul=BolzenUP1
Button=Bolzen
Default=Bolzen01
[Template3]
[Template4]
Modul=SchweißenUP1
Button=Schweißen
Default=Schw01

```

2. Template entspricht dem Softkey Nr.4

Modulname eines im Verzeichnis "C:\Programme\KRC\Template\" vorhandenen Musterprogramms. Eine einzelne "SRC"-Datei als Template ist zulässig.

Beschriftung des Softkeys, bestehend aus maximal 10 Zeichen.

Maximal 8 Zeichen langer Programmname der im Namensfeld des InLine-Formulars vorgeschlagen wird. Diese Angabe ist optional.



Änderungen in der Datei "Template.ini" sind online möglich. Nachdem die Datei bearbeitet und abgespeichert worden ist, braucht die Softkeyleiste nur erneut aufgerufen zu werden.

2.2.3 Zentralarchiv - Datei <Archive.ini>

In der Datei "C:\Programme\KRC\Init\Archive.ini" können die Pfade zum Sichern und Laden über das Zentralarchiv festgelegt werden. In dieser Konfigurationsdatei können die Pfade für Verzeichnisse und Teilbäume der Registry eingetragen werden. Außerdem ist es möglich, die Sprach- und Langtextdatenbank neu einzulesen.

Verzeichnispfade

Insgesamt können bis zu 20 Verzeichnispfade bzw einzelne Dateien gesichert und später wieder zurückgeschrieben werden.

Die Pfade der zu sichernden Verzeichnisse müssen in der Datei "Archive.ini" unter dem Abschnitt "[Save]" angegeben werden. Die Bezeichnung "\" steht hier für alle Dateien, die sich im genannten Pfad befinden.



```

.
.
.
; Specify all file items to be saved in archive directories
[Save]
Source1 = %InstallationDir%\Mada\*
Target1 = d: \Zentral archiv\Mada\*
Source2 = %InstallationDir%\Mada\Steu\*
Target2 = d: \Zentral archiv\Mada\Steu\*
Source3 = %InstallationDir%\Mada\R1\*
Target3 = d: \Zentral archiv\Mada\R1\*
Source4 = %InstallationDir%\PowerOn\*
Target4 = d: \Zentral archiv\PowerOn\*
Source5 = %InstallationDir%\PowerOn\Steu\*
Target5 = d: \Zentral archiv\PowerOn\Steu\*
Source6 = %InstallationDir%\PowerOn\R1\*
Target6 = d: \Zentral archiv\PowerOn\R1\*
Source7 = %InstallationDir%\Init\*
Target7 = d: \Zentral archiv\Init\*
Source8 = %InstallationDir%\ir_spec\*
Target8 = d: \Zentral archiv\ir_spec\*
Source9 = %InstallationDir%\template\*
Target9 = d: \Zentral archiv\template\*
Source10 = %InstallationDir%\lib\menuKuka. ini
Target10 = d: \Zentral archiv\menuKuka. ini
Source11 = %InstallationDir%\lib\softkeyKuka. ini
Target11 = d: \Zentral archiv\softKeyKuka. ini
Source12 = %InstallationDir%\data\Kuka_con. mdb
Target12 = d: \Zentral archiv\data\Kuka_con. mdb
Source13 = %InstallationDir%\data\Kuka_tab. mdb
Target13 = d: \Zentral archiv\data\Kuka_tab. mdb
; Source14 =
.
.
.
; Target20 =

```

Hinter dem Eintrag "Source" wird das jeweilige Quellverzeichnis angegeben, welches gesichert werden soll.

"Target" gibt das eigentliche Zielverzeichnis bzw. die Zieldatei des Zentralarchivs an, auf welches die Daten gesichert werden.

Die Pfade für das Laden vom Zentralarchiv müssen im Abschnitt "[Load]" angegeben werden. Die Bezeichnung "*" steht hier ebenfalls für alle Dateien, die sich im genannten Pfad befinden.



```

.
.
.
; Specify all file items to be loaded from archive directories
[Load]
Source1 = d:\Zentral archiv\Mada\*
Target1 = %InstallationDir%\Mada\*
Source2 = d:\Zentral archiv\Mada\Steu\*
Target2 = %InstallationDir%\Mada\Steu\*
Source3 = d:\Zentral archiv\Mada\R1\*
Target3 = %InstallationDir%\Mada\R1\*
Source4 = d:\Zentral archiv\PowerOn\*
Target4 = %InstallationDir%\PowerOn\*
Source5 = d:\Zentral archiv\PowerOn\Steu\*
Target5 = %InstallationDir%\PowerOn\Steu\*
Source6 = d:\Zentral archiv\PowerOn\R1\*
Target6 = %InstallationDir%\PowerOn\R1\*
Source7 = d:\Zentral archiv\Init\*
Target7 = %InstallationDir%\Init\*
Source8 = d:\Zentral archiv\ir_spec\*
Target8 = %InstallationDir%\ir_spec\*
Source9 = d:\Zentral archiv\template\*
Target9 = %InstallationDir%\template\*
Source10 = d:\Zentral archiv\menuKuka.ini
Target10 = %InstallationDir%\lib\*
Source11 = d:\Zentral archiv\softKeyKuka.ini
Target11 = %InstallationDir%\lib\*
Source12 = %InstallationDir%\data\Kuka_con.mdb
Target12 = d:\Zentral archiv\data\Kuka_con.mdb
Source13 = %InstallationDir%\data\Kuka_tab.mdb
Target13 = d:\Zentral archiv\data\Kuka_tab.mdb
; Source14 =
.
.
.
; Target20 =

```

Hinter dem Eintrag "Source" wird der Pfad der Verzeichnisse bzw. Dateien auf dem Zentralarchiv angegeben,.

"Target" gibt das eigentliche Zielverzeichnis an, wohin die Dateien vom Zentralarchiv zurückgeschrieben werden sollen.

Registry-Pfade

Die Pfade der zu sichernden Verzeichnisse müssen in der Datei "Archive.ini" unter dem Abschnitt "[RegSave]" angegeben werden. Die Bezeichnung "*" steht hier für alle Dateien, die sich im genannten Pfad befinden.



```

.
.
.
; Specify all registry items to be saved in archive files
[RegSave]
Source1 = \CurrentUser\Software\Kuka Roboter GmbH\*
Target1 = d:\Zentral archiv\cu_kuka.txt
Source2 = \CurrentUser\Software\VB and VBA Program Settings\*
Target2 = d:\Zentral archiv\cu_vb.txt
Source3 = \Local Machine\Software\Kuka Roboter GmbH\*
Target3 = d:\Zentral archiv\lm_kuka.txt
; Source4 =
.
.
.
; Target20 =

```

Die Pfade für das Laden vom Zentralarchiv müssen im Abschnitt "[RegLoad]" angegeben werden. Die Bezeichnung "*" steht hier ebenfalls für alle Dateien, die sich im genannten Pfad befinden.



