

Sicherheit im Umgang mit Industrierobotern

Roboterkomponenten

iPendant

Roboter Controller

Robotersteuerung

Roboter Hauptmenü

Frames - Koordinatensysteme

Rechte Hand Regel

Einzelachsmodus Joint

Basis Koordinatensystem

Bewegungsarten Joint, Linear, Circular

Positionsregister

Tool Frame (TCP) / Vermessen

User Frame / Vermessen

Payload Einstellung

Das erste Roboterprogramm

Digitale Ein und Ausgänge

...



## Persönliche Schutzausrüstung



Arbeitsschutzkleidung



Schutzhandschuhe



Arbeitsschutzschuhe



Gehörschutz

### Persönliche Schutzausrüstung (PSA)

Sind Gefährdungen durch technische und organisatorische Schutzmaßnahmen nicht auszuschließen, dann sind der Arbeitgeber und die Arbeitgeberin verpflichtet, ihren Beschäftigten die erforderlichen persönlichen Schutzausrüstungen zur Verfügung zu stellen. Die Beschäftigten müssen die bereitgestellte persönliche Schutzausrüstung verwenden, solange eine Gefährdung besteht.

## Technische Sicherheitseinrichtungen



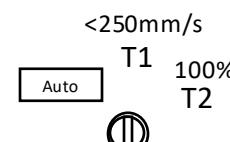
Lichtschranken



Handbediengerät



Zustimmungstasten



Schlüsselschalter

### Technische Sicherheitseinrichtungen

Für die Sicherheit von Industrierobotern gelten die europäisch harmonisierten Normen EN ISO 10218-1 und EN ISO 10218-2. Unter den Geltungsbereich dieser Normen fallen zum Beispiel die bekannten 6-Achsen-Knickarmroboter, Scara-Roboter oder auch Portalroboter. Die Roboternormen gelten für Systeme mit mindestens drei frei programmierbaren Achsen. Für Handhabungssysteme mit weniger Achsen existiert keine eigenständige Norm. Jedoch kann man sich an o.g. Norm anlehnen.

#### Trennende Schutzeinrichtungen:

- Kabinen
- Umzäunungen

#### Bewegliche trennende Schutzeinrichtungen:

- Schutztüren
- Rolltore
- Klappen

#### Nicht trennende Schutzeinrichtungen:

- Lichtvorhänge
- Laserscanner
- Schaltmatten
- Zweihandschaltungen

- **Roboter**

Sockel, Medienverbindungen (Luft, Wasser)  
Tooling, Kamera, Sensoren, Vakuumerzeuger



- **Verbindungsleitungen**

Kabel, Schläuche, Druckleitungen

- **Robotersteuerung**

Schalter, Schlüsselschalter, Lampen, NotAus

- **Programmiergerät**

Display, Lasten, NotAus

- **Software**

Roboterprogramme, SPS

- **Zubehör**



# Robotersteuerung

## Übersicht Robotersteuerung

- 1 Betriebsartenschalter
- 2 Fehler Reset
- 3 Zyklus Start / Programmstart
- 4 Fehleranzeige
- 5 Power ON/OFF
- 6 NOTAUS
- 7 Steuerung ON/OFF



**Ein/Aus Schalter**  
In Kombination mit dem Zustimmungstaster wird das Verfahren des Roboters möglich.

**Funktionstasten**  
wechselnd Menüs

**Reset Taste**

**Numerische Tasten**



**Robotergeschwindigkeit**  
Regelung der Geschwindigkeit

**NotAus Taste**  
Stoppt das laufende Programm, bremst alle Achsen und schaltet den Strom der Servos aus.

**Display**

**Tasten zur Roboterbewegung**  
manuelle Bewegung des Roboters



## Status

- | Status | Funktion   |
|--------|--|
| Fault  | Ein Fehler steht an  |
| Hold   | Die HOLD-Taste ist gedrückt oder das HOLD-Signal der SPS fehlt |
| Step   | Einzeilschrittmodus ist aktiviert                              |
| Busy   | Der Roboter arbeitet bzw. das Programm wird ausgeführt.        |
| Run    | Das Programm wird vom Roboter ausgeführt.                      |

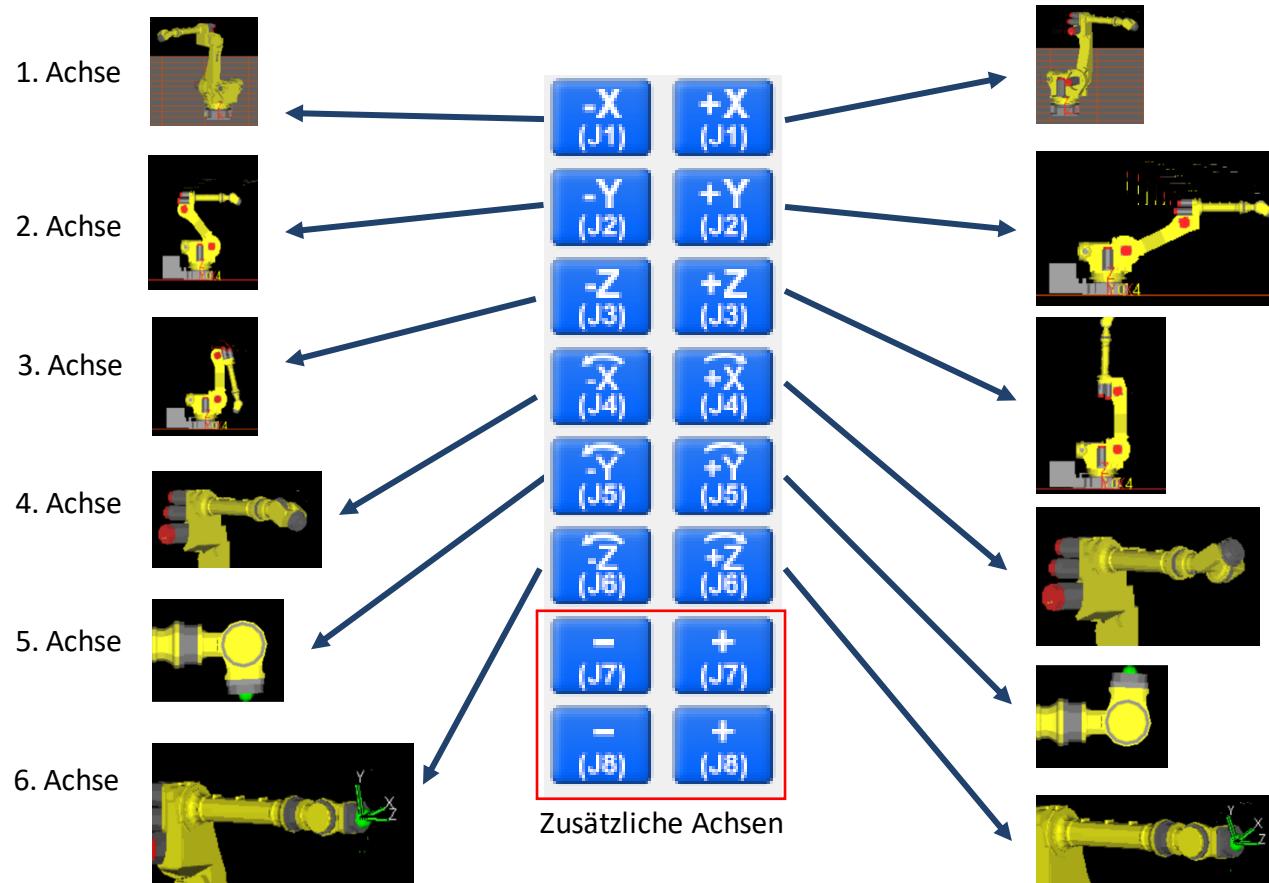
## Erweiterte Funktionen

- | Funktion | Beschreibung  |
|----------|---|
| Weld     | Schweißen ist aktiviert   |
| Joint    | Das Joint-Koordinatensystem als Verfahr Koordinatensystem ist aktiv |
| XYZ      | Das Kartesische Koordinatensystem ist aktiviert (JGFRM,USER, WORLD) |
| TOOL     | Werkzeug-Koordinatensystem wird verwendet.                          |

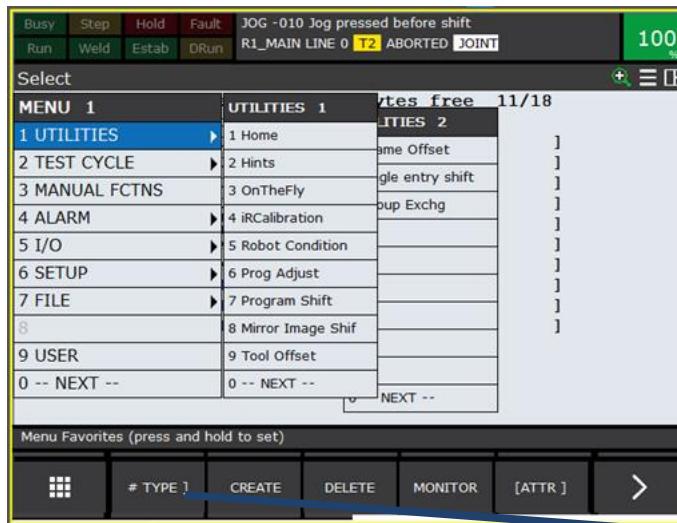
Funktionstasten mit allen Softwarefunktionen zur Programmierung



# Bewegung mit 6 Einzelachsen



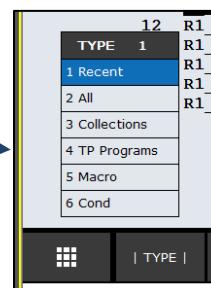
# Menu und Type



## Menu 1

**UTILITIES**  
**TEST CYCLE**  
**MANUAL FUNCTION**  
**ALARM**  
**I/O**  
**SETUP**  
**FILE**  
**USER**  
**NEXT**

Anzeige von Hinweisen  
 Spezifizierung der Test  
 Ausführung von Makro Programmen  
 Darstellung der zeitlichen Abfolge von Alarmmeldungen  
 Anzeige der Ausgänge und simulierte Ein/Ausgänge  
 Einstellung des Systems  
 Dateien lesen und speichern  
 Meldungen für Benutzer  
 Umblättern auf die nächste Seite

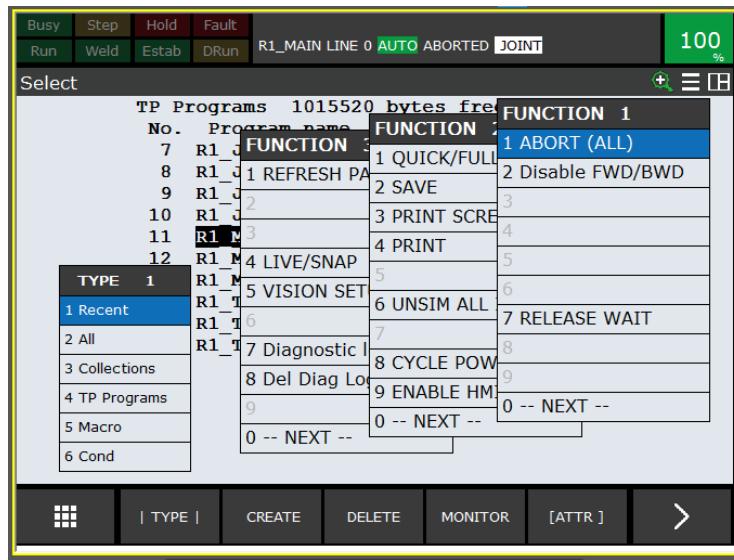


## Menüerweiterung auf Type Taste

## Menu 2

**SELECT**  
**EDIT**  
**DATA**  
**STATUS**  
**4D GRAPHICS**  
**SYSTEM**  
**USER2**  
**BROWSER**  
**NEXT**

wie Select Taste - Programmauswahl  
 Programmeditor  
 Programmdaten und Werte in Registern  
 Status des Systems wird angezeigt  
 Umschalten auf 4D Grafik  
 Anzeige der Ausgänge und simulierte Ein/Ausgänge  
 nicht belegt  
 Browser Links  
 Umblättern auf die nächste Seite



## Menüerklärung der FCTN Taste

### ABORT (ALL)

zwingt ein Programm das ausgeführt wird zum Abbruch  
Starten von Programmen vom Handbediengerät  
erlauben oder verbieten

### CHANGE GROUP

Jog feed Gruppe ändern. Nur angezeigt bei  
mehreren Gruppen.

### RELEASE WAIT

Warteanweisung wird übersprungen, dann Stopp

### QUICK/FULL MENUS

wechselt die Anzeige „FULL MENU“ und „QUICK MENUS“  
Daten des aktuellen Bildschirms sichern

### SAVE

Drucken des aktuellen Bildschirms

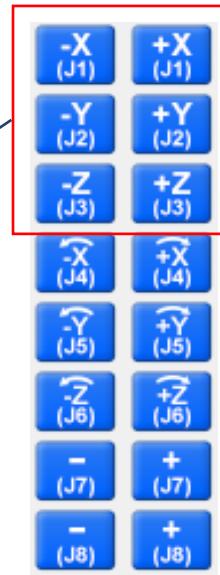
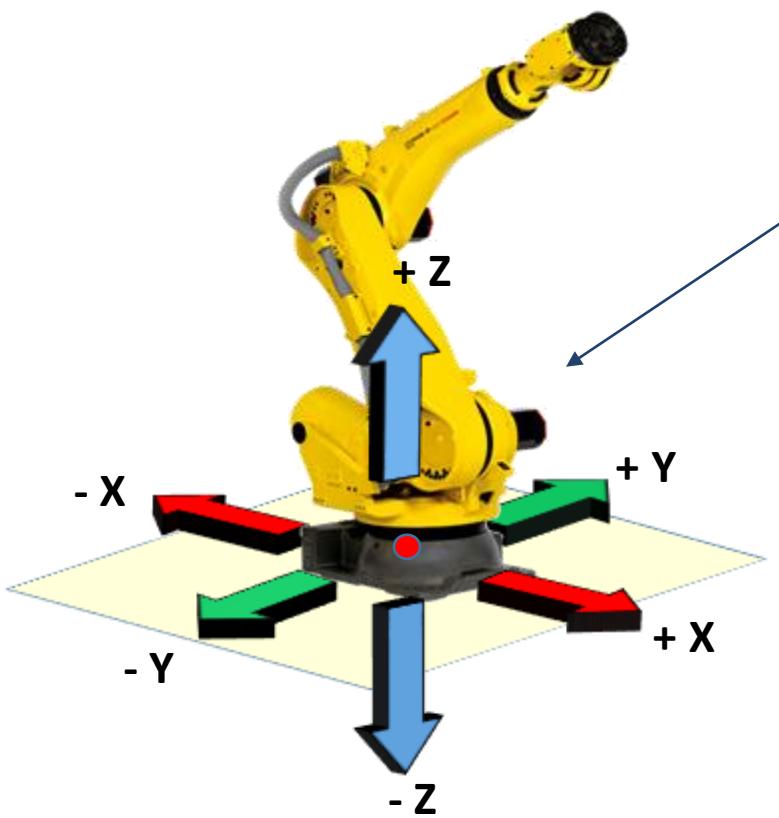
### PRINT SCREEN

