

2 System konfigurieren Experte

2.1 Konfigurationsdateien

2.1.1 Projektierbare Anzeige – Datei <ConfigMon.INI>

Diese Datei liegt standardmäßig im Verzeichnis "C:\KRC\Roboter\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 Nebestehende Tabelle benötigt die folgenden Einträge in der Datei "ConfigMon.ini":

```
[Version]
Version=1.0

.
.
.

[Group4]
GroupTitle=ManuellerSicherheitstest
NmbDataCols=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.

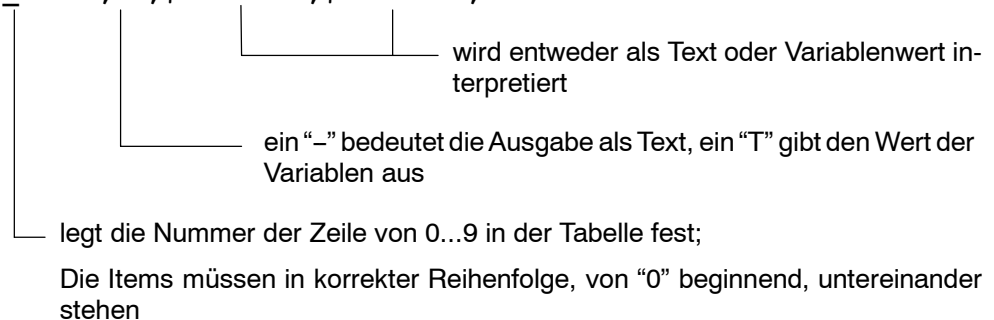
MaxColWidths=1800;1500;

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;

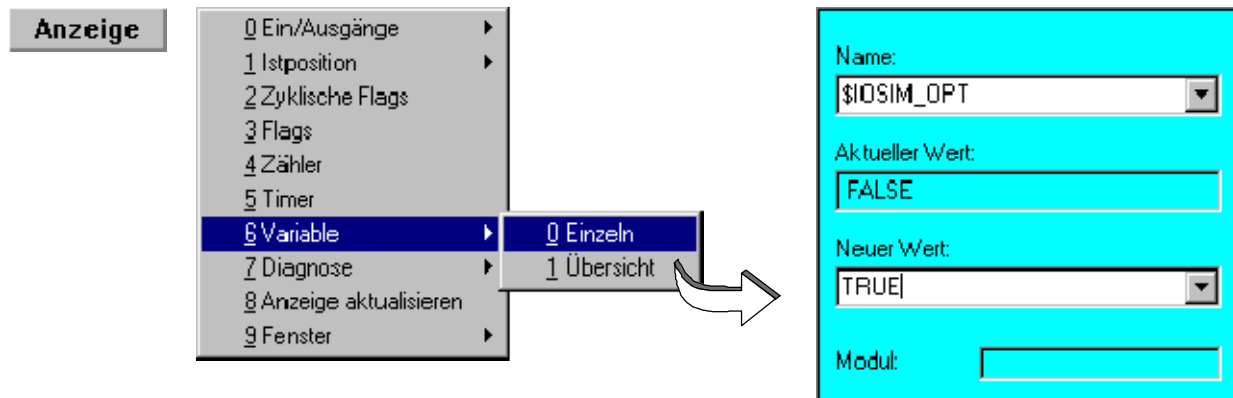


2.2 Simulierte Ein-/Ausgänge (IO-Simulation)

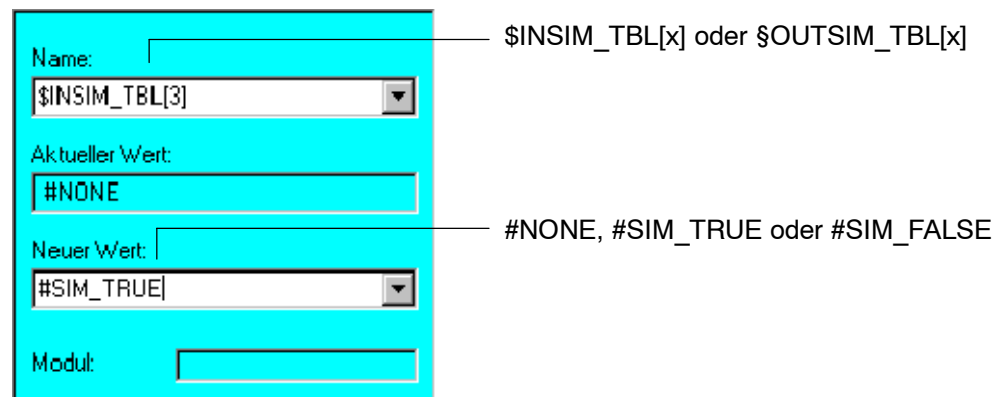
Diese Funktion gestattet es, bestimmte Ein- sowie Ausgänge zu simulieren. Steht beispielsweise die Eingangsperipherie noch nicht zur Verfügung, können die erforderlichen Eingänge einfach per Simulation auf "TRUE" oder "FALSE" gesetzt werden. Das gleiche Prinzip gilt auch für die Ausgänge.

2.2.1 Funktion

Zum Einschalten der Simulation steht die Variable "**\$IOSIM_OPT**" zur Verfügung. Öffnen Sie hierzu die Variablenkorrektur und ändern Sie deren Wert auf "TRUE".



Die Simulation ist damit zwar grundsätzlich eingeschaltet, es müssen aber noch die betroffenen Ein- bzw. Ausgänge gesetzt werden. Hierzu dienen die Variablen "**\$INSIM_TBL[x]**" und "**\$OUTSIM_TBL[x]**".



Jedem der Ein- bzw. Ausgänge kann einer der folgenden Zustände zugewiesen werden:

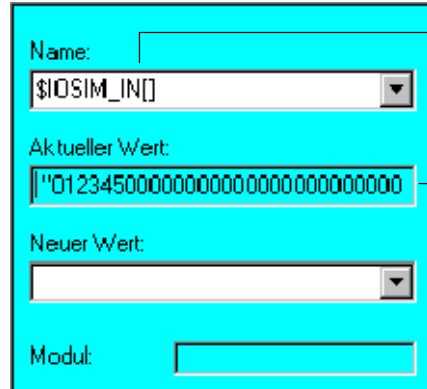
- **#NONE** Der Ein- bzw. Ausgang wird nicht simuliert und bleibt damit unverändert;
- **#SIM_TRUE** Der Ein-/Ausgang wird gesetzt, was in der Anzeige der Digitalen Ein-/Ausgänge abgelesen werden kann;
- **#SIM_FALSE** Der Ein- bzw. Ausgang wird ausgeschaltet, was ebenfalls in der Anzeige der Digitalen Ein-/Ausgänge abgelesen werden kann.



Einige Ein- bzw. Ausgänge dürfen aus Systemgründen nicht verwendet werden. Sie sind mit einem Schreibschutz versehen und können daher nicht verändert werden.

Ausgabe von Arrays

Die Variablen “\$IOSIM_IN[]” und “\$IOSIM_OUT[]” geben alle Ein- bzw. Ausgänge in einer Zeile aus. Jedem Ein- bzw. Ausgang ist eine Ziffer zugeordnet, deren Bedeutung nachfolgend beschrieben wird.



Name: \$IOSIM_IN[] oder \$IOSIM_OUT[]
 Aktueller Wert:
 Neuer Wert:
 Modul:

Ausgabe der Belegung aller Ein-/Ausgänge

Jeder der Ein- bzw. Ausgänge kann einen der folgenden Zustände annehmen:

- 0 Kein Signal (FALSE)
- 1 Signal liegt an (TRUE)
- 2 Simuliertes Signal ist auf “FALSE” gesetzt (ausgeschaltet)
- 3 Simuliertes Signal ist auf “TRUE” gesetzt (eingeschaltet)
- 4 System Signal ist auf “FALSE” gesetzt (ausgeschaltet)
- 5 System Signal ist auf “TRUE” gesetzt (eingeschaltet)

Im Zustandsfenster der Digitalen Eingänge würde diese Belegung folgendermaßen aussehen:



Eingang			
	SYS	SIM	Eingang
1			Eingang
2			Eingang
3		SIM	Eingang
4		SIM	Eingang
5	SYS		Eingang
6	SYS		Eingang
7			Eingang
8			Eingang
9			Eingang

Kein Signal an Eingang 1 (FALSE)
 Signal an Eingang 2 (TRUE)
 Kein Signal an Eingang 3 (FALSE)
 Signal an Eingang 4 (TRUE)
 Kein Signal an Eingang 5 (FALSE)
 Signal an Eingang 6 (TRUE)

Auch die Variablen “\$IN[]” und “\$OUT[]” geben die Belegung der Ein- bzw. Ausgänge in einer Zeile aus, wenn keine Nummer angegeben wird. Allerdings wird nicht zwischen physikalischen und simulierten I/O's unterschieden, sondern nur, ob ein Signal anliegt oder nicht.



Name:

Aktueller Wert:

Neuer Wert:

Modul:

\$IN[] oder \$OUT[]

Belegung der Ein-/Ausgänge

Jeder der Ein- bzw. Ausgänge kann einen der folgenden Zustände annehmen:

- 0 Kein physikalisches Signal (FALSE)
- 1 Physikalisches Signal liegt an (TRUE)

2.2.2 Optionen

Die nachfolgend beschriebenen Optionen sind für die Simulation der Ein-/Ausgänge nicht notwendig, aber hilfreich.

Zustimmtaste (\$OUT_NODRIVE)

Zum Umschalten der Ausgänge muß normalerweise einer der Zustimmtaster gedrückt gehalten werden. Durch Setzen der Variablen “\$OUT_NODRIVE” auf “TRUE” kann dies unterbunden werden.

Betriebsart Automatik Extern (\$IOBLK_EXT)

In Verbindung mit \$OUT_NODRIVE = TRUE gestattet die Variable “\$IOBLK_EXT” das Setzen von Ausgängen in der Betriebsart Automatik Extern, was normalerweise nicht möglich ist. Dazu muß “\$IOBLK_EXT” auf den Wert “FALSE” gesetzt werden.

2.2.3 Verwendete Variablen

Variable	Wertebereich	Bedeutung
\$IOSIM_OPT	TRUE FALSE	Simulation aktiv Simulation inaktiv
\$INSIM_TBL[x] x = 1 ... 1024	#NONE #SIM_TRUE #SIM_FALSE	Eingang wird nicht simuliert Eingang wird eingeschaltet Eingang wird ausgeschaltet
\$OUTSIM_TBL[x] x = 1 ... 1024	#NONE #SIM_TRUE #SIM_FALSE	Ausgang wird nicht simuliert Ausgang wird eingeschaltet Ausgang wird ausgeschaltet
\$IOBLK_EXT *1	TRUE FALSE	Kein Setzen von Ausgängen in #EXT Ausgänge können gesetzt werden
\$OUT_NODRIVE	TRUE FALSE	Zustimmtaster braucht zum Umschalten der Ausgänge nicht gedrückt zu werden Zustimmtaster muß betätigt werden
\$IOSIM_IN[]		Anzeige aller Eingänge
\$IOSIM_OUT[]		Anzeige aller Ausgänge
\$IN[x] x = 1 ... 1026 \$IN[]	TRUE FALSE	Eingang gesetzt Eingang nicht gesetzt Alle Eingänge
\$OUT[x] x = 1 ... 1024 \$OUT[]	TRUE FALSE	Ausgang gesetzt Ausgang nicht gesetzt Alle Ausgänge
*1 Nur in Verbindung mit \$OUT_NODRIVE = TRUE		



Wird ein Ausgang simuliert, kann dieser während der Simulation nicht mehr über die Variablenkorrektur ("OUT[x]") oder im Zustandsfenster umgeschaltet werden. Dazu muß die Simulation für diesen Ausgang zuerst ausgeschaltet werden.

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\\MaDa\\\$machine.dat"

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

