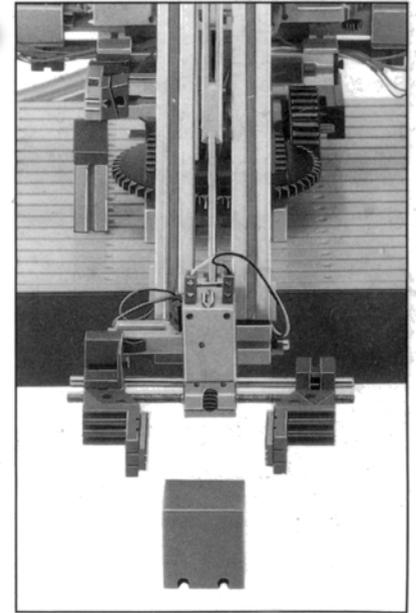


fischertechnik[®] [®] COMPUTING

Bauanleitung Trainingsroboter · Instructions Training Robot · Mode d'emploi du robot d'entraînement



Inhalt

Einführung	3
Was ist ein Roboter?	4
Antriebs- und Positioniersystem	5
Interface und Software	6
Roboter-Systemprogramm	8
Erste Experimente	9
Aufbau des Roboters	10
Steuerung des Roboters	11
Teach-In Verfahren	14
Weitere Experimente	15
Abdruck der Programme	16
Funktionsweise des Interface und Roboter-Systemprogramms	21
Bebilderte Bauanleitung	61
Kabelkonfektionierung	62
Verdrahtungsplan Gabellichtschanke	63
Mechanischer Aufbau	64
Verdrahtungsplan Trainingsroboter	91

Table of Contents

Introduction	33
What is a Robot ?	34
The Driving and Positioning System	35
Interface and Software	36
The Robot System Program	38
First Experiments	39
Assembly of the Robot	40
Control of the Robot	41
Teach-in Procedure	44
Further Experiments	45
Print out of the Programs	46
Operation of the Interface and the Robot System Program	51
Illustrated Assembly Instructions	61
Ribbon Cable Configuration	62
Circuit Layout Photo-Interrupter	63
Mechanical Assembly	64
Circuit Layout Training Robot	91

fischertechnik Trainingsroboter

Lieber fischertechnik-Freund,

kaum ein technisches Instrument läßt sich so vielfältig einsetzen, wie ein Computer. Eines der reizvollsten Gebiete der Computertechnik ist jedoch die Steuerung technischer Modelle. Mit dem fischertechnik computing Bausatz Trainingsroboter haben Sie ein Modell erworben, das Sie in das anspruchsvollste Gebiet der Steuerungstechnik, die Robotik, einführt.

Der Bausatz verbindet verschiedene Forderungen in einem leistungsfähigen Instrument. Zu allererst soll der Roboter realistisch sein. Aus diesem Grund hat der Trainingsroboter keinerlei Ähnlichkeit mit Science-fiction-Robotern, jenen menschenähnlichen Gestalten in Metall. Vielmehr lagen Konstruktionsskizzen heutiger Industrieroboter dem Entwurf zugrunde. Sie können aus dem Bausatz das Modell eines dreiachsigen Industrieroboters mit rotatorischen Bewegungsachsen aufbauen. Was dies genau bedeutet, werden Sie in dem Anleitungsbuch noch lesen.

Zum zweiten sollen Sie nicht nur den Roboter aufbauen und mit Ihrem Heim- oder Personal-Computer steuern können. Sie sollen auch die Möglichkeit haben, die Abläufe zu verstehen. Daher sind die Pro-

gramme so ausgelegt, daß sie zum überwiegenden Teil in BASIC geschrieben und reichlich dokumentiert wurden. Das Verständnis der Programme ist auch die Voraussetzung, eigene Ideen und Vorstellungen zu realisieren. Ich möchte behaupten, daß der Nutzen des Roboters an dieser Stelle zuvörderst zum Tragen kommt.

Ändern, Experimentieren, Ausbauen, Programmieren... hier spielt Ihre eigene Kreativität die entscheidende Rolle.

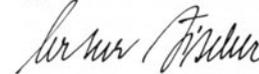
Dieser dritten Anforderung an den Roboter kommt vom mechanischen Aufbau her das fischertechnik-System in idealer Weise entgegen. Gleichgültig, ob Sie die Funktionen der Greifhand ausbauen, den Roboter mit Sensoren versehen oder eine Umgebung zur Computer-integrierten Fertigung aufbauen – die Vielfalt und hohe Präzision der fischertechnik Bauelemente wird Ihnen in allen Bereichen weiterhelfen.

Zur hohen Funktionszuverlässigkeit des Trainingsroboters tragen auch neue Bauelemente bei. An erster Stelle ist hier das Antriebs- und Positioniersystem zu erwähnen. Neue kompakte Gleichstrommotoren mit hoher Leistung verhelfen dem Roboter zu schneller Bewegung. Doch ist es mit der Steuerung des Roboters alleine nicht getan. Das Programm im

Computer muß die Möglichkeit haben, die einmal erreichte Stellung abzufragen. Hierzu dienen die eigens entwickelten Gabellichtschranken. Sie durchleuchten mit Infrarotlicht das mit der Antriebsachse verbundene Rad. Die dort angebrachte Segmentierung erzeugt bei Bewegung letztlich Impulse, die über das fischertechnik Interface dem Computer zugeführt werden. Um diese schnelle Impulsfolge verwerten zu können, wird ein speziell für den Trainingsroboter entwickeltes Softwarepaket eingesetzt. Dieser Teil ist in der Maschinensprache Ihres Computers geschrieben und kann ohne Zählerverluste selbst die gleichzeitige Bewegung aller Roboter-gelenke überwachen. Diese Programme werden von BASIC aufgerufen und sind somit problemlos zu benutzen.

Ich bin sicher, daß der fischertechnik computing Trainingsroboter Sie zu einer Reihe eigener Experimente anregen und Ihr Wissen und Ihre Erfahrung auf diesem Gebiete erheblich erweitern wird.

Ihr



Was ist ein Roboter?

Der Wunsch Roboter zu bauen, ist schier so alt wie die Menschheit. Wie so oft in der Geschichte unserer Kultur finden wir die Wurzeln des Roboters auch schon im antiken Griechenland. Mechanische Tempeldiener wurden dort erstellt, die vorgegebene Bewegungen ausführen konnten. Gesteuert wurden diese durch das Gewicht des herabrieselnden Sandes einer großen, im Roboter eingebauten, Sanduhr. Mit der Blütezeit der Handwerkskunst in der Neuzeit erregten auch immer wieder mechanisch gesteuerte Puppen die Bewunderung der Zeitgenossen und lassen auch uns Zeuge des Geschicks ihrer Erbauer werden. Ein schachspielender Automat zeigte jedoch die Grenzen der damaligen Möglichkeiten: In ihm verbarg sich ein kleinwüchsiger Mensch, der geschickt über Hebel und Gestänge die schachspielende Puppe steuerte. Unser Jahrhundert hat erst die Wortschöpfung „Roboter“ erlebt. In dem Bühnenstück R.U.R. (Rossum's Universal Robot) des tschechischen Autors Karel Capek wurden Maschinensklaven „Roboter“ genannt; abgeleitet von dem slawischen Wort „robota“ für „schwer arbeiten“. In dieser gedanklichen Linie bleibt auch häufig unsere heutige Vorstellung eines Roboters, von einer Maschine mit menschenähnlicher Gestalt, mittlerweile nicht mehr mit Sanduhr oder Federwerk sondern mit einem Computer als Gehirn ausgestattet. Genährt wird diese Vorstellung durch eine vielfältige Science-fiction-Literatur.

Daneben erleben wir in der Praxis unserer Arbeitswelt eine andere Art von Roboter. Es handelt sich hier um „... universell einsetzbare Bewegungsautomaten mit mehreren Achsen, deren Bewegung hinsichtlich Bewegungsfolge und -wegen bzw. -winkeln frei programmierbar und gegebenenfalls sensorgeführt sind“ (VDI-Richtlinie). Roboter können je nach Ausstattung z.B. mit Greifern Werkstücke umsetzen, mit Punktschweißzange oder Lackierpistole Fertigungsaufgaben durchführen, aber auch eine Vielzahl anderer Handreichungen vornehmen.

Trotz der nüchternen Sichtweise der industriellen Praxis kommt doch auch bei diesen Instrumenten die Menschenähnlichkeit wieder ins Spiel. Gerade die universellsten der Roboter lehnen sich an das Vorbild der Natur an. Ihr Bewegungsapparat gleicht häufig einem menschlichen Arm. Deutlich sind vier Teilstücke des Roboters auszumachen: Die Grundplatte entspricht dem Körper, darauf folgt der Oberarm, an den sich nach dem Ellenbogengelenk der Unterarm anschließt. Den Abschluß bildet die Greifhand. Wir werden in der Folge diese menschenbezogenen Begriffe benutzen, um Ihnen eine schnelle Orientierung zu ermöglichen.

Wir wollen nicht verhehlen, daß auch andere Bauformen von Industrierobotern üblich sind. In beliebiger Kombination können die Drehachsen des Roboters auch durch Verschiebungen längs einer Führung ersetzt werden. Erfolgen im Extremfall alle Bewegungen längs einer Bahn, so erhalten wir ein Gerät, das mehr einem Portalkran als einem Arm entspricht. Dieser Robotertyp heißt dann auch Portalroboter und ist meist besser für hohe Lasten geeignet. Auch bei diesem Robotertyp werden die Verschiebemöglichkeiten als Bewegungsachsen bezeichnet, obwohl eine Drehachse im strengen Sinne nicht vorliegt.

Den Robotern ist gemeinsam, daß zur Positionierung der Greifhand immer drei Bewegungsachsen erforderlich sind. Bei einem Knickarmroboter ist dies erst nach längerem Ausprobieren einzusehen. Dagegen leuchtet diese Tatsache bei dem Portalroboter sofort ein. Er kann sich längs der drei Raumdimensionen Höhe, Breite und Tiefe bewegen.

Sie werden vielleicht schon gehört haben, daß Industrieroboter fünf und mehr Achsen haben. In seltenen Fällen kommt diese größere Zahl von Achsen der Bewegungsapparatur zugute, z.B. um in die Ecke einer Autokarosserie hineingreifen zu können. Meist dienen die vierte und fünfte Achse dem Schwenken des Handgelenks. Auf diese Weise kann

die Griffrichtung verändert werden. Diese Möglichkeiten finden Sie beim fischertechnik Trainingsroboter nicht eingebaut, er enthält nur die drei Hauptachsen zur Bewegung. Allerdings stellt dies kein Nachteil dar, da alle Studien zur Robotergeometrie auch an den drei Hauptbewegungsachsen angestellt werden können. Zudem ist die Greifhand mit einem mechanischen Lageausgleich ausgestattet. Vielleicht ist es aber ein für Sie interessantes Unterfangen, die zusätzlichen Achsen zur Orientierung der Greifhand noch anzubauen? Doch zunächst soll das Grundmodell erstellt werden.

Das Antriebs- und Positioniersystem

Zum Antrieb des Roboters dienen Gleichstrommotoren. Die drei größeren Motoren bewirken die Roboterbewegung, der kleinere Motor das Öffnen und Schließen der Greifzange. Gleichstrommotoren haben den Vorteil, daß sie einfach anzusteuern sind und ein hohes Drehmoment bei geringem Eigen-gewicht und -volumen erbringen. Sie tragen letztlich zu der schnellen Bewegung des Roboters bei. Wir können die Motoren vor dem Einbau testen, indem wir sie direkt an das Netzgerät anschließen. Da im nachfolgenden Betrieb des Roboters bis zu vier Motoren gleichzeitig im Betrieb sein können, muß ein entsprechend belastbares Netzgerät verwendet werden. Wir empfehlen das fischertechnik computing Netzgerät oder aber die Verwendung von zwei fischertechnik Netzgeräten mot 4.

Gleichstrommotoren teilen mit den meisten anderen Motoren aber auch einen Nachteil. Sie können zwar über ein Getriebe den Roboter bewegen, jedoch ist nach Durchführung der Bewegung die Stellung des Roboters nur ungefähr bekannt. Lediglich die Laufdauer, während der der Motor angeschaltet ist, kann man über den Computer steuern. Andere Faktoren, wie die genaue Netzspannung und damit die Spannung des Netzgeräts, die Leichtgängigkeit des Roboters und der Lastwechsel an der Greifhand sind Faktoren, die zu einer Änderung der Motorgeschwindigkeit führen. Daher müssen gleiche Schalt-dauern nicht unbedingt immer zu gleichen Bewegungen des Roboters führen. Ein solcher Zustand ist jedoch für ein Präzisionsinstrument wie einen Roboter untragbar. Daher muß unabhängig vom Motor die Position des Roboters noch einmal gemessen werden. Das hierzu dienende Positioniersystem besteht bei dem Trainingsroboter aus Gabellichtschranken. Bild 1 zeigt, wie eine solche Gabellichtschranke mit dem Motor und dem Getriebe verbunden ist. Die Abtriebswelle des Getriebes trägt ein becherförmiges Rad, auf dessen Umfang in regel-

mäßigem Abstand schwarze Linien aufgedruckt sind. Insgesamt sind es 32 Linien.