Druckt alle Daten des aktuellen Bildschirms aus

### PRINT

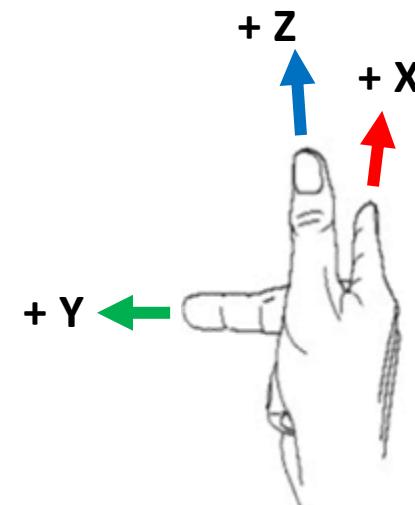
## **Frames - Koordinatensysteme**

## World Frame (Welt Koordinatensystem)



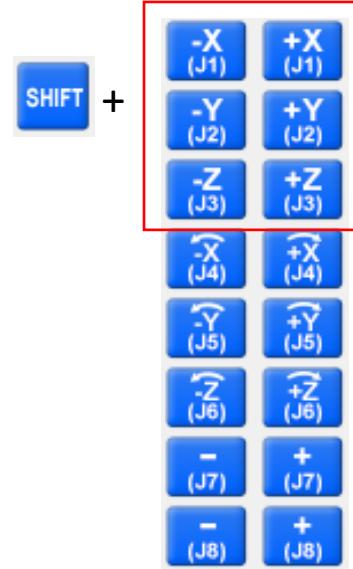
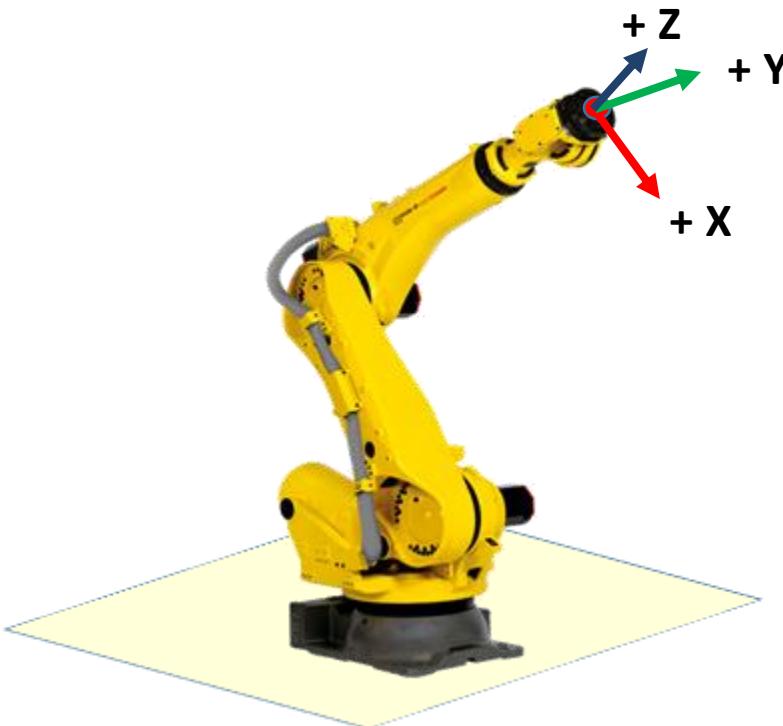
### WORLD Anwählen

1. COORD anwählen
2. WORLD auswählen
3. SHIFT Taste + Zustimmungstaste  
+ X / Y oder Z drücken um den Roboter in WORLD zu verfahren



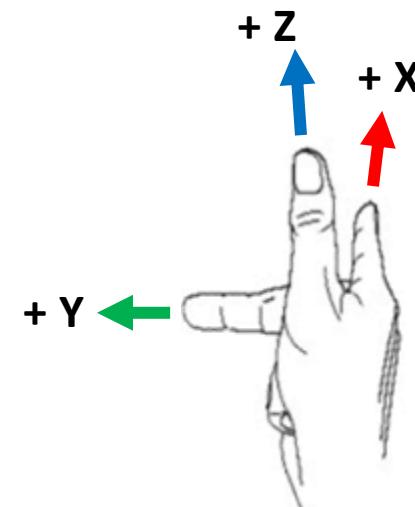
Rechte Hand Regel

## TOOL Frame (Werkzeug Koordinatensystem)



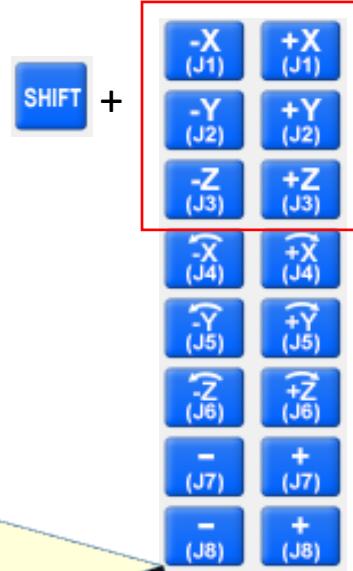
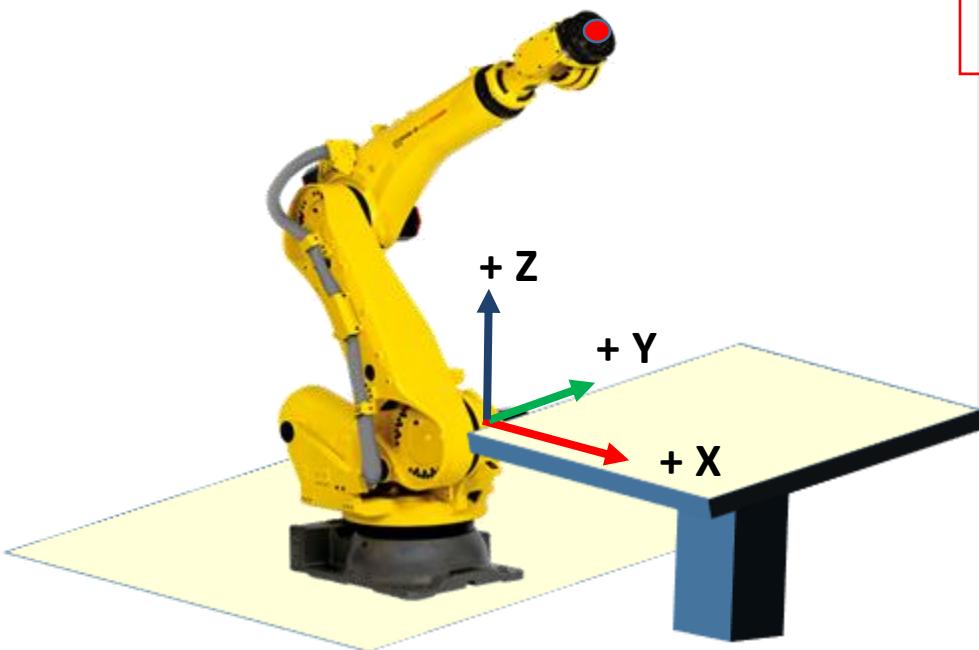
### TOOL Anwählen

1. COORD anwählen
2. TOOL auswählen
3. SHIFT Taste + Zustimmungstaste + X / Y oder Z drücken um den Roboter in TOOL zu verfahren



Rechte Hand Regel

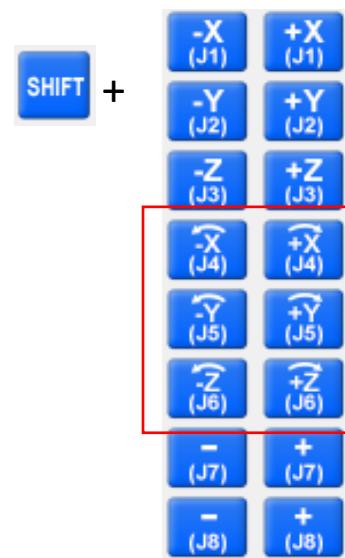
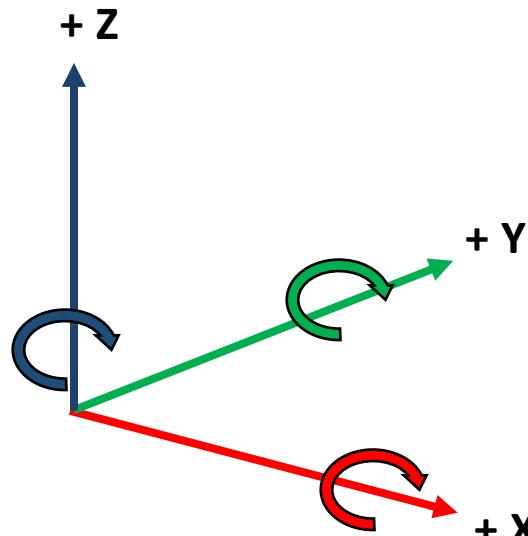
## USERFRAME Frame (Werkstück Koordinatensystem)



### USERFRAME Anwählen

1. COORD anwählen 
2. USER auswählen
3. SHIFT Taste + Zustimmungstaste + X / Y oder Z drücken um den Roboter in USER zu verfahren

Rotation um das Frame – gilt für jedes Koordinatensystem

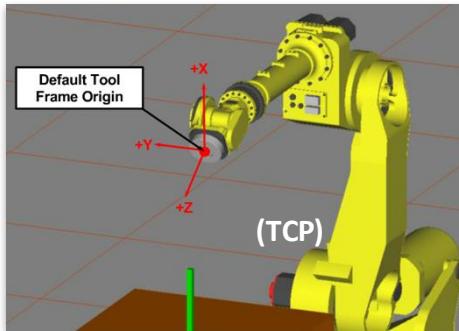


## USERFRAME Anwählen

1. COORD anwählen
2. FRAME auswählen
3. SHIFT Taste + Zustimmungstaste + J4 / J5 oder J6 drücken um den Roboter zu verfahren

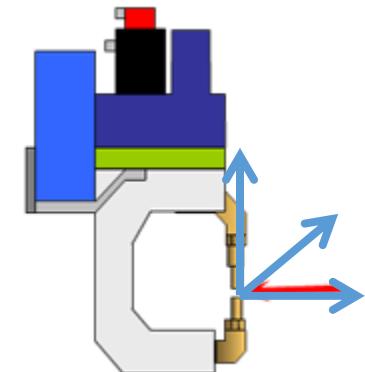
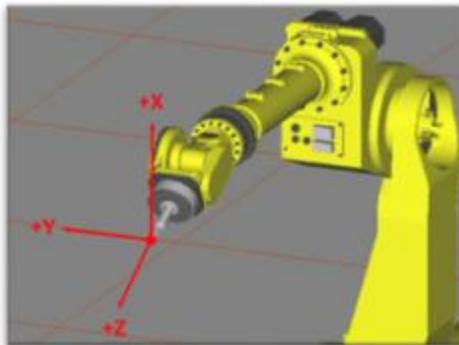


## TOOL Frame (Verschiebung des TCP)

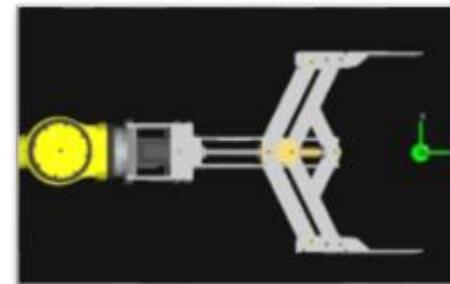
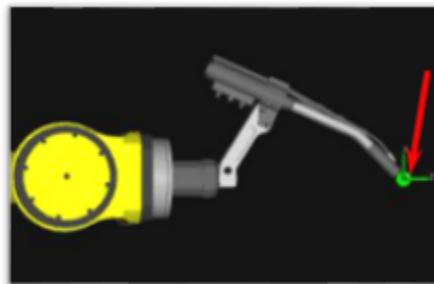


Der TCP (Tool Center Point) liegt bei Tool 0 in der Mitte der 6 Achse. Wird ein neues Tool vermessen, so wird der TCP an das Ende des Tools verschoben.

Warum wird das gemacht?  
Die Bewegung im Tool Koordinatensystem vereinfacht die Bewegung des Roboters beim Teachen von Bahnen. Für den Anwender ist die Stellung des Werkzeuges zum Werkstück wichtig.



Tool Frame Verschiebung



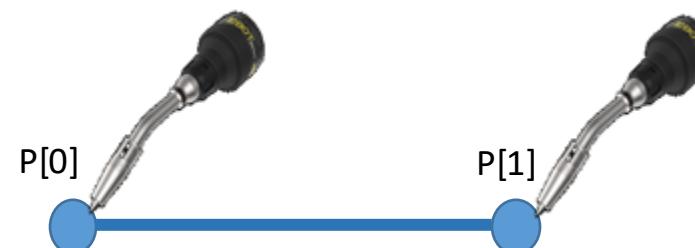
Tool Frame Beispiele

# Bewegungsbefehle

Joint = **J P[1] 100% FINE**



Linear = **L P[1] 2000 mm/sec FINE**

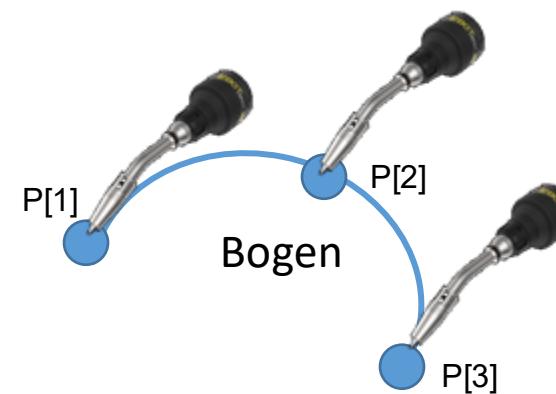


**Startpunkt**

Linear = **L P[1] 2000 mm/sec FINE**

**Bogen**

Zirkular = **C P[2] P[3] 2000 mm/sec FINE**



## Positionsbezeichnungen

Punktbezeichnung mit Ziffer und Name      L P[1] 2000 mm/sec FINE

Punktbezeichnung mit Positionsregister      PR[1 HOME]

## Roboter Positionsgeschwindigkeit

Linear =      L P[1] 2000 mm/sec FINE

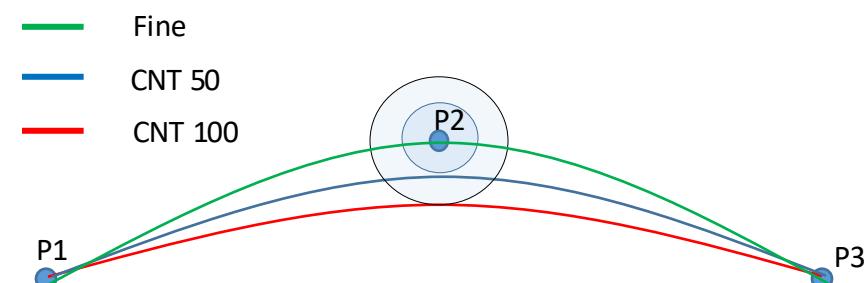
Joint =      J P[1] 100% CNT100

Zirkular =      C P[2] P[3] 2000 mm/sec FINE

## Roboter Positionsgenauigkeit

Linear =      L P[1] 2000 mm/sec FINE

Joint =      J P[1] 100% CNT100



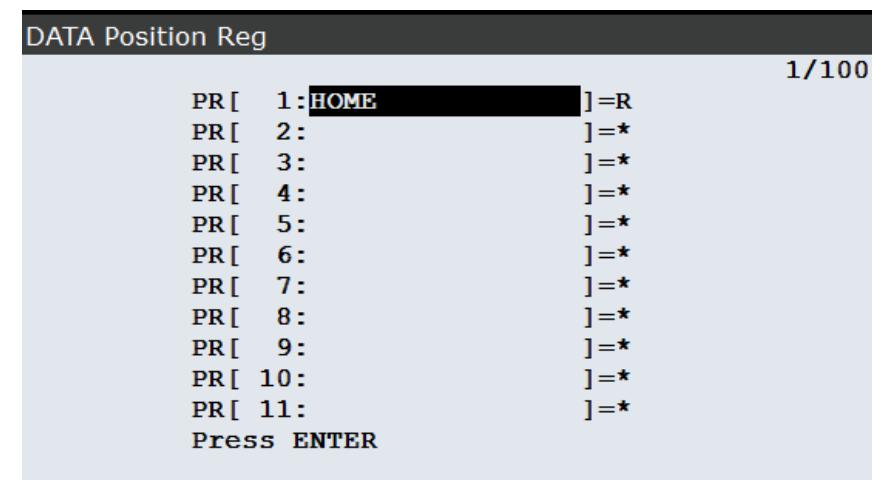
## Roboter Positionsregister

Die globalen Positionen werden in einem Positionsregister gespeichert.

Positionsregister können bei der Programmierung einfach eingesetzt werden aber nicht ausversehen umgeteacht.

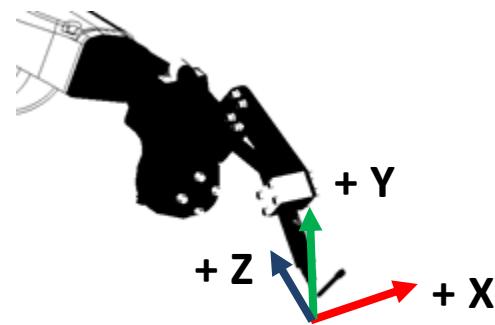
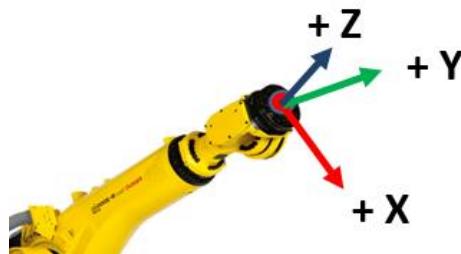


Anzeige: DATA -> TYPE -> 3 Positions Reg



Anzeige: Positionsregister 1 = HOME

## Werkzeug - Tool vermessen



## Werkzeug Vermessen

Damit der Roboter sich entlang eines kartesischen Koordinatensystems in Bezug auf das Tool bewegt, muss dieses Vermessen werden. Durch das Vermessen wird der TCP =Tool Center Point von der Mitte der 6 Achse an die Spitze des Tools verschoben. Damit der Roboter bei der Bedienung sich in Stoßrichtung des Tools bewegt wird die Orientierung des Tool Koordinatensystems festgelegt.