```

.
.
.
; Specify all files to load registry items
[RegLoad]
Source1 = d:\Zentral archiv\cu_kuka.txt
Source2 = d:\Zentral archiv\cu_vb.txt
Source3 = d:\Zentral archiv\lm_kuka.txt
; Source4 =
.
.
.
; Source20 =

```

Sprach- und Langtextdatenbank

Diese Option gibt an, ob die Sprach- bzw. Langtextdatenbank neu eingelesen werden soll (TRUE) oder nicht (FALSE). In der Datei "Archiv.ini" befinden sich die Einträge unter dem Abschnitt "[DBLoad]".



Nur die Registry-Schlüssel "\LocalMachine\Software\" (entspricht "HKEY_LOCAL_MACHINE") und "\CurrentUser\Software\" (entspricht "HKEY_CURRENT_USER") sind zugelassen.



```

.
.
.
; Specify whether databases should be refreshed
[DBLoad]
Config = FALSE
Language = FALSE

```



Netzwerkverzeichnisse können sowohl als sogenanntes "gemapptes" Laufwerk wie z.B. "H:\...", als auch in der URL-Notation wie z.B. "\\Server25\Zentralarchiv\..." angegeben werden.

Pfade von Verzeichnissen bzw. Dateien die innerhalb des KRC-Verzeichnisses liegen, müssen mit "%InstallationDir%" beginnen. Dies entspricht standardmäßig dem Pfad "C:\Programme\KRC\".

Kommentare beginnen grundsätzlich mit ";".

2.3 5 Home-Positionen

Ab der Softwareversion 2.3 kann der Anwender zusätzlich zur Home-Position weitere 5 Home-Positionen definieren. Insgesamt stehen dann 6 verschiedene Positionen zur Auswahl.

Analog zur bisherigen Home-Position "\$H_POS", bei deren Erreichen die Variable "\$IN_HOME" auf "TRUE" gesetzt wird, werden für die Positionen "\$AXIS_HOME[1]" ... "\$AXIS_HOME[5]" jeweils die Variablen "\$IN_HOME1" ... "\$IN_HOME5" auf "TRUE" gesetzt.

Wie bei "\$H_POS" gilt auch bei den 5 zusätzlichen Home-Positionen das durch die Variable "\$H_POS_TOL" definierte Toleranzband. Befinden sich alle Achsen innerhalb dieses Toleranzfensters, wird der zugehörige Ausgang gesetzt.



Die entsprechenden Einstellungen bzw. Änderungen können nur im Expertenmodus vorgenommen werden.

2.3.1 Datei "\R1\\$_machine.dat"

Hinter den hervorgehobenen Bereichen "\$AXIS_HOME[x]" werden die Koordinaten der Achsen 1...6 bzw. Zusatzachsen E1...E6 angegeben.

```
&PARAM VERSION=13. 0. 0
DEFDAT  $MACHINE PUBLIC
.
.
.
E6AXIS  $H_POS={A1  0. 0, A2  -90. 0, A3  90. 0, A4  0. 0, A5  0. 0, A6  0. 0, E1
0. 0, E2  0. 0, E3  0. 0, E4  0. 0, E5  0. 0, E6  0. 0}
E6AXIS  $H_POS_TOL={A1  2. 0, A2  2. 0, A3  2. 0, A4  2. 0, A5  2. 0, A6  2. 0, E1
2. 0, E2  2. 0, E3  2. 0, E4  2. 0, E5  2. 0, E6  2. 0}
E6AXIS  $AXIS_HOME[ 5]
$AXIS_HOME[ 1]={A1  0. 0, A2  -90. 0, A3  90. 0, A4  0. 0, A5  0. 0, A6  0. 0, E1
0. 0, E2  0. 0, E3  0. 0, E4  0. 0, E5  0. 0, E6  0. 0}
$AXIS_HOME[ 2]={A1  0. 0, A2  -90. 0, A3  90. 0, A4  0. 0, A5  0. 0, A6  0. 0, E1
0. 0, E2  0. 0, E3  0. 0, E4  0. 0, E5  0. 0, E6  0. 0}
$AXIS_HOME[ 3]={A1  0. 0, A2  -90. 0, A3  90. 0, A4  0. 0, A5  0. 0, A6  0. 0, E1
0. 0, E2  0. 0, E3  0. 0, E4  0. 0, E5  0. 0, E6  0. 0}
$AXIS_HOME[ 4]={A1  0. 0, A2  -90. 0, A3  90. 0, A4  0. 0, A5  0. 0, A6  0. 0, E1
0. 0, E2  0. 0, E3  0. 0, E4  0. 0, E5  0. 0, E6  0. 0}
$AXIS_HOME[ 5]={A1  0. 0, A2  -90. 0, A3  90. 0, A4  0. 0, A5  0. 0, A6  0. 0, E1
0. 0, E2  0. 0, E3  0. 0, E4  0. 0, E5  0. 0, E6  0. 0}
.
.
.
```



Diese Datei befindet sich standardmäßig im Verzeichnis "C:\Programme\KRC\MaDa\R1\".

2.3.2 Datei “\Steu\\$machine.dat”

In der Datei “\$machine.dat” wird jeder Variablen “\$IN_HOME1” ... “\$IN_HOME5” ein bestimmter Ausgang zugewiesen.