Die Gabellichtschranke greift nun über den Rand des Bechers. Auf der einen Seite der Gabel befindet sich eine Leuchtdiode, die Infrarotlicht sendet. Auf der anderen Seite der Gabel ist ein Fototransistor eingebaut, der auf Infrarotlicht empfindlich ist. Befindet sich kein Gegenstand zwischen der Gabel, so erscheint, vorausgesetzt daß die Betriebsspannung der Lichtschranke richtig angeschlossen ist (s.u.), an dem mit \perp bezeichneten Ausgang ein High-Signal. Wird dagegen ein lichtundurchlässiger Gegenstand zwischen die Zinken der Gabel eingeschoben, so ist der Lichtstrahl unterbrochen. An dem Ausgang erscheint ein Low-Signal. Genauso reagiert die Lichtschranke auch auf die verschiedenen Zonen des Rades. Die schwarz bedruckte Stelle unterbricht den Lichtstrahl, die nicht bedruckten Stellen schwächen zwar das Infrarotlicht, lassen aber eine ausreichende Menge durch.

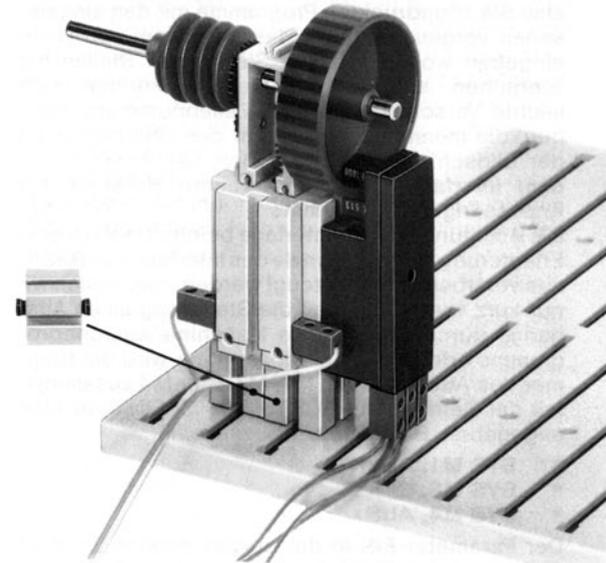
Wenn der Motor läuft, so erscheint an dem Ausgang der Lichtschranke eine Impulsfolge, die genau den dunklen und hellen Stellen des Rades entspricht. Nun sind wir in der Lage, den Roboter mit solch einem Aggregat präzise zu steuern. Wir messen nicht die Laufdauer des Motors, sondern zählen die Impulswechsel am Ausgang der Lichtschranke. Diese Zahl ist ein exaktes Maß für die Umdrehung der Abtriebsachse und damit für die Stellung des Roboters. Allerdings erfordert das Zählen der Impulse eine hohe Arbeitsgeschwindigkeit des Computers. Eine solche Aufgabe muß direkt in der Maschinensprache des Computers programmiert werden, wie weiter unten noch erläutert wird.

Im Moment wollen wir erste Erfahrungen mit dem Positioniersystem sammeln. Wie oben geschildert, muß der Infrarotstrahl das Plastikmaterial durchleuchten, nicht aber den schwarzen Aufdruck. Auch Sonnenlicht und das Licht von Neonröhren enthält Infrarotanteile, die natürlich störend wirken. Wir

müssen daher den optimalen Arbeitspunkt der Lichtschranke suchen. Hierzu kann ihre Empfindlichkeit eingestellt werden. Mit dem beige-fügten Schraubendreher können Sie durch das Loch in der Gehäuseoberseite ein Potentiometer verstellen. Gehen Sie dabei aber mit Vorsicht und Gefühl voran, um die elektronischen Bauteile nicht zu beschädigen.

Als Einstellhilfe finden Sie auf der Diskette bzw. Kassette das Programm ROBOT.JUST. Zu seiner Benutzung ist das fischertechnik Interface notwendig, so daß wir an dieser Stelle uns zunächst mit diesem wichtigen Instrument vertraut machen wollen.

Bild 1



Interface und Software

An dieser Stelle wollen wir eine kurze Bemerkung zu der Dokumentation der Programme bei fischertechnik computing einfließen lassen. Die Programme sind in dem Anleitungsheft in der Schreibweise des Commodore 64 abgedruckt. Mit dem Interface, das zu Ihrem Computer paßt, wird eine Diskette oder Kassette mitgeliefert, auf der die Programme auch vorliegen. Die BASIC-Schreibweisen der verschiedenen Computer unterscheiden sich leicht. Wenn Sie keinen Commodore 64, sondern einen anderen Computer haben, wird das Programm auf der Diskette oder der Kassette nicht ganz identisch mit dem hier abgedruckten Programm sein. Es ist schon an den entsprechenden Computertyp angepaßt. Die Stellen, wo sich auf jeden Fall Abweichungen ergeben, sind in dem Abdruck des Programms mit einem Sternchen vor der Zeile gekennzeichnet. Wenn Sie also die abgedruckten Programme mit den eingeleiteten verglichen oder das Programm von Hand eingeben wollen, müssen Sie an den Stellen mit Sternchen aufpassen. Außerdem können sich leichte Verschiebungen der Zeilennummern ergeben, die meist ihre Ursache in den Unterschieden der Bildschirmsteuerung haben. Die Anleitung zu dem Interface gibt Ihnen weitere Hinweise zur Anpassung der Programme.

Die Anleitung zu dem Interface beinhaltet auch eine Erläuterung, wie die Signale des Interface von BASIC aus verarbeitet bzw. erzeugt werden. Hier wollen wir nur kurz vermerken, daß die Steuerung eines Ausgangs durch Aufruf eines Maschinenspracheprogramms erfolgt. Als Aufrufparameter wird die Nummer des Ausgangs M1, M2, M3 oder M4 zusammen mit der Betriebsart RECHTS, LINKS, EIN oder AUS angegeben. Beispiele sind:

* **SYS M1, RECHTS**

* **SYS M3, EIN**

* **SYS M4, AUS**

Der Parameter EIN in der obigen zweiten Zeile ist übrigens gleichbedeutend mit dem Parameter

RECHTS. Als allererstes muß jedoch immer der Befehl

* **SYS INIT**

erfolgen, der das Interface in einen Anfangszustand versetzt. Dabei werden alle Motoren ausgeschaltet, so daß dieser Befehl auch zum gleichzeitigen Abschalten der Motoren dient.

Die Eingänge des Interface werden mit der USR-Funktion erfaßt. Mit den Parametern E1, E2 bis E8 werden die acht Eingänge abgefragt, an die die mini-Taster angeschlossen werden. Auch andere Ein-Aus-Signale können dort eingespeist werden. Die Funktionen USR(EX) und USR(EY) hingegen dienen der Eingabe stufenlos veränderlicher elektrischer Werte. Diese Eingänge werden bei dem hier beschriebenen Roboter nicht benötigt. Sie können aber wichtig werden, wenn Sie Sensoren anschließen wollen, z.B. Fotowiderstände zur Objekterkennung.

Wichtig zu wissen ist auch, daß das Interface eine Überwachungsschaltung des Datenverkehrs besitzt. Immer wenn innerhalb einer halben Sekunde kein neuer Befehl, sei es ein Ausgabe- oder Eingabebefehl, kommt, schaltet es alle Motoren ab. Beim Stoppen des Computerprogramms brauchen Sie daher nicht eigens die Stromversorgung der Motoren abzustellen. Setzt der Datenaustausch wieder ein, nimmt das Interface alle Motoren wie zuletzt wieder in Betrieb.

Das Maschinenspracheprogramm, das den Datenaustausch zwischen Computer und Interface bewirkt, muß natürlich auch in dem Computer abgespeichert sein. Hierzu dient das sogenannte Grundprogramm, das sich ebenfalls auf der Diskette oder Kassette befindet. Gleichzeitig ist es Bestandteil eines jeden weiteren fischertechnik computing Programms und belegt die Zeilennummer 1 bis 500. In den Programmlisten dieses Anleitungsbuches erscheint dieser Teil jedoch nicht, da er für jeden Computertyp anders aussieht. Das Maschinenpro-

gramm muß ganz detailliert auf den Hard- und Softwareaufbau des Computers eingehen. Sie finden das Grundprogramm in der Anleitung zu Ihrem Interface dokumentiert.

Der Trainingsroboter verwendet aufgrund des oben beschriebenen Positioniersystems ein erweitertes Grundprogramm, das Roboter-Systemprogramm. Es liegt auf der Diskette bzw. Kassette als ROBOT.SYSTEM vor. Es kommen acht weitere Kommandos hinzu: Ganz ähnlich zu den Ausgabekommandos SYS M1, RECHTS oder SYS M4, AUS verhält sich das Kommando

* **SYS P1, nnnn**

Dieses Kommando läßt den Motor M1 solange laufen, bis der Roboter die Position nnnn erreicht hat. nnnn ist dabei eine positive Ganzzahl und gibt den Stand des oben erwähnten Impulszählers wieder. Das Roboter-Systemprogramm „weiß“ also immer den jeweiligen Zählerstand. Wenn mit dem Kommando eine neue Position angefordert wird, so berechnet das Roboter-Systemprogramm die Differenz der Positionen. Aus dem absoluten Zahlenwert ergibt sich die Anzahl der abzuwartenden Impulse von der Lichtschranke, aus dem Vorzeichen die Laufrichtung des Motors. Allerdings rührt sich der Motor noch nicht. Als Programmierer haben Sie nämlich die Gelegenheit, für weitere Motoren die neue Position anzufordern. Erst wenn alle Aufträge an das Roboter-Systemprogramm ausgeteilt sind, wird das Kommando

* **SYS ROBOT**

aufgerufen. Nun laufen alle Motoren, die eine neue Position erreichen sollen, gleichzeitig los. Von allen in Betrieb befindlichen Motoren werden die Impulselingänge überwacht und ausgezählt. Jeder Motor wird ausgeschaltet, wenn er die neue Position erreicht hat. Sind alle Positionen erreicht, gibt das Kommando die Kontrolle wieder an das BASIC-Programm zurück.

Noch einige weitere Bemerkungen zu dem Roboter-Systemprogramm sind angebracht. Ein Motor wird auch schon vor Erreichen der Position abgeschaltet, wenn der dem Motor zugeordnete Endtaster angesprochen hat. Diese Endtaster haben einerseits die Funktion, eine „Heimposition“ des Roboters festzulegen, andererseits unzulässige Bewegungen des Roboters zu vermeiden. Die Endtaster müssen im nicht betätigten Zustand geschlossen sein. Sie öffnen den Kontakt bei Betätigung. Bei den dem Modell beigefügten mini-Tastern sind somit meist die Kontaktnummern 1 und 2 anzuschließen.

Weiter hatten wir geschildert, daß bei Erreichen der Zielposition der Motor einer jeder Bewegungsachse abgeschaltet wird. Wer sich jedoch schon genauer mit der motorischen Steuerung befaßt hat, wird wissen, daß der Motor nach dem Abschalten noch etwas nachläuft. Aus diesem Grund überwacht das Roboter-Systemprogramm den Impulseingang noch für eine festgelegte Zeit nach Abschalten des Motors. Jeder dann noch eintreffende Impuls wird noch gezählt. Letztlich wird sich also nicht die Zielposition ergeben, sondern eine klein wenig danebenliegende Position. Diese tatsächlich erreichte Position kann durch die Funktion

*** USR(P1) (dto. für P2, P3 und P4)**

abgefragt werden.

Die den Motoren zugeordneten Positionszähler können auch auf Null gestellt werden. Dies erfolgt mit dem Kommando

*** SYS INIT**

das Sie schon aus dem einfacheren Grundprogramm kennen. Da es im Rahmen des Roboter-Systemprogramms noch diese Nebenwirkung hat, kann es nicht mehr bedenkenlos zum gleichzeitigen Abschalten aller Motoren verwendet werden. Allerdings wird sich diese Notwendigkeit auch gar nicht mehr ergeben, da die Motoren der zuverlässigen

Kontrolle des Roboter-Systemprogramms unterliegen.

Im Gegensatz zu den einfacheren Kommandos des Grundprogramms sind in den Kommandos des Roboter-Systemprogramms Ein- und Ausgaben miteinander verwoben, um so die hohe Leistungsfähigkeit zu erzielen. Dies erfordert aber auch eine feste Zuordnung von Ein- und Ausgängen des Interface, wie die nachstehende Tabelle zeigt:

Motor	End- taster	Impuls- eingang	Kommandos
M1	E1	E2	SYS P1, nnnn USR(P1)
M2	E3	E4	SYS P2, nnnn USR(P2)
M3	E5	E6	SYS P3, nnnn USR(P3)
M4	E7	E8	SYS P4, nnnn USR(P4)

Wie aus der Tabelle zu sehen ist, sind die Roboter-Systemkommandos für alle vier Ausgänge des Interface vorhanden, wenn auch der Trainingsroboter das Positioniersystem nur für drei Motoren benutzt. In diesem Fall werden Motor 1 bis 3 über die Positionierkommandos gesteuert. Der Motor 4 treibt die Greifzange an und wird mit den einfacheren Kommandos des Grundprogramms gesteuert.

Sollten Sie nicht mit dem fischertechnik computing Interface arbeiten, sondern mit einer anderen Interfaceschaltung, gilt das bisher Gesagte natürlich nicht in jedem Detail. Dennoch können Sie die hier skizzierten Ideen auch auf jeder anderen Hardware realisieren.

