



Fachschule für Technik

an den Beruflichen Schulen des Unstrut- Hainich- Kreises
Sondershäuser Landstraße 39
99974 Mühlhausen



Fachhochschule Jena
University of Applied Sciences Jena

Carl-Zeiss-Promenade 2
07745 Jena

Thema

**Entwurf und Inbetriebnahme eines Praktikumsversuches für das Labor
Maschinenlehre unter Verwendung des vorhandenen KUKA-Roboters vom Typ
KR30/2.**

Praktikumsbericht von
Michél Eberhardt
ET07VDV

Fachlehrer: Dipl.-Ing. D. Gerst
Betreuer Betrieb: Dipl.-Ing.(FH) Andreas Riess
März 2009

Inhaltsverzeichnis

1	Aufgabenstellung und Einleitung	1
2	Aktueller Stand	1
2.1	Kurzbeschreibung des Systems	2
2.2	Werkzeug	3
3	Durchführung	3
3.1	Einarbeitung	3
3.2	Dokumentation	4
3.3	Inbetriebnahme des Kollisionsschutzes	4
3.4	Kalibrieren der Achsen	5
3.5	Einrichten der Arbeitsumgebung	5
3.6	Programmiersprache und Syntax	5
3.7	Entwurf der Programmieraufgabe „Würfel“	6
4	Ergebnisse	7
5	Literaturverzeichnis	8
6	Anhang	9
6.1	Anhang A	9
6.2	Anhang B	10
6.3	Anhang C	17
6.4	Anhang D	22
6.5	Anhang E	34

1 Aufgabenstellung und Einleitung

Ziel des Praktikums war der Entwurf und die Inbetriebnahme eines Praktikumsversuches für das Labor Maschinenlehre unter Verwendung des vorhandenen KUKA-Roboters vom Typ KR30/2 (siehe Bild 1). Folgende Aufgaben waren während des Praktikums zu lösen:

- Entwurf des Versuches
- Inbetriebnahme des Kollisionsschutzsensors
- Programmierung unter Verwendung der Maschinenkoordinatensysteme
- Erarbeiten der Versuchsanleitung
- ausführlicher Test des Versuches
- Dokumentation
- Übergabe des Gesamtaufbaus an den Betreuer

Die Motivation, aus der vorhandenen Technik im Labor Maschinenlehre einen weiteren Praktikumsversuch zu erarbeiten, ist die Erweiterung des Praktikums Maschinenlehre im Fachbereich Maschinenbau. In dem Versuch sollen die Grundkenntnisse der Robotik anschaulich vermittelt werden können.

Außerdem werden die Studenten in die Lage versetzt, den Roboter selbstständig in Betrieb zu nehmen, ein eigenständiges Programm zu erstellen und im praktischen Testlauf umzusetzen. Dabei sollen Grundkenntnisse der Steuerung des KR30/2, sowie der sichere Umgang mit verschiedenen Koordinatensystemen und deren Transformation erlangt werden. Weiterhin werden grundlegende Programmierkenntnisse und die Eigenheiten der KUKA-Programmierung vermittelt.

2 Aktueller Stand

Ausgangspunkt für das Betriebspraktikum war der KUKA-KR30/2, der sich nach dem hausinternen Umzug des Fachbereiches MB ungenutzt im Maschinenlabor befand. Der Spezialist für den Roboter verließ altersbedingt die Fachhochschule, wobei eine ausführliche Einarbeitung des Nachfolgers, meines Betreuers Herrn Dipl.-Ing. (FH) A. Riess, nicht erfolgte. Folglich konnte die Inbetriebnahme des KUKA ausschließlich anhand der technischen Dokumentation erfolgen.

2.1 Kurzbeschreibung des Systems

Der Roboter KR30/2 ist ein sechachsiger Industrieroboter und gehört zur Kategorie der Knickarmroboter. Haupteinsatzgebiete sind: Montieren, Auftragen von Kleb- Dicht- und Konservierungsstoffen, Bearbeiten, MIG/MAG- Schweißen und YAG- Laserstrahlschweißen.

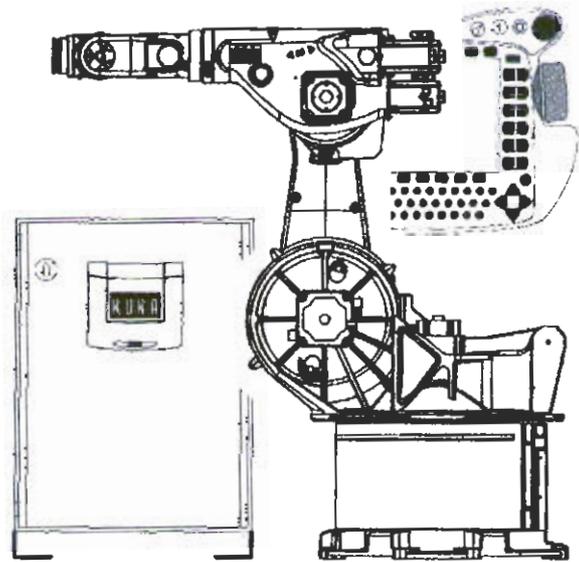


Bild 1 Das KUKA KR 30/2 System

Nenn-, Traglasten und Zusatzlasten können auch bei maximaler Armausladung mit maximaler Geschwindigkeit bewegt werden (siehe Bild 2). Alle Grundkörper der beweglichen Hauptbaugruppen bestehen aus Leichtmetallguss. Dieses Auslegungskonzept ist im Hinblick auf wirtschaftlichen Leichtbau und hohe Torsions- und Biegefestigkeit CAD- und FEM- optimiert. Gelenke und Getriebe bewegen sich weitgehend spielfrei, alle bewegten Teile sind abgedeckt. Alle Antriebsmotoren sind steckbare, bürstenlose AC- Servomotoren und somit weitgehend wartungsfrei und gegen Überlastung geschützt. Die Grundachsen sind dauergeschmiert, d.h. ein Ölwechsel ist frühestens nach 20.000 Betriebsstunden erforderlich. Die durchschnittliche Lebensdauer liegt bei 10 bis 15 Jahren.¹ Notwendige Sensor- und Pneumatikleitungen werden direkt an den Achsen entlang bis zu dem Werkzeug geführt.

Die Steuer- und Leistungselektronik ist in dem externen Steuerschrank KR C1 untergebracht, dieser beinhaltet auch den Windows 95 gestützten Hauptrechner.

Robotertyp	KR 30/2
Gesamtgewicht [kg]	880
Nenn- Traglast [kg]	30
Zusatzlast bei Nenntlast [kg]	35
Max. Gesamtlast [kg]	65

Bild 2 Technische Daten

Das KUKA-Control-Panel KCP bildet die Schnittstelle zwischen Mensch und Maschine zur Bedienung der Robotersteuerung. Alle Elemente zur Programmierung und Bedienung des

¹ vgl. KUKA Roboter GmbH: www.kukakorea.com/data/sheet/kr30_60_3_de_en_fr.pdf, S. 3.

Robotersystems sind - mit Ausnahme des Hauptschalters - direkt am KCP angebracht. Wahlweise kann die Programmierung direkt über den Windows- Rechner durchgeführt werden.

2.2 Werkzeug

Am Roboter ist eine 220 mm lange Pneumatikzange befestigt, die das Greifen und Umsetzen verschiedenster Gegenstände im Schwenkbereich des Armes ermöglicht. Hiermit lassen sich Traglasten bis 5,4 kg problemlos befördern.² Aus Sicherheitsgründen ist das Werkzeug mit einen Kollisions- und Überlastschutz vom Typ OPS 160 gekoppelt, der den

Geifer pneumatisch in der Mittelstellung arretiert. Bei plötzlichem Einwirken großer Kräfte, beispielsweise einer Kollision, wird diese Stellung verlassen und ein Read- Sensor übermittelt eine Fehlermeldung an das System, die dann von der Steuerung entsprechend weiter verarbeitet wird.



Bild 3 Schunk Greifwerkzeug mit Kollisions- und Überlastschutz

3 Durchführung

3.1 Einarbeitung

Zu Beginn der Praktikumstätigkeit erfolgte eine Sichtung sämtlicher vorhandener Handbücher und Schaltungsunterlagen. Hinzu kam das Kennenlernen der Roboterkomponenten, insbesondere des Control-Panels, dass die vielseitigen Bedienfunktionen vereint.

Um die Eigenheiten des KUKA- Basic Dialektes schneller zu verstehen, wurden Teile der alten, nicht dokumentieren Programme gesichtet und im sicheren Schrittbetrieb abgefahren. Die daraus gewonnenen Erkenntnisse wurden beim späteren Programmieren vertieft und teilweise korrigiert.

² vgl. Schunk: Datenblatt PGN-plus 125, S.1.

3.2 Dokumentation

Eine der Hauptaufgaben war die Erstellung einer Versuchsanleitung einschließlich der Aufgabenstellung für den Versuch „KUKA-Roboter“. Das setzte die Inbetriebnahme des Roboters, einen Programmwurf und die praktische Realisierung des Versuchs voraus. Die entsprechende Dokumentation erfolgte analog des Praktikumsverlaufes.

3.3 Inbetriebnahme des Kollisionsschutzes

Der oben genannte Kollisionssensor am Greifwerkzeug war bei Antritt des Praktikums nicht angeschlossen, damit war die Situation bezüglich der Sicherheit des Gesamtsystems nicht zufriedenstellend und die Inbetriebnahme dringend erforderlich. Nach dem ausführlichen Studium der Stromlaufpläne wurde die Schaltung „Kollisionsschutz“ (siehe Anhang A) entwickelt und an das bestehende System angepasst.

Hierzu wurden die externen Notaus- Anschlüsse - „Not-Aus Kanal A extern“ und „Not-Aus Kanal B extern“ - mit zusätzlichen Relais- Schließerkontakten beschalten. Demzufolge werden die Steuerzustände des Relais vom System erkannt und der Low- Pegel als Fehler „Externer Not- Aus“ interpretiert. Das Freigabesignal ist derart realisiert, dass Hardwarefehler der Ansteuerung und der vorgeschalteten Bauteile mit einbezogen wurden.