Die Werkzeugkoordinaten werden mit X, Y, Z bezeichnet. Die „0“ Position des Koordinatensystems befindet sich am Endpunkt des Tools.

### Dreipunktmethode

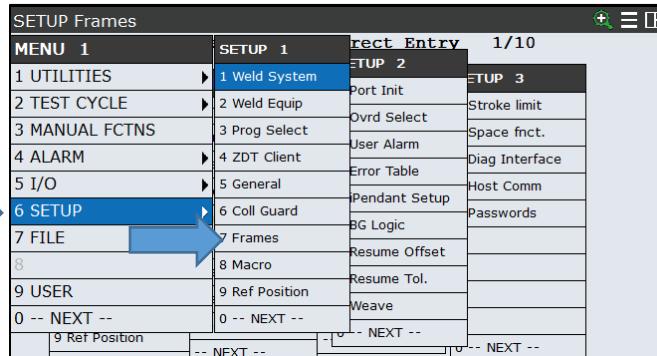
Durch diese Methode wird der Werkzeugmittelpunkt definiert. Dabei wird ein Punkt mit dem Werkzeug exakt aus drei Richtungen angefahren. Automatisch errechnet sich dann die TCP Verschiebung. Bei dieser Methode kann der TCP verschoben werden aber nicht die Orientierung des Tool Koordinatensystems.

### Sechspunktmethode

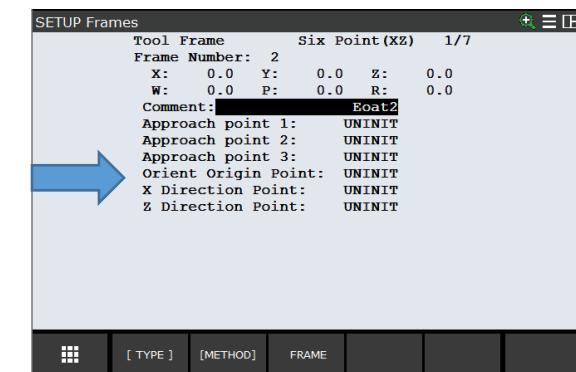
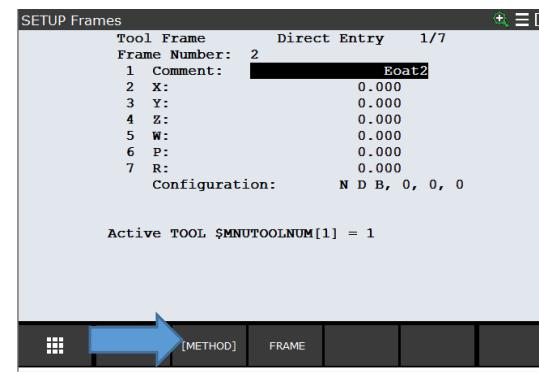
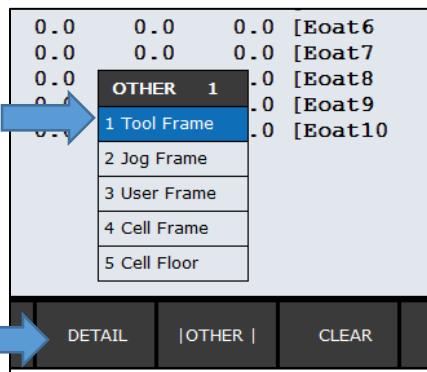
Bei dieser Methode wird erst der TCP festgelegt und dann die Orientierung des Werkzeuges.

## Werkzeug - Tool vermessen

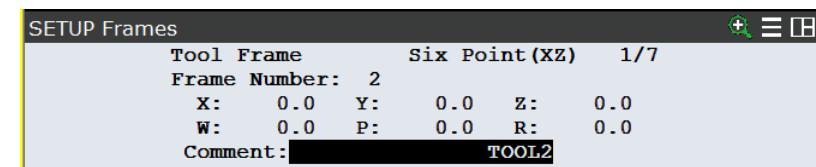
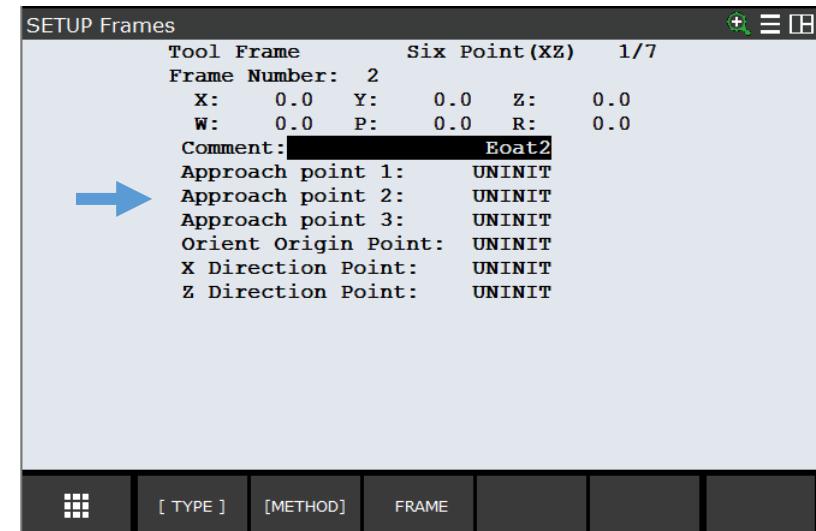
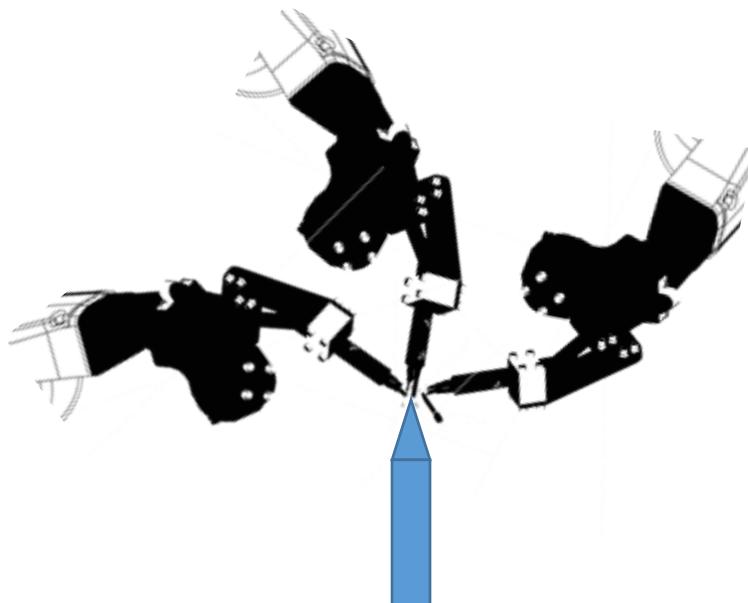
Anwahl: Menu -> Setup -> Frames -> Tool Frame -> DETAIL -> METHOD -> Six Point (XZ)



SETUP Frames				
Tool	Frame	/ Direct Entry	1/10	Comment
1	-17.1	.0	355.2	[tWeld ]
2	0.0	0.0	0.0	[Eoat2 ]
3	0.0	0.0	0.0	[Eoat3 ]
4	0.0	0.0	0.0	[Eoat4 ]
5	0.0	0.0	0.0	[Eoat5 ]
6	0.0	0.0	0.0	[Eoat6 ]
7	0.0	0.0	0.0	[Eoat7 ]
8	0.0	0.0	0.0	[Eoat8 ]
9	0.0	0.0	0.0	[Eoat9 ]
10	0.0	0.0	0.0	[Eoat10 ]



# Werkzeug - Tool vermessen



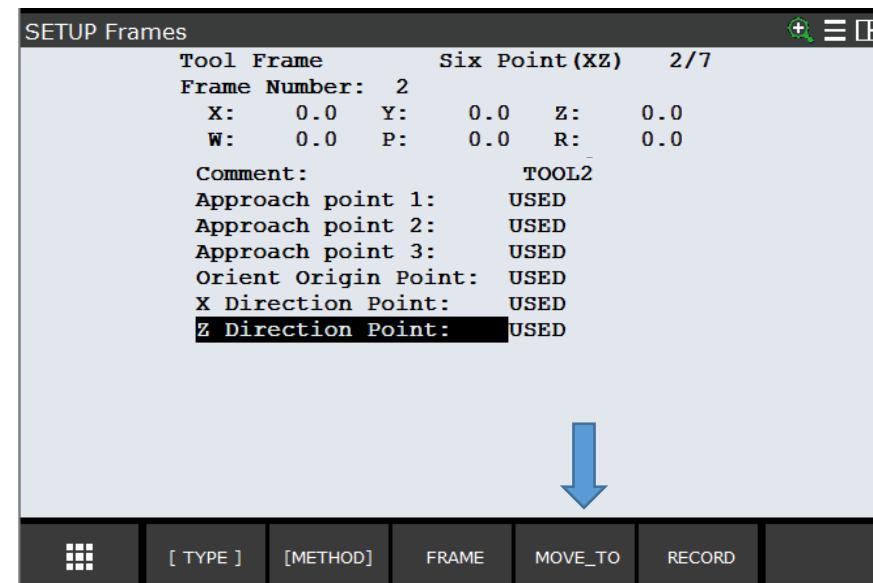
Tool wurde benannt mit = TOOL2

RECORD

Alle Positionen immer mit RECORD sichern!



- Die Einmeßpunkte können zu Kontrollzwecken angefahren werden. (MOVE\_TO)



[ TYPE ]

[METHOD]

FRAME

MOVE\_TO

RECORD

Menu -> SETUP -> FRAMES

SETUP Frames					
Tool	Frame	/ Direct Entry			2/10
		X	Y	Z	
1	-17.1	.0	355.2	[tWeld	]
2	12.0	0.0	100.0	[TOOL2	]
3	0.0	0.0	0.0	[Eoat3	]
4	0.0	0.0	0.0	[Eoat4	]
5	0.0	0.0	0.0	[Eoat5	]
6	0.0	0.0	0.0	[Eoat6	]
7	0.0	0.0	0.0	[Eoat7	]
8	0.0	0.0	0.0	[Eoat8	]
9	0.0	0.0	0.0	[Eoat9	]
10	0.0	0.0	0.0	[Eoat10	]

Cursor auf die gewünschte Werkzeugnummer setzen. (z.B. = 2)

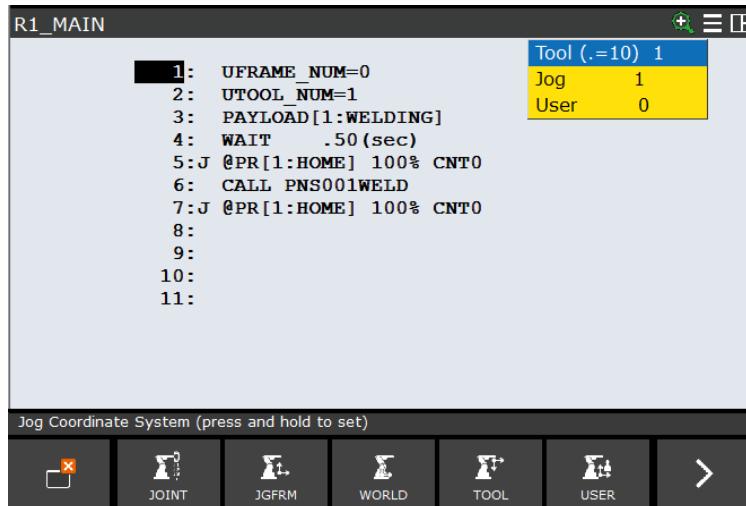
CLEAR drücken und mit YES bestätigen

(es werden die Koordinaten aber nicht der Kommentar gelöscht)

Im Anschluss kann der Kommentare durch Bestätigung mit „YES“ gelöscht werden.

# Werkzeug aktivieren

Drücke die SHIFT + COORD Taste



Das Jog Menü wird angezeigt

Jetzt können das Tool und damit die Werkzeugnummer eingegeben werden.  
Nach der Eingabe schließt das Fenster automatisch.

Tool (.=10) 2  
Jog 1  
User 0



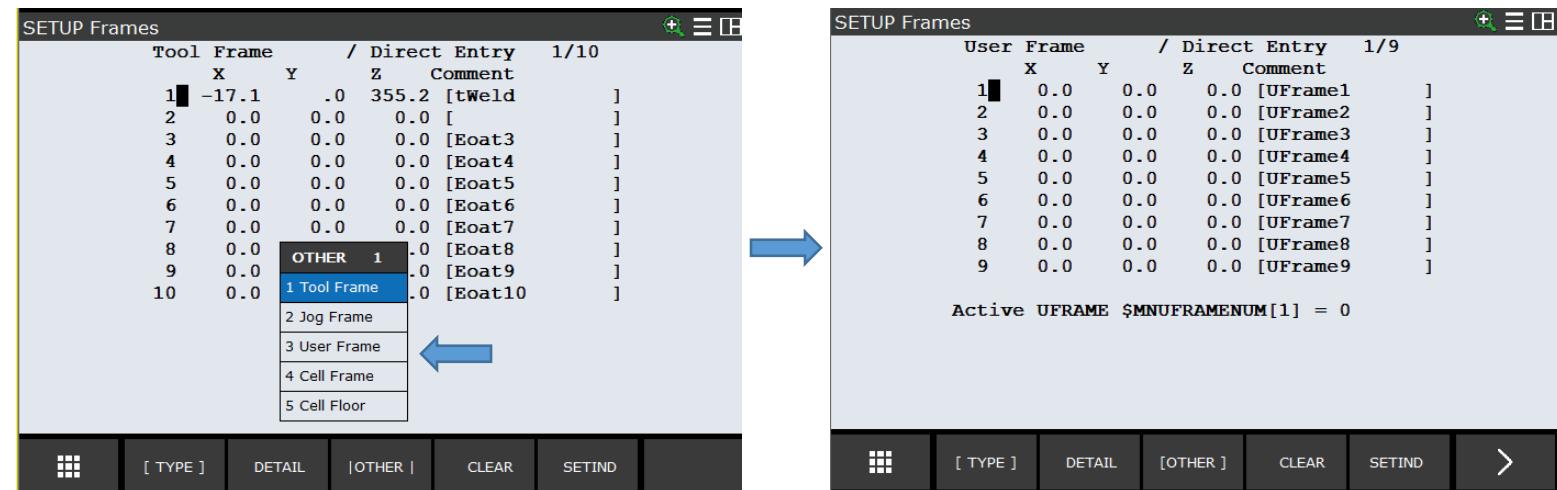
Tool (.=10) 2  
Jog 1  
User 0

Tool (.=10) 2  
Jog 1  
User 1

## Userframe = Werkstück (Arbeitsfläche)

Der Roboter benötigt für die Bewegung im Raum nicht nur das vermessene Tool sondern auch einen Bezug dazu. Der Bezug ist die Ebene an der der Roboter mit dem Tool arbeitet. Die Einstellung des Userframes ist deutlich einfacher als die des Tools. Das Userframe 0 ist die Ebene auf der der Roboters steht. Möchte man mehrere Flächen nutzen so ist das Vermessen von Userframes notwendig. Das Userframe muss vor dem Teachen im Programm immer eingestellt werden.  
Das Vermessen ist ähnlich der Dreipunktmethode beim Tool vermessen.