Roboter-Systemprogramm

Nachfolgend ist das BASIC-Programm wiedergegeben, das das Roboter-Systemprogramm für den Commodore 64 erzeugt. Dieses Programm sowie die ab Seite 16 noch abgedruckten Programme können auch von der fischertechnik Diskette Trainingsroboter/Plotter/Scanner geladen werden. Dies gilt auch für die entsprechenden Programme für andere Computer. Fordern Sie die Diskette unter Angabe des Typs Ihres Computers und Laufwerks bitte bei

fischerwerke Artur Fischer GmbH & Co. KG
Abt. fischertechnik
7244 Tumlingen/Waldachtal

an. Sie müssen hierzu den beigefügten Gutschein verwenden.

```
* 1 PRINT CHR$(147):POKE 53280,3:POKE 53281,1
* 2 FOR I=1 TO 7
* 3 PRINT
* 4 NEXT
* 5 PRINT TAB(6);"BITTE EINEN MOMENT WARTEN"
* 6 PRINT
* 7 PRINT TAB(11);"TRAININGSROBOTER"
* 8 PRINT
* 9 PRINT TAB(5);"SYSTEMPROGRAMM WIRD GELADEN"
* 10 REM ROBOTER SYSTEMPROGRAMM FUER COMMODORE 64
* 13 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1985
* 15 REM AUFRUF DES PROGRAMMS MIT
* 20 REM SYS M1,EIN SYS M1,AUS
* 25 REM SYS M1,LINKS SYS M1,RECHTS
* 30 REM USR(E1) USR(EX) USR(EY)
* 35 REM SPEZIELLE ROBOTERBEFEHLE
* 40 REM SYS P1,NNNN SOLLPOSITION ABLEGEN
* 45 REM USR(P1) ISTPOSITION ABFRAGEN
* 50 REM SYS ROBOT START DES ROBOTERS
* 55 DATA 52656,0,0,169,0,162,23,157,208,53375
* 60 DATA 207,202,16,250,48,46,169,3,54316
* 65 DATA 208,10,169,12,208,6,169,48,55146
* 70 DATA 208,2,169,192,120,141,177,205,56360
* 75 DATA 32,253,174,173,176,205,13,177,57563
* 80 DATA 205,141,176,205,32,158,183,138,58801
* 85 DATA 45,177,205,141,177,205,173,176,60100
* 90 DATA 205,77,177,205,32,241,205,88,61330
* 95 DATA 96,141,176,205,72,169,63,141,62393
* 100 DATA 3,221,162,8,169,48,14,176,63194
* 105 DATA 205,144,2,9,4,141,1,221,63921
* 110 DATA 9,8,141,1,221,202,208,236,64947
* 115 DATA 169,57,141,1,221,104,141,176,65957
* 120 DATA 205,96,120,32,247,183,201,0,67041
* 125 DATA 208,118,192,162,240,57,192,146,68356
* 130 DATA 240,53,140,177,205,32,61,206,69470
* 135 DATA 45,177,205,168,240,2,160,1,70468
* 140 DATA 32,162,179,88,96,169,50,141,71385
* 145 DATA 1,221,9,8,141,1,221,162,72149
* 150 DATA 8,10,44,1,221,16,2,9,72460
* 155 DATA 1,160,48,140,1,221,160,56,73247
* 160 DATA 140,1,221,202,208,235,96,169,74519
* 165 DATA 255,141,4,221,141,5,221,169,75676
* 170 DATA 185,141,14,221,140,1,221,160,76759
* 175 DATA 58,140,1,221,173,4,221,162,77739
* 180 DATA 3,202,208,253,56,237,4,221,78923
* 185 DATA 208,242,162,56,142,1,221,56,80011
* 190 DATA 169,255,237,4,221,168,169,255,81489
* 195 DATA 237,5,221,32,145,178,88,96,82482
* 200 DATA 162,8,192,194,240,18,162,2,83462
* 205 DATA 192,198,240,12,162,4,192,202,84664
* 210 DATA 240,6,162,6,192,206,208,11,85695
* 215 DATA 188,208,207,189,209,207,32,145,87080
* 220 DATA 179,88,96,140,176,205,141,177,88282
* 225 DATA 205,96,169,0,240,10,169,2,89173
* 230 DATA 208,6,169,4,208,2,169,6,89945
* 235 DATA 141,177,205,32,253,174,32,138,91097
* 240 DATA 173,32,247,183,174,177,205,157,92445
* 245 DATA 217,207,152,157,216,207,96,162,93859
* 250 DATA 6,56,189,208,207,253,216,207,95201
* 255 DATA 141,240,207,189,209,207,253,217,96864
* 260 DATA 207,141,241,207,16,5,189,201,98071
* 265 DATA 207,208,11,173,240,207,13,241,99371
* 270 DATA 207,240,3,189,208,207,157,224,100799
* 275 DATA 207,157,232,207,202,202,16,209,102230
* 280 DATA 32,61,206,141,241,207,32,61,103211
* 285 DATA 206,168,77,241,207,141,240,207,104698
* 290 DATA 140,241,207,169,0,141,176,205,105977
* 295 DATA 162,6,173,241,207,61,201,207,107235
* 300 DATA 208,6,157,224,207,157,225,207,108626
* 305 DATA 173,240,207,61,200,207,240,76,110030
* 310 DATA 189,232,207,221,200,207,208,20,111514
* 315 DATA 56,189,208,207,233,1,157,208,112773
* 320 DATA 207,189,209,207,233,0,157,209,114184
* 325 DATA 207,56,176,17,24,189,208,207,115268
* 330 DATA 105,1,157,208,207,189,209,207,116551
* 335 DATA 105,0,157,209,207,189,208,207,117833
* 340 DATA 221,216,207,208,23,189,209,207,119313
* 345 DATA 221,217,207,208,15,169,0,221,120571
* 350 DATA 224,207,240,8,157,224,207,169,122007
* 355 DATA 255,157,225,207,169,0,221,225,123466
* 360 DATA 207,240,3,222,225,207,173,176,124919
* 365 DATA 205,29,224,207,141,176,205,202,126308
* 370 DATA 202,16,135,173,176,205,32,241,127488
* 375 DATA 205,240,3,76,30,207,162,6,128417
* 380 DATA 29,225,207,202,202,16,249,201,129748
* 385 DATA 0,240,3,76,30,207,88,96,130488
* 390 DATA 2,1,8,4,32,16,128,64,130743
* 395 DATA 1,2,4,8,16,32,64,128,130998
* 400 DATA 162,146,255,170,85,85,26,206,132133
* 405 DATA 52930,52934,52938,52942,52967,396844
* 410 READ INIT: M1=INIT
* 415 FOR M3=0 TO 67: FOR M2=0 TO 7
* 420 READ M4: POKE INIT+M3*8+M2,M4
* 425 M1=M1+M4: NEXT
* 430 READ M4: IF M1<>M4 THEN PRINT"DATAFEHLER
IN ZEILE":M3*5+55:PRINT M1:END
* 435 NEXT
* 440 READ E1,E2,E3,E4,E5,E6,E7,E8
* 445 M1=M1+E1+E2+E3+E4+E5+E6+E7+E8
* 450 READ M4: IF M1<>M4 THEN PRINT"DATAFEHLER
IN ZEILE 395":PRINT M1:END
* 455 READ EX,EY,AUS,LINKS,RECHTS,EIN,M2,M3
* 460 M1=M1+EX+EY+AUS+LINKS+RECHTS+EIN+M2+M3
* 465 READ M4: IF M1<>M4 THEN PRINT"DATAFEHLER
IN ZEILE 400":PRINT M1:END
* 470 POKE 785,M2:POKE 786,M3
* 475 READ P1,P2,P3,P4,ROBOT
* 480 M1=M1+P1+P2+P3+P4+ROBOT
* 485 READ M4: IF M1<>M4 THEN PRINT"DATAFEHLER
IN ZEILE 405":PRINT M1:END
* 490 INIT=INIT+2
* 495 M1=INIT+12: M2=M1+4: M3=M2+4: M4=M3+4
* 500 SYS INIT
```

Erste Experimente

Nach soviel Theorie zur Programmierung wollen wir nun die ersten Experimente durchführen. Wir nehmen den zuvor erstellten Versuchsaufbau zur Hand. Motor und Lichtschranke müssen mit Hilfe des beigefügten 20-adrigen Kabels mit dem Interface verbunden werden. Das Kabel sollten wir jedoch gleich so herrichten, daß wir es nachher in Form eines Kabelbaums in den Roboter einziehen können. Der Zuschnitt ist auf S. 62 gezeigt. Aus dem abgeschnittenen Teil wird einerseits ein weiterer Kabelbaum angefertigt. Zum anderen werden die nun noch verbleibenden Kabelreste in Einzeladern auseinandergezogen. Diese Kabel dienen zum weiteren Verdrahten des Modells, insbesondere um die gemeinsame Masse- und +5V-Leitung an alle Taster und Lichtschranken zu bringen.

Insgesamt vier Leitungen werden bei dem Aufbau des Trainingsroboters nicht benutzt. Sie sind den Eingängen EX, EY und E8 zugeordnet. Wenn Sie sich jetzt schon gedanklich mit Ausbauplänen für Ihren Roboter befassen, so sollten Sie diese Leitungen nicht abschneiden, sondern in voller Länge belassen, aufwickeln und z. B. mit Klebestreifen unter dem Hauptkabelbaum fixieren. Die Kabelenden werden nun vorsichtig auf eine Länge von ca. 3 bis 5 mm abisoliert, ohne die feinen Adern der Litze zu beschädigen. Anschließend werden die Adern verdreht. Mit dem Anschrauben der Stecker sollten Sie noch warten, bis Sie das Kabel in den Roboter eingezogen haben.

Für unseren ersten Test genügt es, wenn der Ausgang M1, der Eingang E1 und E2 sowie die Leitung +5V mit Steckern versorgt wird. Welche Kabeladern dies sind, geht aus dem Verdrahtungsplan sowie der Deckelbeschriftung des Interface hervor. Biegen Sie die Litze auf die Isolation um. Lösen Sie das Schraubchen des Steckers und führen Sie das Kabelende in die Hülse ein. Danach wird die Schraube wieder angezogen, aber nicht so fest, daß das Kabel abgequetscht wird. Außerdem richten Sie

sich aus den Kabelabschnitten zwei Kabel, die an beiden Enden mit fischertechnik-Steckern versehen werden.

Verbinden Sie nun den Motor mit den beiden zugehörigen Kabeln orange und gelb. Die Buchse mit dem \oplus -Symbol der Lichtschranke wird mit der roten +5V-Leitung verbunden. Außerdem wird die +5V-Leitung weitergeführt an den Kontakt 2 eines mini-Tasters. Die Buchse mit dem \ominus -Symbol der Lichtschranke wird mit der getrennten Massebuchse des Interface verbunden. Zum Schluß verbinden Sie noch die Leitung E1 (braun) mit dem Kontakt 1 des mini-Tasters und die Leitung E2 (rot) mit dem Impulsausgang der Lichtschranke \perp .

Wir kommen nun zurück auf die zuvor erwähnte Empfindlichkeitseinstellung der Lichtschranke. Wenn alle Verbindungen hergestellt sind und das Interface an den ausgeschalteten (!) Computer angeschlossen ist, schalten Sie diesen ein und laden das Programm ROBOT.JUST und starten es. Auf die Frage, welche Lichtschranke einjustiert werden soll, antworten Sie mit 1. Nun muß der Motor laufen und auf dem Bildschirm ein Meßinstrument erscheinen. Der Zeiger des Meßinstruments muß sich in dem grünen Bereich bewegen, optimalerweise auf 0,5 zeigen. Sollte das nicht der Fall sein, so können Sie wie zuvor beschrieben die Empfindlichkeit an der Lichtschranke einjustieren.

Wenn die Einstellung korrekt ist, verfahren Sie mit den anderen Lichtschranken auf die gleiche Weise. Sie können auch später die Lichtschranke im eingebauten Zustand am Roboter einstellen, müssen dazu aber das Gestänge des Roboters ausklinken. Sind diese Vorarbeiten zur Zufriedenheit abgeschlossen, so können Sie sich von der Funktion der Roboter-Systemkommandos überzeugen. Laden Sie hierzu das Programm ROBOT.SYSTEM und starten es. Ähnlich wie bei dem Grundprogramm meldet sich das Roboter-Systemprogramm zurück als wäre

nichts geschehen. Doch nun können Sie die zuvor besprochenen Kommandos im Direktmodus ausprobieren. Denken Sie daran, daß die Sternchen **nicht** Bestandteil des Kommandos sind, sondern Sie an die unterschiedliche Schreibweise der Computer erinnern sollen.

Geben Sie ein:

* **SYS P1, 64**

und anschließend:

* **SYS ROBOT**

Der Motor muß für eine kurze Zeit laufen und das Rad um etwas mehr als eine Umdrehung drehen. 64 Pegelwechsel entsprechend den 32 schwarzen und 32 hellen Sektoren machen gerade eine Umdrehung aus. Wegen des Motornachlaufs wurde eine höhere Positionszahl erreicht. Ermitteln Sie diese durch das Kommando

* **PRINT USR(P1)**

Vielleicht werden Sie über die Größe des Nachlaufs erstaunt sein. Jedoch können Sie davon ausgehen, daß der Nachlauf später im Roboter aufgrund der Last- und Reibungsverhältnisse geringer sein wird. Im nächsten Versuch geben Sie ein:

* **SYS P1, 10000**

und

* **SYS ROBOT**

Der Motor läuft nun längere Zeit und Sie haben Gelegenheit, den angeschlossenen mini-Taster zu drücken. Sofort bleibt der Motor stehen und das Kommando ist beendet. Mit

* **PRINT USR(P1)**

können Sie sehen, wie weit der Positionszähler kam, bevor Sie die Taste drückten.