```
&PARAM VERSION=3.4.0
&REL 4
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 $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:\\KRC\\Roboter\\KRC\\R1\\MaDa**".

2.3.2 Datei “\Steu\MaDa\\$machine.dat”

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

```
&PARAM VERSION=3.4.0
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:\KRC\Roboter\KRC\Steu\MaDa”.

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:

```
C:\KRC\Roboter\KRC\STEU\MADA\%CUSTOM.DAT :
```

```
DEFDAT %CUSTOM 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}
```

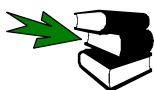
```
...
```

```
ENDDAT
```



Arbeitsräume können auch in *.SRC-Dateien definiert, bzw. ein- und ausgeschaltet werden. Die dort angegebenen Werte werden automatisch in die Datei "%CUSTOM.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. n 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. n verletzt" ausgegeben.

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

C:\KRC\Roboter\KRC\STEU\MADA\MACHINE.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

```

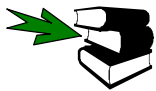
Aus der Komponente \$WORKSPACE[n].STATE kann der Zustand des zugeordneten Ausgangs ausgelesen werden.

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"**.

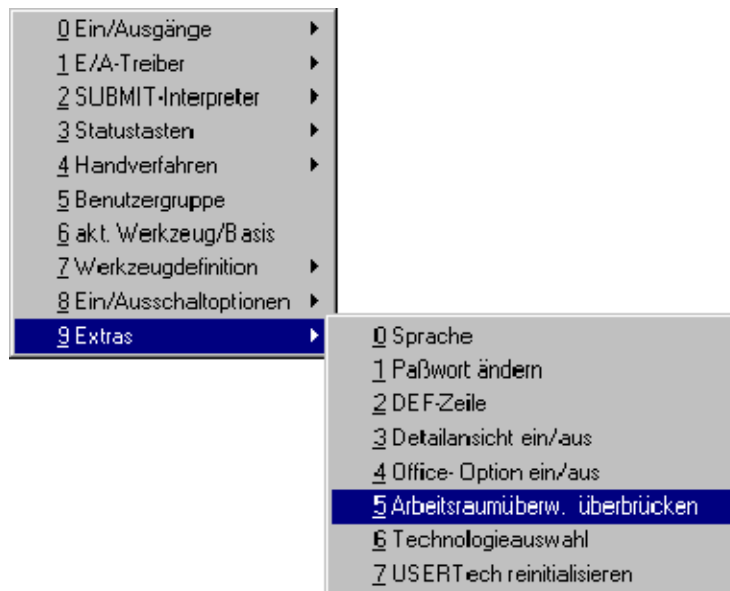


Die Überwachung des betreffenden Arbeitsraumes bleibt solange abgeschaltet, bis die Komponente wieder auf einen Wert ungleich "#OFF" gesetzt wird.

Überbrücken der Arbeitsraumüberwachung über Menü

Konfig.

Das "Überbrücken" der Arbeitsraumüberwachung ist über die Menüfunktion "Konfig." -> "Extras" -> "Arbeitsraumüberw. überbrücken" 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 überbrückt, 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.

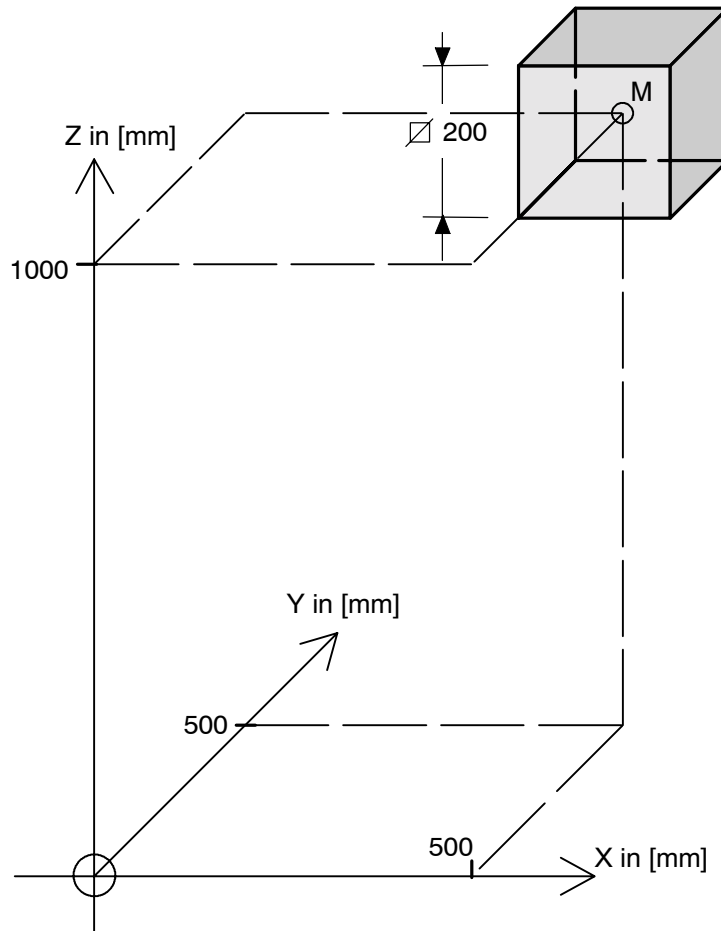


Fehlerhafte “\$TOOL”-Daten können zu unvorhersehbaren Situationen führen!