Menu -> Frames -> OTHER -> User Frame



SETUP Frames					
Tool	Frame	/	Direct	Entry	1/10
	X	Y	Z	Comment	
1	-17.1	.0	355.2	[tWeld	]
2	0.0	0.0	0.0	[	]
3	0.0	0.0	0.0	[Boat3	]
4	0.0	0.0	0.0	[Boat4	]
5	0.0	0.0	0.0	[Boat5	]
6	0.0	0.0	0.0	[Boat6	]
7	0.0	0.0	0.0	[Boat7	]
8	0.0	0.0	0.0	[Boat8	]
9	0.0	0.0	0.0	[Boat9	]
10	0.0	0.0	0.0	[Boat10	]

SETUP Frames					
User	Frame	/	Direct	Entry	1/9
	X	Y	Z	Comment	
1	0.0	0.0	0.0	[UFrame1	]
2	0.0	0.0	0.0	[UFrame2	]
3	0.0	0.0	0.0	[UFrame3	]
4	0.0	0.0	0.0	[UFrame4	]
5	0.0	0.0	0.0	[UFrame5	]
6	0.0	0.0	0.0	[UFrame6	]
7	0.0	0.0	0.0	[UFrame7	]
8	0.0	0.0	0.0	[UFrame8	]
9	0.0	0.0	0.0	[UFrame9	]

Active UFRAME \$MNUFRAMENUM[1] = 0

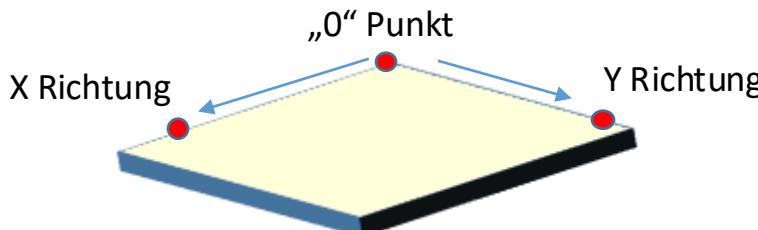
## Userframe = Werkstück (Arbeitsfläche)

SETUP Frames

User Frame	Direct Entry	1/7
Frame Number:	1	
1 Comment:	UFrame1	
2 X:	0.000	
3 Y:	0.000	
4 Z:	0.000	
5 W:	0.000	
6 P:	0.000	
7 R:	0.000	
Configuration:	N D B, 0, 0, 0	

Active UFRAME \$MNUFRAMENUM[1] = 0

[ TYPE ] [METHOD] FRAME MOVE\_TO RECORD



Comment: Benennung des Userframe

Methode zur Vermessung festlegen: Three Point

1. „0“ Punkt = Ursprung des Koordinatensystems festlegen
2. In X Richtung verfahren und Punkt festlegen
3. In Y Richtung verfahren und Punkt festlegen

SETUP Frames

User Frame	Three Point	1/4
Frame Number:	1	
X:	0.0	Y: 0.0
W:	0.0	Z: 0.0
P:	0.0	R: 0.0

Comment: UFrame1

Orient Origin Point: UNINIT

X Direction Point: UNINIT

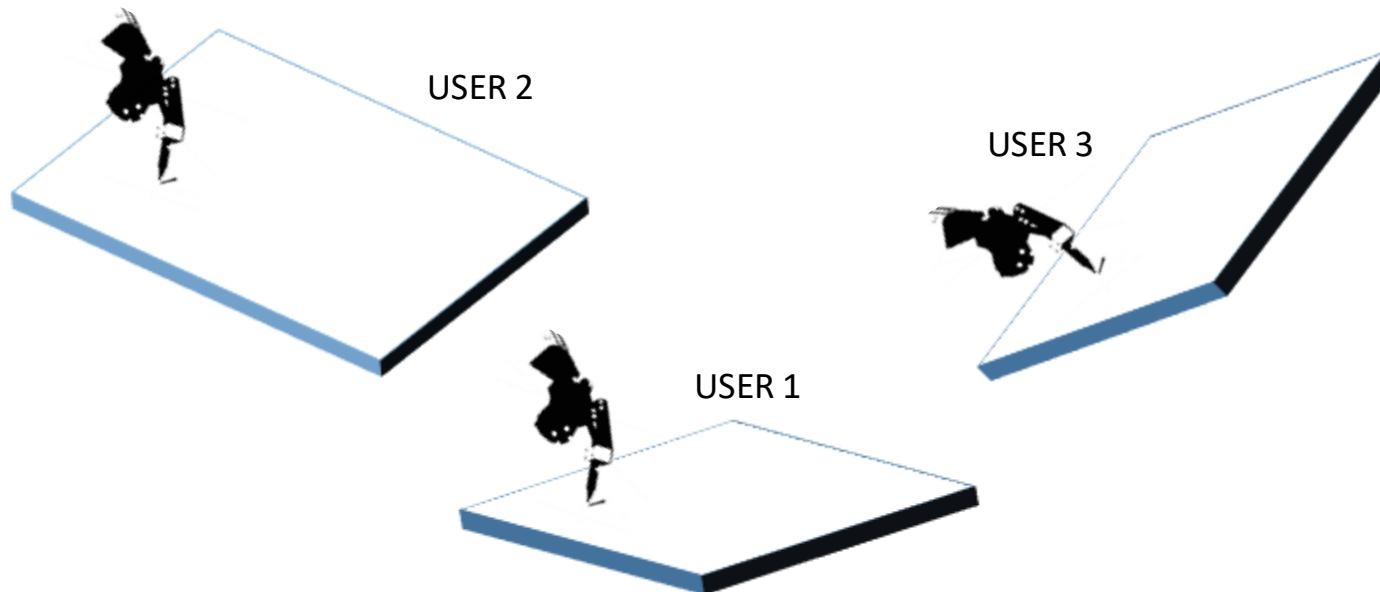
Y Direction Point: UNINIT

## Userframe = Arbeitsfläche einstellen

Jetzt können das Userframe und damit die Arbeitsfläche eingegeben werden.

Nach der Eingabe schließt das Fenster automatisch.

Tool (.=10) 2
Jog 1
User 1



## Userframe löschen

SETUP Frames					
User	Frame	/	Direct Entry	2/9	
	X	Y	Z	Comment	
1	0.0	0.0	0.0	[UFrame1]	
2	0.0	0.0	0.0	[UFrame2]	
3	0.0	0.0	0.0	[UFrame3]	
4	0.0	0.0	0.0	[UFrame4]	
5	0.0	0.0	0.0	[UFrame5]	
6	0.0	0.0	0.0	[UFrame6]	
7	0.0	0.0	0.0	[UFrame7]	
8	0.0	0.0	0.0	[UFrame8]	
9	0.0	0.0	0.0	[UFrame9]	

Active UFRAME \$MNUFRAMENUM[1] = 0

[ TYPE ]    DETAIL    [ OTHER ]    CLEAR    SETIND    >

Cursor auf das gewünschte Userframe setzen. (z.B. = 2)

CLEAR drücken und mit YES bestätigen

(es werden die Koordinaten aber nicht der Kommentar gelöscht)

Im Anschluss kann der Kommentare durch Bestätigung mit „YES“ gelöscht werden.

## Roboterprogramm erstellen

# Programm erstellen

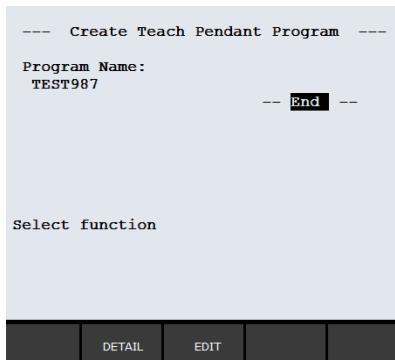
Damit Bewegungsabläufe mehrfach wiederholt werden können werden die Bewegungspunkte des Roboters in Programmen gespeichert.

## Hinweis:

**Zeichen** Für den Programmnamen dürfen Alphabetische Zeichen, Zahlen verwendet werden. Ein Programmname darf nicht mit einer Zahl beginnen.

**Symbole** Es dürfen nur \_ @ \* als Symbol verwendet werden.

*Programme die von der SPS aufgerufen werden sollen, müssen mit RSR oder PNS beginnen.*



Neus Programm erstellen:

1. SELECT Taste drücken
2. CREATE auswählen
3. Programmname eintragen
4. Enter

Programmname *TEST987* wurde eingegeben

5. Details

Ein Kommentar kann eingegeben werden „*TEST*“

PREV -> Weiter

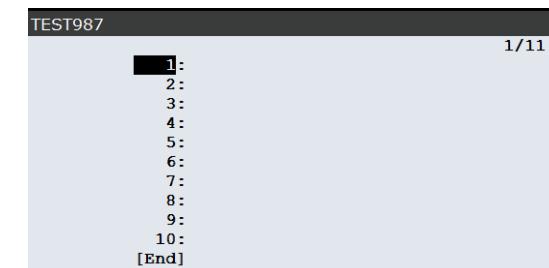
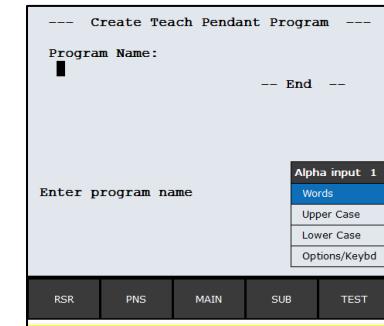
Einstellungen für Schreibschutz usw. können vorgenommen werden.

Diese Einstellungen können aber jederzeit geändert werden.

Mit **END** abschließen

6. **EDIT** wird automatisch aufgerufen
7. Mit **NEXT** Taste zu **EDCMD** wechseln
8. **Insert** -> Zeilen einfügen in das Programm
9. **ENTER** Taste drücken

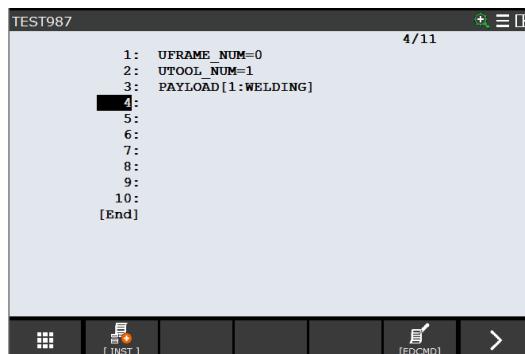
Anzahl der Zeilen eingeben z.B. **10** Stück <**Enter**>



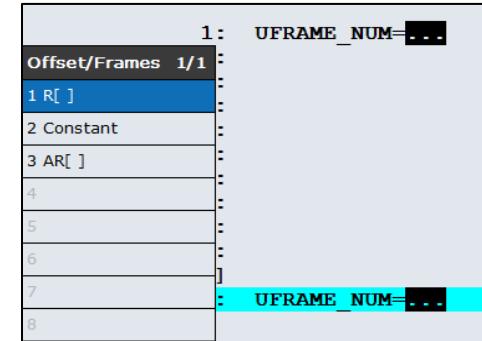
# Programm erstellen

Das Programm soll aus 5 Punkten bestehen.

Als erstes aber sind das **Userframe**, das **Utool** und **Payload** einzustellen.



- INST für Instruktionen anwählen
- Auf Instruktion 2 -> 4 wechseln
- Offset/Frames auswählen



- Constant = Konstante auswählen
- UFRAME\_NUM = 0 setzen <ENTER>
- Utool Frame auswählen
- CHOICE -> Constant wählen um Ziffer einzugeben
- 1 eingeben <ENTER>
- Payload auswählen <ENTER>
- 1 Eingeben - Gewicht für Werkzeug ist gesetzt

# Programm erstellen

TEST987 5/11

```
1: UFRAME_NUM=0
2: UTOOL_NUM=1
3: PAYLOAD [1:WELDING]
4:
5:
6:
7:
8:
9:
10:
[End]
```



POINT WELD\_ST WELD\_PT WELDEND TOUCHUP >

- Einen Punkt in Zeile 5 erstellen
- POINT + drücken

Select Motion 1/1
1 J P[] 100% FINE
2 J P[] 100% CNT100
3 L P[] 100mm/sec FINE
4 L P[] 100mm/sec CNT100

Motion Modify 1/1
1 P[ ]
2 PR[ ]
3
4
5
6
7
8 --next page--



- Auswählen welche Bewegungsart notwendig ist
  - Als 1. die Home Position erstellen
  - J PR[1:HOME] 100% CNT100
  - Die Home Position ist im Positionsregister 1 gespeichert
- Jetzt weitere Punkte einfügen

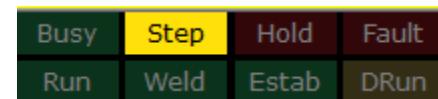
1: UFRAME_NUM=0
2: UTOOL_NUM=1
3: PAYLOAD [1:WELDING]
4:
5: J @PR[1:HOME] 100% CNT100
6: L @P[1] 100mm/sec CNT0
7: L @P[2] 100mm/sec FINE
8: L @P[3] 100mm/sec FINE
9: L @P[4] 100mm/sec FINE
10: [End]

- Als nächstes wieder Zeile 6 anwählen
- Roboter auf einen gewählten Punkt bewegen
- SHIFT Taste + TOCHUP drücken um Punkt zu speichern
- Weiter zur Zeile 7 mit Cursor Taste nach Unten
- Roboter zur gewünschten Position bewegen
- SHIFT Taste + TOCHUP drücken um Punkt zu speichern
- Usw.
- Wenn alle Punkte geteacht sind kann der Ablauf getestet werden

```
1: UFRAME_NUM=0
2: UTOOL_NUM=1
3: PAYLOAD[1:WELDING]
4:
5:J @PR[1:HOME] 100% CNT100
6:L @P[1] 100mm/sec CNT0
7:L @P[2] 100mm/sec FINE
8:L @P[3] 100mm/sec FINE
9:L @P[4] 100mm/sec FINE
10:
[End]
```

## Testen

- 1. Zeile mit Cursor Tasten anwählen
- STEP Betrieb anwählen -> STEP Taste drücken -> LED STEP leuchtet
- SHIFT Taste + Zustümmungsschalter + FWD Taste -> der Roboter bewegt sich zu von Zeile zu Zeile
- Dabei FWD Taste loslassen und erneut drücken

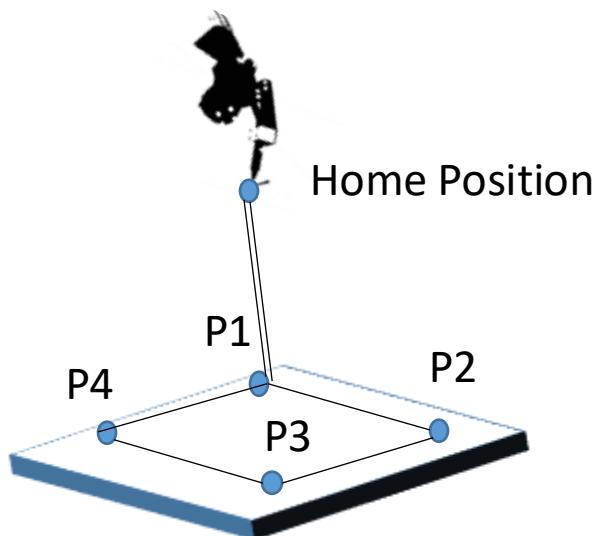


PAUSED

```
1: UFRAME_NUM=0
2: UTOOL_NUM=1
3: PAYLOAD[1:WELDING]
4:
5:J @PR[1:HOME] 100% CNT100
6:L @P[1] 100mm/sec CNT0
7:L @P[2] 100mm/sec FINE
8:L @P[3] 100mm/sec FINE
9:L @P[4] 100mm/sec FINE
10:
[End]
```

## Übung 7

- Werkzeugnummer und Userframe überprüfen -> ggf. die richtigen aktivieren
- Betriebsartenschalter auf T1?
- iPendant eingeschalten?
- Erstelle ein Programm mit Namen „RechteckXX“?
- Testen eines Programms mit Home Position, 5 Punkten (2 x P1) und zurück zur Home



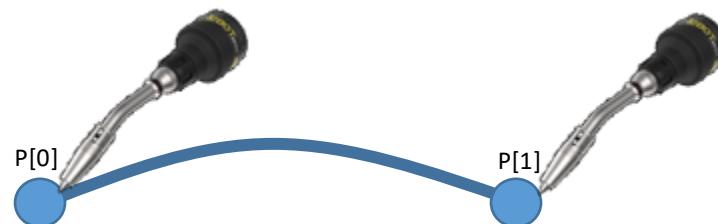
### RECHTECK1

1/12

```
1: UFRAME_NUM=1
2: UTOOL_NUM=2
3: PAYLOAD[1:Grp without part]
4:
5:J PR[1:Homepos] 100% CNT0
6:L P[1] 3000mm/sec CNT0
7:L P[2] 3000mm/sec CNT0
8:L P[3] 3000mm/sec CNT0
9:L P[4] 3000mm/sec CNT0
10:L P[1] 3000mm/sec CNT0
11:J PR[1:Homepos] 100% CNT0
```

## Übung

- Erstelle ein neues Programm Name: Bewegungen
- Setze das Tool, Userframe und Payload
- Erstelle zwei Joint Punkte
- Ändere die Geschwindigkeit und Überschleifung der Punkte
- Bewege jedes mal den Roboter zwischen den Punkten
- Was kannst du beobachten?



