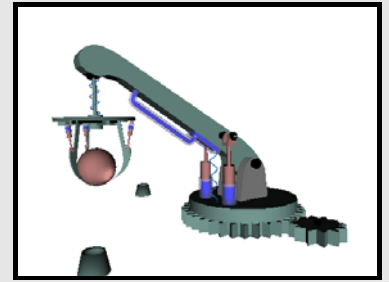


Der programmierbare Roboterarm IV

Programm von J. Tiburski

Einführung in die Automatenprogrammierung



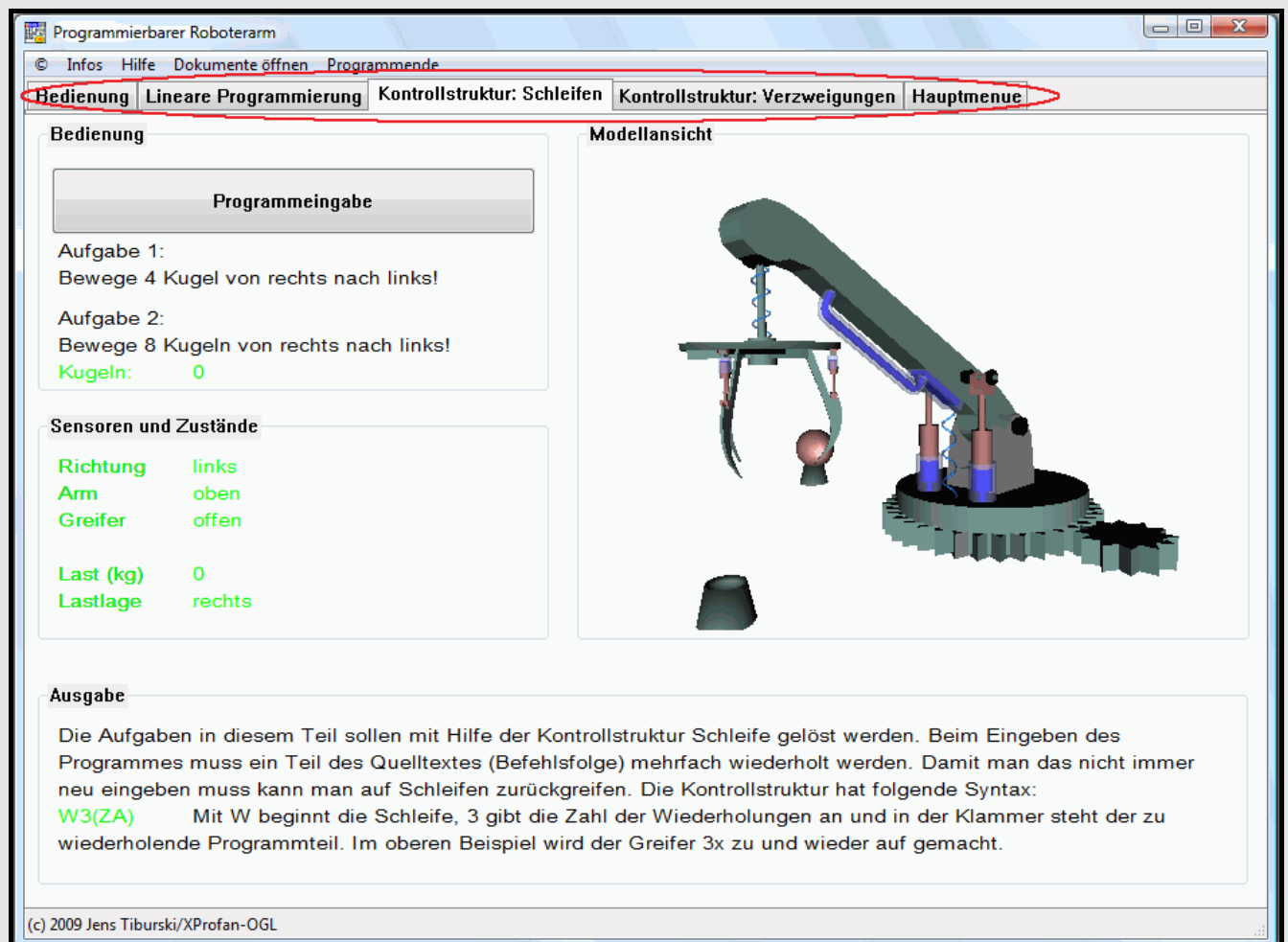
Der programmierbare Roboterarm – strukturiertes Programmieren