2.4.2 Beispiele



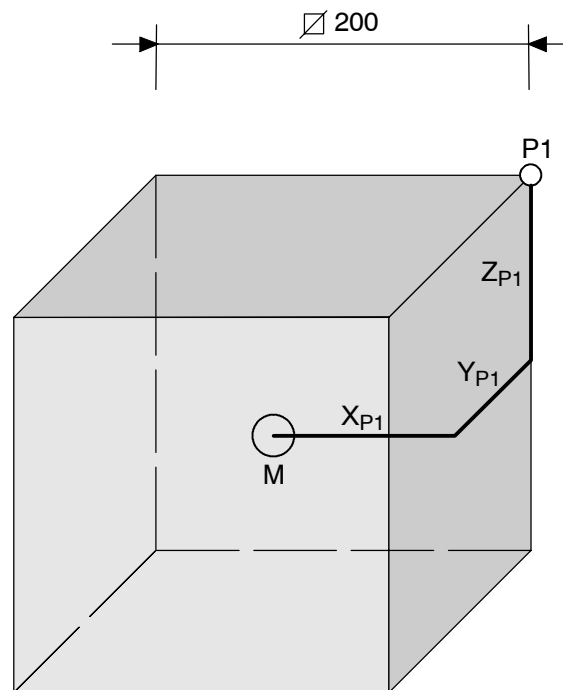
Es soll ein kubischer Arbeitsraum mit 200 mm Kantenlänge überwacht werden. Sein Mittelpunkt soll bei $X=500\text{mm}$, $Y=500\text{mm}$, $Z=1000\text{mm}$ liegen. Die Winkel A, B und C besitzen den Wert "0".



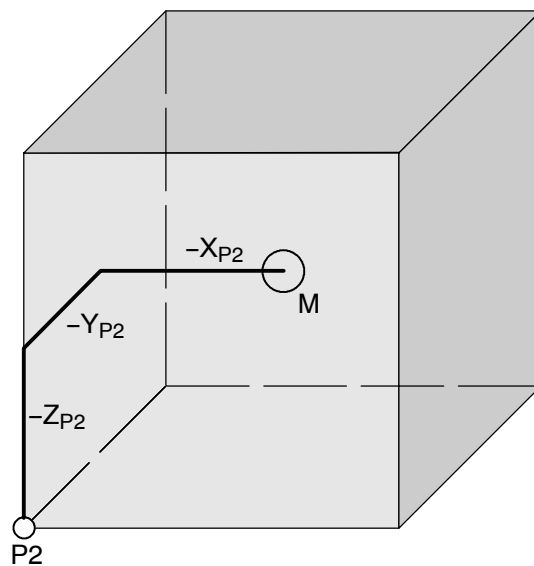
Der Mittelpunkt des Arbeitsraums (Punkt "M") wird im Workspace-Befehl folgendermaßen definiert:

```
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}
```

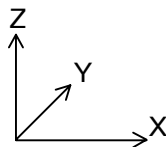
Der Arbeitsraum soll eine Seitenlänge von 200mm besitzen. Vom Mittelpunkt "M" ausgehend wird der Bereich über die Punkte "P1" in positiver und "P2" in negativer Richtung definiert.



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



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



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

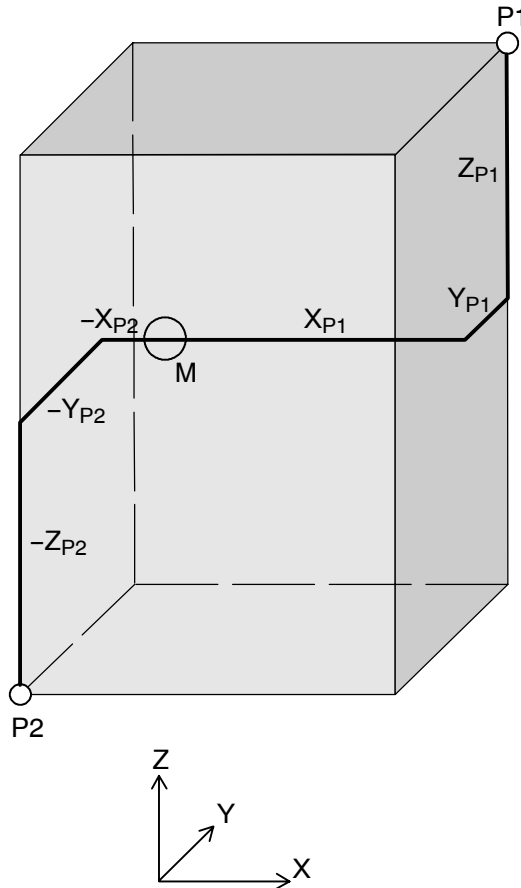
```
WORKSPACE[n]={X 500, Y 500, Z 1000, A 0, B 0, C 0, x1 100, y1 100, z1 100, x2 -100, y2 -100, z2 -100, MODE #INSIDE}
```

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.