```
&PARAM VERSION=13. 0. 0
DEFDAT  $MACHINE PUBLIC
.
.
.
SIGNAL $IN_HOME $OUT[ 1000] ; ROB IN HOMEPOSITION
.
.
.
SIGNAL $IN_HOME1 $OUT[ 977]
SIGNAL $IN_HOME2 $OUT[ 978]
SIGNAL $IN_HOME3 $OUT[ 979]
SIGNAL $IN_HOME4 $OUT[ 980]
SIGNAL $IN_HOME5 $OUT[ 981]
.
.
.
ENDDAT
```



Diese Datei befindet sich standardmäßig im Verzeichnis “C:\Programme\KRC\MaDa\Steu\”.

2.4 Arbeitsraumüberwachung

2.4.1 Definition

Es können bis zu acht kubische Arbeitsräume überwacht werden, die sich auch überlappen dürfen, um so z.B. komplexe Formen zu bilden. Tritt das Werkzeug oder Werkstück mit seinem Bezugspunkt in einen dieser Arbeitsräume ein, bzw. aus einem Arbeitsraum heraus, wird ein zuvor zugeordneter Ausgang der Steuerung gesetzt. Der Roboter kann dann auch angehalten und eine Fehlermeldung ausgegeben werden.



Je nach Geschwindigkeit des Roboters kann dabei sein Bremsweg variieren und das Werkzeug / Werkstück kann so in den Arbeitsraum gelangen !

Das bereitgestellte Ausgangssignal kann dann vom KRL-Programm oder von einem externen Leitrechner weiterverarbeitet werden.

Die Definition eines Arbeitsraums erfolgt durch einen Eintrag in der Datei:

```
... \KRC\MADA\STEU\SCUSTOM.DAT :
```

```
DEFDAT SCUSTOM PUBLIC
```

```
...
```

```
WORKSPACE[1]={X 0.0, Y 0.0, Z 0.0, A 0.0, B 0.0, C 0.0, X1 0.0, Y1 0.0, Z1 0.0, X2 0.0, Y2 0.0, Z2 0.0, MODE #OFF}
```

```
WORKSPACE[2]={X 0.0, Y 0.0, Z 0.0, A 0.0, B 0.0, C 0.0, X1 0.0, Y1 0.0, Z1 0.0, X2 0.0, Y2 0.0, Z2 0.0, MODE #OFF}
```

```
WORKSPACE[3]={X 0.0, Y 0.0, Z 0.0, A 0.0, B 0.0, C 0.0, X1 0.0, Y1 0.0, Z1 0.0, X2 0.0, Y2 0.0, Z2 0.0, MODE #OFF}
```

```
WORKSPACE[4]={X 0.0, Y 0.0, Z 0.0, A 0.0, B 0.0, C 0.0, X1 0.0, Y1 0.0, Z1 0.0, X2 0.0, Y2 0.0, Z2 0.0, MODE #OFF}
```

```
WORKSPACE[5]={X 0.0, Y 0.0, Z 0.0, A 0.0, B 0.0, C 0.0, X1 0.0, Y1 0.0, Z1 0.0, X2 0.0, Y2 0.0, Z2 0.0, MODE #OFF}
```

```
WORKSPACE[6]={X 0.0, Y 0.0, Z 0.0, A 0.0, B 0.0, C 0.0, X1 0.0, Y1 0.0, Z1 0.0, X2 0.0, Y2 0.0, Z2 0.0, MODE #OFF}
```

```
WORKSPACE[7]={X 0.0, Y 0.0, Z 0.0, A 0.0, B 0.0, C 0.0, X1 0.0, Y1 0.0, Z1 0.0, X2 0.0, Y2 0.0, Z2 0.0, MODE #OFF}
```

```
WORKSPACE[8]={X 0.0, Y 0.0, Z 0.0, A 0.0, B 0.0, C 0.0, X1 0.0, Y1 0.0, Z1 0.0, X2 0.0, Y2 0.0, Z2 0.0, MODE #OFF}
```

```
...
```

```
ENDDAT
```



Arbeitsräume können auch in *.SRC-Dateien definiert, bzw. ein- und ausgeschaltet werden. Die dort angegebenen Werte werden automatisch in die Datei "SCUSTOM.DAT" eingetragen und stehen beim nächsten Hochlauf der Steuerung wieder zur Verfügung.

Auch über die Korrektur von Variablen können die Einstellungen von Arbeitsräumen verändert werden.



Nähere Informationen darüber, wie der Wert von Systemvariablen während des Betriebs der Steuerung geändert werden kann, finden Sie im Kapitel **[Anzeigen]**, Abschnitt "Variablen".

Die Bedeutung der Komponenten der Struktur **\$WORKSPACE**:

| | | | |
|----------|-------------------------------|----------|---------------------------------|
| X | Position des Arbeitsraumes in | A | Orientierung des Arbeitsraumes, |
| Y | den Hauptachsen, bezogen auf | B | bezogen auf das Welt- |
| Z | das Welt-Koordinatensystem | C | Koordinatensystem |

| | | | |
|-----------|--|-----------|--|
| X1 | bestimmt Δx_1 , Δy_1 , Δz_1 in Bezug auf den Punkt X, Y, Z, A, | X2 | bestimmt Δx_2 , Δy_2 , Δz_2 in Bezug auf den Punkt X, Y, Z, A, |
| Y1 | | Y2 | |
| Z1 | B, C und spannt Quader auf | Z2 | B, C und erweitert Quader |

| Einstellungsmöglichkeiten für "MODE": | |
|---------------------------------------|--|
| #OFF | Die Überwachung des betreffenden Arbeitsraumes wird abgeschaltet. |
| #INSIDE | Vorgegebener Ausgang wird gesetzt, wenn sich der Bezugspunkt (TCP) des Werkzeugs / Werkstücks <u>innerhalb</u> des Arbeitsraumes befindet. |
| #OUTSIDE | Vorgegebener Ausgang wird gesetzt, wenn sich der Bezugspunkt (TCP) des Werkzeugs / Werkstücks <u>außerhalb</u> des Arbeitsraumes befindet. |
| #INSIDE_STOP | Vorgegebener Ausgang wird gesetzt, wenn sich der Bezugspunkt (TCP) des Werkzeugs / Werkstücks, bzw. der Handwurzepunkt <u>innerhalb</u> des Arbeitsraumes befindet. Zusätzlich wird der Roboter gestoppt und die Fehlermeldung 114 "Arbeitsraum Nr. <i>n</i> verletzt" ausgegeben. |
| #OUTSIDE_STOP | Vorgegebener Ausgang wird gesetzt, wenn sich der Bezugspunkt (TCP) des Werkzeugs / Werkstücks <u>außerhalb</u> des Arbeitsraumes befindet. Zusätzlich wird der Roboter gestoppt und die Fehlermeldung 114 "Arbeitsraum Nr. <i>n</i> verletzt" ausgegeben. |

Die Zuordnung der Signale zu den Ausgängen erfolgt ausschließlich in der Datei

C: \PROGRAMME\KRC\MADA\STEU\SMACHINE.DAT



Ändern Sie die Zuordnungen der Variablen **\$WORKSTATE n** nur in der Datei "**\$MACHINE.DAT**".

Änderungen in anderen Dateien führen zu Fehlfunktionen!