Aufbau des Roboters

Nun wollen wir den Motor wieder zurücklaufen lassen:

* **SYS P1,0 : SYS ROBOT**

Der Motor wird sich nun in der anderen Richtung drehen und der Zähler über die Nullposition in den negativen Zahlenbereich überschwingen. Sie können sich davon überzeugen:

* **PRINT USR(P1)**

Zum Abschluß noch ein Hinweis zur genaueren Positionierung. Hierzu müssen Sie die Getriebewelle bremsen. Sie können eine Bremse aufbauen oder aber einfach von Hand bremsen. Geben Sie nun ein:

* **SYS P1, 64 : SYS ROBOT : SYS ROBOT : SYS ROBOT**

Da das Roboter-Kommando nun dreimal mit dem gleichen Sollwert aufgerufen wird, wird die Position noch zweimal nachkorrigiert. Aufgrund der Bremse kommt aber der Motor nicht mehr auf volle Drehzahl und stoppt jedesmal genauer. In dieser Richtung können Sie noch weitere Experimente unternehmen, wenn der Roboter aufgebaut ist und die Motoren unter der Last des Roboterarms stehen.

Zum Aufbau des Roboters verfahren Sie nach der anschließend folgenden bebilderten Bauanleitung. Wenn der Roboter aufgebaut ist, prüfen Sie noch einmal, ob alle Teile exakt ausgerichtet sind und die Antriebsgestänge leichtgängig arbeiten. Hierzu können Sie die Motoren ein klein wenig aus dem Eingriff in die Getriebeverzahnung ausklinken. Danach beginnt die Verkabelung des Roboters. Verfahren Sie nach dem Verdrahtungsplan auf Seite 91.

Beginnen Sie damit, daß Sie den Hauptkabelbaum in der Zugentlastung arretieren. Alle Motorausgänge werden zunächst mit der Lampenreihe verbunden. Danach werden die mit * gekennzeichneten Leitungen, insbesondere der kleine Kabelbaum, durch die zentrale Öffnung des Drehkranzes und entlang dem Aufbau des Roboters geführt. Eine Pinzette ist zum Durchziehen der Kabel hilfreich. Mit Hilfe der grauen Bauplatten können Sie die Kabel in den Nuten der Metallbaustäbe arretieren.

Danach kommt der elektrische Test des Trainingsroboters. Laden Sie hierzu das Diagnoseprogramm DIAGNOSE. Überprüfen Sie die Endtaster, Eingänge E1, E3 und E5. Im nicht betätigten Zustand sollte eine Eins am Bildschirm erscheinen. Bei dem Taster zur Überwachung des Greifers ist es umgekehrt, hier erscheint die Eins, wenn der Taster gedrückt ist. Allen Endtastern ist aber gemeinsam, daß die Eins in dem zulässigen Arbeitsbereich des Roboters erscheint. Warum wurde gerade diese Polarität? Wenn während des Betriebs des Roboters ein Kabel reißt, so ergibt sich die gleiche Situation, wie wenn der Endtaster öffnet. Das Roboter-Systemprogramm reagiert auf Kabelriß daher auch mit Abschalten des Motors.

In die Plusleitung, die zu allen Endtastern führt, ist noch einmal ein Taster eingeführt. Wird dieser Kontakt geöffnet, so gehen sämtliche Eingänge auf Null und das Robotersystemprogramm schaltet alle Motoren ab. Der Taster hat also eine Not-Aus-Funktion. Auch ihn sollten Sie überprüfen.

Von der Funktionsfähigkeit der Gabellichtschranken können Sie sich nochmals durch leichtes Drehen des bedruckten Rades überzeugen. Die zugeordnete Anzeige muß zwischen Eins und Null hin- und herspringen.

Nun zu den Motoren. Für erste Versuche ist es vielleicht besser, wenn das Getriebe des Roboters noch ausgeklinkt ist. Mit den Zahlentasten wird ein Motor angewählt. Nun kann für diesen Motor mit den Tasten R und L Rechts- bzw. Linkslauf gewählt werden. Mit der Taste A wird der Motor abgeschaltet. Auf diese Weise können Sie den Roboter bewegen. Überzeugen Sie sich, daß bei Rechtslauf die Spindelmutter des Oberarm- und Unterarmtriebs nach unten wandert. Ist dies nicht der Fall, müssen die Anschlüsse am Motor umgepolt werden. Desgleichen muß die Drehbewegung des Roboters von oben gesehen im Uhrzeigersinn erfolgen, wenn Rechtslauf gewählt wird. Die Greifzange muß schließlich bei Rechtslauf öffnen.

Sind alle Kontrollen abgeschlossen können wir zur Robotersteuerung übergehen.

Steuerung des Roboters

Um den Roboter programmieren zu können, sollten wir uns zunächst mit den Bewegungsformen des Roboters vertraut machen. Besser als das Diagnoseprogramm ist hierzu das Programm ROBOT.HAND geeignet. In diesem Programm steht der Roboter unter der Kontrolle des Roboter-Systemprogramms, so daß z.B. die Endtaster und die Not-Aus-Schaltung automatisch überwacht werden. Das Programm zeigt Ihnen am Bildschirm die Bedienung des Roboters an und steuert den Roboter in seine Heimposition. Die Heimposition des Roboters wird durch das Ansprechen der Endtaster gekennzeichnet. In dieser Position ist der Arm des Roboters erhoben, die Zange geöffnet und der Roboter zeigt auf die dem Anschlußkabel gegenüberliegende Seite. Allerdings sind in der eigentlichen Heimposition die Endtaster gerade eben wieder freigegeben, da im Arbeitsbereich des Roboter-Systemprogramms die Endtaster nicht ansprechen dürfen. Nach Erreichen der Heimposition wird der Befehl

* SYS INIT

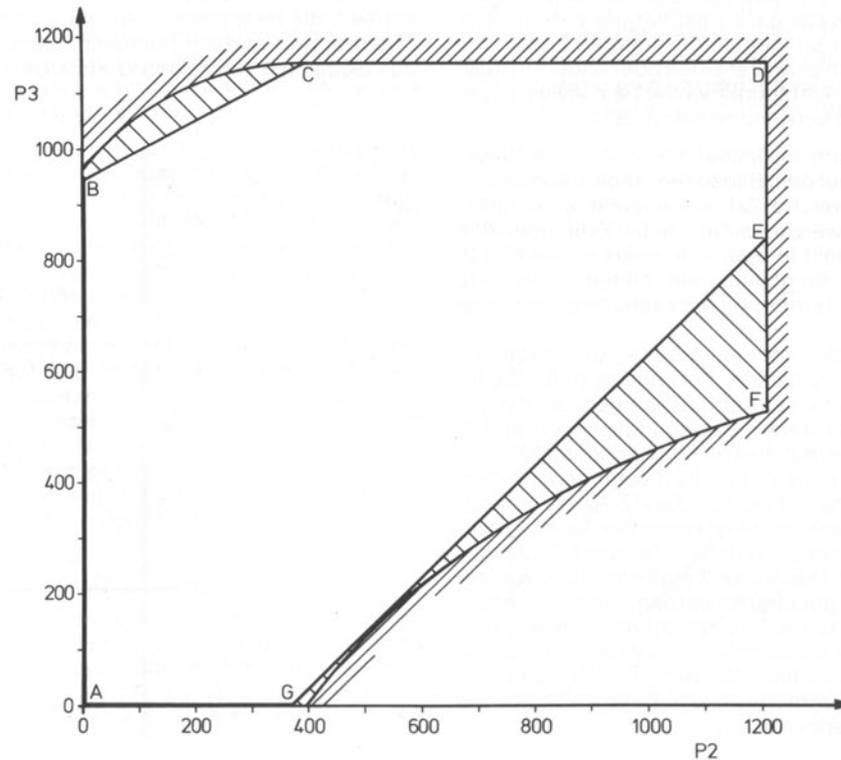
durchlaufen, so daß die Heimposition den Positionswert Null für alle drei Bewegungsachsen erhält.

Die Bewegungen des Roboters werden mit den Zahlentasten angesteuert. Die Wegstrecke pro Tastendruck kann mit Hilfe der Funktionstasten gewählt werden, so daß Sie auf einfache Weise grob und fein positionieren können. Der Greifzangenmotor wird beim Schließen im Gegensatz zu den Roboterachsen immer über feste Zeitintervalle angesteuert. Die Dauer des Zeitintervalls können Sie per Kommando ändern. Wählen Sie sie so, daß der Gegenstand sicher ergriffen wird, jedoch das Zangengetriebe noch nicht durch Verkanten blockiert. Beim Öffnen der Greifzange wartet das Programm auf ein 0-Signal an E7 (Taster nicht betätigt). Die Betätigung der HOME-Taste bringt den Roboter immer wieder in seine Heimposition. Aus der Heimposition kann der Roboter nur in Richtung positiver Posi-

tionsdaten bewegt werden. Dies ist bei den beiden Armachsen einleuchtend, denn ein weiteres Zurückfahren ist nicht möglich. Anders bei der Drehbewegung. Die Heimposition kann freizügig auf jeden Drehwinkel des Roboterkörpers relativ zur Grundplatte festgelegt werden, da der Roboter in seiner Drehbewegung nicht eingeschränkt ist. In den Bau-stufen ist sie so angelegt, daß der Roboter von den Anzeigelampen wegzeigt. Durch Versetzen des

Betätigungs-nockens sind aber auch andere Positionen möglich. Um ein Verdrehen des Kabelbaums zu verhindern, gilt jedoch die gleiche Einschränkung wie bei den Armachsen auch: es können nur positive Positionsdaten angefahren werden. Wenn Sie den Roboter wie gezeigt aufbauen, empfiehlt sich daher, den Arbeitsbereich hauptsächlich vor der Breitseite der Grundplatte anzuordnen, um dann freizügig nach rechts und links schwenken zu können.

Bild 2



Experimentieren Sie mit dem Roboter und versuchen Sie, die beigelegten Werkstücke zu greifen und umzusetzen. Sie werden feststellen, daß Sie hierzu sorgfältig und mit Bedacht vorgehen müssen. Sie werden auch die Wirkung der beiden Bewegungsachsen von Ober- und Unterarm einzuschätzen lernen. Der Oberarm bewirkt hauptsächlich das Vorstrecken und das Zurückziehen der Greifhand, während der Unterarm mehr für eine Auf-/Ab-Bewegung zuständig ist.

Sie werden weiter feststellen, daß die beiden Bewegungsachsen nicht ganz unabhängig voneinander bedient werden können. In bestimmten Zuständen stößt das Gestänge an der einen oder anderen Stelle an. Diesen Gesichtspunkt wollen wir etwas näher erforschen und verwenden hierzu Bild 2.

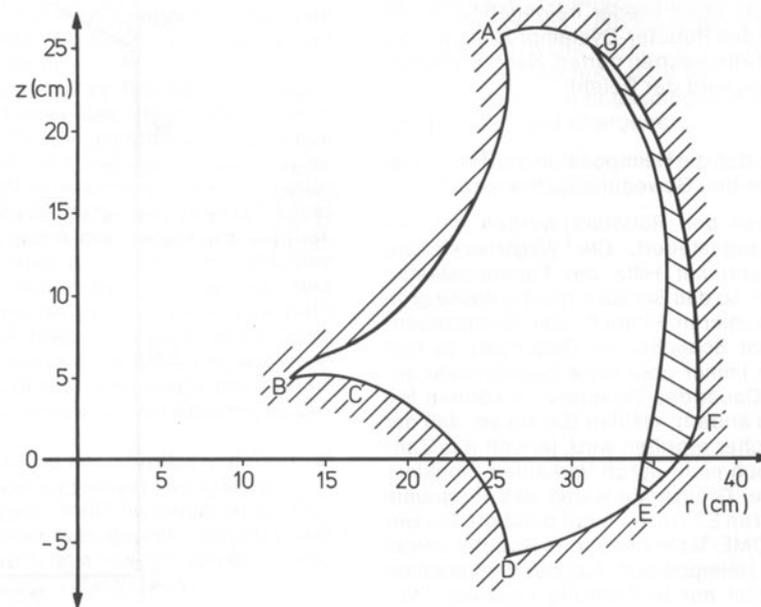
Da das Programm uns immer die Position der Bewegungsachsen auf dem Bildschirm angibt, können wir untersuchen, welche Zahlenkombinationen zulässig sind und welche nicht. Jede Zahlenposition ergibt einen Punkt in dem Achsenkreuz von Bild 2. Alle erlaubten Kombinationen bilden dann eine Fläche, die wir als internen Arbeitsraum bezeichnen werden.