Da die Ansteuerung der Achsen im Fehlerfall von der Software verriegelt wird, ist es notwendig, eine Freischaltmöglichkeit einzubauen, die die Fehlermeldung beendet und ein manuelles Zurücksetzen ermöglicht. Dies wird mit dem im Schaltschrank untergebrachten Taster „S ENA“ verwirklicht (siehe Bild 4), der so eingebaut wurde, dass ein permanentes Überbrücken der Schutzfunktion ohne absichtliche Manipulation der Schaltung unmöglich ist.

Bauteilliste:

- 1x Netzteil Siemens LOGO! Power 24V
- 1x Steuerrelais Omron LY2 2W
- 1x Gehäuse MR2
- 1x Drucktaster 1Ö 1S
- 1.5m Kabel



Bild 4 Externer Notaus- Taster „S ENA“ im Schaltschrank C2

Nachdem die Schaltung in das System integriert wurde, trat ein unvorhersehbarer Effekt am Roboter auf, der die Greiffunktion des Werkzeugs blockierte, was wiederum eine nähere

Untersuchung der Schaltschrankverkabelung erforderlich machte. Da ein Großteil der Einbaumaßnahmen am Stecker X11 erfolgte, konnte die Fehlersuche auf diesen Bereich eingegrenzt werden. Dabei wurde eine kalte Lötstelle am Kontakt 24 entdeckt und wieder angelötet, woraufhin die Funktionstüchtigkeit des Roboters voll hergestellt war.

3.4 Kalibrieren der Achsen

Bevor der Roboter die programmierten Aufgaben ausführen kann, müssen die Achsen in eine definierte Grundposition gebracht werden. Die detaillierte Beschreibung dieses Vorgangs ist im Anhang C unter „2 Justieren und Dejustieren“ nachzulesen. Wenn die Justierung erfolgt ist, kann der Roboter jederzeit über die im System vorhandene Positionsvariable „HOME“ in seine Grundstellung gefahren werden. Die Koordinaten der später geteachten Positionen beziehen sich auf diese Stellung.

3.5 Einrichten der Arbeitsumgebung

Nachdem der Roboter justiert wurde, erfolgte die Ausrichtung der Arbeitsebene unter Verwendung linearer Relativbewegungen, d.h. das der Roboterarm auf die Höhe der Arbeitsplatte abgesenkt wird und dort eine geradlinige Bewegung parallel zum eigenen Standort ausführt (Anhang D „tisch_ausrichten.src“). Dabei musste berücksichtigt werden, dass die komplette Arbeitsebene im Schwenkbereich des Roboters liegt. Die Echtzeiterfassung der Arbeitsumgebung, z.B. über eine Kamera, ist im System nicht vorgesehen. Um zu gewährleisten, dass die erstellten Programme in Zukunft genutzt werden können, musste die Arbeitsebene im Labor fixiert werden.

3.6 Programmiersprache und Syntax

Die von KUKA verwendete Programmiersprache ist der Hochsprache BASIC angelehnt, wobei diese im Speziellen signifikant abweicht. Dies wiederum bedeutet, dass eine entsprechende Einarbeitungszeit vorzusehen ist. Da das einfache Aneinanderreihen geteachter Positionen und redundanter Quellcode nicht den Ansprüchen der Programmieraufgabe genügen, ist es notwendig, grundlegende Algorithmen der Programmierung in der KUKA- Sprache zu verwenden.

Im Fall des Würfel- Programms wurden die oft genutzten Bewegungen in einer Schleife zusammengefasst und teilweise in verschiedene Unterprogramme ausgelagert (siehe Anhang C). Ein KUKA- Programm besteht aus zwei Teilen einer SRC- Datei, in die der eigentliche Ablauf geschrieben wird, und einer DAT- Datei, die beispielsweise die Koordinaten geteachter Positionen enthält. Unterprogramme werden in speziellen SUB- Dateien gespeichert, da diese keine eigene DAT- Datei benötigen. Diese SUB- Programme

können auf globale Variablen in der DAT- Datei des Hauptprogramms zugreifen, wenn diese durch den Befehl „PUBLIC“ gekennzeichnet sind.

3.7 Entwurf der Programmieraufgabe „Würfel“

Der spätere Laborversuch, der im Lehrbetrieb auf Basis dieser Arbeit durchgeführt werden soll, zielt darauf ab, die Grundlagen der Robotik zu vermitteln, was bedeutet, dass der Versuch eine Vielzahl von Funktionen mit einbeziehen muss. Der eigentliche Versuch ist von den Studenten innerhalb von 2 Praktikumsterminen à 1,5h zu bearbeiten, wodurch nicht auf alle Möglichkeiten der Steuerung eingegangen werden kann.

Für den Versuchsentwurf stehen neun Würfel, die auf der Arbeitsebene in einer 3x3 Matrix angeordnet sind, und ein rechteckiges 0,7m hohes Aluminiumprofil, das zur Zwischenlagerung dient, zur Verfügung (siehe Bild 5).

Im geplanten Ablauf wird der Roboter über dem ersten Würfel positioniert und dieser Punkt in der Variablen „PTISCH“ gespeichert. Später wird die Aufnahmeposition in jedem Schleifendurchlauf berechnet. Der Würfel wird vom Greifer fixiert und in das Zwischenlager gebracht, der Vorgang wird nun mit den restlichen Würfeln wiederholt. Dabei werden entsprechende Schleifen, welche die 3x3 Matrix abbilden, verwendet. Danach werden die Würfel der Reihe nach an der unteren Öffnung des Profils entnommen und auf die ursprünglichen Positionen zurück sortiert. Am Ende des Versuches liegen alle Würfel wieder auf der Arbeitsebene und der Roboter fährt in seine HOME- Position.

Darüber hinaus beinhaltet der Entwurf die Erstellung mehrerer Beispielprogramme, die im Einzelnen bestimmte Funktionen veranschaulichen und die Einarbeitungsphase im späteren Laborbetrieb verkürzen. Hierzu gehören die Programme:

„bewegung_mit_winkel.src“,
„bewegungsarten.src“ und
„dynamische_position.src“,

die Beispiele für den Umgang mit verschiedenen Bewegungsarten enthalten (Anhang D). Unterschiedliche Schleifentypen werden in der Datei „schleifen.src“ veranschaulicht. Das



Bild 5 Versuchsaufbau

Verwenden von Unterprogrammen und globalen Variablen, die im Haupt- und Unterprogramm bekannt sind, wird mit den Dateien „testh.src“, „testh.dat“ und „testu.sub“ vermittelt.

4 Ergebnisse

Das Ergebnis meines Praktikums ist ein funktionstüchtiger Laborversuch, der anhand der von mir erstellten Dokumente - Anleitung, Aufgabenstellung (Anhang E) und Programmdokumentationen - auf strukturierte Art und Weise die Grundlagen der Robotersteuerung, -programmierung und Robotik vermittelt.

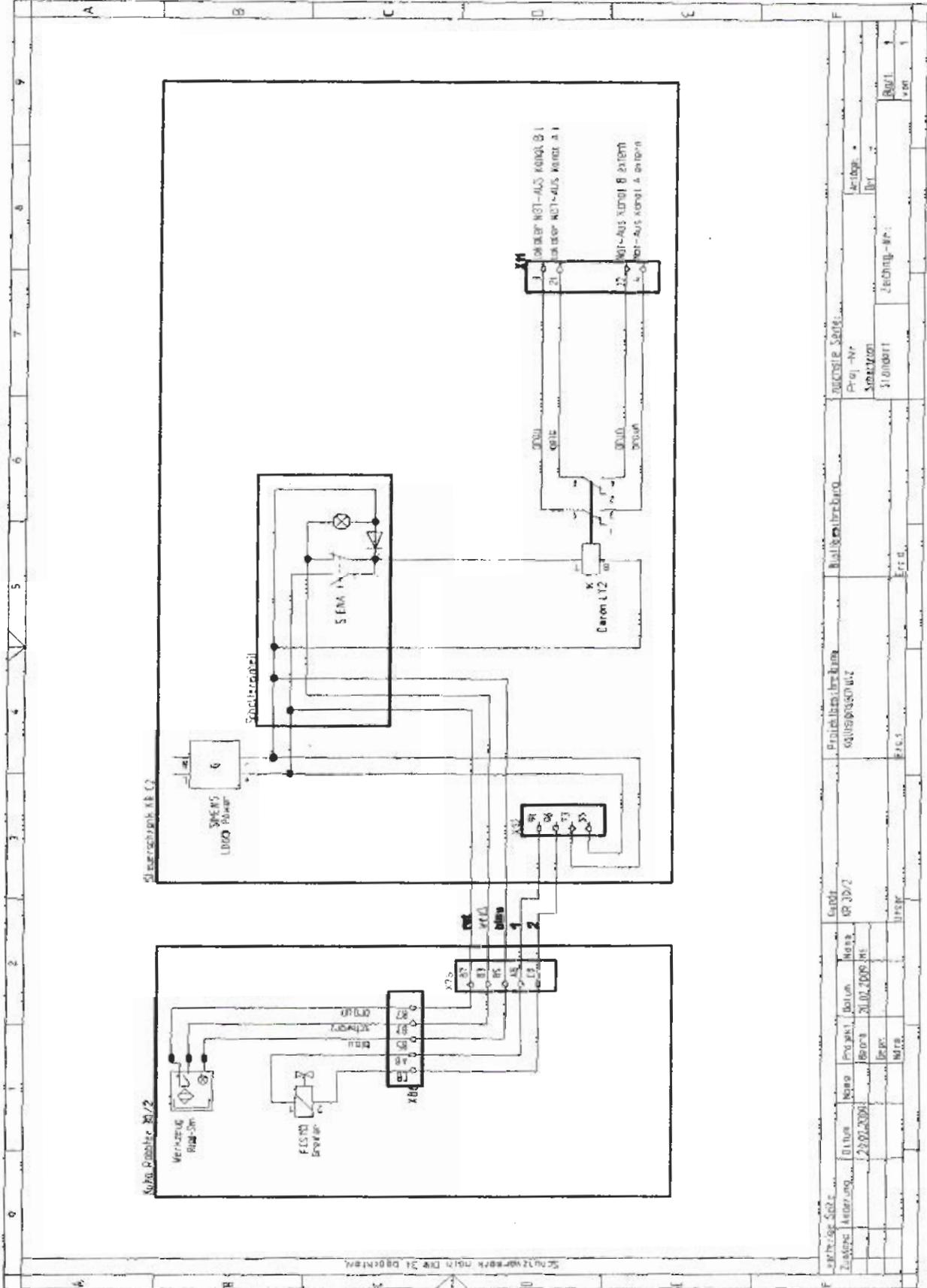
Wesentlich ist, dass durch die erstellte Dokumentation eine selbstständige Einarbeitung der Studenten in die Thematik möglich ist, was den Betreuungsaufwand während des Praktikums erheblich reduziert.