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

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

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

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

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

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

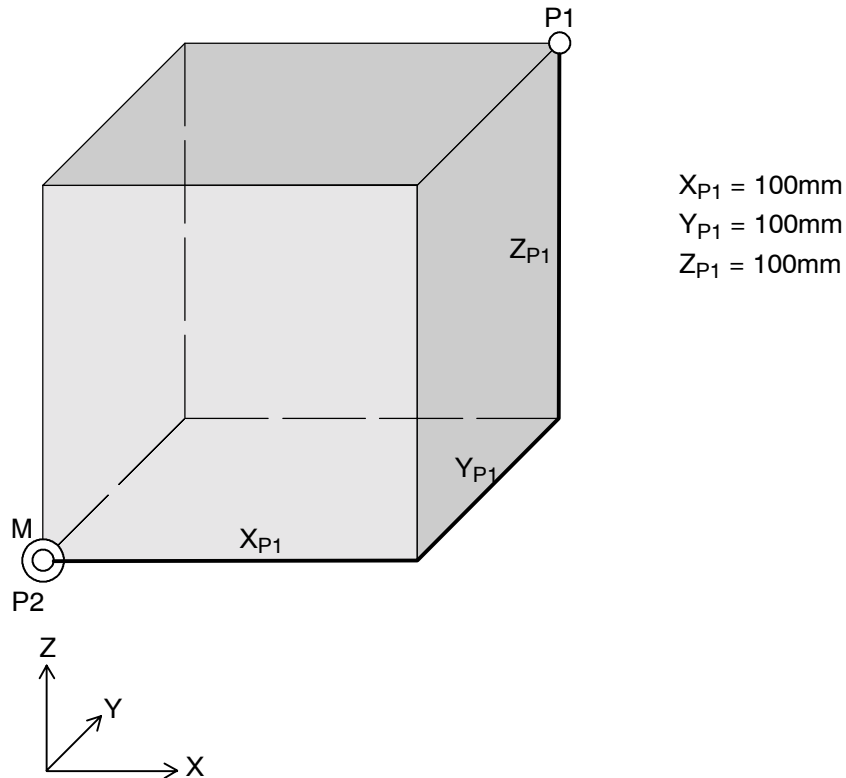
Der zugeordnete Ausgang soll wieder gesetzt werden, wenn sich der Bezugspunkt von Werkzeug, bzw. Werkstück innerhalb des Arbeitsraums befindet. Gleichzeitig soll der Roboter stoppen und eine Fehlermeldung ausgeben.

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.



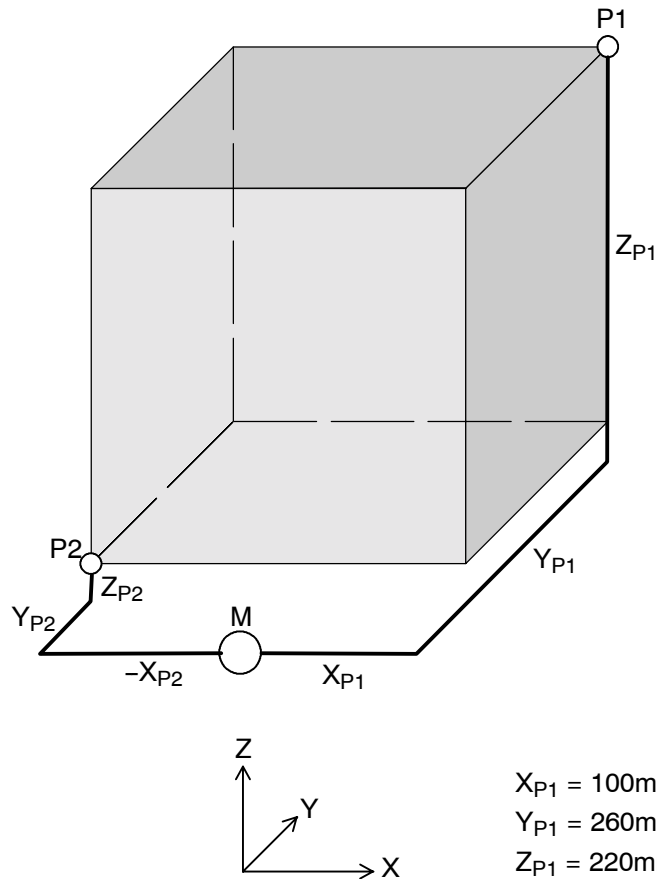
Befindet sich der Bezugspunkt von Werkzeug bzw. Werkstück außerhalb des Arbeitsraums, soll der zugeordnete Ausgang gesetzt werden. Gleichzeitig soll der Roboter stoppen und eine Fehlermeldung ausgeben.

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, MODE #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 Momentenbetrieb (Soft Servo)

2.5.1 Allgemein

Ab den Softwareständen KR C1, Release 2.2 und VKR C1, Release 1.2 können einzelne oder auch mehrere Achsen in den Momentenbetrieb geschaltet werden.



Im Normalbetrieb arbeitet der Roboter von außen wirkenden Kräften entgegen, um die programmierte Bahn zu fahren.

Dies kann bei zu hohen Kräften zu Schäden am Teil, Werkzeug oder Roboter führen! Gleiches gilt, wenn der Roboter innerhalb der programmierten Bahn gegen das Bauteil oder ein Hindernis fährt.

Bei bestimmten Applikationen – beispielsweise das Entladen von Druckgußmaschinen oder das Arbeiten mit elektromotorischen Punktschweißzangen – ist es jedoch erforderlich, daß der Roboter "nachgibt".