Zur Ermittlung des internen Arbeitsraums beginnen wir bei der Heimposition, Koordinate (0,0). Weiter zurück kann keine der Antriebsspindeln, so daß die negativen Werte beider Bewegungsachsen entfallen. Erhöhen Sie nun die Position der Achse 2 Schritt für Schritt, ohne die Achse 3 zu verändern. In dem Diagramm wandern Sie längs der P2-Achse. Irgendwann wird eine weitere Verstellung der Achse 2 nicht mehr möglich sein, ohne daß ein Teil des Antriebsgestänges aneckt. Die Achse 3 muß ein Stück von der Nullposition weggesteuert werden, um eine weitere Bewegung der Achse 2 zu ermöglichen. In dem Diagramm löst sich die Randlinie des Arbeitsraums von der Koordinatenachse. Mit viel Sorgfalt und unter ständiger Beobachtung des Roboters können Sie die Randlinie verfolgen, bis die Spindel der Achse 2

ganz ausgefahren ist. Nun wird die Bewegungsachse 3 verstellt, bis auch diese ganz ausgefahren ist. In dem Diagramm verläuft die rechte Randbegrenzung senkrecht nach oben. An dem oberen Rand entlang geht es nun durch Verringerung der Positionswerte der Bewegungsachse 2 wieder nach links. Aber auch hier kann Bewegungsachse 2 nicht ganz bis Null zurückgenommen werden, ohne Bewegungsachse 3 ebenfalls zurückzunehmen. Am linken Rand des internen Arbeitsraums angekommen, kann durch Zurückfahren der Bewegungsachse 3 die Heimposition wieder erreicht werden. Nachdem wir nun ein Diagramm besitzen, das uns das Gebiet der zulässigen Werte angibt, interessiert

uns auch, welche Bewegungen des Roboters diesem Gebiet zuzuordnen sind. Die Position des Roboters beschreibt man am leichtesten durch die Position eines idealisierten Punktes, dem „tool-center-point“, TCP. Dieser Punkt ist die Mitte zwischen den Backen der Greifzange. Wird die Lage dieses Punktes im Raum angegeben, so sind die Positionen aller drei Bewegungsachsen daraus ableitbar. Der Zusammenhang zwischen der Höhe des TCP über der Grundplatte, z , und der Entfernung vom Drehmittelpunkt, r , einerseits und den Achspositionen P2 und P3 andererseits ergibt sich aus den nachfolgenden Formeln, die wir hier nicht weiter ableiten wollen:

Bild 3



$$r = 120 \cdot \cos \alpha + 180 \cdot \cos \beta + 110$$

$$z = 120 \cdot \sin \alpha + 180 \cdot \sin \beta + 117,5$$

$$\alpha = 126^\circ - \delta$$

$$\beta = 36^\circ - \varepsilon$$

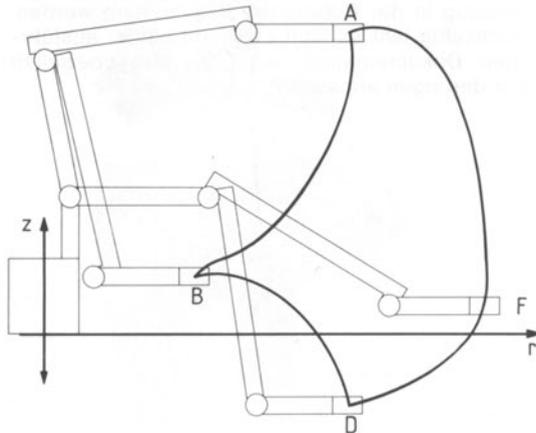
$$\delta = \cos^{-1} \left[\frac{(14004 - (P2 \cdot 0,07363 + 60)^2)}{12240} \right]$$

$$\varepsilon = \cos^{-1} \left[\frac{(14004 - (P3 \cdot 0,07363 + 60)^2)}{12240} \right]$$

(alle Maße in mm)

Bei der Anwendung dieser Gleichungen müssen Sie berücksichtigen, daß diese den idealisierten Zusammenhang wiedergeben. In der Praxis wird sich eine Abweichung durch das notwendige Lagerspiel ergeben. Auch aus dem Vergleich von Bild 2 und Bild 3 können Sie den Zusammenhang zwischen internem und realem Arbeitsraum ablesen. Die charakteristischen Eckpunkte sind in beiden Bildern mit den gleichen Buchstaben bezeichnet. Den realen Arbeitsraum können Sie auch selbst ermitteln. Sie verfolgen gemäß Bild 2 die Randlinie des internen Arbeitsraums und messen die Position des TCP.

Bild 4



Damit Sie eine Vorstellung von der Stellung des Roboters gewinnen, ist in Bild 4 der reale Arbeitsraum zusammen mit schematischen Skizzen des Roboters aufgetragen.

Wir wollen nun das Programm ROBOT.HAND so abändern, daß wir den Roboter nicht versehentlich beschädigen können. Die Randlinien des internen Arbeitsraums müssen abgefragt werden und jede Steuerung außerhalb dessen vermieden werden. Dazu wird die Randlinie durch Geraden angenähert. Während dies parallel zu den Koordinatenachsen und in der linken oberen Ecke des internen Arbeitsraums problemlos möglich ist, verdient der rechte untere Eckabschnitt besondere Beachtung. Wir müssen bei der Definition auch die Arbeitsweise des Positioniersystems beachten.

Wenn das Roboterprogramm gestartet wird, kann das Programm keinerlei Kenntnisse über die vorliegende Position des Roboters haben (in Fachsprache: er besitzt kein absolutes Positioniersystem). Alle Positionskenntnisse werden ja nur über das Zählen von Impulsen, beginnend mit einer bekannten Position, gewonnen (inkrementales Positioniersystem). Diese unabhängig bekannte Position ist die Heimposition, wie sie sich aus dem Ansprechen der Endtaster ergibt. Daher steuert das Programm den Roboter immer zu Beginn an die Heimposition. Die Vorgehensweise ist dabei folgende: Alle Motoren werden gestartet und laufen jeweils solange, bis der zugeordnete Endtaster anspricht. Die Programmierung erfolgt mit den einfacheren Kommandos, ohne Überwachung des Impulseingangs, z.B.:

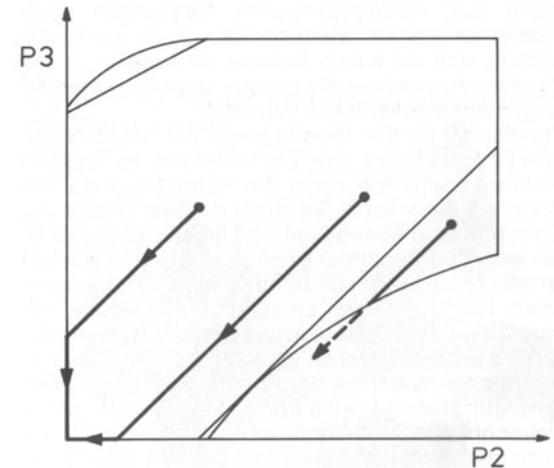
*** SYS M1, RECHTS**

Anschließend laufen sie wieder, wie zuvor beschrieben, vom Endschalter weg. Diese Bahn, im internen Arbeitsraum aufgezeichnet, bildet eine Diagonallinie, da bis auf kleine Exemplarstreuungen beide Motoren der Armantriebe gleich schnell laufen.

Sobald die Bahn auf eine Koordinatenachse stößt, verläuft sie längs dieser bis in den Nullpunkt (Bild 5). Nehmen wir nun an, der Roboter sei nahe dem Punkt F abgestellt worden. Bei dem Anfahren der Heimposition würde die Bahn das erlaubte Gebiet verlassen. Daher muß der Eckabschnitt bei Punkt F so gewählt werden, daß er von einer Geraden mit 45° Neigung gebildet wird. Diese Begrenzungen des Arbeitsraums sind in den Bildern 2 und 3 als Verbindungslinie EG eingezeichnet.

Die Berücksichtigung dieser Begrenzungen erfordert nur wenige zusätzliche Programmzeilen. Wenn Sie nun das Programm ROBOT.RAUM laden, so haben Sie die gleichen Steuerungsmöglichkeiten des Roboters wie zuvor, jedoch wird er sich weigern, seinen Arbeitsraum zu überschreiten.

Bild 5



Teach-in Verfahren

Die Benutzung der bisherigen Programme war zwar sehr hilfreich zum Verständnis der Geometrie des Roboters, stellte jedoch keine Programmierung des Roboters im eigentlichen Sinne dar. Für jede Bewegung des Roboters war der Eingriff des Bedieners notwendig. Ein Kennzeichen der Roboter-Programmierung ist jedoch, daß der Roboter die aufgetragene Tätigkeit selbständig durchführt.

In dem vorangegangenen Abschnitt hatten wir auch die Gleichungen angegeben, die den Zusammenhang zwischen den Positionen der Bewegungsachsen im internen Arbeitsraum und der Position des TCP im realen Arbeitsraum vermitteln. Mit dieser Kenntnis wäre es möglich, eine bestimmte Bewegungsabfolge des Roboters festzulegen und in einer Tabelle zu codieren. Danach kann das Roboterprogramm der Reihe nach die Positionsdaten aus der Tabelle entnehmen und den Roboter entsprechend steuern. Diese Methode wird auch in der Praxis eingesetzt, insbesondere wenn die Positionsdaten nicht nur aus einer Tabelle ermittelt, sondern vielleicht aus vorangegangenen Messungen noch modifiziert werden. Denken Sie z.B. an ein Sichtsystem, das mit einem Roboter verbunden ist und ihm Kenntnisse über die genaue Lage und Orientierung eines Werkstückes vermittelt.

Praktischer für die überwiegende Zahl der Einsatzfälle ist jedoch das Teach-In Verfahren. Im Teach-In Verfahren wird wie zuvor die Handsteuerung des Roboters eingesetzt. Sie dient darüber hinaus zur Erzeugung der obengenannten Tabelle. Immer wenn ein wichtiger Punkt der Bahn des Roboters erreicht ist wird er auf ein besonderes Kommando abgespeichert. Nach und nach entsteht so die Tabelle entsprechend dem Geschick des Bedieners. Wenn die Tabelle komplett erstellt ist, kann sie dann als Vorlage dienen, nach der der Roboter selbständig den Bewegungsablauf wiederholt. Eine Kenntnis der Umrechnungsgleichungen ist nicht erforderlich und auch die Fehlerhäufigkeit bei der Festlegung der

Bahn wird geringer sein. Der Roboterinstructor hat ja den Roboter ständig vor Augen.

Ein solches Programm liegt unter dem Namen ROBOT.TEACH vor. Über das Beschriebene hinaus sind noch weitere Komfortstufen zur Pflege der Bewegungstabellen eingebaut. Die Tabelle kann z.B. auf Diskette oder Kassette abgespeichert und von dort auch wieder geladen werden. Sie kann über einen angeschlossenen Drucker ausgedruckt werden. Tabellenpunkte können gelöscht und mehrere Tabellen im Speicher des Computers vereinigt werden.

Wenn Sie das Programm laden und starten, wird Ihnen zunächst ein Hauptmenü auf dem Bildschirm angezeigt, mit dessen Hilfe Sie die gewünschte Funktion wählen können. Beim ersten Start können Sie auf eine beispielhafte Bewegung zurückgreifen, die schon auf Diskette bzw. Kassette abgespeichert ist. Wählen Sie daher zunächst den Menüpunkt „L“ zum Laden eines Files an. Sie müssen anschließend den Filenamen angeben; er lautet BEISPIEL 1. Nach dem Laden des Files erscheint wieder das Menü. Sie können als nächstes die Bewegungstabelle ausdrucken, sofern Sie einen Drucker an Ihren Computer angeschlossen haben. Dies erfolgt durch Anwahl des Menüpunktes „P“.

Jetzt wollen wir auch den Roboter in Betrieb nehmen. Mit dem Menüpunkt „R“ wird die Ausführung der Bewegung angewählt. Nach Angabe der Anzahl der Durchläufe durch die Bewegungstabelle beginnt der Roboter die codierte Bewegung auszuführen. Wollen Sie nicht das Ende der Bewegung abwarten, so können Sie auch vorzeitig durch Drücken der Taste „M“ wieder in das Hauptmenü zurückgelangen.

Doch nun zum Teach-In Verfahren. Wählen Sie Menüpunkt „T“ an. Das Programm fragt Sie, ob die bisherige Bewegungsfolge gelöscht werden soll. Sie haben nun die Wahl, den eben geladenen Bewe-

gungsablauf zu verlängern oder ganz von neuem anzufangen. Nach Beantwortung der Frage erscheint ein neues Menü, das Ihnen in groben Zügen schon von der Handsteuerung bekannt ist. Wie schon gewohnt werden die Motoren über die Zahlentasten, die Schrittweite über die Funktionstasten gesteuert. Auch das Anfahren der Heimposition ist vorgesehen. Neu kommt hinzu, daß mit jedem Betätigen der RETURN-Taste die gerade vorliegende Roboterposition an das bisherige Tabellenende angefügt wird. Mit der Löschtaste DEL wird jedoch der letzte Tabellenpunkt wieder ausgetragen, so daß Sie fehlerhafte Bahnen auch wieder korrigieren können.

Haben Sie einen Bewegungsablauf nach Ihren Vorstellungen eingegeben, so gelangen Sie durch Drücken der Taste „M“ wieder in das Hauptmenü. So wie das Menü angelegt ist, können Sie aber auch immer wieder zwischendurch einen Probedurchlauf anfordern und anschließend den Tabellenaufbau fortsetzen. Oder aber zwischendurch den Bewegungsablauf auf Diskette sichern oder ausdrucken. Das Programm ROBOT.TEACH wird Ihr Standardwerkzeug in der Roboterprogrammierung werden. Gleichzeitig läßt es sich aufgrund seiner ausführlichen Dokumentation leicht an Ihre speziellen Anforderungen anpassen.