5 Literaturverzeichnis

- a) FST Mühlhausen. 2009. „Einführung in die Roboter-Technik I“. <<[http://www.fachschule-fuer-technik-mhl.de/elp/html/home.htm/Robotertechnik/Einführung in die Robotik](http://www.fachschule-fuer-technik-mhl.de/elp/html/home.htm/Robotertechnik/Einfuehrung_in_die_Robotik)>>. 6.2.2009
- b) KUKA Roboter GmbH. 04.2001. Schaltungsunterlagen. KUKA Roboter KR 30/2
- c) KUKA Roboter GmbH. 05.2001. Seminarhandbuch. Fortgeschrittene Roboterprogrammierung Steuerungstyp KR C2
- d) KUKA Roboter GmbH. 05.2001. Seminarhandbuch. Grundlagen der Roboterprogrammierung Steuerungstyp KR C2
- e) KUKA Roboter GmbH. 11.2004. Spezifikation. <<www.kukakorea.com/data/sheet/kr30_60_3_de_en_fr.pdf>>. 15.2.2009
- f) Schunk GmbH. 07.2002. Datenblatt. PGN-plus 125

6 Anhang

6.1 Anhang A



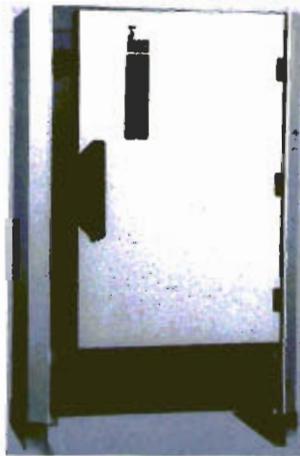
Zeichnungs-Nr.: Blatt: 1 von 1		Projekt-Nr.: Blatt: 1 von 1		Zeichnungs-Nr.: Blatt: 1 von 1	
Projektionsart: 1. Ordnung		Projektionsrichtung: nach unten		Maßstab: 1:1	
Auftraggeber: ...		Projektionsart: 1. Ordnung		Projektionsrichtung: nach unten	
Zeichner: ...		Gezeichnet: ...		Geprüft: ...	
Datum: ...		Projektionsart: 1. Ordnung		Projektionsrichtung: nach unten	

6.2 Anhang B

KUKA KR 30-2 Versuchsanleitung

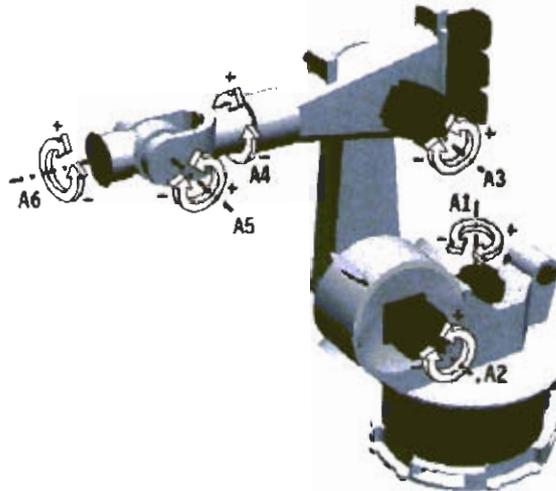
Inbetriebnahme | Programmieren | Testen

0 Bestandteile



KR C1 Steuerschrank mit
Hauptschalter

Bild I



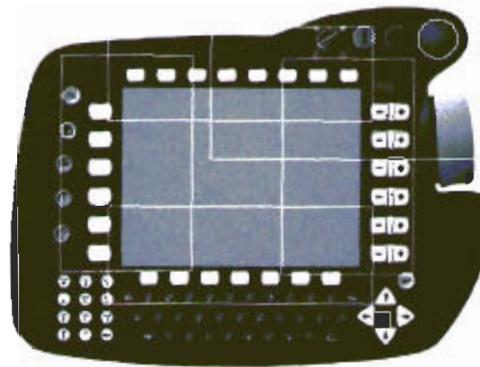
KUKA Roboter / Achsenbezeichnung

Bild II



KUKA Control Panel KCP

Bild III



KCP mit Orientierungsfenstern

Bild IV

1 Einschalten

- Netzstecker und Druckluftanschluss überprüfen (Klimaanlage muss eingeschaltet sein)
- Hauptschalter auf **ON** → Lüfter läuft, Rechner fährt hoch
!Achtung: Wenn die Anlage nicht reagiert Sicherung 2F1.1 im Sicherungskasten einschalten.

- Monitor ein

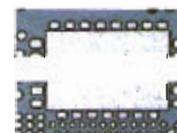
- Schlüsselschalter  auf dem Panel in die untere rechte Stellung.



- Desktopverknüpfung „Verknüpfung mit Cross3“ doppelklicken (dauert mehrere Min.)

- Wechseln in den Expertenmodus: **Konfig.** → **5 Benutzergruppe**

→ **Experte** → Passwort „kuka“ → **Enter**



2 Justieren und Dejustieren

- Justieren

Um den Roboter in eine definierte Grundstellung zu bringen, müssen alle sechs Achsen mit der **Messuhr** (am Programmierplatz) eingestellt werden. Die Justierung erfolgt der Reihe nach von Achse 1 [A1] bis Achse 6 [A6] (siehe Bild II).

- Programm Oberfläche **Inbetriebn.** → **1Justieren** → **0Uhr**
→ die Liste der zu justierenden Achsen erscheint in der rechten Bildschirmhälfte



- Verschlusskappe der jeweilige Messpatrone abnehmen und die **Messuhr** anschrauben. Die **Messpatrone** der Achse 5 ist unter der Verschlusskappe rechts neben dem KUKA Schriftzug zu finden. Der passende Schraubendreher liegt am Platz.



Hinweis: Manuelles Steuern mit der Space- Mouse

1. Voreinstellung

	Handsteuerung einschalten	→	
	Achsspezifisches Koordinatensystem	→	
	Drehgeschwindigkeit +/-		
	Greifwerkzeug auf / zu		

aktuelle Geschwindigkeit 50%

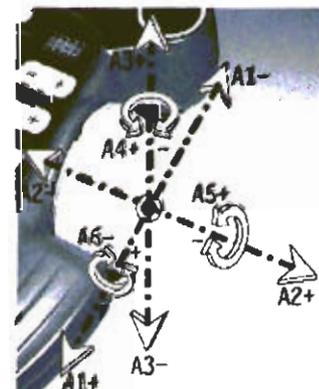
2. Bewegen der sechs Achsen

Eine der Zustimmungstasten (Bild III) in die **Mittelstellung** drücken und halten.

→ nach wenigen Sek. zieht ein Relais hörbar an

→ 2. Feld wird grün

→ solange die **Zustimmungstaste** gehalten wird kann der Roboter mit der **Space- Mouse** gesteuert werden

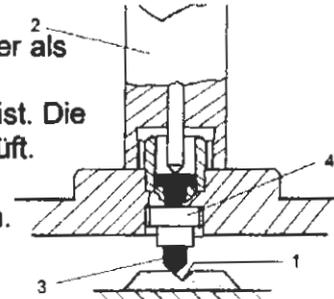


Achtung: Wenn die Zustimmungstaste zu **stark** oder zu **leicht** gedrückt wird schaltet der Roboter ab! → 2. Feld wird rot → Vorgang wiederholen

- Grobe Vorjustage mit relativ hoher Geschwindigkeit >10%
→ **Markierungen** der Achse angleichen (meist mit Weiß gekennzeichnet)

- Bei der Feinjustierung sollte die Geschwindigkeit nicht größer als 3% gewählt werden. Die Achse wird so lang gedreht bis der **Messstift(3)** am **Tiefpunkt** der Messkerbe(1) angekommen ist. Die Einstellung wird mit der angeschraubten Messuhr(2) überprüft.

!Achtung: Die Messkerbe (1) ist nicht an allen Achsen gleich.



- Die Einstellung wird im Rechner durch **doppelklicken** der Achse gespeichert und verschwindet aus der Liste.
- Vorgang für alle Achsen wiederholen bis die Liste leer ist.
- Mit **Schließen** Fertigstellen.



• Dejustieren

Wenn der Roboter falsch justiert wurde, kann man die Einstellung mit dieser Option zurücksetzen.

- **Inbetriebn. → 2 Dejustieren**
→ Achse/n doppelklicken
- **Schließen**



!Achtung: Roboter erst in Grundstellung bringen. Die Achsen lassen sich nur bedingt einzeln dejustieren.

3 Programm anlegen

- Ordner erstellen

Cursor in die linke Bildschirmhälfte → Speicherort wählen → **Neu**



- Programm anlegen

Rechter Bildschirm → Ordner öffnen → **Neu** → **Modul**
Zwei gleichnamige Dateien werden angelegt **.src** und **.dat**.



- Anwählen oder Öffnen

- → **Anwählen**

Das Programm wird in den Roboter geladen und kann abgefahren werden.



Zum Hauptmenü **Bearbeiten** → **7 Programm abwählen**



!Achtung: Änderungen im Quellcode werden nur bedingt übernommen.

Hinweis: Programm testen



Betriebsart wählen Durchlauf oder Schrittweise (Starttaste muss nach jedem Schritt und Hilfspunkt neu gedrückt werden).

Anwählen →

Zustimmtaste (Bild III) +  halten
wie im Hinweis oben



Nachdem mindestens eine Position angefahren wurde, kann das Programm über die Schaltfläche **Satzanwahl**, an einer beliebigen Stelle weiterfahren.