```

DEFDAT  $MACHINE PUBLIC
...
SIGNAL $WORKSTATE1 $OUT[ n ]
SIGNAL $WORKSTATE2 $OUT[ n ]
SIGNAL $WORKSTATE3 $OUT[ n ]
SIGNAL $WORKSTATE4 $OUT[ n ]
SIGNAL $WORKSTATE5 $OUT[ n ]
SIGNAL $WORKSTATE6 $OUT[ n ]

```

```
SIGNAL $WORKSTATE7 $OUT[ n]
SIGNAL $WORKSTATE8 $OUT[ n]
```

```
...
```

```
ENDDAT
```

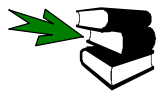
Aus der Komponente \$WORKSPACE[n].STATE kann der Zustand des zugeordneten Ausgangs ausgelesen werden.

Wird in den Modi "INSIDE_STOP" oder "OUTSIDE_STOP" ein Arbeitsraum verletzt, kann der Roboter erst dann wieder verfahren werden, wenn die betreffende Arbeitsraumüberwachung ausgeschaltet wird. Dazu gibt es zwei Möglichkeiten:

Ausschalten der Arbeitsraumüberwachung über Variablenkorrektur bzw. KRL-Programm

Ändern Sie den Wert der Komponente "MODE" des betroffenen Arbeitsraumes z.B.

"\$WORKSPACE[1]. MODE" in "#OFF".



Nähere Informationen darüber, wie der Wert von Systemvariablen während des Betriebs der Steuerung geändert werden kann, finden Sie im Kapitel **[Anzeigen]**, Abschnitt **"Variablen"**.

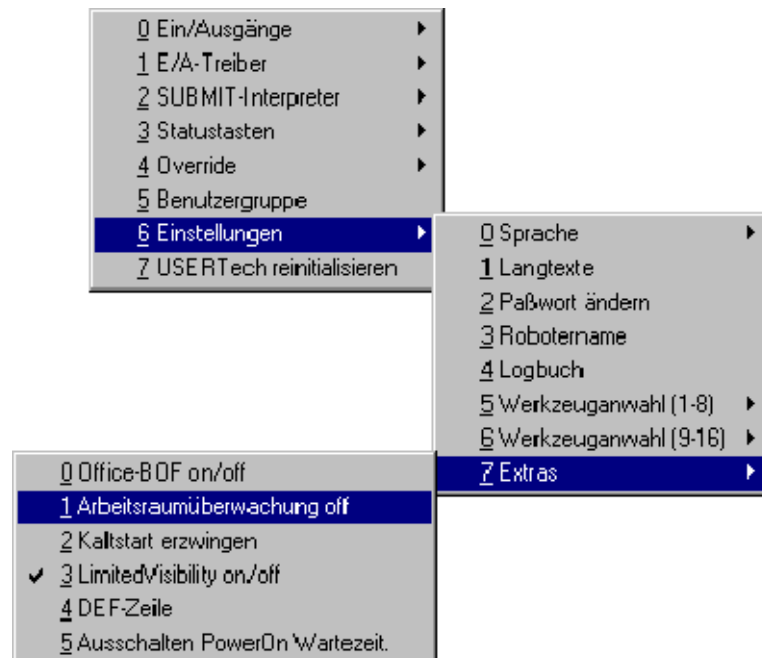


Die Überwachung des betreffenden Arbeitsraumes bleibt solange abgeschaltet, bis die Komponente wieder auf einen Wert ungleich "#OFF" gesetzt wird.

Ausschalten der Arbeitsraumüberwachung über Menü

Konfig.

Das "Überbrücken" der Arbeitsraumüberwachung ist über die Menüfunktion "Konfig." -> "Einstellungen" -> "Extras" -> "Arbeitsraumüberwachung off" möglich.



Diese Funktion ermöglicht es, den Roboter aus dem verletzten Arbeitsraum wieder herauszufahren.



Dies ist nur in der Betriebsart TEST (T1) möglich.

Wurde ein Arbeitsraum verletzt, so erscheint die Fehlermeldung 114:

“Arbeitsraum Nr. n verletzt”

Wird die Arbeitsraumüberwachung dann ausgeschaltet, wird diese Meldung durch die Zustandsmeldung 115:

“Arbeitsraum Nr. n freifahren”

ersetzt. Nach dem Verlassen des verletzten Arbeitsraumes wird diese Meldung gelöscht.

Ist “\$TOOL” ungültig und mindestens ein Arbeitsraum aktiv, so erscheint die Fehlermeldung 112

“\$TOOL ungültig: Keine Arbeitsraumüberwachung möglich” im Meldungsfenster.

Bereits gesetzte Ausgänge werden zurückgesetzt und eventuell anstehende Meldungen gelöscht.

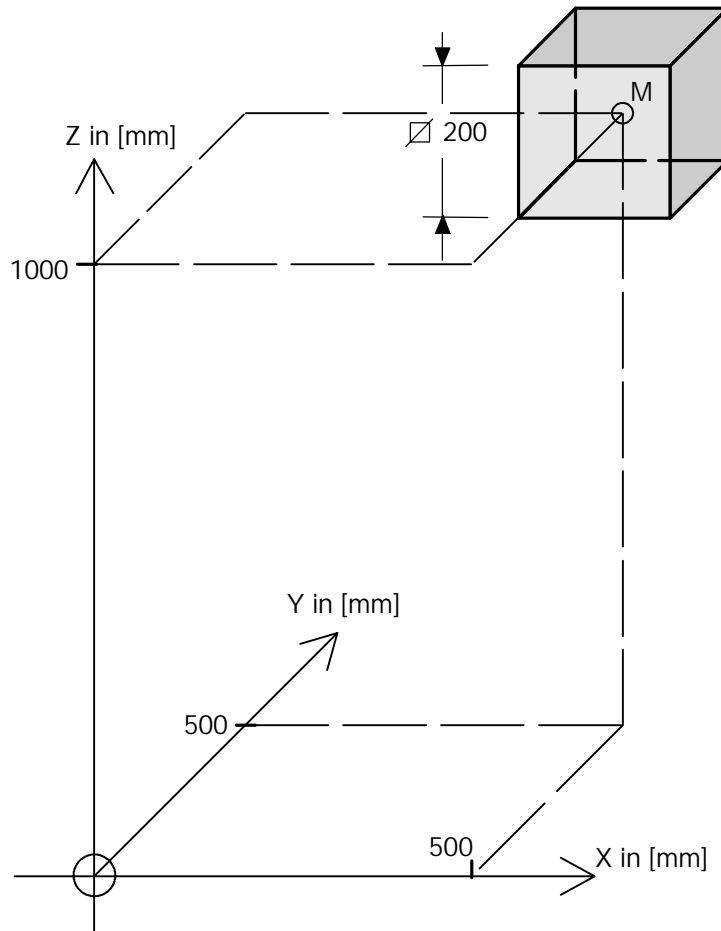


Fehlerhafte “\$TOOL”-Daten können zu unvorhersehbaren Situationen führen!

2.4.2 Beispiele



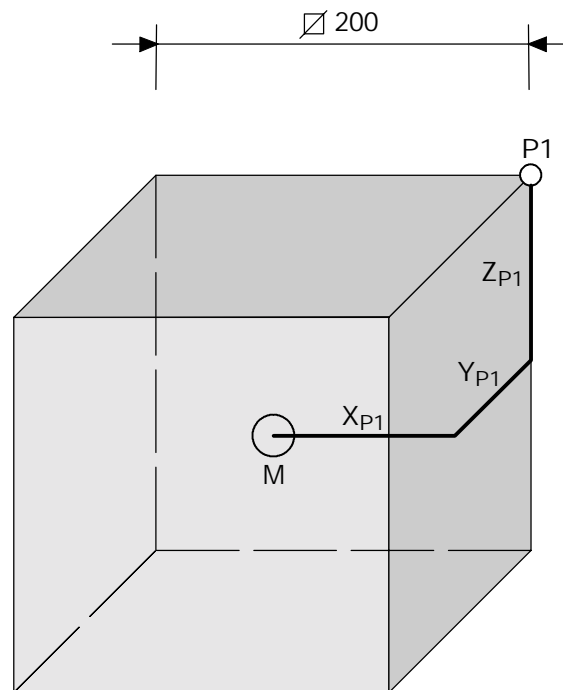
Es soll ein kubischer Arbeitsraum mit 200 mm Kantenlänge überwacht werden. Sein Mittelpunkt soll bei $X=500\text{mm}$, $Y=500\text{mm}$, $Z=1000\text{mm}$ liegen. Die Winkel A, B und C besitzen den Wert "0".



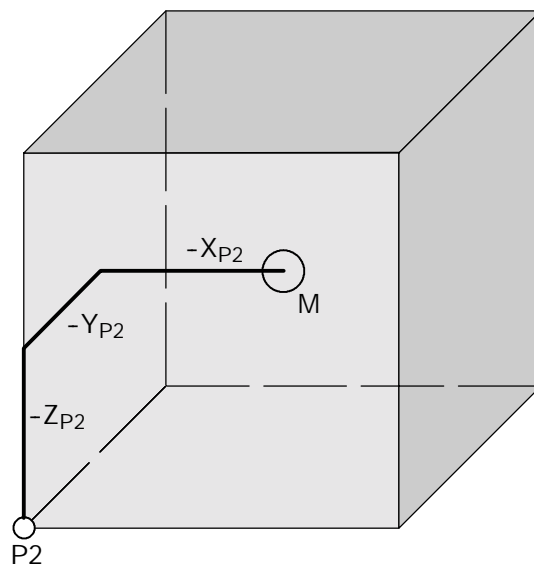
Der Mittelpunkt des Arbeitsraums (Punkt "M") wird im Workspace-Befehl folgendermaßen definiert:

```
WORKSPACE[n]={X 500, Y 500, Z 1000, A 0, B 0, C 0, X1 100, Y1 100, Z1 100, X2 -100, Y2 -100, Z2 -100, MODE #INSIDE}
```

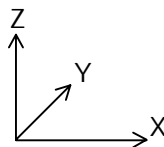
Der Arbeitsraum soll eine Seitenlänge von 200mm besitzen. Vom Mittelpunkt "M" ausgehend wird der Bereich über die Punkte "P1" in positiver und "P2" in negativer Richtung definiert.



$$\begin{aligned} X_{P1} &= 100\text{mm} \\ Y_{P1} &= 100\text{mm} \\ Z_{P1} &= 100\text{mm} \end{aligned}$$



$$\begin{aligned} -X_{P2} &= -100\text{mm} \\ -Y_{P2} &= -100\text{mm} \\ -Z_{P2} &= -100\text{mm} \end{aligned}$$



Die erforderlichen Parameter finden sich in der Parameterzeile an folgender Stelle wieder:

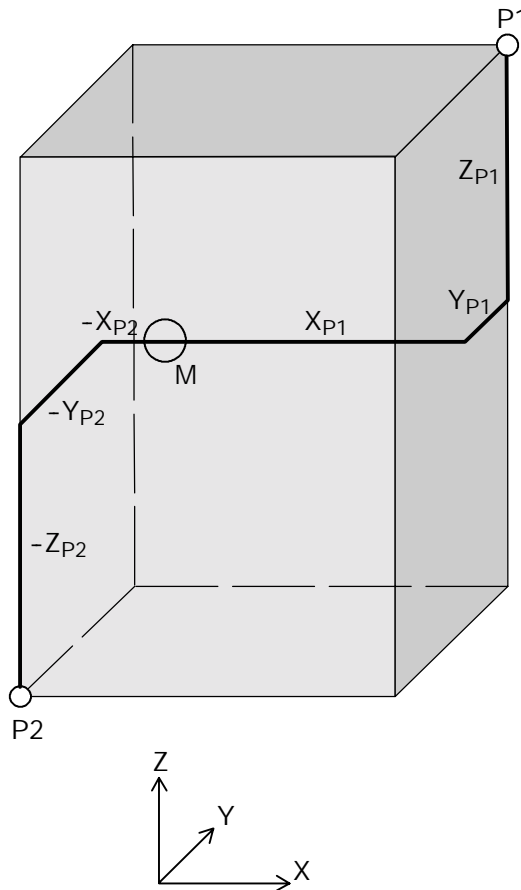
WORKSPACE[n]={X 500, Y 500, Z 1000, A 0, B 0, C 0, X1 100, Y1 100, Z1 100, X2 -100, Y2 -100, Z2 -100, MODE #INSIDE}

Der zugeordnete Ausgang soll gesetzt werden, sobald sich der Bezugspunkt von Werkzeug, bzw. Werkstück innerhalb des Arbeitsraums befindet. Dies stellt den letzten Eintrag im Workspace-Befehl dar:

WORKSPACE[n]={X 500, Y 500, Z 1000, A 0, B 0, C 0, X1 100, Y1 100, Z1 100, X2 -100, Y2 -100, Z2 -100, **MODE #INSIDE}**



In diesem Beispiel besitzt der Arbeitsraum die Abmessungen $x=300\text{mm}$, $y=250\text{mm}$ und $z=450\text{mm}$. Der Punkt "M" liegt in diesem Fall NICHT im Zentrum des Quaders.



$$X_{P1} = 250\text{mm}$$

$$Y_{P1} = 150\text{mm}$$

$$Z_{P1} = 200\text{mm}$$

$$-X_{P2} = -50\text{mm}$$

$$-Y_{P2} = -100\text{mm}$$

$$-Z_{P2} = -250\text{mm}$$

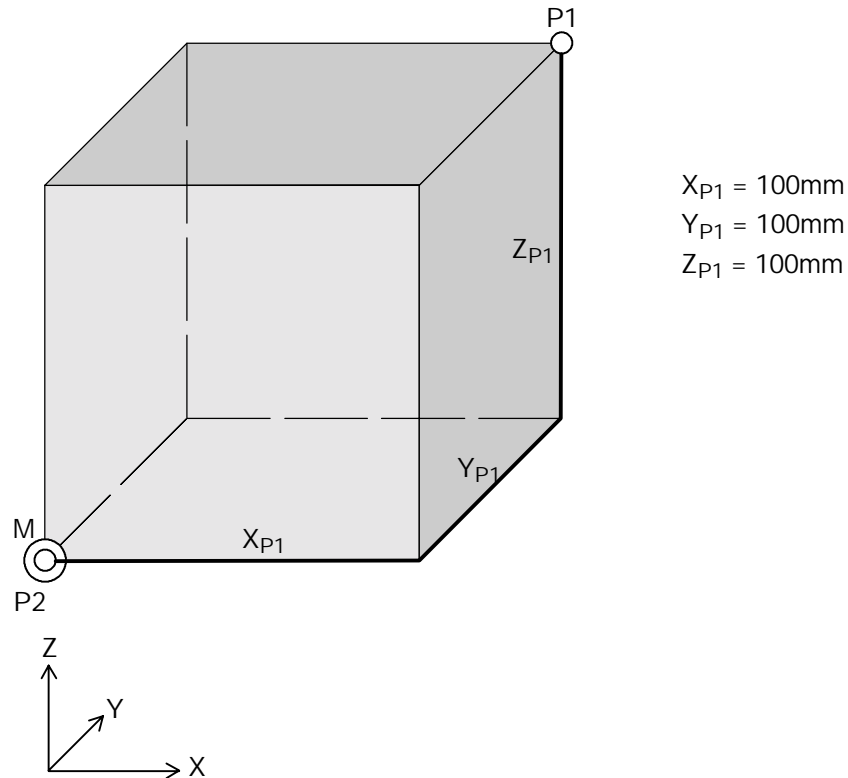
Der zugeordnete Ausgang soll wieder gesetzt werden, wenn sich der Bezugspunkt von Werkzeug, bzw. Werkstück innerhalb des Arbeitsraums befindet. Gleichzeitig soll der Roboter stoppen und eine Fehlermeldung ausgeben.

Die entsprechende Anweisung lautet:

WORKSPACE[n] = {X 500, Y 500, Z 2000, A 0, B 0, C 0, X1 250, Y1 150, Z1 200, X2 -50, Y2 -100, Z2 -250, MODE #INSIDE_STOP}



Wird der Punkt "M" beispielsweise auf den Punkt "P2" gelegt, sind ausschließlich die Koordinaten des Punktes "P1" für den Arbeitsraum ausschlaggebend.



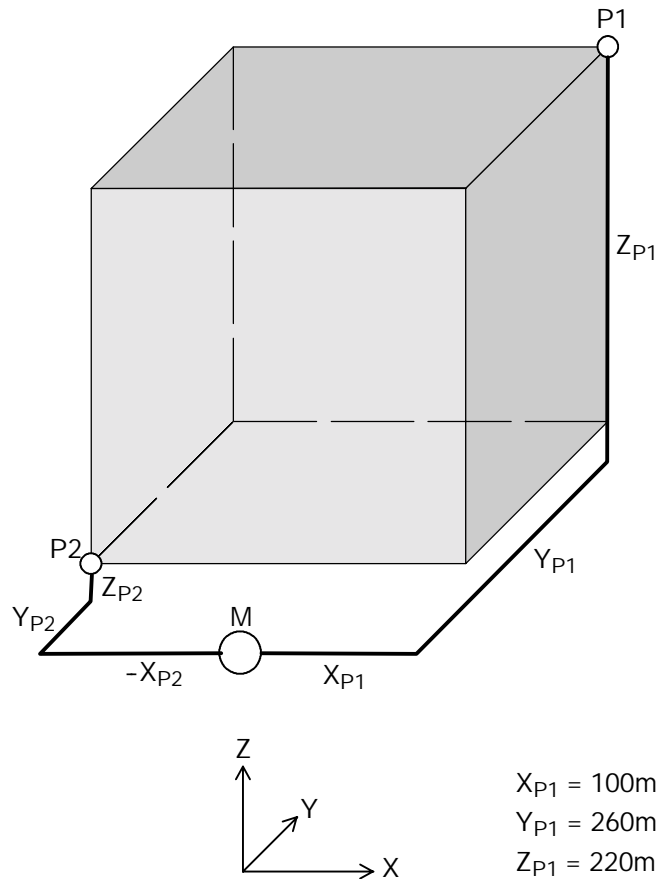