Weitere Experimente

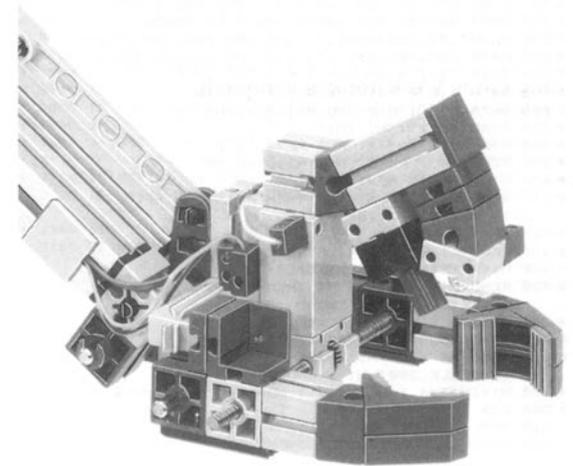
Zum Schluß noch einige Hinweise, wie Sie den Trainingsroboter auch noch einsetzen können. Hierzu sollen Ihnen die folgenden Abbildungen einige Anregungen vermitteln. So kann die Greifzange auch nach unten zeigend montiert werden. Diese Greiferhaltung ist z.B. zum Greifen von Schachfiguren geeignet. Sicherlich eine besondere Herausforderung für einen Programmierer, einen Schachroboter zu bauen.



Im nächsten Bild ist ein Elektromagnet angebaut. Auf diese Weise lassen sich Eisenteile recht einfach greifen.



Zum Schluß der Anbau einer Reflexlichtschranke an den Greifarm. Deutlich ist die Lampe zu erkennen, die den Raum zwischen den Greiferbacken ausleuchtet. Sie wird direkt an das Netzgerät angeschlossen. Der benachbarte Fotowiderstand ist durch eine Kappe und einen Tubus so geschützt, daß er nicht das direkte Licht der Lampe registrieren kann. Er reagiert jedoch mit einer deutlichen Widerstandsänderung, wenn ein heller Gegenstand in den Greifbereich kommt. Diese Widerstandsänderung kann durch den Interfaceeingang EX oder EY registriert und dem Computer mitgeteilt werden. Durch geeignete Suchprogramme kann auf diese Weise ein Objekt erkannt werden und der Greifarm genau über diesem ausgerichtet werden. Danach muß der Arm nur noch abgesenkt werden, um das Objekt zu greifen. Auch diese Aufgabe ist eine rechte Herausforderung an Ihre Programmierkunst.



Prog. ROBOT.JUST

```
* 1 PRINT CHR$(147)
* 2 PRINT CHR$(31)
* 3 POKE53280,0
* 4 POKE53281,0
* 5 PRINT"GRUNDPROGRAMM WIRD BELADEN"
* 10 REM MODIFIZIERTES INTERFACE PROGRAMM ZUR
* 15 REM LICHTSCHRANKENJUSTAGE (COMMODORE 64)
* 20 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
* 30 REM AUFRUF DES PROGRAMMS MIT
* 40 REM SYS M1,EIN SYS M1,AUS
* 50 REM SYS M1,RECHTS SYS M1,LINKS
* 60 REM USR(E1) USR(EX) USR(EY)
* 70 REM M1 BIS M4 SIND MOTORANSTEUERUNGEN
* 80 REM E1 BIS E8 SIND DIGITALEINGANGS
* 90 REM EX UND EY SIND ANALOGEINGANGS
* 100 DATA 52736,169,0,240,38,169,3,208,10,53573
* 110 DATA 169,12,208,6,169,48,208,2,54395
* 120 DATA 169,192,120,133,255,32,253,174,55723
* 130 DATA 165,254,5,255,133,254,32,158,56979
* 140 DATA 183,138,37,255,133,255,165,254,58399
* 150 DATA 69,255,133,254,168,169,63,141,59651
* 160 DATA 3,221,162,8,169,48,6,254,60522
* 170 DATA 144,2,9,4,141,1,221,9,61053
* 180 DATA 8,141,1,221,202,208,237,169,62240
* 190 DATA 57,141,1,221,132,254,88,96,63230
* 200 DATA 120,32,161,183,134,255,169,0,64284
* 210 DATA 141,156,206,169,255,141,155,206,65713
* 220 DATA 169,50,141,1,221,9,8,141,66453
* 230 DATA 1,221,162,8,10,44,1,221,67121
* 240 DATA 16,2,9,1,160,48,140,1,67498
* 250 DATA 221,160,56,140,1,221,202,208,68707
* 260 DATA 235,37,255,240,2,159,1,24,69670
* 270 DATA 109,156,206,141,155,206,206,155,71005
* 280 DATA 206,208,205,172,155,206,32,162,72352
* 290 DATA 179,88,96,0,0,0,0,0,72157
* 300 DATA 1,2,4,8,16,32,64,128,72970
* 360 DATA 255,170,85,85,80,206,73851
* 370 READ INIT : M1=INIT
* 380 FOR M3=0 TO 19: FOR M2=0 TO 7
* 390 READ M4 : POKE INIT+M3*8+M2,M4
* 400 M1=M1+M4 : NEXT
* 410 READ M4 : IF M1<>M4 THEN PRINT"DATAFEHLER
IN ZEILE" ;M3*10+100:END
* 420 NEXT
* 430 READ E1,E2,E3,E4,E5,E6,E7,E8
* 440 M1=M1+E1+E2+E3+E4+E5+E6+E7+E8
* 450 READ M4 : IF M1<>M4 THEN PRINT"DATAFEHLER
IN ZEILE 350" : END
* 460 READ AUS,LINKS,RECHTS,EIN,M2,M3
* 470 M1=M1+AUS+LINKS+RECHTS+EIN+M2+M3
* 480 READ M4 : IF M1<>M4 THEN PRINT"DATAFEHLER
IN ZEILE 360",M1 : END
* 490 M1=INIT+4 : M2=M1+4 : M3=M2+4 : M4=M3+4
* 500 SYS INIT
* 510 REM
* 520 REM FISCHERTECHNIK COMPUTING

530 REM
540 REM LICHTSCHRANKENJUSTAGE
550 REM
560 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1985
570 REM
580 REM BELEGUNG DES INTERFACE
590 REM MOTOR IMPULS KOMMANDO
600 REM EINGANG
* 610 REM M1 E2 USR(E2)
* 620 REM M2 E4 USR(E4)
* 630 REM M3 E6 USR(E6)
* 640 REM M4 E8 USR(E8)
650 REM
660 REM FUNKTION
670 REM DAS PROGRAMM DIENT ZUR PUSBREITENJUSTAGE
680 REM DER GABELLICHTSCHRANKE.
690 REM UM OPTIMALE AUSWERTBEDINGUNGEN ZU ERHALTEN
700 REM SOLLTE DAS TASTVERHAELTNIS 1:1 SEIN.
710 REM DIES WIRD EINGESTELLT INDEM MAN MIT EINEM
720 REM SCHRAUBENDREHER VORSICHTIG AM INNENLIEGENDEN
730 REM POTENTIOMETER DREHT. DER FEHLER DER ANZEIGE
740 REM SOLLTE SICH IM GRUENEN FELD BEWEGEN.
750 REM
* 1000 PRINT CHR$(147)
1010 PRINT"WELCHE LICHTSCHRANKE SOLL"
1020 INPUT"EINJUSTIERT WERDEN ";L
1030 IF L=1THEN LET E=E2:M=M1
1040 IF L=2THEN LET E=E4:M=M2
1050 IF L=3THEN LET E=E6:M=M3
1060 IF L=4THEN LET E=E8:M=M4
1070 IF L>4 OR L<1 THEN GOTO 1020
* 1080 PRINT CHR$(147)
* 1090 PRINT CHR$(28)" F I S C H E R"CHR$(31)
" T E C H N I K "
1100 PRINT
* 1110 PRINT CHR$(158)" C O M P U T I N G "
CHR$(31)
1120 PRINT
1130 PRINT" PULSBREITENJUSTAGE GABELLICHTSCHRANKE"
1140 PRINT:PRINT
1150 PRINT"0 0.25 0.5 0.75 1"
* 1160 DATA 180,160,160,160,160,160,160,160,160,160
* 1170 DATA 160,160,160,160,160,160,160,160,160,160
* 1180 DATA 160,160,160,160,160,160,160,160,160,160
* 1190 DATA 160,160,160,160,160,160,160,160,160,160
* 1200 DATA 160,160,160,160,160,160,160,160,160,160
1210 FOR C=0 TO 39
1220 READ D
1230 PRINT CHR$(D);
1240 NEXT C
* 1250 DATA 28,111,183,183,183,183,183,183,183
* 1260 DATA 183,183,183,183,183,183,183,183,183,183
* 1270 DATA 183,183,183,183,183,183,183,183,183,183
* 1280 DATA 183,183,183,183,183,183,183,28,183
* 1290 DATA 183,183,183,183,183,183,183,183,183,183
* 1300 DATA 183,183,112
1310 FOR C=0 TO 42
1320 READ D
1330 PRINT CHR$(D);
1340 NEXT C
1350 PRINT
* 1360 DATA 28,108,175,175,175,175,175,175,175,175
* 1370 DATA 175,175,175,175,175,175,175,175,175,175
* 1380 DATA 175,175,175,175,175,175,175,175,175,175
* 1390 DATA 175,175,175,175,175,175,28,175
* 1400 DATA 175,175,175,175,175,175,175,175,175,175
* 1410 DATA 175,175,188
1420 FOR C=0 TO 42
1430 READ D
1440 PRINT CHR$(D);
1450 NEXT C
* 2000 SYS M,EIN
2010 PRINT:PRINT
* 2020 PRINT CHR$(31)"BITTE ANZEIGEPFEIL IN GRUENEN"
2030 PRINT"BEREICH BRINGEN"
* 2040 PRINT"OBERE"CHR$(18);CHR$(28)" VORSICHTIG";
* 2050 PRINT CHR$(31);CHR$(146)"MIT DEM SCHRAUBEN-"
2060 PRINT"DREHER NACH LINKS ODER RECHTS DREHEN!"
2070 PRINT:PRINT
2080 PRINT"ENDE DER MESSUNG MIT "CHR$(18)"F1"
CHR$(146)
2090 FOR I=1 TO 11
* 2100 PRINT CHR$(145);
2110 NEXT I
* 2120 PRINT SPC(38/255*USR(E));CHR$(5)"+"CHR$(31);
* 2130 PRINT CHR$(145)
2140 PRINT"
* 2150 PRINT CHR$(145);
* 2160 GET A#:IF A#=CHR$(133)THEN RESTORE:GOTO 10
2170 GOTO 2120
```