Beim linearen Programmieren galt bisher: Ein Programm für den programmierbaren Roboterarm besteht also aus einer Reihe von Großbuchstaben die von links nach rechts abgearbeitet werden.

Beim strukturierten Programmieren kommen nun jedoch Schleifen und Verzweigungen mit hinzu. Da Schleifen und Verzweigungen den linearen Ablauf des Programms widersprechen spricht man nicht mehr von **linearer Programmierung** sondern von **strukturierter Programmierung**. Im Teil 4 des Programms Roboterarm werden nacheinander die folgenden Kapitel behandelt:

1. **Bedienung** des Roboterarms über die Button
2. **lineares Programmieren** mit Kugeltransport
3. Kontrollstruktur **Schleifen**
4. Kontrollstruktur **bedingte Verzweigungen**

Die einzelnen Kapitel werden im Programm Roboterarm über die Reiter aktiviert:



Der programmierbare Roboterarm – Bedienung des Roboterarms über die Button

Die bereits aus dem **Teil 1** bekannten **Programmierbefehle** werden beibehalten - auch hier ist während der Bedienung darauf zu achten, dass die Anzeige der Sensoren berücksichtigt wird. Wenn z.B. der Richtungs-Sensor bereits **links** anzeigt kann der Befehl **L** (**Roboterarm nach links schwenken**) nicht mehr ausgeführt werden. Das Programm gibt eine Fehlermeldung aus.

Hinzu kommt nun die Last in Form einer Kugel die vom rechten Lager auf das linke Lager befördert werden soll. Jede der Kugeln hat die Masse 300 kg. Sobald der Greifer um die Kugel geschlossen wird zeigt der Last-Sensor die aktuelle Last an. Wenn die Kugel mittels Roboterarm vom rechten Lager auf das linke Lager befördert wurde kann der Kugeltausch (neuer Befehl der Programmiersprache K) durchgeführt werden.

Merke:

Für den problemlosen Kugeltausch müssen zwei Bedingungen erfüllt sein. Dazu sind zwei neue Sensoren hinzugekommen die die Last betreffen. Der Roboterarm verfügt nun über einen Last-Sensor der die Masse der Last in kg ermittelt. Der zweite Sensor ist der Lastlage-Sensor der über die aktuelle Lage der Kugel informiert.

- Die Kugel muss vom rechten Lager auf das linke Lager befördert worden sein.
(Lastlage-Sensor: **links**)
- Die Kugel muss freigegeben sein.
(Roboterarm-Sensor: **oben**)

Bedienung | **Lineare Programmierung** | **Kontrollstruktur: Schleifen** | **Kontrollstruktur: Verzweigungen** | **Hauptmenue**

Bedienung

L (nach links drehen) | R (nach rechts drehen)

H (Arm hoch) | T (Arm tiefer)

Z (Greifer zu) | A (Greifer auf)

K (neue Kugel)

Sensoren und Zustände

Richtung	links
Arm	oben
Greifer	offen
Last (kg)	0
Lastlage	rechts

Ausgabe

Ähnlich wie bei Teil 1 dieses Programms soll der Roboterarm über die Befehlseingabe per Buttons gesteuert werden. Dabei sollen die Kugeln vom rechten Lager in das linke Lager bewegt werden. Alle Kugeln haben die Masse 300 kg - die Obergrenze der Belastung des Roboterarms. Der Austausch der Kugeln K (für den Befehl: Kugelwechsel) kann nur erfolgen, wenn die Last sich auf dem linken Lager befindet. Der Roboterarm muss die Kugel freigegeben haben und nach oben geschwenkt worden sein.