Befindet sich der Bezugspunkt von Werkzeug bzw. Werkstück außerhalb des Arbeitsraums, soll der zugeordnete Ausgang gesetzt werden. Gleichzeitig soll der Roboter stoppen und eine Fehlermeldung ausgeben.

Die entsprechende Anweisung lautet:

WORKSPACE[n]={X 500, Y 500, Z 2000, A 0, B 0, C 0, X1 100, Y1 100, Z1 100, X2 0, Y2 0, Z2 0, MDE #OUTSIDE_STOP}



Hat eines der Punktkoordinatenpaare " X_{P1} " und " X_{P2} ", " Y_{P1} " und " Y_{P2} " oder " Z_{P1} " und " Z_{P2} " gleiche Vorzeichen, liegt der Punkt "M" außerhalb des Arbeitsraums. Die Kantenlänge des Quaders ergibt sich folglich aus der Differenz der X, Y oder Z-Komponenten.



Auch hier soll der zugeordnete Ausgang gesetzt werden, wenn sich der Bezugspunkt von Werkzeug bzw. Werkstück außerhalb des Arbeitsraums befindet.

Die entsprechende Anweisung lautet:

WORKSPACE[n]={X 500, Y 500, Z 2000, A 0, B 0, C 0, X1 100, Y1 260, Z1 220, X2 -100, Y2 60, Z2 20, MODE #OUTSIDE }

2.5 Getriebemomentenüberwachung

2.5.1 Funktion

Die Getriebemomente des Roboters werden indirekt überwacht. Dies geschieht, indem aus den vorhandenen Motorströmen die jeweiligen Getriebemomente errechnet werden. Überschreiten diese berechneten Momente einen bestimmten Wert, wird eine Stopreaktion ausgelöst und eine entsprechende Meldung im Meldungsfenster ausgegeben.



Die Zusatzachsen werden nicht überwacht.

Da die Getriebemomente indirekt überwacht werden, entsprechen sie nicht in allen Fällen der Realität, da das tatsächliche Moment nicht meßbar ist. Die Überwachung ist beim CP- und PTP-Fahren aktiv. Hierfür muß die Beschleunigungsanpassung eingeschaltet sein.



Die Überwachung ist bei NOT-Aus und Drehzahlstop nicht aktiv!

Spricht die Momentenüberwachung an, so kann dies folgende Ursachen haben:

- G Überlastung des Roboters aufgrund eines nicht zulässigen Werkzeugs;
- G Fehlerhafter Druckausgleich der Achse 2;
- G Mechanisch schwergängige Achse.

2.5.2 Konfigurieren

Die Überwachung kann über das Maschinendatum "C:\Programme\KRC\MaDa\R1\SMA-CHINE.DAT" ein- bzw. ausgeschaltet werden. Suchen Sie in der Datei folgende Zeile:

```
:
  BOOL $GEAR_TORQ_MON=FALSE ; GETRIEBEMOMENTENUEBERWACHUNG AUS
:
```

und schalten die Überwachung ein, indem Sie der Variable den Wert "TRUE" zuweisen:

```
:
  BOOL $GEAR_TORQ_MON=TRUE ; GETRIEBEMOMENTENUEBERWACHUNG EIN
:
```

Um die Getriebemomentenüberwachung verwenden zu können, muß auch die Beschleunigungsanpassung eingeschaltet sein. Dies ist der Fall, wenn die Variable "\$ADAP_ACC" den Wert "#STEP1" besitzt. Sie finden diese Variable in der Datei "C:\Programme\KRC\MaDa\R1\ROBCOR.DAT":

```
:
  DECL ADAP_ACC $ADAP_ACC=#STEP1 ; BESCHLEUNIGUNGSANPASSUNG
:
```

Das maximal zulässige Getriebemoment errechnet sich aus dem maximalen Getriebemoment für die Bahnplanung multipliziert mit dem vorgegebenen Prozentfaktor "\$DYN_DATA[]". Dieser Faktor muß $\geq 100\%$ sein. Der Prozentfaktor wird in der Datei "C:\Programme\KRC\MaDa\R1\ROBCOR.DAT" hinterlegt und beträgt standardmäßig 150%.

```
:
  $DYN_DATA[231]=150.0
:
```

```

$DYN_DATA[ 232 ] = 150.0
$DYN_DATA[ 233 ] = 150.0
$DYN_DATA[ 234 ] = 150.0
$DYN_DATA[ 235 ] = 150.0
$DYN_DATA[ 236 ] = 150.0
:
```



Für Schäden, die aufgrund falscher Einstellungen entstehen, wird keine Haftung übernommen!

2.6 Kollisionsüberwachung

2.6.1 Funktion

Falls der Roboter mit einem Bauteil kollidiert oder mit seinem Werkzeug an einem Bauteil hängen bleibt, reagiert der Lage- und Drehzahlregler entsprechend. Die Sollmomente der beteiligten Achsen werden automatisch erhöht. Abhängig vom Anwendungsfall kann der Roboter den Widerstand überwinden und seine Bewegung fortsetzen. Hierbei können Werkzeug bzw. Bauteil beschädigt werden.



Die Zusatzachsen werden nicht überwacht.

Der Anwender kann sowohl die Momentengrenzen als auch die Ansprechzeit konfigurieren. Über eine KRL-Variable wird ein Bereich (ein sogenannter "Überwachungsschlauch") um das jeweilige Moment gelegt. Verläßt das Moment diesen Überwachungsschlauch, wird nach der angegebenen Ansprechzeit eine bahntreue Stopreaktion ausgelöst und eine entsprechende Meldung im Meldungsfenster ausgegeben.

Im Normalfall ist die Überwachung eingeschaltet, der Standardwert des Überwachungsschlauchs beträgt jedoch hierbei 200%. Für einzelne Bewegungen oder Programmteile kann die Überwachung nun sensibler eingestellt werden.



Bei einem Reset, Satzanwahl und Programmbwahl werden die eingestellten Grenzen auf den Standardwert von 200% zurückgesetzt.



Die Kollisionsüberwachung bietet keine Garantie gegen Beschädigungen, kann aber die Größe des jeweiligen Schadens reduzieren. Ausschlaggebend für die Art der jeweiligen Beschädigung sind sowohl die gefahrene Geschwindigkeit als auch die aufgetretenen Momente.

2.6.2 Konfigurieren

Um die Kollisionsüberwachung verwenden zu können, muß auch die Beschleunigungsanpassung eingeschaltet sein. Dies ist der Fall, wenn die Variable "\$ADAP_ACC" den Wert "#STEP1" besitzt. Sie finden diese Variable in der Datei "C:\Programme\KRC\MaDa\R1\\$_ROBCOR.DAT":

```

:
DECL ADAP_ACC $ADAP_ACC=#STEP1 ; BESCHLEUNIGUNGSANPASSUNG
:
```



Die Lastdaten sind für die Kollisionsüberwachung korrekt zu ermitteln. Außerdem müssen die Momentengrenzen an den jeweiligen Anwendungsfall angepaßt werden.

Die Ansprechzeit der Überwachung kann mit der Variablen "\$TORQMON_TIME" in [ms] eingestellt werden. Diese Variable befindet sich in der Datei "C:\Programme\KRC\MaDa\Steu\\$CUSTOM.DAT":

```
:
REAL $TORQMON_TIME=0.0 ; ZEIT FUER FAHRMOMENTENUEBERWACHUNG
:
```

Die Größe des Überwachungsschlauchs wird durch folgende Variablen in der Datei "C:\Programme\KRC\MaDa\\$CUSTOM.DAT" vorgegeben:

G Standardwerte des Überwachungsschlauchs für den Programmbetrieb in Prozent

```
:
$TORQMON_DEF[1]=200
$TORQMON_DEF[2]=200
$TORQMON_DEF[3]=200
$TORQMON_DEF[4]=200
$TORQMON_DEF[5]=200
$TORQMON_DEF[6]=200
:
```

G Standardwerte des Überwachungsschlauchs für den Kommandobetrieb in Prozent