Momentenbetrieb bedeutet, daß eine Achse entweder ein definiertes Moment, vorgegeben durch die Begrenzung des Drehzahlreglerausganges (positive und negative Grenze), gegen einen Widerstand aufbauen kann (Drücken oder Ziehen) oder, daß die Achse durch äußere Krafteinwirkung bewegt werden kann (Wegdrücken der Roboterhand bzw. des daran befindlichen Werkzeuges).



Achsen, die in den Momentenbetrieb geschaltet sind, können **nicht** in Abhängigkeit mit anderen Achsen gefahren werden.

Bei Abschalten des Momentenbetriebes fährt die Achse von der aktuellen Position auf die nächste Position im Programm.

2.5.1.1 Einschränkungen, Risiken

Technisch ist der Momentenbetrieb grundsätzlich für alle Roboterachsen möglich, aufgrund technisch bedingter Einschränkungen sowie teils damit verbundener Gefahren jedoch nicht für jede Achse sinnvoll. Nachfolgende Tabelle gibt über die Möglichkeiten und Einschränkungen Aufschluß.

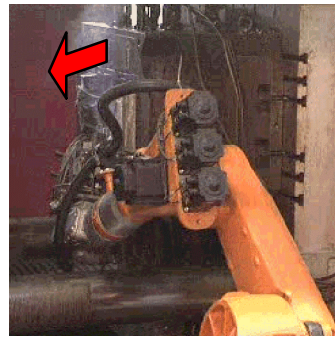
Achse	Momentenbetrieb, Einschränkungen, Risiken
1	Bei Boden- oder Deckenmontage uneingeschränkt und ohne besondere Gefahren möglich . Bei Wandmontage hingegen ist kein Momentenbetrieb möglich, weil die Gefahr des Durchsackens besteht.
2	Kein Momentenbetrieb möglich . Aufgrund unterschiedlicher, von der Position des Roboterarms abhängiger Momente und ggf. des positionsabhängigen Gewichtsausgleiches besteht die Gefahr des Durchsackens.
3	Momentenbetrieb grundsätzlich möglich , sollte jedoch wegen der Gefahr des Durchsackens aufgrund unterschiedlicher, von der Position der Schwinge abhängiger Momente vermieden werden.
4 5 6	Momentenbetrieb grundsätzlich möglich , jedoch nicht sinnvoll. Aufgrund des geringen Rückwirkungsgrades der Handachsgetriebe sind sehr hohe Kräfte erforderlich.

2.5.1.2 Beispiel Momentenbetrieb

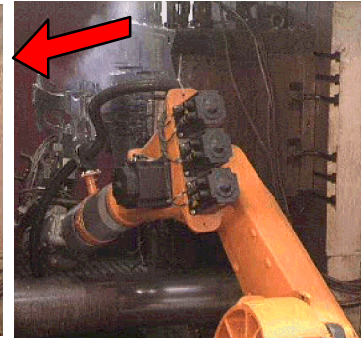
Entladung einer Druckgußmaschine



Roboter greift Bauteil.



Teil wird aus der Form gelöst. Der Ausstoßhub wird durch Drehung des Roboters um Achse 1 abgefangen.



Teil wird vom Roboter übernommen und aus dem Bereich der Druckgußmaschine heraus gefahren.

- Der Roboter greift das noch in einer der Formhälften sitzende Bauteil.
- Das Lösen des Teiles aus der Form geschieht durch einen hydraulischen Ausstoßer.
- Der Hub des Ausstoßers kann in der kurzen Zeit, in der dieser erfolgt, nicht meßtechnisch erfaßt und von der Robotersteuerung ausgewertet werden.
- Durch den Momentenbetrieb der Achse 1 gibt der Roboter bei entsprechender Anordnung nach, so daß der Ausstoßhub durch eine Drehung um Achse 1 abgefangen wird und keine schädlichen Belastungen auf den Roboter wirken.



Eine schräg in den Raum wirkende Ausstoßbewegung, die eine programmierte Bahn unter Einbeziehung mehrerer Achsen für den Momentenbetrieb erfordern würde, ist nicht umsetzbar.

Wird jedoch der Roboter so angeordnet, daß die Bodenflanschfläche parallel zur Ebene der Ausstoßbewegung liegt, beispielsweise durch schräge Montage des Roboters, ist Momentenbetrieb der Achse 1 möglich.



Die Schräglage des Roboters ist nur innerhalb bestimmter Grenzen zulässig. Siehe hierzu Abschnitt 2.5.1.

Durch entsprechendes Einstellen der negativen und positiven Grenze des Drehzahlreglerausganges kann erreicht werden, daß der Roboter das aus der Schwerkraft resultierende Moment ausgleicht, ja sogar am Teil zieht und sich ohne Gegenkraft vom Ausstoßer wegdrücken läßt.

2.5.2 Funktionsweise

Mit der Systemvariablen " $\$CURR_RED[x, x]$ " kann die Begrenzung des Drehzahlregler-Ausgangs verändert werden. Für die positive sowie die negative Grenze ist jeweils ein Index vorhanden. Die Aktivierung der jeweiligen Achse geschieht im Bitfeld " $\$TORQUE_AXIS$ ".

Durch Manipulation der Drehzahlreglergrenze ist definierbar, daß die Achse entweder weich ist oder mit einer definierten Kraft drückt.

Wird der Momentenbetrieb für eine Achse aktiviert, sind die Überwachungen "Stellgröße", "Stillstand", "Positionierzeit" und "Motor blockiert" für diese Achse abgeschaltet. Dabei wird nur noch die Istgeschwindigkeit dieser Achse überwacht. In der Betriebsart "Test 1" darf die Geschwindigkeit nicht höher sein als

$$\$RED_T1 * \$VEL_ACT_MA$$

der maximalen Achsgeschwindigkeit (Maschinendatum $10\% * 110\% = 11\%$) werden, weil sonst eine Überwachung anspricht und die Antriebe abschalten.