Modellansicht

Der programmierbare Roboterarm – lineare Programmierung des Roboterarms

Solange immer nur ein Methodenaufruf nach dem anderen ausgeführt werden kann spricht man von **linearer Programmierung**. An dieser Stelle stehen die folgenden „Programmierbefehle“ zur Verfügung:

- R – Roboterarm nach rechts schwenken.
- L – Roboterarm nach links schwenken.
- H – Roboterarm nach oben (hoch) schwenken.
- T – Roboterarm nach unten (tiefer) schwenken.
- A – Greifer öffnen (auf).
- Z – Greifer schließen (zu).
- K – Kugelwechsel.

Aus diesen Befehlen ist das gewünschte Programm zusammenzusetzen. Die oben vereinbarten Bedingungen zur Beachtung der Sensorzustände sowie zum Kugeltausch müssen auch hier wieder eingehalten werden.

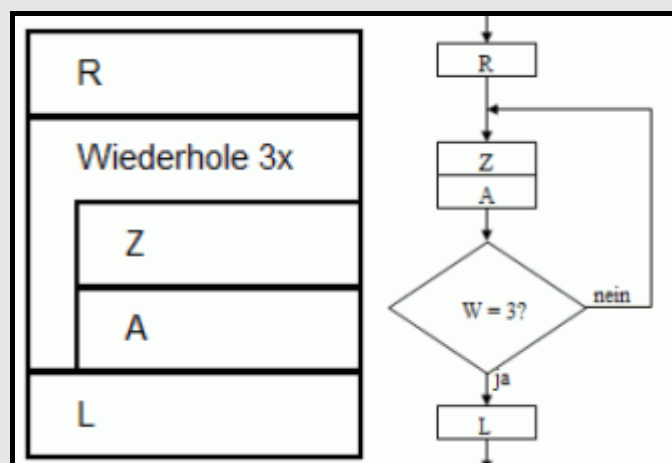
Man kann nun versuchen, die Kugel von rechts nach links zu befördern und anschließend auszutauschen oder von rechts nach links und wieder zurück von links nach rechts (Wie im kleinen Intro des Programms Roboterarm!).

Der programmierbare Roboterarm - Wiederholungen

Der Abschnitt „Wiederholungen“ bearbeitet zum ersten Mal Programmstrukturen, die von der linearen Programmierung abweichen. Wenn z.B. nicht nur eine Kugel bewegt werden soll sondern eine beliebige Anzahl, so ist es – bisher – unabwendbar, einen bestimmten Programmteil mehrfach zu wiederholen. Das kann man eine **Schleife** erledigen lassen! Eine Schleife im Programm bezeichnet einen Programmteil, der mehrfach nacheinander ausgeführt werden kann. Der Roboterarm erkennt die Kontrollstruktur Schleife an der Syntax **W3(ZA)**. Das W kennzeichnet die Struktur als Wiederholung, die darauf folgende Ziffer gibt die Anzahl der Wiederholungen an. Die Befehle in der Klammer sind der zu wiederholende Programmteil. Beim Programmablauf würde **W3(ZA)** dasselbe ergeben wie **ZAZAZA**. Allerdings kann man bei vielen Wiederholungen oder größeren Schleifen Befehle sparen.

Schematische Darstellung einer Schleife für die Befehlsfolge: **RW3(ZA)L**

(Struktogramm und Programmablaufplan der Sequenz)



Es dürfen mehrere Schleifen **nacheinander** - aber **nicht geschachtelt** - eingegeben werden!

Der programmierbare Roboterarm – bedingte Verzweigungen

Mit den „Wiederholungen“ kennt man bereits eine Kontrollstruktur, die das Verzeigen eines Programms ermöglicht. Eine echte Verzweigung jedoch - die sich aus der unterschiedlichen Beantwortung einer Bedingung mit mehreren Alternativen ergibt - ist das noch nicht.