Prog. ROBOT.HAND

```

* 500 SYS INIT
510 REM
520 REM FISCHERTECHNIK COMPUTING
530 REM
540 REM HANDSTEUERUNG DES TRAININGSROBOTERS
550 REM MIT POSITIONSERKENNUNG
560 REM
570 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1985
580 REM
590 REM FUNKTION
600 REM DER ROBOTER WIRD UEBER DIE TASTATUR VON HA
ND GESTEUERT.
610 REM GLEICHZEITIG WIRD UEBER DIE SEKTORSCHIEBE
620 REM DIE POSITION DES ROBOTERS EINGELESEN.
630 REM ZUR STEUERUNG DIENEN DABEI DIE TASTEN 1-8.
640 REM DIE HOMETASTE DIENET ZUM ANFAHREN DER HEIMP
OSITION.
* 650 REM DIE TASTEN F1-F7 LEGEN DEN WEG PRO TASTEND
RUCK FEST.
660 REM TASTENBELEGUNG
670 REM 1 UND 2 = M1 RECHTS UND LINKS
680 REM 3 UND 4 = M2 RECHTS UND LINKS
690 REM 5 UND 6 = M3 RECHTS UND LINKS
700 REM 7 UND 8 = M4 RECHTS UND LINKS
* 710 REM F1=256 SEKTOREN WEG PRO TASTENDRUCK
* 720 REM F3=64 SEKTOREN WEG PRO TASTENDRUCK
* 730 REM F5=16 SEKTOREN WEG PRO TASTENDRUCK
* 740 REM F5=4 SEKTOREN WEG PRO TASTENDRUCK
* 750 REM HOME=HEIMPOSITION ANFAHREN
* 760 PRINT CHR$(147)
770 PRINT:PRINT
* 780 PRINT CHR$(28)"      F I S C H E R"CHR$(31)" T
      E C H N I K"CHR$(144)
790 PRINT
900 PRINT"      C O M P U T I N G"
910 PRINT:PRINT
920 PRINT"      D R E I A C H S I G E R"
930 PRINT
940 PRINT"      T R A I N I N G S R O B O T E R"
950 PRINT:PRINT
960 PRINT" STEUERUNG UEBER DIE TASTATUR MIT"
970 PRINT" AUFEZEICHEN DER ROBOTERPOSITION "
980 PRINT
990 PRINT" ROBOTER FAHRT IN HEIMPOSITION"
910 GOSUB 3000
920 LET ZZ=200
* 1000 PRINT CHR$(147)
1010 PRINT"TASTENFUNKTIONEN:      POSITION:"
1020 PRINT:
* 1030 PRINT CHR$(18)"1"CHR$(146)"  ROBOTER NACH LIN
      KS"
* 1040 PRINT CHR$(18)"2"CHR$(146)"  ROBOTER NACH REC
      HTS"
1050 PRINT
* 1060 PRINT CHR$(18)"3"CHR$(146)"  OBERARM VOR"
* 1070 PRINT CHR$(18)"4"CHR$(146)"  OBERARM ZURUECK"
1080 PRINT
* 1090 PRINT CHR$(18)"5"CHR$(146)"  UNTERARM AB"
* 1100 PRINT CHR$(18)"6"CHR$(146)"  UNTERARM AUF"
1110 PRINT
* 1120 PRINT CHR$(18)"7"CHR$(146)"  ZANGE AUF"
* 1130 PRINT CHR$(18)"8"CHR$(146)"  ZANGE ZU"
1140 PRINT
1150 PRINT "SCHRIITUEITEN"
* 1155 PRINT "ZEITKONSTANTE GREIFZANGE  "CHR$(18)"+/-
      "CHR$(146)
1160 PRINT
* 1170 PRINT CHR$(18)"F1"CHR$(146)"  256 SEGMENTE"
* 1180 PRINT CHR$(18)"F3"CHR$(146)"   64 SEGMENTE"
* 1190 PRINT CHR$(18)"F5"CHR$(146)"   16 SEGMENTE"
* 1200 PRINT CHR$(18)"F7"CHR$(146)"    4 SEGMENTE"
1210 PRINT
* 1220 PRINT CHR$(18)"HOME"CHR$(146)"  HEIMPOSITION
      ANFAHREN"
2000 LET Q%=16
* 2010 LET CL$="      "+CHR$(157)+CHR$(157)+CHR$(157)
      +CHR$(157)+CHR$(157)
* 2020 LET Q1%=USR(P1):IF Q1%<0 THEN Q1%=0
* 2030 LET Q2%=USR(P2):IF Q2%<0 THEN Q2%=0
* 2040 LET Q3%=USR(P3):IF Q3%<0 THEN Q3%=0
2050 REM STEUERUNG UEBER TASTATUR
* 2060 LET A=PEEK(203):REM TASTATURREGISTER LESEN
* 2070 IF A=56 THEN Q1%=Q1%+Q% : REM DREHUNG NACH LI
      NKS
* 2080 IF A=59 THEN Q1%=Q1%-Q% : REM DREHUNG NACH RE
      CHTS
2090 REM ARBEITSRAUM DREHUNG
2100 IF Q1%<0 THEN Q1%=0
* 2120 PRINT CHR$(19)
* 2130 PRINT CHR$(17);CHR$(17);
* 2140 PRINT TAB(30);CL$;Q1%
* 2150 SYS P1,Q1%
* 2160 IF A=8 THEN Q2%=Q2%+Q% : REM OBERARM NACH VO
      RNE
* 2170 IF A=11 THEN Q2%=Q2%-Q% : REM OBERARM NACH HI
      NTEN
2180 REM ARBEITSRAUM OBERARM
2220 IF Q2%<0 THEN Q2%=0
* 2230 PRINT CHR$(17);CHR$(17);
* 2240 PRINT TAB(30);CL$;Q2%
* 2250 SYS P2,Q2%
* 2260 IF A=16 THEN Q3%=Q3%+Q% : REM UNTERARM NACH U
      NTEN
* 2270 IF A=19 THEN Q3%=Q3%-Q% : REM UNTERARM NACH O
      BEN
2280 REM ARBEITSRAUM UNTERARM
2320 IF Q3%<0 THEN Q3%=0
* 2330 PRINT CHR$(17);CHR$(17);
* 2340 PRINT TAB(30);CL$;Q3%
* 2350 SYS P3,Q3%
2360 REM ROBOTERROUTINE STARTEN
* 2370 SYS ROBOT
2390 REM ZANGENROUTINE
* 2390 IF USR(E7)=0 THEN PRINT CHR$(17);CHR$(17);TAB
      (28);"AUF"
* 2400 IF USR(E7)=1 THEN PRINT CHR$(17);CHR$(17);TAB
      (28);"ZU "
* 2410 IF A<>24 THEN GOTO 2470
2420 IF ZA$="AUF" THEN GOTO 2470
2430 LET ZA$="AUF"
* 2440 SYS M4,RECHTS
* 2450 IF USR(E7)=1 THEN GOTO 2440
* 2460 SYS M4,AUS
* 2470 IF A<>27 THEN GOTO 2540
2480 IF ZA$="ZU " THEN GOTO 2540
2490 LET ZA$="ZU "
2500 FOR Z=1 TO ZZ
* 2510 SYS M4,LINKS
2520 NEXT
* 2530 SYS M4,AUS
2540 REM SCHRIITUEITE EINSTELLEN
* 2542 GET A#
2543 IF A#="+" THEN IF ZZ<500 THEN LET ZZ=ZZ+50
2544 IF A#="-" THEN IF ZZ>50 THEN LET ZZ=ZZ-50
* 2550 IF A#4 THEN Q%=256
* 2560 IF A#5 THEN Q%=64
* 2570 IF A#6 THEN Q%=16
* 2580 IF A#3 THEN Q%=4
* 2590 PRINT CHR$(17);CHR$(17);
* 2600 PRINT TAB(30);CL$;Q%
* 2605 PRINT TAB(29);CL$;ZZ
* 2610 IF A=51 THEN GOSUB 3000
2620 GOTO 2020
3000 REM HEIMPOSITION ANFAHREN
3010 LET H1=1:H2=1:H3=1:H4=1
* 3020 IF USR(E1)=1 AND H1=1 THEN SYS M1,RECHTS
* 3030 IF USR(E3)=1 AND H2=1 THEN SYS M2,RECHTS
* 3040 IF USR(E5)=1 AND H3=1 THEN SYS M3,RECHTS
* 3050 IF USR(E7)=1 AND H4=1 THEN SYS M4,RECHTS
* 3060 IF USR(E1)=0 THEN SYS M1,LINKS:H1=-1
* 3070 IF USR(E3)=0 THEN SYS M2,LINKS:H2=-1
* 3080 IF USR(E5)=0 THEN SYS M3,LINKS:H3=-1
* 3090 IF USR(E7)=0 THEN SYS M4,AUS:H4=0
* 3100 IF USR(E1)=1 AND H1=-1 THEN SYS M1,AUS:H1=0
* 3110 IF USR(E3)=1 AND H2=-1 THEN SYS M2,AUS:H2=0
* 3120 IF USR(E5)=1 AND H3=-1 THEN SYS M3,AUS:H3=0
3130 IF H1<>0 OR H2<>0 OR H3<>0 OR H4<>0 THEN GOTO
      3020
* 3140 SYS INIT
3150 RETURN

```

Prog. ROBOT.RAUM

```

560 REM UEBERWACHUNG DES ARBEITSRAUM

880 PRINT "UND UEBERWACHUNG DES ARBEITSRAUM"

2110 IF Q1%>3600 THEN Q1%=3600

2190 IF Q2%>370+03% THEN Q2%=370+03%
2200 IF Q2%>1210 THEN Q2%=1210
2210 IF Q2%<-1709+1.82*03% THEN Q2%=-1709+1.82*03%

2290 IF Q3%>940+0.55*02% THEN Q3%=940+0.55*02%
2300 IF Q3%>1160 THEN Q3%=1160
2310 IF Q3%<-370+02% THEN Q3%=-370+02%

```

Prog. ROBOT.TEACH

```

* 500 SYS INIT
510 REM
520 REM FISCHERTECHNIK COMPUTING
530 REM
540 REM TRAININGSROBOTER IM TEACH IN MODUS
550 REM
560 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1985
570 REM
580 REM FUNKTION
590 REM DAS PROGRAMM ZEIGT ZUNAECHST DAS HAUPTMENU
E.
600 REM FOLGENDE FUNKTIONEN KOENNEN GEWAERHLT WERDE
N:
610 REM T = ROBOTER TEACH IN MODUS
620 REM R = TEACH PROGRAMM AUSFUEHREN
630 REM S = TEACH PROGRAMM ABSPEICHERN
640 REM L = TEACH PROGRAMM LADEN
650 REM D = DISKETTENINHALT
660 REM P = TEACH PROGRAMM AUSDRUCKEN
670 REM E = PROGRAMMENDE
680 REM
690 REM DER ROBOTER TEACH IN MODUS FUEHRT ZU EINEM
UNTERMENUE ZUR STEUERUNG
700 REM DES ROBOTERS PER HAND UND ZUR ABSPEICHERUN
G DER BAHN.
710 REM TASTENBELEGUNG:
720 REM 1 UND 2 = M1 RECHTS UND LINKS (DREHUNG)
730 REM 3 UND 4 = M2 RECHTS UND LINKS (OBERARM)
740 REM 5 UND 6 = M3 RECHTS UND LINKS (UNTERARM)
750 REM 7 UND 8 = M4 RECHTS UND LINKS (ZANGE)
* 760 REM F1 = 256 SEKTOREN WEG PRO TASTENDRUCK)
* 770 REM F3 = 64 SEKTOREN WEG PRO TASTENTRUCK
* 780 REM F5 = 16 SEKTOREN WEG PRO TASTENDRUCK
* 790 REM F5 = 4 SEKTOREN WEG PRO TASTENDRUCK
* 900 REM HOME = HEIMPOSITION ANFAHREN
810 REM RETURN = ABSPEICHERN
* 820 REM DEL = LETZTE ABSPEICHERUNG LOESCHEN
830 REM M = ZURUECK ZUM HAUPTMENUE
1000 DIM DR(100),DA(100),UA(100),ZA(100)
1010 LET ID=-1:REM ZEIGER IN TEACH TABELLE
1015 LET Z2=200:REM ZEITKONSTANTE GREIFZANGE
1020 GOSUB 10000
1030 PRINT
* 1040 PRINT CHR$(18)"T"CHR$(146)" ROBOTER TEACH I
N MODUS"
1050 PRINT
* 1060 PRINT CHR$(18)"R"CHR$(146)" TEACH PROGRAMM
AUSFUEHREN"
1070 PRINT
* 1080 PRINT CHR$(18)"S"CHR$(146)" TEACH PROGRAMM
ABSPEICHERN"
1090 PRINT
* 1100 PRINT CHR$(18)"L"CHR$(146)" TEACH PROGRAMM
LADEN"
1110 PRINT
* 1120 PRINT CHR$(18)"D"CHR$(146)" DISKETTENINHALT
"
1130 PRINT
* 1140 PRINT CHR$(18)"P"CHR$(146)" TEACH PROGRAMM
AUSDRUCKEN"
1150 PRINT
* 1160 PRINT CHR$(18)"E"CHR$(146)" PROGRAMMENDE";
1170 REM TASTATURABFRAGE
* 1180 GET A#
1190 IF A#="" THEN GOTO 1180
1200 IF A#="T" THEN GOTO 2010:REM TEACH MODUS
1210 IF A#="R" THEN GOTO 4010:REM AUSFUEHRMODUS
1220 IF A#="S" THEN GOTO 5010:REM ABSPEICHERN
1230 IF A#="L" THEN GOTO 6010:REM LADEN
1240 IF A#="D" THEN GOTO 8010:REM DISKETTENINHALT
1250 IF A#="P" THEN GOTO 7010:REM DRUCKERAUSGABE
1260 IF A#="E" THEN GOTO 8010:REM PROGRAMMENDE
1270 GOTO 1180
2000 REM TEACH IN MODUS
2010 IF ID=-1 THEN GOTO 2100
* 2020 PRINT CHR$(147)
2030 FOR I=1 TO 5
2040 PRINT
2050 NEXT I
2060 INPUT"ALTES TEACH PROGRAMM LOESCHEN (J/N)";K#
2070 IF K#="N"THEN 2130
2080 IF K#(">")J"THEN 2060
2090 LET ID=-1
* 2100 PRINT CHR$(147)
2110 PRINT"ROBOTER FAEHRT IN HEIMPOSITION"
2120 GOSUB 11000:REM HOMEROUTINE
* 2130 PRINT CHR$(147)
2140 PRINT"TASTENFUNKTIONEN: POSITION:"
2150 PRINT
* 2160 PRINT CHR$(18)"1"CHR$(146)" ROBOTER NACH LIN
KS"
* 2170 PRINT CHR$(18)"2"CHR$(146)" ROBOTER NACH REC
HTS"
* 2180 PRINT CHR$(18)"3"CHR$(146)" OBERARM VOR"
* 2190 PRINT CHR$(18)"4"CHR$(146)" OBERARM ZURUECK"
* 2200 PRINT CHR$(18)"5"CHR$(146)" UNTERARM AB"
* 2210 PRINT CHR$(18)"6"CHR$(146)" UNTERARM AUF"
* 2220 PRINT CHR$(18)"7"CHR$(146)" ZANGE AUF"
* 2230 PRINT CHR$(18)"8"CHR$(146)" ZANGE ZU"
2240 PRINT
2250 PRINT "SCHRITTWEITE"
* 2255 PRINT "ZEITKONSTANTE GREIFZANGE "CHR$(18)"+/
-"CHR$(146)
2260 PRINT
* 2270 PRINT CHR$(18)"F1"CHR$(146)" 256 SEGMENTE";
* 2280 PRINT TAB(20);CHR$(18)"M "CHR$(146)" MENU
E"
* 2290 PRINT CHR$(18)"F3"CHR$(146)" 64 SEGMENTE "
;
* 2300 PRINT TAB(20);CHR$(18)"HOME"CHR$(146)" HEIM
POSITION"
* 2310 PRINT CHR$(18)"F5"CHR$(146)" 16 SEGMENTE";
* 2320 PRINT TAB(20);CHR$(18)"RETURN"CHR$(146)" LERN
E"