In der Betriebsart Test 2 und Automatik kann die Überwachungsgrenze mit der Variablen “\$TORQ_VEL[]” im Programm beeinflusst werden. Dabei sind Geschwindigkeiten bis 150% zulässig.

Wird der Momentenbetrieb abgeschaltet, fährt die Achse von der aktuellen Position zur nächsten Position im Programm weiter. Ein Beschreiben der Variablen “\$TORQUE_AXIS” löst einen Vorlaufstop aus.



Bei Programmabwahl oder Reset wird der Momentenbetrieb automatisch abgeschaltet.

Beim Handverfahren wird der Momentenbetrieb während des Handverfahrens abgeschaltet und beim Rückpositionieren wieder eingeschaltet.

Beim Rückpositionieren bewegt sich die Achse nicht, falls diese weich geschaltet ist. Das System fährt intern jedoch auf SAK, d. h. die Starttaste muß solange festgehalten werden, bis SAK erreicht ist.

2.5.3 Beispiele für die Weichschaltung von Achsen

2.5.3.1 Weichschaltung Achse 1

Zur Weichschaltung der Achse 1 wird der Drehzahlreglerausgang auf 0% begrenzt. Dies hat zur Folge, daß die Kontrolle der Achse durch Lage- und Drehzahlregler entfällt, wodurch die Achse von außen bewegt werden kann.

\$TORQUE_AXIS = 'B000001'	; Achse 1 in Momentenbetrieb schalten
\$CURR_RED[1, 1] = 0	;positive Momentengrenze auf 0
\$CURR_RED[1, 2] = 0	;negative Momentengrenze auf 0
PTP {A1 90}	;Bewegung, das sich die Bremsen öffnen ;Achse ist weich und kann bewegt werden
WAIT FOR \$IN[17]	;warten auf Signal um Momentenbetrieb zu ;beenden (z.B. Ausstoßer)
\$TORQUE_AXIS = 0	;Achse wieder hart schalten
PTP {A1 -20}	;nächste Position anfahren

2.5.3.2 Weichschaltung Achse 3

Achse 3 kann weich geschaltet werden, indem man den Drehzahlreglerausgang auf das aktuelle Haltemoment begrenzt. Eine Kontrolle durch Lage- und Drehzahlregler findet nur mehr im Bereich des Haltemomentes die Achse statt, wodurch die Achse von außen bewegt werden kann.

Das größte Haltemoment wird erreicht, wenn die Achse 3 waagrecht steht.



Bei Beladung der Achse mit einer höheren Last als zulässig, sowie bei falscher Einstellung der Stromgrenze, sackt der Roboterarm nach unten durch!

PTP {A3 90}	; Achse 3 waagrecht
\$TORQUE_AXIS = 'B000100'	; Achse 3 in Momentenbetrieb schalten
\$CURR_RED[3, 1] = ABS(\$CURR_ACT[3])	;positive Momentengrenze auf ;Haltemoment
\$CURR_RED[3, 2] = ABS(\$CURR_ACT[3])	;negative Momentengrenze auf ;Haltemoment
PTP {A3 0}	;Bewegung A3 senkrecht ;Achse ist weich und kann bewegt werden

```

WAIT FOR $IN[17]                ;warten auf Signal um Momentenbetrieb zu
                                ;beenden (z.B. Ausstoßer)

$TORQUE_AXIS = 0                ;Achse wieder hart schalten

PTP {A1 -20, A3 80}            ;nächste Position anfahren

```

Die Achse 3 muß in jedem Fall in eine Position außerhalb des Druckbereiches gebracht werden (im Beispiel senkrecht), ansonsten kann die Achse nicht nach oben gedrückt werden, da der Lageregler mit dem Haltemoment versucht die Position zu halten und wenn man die Achse wegdrücken möchte, man das Gewicht der Achse überwinden muß.

Die Handachsen können ohne Gefahr mit Moment 0% weich geschaltet werden, jedoch hat das Getriebe der Handachsen einen für diese Maßnahme nicht gut geeigneten Rückwirkungsgrad; diese Achsen lassen sich relativ schwer von außen bewegen.



Achse 2 sollte wegen des Druckausgleichs und da sie in senkrechter Stellung das geringste Moment hat, niemals weich geschaltet werden. Die Gefahr, daß die Achse nach unten durchsackt, ist sehr groß!

2.5.4 Beispiel für Achse mit definiertem Moment



Nachfolgendes Beispiel zeigt den Einsatzes einer elektromotorischen Punktschweißzange:

Bei der elektromotorischen Punktschweißzange soll am Bauteil ein definiertes Moment aufgebaut werden. Dazu wird eine innerhalb des Bauteiles befindliche Position angefahren, und der aus dem definierten Moment resultierende Strom eingestellt.