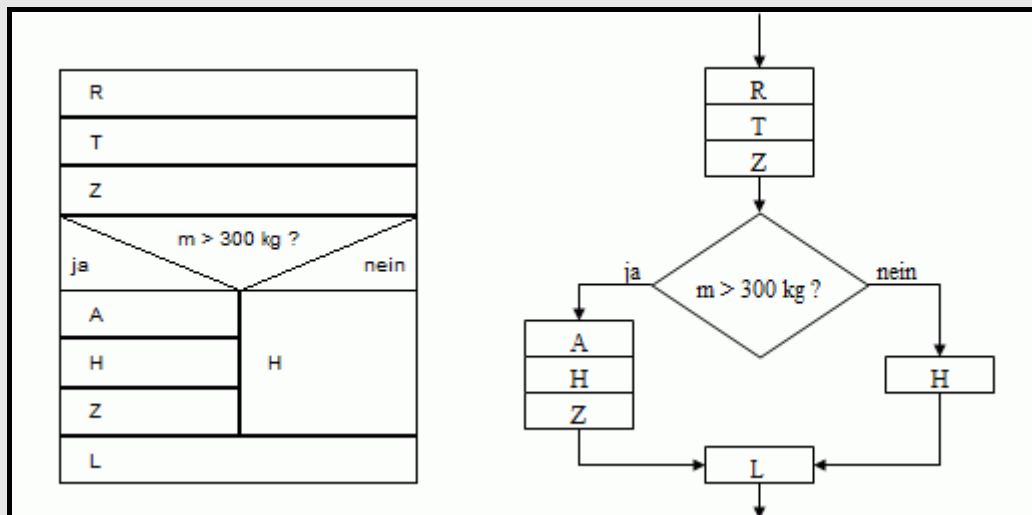
Im Folgenden werden dem Roboterarm Kugeln mit unterschiedlichen Lasten (0kg-500kg) vorgelegt. Die Masse von 300 kg ist dabei die Obergrenze die der Roboterarm heben kann ohne Schaden zu nehmen. Sobald der Roboterarm eine Kugel im Greifer hat (Richtung: **links**, Arm: **unten**, Greifer: **zu**) wechselt die Lastposition auf den Zustand **im_Greifer** und nun muss der Test erfolgen, ob die Last vom Roboterarm bewegt werden kann oder nicht. Der Roboterarm erkennt die Kontrollstruktur **Verzweigung** an der Syntax **I<erste Befehlsfolge/zweite Befehlsfolge>**. Der Großbuchstabe I steht für das englische *if ... then*. Die erste Befehlsfolge wird ausgeführt, wenn die Kugel die erlaubte Masse von höchstens 300 kg hat. Bei der anderen Alternative ($m > 300 \text{ kg}$) wird die zweite Befehlsfolge ausgeführt. Es ist darauf zu achten, dass der Roboterarm am Ende des Tests bei beiden Alternativen dieselben Sensoren-Zustände hat, damit das Programm anschließend fehlerfrei weiter ausgeführt werden kann.

Für die Bearbeitung der Aufgabe 2 (Transport von 5 Kugeln) kann der Test problemlos in die Schleife eingebaut werden – es darf aber nur **ein Test** im Programm durchgeführt werden.

Wenn die Kugel zu schwer ist muss sie rechts liegen bleiben bis der Roboterarm wieder nach oben geschwenkt ist. Nun kann die Kugel auch in der rechten Position getauscht werden und der Kugeltausch wird für die Erfüllung der Aufgabe mitgezählt.

Schematische Darstellung der Verzweigung für die Befehlsfolge: **RTZI<H/AHZ>L**

(Struktogramm und Programmablaufplan der Sequenz)



Sensoren-Zustände vor der Verzweigung: Richtung: **rechts** Arm: **unten** Greifer: **zu**
Sensoren-Zustände nach der Verzweigung: Richtung: **rechts** Arm: **oben** Greifer: **zu**

Ich wünsche viel Erfolg bei der Programmierung des Roboterarms und hoffe, dass dieses Programm beim Einstieg in die Welt der Algorithmen geholfen hat – oder einfach nur Spaß gemacht hat.

Jens Tiburski