Mit **Bearbeiten** → **8 Programm zurücksetzen** kann erneut gestartet werden.



Ablauf mit hoher Geschwindigkeit: Schlüsselschalter auf **linke untere** Stellung
!Achtung: **Nur mit getesteten Programmen!!!**



- → **Öffnen**

Editor zum erstellen und ändern des Programms.



Mit **Schließen** → Speichern **ja / nein** wird der Editor beendet.



4 Programmieren

Die Programme werden in einen Basicdialekt geschrieben. Um den Einstieg zu erleichtern sollten die vorhandenen Beispielprogramme `KRC:\R1\Beispielprogramme` genutzt werden.

- Arbeiten im Editor

Die aus Win bekannten Funktionen **Ausschneiden**, **Kopieren**, **Einfügen** und **Löschen** (der kompletten Zeile) lassen sich ausschließlich über **Bearbeiten** nutzen.  Siehe auch 3 Öffnen. Kommentare mit Semikolon „;Text...“

- Variablen

Müssen über die Programmzeile INI geschrieben werden. Tipp: Dazu die SRC- Datei im Win- Editor bearbeiten. (siehe „5 Abschalten der Anlage“ Kuka- Programm minimieren)

- Bewegen

- `Beispielprogramme\bewegungsarten.src`
- `Beispielprogramme\bewegen_mit_winkel.src`
- `Beispielprogramme\dynamische_position.src`

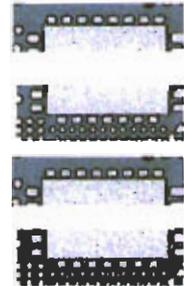
- Teachen (**Position manuell anfahren** und im Prog. abspeichern)

- Position fest: Cursor in gewünschte Programmzeile → **Befehl**

→ **1 Bewegen** → Bewegungsart → **Touch Up** → **Ja** → **Befehl OK**

oder
- Position veränderbar: Cursor... → **Befehl** → **6 KRL Assistent**

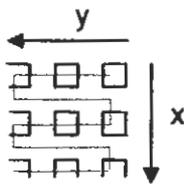
→ Bewegungsart → **{?}** (Datentyp wählen z.B. POS) → **Enter**



- Schleifen

`Beispielprogramme\schleifen.src`

Bsp. Matrix



```

for i=0 to ... ; erste Zählschleife (↓)
for j=0 to ... ; zweite Zählschleife (←)
...
y=y+1          ; y-Wert erhöhen
endfor
...
y=y-...        ; y-Wert zurücksetzen
x=x+1          ; x-Wert erhöhen
endfor
    
```

- Unterprogramm/ SUB- Programm (siehe auch 3 Programm anlegen)

Modul + Modul `Beispielprogramme\einfaches_unterprog\hauptprog.src`

Modul + Expert Submit (`Beispielprogramme\einfaches_unprog_mit_var_uebergabel`
`testh.src`, globale Variable in `testh.dat`)

- Testlauf

Programm abspeichern → (siehe Hinweis: Programm testen)

5 Abschalten der Anlage

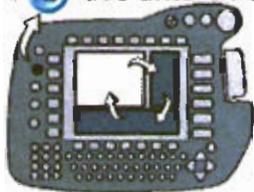
- Roboter in Grundstellung fahren z.B. mit KRC:\R1\Beispielprogramme\home.src
- In den grauen Bereich, über den unteren Schaltflächen, klicken und das KRL- Programm **Beenden**. An dieser stelle ist auch **Minimier.** möglich.
- PC **herunterfahren** und anschließend Hauptschalter **OFF**



6 Mögliche Fehler

- Roboter reagiert nicht.

- Mit  das untere Fenster aktivieren → **Alles Quitt**



!Achtung: Einige Fehlermeldungen wie z.B. Not-Aus müssen auf diese Weise bestätigt werden.



- Programmende siehe Hinweis: Programm testen
 - Stecker am Greifer prüfen muss grün und orange Leuchten (Wackelkontakt)
- Kollision
 - Stecker am Greifer leuchtet grün und orange → manuell aus der Gefahrenbereich bringen (wie im Hinweis: Manuelles Steuern mit der Space- Mouse)
 - Kollisionsschutz hat ausgelöst (externer Not-Aus) → Tasterbeleuchtung erlicht, Greiferstecker leuchtet nur grün → **Taster** im Steuerschrank gedrückt halten (Kollisionsschutz wird solange überbrückt) → zweite Person kann Roboter steuern (siehe oben) → **Greifer** in Mittelstellung bringen → Taster leuchtet, Greiferstecker grün und orange (alles i.O.)
 - Bewegung wird nicht ausgeführt
 - Geschriebene Position außer Reichweite

6.3 Anhang C

Eine Lösungsmöglichkeit des „Würfel- Versuches“ mit zwei Unterprogrammen. Der grau markierte Quellcode wurde automatisch erstellt.

Hauptp.src

```
&ACCESS RVP
&REL 20
&PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe
&PARAM EDITMASK = *
DEF hauptp( )

EXT up1( ) ;unterprogramme bekannt geben
EXT up2( )
DECL E6POS VPTISCH ;var für position decl

;FOLD INI;%{PE}%V3.2.0,%MKUKATPBASIS,%CINIT,%VCOMMON,%P
;FOLD BAS INI;%{E}%V3.2.0,%MKUKATPBASIS,%CINIT,%VINIT,%P
GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
INTERRUPT ON 3
BAS {#INITMOV,0 }
;ENDFOLD (BAS INI)
;FOLD A20 INI;%{E}%V3.2.0,%MKUKATPA20,%CINIT,%VINIT,%P
IF ARC20==TRUE THEN
A20 {ARC_INI}
INTERRUPT DECL 6 WHEN $CYCFLAG[3]==FALSE DO A20(TECH_STOP2)
ENDIF
;ENDFOLD (A20 INI)
;FOLD A10 INI;%{E}%V3.2.0,%MKUKATPARC,%CINIT,%VINIT,%P
IF A10_OPTION==#ACTIVE THEN
INTERRUPT DECL 4 WHEN $CYCFLAG[2]==FALSE DO A10 (#APPL_ERROR)
INTERRUPT DECL 7 WHEN A_ARC_SWI==#ACTIVE DO A10 (#ARC_SEAM)
INTERRUPT DECL 5 WHEN A_FLY_ARC==TRUE DO A10 (#HPU_ARC)
INTERRUPT ON 5
A10_INI ( )
ENDIF
;ENDFOLD (A10 INI)
;FOLD GRIPPER INI;%{E}%V3.2.0,%MKUKATPGRIPPER,%CINIT,%VINIT,%P
USER_GRP(0,DUMMY,DUMMY,GDEFAULT)
;ENDFOLD (GRIPPER INI)
;FOLD SPOT INI;%{E}%V3.2.0,%MKUKATPSPOT,%CINIT,%VINIT,%P
USERSPOT{#INIT}
;ENDFOLD (SPOT INI)
;FOLD TOUCHSENSE INI;%{E}%V3.2.0,%MKUKATPTS,%CINIT,%VINIT,%P
IF H70_OPTION THEN
INTERRUPT DECL 15 WHEN $MEAS_PULSE[TOUCH_I[TOUCH_ACTIVE].IN_NR] DO
H70 (6,CD0 )
INTERRUPT DECL 16 WHEN $ZERO_MOVE DO H70 (7,CD0 )
INTERRUPT DECL 17 WHEN $TECHPAR_C[FG_TOUCH,%}>0.5 DO H70 (8,CD0)
H70 (1,CD0 )
ENDIF
;ENDFOLD (TOUCHSENSE INI)
;FOLD USER INI;%{E}%V3.2.0,%MKUKATPUSER,%CINIT,%VINIT,%P
;Make your modifications here
```

```

;ENDFOLD (USER INI)
;ENDFOLD (INI)
;FOLD          PTP          HOME          Vel=          100          %
DEFAULT;%{PE}%V3.2.0,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME,
3:, 5:100, 7:DEFAULT
$BWDSTART = FALSE
$H_POS=XHOME
PDAT_ACT=PDEFAULT
BAS (#PTP_DAT )
FDAT_ACT=FHOME
BAS (#FRAMES )
BAS (#VEL_PTP,100 )
PTP XHOME
;ENDFOLD

```

```

PTP {AXIS:A2 -90,A3 90,A5 90} ; Achse 5 90 Grad
PTP_REL {x -291,y -105,z 570} ;pos über Würfel 1
VPTISCH=$POS_ACT             ;pos speichern
x=VPTISCH.x                  ;array in var auslesen
y=VPTISCH.y
z=VPTISCH.z
a=VPTISCH.a
b=VPTISCH.b
c=VPTISCH.c

```

```

up1()                        ;erstes unterprogramm aufrufen
up2()                        ;zweites unterprogramm aufrufen

```

```

;FOLD          PTP          HOME          Vel=          100          %
DEFAULT;%{PE}%V3.2.0,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME,
3:, 5:100, 7:DEFAULT
$BWDSTART = FALSE
$H_POS=XHOME
PDAT_ACT=PDEFAULT
BAS (#PTP_DAT )
FDAT_ACT=FHOME
BAS (#FRAMES )
BAS (#VEL_PTP,100 )
PTP XHOME
;ENDFOLD
END

```

hauptp.dat

```

&ACCESS RVP
&REL 20
&PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe
&PARAM EDITMASK = *

```

```

DEFDAT hauptp PUBLIC          ;var für sub-prog freigeben

```

```

;FOLD EXTERNAL
DECLARATIONS;%{PE}%V3.2.0,%MKUKATPBASIS,%CEXT,%VCOMMON,%P
;FOLD BAS EXT;%{PE}%V3.2.0,%MKUKATPBASIS,%CEXT,%VEXT,%P
EXT BAS {BAS_COMMAND :IN,REAL :IN }
DECL INT SUCCESS