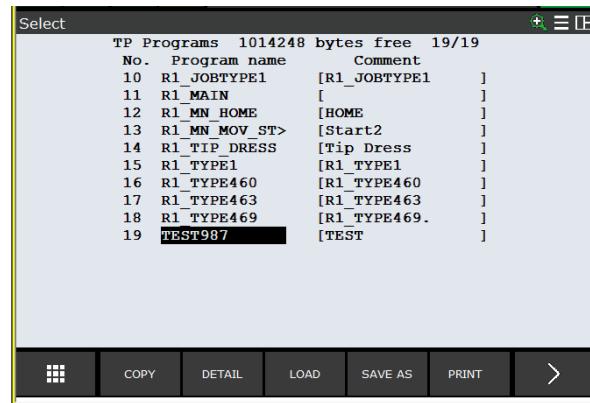
- Wiederhole die Übung mit zwei Linearen Bewegungen.
- Was kannst du beobachten?



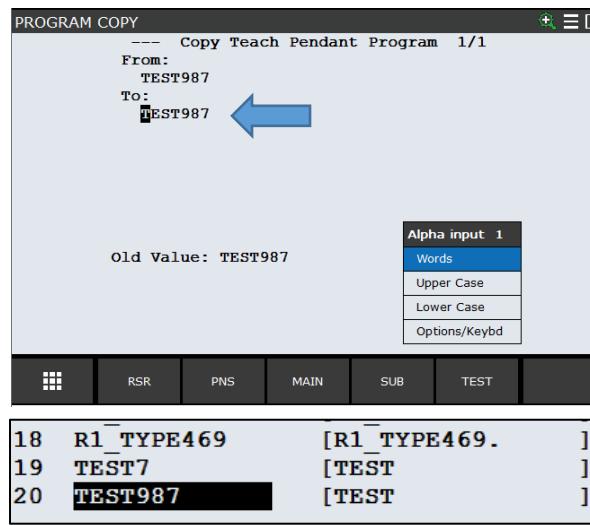
# Übung 3

## Übung

### Programm Kopieren – vorher Programm mit FCTN -> ABORT (ALL) abbrechen



COPY drücken



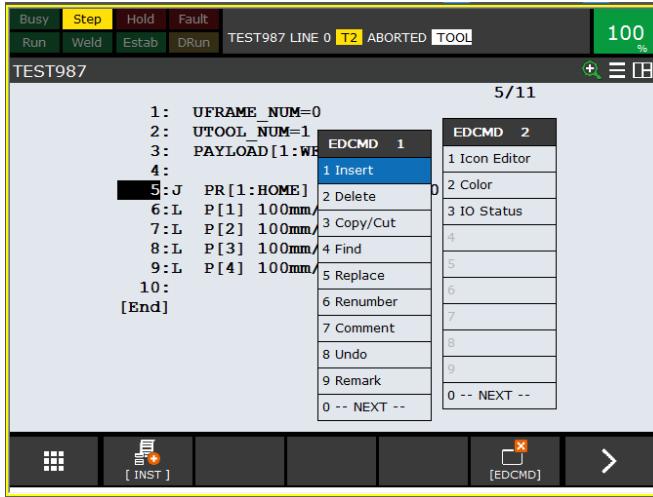
Neuen Programmnamen erstellen

To:  
**TEST7**  <Enter> Yes drücken



## EDCMD Funktion

Beim Programmieren ist es hilfreich zusätzliche Funktionen für das Kopieren, Einfügen, Löschen usw. nutzen zu können. Im Menü EDIT findet man diese Funktionen.



- **Insert** hiermit können Zeilen vor dem Satzzeiger eingefügt werden
- **Delete** die Zeile wird gelöscht
- **Copy/Cut** die Zeilen werden Kopiert oder ausgeschnitten.
- **Find** ein Befehl im Programm wird gesucht
- **Replace** ersetzen der einer Variablen im Programm
- **Renumber** die Positionsnummern werden neu durchnummieriert  
die Positionsdaten bleiben dabei erhalten
- **Comment** einfügen eines Kommentars
- **Undo** letzte Aktion rückgängig machen
- **Remark** auskommentieren von Programmzeilen
- **Icon Editor** umschalten in den Icon Editor

## Übung

Dupliziere dir ein beliebiges Programm und nutze die beschriebenen Funktionen.

Über solange, bis du die Funktionen optimal für die Erstellung deine Programme einsetzen kannst.

## Ein und Ausgänge

# Ein - und Ausgänge

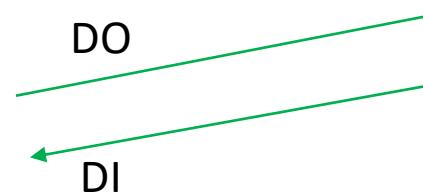
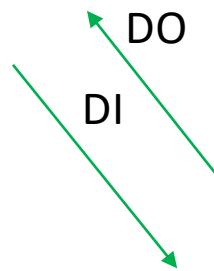
## Externe Steuerung



RO  
RI

Interne Signale  
Ventile, Sensoren, Greifer,  
Vakuumeinheiten

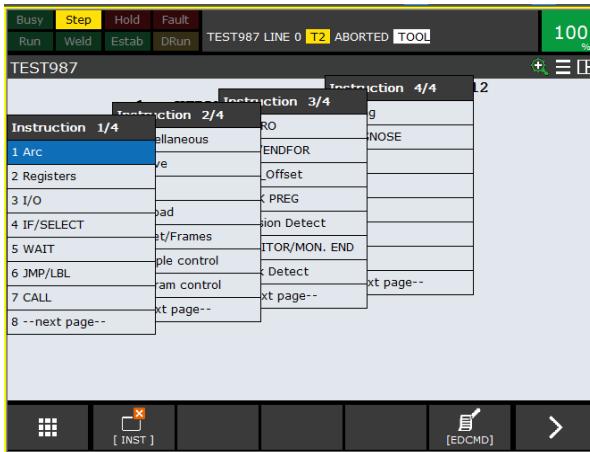
## Roboter Steuerung



DO = Digitale Ausgänge  
DI = Digitale Eingänge

# Ein - und Ausgänge

## Instruktionen zur Steuerung der Signale



INSTR → I/O drücken

```
PAUSED 10/16
3: PAYLOAD[1:WELDING]
4:
5:J PR[1:HOME] 100% CNT100
6:L P[1] 100mm/sec CNT0
7:L P[2] 100mm/sec FINE
8:L P[3] 100mm/sec FINE
9:L P[4] 100mm/sec FINE
10: DO[10:ON ]=ON
11: DO[100:OFF]=PULSE, 0.1sec
12:
13:
```

I/O statement 1/2		
1 DO[ ]=...		Steuert einen Ausgang
2 R[ ]=DI[ ]		Ein Register übernimmt den Wert des Eingangs
3 RO[ ]=...		Der Roboterausgang übernimmt den Wert
4 R[ ]=RI[ ]		Das Register übernimmt den Robotereingangswert
5 GO[ ]=...		Der Gruppenausgang erhält den Wert
6 R[ ]=GI[ ]		Dem Register wird der Wert des Gruppeneingangs übergeben
7 WO[ ]=...		
8 --next page--		

#	SIM	STATUS	[ ]
DO[ 1]	U	OFF	[ ]
DO[ 2]	U	OFF	[ ]
DO[ 3]	U	OFF	[ ]
DO[ 4]	U	OFF	[ ]
DO[ 5]	U	OFF	[ ]
DO[ 6]	U	OFF	[ ]
DO[ 7]	U	OFF	[ ]
DO[ 8]	U	OFF	[ ]
DO[ 9]	U	OFF	[ ]
DO[ 10]	U	ON	[ ]
DO[ 11]	U	OFF	[ ]

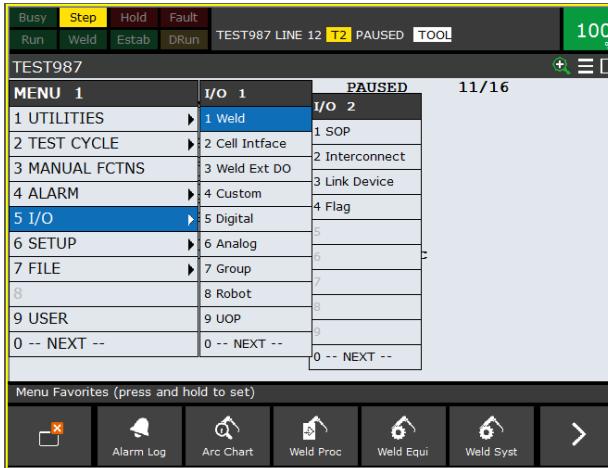
Sorted by port number.

In Zeile 10 des Programms wird der Ausgang 10 eingeschaltet. Zur Kontrolle werden die Ausgänge in diesem Bereich angezeigt. In Zeile 11 wird der Ausgang 100 für 0.1 Sekunde eingeschalten.

DO[100:OFF] zeigt den derzeitigen Zustand des Ausgangs = OFF = ausgeschaltet

# Ein - und Ausgänge

## Anzeige von Digitalen Signalen



Busy	Step	Hold	Fault	TEST987 LINE 12	T2 PAUSED	TOOL	100 %			
Run	Weld	Estab	DRun	I/O Digital Out						
#	SIM	STATUS		103/512						
DO[ 93]	U	OFF	[	]						
DO[ 94]	U	OFF	[	]						
DO[ 95]	U	OFF	[	]						
DO[ 96]	U	OFF	[	]						
DO[ 97]	U	OFF	[	]						
DO[ 98]	U	OFF	[	]						
DO[ 99]	U	OFF	[	]						
DO[ 100]	U	OFF	[	]						
DO[ 101]	U	OFF	[	]						
DO[ 102]	U	OFF	[	]						
DO[ 103]	U	OFF	[	]						

Sorted by port number.

Mit den Cursortasten kann man sich zwischen den Signalen bewegen. Durch zusätzlichen Drücken der Shift Taste ist ein Seitensprung möglich.

TYPE	1
1 Weld	
2 Cell Intface	
3 Weld Ext DO	
4 Custom	
5 Digital	
6 Analog	
7 Group	
8 Robot	
9 UOP	
0 -- NEXT --	

Mit **TYPE** kann man zwischen den Signalarten wechseln.

**Config** zeigt die Konfiguration der Signale an.

**IN/OUT** Wechsel zwischen der Anzeige der Ein- und Ausgängen

**ON** setzt einen Ausgang

**OFF** setzt den Ausgang wieder zurück

**Next** ein weiteres Menü wird aufgerufen

**DETAIL** Anzeige von Zusätzlichen Informationen oder **Kommentar** einfügen

Port Detail	1/3
Digital Output	[ 103]
1 Comment: [TEST]	
2 Polarity: NORMAL	
3 Complementary: FALSE [ 103 - 104]	



Port Detail	1/3
Digital Output	[ 103]
1 Comment: [TEST]	
2 Polarity: NORMAL	
3 Complementary: FALSE [ 103 - 104]	

# Ein - und Ausgänge

## Simulieren von Digitalen Signalen

Busy	Step	Hold	Fault	Run	Weld	Estab	DRun	TEST987 LINE 12	T2 PAUSED	TOOL	100 %
I/O Digital Out											
# SIM STATUS											
103/512											
DO[ 93] U OFF [ ]											
DO[ 94] U OFF [ ]											
DO[ 95] U OFF [ ]											
DO[ 96] U OFF [ ]											
DO[ 97] U OFF [ ]											
DO[ 98] U OFF [ ]											
DO[ 99] U OFF [ ]											
DO[ 100] U OFF [ ]											
DO[ 101] U OFF [ ]											
DO[ 102] U OFF [ ]											
DO[ 103] U OFF [ ]											
Sorted by port number.											
<input type="button" value="["/> <input type="button" value="TYPE"/> <input type="button" value="CONFIG"/> <input type="button" value="IN/OUT"/> <input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="&gt;"/>											

Der zu Simulierende Ausgang wird mit den Cursor Tasten ausgewählt. Mit der Next Taste wird SIMULATE angezeigt. Durch drücken der Taste kann der Ausgang oder Eingang simuliert werden. Das heißt, der Bediener kann bestimmte gewünschte Signalzustände für Tests einstellen. Simulierte Signale sind gelb hinterlegt. Durch UNSIM kann die Simulation zurückgesetzt werden.

Busy	Step	Hold	Fault	Run	Weld	Estab	DRun	TEST987 LINE 12	T2 PAUSED	TOOL	100 %
I/O Digital Out											
# SIM STATUS											
103/512											
DO[ 98] U OFF [ ]											
DO[ 99] U OFF [ ]											
DO[ 100] U OFF [ ]											
DO[ 101] U OFF [ ]											
DO[ 102] U OFF [ ]											
DO[ 103] U OFF [ TEST ]											
DO[ 104] U OFF [ ]											
DO[ 105] U OFF [ ]											
DO[ 106] U OFF [ ]											
DO[ 107] U OFF [ ]											
DO[ 108] U OFF [ ]											
<input type="button" value="["/> <input type="button" value="TYPE"/> <input type="button" value="CONFIG"/> <input type="button" value="IN/OUT"/> <input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="&gt;"/>											

Busy	Step	Hold	Fault	Run	Weld	Estab	DRun	TEST987 LINE 12	T2 PAUSED	TOOL	100 %
I/O Digital In											
# SIM STATUS											
2/512											
DI[ 1] S OFF [ ]											
DI[ 2] U OFF [ ]											
DI[ 3] U OFF [ ]											
DI[ 4] U OFF [ ]											
DI[ 5] U OFF [ ]											
DI[ 6] U OFF [ ]											
DI[ 7] U OFF [ ]											
DI[ 8] U OFF [ ]											
DI[ 9] U OFF [ ]											
DI[ 10] U OFF [ ]											
DI[ 11] U OFF [ ]											
<input type="button" value="["/> <input type="button" value="TYPE"/> <input type="button" value="CONFIG"/> <input type="button" value="IN/OUT"/> <input type="button" value="ON"/> <input type="button" value="OFF"/> <input type="button" value="&gt;"/>											

Der zu Simulierende Ausgang wird mit den Cursor Tasten ausgewählt. Mit der Next Taste wird SIMULATE angezeigt. Durch drücken der Taste kann der Ausgang oder Eingang simuliert werden. Das heißt, der Bediener kann bestimmte gewünschte Signalzustände für Tests einstellen. Simulierte Signale sind gelb hinterlegt. Durch UNSIM kann die Simulation zurückgesetzt werden.

## Ein - und Ausgänge

## Digitale Gruppen Ein- und Ausgänge

Busy	Step	Hold	Fault	TEST987 LINE 12	T2 PAUSED	TOOL	100%
Run	Weld	Estab	dRun				
I/O Group Out							
	#	SIM	VALUE		1/300		
GO[	1]	*	*	[PROGRAM_NR	]		
GO[	2]	*	*	]			
GO[	3]	*	*	[	]		
GO[	4]	*	*	]			
GO[	5]	*	*	[	]		
GO[	6]	*	*	]			
GO[	7]	*	*	[	]		
GO[	8]	*	*	]			
GO[	9]	*	*	[	]		
GO[	10]	*	*	]			
GO[	11]	*	*	[	]		
Port is not defined.							
	[ TYPE ]	NUM-SRT	CMT-SRT	DETAIL		HELP	>

Digitale Ein- und Ausgänge können nur zwei Zustände abbilden. Um komplexere Daten zu übermitteln können Signale zu Gruppen zusammengefasst werden.

Jedes Signal kann dabei einen festen Wert in der Darstellung übernehmen. Die Gruppe 1 soll von der SPS die Programmnummer an den Roboter übermitteln. Hierzu wurden verschieden Eingänge zu dieser Gruppe zusammengefasst. (DI10-15)

Der digitale Zustand wird binär dargestellt **2<sup>X</sup>**

DI	10	11	12	13	14	15	
Wert	1	2	4	8	16	32	
on/off	1	0	1	0	0	0	
Wert	1 + 0 + 4 + 0 + 0 + 0 = 5						

# Ein - und Ausgänge



I/O Taste drücken

## Verschiedene Signale

TYPE	RO[1]	TYPE	2	SIM	STATUS
1 Cell Interface		Link Device			[Greifer ZU]
2 Custom		Flag			[Greifer AUF]
3 Digital					[ ]
4 Analog					[ ]
5 Group					[ ]
6 Robot					[ ]
7 UOP					[ ]
8 SOP					[ ]
9 Interconnect					[ ]
0 -- NEXT --					[ ]

## Beispiele für digitale Ein- und Ausgänge

### Roboter Eingänge

RI[3:Offen]

RI[4:Geschlossen]

### Roboter Ausgänge

RO[1:Greifer ZU]

RO[2:Greifer AUF]

#	SIM	STATUS	1/8
RI[ 1]	U	OFF	[ ]
RI[ 2]	U	OFF	[ ]
RI[ 3]	U	OFF	[Offen ]
RI[ 4]	U	OFF	[Geschlossen ]
RI[ 5]	U	OFF	[ ]
RI[ 6]	U	OFF	[ ]
RI[ 7]	U	OFF	[ ]
RI[ 8]	U	OFF	[ ]

Sorted by port number.