Da diese "Achse" die programmierte Position nicht erreichen kann, weil die Zangenelektrode vorher das Blech berührt, drückt sie mit der definierten Kraft auf das Bauteil. Wird nur die Stromgrenze zum Hineinfahren begrenzt, und beträgt die zweite Stromgrenze 100 %, kann mit voller Geschwindigkeit wieder aus dem Bauteil heraus gefahren werden.

```

PTP {E1 0}                      ;Zange kurz vor Blechberührung fahren

$TORQUE_AXIS = 'B1000000'       ;Momentenbetrieb einschalten

$CURR_RED[7,1] = 20             ;definiertes Moment einstellen
                                ;positive Grenze baut Moment auf

PTP {E1 -10}                   ;10 mm in das Bauteil "fahren"
                                ;Moment wird aufgebaut

WAIT FOR SCHWEISSENDE           ;Signal "Schweißende" abwarten
                                ;"fliegend" positive Grenze wieder hochsetzen

TRIGGER WHEN DISTANCE=0 DELAY=50 DO $CURR_RED[7,1]=100

PTP {E1 20}                    ;Schweißzange öffnen

$TORQUE_AXIS = 0                ;Momentenbetrieb abschalten

```

2.5.5 Variablen für den Momentenbetrieb

Folgende Variablen stehen für den Momentenbetrieb zur Verfügung:

REAL \$CURR_ACT[12]

Aktueller Strom der Achsen 1 bis 12 in Prozent vom Verstärker-Strom

\$CURR_MAX * \$CURR_LIM (-100% bis +100%).

REAL \$CURR_RED[12,2]

Strombegrenzung der Achsen 1 bis 12 in Prozent vom maximal Strom (0% bis +100%).

Dabei ist Index 1 die positive und Index 2 die negative Grenze. Diese Grenzen sind Absolutwerte zwischen 0% und 100%.



Die Strombegrenzung birgt die Gefahr, daß die Achse nicht mehr das erforderliche Moment zum Halten, Bremsen oder Fahren aufbringen kann. Dadurch können Menschen und Maschinen gefährdet werden! Eine Begrenzung des Stroms darf nur im Zusammenhang mit "\$TORQUE_AXIS" verwendet werden.

INT \$TORQUE_AXIS

Dies ist ein Bitfeld für die momentenbetriebene Roboterachsen A1 bis A6 sowie die externen Achsen E1 – E6.

Das Setzen eines Bits für die entsprechende Achse schaltet diese Achse in den Momentenbetrieb. Die Überwachungsfunktionen für diese Achse werden abgeschaltet.



Ein Beschreiben dieser Variablen löst einen Vorlaufstop aus.

REAL \$TORQ_VEL[12]

Geschwindigkeitsgrenze in Prozent von der maximalen Geschwindigkeit zur Überwachung der momentenbetriebenen Achse.



Wenn eine Achse im Momentenbetrieb ist, sind alle Überwachungsfunktionen abgeschaltet.

Um sicherzustellen, daß im Falle eines Defektes oder Fehlers in der Hardware oder ein Durchsacken der Achse erkannt wird, findet eine Überwachung der Geschwindigkeit statt.

Mittels der Variablen "\$TORQ_VEL" kann im Programm die maximal zulässige Geschwindigkeit in den Betriebsarten T2 und Automatik eingestellt werden.

Für die Betriebsart T1 gilt die Geschwindigkeit, welche in den Maschinendaten eingestellt ist. Bei Überschreitung dieser Geschwindigkeit werden die Antriebe abgeschaltet und eine entsprechende Fehlermeldung ausgegeben.

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 ausgeschaltet, der Standardwert des Überwachungsschlauchs beträgt 200%. Für einzelne Bewegungen oder Programmteile kann die Überwachung nun sensibler eingestellt werden.



Bei einem Reset, Satzanwahl und Programmwahl werden die eingestellten Grenzen auf den Standardwert aus der Custom.dat 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.

Weiterhin stehen die Signalausgänge \$COLL_ENABLE und \$COLL_ALARM unter KRC:\STEU\MACHINE.DAT zur Verfügung. COLL_ENABLE wird gesetzt, wenn die aktuelle Überwachungsgrenze einer Achse für Programmbetrieb \$torqmon kleiner 200% ist. \$COLL_ALARM wird gesetzt, wenn die Meldung "117 Momentüberschreitung Ax" für eine Achse ausgegeben wird. Dieser Ausgang bleibt solange gesetzt wie \$STOPMESS ansteht.

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:\KRC\Roboter\KRC\R1\MaDa\ROBCOR.DAT":

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



Die Lastdaten sind für die Kollisionsüberwachung korrekt zu ermitteln. Außerdem muß der Überwachungsschlauch 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:\KRC\Roboter\KRC\Steu\MaDa\SCUSTOM.DAT":


```

:
REAL $TORQMON_TIME=0.0 ; ZEIT FUER FAHRMOMENTENUEBERWACHUNG
:

```

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

- 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
:

```

- 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_...".

Eine Überwachung der Fahrmomente erfolgt im

- 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.
- Kommandobetrieb
 - Standardmäßig sind für die Momentenüberwachung die Werte der Variablen "\$TORQMON_COM_DEF[1]...[6]" aus der Datei "SCUSTOM.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 →

- 0 Ein/Ausgänge
- 1 Istposition
- 2 Zyklische Flags
- 3 Flags
- 4 Zähler
- 5 Timer
- 6 Variable**
 - 0 **Einzel**
 - 1 Übersicht
- 7 Diagnose
- 8 Anzeige aktualisieren
- 9 Fenster

→

Name:

Aktueller Wert:

Neuer Wert:

Modul:

Nun führen Sie den Bewegungssatz aus und lesen die Variable erneut aus. Der ausgegebene Wert entspricht der aufgetretenen maximalen Momentabweichung.

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



Der Variable "\$TORQ_DIFF[]" 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	129		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:\KRC\Roboter\KRC\R1\MaDa\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  
$CURR_MON[ 7 ]=0.0  
$CURR_MON[ 8 ]=0.0  
$CURR_MON[ 9 ]=0.0  
$CURR_MON[ 10 ]=0.0  
$CURR_MON[ 11 ]=0.0  
$CURR_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!



Konfiguration