```

```

;ENDFOLD
;FOLD A10 EXT;%{E}%V3.2.0,%MKUKATPARC,%CEXT,%VEXT,%P
EXT A10 (A_CMD_T:IN,A_STRT_T:IN,A_WELD_T:IN,A_END_T:IN,INT:IN)
;ENDFOLD
;FOLD A20 EXT;%{E}%V3.2.0,%MKUKATPA20,%CEXT,%VEXT,%P
EXT A20 (INT :IN,WELD_ST:IN,WELD_FI:IN,INT:IN)
;ENDFOLD
;FOLD GRIPPER EXT;%{E}%V3.2.0,%MKUKATPGRIPPER,%CEXT,%VEXT,%P
EXT H50 (INT:IN, INT:IN ,INT:IN ,GRP_TYP:IN)
;ENDFOLD
;FOLD SPOT EXT;%{E}%V3.2.0,%MKUKATPSPOT,%CEXT,%VEXT,%P
EXT USERSPOT (S_COMMAND :IN, SPOT_TYPE:IN)
;ENDFOLD
;FOLD TOUCHSENSE EXT;%{E}%V3.2.0,%MKUKATPTS,%CEXT,%VEXT,%P
EXT H70 (INT :IN,SRCH_TYP_2 :OUT,E6POS :IN,SRCH_TYP_3
:IN,SRCH_TYP_2 :IN,SRCH_TYP_2 :IN,SRCH_TYP_2 :IN,SRCH_TYP_2
:IN,SRCH_TYP_2 :IN,INT :IN )
;ENDFOLD
;FOLD USER EXT;%{E}%V3.2.0,%MKUKATPUSER,%CEXT,%VEXT,%P
;Make here your modifications
;ENDFOLD
;ENDFOLD

```

```
GLOBAL REAL x,y,z,a,b,c ;globale var
```

```
ENDDAT
```

up1.sub

```

&ACCESS RV1
&REL 14
&PARAM TEMPLATE = C:\KRC\Roboter\Template\ExpertSubmit
&PARAM EDITMASK = *
DEF up1( )

INT I,J ;var für die for schleife nur in up1
VPTISCH.x=x ;arraywerte einzeln einlesen
VPTISCH.y=y
VPTISCH.z=z
VPTISCH.a=a
VPTISCH.b=b
VPTISCH.c=c

for I = 0 to 2 ;schleife bewegung in x-richtung
for J = 0 to 2 ;bewegung in y-richtung
LIN_REL {z 55}
$OUT[17]=TRUE ;würfel greifen
WAIT SEC 1 ;warten
LIN_REL {z -55}
PTP {A1 2.11, A2 -70.89, A3 90.35, A4 -89.49, A5 92.14, A6 19.56}
PTP {A1 -2.37, A2 -70.5, A3 91.84, A4 -91.04, A5 87.95, A6 21.41}
$OUT[17]=FALSE ;greifer öffnen
WAIT SEC 1 ;warten
PTP {A1 2.11, A2 -70.89, A3 90.35, A4 -89.49, A5 92.14, A6 19.56}
IF J < 2 THEN ;überspringt progteil im letzten
;durchlauf

```

```

y=y -90
VPTISCH.y=y
PTP VPTISCH
ENDIF
ENDFOR ;ende der inneren schleife

```

```

IF I < 2 THEN
y=y+2*90
x=x-115
VPTISCH.x=x
VPTISCH.y=y
PTP VPTISCH
ENDIF
ENDFOR ;ende der Äußeren schleife

```

```

x=VPTISCH.x ;array in var auslesen
y=VPTISCH.y
z=VPTISCH.z
a=VPTISCH.a
b=VPTISCH.b
c=VPTISCH.c

```

```
END
```

up2.sub

```

&ACCESS RV1
&REL 9
&PARAM TEMPLATE = C:\KRC\Roboter\Template\ExpertSubmit
&PARAM EDITMASK = *
DEF up2( )

```

```

INT I,J ;var für die for-schleife nur in up2
VPTISCH.x=x ;arraywerte einzeln einlesen
VPTISCH.y=y ;x,y,z,a,b,c sind GLOBALE var
VPTISCH.z=z
VPTISCH.a=a
VPTISCH.b=b
VPTISCH.c=c

```

```

for I = 0 to 2
for J = 0 to 2

```

```

PTP {A1 -9.9, A2 -102.49, A3 147.33, A4 165.62, A5 45.77, A6 -80.69}
PTP {A1 -26.77, A2 -58.61, A3 139.62, A4 152.65, A5 81.99, A6 -
86.82}
LIN_REL {x -90}
$OUT[17]=TRUE
WAIT SEC 1
LIN_REL {x 90}
PTP {A1 -9.9, A2 -102.49, A3 147.33, A4 165.62, A5 45.77, A6 -80.69}
PTP VPTISCH
LIN_REL {z 55}
$OUT[17]=FALSE
WAIT SEC 1

```

```
LIN_REL (z -55)
```

```
IF J < 2 THEN  
y=y +90  
VPTISCH.y=y  
ENDIF  
ENDFOR
```

```
IF I < 2 THEN  
y=y-2*90  
x=x+115  
VPTISCH.x=x  
VPTISCH.y=y  
ENDIF  
ENDFOR
```

```
END
```

6.4 Anhang D

bewegung_mit_winkel.src

```
&ACCESS RVP
&PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe
&PARAM EDITMASK = *
DEF bewegung_mit_winkel( )
;FOLD INI;{%PE}%V3.2.0,%MKUKATPBASIS,%CINIT,%VCOMMON,%P
;FOLD BAS INI;{%E}%V3.2.0,%MKUKATPBASIS,%CINIT,%VINIT,%P
GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
INTERRUPT ON 3
BAS (#INITMOV,C )
;ENDFOLD (BAS INI)
;FOLD A20 INI;{%E}%V3.2.0,%MKUKATPA20,%CINIT,%VINIT,%P
IF ARC20==TRUE THEN
A20 (ARC_INI)
INTERRUPT DECL 6 WHEN $CYCFLAG[3]==FALSE DO A20 (TECH_STOP2)
ENDIF
;ENDFOLD (A20 INI)
;FOLD A10 INI;{%E}%V3.2.0,%MKUKATPARC,%CINIT,%VINIT,%P
IF A10_OPTION==#ACTIVE THEN
INTERRUPT DECL 4 WHEN $CYCFLAG[2]==FALSE DO A10 (#APPL_ERROR)
INTERRUPT DECL 7 WHEN A_ARC_SWI==#ACTIVE DO A10 (#ARC_SEAM)
INTERRUPT DECL 5 WHEN A_FLY_ARC==TRUE DO A10 (#HPU_ARC)
INTERRUPT ON 5
A10_INI ( )
ENDIF
;ENDFOLD (A10 INI)
;FOLD GRIPPER INI;{%E}%V3.2.0,%MKUKATPGRIPPER,%CINIT,%VINIT,%P
USER_GRP(0,DUMMY,DUMMY,GDEFAULT)
;ENDFOLD (GRIPPER INI)
;FOLD SPOT INI;{%E}%V3.2.0,%MKUKATPSPOT,%CINIT,%VINIT,%P
USERSPOT(#INIT)
;ENDFOLD (SPOT INI)
;FOLD TOUCHSENSE INI;{%E}%V3.2.0,%MKUKATPTS,%CINIT,%VINIT,%P
IF H70_OPTION THEN
INTERRUPT DECL 15 WHEN SMEAS_PULSE[TOUCH_I[TOUCH_ACTIVE].IN_NR] DO
H70 (6,CD0 )
INTERRUPT DECL 16 WHEN $ZERO_MOVE DO H70 (7,CD0 )
INTERRUPT DECL 17 WHEN $TECHPAR_C{FG_TOUCH,8}>0.5 DO H70 (8,CD0)
H70 (1,CD0 )
ENDIF
;ENDFOLD (TOUCHSENSE INI)
;FOLD USER INI;{%E}%V3.2.0,%MKUKATPUSER,%CINIT,%VINIT,%P
;Make your modifications here
;ENDFOLD (USER INI)
;ENDFOLD (INI)
;FOLD PTP HOME Vel= 100 %
DEFAULT;{%PE}%V3.2.0,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME,
3:, 5:100, 7:DEFAULT
$BWDSTART = FALSE
$H_POS=XHOME
PDAT_ACT=PDEFAULT
BAS (#PTP_DAT )
FDAT_ACT=FHOME
```

```

BAS (#FRAMES )
BAS (#VEL_PTP,100 )
PTP XHOME
;ENDFOLD

$VEL_AXIS[5]=100 ;achsengeschwindigkeit festlegen
PTP {AXIS:A2 -90,A3 90,A5 90} ;Arm 90grad anwinkeln

;PTP {AXIS:A1 0,A2 -90,A3 90,A4 0,A5 90,A6 0} ausgeschrieben
;FOLD PTP HOME Vel= 100 %
DEFAULT;#{PE}%V3.2.0,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME,
3:, 5:100, 7:DEFAULT
$BWDSTART = FALSE
$H_POS=XHOME
FOAT_ACT=PDEFAULT
BAS (#PTP DAT )
FDAT_ACT=FHOME
BAS (#FRAMES )
BAS (#VEL_PTP,100 )
PTP XHOME
;ENDFOLD
END

```

bewegungsarten.src

```

&ACCESS RVP
&REL 47
&PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe
&PARAM EDITMASK = *
DEF bewegungsarten( )
;FOLD INI;#{PE}%V3.2.0,%MKUKATPBASIS,%CINIT,%VCOMMON,%P
;FOLD BAS INI;#{E}%V3.2.0,%MKUKATPBASIS,%CINIT,%VINIT,%P
GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
INTERRUPT ON 3
BAS (#INITMOV,0 )
;ENDFOLD (BAS INI)
;FOLD A20 INI;#{E}%V3.2.0,%MKUKATPA20,%CINIT,%VINIT,%P
IF ARC20==TRUE THEN
A20 (ARC_INI)
INTERRUPT DECL 6 WHEN $CYCFLAG[3]==FALSE DO A20(TECH_STOP2)
ENDIF
;ENDFOLD (A20 INI)
;FOLD A10 INI;#{E}%V3.2.0,%MKUKATPARC,%CINIT,%VINIT,%P
IF A10_OPTION==#ACTIVE THEN
INTERRUPT DECL 4 WHEN $CYCFLAG[2]==FALSE DO A10 (#APPL_ERROR)
INTERRUPT DECL 7 WHEN A_ARC_SWI==#ACTIVE DO A10 (#ARC_SEAM)
INTERRUPT DECL 5 WHEN A_FLY_ARC==TRUE DO A10 (#HPU_ARC)
INTERRUPT ON 5
A10_INI ( )
ENDIF
;ENDFOLD (A10 INI)
;FOLD GRIPPER INI;#{E}%V3.2.0,%MKUKATPGRIPPER,%CINIT,%VINIT,%P
USER_GRP(0,DUMMY,DUMMY,GDEFAULT)
;ENDFOLD (GRIPPER INI)
;FOLD SPOT INI;#{E}%V3.2.0,%MKUKATPSPOT,%CINIT,%VINIT,%P