### Programmausschnitte

#### OPEN

```
1: IF RI[4:Geschlossen]=OFF,JMP LBL[1] ;  
2: RO[1:Greifer ZU]=OFF ;  
3: LBL[1] ;
```

#### CLOSE

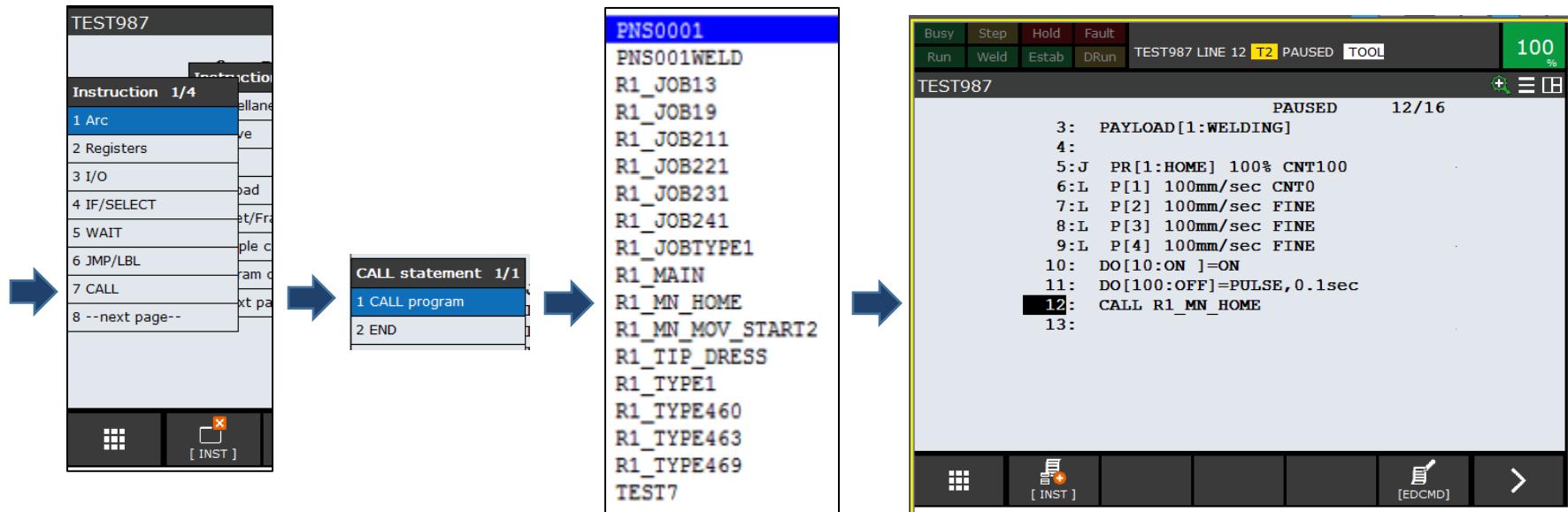
```
1: RO[1:Greifer ZU]=ON ;
```

# Unterprogramme aufrufen

## Aufruf von Programmen

Durch CALL ist es möglich verschiedene Programme aus einem Hauptprogramm aufzurufen.

INST -> 7 CALL

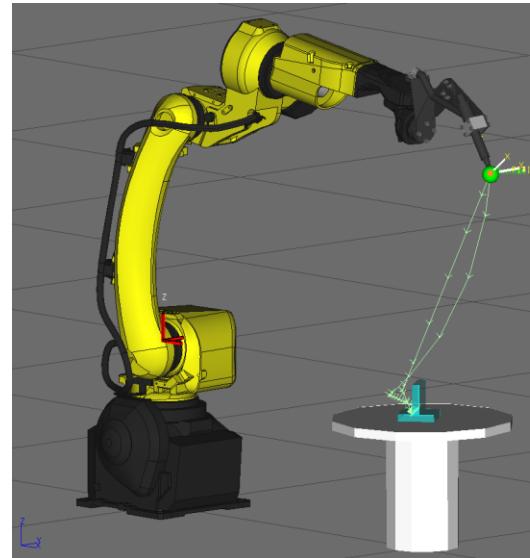


Durch CALL wird das „R1\_MN\_HOME“ Programm aufgerufen.

## Erstellung eines Programms das eine Schweißnaht zieht

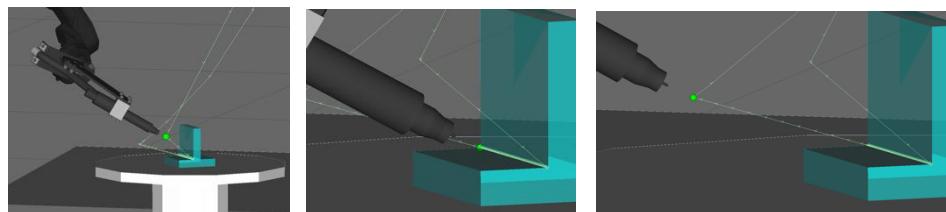
1. Erstelle ein neues Programm mit dem Namen: R1WELD
2. SELECT -> CREATE -> NAMEN eingeben <ENTER>
3. als Erstes: UTOOL, UFRAME, PAYLOAD eintragen
4. Der Startpunkt soll die HOME Position sein
5. P[2] - Anfahren des Werkstücks
6. P[3] - Start Position zum Schweißen
7. P[4] - Ende Schweißen
8. P[5] – Wegfahren
9. Home

```
7:J PR[1:HOME] 100% FINE
8:L P[2:OverWeld] 1500mm/sec CNT0
:
9:L P[3] 800mm/sec FINE
: Weld Start[1,1]
10:L P[5] WELD_SPEED FINE
: Weld End[1,1]
11:L P[6] 100mm/sec FINE
12:J PR[1:HOME] 100% FINE
```



MENU 2	DATA 1
1 SELECT	1 Weld Procedure
2 EDIT	2 Registers
3 DATA	3 Position Reg
4 STATUS	4 String Reg
5 4D GRAPHICS	5 KAREL Vars
6 SYSTEM	6 KAREL Posns
7 USER2	7 Weave Sched
8 BROWSER	Weld Start[1,1]
9	
0 -- NEXT --	

Kleine  
Hilfe



## Was ist ein Register?

Ein Register ist zur Speicherung von Zahlen gedacht. Anhand von Nummern können 200 Register Zahlen speichern. Die Adressierung kann dabei direkt als auch indirekt erfolgen. Der Datentyp ist ein Doppelwort (32Bit) und Register sind Global angelegt.

### Register direkt ansprechen

R[11] = 5 Durch die Nummer 11 wird das Register 11 angesprochen. Mit der Zuweisung der Ziffer 5 wird im Register die Zahl 5 gespeichert. Wird anschließend R[11] = 15 aufgerufen, so wird der alte Wert 5 durch den neuern Wert 15 überschrieben.

### Register indirekt ansprechen

R[R[4]] = 5 Es wird nicht direkt das Register mit der Nummer aufgerufen. Im Register 4 steht der Wert 11. Dieser wird jetzt als Registernummer eingesetzt. Das Register 11 erhält den Wert 5 übergeben.

```
R[ 1: ]=0
R[ 2: ]=0
R[ 3: ]=0
R[ 4: ]=11
R[ 5: ]=0
R[ 6: ]=0
R[ 7: ]=0
R[ 8: ]=0
R[ 9: ]=0
R[ 10: ]=0
R[ 11:TEST ]=5

R[ 11:TEST ]=5

R[ 4: ]=11
```

```
9:L P[3] 800mm/sec FINE
: Weld Start[1,1]
10:L P[5] WELD_SPEED FINE
: Weld End[1,1]
11:L P[6] 100mm/sec FINE
12:J @PR[1:HOME] 100% FINE
13: R[4]=11
14: R[11:TEST]=5
15: R[R[4]]=5
16:
[End]
Enter value
```

REGISTER statement 1/1

1 R[ ]
2 PR[ ]
3 PR[i,j]
4 SR[ ]
5
6
7
8 --next page--

123 : DIRECT : INDIRECT : [CHOICE] : [LIST]

Positionsregister können wie Zahlen eingesetzt werden!

REGISTER statement 1/1
1 ...=...
2 ...=...+...
3 ...=...-...
4 ...=...*...
5 ...=.../...
6 ...=...DIV...
7 ...=...MOD...
8 ...=(...)

## Bedingte Verzweigung (if –then- else )

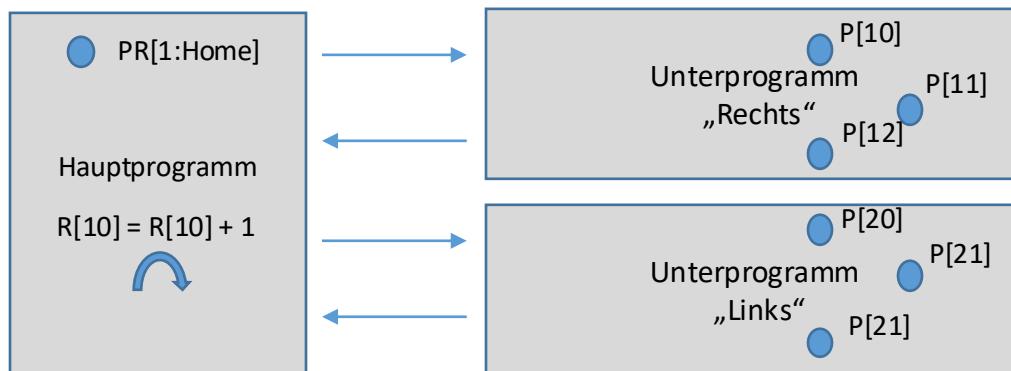
Um die Arbeit mit Registern zu vertiefen werden wir die Bedingte Verzweigung (IF) mit Registern umsetzen.

Als erstes wird ein Hauptprogramm benötigt das eine Bedingte Verzweigung enthält. Aufgabe ist es dieses Hauptprogramm und zwei weitere Unterprogramme zu erstellen.

Das Hauptprogramm hat die Registeranweisung  $R[10] = R[10] + 1$  mit der die Zyklen gezählt werden.

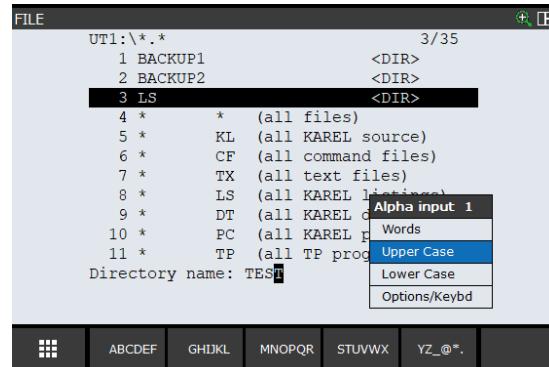
Das Hauptprogramm soll 3 Zyklen durchlaufen bevor es wieder abbricht. Der Zyklus beinhaltet den Aufruf eines Unterprogramms „Recht“ mit 3 Punkten und ein Unterprogramm „Links“ mit 3 Punkten. Der Roboter wechselt immer von Links nach Rechts.

Sind 3 Zyklen durchlaufen hält der Roboter an.



# Programme sichern

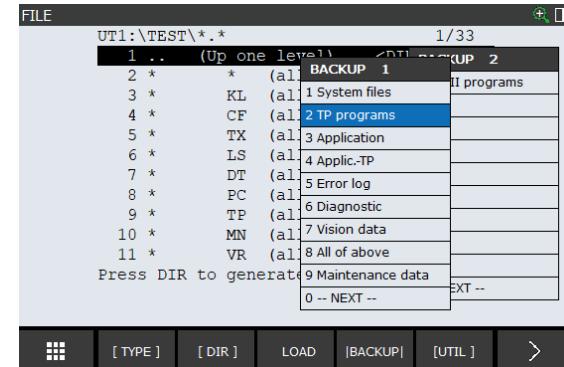
## Verzeichnis „Test“ erstellen



```
FILE UT1:\*.* 3/35
1 BACKUP1 <DIR>
2 BACKUP2 <DIR>
3 LS <DIR>
4 * * (all files)
5 * KL (all KAREL source)
6 * CF (all command files)
7 * TX (all text files)
8 * LS (all KAREL listings)
9 * DT (all KAREL P Words
10 * PC (all KAREL P Words
11 * TP (all TP progs)
Directory name: TEST
ABCDEF GHIJKL MNOPQR STUVWX YZ_@*. 
```

Screenshot of a Fanuc Robo Trainer terminal window. The command '3 LS' is being entered, and a context menu is open with options: Alpha input 1, Upper Case, Lower Case, and Options/Keybd. The status bar shows 'ABCDEF GHIJKL MNOPQR STUVWX YZ\_@\*.'.

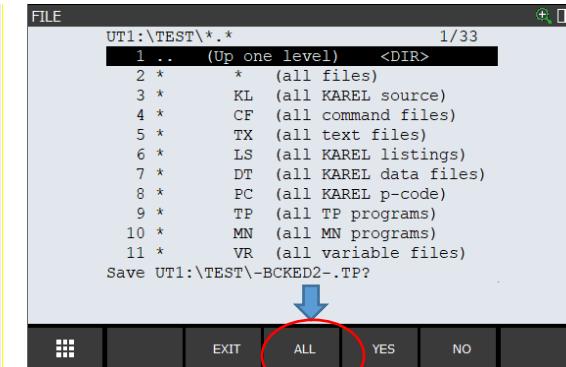
## Sichern aller „TP“ Programme



```
FILE UT1:\TEST\*.* 1/33
1 .. (Up one level) <DIR> UP 2
2 * * (all files)
3 * KL (all KAREL source)
4 * CF (all command files)
5 * TX (all text files)
6 * LS (all KAREL listings)
7 * DT (all KAREL data files)
8 * PC (all KAREL p-code)
9 * TP (all TP programs)
10 * MN (all MN programs)
11 * VR (all variable files)
Press DIR to generate EXT -- 
```

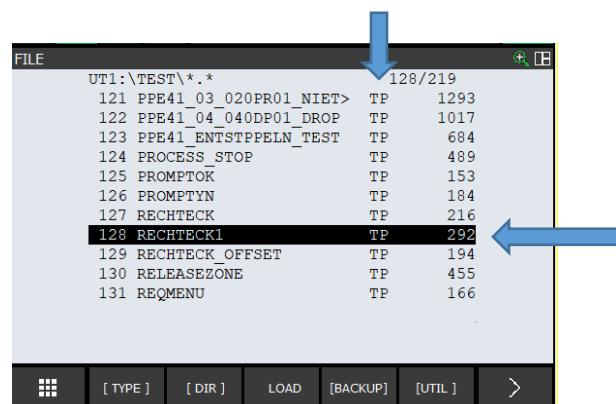
Screenshot of a Fanuc Robo Trainer terminal window showing a file list. The 'TP' program is selected in the list. The status bar shows '[ TYPE ] [ DIR ] LOAD [BACKUP] [UTIL ] >'. A blue arrow points from the 'TP' entry in the list to the 'TP' entry in the bottom status bar.

## „ALL“ auswählen



```
FILE UT1:\TEST\*.* 1/33
1 .. (Up one level) <DIR>
2 * * (all files)
3 * KL (all KAREL source)
4 * CF (all command files)
5 * TX (all text files)
6 * LS (all KAREL listings)
7 * DT (all KAREL data files)
8 * PC (all KAREL p-code)
9 * TP (all TP programs)
10 * MN (all MN programs)
11 * VR (all variable files)
Save UT1:\TEST\~BACKED2~.TP?
ABCDEF GHIJKL MNOPQR STUVWX YZ_@*. 
```

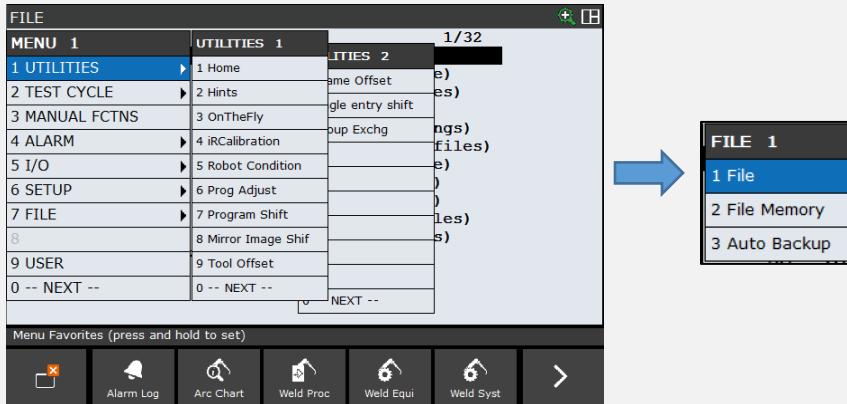
Screenshot of a Fanuc Robo Trainer terminal window showing a file list. The 'ALL' option is highlighted in the bottom status bar. A blue arrow points from the 'ALL' entry in the status bar to the 'ALL' entry in the bottom status bar. A red circle highlights the 'ALL' button in the status bar.



```
FILE UT1:\TEST\*.* 128/219
121 PFE41_03_020PR01_NIET> TP 1293
122 PFE41_04_040DP01_DROP TP 1017
123 PFE41_ENTSTPPELN_TEST TP 684
124 PROCESS_STOP TP 489
125 PROMPTOK TP 153
126 PROMPTYN TP 184
127 RECHTECK TP 216
128 RECHTECK1 TP 292
129 RECHTECK_OFFSET TP 194
130 RELEASEZONE TP 455
131 REQMENU TP 166 
```

Screenshot of a Fanuc Robo Trainer terminal window showing a file list. The status bar shows '[ TYPE ] [ DIR ] LOAD [BACKUP] [UTIL ] >'. A blue arrow points from the 'BACKUP' entry in the status bar to the 'BACKUP' entry in the bottom status bar.