```
:
$TORQMON_COM_DEF[1]=200
$TORQMON_COM_DEF[2]=200
$TORQMON_COM_DEF[3]=200
$TORQMON_COM_DEF[4]=200
$TORQMON_COM_DEF[5]=200
$TORQMON_COM_DEF[6]=200
:
```



Die Breite des Toleranzbandes ergibt sich aus dem maximalen Moment in [Nm] multipliziert mit der Prozentsatz von "\$TORQMON_...".

In der Datei "\$OPERATE.SRC", die nur über das Dateifenster der Bedienoberfläche zu erreichen ist, stehen die Variablen "\$TORQMON[1]...[6]" und "\$TORQMON_COM[1]...[6]" zur Verfügung.

Eine Überwachung der Fahrmomente erfolgt im

G Programmbetrieb

- Vorlauf:
Innerhalb des KRL-Programms kann der Variable "\$TORQMON[]" ein satzbezogenes Toleranzband für das Moment vorgegeben werden.
- Hauptlauf:
Die Überwachung wird sofort aktiv bzw. inaktiv, wenn die Variable im Interruptprogramm beschrieben wird.

G Kommandobetrieb

- Standardmäßig sind für die Momentenüberwachung die Werte der Variablen "\$TORQMON_COM_DEF[1]...[6]" aus der Datei "\$CUSTOM.DAT" gültig. Der Anwender kann über die Variablenkorrektur die Überwachungsgrenze jederzeit ändern, indem er die Werte von "\$TORQMON_COM[1]...[6]" ändert.

Über die Systemvariable "\$TORQ_DIFF[1]...[6]" kann die maximale aufgetretene Momentenabweichung in Prozent abgelesen werden. Diese Variable kann daher zum optimieren der Momentenüberwachung genutzt werden:



Vor einem Bewegungssatz bzw. Bewegungsabschnitts setzen Sie die Variable mit Hilfe der Variablenkorrektur auf "0".

Anzeige

The diagram illustrates the steps to access the variable configuration window:

- Click **Anzeige**.
- Open the menu:
 - 0 Ein/Ausgänge
 - 1 Istposition
 - 2 Zyklische Flags
 - 3 Flags
 - 4 Zähler
 - 5 Timer
 - 6 Variable** (selected)
 - 7 Diagnose
 - 8 Werkzeugdefinition
- Open the sub-menu:
 - 0 Korrigieren
 - 1 Anzeigen

The resulting window shows:

- Name:** \$TORQ_DIFF[1]
- Aktueller Wert:** 137
- Neuer Wert:** 0
- Modul:** "/R1/TEST_SRC"

Nun führen Sie den Bewegungssatz aus und lesen die Variable erneut aus. Der ausgegebene Wert entspricht dem aufgetretenen Momentenmaximum.

Setzen Sie nun den Überwachungsschlauch auf den Wert von "\$TORQ_DIFF[]" zusätzlich einer Sicherheit von 5-10%.



Die Variable "\$TORQ_DIFF[]" kann nur von "R_INT", "S_INT" sowie der Variablenkorrektur ausgelesen werden. Der Variable kann nur der Wert "0" zugewiesen werden.



Für Schäden aufgrund falscher Einstellungen wird keine Haftung übernommen!

2.7 Motorstromüberwachung (I²t-Überwachung)

2.7.1 Funktion

Diese Funktion dient zur Überwachung des Motorstroms in der Motorleitung vom Verstärker bis zum Motor. Sie ist als Ergänzung zur bisherigen Temperatur-Überwachung in den Motoren gedacht. Der Vorteil dieser Leitungsabsicherung besteht darin, daß eine exakte Anpassung an den jeweiligen Leitungsquerschnitt möglich ist.

Aus den Sollströmen des Stromreglers wird über den Zeitraum von einer Minute jeweils ein Mittelwert gebildet.

Überschreitet der Motorstrom 95% bzw. 100% des maximal zulässigen Dauerstroms, wird der Roboter angehalten und es erfolgt die Ausgabe einer Zustandsmeldung im Meldungs-fenster:

| | Zeit | Nr. | Abs. | Meldung |
|---|-------|------|------|---|
| ! | 12:07 | 1350 | | SAK erreicht |
| i | 12:07 | 123 | | i ² t Überwachung, Strombelastung der Motorleitung A2 nach 59.5 s 95 % erreicht. |

| | Zeit | Nr. | Abs. | Meldung |
|---|-------|------|------|--|
| ! | 13:39 | 1350 | | SAK erreicht |
| i | 13:46 | 125 | | i ² t Überwachung, Strombelastung der Motorleitung A2 nach 60.0 s 100 % |

Unterschreitet der Motorstrom die entsprechenden Werte wieder, wird eine Quittierungsmeldung ausgegeben.

| | Zeit | Nr. | Abs. | Meldung |
|------|-------|------|------|--|
| ! | 12:07 | 1350 | | SAK erreicht |
| STOP | 13:40 | 1242 | | Quit i ² t Überwachung, Strombelastung der Motorleitung A2 nach 59.5 s 95 % |

| | Zeit | Nr. | Abs. | Meldung |
|------|-------|------|------|---|
| ! | 13:39 | 1350 | | SAK erreicht |
| STOP | 13:40 | 1241 | | Quit i ² t Überwachung, Strombelastung der Motorleitung A2 nach 60.0 s 100 % |

Quitt

Erst nach quittieren wird die Zeile aus dem Meldungs-fenster entfernt, und der Roboter kann wieder verfahren werden.

2.7.2 Konfigurieren

Der zulässige Dauerstrom wird in der Datei "C:\Programme\KRC\MaDa\R1\SMACHINE.DAT" für bis zu 12 Achsen vorgegeben. Die Einheit wird in [A] angegeben:

```

:
REAL $CURR_MON[ 12] ; ZULAESSIGER NENNSTROM
$CURR_MON[ 1]=12.8
$CURR_MON[ 2]=12.8
$CURR_MON[ 3]=12.8
$CURR_MON[ 4]=6.7
$CURR_MON[ 5]=6.7
$CURR_MON[ 6]=6.7

```

```
$CURRE_MON[ 7]=0.0  
$CURRE_MON[ 8]=0.0  
$CURRE_MON[ 9]=0.0  
$CURRE_MON[10]=0.0  
$CURRE_MON[11]=0.0  
$CURRE_MON[12]=0.0  
:
```



Die Stromgrenze ist in Abhängigkeit vom Motorleitungsquerschnitt sowie dem maximal zulässigen Motorstrom einzustellen.

Für Schäden aufgrund falscher Einstellungen wird keine Haftung übernommen!