```

```

USERSPOT{#INIT)
;ENDFOLD (SPOT INI)
;FOLD TOUCHSENSE INI;{%E}%V3.2.0,%MKUKATPTS,%CINIT,%VINIT,%P
IF H70_OPTION THEN
INTERRUPT DECL 15 WHEN $MEAS_PULSE[TOUCH_I[TOUCH_ACTIVE].IN_NR] DO
H70 (6,CD0 )
INTERRUPT DECL 16 WHEN $ZERO_MOVE DO H70 (7,CD0 )
INTERRUPT DECL 17 WHEN $TECHPAR_C{FG_TOUCH,8}>0.5 DO H70 (8,CD0)
H70 (1,CD0 )
ENDIF
;ENDFOLD (TOUCHSENSE INI)
;FOLD USER INI;{%E}%V3.2.0,%MKUKATPUSER,%CINIT,%VINIT,%P
;Make your modifications here
;ENDFOLD (USER INI)
;ENDFOLD (INI)
;FOLD PTP HOME Vel= 100 % DEFAULT;{%PE}%R
3.2.42,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME, 3:, 5:100,
7:DEFAULT
$BWDSTART = FALSE
PDAT_ACT=PDEFAULT
BAS(#PTP_DAT)
FDAT_ACT=FHOME
BAS(#FRAMES)
BAS(#VEL_PTP,100)
$H_POS=XHOME
PTP XHOME
;ENDFOLD

;grundbewegungsarten mit Bezugspunkt
PTP_REL {z 700} ;bewegung ungenau
LIN_REL {z -700} ;bewegung linear
PTP_REL {z 350}
CIRC_REL {z -175,x 170},{z -350,x 0};kreisbahn (nicht vom home
punkt)

;grundbewegung mit geteachten koordinaten typ: POS
;FOLD PTP P5 Vel= 100 % PDAT5 Tool[1]:Greifer Base[1];{%PE}%R
3.2.42,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:P5, 3:, 5:100, 7:PDAT5
$BWDSTART = FALSE
PDAT_ACT=PPDAT5
BAS(#PTP_DAT)
FDAT_ACT=FP5
BAS(#FRAMES)
BAS(#VEL_PTP,100)
PTP XP5
;ENDFOLD
PTP {X 48.44, Y 186.88, Z 324.23, A 110.61, B -89.63, C -110.45, S
2, T 35}
LIN {X -135.07, Y -99.68, Z 550.83, A 110.55, B -89.64, C -110.39, S
2, T 34}

CIRC {X -135.08, Y -299.7, Z 350.61, A 110.76, B -89.64, C -110.6, S
2, T 34},{X -135.09, Y -99.7, Z 150.63, A 110.77, B -89.64, C -
110.61, S 2, T 34}
CIRC {X -135.09, Y 99.7, Z 350.64, A 110.61, B -89.64, C -110.45, S
2, T 34},{X -135.09, Y -99.7, Z 550.64, A 110.61, B -89.64, C -
110.45, S 2, T 34}

```

```

;FOLD PTP P6 Vel= 100 % PDAT6 Tool[1]:Greifer Base[1];%{PE}%R
3.2.42,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:P6, 3:, 5:100, 7:PDAT6
$BWDSTART = FALSE
PDAT_ACT=PPDAT6
BAS(#PTP_DAT)
FDAT_ACT=FP6
BAS(#FRAMES)
BAS(#VEL_PTP,100)
PTP XP6
;ENDFOLD

```

;CONT? mit geteachten pos

```

;FOLD PTP P2 CONT Vel= 100 % PDAT2 Tool[1]:Greifer Base[1];%{PE}%R
3.2.42,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:P2, 3:C_PTP, 5:100,
7:PDAT2
$BWDSTART = FALSE
PDAT_ACT=PPDAT2
BAS(#PTP_DAT)
FDAT_ACT=FP2
BAS(#FRAMES)
BAS(#VEL_PTP,100)
PTP XP2 C_PTP
;ENDFOLD

```

;P2 wird nur ungenau angefahren (betriebsart durchlaufen)

```

;FOLD PTP P3 Vel= 100 % PDAT3 Tool[1]:Greifer Base[1];%{PE}%R
3.2.42,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:P3, 3:, 5:100, 7:PDAT3
$BWDSTART = FALSE
PDAT_ACT=PPDAT3
BAS(#PTP_DAT)
FDAT_ACT=FP3
BAS(#FRAMES)
BAS(#VEL_PTP,100)
PTP XP3
;ENDFOLD
;p3 wird genau berührt

```

```

;FOLD PTP HOME Vel= 100 %
DEFAULT;%{PE}%V3.2.0,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME,
3:, 5:100, 7:DEFAULT
$BWDSTART = FALSE
$H_POS=XHOME
PDAT_ACT=PDEFAULT
BAS(#PTP_DAT)
FDAT_ACT=FHOME
BAS(#FRAMES)
BAS(#VEL_PTP,100)
PTP XHOME
;ENDFOLD
END

```

dynamische_position.src

```

&ACCESS RVP
&PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe
&PARAM EDITMASK = *

```

```
DEF dynamische_position( )
```

```
REAL DELTA ;var mit Kommastellen
```

```
;FOLD INI;{%PE}%V3.2.0,%MKUKATPBASIS,%CINIT,%VCOMMON,%P
;FOLD BAS INI;{%E}%V3.2.0,%MKUKATPBASIS,%CINIT,%VINIT,%P
GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
INTERRUPT ON 3
BAS (#INITMOV,0 )
;ENDFOLD (BAS INI)
;FOLD A20 INI;{%E}%V3.2.0,%MKUKATPA20,%CINIT,%VINIT,%P
IF ARC20==TRUE THEN
A20 (ARC_INI)
INTERRUPT DECL 6 WHEN $CYCFLAG[3]==FALSE DO A20(TECH_STOP2)
ENDIF
;ENDFOLD (A20 INI)
;FOLD A10 INI;{%E}%V3.2.0,%MKUKATPARC,%CINIT,%VINIT,%P
IF A10_OPTION==#ACTIVE THEN
INTERRUPT DECL 4 WHEN $CYCFLAG[2]==FALSE DO A10 (#APPL_ERROR)
INTERRUPT DECL 7 WHEN A_ARC_SWI==#ACTIVE DO A10 (#ARC_SEAM)
INTERRUPT DECL 5 WHEN A_FLY_ARC==TRUE DO A10 (#HPU_ARC)
INTERRUPT ON 5
A10_INI ( )
ENDIF
;ENDFOLD (A10 INI)
;FOLD GRIPPER INI;{%E}%V3.2.0,%MKUKATPGRIPPER,%CINIT,%VINIT,%P
USER_GRP(0,DUMMY,DUMMY,GDEFAULT)
;ENDFOLD (GRIPPER INI)
;FOLD SPOT INI;{%E}%V3.2.0,%MKUKATPSPOT,%CINIT,%VINIT,%P
USERSPOT(#INIT)
;ENDFOLD (SPOT INI)
;FOLD TOUCHSENSE INI;{%E}%V3.2.0,%MKUKATPTS,%CINIT,%VINIT,%P
IF H70_OPTION THEN
INTERRUPT DECL 15 WHEN SMEAS_PULSE[TOUCH_I[TOUCH_ACTIVE].IN_NR] DO
H70 (6,CD0 )
INTERRUPT DECL 16 WHEN $ZERO_MOVE DO H70 (7,CD0 )
INTERRUPT DECL 17 WHEN $TECHPAR_C[FG_TOUCH,8]>0.5 DO H70 (8,CD0)
H70 (1,CD0 )
ENDIF
;ENDFOLD (TOUCHSENSE INI)
;FOLD USER INI;{%E}%V3.2.0,%MKUKATPUSER,%CINIT,%VINIT,%P
;Make your modifications here
;ENDFOLD (USER INI)
;ENDFOLD (INI)
;FOLD PTP HOME Vel= 100 %
DEFAULT{%PE}%V3.2.0,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME, 3:,
5:100, 7:DEFAULT
$BWDSTART = FALSE
$H_POS=XHOME
PDAT_ACT=PDEFAULT
BAS (#PTP_DAT )
FDAT_ACT=FHOME
BAS (#FRAMES )
BAS (#VEL_PTP,100 )
PTP XHOME
;ENDFOLD
```

```

PTP {AXIS:A2 -90,A3 90,A5 90} ;achse2 -90grad
PTP_REL {x -291,y -105,z 570} ;pos über ersten Würfel
Pl=$POS_ACT ;aktuelle pos speichern
DELTA=Pl.z ;z-koordinate in var
DELTA=DELTA-70 ;70mm abgezogen
Pl.z=DELTA ;neuen z-wert in Pl einlesen
PTP Pl ;bewegung zur errechneten pos

```

```

;FOLD PTP HOME Vel= 100 $
DEFAULT;%{PE}%V3.2.0,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME,
3:, 5:100, 7:DEFAULT
$BWDSTART = FALSE
$H_POS=XHOME
PDAT_ACT=PDEFAULT
BAS (#PTP_DAT )
FDAT_ACT=FHOME
BAS (#FRAMES )
BAS (#VEL_PTP,100 )
PTP XHOME
;ENDFOLD
END