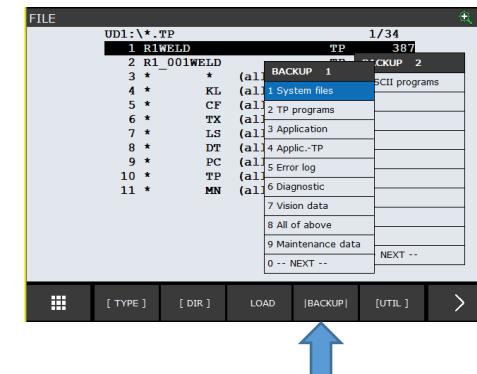
## Menu -> 7 FILE



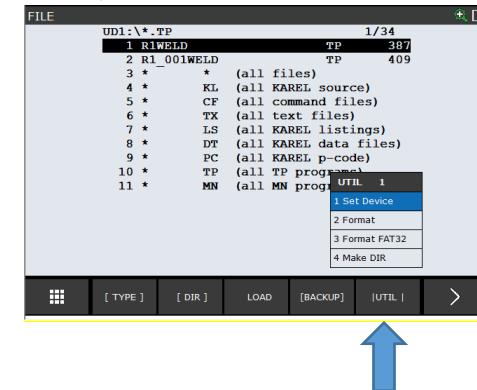
## Auflistung der File Typen



## Backup



## Setup



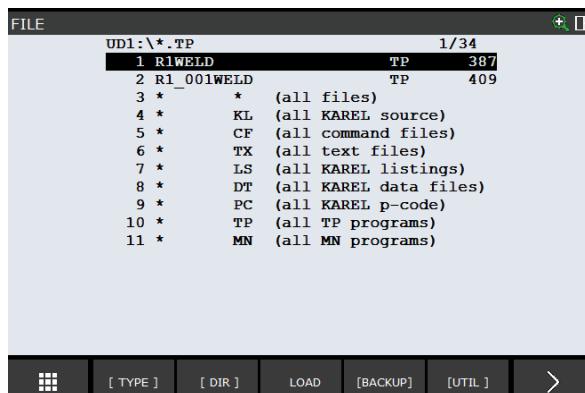
Taste **Menu** drücken, **FILE** auswählen im Menu FILE 1 wieder **File** auswählen dann werden alle Dateitypen angezeigt.  
Unter **UTIL** wird mit **Set Device** das Speichermedium eingestellt. z.B.: **USB Disk**

Anschließend wird der Dateityp der gesichert werden soll bestimmt. z.B.: **TP**

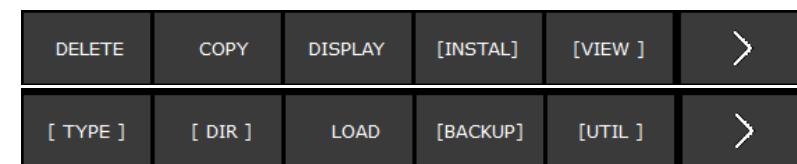
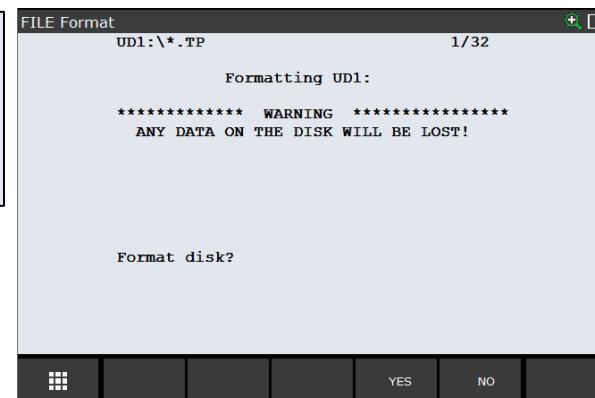
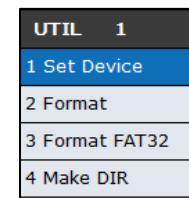
Alle TP Programme werden angezeigt.

**Backup** – sichern der TP Programme auf **USB**

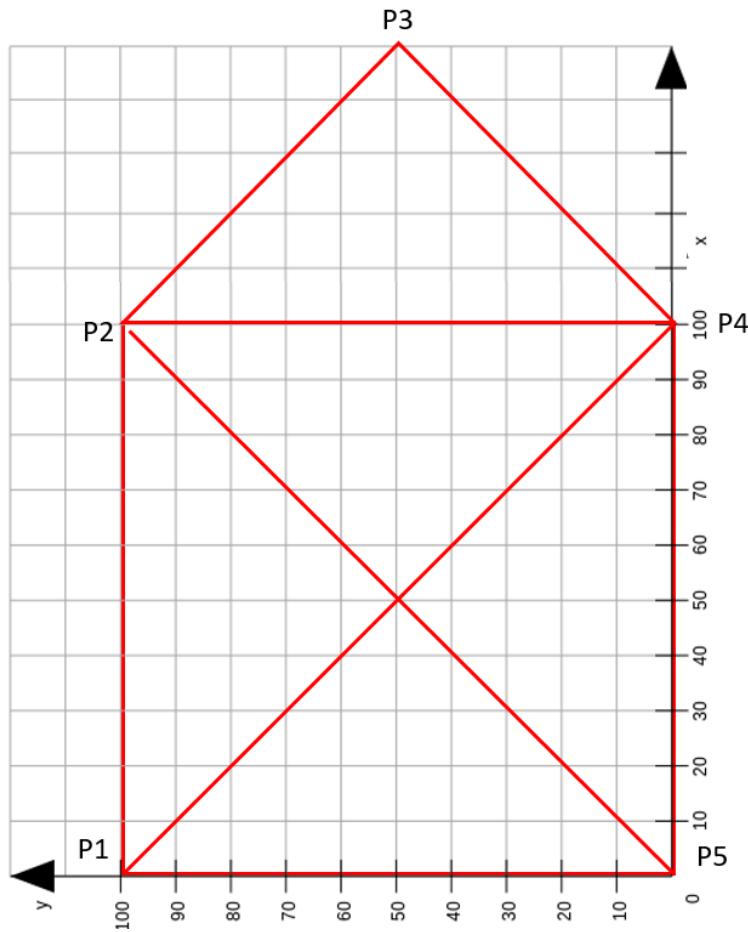
Durch **ALL YES NO** kann entschieden werden, welche Programme gesichert werden.



Formatierung / Verzeichnis erstellen



# Haus vom Nikolaus



P5 Teachen

P1 (X=0 , Y=100 , Z= 0 )

P2 ( X= , Y= , Z= )

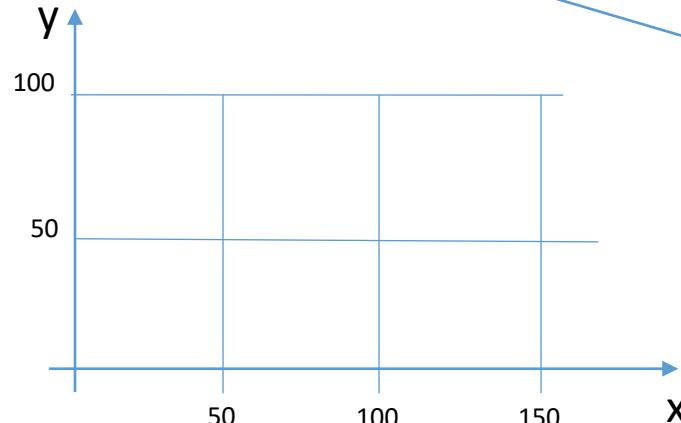
P3 ( X= , Y= , Z= )

P4 ( X= , Y= , Z= )

Sie finden ein Programm vor und analysieren es. Zeichnen sie in das Koordinatensystem die Bewegungen des Roboters mit Pfeilen ein.

RECHTECK

1: UFRAME\_NUM=1  
2: UTOOL\_NUM=2  
3: PAYLOAD[1:Gpr without part]  
4: TIMER[1]=RESET  
5: TIMER[1]=START  
6: !Offset PR()  
7:L @PR[15:TEACH] 100mm/sec FINE  
8:L @PR[15:TEACH] 100mm/sec FINE  
: Offset,PR[17:Y100]  
9:L @PR[15:TEACH] 100mm/sec FINE  
: Offset,PR[18:X100 Y100]  
10:L @PR[15:TEACH] 100mm/sec FINE  
: Offset,PR[19:X150 Y 50]  
11:L @PR[15:TEACH] 100mm/sec FINE  
: Offset,PR[18:X100 Y100]  
12:L @PR[15:TEACH] 100mm/sec FINE  
: Offset,PR[19:X150 Y 50]  
13:J @PR[15:TEACH] 100% CNT100  
14: R[31]=R[30]\*R[30]  
15: PR[40,1:=0]=R[31]



## Daten

DATA Position Reg

15/100

PR[ 10:P2	]=R
PR[ 11:P3	]=R
PR[ 12:P4	]=R
PR[ 13:P5	]=R
PR[ 14:	]=R
PR[ 15:TEACH	]=R
PR[ 16:X100	]=R
PR[ 17:Y100	]=R
PR[ 18:X100 Y100	]=R
PR[ 19:X150 Y 50	]=R
PR[ 20:HP1	]=R

Press ENTER

R[ 30: ]=50  
R[ 31: ]=2500

DATA Position Reg

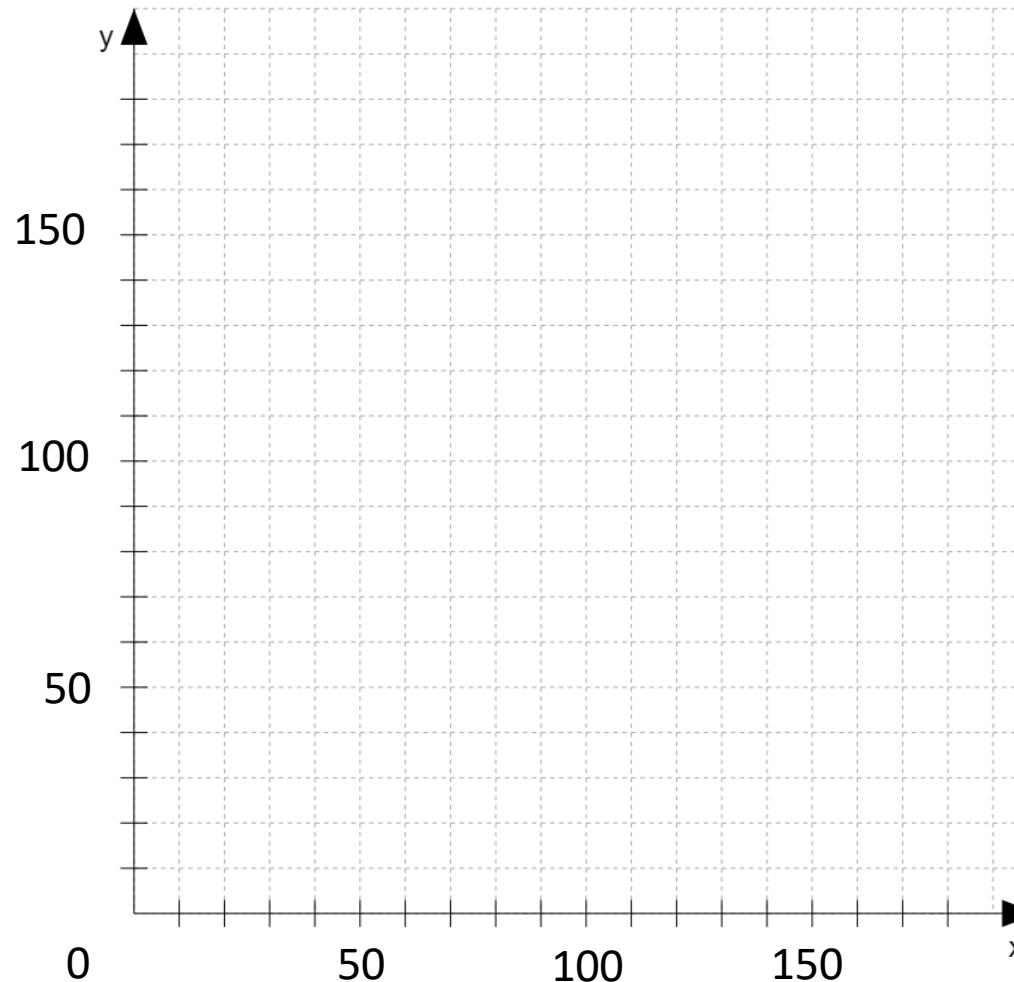
PR[40] UF:F UT:F CONF:NDB 000	
X 2500.000 mm	W 0.000 deg
Y 0.000 mm	P 0.000 deg
Z 0.000 mm	R 0.000 deg

Position Detail

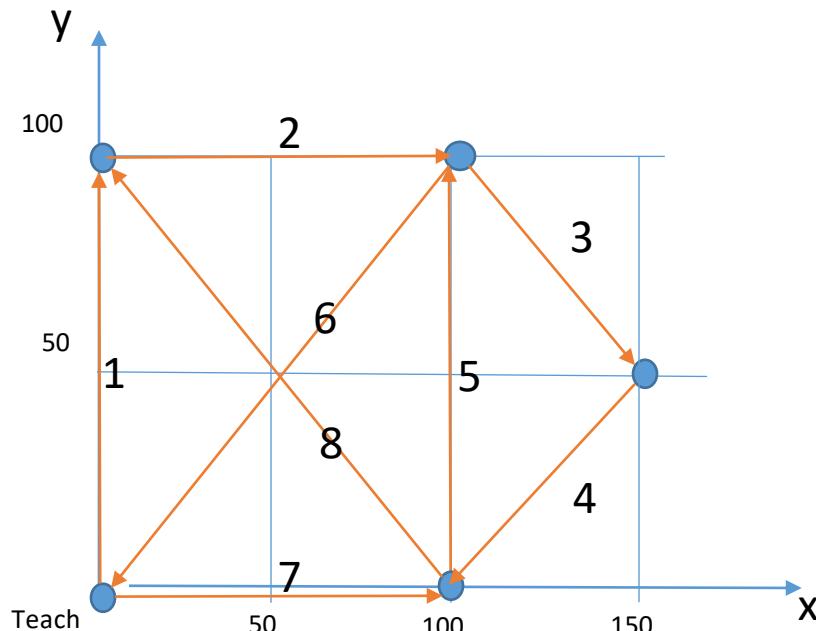
PR[ 38:	]=*
PR[ 39:	]=*
PR[ 40:=0	]=R
PR[ 41:	]=*
PR[ 42:	]=*
PR[ 43:	]=*
PR[ 44:	]=*

Enter value

# Zeichne die Programmfolge ein



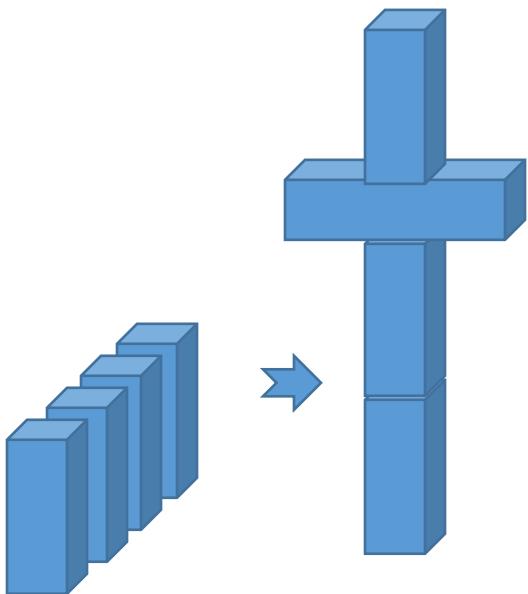
Ändern sie das Programm so, dass der Roboter „das Haus vom Nikolaus“ zeichnet.