```

schleifen.src

```

&ACCESS RVP
&REL 1
&PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe
&PARAM EDITMASK = *
DEF schleifen( )

DECL INT I,J,K ;var für die for-schleife decl
J=1
K=1

;FOLD INI;%{PE}%V3.2.0,%MKUKATPBASIS,%CINIT,%VCOMMON,%P
;FOLD BAS INI;%{E}%V3.2.0,%MKUKATPBASIS,%CINIT,%VINIT,%P
GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
INTERRUPT ON 3
BAS (#INITMOV,0 )
;ENDFOLD (BAS INI)
;FOLD A20 INI;%{E}%V3.2.0,%MKUKATPA20,%CINIT,%VINIT,%P
IF ARC20==TRUE THEN
A20 (ARC_INI)
INTERRUPT DECL 6 WHEN $CYCFLAG[3]==FALSE DO A20(TECH_STOP2)
ENDIF
;ENDFOLD (A20 INI)
;FOLD A10 INI;%{E}%V3.2.0,%MKUKATPARC,%CINIT,%VINIT,%P
IF A10_OPTION==#ACTIVE THEN
INTERRUPT DECL 4 WHEN $CYCFLAG[2]==FALSE DO A10 (#APPL_ERROR)
INTERRUPT DECL 7 WHEN A_ARC_SWI==#ACTIVE DO A10 (#ARC_SEAM)
INTERRUPT DECL 5 WHEN A_FLY_ARC==TRUE DO A10 (#HPU_ARC)
INTERRUPT ON 5
A10_INI ( )
ENDIF
;ENDFOLD (A10 INI)

```

```

;FOLD GRIPPER INI;%(E)%V3.2.0,%MKUKATPGRIPPER,%CINIT,%VINIT,%P
USER_GRP(0,DUMMY,DUMMY,GDEFAULT)
;ENDFOLD (GRIPPER INI)
;FOLD SPOT INI;%(E)%V3.2.0,%MKUKATPSPOT,%CINIT,%VINIT,%P
USERSPOT(#INIT)
;ENDFOLD (SPOT INI)
;FOLD TOUCHSENSE INI;%(E)%V3.2.0,%MKUKATPTS,%CINIT,%VINIT,%P
IF H70_OPTION THEN
INTERRUPT DECL 15 WHEN $MEAS_PULSE[TOUCH_I[TOUCH_ACTIVE].IN_NNR] DO
H70 (6,CD0 )
INTERRUPT DECL 16 WHEN $ZERO_MOVE DO H70 (7,CD0 )
INTERRUPT DECL 17 WHEN $TECHPAR_C[FG_TOUCH,8]>0.5 DO H70 (8,CD0)
H70 (1,CD0 )
ENDIF
;ENDFOLD (TOUCHSENSE INI)
;FOLD USER INI;%(E)%V3.2.0,%MKUKATPUSER,%CINIT,%VINIT,%P
;Make your modifications here
;ENDFOLD (USER INI)
;ENDFOLD (INI)
;FOLD      PTP      HOME      Vel=      100      %
DEFAULT%(PE)%V3.2.0,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME, 3:,
5:100, 7:DEFAULT
$BWDSTART = FALSE
$H_POS=XHOME
PDAT_ACT=PDEFAULT
BAS (#PTP_DAT )
FDAT_ACT=FHOME
BAS (#FRAMES )
BAS (#VEL_PTP,100 )
PTP XHOME
;ENDFOLD

FOR I = 0 TO 4 STEP 1      ;5mal leerzeichen beachten!
WAIT SEC 1      ;wartet 1 sek
;FOLD      OUT      17      ''      State=      TRUE      CONT;%(PE)%R
3.2.42,%MKUKATPBASIS,%COUT,%VOUTX,%P 2:17, 5:TRUE, 6:CONTINUE
CONTINUE
$OUT[17]=TRUE
;ENDFOLD
;FOLD WAIT Time= 1 sec;%(PE)%R 3.2.42,%MKUKATPBASIS,%CWAIT,%VWAIT,%P
2:1
WAIT SEC 1
;ENDFOLD
;FOLD      OUT      17      ''      State=      FALSE      CONT;%(PE)%R
3.2.42,%MKUKATPBASIS,%COUT,%VOUTX,%P 2:17, 5:FALSE, 6:CONTINUE
CONTINUE
$OUT[17]=FALSE
;ENDFOLD
;FOLD WAIT Time= 1 sec;%(PE)%R 3.2.42,%MKUKATPBASIS,%CWAIT,%VWAIT,%P
2:1
WAIT SEC 1
;ENDFOLD
ENDFOR      ;schleifenende

WAIT SEC 10

WHILE J<3      ;bedingung (2mal)

```

```

$OUT[17]=TRUE
WAIT SEC 1
$OUT[17]=FALSE
WAIT SEC 1
J=J+1
ENDWHILE

```

```
WAIT SEC 10
```

```

REPEAT                                     ;3mal
$OUT[17]=TRUE
WAIT SEC 1
$OUT[17]=FALSE
WAIT SEC 1
K=K+1
UNTIL K==4                                ;bedingung

```

```

;FOLD          PTP          HOME          Vel=          100          %
DEFAULT;%(PE)%V3.2.0,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME,
3:, 5:100, 7:DEFAULT
$BWDSTART = FALSE
$H_POS=XHOME
PDAT_ACT=PDEFAULT
BAS (#PTP_DAT )
FDAT_ACT=FHOME
BAS (#FRAMES )
BAS (#VEL_PTP,100 )
PTP XHOME
;ENDFOLD
END

```

tisch_ausrichten.src

```

&ACCESS RVP
&REL 10
&PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe
&PARAM EDITMASK = *
DEF tisch_ausrichten( )
;FOLD INI;%(PE)%V3.2.0,%MKUKATPBASIS,%CINIT,%VCOMMON,%P
;FOLD BAS INI;%(E)%V3.2.0,%MKUKATPBASIS,%CINIT,%VINIT,%P
GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
INTERRUPT ON 3
BAS (#INITMOV,0 )
;ENDFOLD (BAS INI)
;FOLD A20 INI;%(E)%V3.2.0,%MKUKATPA20,%CINIT,%VINIT,%P
IF ARC20==TRUE THEN
A20 (ARC_INI)
INTERRUPT DECL 6 WHEN $CYCFLAG[3]==FALSE DO A20(TECH_STOP2)
ENDIF
;ENDFOLD (A20 INI)
;FOLD A10 INI;%(E)%V3.2.0,%MKUKATPARC,%CINIT,%VINIT,%P
IF A10_OPTION==#ACTIVE THEN
INTERRUPT DECL 4 WHEN $CYCFLAG[2]==FALSE DO A10 (#APPL_ERROR)
INTERRUPT DECL 7 WHEN A_ARC_SWI==#ACTIVE DO A10 (#ARC_SEAM)
INTERRUPT DECL 5 WHEN A_FLY_ARC==TRUE DO A10 (#HPG_ARC)

```

```

INTERRUPT ON 5
A10_INI ( )
ENDIF
;ENDFOLD (A10 INI)
;FOLD GRIPPER INI;%{E}%V3.2.0,%MKUKATPGRIPPER,%CINIT,%VINIT,%P
USER_GRP(0,DUMMY,DUMMY,GDEFAULT)
;ENDFOLD (GRIPPER INI)
;FOLD SPOT INI;%{E}%V3.2.0,%MKUKATPSPOT,%CINIT,%VINIT,%P
USERSPOT(#INIT)
;ENDFOLD (SPOT INI)
;FOLD TOUCHSENSE INI;%{E}%V3.2.0,%MKUKATPTS,%CINIT,%VINIT,%P
IF H70_OPTION THEN
INTERRUPT DECL 15 WHEN $MEAS_PULSE{TOUCH_I{TOUCH_ACTIVE}.IN_NR} DO
H70 (6,CD0 )
INTERRUPT DECL 16 WHEN $ZERO_MOVE DO H70 (7,CD0 )
INTERRUPT DECL 17 WHEN $TECHPAR_C[FG_TOUCH,8]>0.5 DO H70 (8,CD0)
H70 (1,CD0 )
ENDIF
;ENDFOLD (TOUCHSENSE INI)
;FOLD USER INI;%{E}%V3.2.0,%MKUKATPUSER,%CINIT,%VINIT,%P
;Make your modifications here
;ENDFOLD (USER INI)
;ENDFOLD (INI)
;FOLD PTP HOME Vel= 100 %
DEFAULT;%{PE}%V3.2.0,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME,
3:, 5:100, 7:DEFAULT
$BWDSTART = FALSE
$H_POS=XHOME
PDAT_ACT=PDEFAULT
BAS (#PTP_DAT )
FDAT_ACT=FHOME
BAS (#FRAMES )
BAS (#VEL_PTP,100 )
PTP XHOME
;ENDFOLD

PTP {AXIS: A2 -90,A3 90,A5 90}
PTP_REL {Z 750}
PTP_REL {x -150}
LIN_REL {y -300}
LIN_REL {y 600}
PTP_REL {x 200}

;FOLD PTP HOME Vel= 100 % DEFAULT;%{PE}%R
3.2.42,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME, 3:, 5:100,
7:DEFAULT
$BWDSTART = FALSE
PDAT_ACT=PDEFAULT
BAS (#PTP_DAT)
FDAT_ACT=FHOME
BAS (#FRAMES)
BAS (#VEL_PTP,100)
$H_POS=XHOME
PTP XHOME
;ENDFOLD
END