## Daten

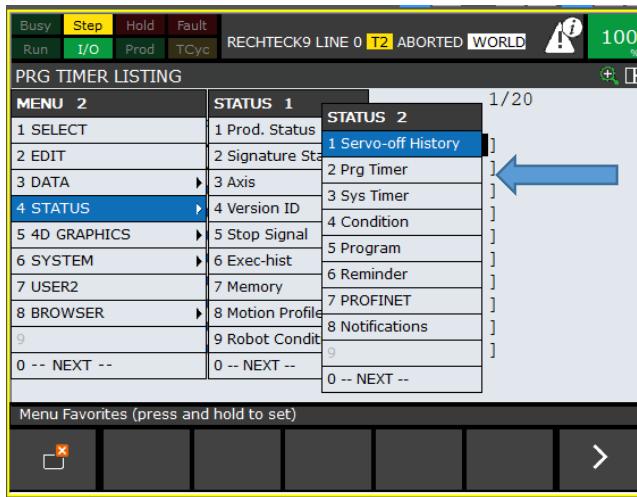
```
DATA Position Reg 15/100
PR[ 10:P2      ]=R
PR[ 11:P3      ]=R
PR[ 12:P4      ]=R
PR[ 13:P5      ]=R
PR[ 14:        ]=R
PR[ 15:TEACH   ]=R
PR[ 16:X100    ]=R
PR[ 17:Y100    ]=R
PR[ 18:X100 Y100 ]=R
PR[ 19:X150 Y 50 ]=R
PR[ 20:HP1      ]=R
Press ENTER
```

# Stapeln von Holzklötzchen



```
1:  UFRAME_NUM=1 ;
2:  UTOOL_NUM=5 ;
3:  ;
4:  ;
5:  CALL GR_OPEN ;
6:  WAIT RI[3:Offen]=ON ;
7:  ;
8:J PR[6:HOME] 100% FINE ;
9:  ;
10:L P[2] 3000mm/sec FINE ;
11:L P[1] 100mm/sec FINE ;
12: CALL GR_CLOSE ;
13:  ;
14:L P[3] 100mm/sec FINE ;
15:J P[4] 100% CNT10 ;
16:L P[5] 100mm/sec FINE ;
17: CALL GR_OPEN ;
18:L P[6] 300mm/sec CNT0 ;
19:  ;
20:J P[9] 100% CNT0 ;
21:L P[11] 100mm/sec FINE ;
22: CALL GR_CLOSE ;
23:L P[12] 100mm/sec FINE ;
24:L P[14] 3000mm/sec CNT10 ;
25:L P[13] 100mm/sec FINE ;
26: CALL GR_OPEN ;
27:  ;
28:L P[15] 100mm/sec FINE ;
29:L P[16] 3000mm/sec CNT0 ;
30:L P[18] 100mm/sec FINE ;
31: CALL GR_CLOSE ;
32:J P[19] 100% FINE ;
33:  ;
34:L P[20] 3000mm/sec CNT0 ;
35:L P[21] 100mm/sec FINE ;
36: CALL GR_OPEN ;
37:L P[22] 100mm/sec FINE ;
38:  ;
39:J P[23] 100% CNT10 ;
40:L P[24] 3000mm/sec CNT0 ;
41:L P[25] 100mm/sec FINE ;
42: CALL GR_CLOSE ;
43:L P[26] 100mm/sec FINE ;
44:J P[28] 100% CNT0 ;
45:L P[27] 100mm/sec FINE ;
46: CALL GR_OPEN ;
47:J P[29] 100% FINE ;
48:J PR[6:HOME] 100% FINE ;
```

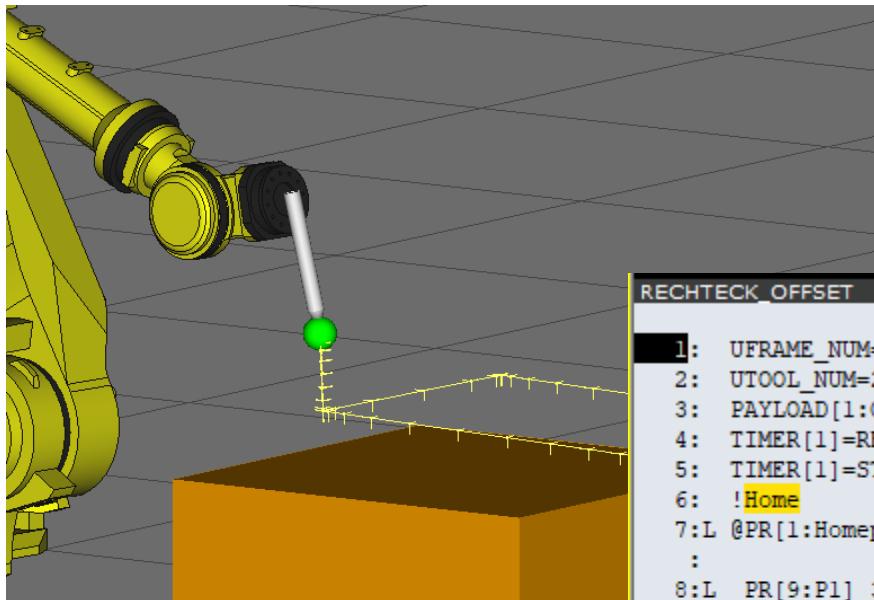
## Timer Listing



TIMER	count	comment
1	34.94	(s) [ ]
2	4.00	(s) [ ]
3	10.17	(s) [Gripper]
4	0.00	(s) [ ]
5	0.00	(s) [ ]
6	0.00	(s) [ ]
7	0.00	(s) [ ]
8	0.00	(s) [ ]
9	0.00	(s) [ ]
10	0.00	(s) [ ]

Busy Step Hold Fault  
Run I/O Prod TCyc RECHTECK9 LINE 0 T2 ABORTED WORLD 100 %

## Rechteck mit Positionsregistern



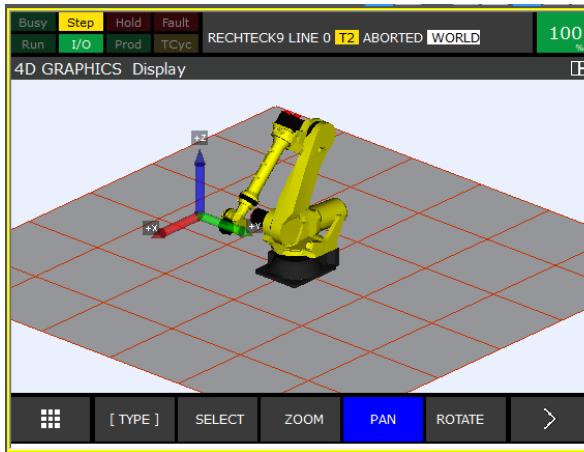
RECHTECK\_OFFSET  
1/15

```
1: UFRAME_NUM=1
2: UTOOL_NUM=2
3: PAYLOAD[1:Grp without part]
4: TIMER[1]=RESET
5: TIMER[1]=START
6: !Home
7:L @PR[1:Homepos] 3000mm/sec CNT0
:
8:L PR[9:P1] 3000mm/sec CNT0
9:L PR[10:P2] 3000mm/sec CNT0
10:L PR[11:P3] 3000mm/sec CNT0
11:L PR[12:P4] 3000mm/sec CNT0
12:L PR[9:P1] 3000mm/sec CNT0
13:L @PR[1:Homepos] 3000mm/sec CNT0
:
14: R[25:TIMER]=TIMER[1]
[End]
```

DATA Position Reg  
1/100

PR[ 1:Homepos ]=R
PR[ 2:Maintenance ]=R
PR[ 3:TPM ]=R
PR[ 4:Start_1 ]=R
PR[ 5:HOME1 ]=R
PR[ 6:RackPos1 ]=R
PR[ 7:OffsetRack ]=R
PR[ 8:WARTEN ]=R
PR[ 9:P1 ]=R
PR[ 10:P2 ]=R
PR[ 11:P3 ]=R
PR[ 12:P4 ]=R
PR[ 13:P5 ]=R
PR[ 14:x100 ]=R
PR[ 15:y100 ]=R
PR[ 16:x100y100 ]=R
PR[ 17:x150y50 ]=R
PR[ 18:X100 Y100 ]=R
PR[ 19:X150 Y 50 ]=R
PR[ 20:HP1 ]=R

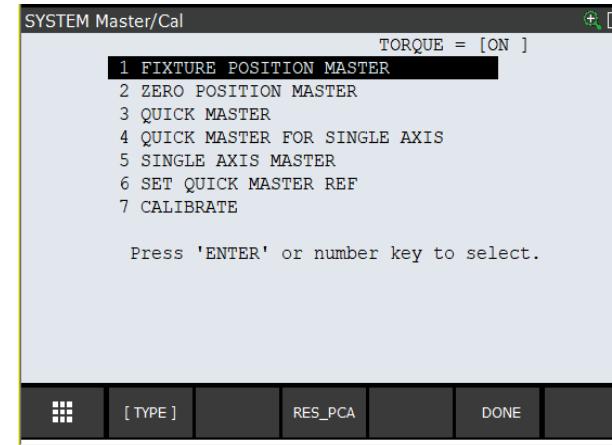
## 4D Grafik



## Payload

MOTION/PAYOUT SET		1/8			
Group 1					
1 Schedule No[ 1]:[Grp without part]					
2 PAYLOAD	[kg]	160.00			
3 PAYLOAD CENTER X	[cm]	2.64			
4 PAYLOAD CENTER Y	[cm]	2.87			
5 PAYLOAD CENTER Z	[cm]	13.10			
6 PAYLOAD INERTIA X	[kgf cms^2]	163.12			
7 PAYLOAD INERTIA Y	[kgf cms^2]	573.79			
8 PAYLOAD INERTIA Z	[kgf cms^2]	648.50			

## Roboter Mastern



## Roboter Axis Limits

SYSTEM Axis Limits				
AXIS	GROUP	LOWER	UPPER	1/84
1	1	-185.00	185.00	dg
2	1	-60.00	76.00	dg
3	1	-132.04	180.00	dg
4	1	-360.00	360.00	dg
5	1	-125.00	125.00	dg
6	1	-360.00	360.00	dg
7	0	0.00	0.00	mm
8	0	0.00	0.00	mm
9	0	0.00	0.00	mm
10	0	0.00	0.00	mm

## Greifer öffnen / schließen

The screenshot shows a software interface for a Fanuc robot. At the top, there are tabs for Busy, Step, Hold, Fault, Run, I/O, Prod, and TCyc. The status bar indicates "WUERFEL1 LINE 0 T2 ABORTED WORLD" and "100 %".

The main area displays a program list for "WUERFEL1" with 31 lines of code. The code is as follows:

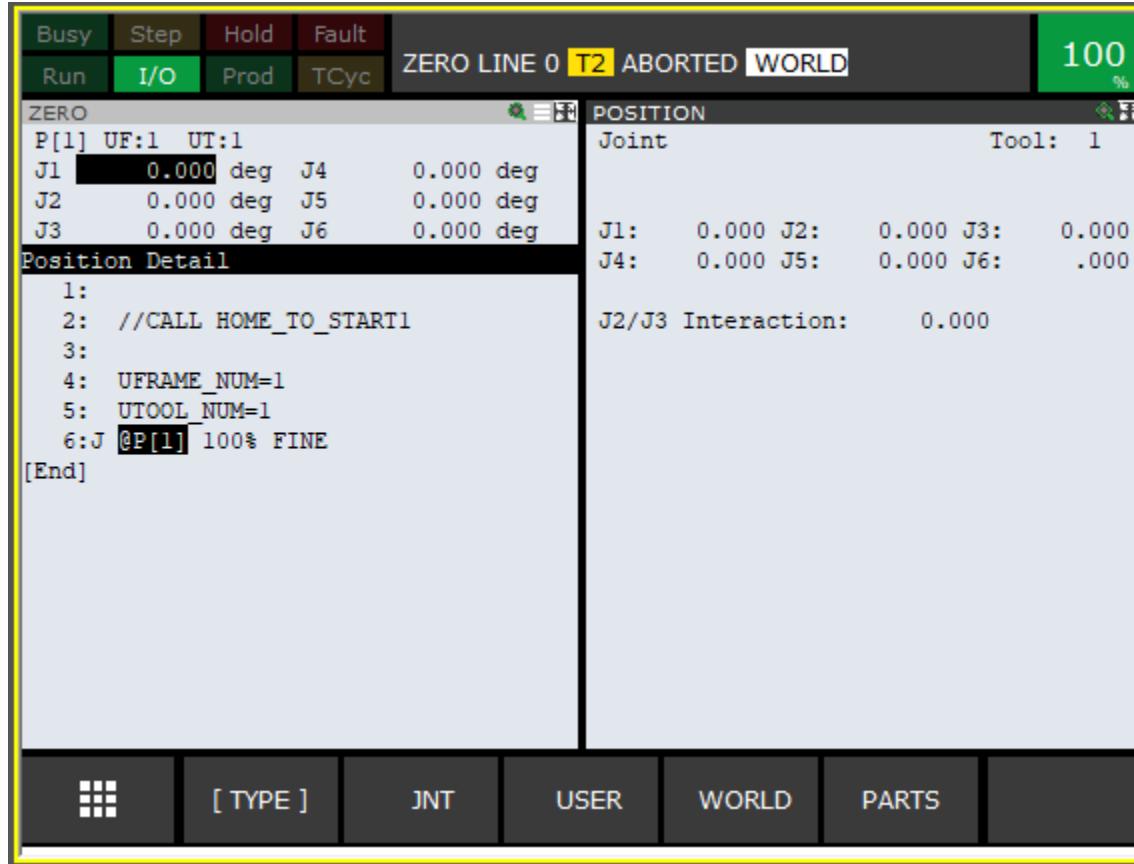
```
1: UFRAME_NUM=1
2: UTOOL_NUM=5
3: PAYLOAD[1:Grp without part]
4: IF (RI[4:Geschlossen]) THEN
5: RO[2:Greifer AUF]=ON
6: ENDIF
7:
8: LBL[1]
9:J P[1] 100% CNT100
10:J @P[2] 100% CNT0
11:L P[3] 3000mm/sec CNT10
12: WAIT .25(sec)
13: RO[1:Greifer ZU]=ON
14:J P[4] 100% CNT0
15:J P[5] 100% CNT100
16:J P[6] 100% CNT0
17: RO[1:Greifer ZU]=OFF
18:L P[7] 3000mm/sec CNT10
19:J P[8] 100% CNT100
20:J P[9] 100% CNT0
```

Below the code, there is an "I/O Robot Out" table with 8 rows. The table has columns for #, SIM, STATUS, and a description. The status for row 2 is "ON" (green), while others are "OFF" (red). The descriptions are: "Greifer ZU" for row 1, "Greifer AUF" for row 2, and empty brackets for rows 3-8.

#	SIM	STATUS	
RO[ 1]	U	OFF	[Greifer ZU ]
RO[ 2]	U	ON	[Greifer AUF ]
RO[ 3]	U	OFF	[ ]
RO[ 4]	U	OFF	[ ]
RO[ 5]	U	OFF	[ ]
RO[ 6]	U	OFF	[ ]
RO[ 7]	U	OFF	[ ]
RO[ 8]	U	OFF	[ ]

At the bottom, there are buttons for [ TYPE ], IN/OUT, ON (green), OFF (red), and a right-pointing arrow.

## Programm ZERO J1 bis J6 = 0° zum Mastern



The screenshot shows the Fanuc Robo Trainer software interface. The top bar displays 'Busy', 'Step', 'Hold', 'Fault', 'Run' (highlighted in green), 'I/O', 'Prod', 'TCyc', 'ZERO LINE 0 T2 ABORTED WORLD', and a battery icon at 100%. The main area is titled 'POSITION' and shows joint positions: J1: 0.000 deg, J2: 0.000 deg, J3: 0.000 deg, J4: 0.000 deg, J5: 0.000 deg, J6: 0.000 deg. The 'Tool: 1' is also indicated. Below this, a 'Position Detail' section shows the following program code:

```
P[1] UF:1 UT:1
J1 0.000 deg J4 0.000 deg
J2 0.000 deg J5 0.000 deg
J3 0.000 deg J6 0.000 deg
Position Detail
1:
2: //CALL HOME_TO_START1
3:
4: UFRAME_NUM=1
5: UTOOL_NUM=1
6:J @P[1] 100% FINE
[End]
```

The bottom navigation bar includes icons for HOME, TYPE, JNT, USER, WORLD, and PARTS.