```

Unterprogramm und globale Variable

testh.src

```
&ACCESS RVP
&REL 5
&PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe
&PARAM EDITMASK = *
DEF testh( )

EXT testu( ) ;unterprogramm bekannt geben

;FOLD INI;%(PE)%V3.2.0,%MKUKATPBASIS,%CINIT,%VCOMMON,%P
;FOLD BAS INI;%(E)%V3.2.0,%MKUKATPBASIS,%CINIT,%VINIT,%P
GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
INTERRUPT ON 3
BAS (#INITMOV,0 )
;ENDFOLD (BAS INI)
;FOLD A20 INI;%(E)%V3.2.0,%MKUKATPA20,%CINIT,%VINIT,%P
IF ARC20==TRUE THEN
A20 (ARC_INI)
INTERRUPT DECL 6 WHEN $CYCFLAG[3]==FALSE DO A20(TECH_STOP2)
ENDIF
;ENDFOLD (A20 INI)
;FOLD A10 INI;%(E)%V3.2.0,%MKUKATPARC,%CINIT,%VINIT,%P
IF A10_OPTION==#ACTIVE THEN
INTERRUPT DECL 4 WHEN $CYCFLAG[2]==FALSE DO A10 (#APPL_ERROR)
INTERRUPT DECL 7 WHEN A_ARC_SWI==#ACTIVE DO A10 (#ARC_SEAM)
INTERRUPT DECL 5 WHEN A_FLY_ARC==TRUE DO A10 (#HPU_ARC)
INTERRUPT ON 5
A10_INI ( )
ENDIF
;ENDFOLD (A10 INI)
;FOLD GRIPPER INI;%(E)%V3.2.0,%MKUKATPGRIPPER,%CINIT,%VINIT,%P
USER_GRP(0,DUMMY,DUMMY,GDEFAULT)
;ENDFOLD (GRIPPER INI)
;FOLD SPOT INI;%(E)%V3.2.0,%MKUKATPSPOT,%CINIT,%VINIT,%P
USERSPOT(#INIT)
;ENDFOLD (SPOT INI)
;FOLD TOUCHSENSE INI;%(E)%V3.2.0,%MKUKATPTS,%CINIT,%VINIT,%P
IF H70_OPTION THEN
INTERRUPT DECL 15 WHEN $MEAS_PULSE[TOUCH_I[TOUCH_ACTIVE].IN_MR] DO
H70 (6,CD0 )
INTERRUPT DECL 16 WHEN $ZERO_MOVE DO H70 (7,CD0 )
INTERRUPT DECL 17 WHEN $TECHPAR_C[EG_TOUCH,8]>0.5 DO H70 (8,CD0)
H70 (1,CD0 )
ENDIF
;ENDFOLD (TOUCHSENSE INI)
;FOLD USER INI;%(E)%V3.2.0,%MKUKATPUSER,%CINIT,%VINIT,%P
;Make your modifications here
;ENDFOLD (USER INI)
;ENDFOLD (INI)
;FOLD PTP HOME Vel= 100
DEFAULT;%(PE)%V3.2.0,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME,
3:, 5:100, 7:DEFAULT
$BWDSTART = FALSE
$H_POS=XHOME
```

```

PDAT_ACT=PDEFAULT
BAS (#PTP_DAT )
FDAT_ACT=FHOME
BAS (#FRAMES )
BAS (#VEL_PTP,100 )
PTP XHOME
;ENDFOLD

```

```

I = 23 ;var I ist GLOBAL in testh.dat
testu() ;aufruf des unterprogrammes

```

```

;FOLD PTP HOME Vel= 100 3
DEFAULT;#{PE}%V3.2.0,%MKUKATPBASIS,%CMOVE,%VPTP,%P 1:PTP, 2:HOME,
3:, 5:100, 7:DEFAULT
$BWDSTART = FALSE
$H_POS=XHOME
PDAT_ACT=PDEFAULT
BAS (#PTP_DAT )
FDAT_ACT=FHOME
BAS (#FRAMES )
BAS (#VEL_PTP,100 )
PTP XHOME
;ENDFOLD
END

```

testh.dat

```

&ACCESS RVP
&REL 5
&PARAM TEMPLATE = C:\KRC\Roboter\Template\vorgabe
&PARAM EDITMASK = *
DEFDAT testh PUBLIC
;FOLD EXTERNAL
DECLARATIONS;#{PE}%V3.2.0,%MKUKATPBASIS,%CEXT,%VCOMMON,%P
;FOLD BAS EXT;#{PE}%V3.2.0,%MKUKATPBASIS,%CEXT,%VEXT,%P
EXT BAS (BAS_COMMAND :IN,REAL :IN )
DECL INT SUCCESS
;ENDFOLD
;FOLD A10 EXT;#{E}%V3.2.0,%MKUKATPARC,%CEXT,%VEXT,%P
EXT A10 (A_CMD_T:IN,A_STRT_T:IN,A_WELD_T:IN,A_END_T:IN,INT:IN)
;ENDFOLD
;FOLD A20 EXT;#{E}%V3.2.0,%MKUKATPA20,%CEXT,%VEXT,%P
EXT A20 (INT :IN,WELD_ST:IN,WELD_FI:IN,INT:IN)
;ENDFOLD
;FOLD GRIPPER EXT;#{E}%V3.2.0,%MKUKATPGRIPPER,%CEXT,%VEXT,%P
EXT H50 (INT:IN, INT:IN ,INT:IN ,GRP_TYP:IN)
;ENDFOLD
;FOLD SPOT EXT;#{E}%V3.2.0,%MKUKATPSPOT,%CEXT,%VEXT,%P
EXT USERSPOT ($_COMMAND :IN, SPOT_TYPE:IN)
;ENDFOLD
;FOLD TOUCHSENSE EXT;#{E}%V3.2.0,%MKUKATPTS,%CEXT,%VEXT,%P
EXT H70 (INT :IN,SRCH_TYP_2 :OUT,E6POS :IN,SRCH_TYP_3
:IN,SRCH_TYP_2 :IN,SRCH_TYP_2 :IN,SRCH_TYP_2 :IN,SRCH_TYP_2
:IN,SRCH_TYP_2 :IN,INT :IN )
;ENDFOLD

```

```

;FOLD USER EXT;%(E)%V3.2.0,%MKUKATPUSER,%CEXT,%VEXT,%P
;Make here your modifications
;ENDFOLD
;ENDFOLD

```

Global INT I

```
ENDDAT
```

testu.sub

```

&ACCESS RV01
&REL 3
&PARAM TEMPLATE = C:\KRC\Roboter\Template\submit
&PARAM EDITMASK = *
DEF testu( )
;FOLD DECLARATIONS;%(PE)%3.2.0,%MKUKATPBASIS,%CINIT,%VCOMMON,%P
;FOLD USER DECL;%(PE)%3.2.0,%MKUKATPBASIS,%CINIT,%VCOMMON,%P
; Please insert user defined declarations
;ENDFOLD (USER DECL)
;ENDFOLD (DECLARATIONS)
;FOLD INI;%(PE)%3.2.0,%MKUKATPBASIS,%CINIT,%VCOMMON,%P
;FOLD USER INIT;%(E)%3.2.0,%MKUKATPBASIS,%CSPS,%VAUTOEXT,%P
; Please insert user defined initialization commands
;ENDFOLD (USER INIT)
;ENDFOLD (INI)

IF I==23 THEN
PTP_REL {x 200}
ENDIF

;FOLD ;%(H);
END
;ENDFOLD
;FOLD USER SUBROUTINE;%(H)%V3.2.0,%MKUKATPUSER,%CSPS,%VSPS,%P
;Integrate your user defined subroutines
;ENDFOLD (USER SUBROUTINE)

```

Fachbereich Maschinenbau

Fachgebiet Allgemeine Maschinenlehre, Getriebe-
technik, Dynamik, Maschinendynamik

Fachgebietsleiter Prof. Dr.-Ing. Werner Laumann

Praktikum Maschinenlehre

Versuch ML 1

Inbetriebnahme und Programmierung eines KUKA KR 30-2

Aufgabe

Nehmen Sie anhand der am Roboter ausliegenden Versuchsanleitung den Roboter in Betrieb und justieren Sie die 6 Achsen des Roboters unter Verwendung des Feinzeigers und der manuellen Steuerungskonsole.

Erstellen Sie ein einfaches Programm, in dem die Grundfunktionen des Roboters vorkommen und stellen Sie dies dem Betreuer vor, verwenden Sie hier ebenfalls die Versuchsanleitung.

Erstellen Sie ein Programm mit folgender Funktionalität:

Die 9 auf dem Tisch befindlichen Würfel werden, unter Verwendung von Unterprogrammen und Schleifen, in das am Tisch befestigte Aluminiumprofil befördert. Anschließend werden die Würfel am unteren Ende des Profils entnommen und zurück auf den Tisch sortiert.

Versuchsvorbereitung

Die Programmiersprache des KUKA ist der Hochsprache BASIC angelehnt, machen Sie sich mit den grundlegenden Befehlen vor Beginn des Versuches vertraut.

Kolloquium (Schlagworte)

- Kinematik
- Koordinatentransformation
- Gelenkketten
- Freiheitsgrade
- Robotik

Eidesstattliche Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Textstellen, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, sind als solche gekennzeichnet.

Die Arbeit wurde weder in der vorliegenden, noch in einer vergleichbaren Form anderweitig vorgelegt.

Mühlhausen 18.3.2009

Zeugnis für das Praktikum

Schüler: **Michél Eberhardt**, geb.: 06.10.1984
Zeit: 02.02. bis 27.02 2009
Ort: Fachhochschule Jena, Fachbereich Maschinenbau

Im Maschinenlabor des Fachbereiches Maschinenbau war Herr Eberhardt im Rahmen seiner Tätigkeit mit folgender Aufgabe beauftragt:

Entwurf und Inbetriebnahme eines Praktikumsversuches für das Labor Maschinenlehre unter Verwendung des vorhandenen KUKA Roboters. Folgende Teilaufgaben waren während des Praktikums zu lösen:

- Entwurf des Versuches
- Erarbeiten der Versuchsanleitung
- Programmierung des Roboters unter Verwendung der Maschinenkoordinatensysteme
- ausführlicher Test des Versuches
- Dokumentation aller notwendigen Arbeitsschritte
- Übergabe des Gesamtaufbaus an den Betreuer

Herr Eberhardt erledigte die ihm übertragene Aufgabe selbstständig, wobei er strukturiert und methodisch vorgeht. Die sehr gute Auffassungsgabe und die Fähigkeit logische Zusammenhänge schnell zu erfassen, konnte Herr Eberhardt bei der Lösung der Aufgabe erfolgreich anwenden.

Insgesamt sind die Leistungen von Herrn Eberhardt mit „Sehr Gut“ bzw. der Note 1 zu bewerten.

Jena, den 27.02.2009



Betreuer während des Praktikums: Dipl.-Ing.(FH) Andreas Riess



Verantwortlicher Hochschullehrer: Prof. Dr.-Ing. Werner Laumann

Fachhochschule Jena
UNIVERSITY OF APPLIED SCIENCES
Fachbereich Maschinenbau
Carl-Zeiss - Promenade 2
07745 Jena