```

```

* 2330 PRINT CHR$(18)"F7"CHR$(146)"      4 SEGMENTE";
* 2340 PRINT TAB(20);CHR$(18)"DEL"CHR$(146)"  LOES
      CHEN"
2350 PRINT
2360 PRINT
2370 PRINT" NR DREH      OARM      UARM      ZANGE";
2380 PRINT*
      *;
* 2390 PRINT CHR$(145);
3000 LET Q%=16
* 3010 LET CL#=""      "+CHR$(157)+CHR$(157)+CHR$(157)
      +CHR$(157)+CHR$(157)
3020 REM ABFRAGESCHLEIFE
* 3030 LET Q1%=USR(P1)
* 3040 LET Q2%=USR(P2)
* 3050 LET Q3%=USR(P3)
3060 REM TASTATURABFRAGE (BEWEGUNGEN)
* 3070 LET A=PEEK(203);REM TASTATURREGISTER LESEN
* 3080 IF A=56 THEN Q1%=Q1%+Q% : REM TASTE LINKS DRE
      HEN
* 3090 IF A=59 THEN Q1%=Q1%-Q% : REM TASTE RECHTS DR
      EHEN
3100 REM ARBEITSRAUM DREHUNG
3110 IF Q1%<0 THEN Q1%=0
3120 IF Q1%>3600 THEN Q1%=3600
* 3130 PRINT CHR$(19)
* 3140 PRINT CHR$(17);CHR$(17);
* 3150 PRINT TAB(30);CL#;Q1%
* 2160 SYS P1,Q1%
* 3170 IF A=8 THEN Q2%=Q2%+Q% : REM OBERARM VOR
* 3180 IF A=11 THEN Q2%=Q2%-Q% : REM OBERARM ZURUECK
3190 REM ARBEITSRAUM OBERARM
3200 IF Q2%>370+Q3% THEN Q2%=370+Q3%
3210 IF Q2%<1210 THEN Q2%=1210
3220 IF Q2%<-1709+1.82*Q3% THEN Q2%=-1709+1.82*Q3%
3230 IF Q2%<0 THEN Q2%=0
* 3240 PRINT CHR$(17);
* 3250 PRINT TAB(30);CL#;Q2%
* 3260 SYS P2,Q2%
* 3270 IF A=16 THEN Q3%=Q3%+Q% : REM UNTERARM AB
* 3280 IF A=19 THEN Q3%=Q3%-Q% : REM UNTERARM AUF
3290 REM ARBEITSRAUM UNTERARM
3300 IF Q3%>940+0.55*Q2% THEN Q3%=940+0.55*Q2%
3310 IF Q3%>1160 THEN Q3%=1160
3320 IF Q3%<-370+Q2% THEN Q3%=-370+Q2%
3330 IF Q3%<0 THEN Q3%=0
* 3340 PRINT CHR$(17);
* 3350 PRINT TAB(30);CL#;Q3%
* 3360 SYS P3,Q3%
3370 REM ROBOTERBEWEGUNG STARTEN
* 3380 SYS ROBOT
3390 REM ROUTINE ZANGE
* 3400 IF USR(E7)=0 THEN PRINT CHR$(17);TAB(30);"AUF
      "
* 3410 IF USR(E7)=1 THEN PRINT CHR$(17);TAB(30);"ZU
      "
* 3420 IF A<>24 THEN GOTO3480
3430 IF ZA#="AUF" THEN GOTO 3480
3440 LET ZA#="AUF"
* 3450 SYS M4,RECHTS
* 3460 IF USR(E7)=1 THEN GOTO3450
* 3470 SYS M4,AUS
* 3480 IF A<>27 THEN GOTO 3560
3490 IF ZA#="ZU " THEN GOTO 3560
3500 LET ZA#="ZU "
3510 FOR Z=1 TO 1.4*Z2
* 3520 SYS M4,LINKS
3530 NEXT
* 3540 SYS M4,AUS
3550 REM SCHRITTWEITE EINSTELLEN
* 3560 GET A#;REM TASTATURABFRAGE(KOMMANDOS)
3563 IF A#="" THEN IF ZZ<500 THEN LET ZZ=ZZ+50
3567 IF A#="--" THEN IF ZZ>50 THEN LET ZZ=ZZ-50
* 3570 IF A#=CHR$(133) THEN Q%=256
* 3580 IF A#=CHR$(134) THEN Q%=64
* 3590 IF A#=CHR$(135) THEN Q%=16
* 3600 IF A#=CHR$(136) THEN Q%=4
* 3610 PRINT CHR$(17);CHR$(17);
* 3620 PRINT TAB(30);CL#;Q%
* 3625 PRINT TAB(29);CL#;Z2
* 3630 IF A#=CHR$(19) THEN GOSUB 11000;REM HOME TAST
      E
3640 IF A#<>"M" THEN GOTO 3680; REM M =MENJETASTE
3650 LET IMAX=10
* 3660 PRINT CHR$(147)
3670 GOTO 1020
* 3680 IF A#<>CHR$(13) THEN GOTO 3850;REM LERNETASTE
3690 REM AKUST.SIGNAL 'ABSPEICHERN'
* 3700 S=54272
* 3710 POKE S+24,15
* 3720 POKE S+6,240
* 3730 POKE S+1,90
* 3740 POKE S+4,17
* 3750 POKE S+24,0;POKE S+1,0;POKE S+4,0;POKE S+6,0
3760 REM ABLAGE IN TEACH TABELLE
3770 LET ID=ID+1
* 3780 LET DR(ID)=USR(P1)
* 3790 LET OA(ID)=USR(P2)
* 3800 LET UA(ID)=USR(P3)
3810 LET ZA#(ID)=ZA#
3820 GOSUB 12010; REM AKTUELLE POSITION AUSDRUCKEN
3830 GOTO 3030
3840 REM LETZTE POSITION LOESCHEN
* 3850 IF A#<>CHR$(20) THEN GOTO 3030
3860 IF ID=0 THEN LET ID=ID-1
3870 REM AKUST. SIGNAL 'LOESCHEN'
* 3880 POKE S+24,15
* 3890 POKE S+6,240
* 3900 POKE S+1,70
* 3910 POKE S+4,17
* 3920 POKE S+24,0;POKE S+1,0;POKE S+4,0;POKE S+6,0
3930 GOSUB 12010; REM AKTUELLE POSITION AUSDRUCKEN
3940 GOTO 3030
4000 REM AUSFUEHRMODUS
4010 GOSUB 10000 : REM TITELMELDUNG
4020 FOR T=1 TO5
* 4030 PRINT CHR$(17)
4040 NEXT T
4050 PRINT"> M < MENUE"
* 4060 PRINT CHR$(19)
4070 FOR U =1 TO 5
* 4080 PRINT CHR$(17)
4090 NEXT U
4100 PRINT
4110 PRINT"      AUSFUEHRMODUS"
4120 PRINT
4130 INPUT"WIEVIELE DURCHLAEUFE ";D
4140 FOR Y=1 TO D
4150 GOSUB 11000; REM HOMEROUTINE
* 4160 PRINT CHR$(147)
4170 PRINT"PROGRAMMTABELLE"
4180 PRINT
4190 PRINT" NR DREH      OARM      UARM      ZANGE"
4200 PRINT
4210 FOR I=0 TO IMAX
* 4220 GET A#
4230 IF A#="M" THEN 1020
4240 PRINT I;TAB(3);DR(I);TAB(12);OA(I);TAB(21);UA
      (I);TAB(31);ZA#(I)
* 4245 11=USR(P1); IF 11<0 THEN LET 11=0
* 4250 SYS P1,11 : IF ABS(DR(I)-11)>10 THEN SYS P1,DR
      (I)
* 4255 12=USR(P2); IF 12<0 THEN LET 12=0
* 4260 SYS P2,12 : IF ABS(OA(I)-12)>10 THEN SYS P2,OA
      (I)
* 4265 13=USR(P3); IF 13<0 THEN LET 13=0
* 4270 SYS P3,13 : IF ABS(UA(I)-13)>10 THEN SYS P3,UA
      (I)
* 4280 SYS ROBOT
4290 REM ZANGENROUTINE
4300 IF ZA#(I)<>"ZU " THEN GOTO4360
4310 IF ZA#="ZU " THEN4360
4320 FOR Z=1 TO ZZ
* 4330 SYS M4,LINKS
4340 NEXT Z
4350 LET ZA#="ZU "
4360 IF ZA#(I)<>"AUF" THEN GOTO 4420
4370 IF ZA#="AUF" THEN 4420
* 4380 SYS M4,RECHTS
* 4390 IF USR(E7)=1 THEN GOTO4370
4400 LET ZA#="AUF"
* 4410 SYS M4,AUS
4420 NEXT I
4430 NEXT Y
4440 GOTO 1020
5000 REM TEACH PROGRAMM ABSPEICHERN
* 5010 PRINT CHR$(147)
5020 PRINT
5030 PRINT"TEACH PROGRAMM AUF DISKETTE ABSPEICHERN
      "
5040 PRINT

```

```

5050 LET F$=""
5060 INPUT"FILENAME";F$
5070 IF F$="" THEN GOTO 1020
* 5080 OPEN 15,8,15
* 5090 OPEN 8,8,8,F$+",.W"
* 5100 INPUT#15,FE,FT$,SP,SE
* 5110 IF FE=0 THEN GOTO 5200
* 5120 IF FE<>63 THEN GOTO 6250
* 5130 PRINT"FILE EXISTIERT BEREITS."
* 5140 INPUT"ALTES FILE LOESCHEN(J/N)";C$
* 5150 IF C$="N" THEN GOTO 6200
* 5160 IF C$<>"J" THEN 5270
* 5170 PRINT#15,"S:"+F$
* 5180 CLOSE 8
* 5190 OPEN 8,8,8,F$+",.W"
* 5200 PRINT#8,IMAX
5210 FOR I=0 TO IMAX
* 5220 PRINT#8,DR(I)
* 5230 PRINT#8,OA(I)
* 5240 PRINT#8,UA(I)
* 5250 PRINT#8,ZA$(I)
5260 NEXT I
* 5270 CLOSE 8
* 5280 CLOSE 15
5290 GOTO 1020
6000 REM TEACH PROGRAMM LADEN
* 6010 PRINT CHR$(147)
6020 PRINT
6030 PRINT"TEACH PROGRAMM VON DISKETTE LADEN"
6040 PRINT
6050 LET F$=""
6060 INPUT"FILENAME";F$
6070 IF F$="" THEN GOTO 1020
* 6080 OPEN 8,8,8,F$+",.R"
* 6090 OPEN 15,8,15
* 6100 INPUT#15,FE,FT$,SP,SE
* 6110 IF FE<>0 THEN GOTO 6250
* 6120 INPUT#8,IMAX
6130 PRINT IMAX;"POSITIONSDATEN"
6140 FOR I=0 TO IMAX
* 6150 INPUT#8,DR(I)
* 6160 INPUT#8,OA(I)
* 6170 INPUT#8,UA(I)
* 6180 INPUT#8,ZA$(I)
6190 NEXT I
* 6200 CLOSE 8
* 6210 CLOSE 15
6220 LET ID=IMAX:ZA$=ZA$(IMAX)
6230 GOTO 1020
6240 REM DISKETTE FEHLERMELDUNG
* 6250 PRINT FT$
6260 PRINT" M < MENUE"
* 6270 GET A$
6280 IF A$="M" THEN GOTO 6200
6290 GOTO 6270
7000 REM TEACH PROGRAMM AUSDRUCKEN
* 7010 PRINT CHR$(147)

```

```

7020 PRINT"SOLL DER AUSDRUCK AUF DRUCKER ODER "
7030 INPUT"BILDSCHIRM ERFOLGEN <D/B>";S$
7040 IF S$="D" THEN 7180
7050 GOSUB 10010
7060 PRINT
7070 PRINT"TEACH TABELLE"
7080 PRINT
7090 PRINT"NR. DREH OARM UARM ZANGE"
7100 FOR I=0 TO IMAX
7110 PRINT I;TAB(3);DR(I);TAB(12);OA(I);TAB(21);UA
(I);TAB(31);ZA$(I)
7120 NEXT I
7130 PRINT
7140 PRINT" M < MENUE"
* 7150 GET A$
7160 IF A$="M" THEN GOTO 1020
7170 GOTO 7150
* 7180 OPEN 4,4,0
* 7190 PRINT#4,"F I S C H E R T E C H N I K"
* 7200 PRINT#4,
* 7210 PRINT#4," C O M P U T I N G"
* 7220 PRINT#4,
* 7230 PRINT#4,"3 ACHSIGER TRAININGSROBOTER"
* 7240 PRINT#4,
* 7250 PRINT#4,"TEACH TABELLE"
* 7260 PRINT#4," "
* 7270 PRINT#4," NR. DREH OARM
UARM ZANGE"
7280 FOR I=0 TO IMAX
* 7290 PRINT#4,I,DR(I),OA(I),UA(I),ZA$(I)
7300 NEXT I
* 7310 CLOSE 4
7320 GOTO 1020
8000 REM PROGRAMMENDE
* 8010 PRINT CHR$(147)
8020 FOR I=1 TO 12
8030 PRINT
8040 NEXT I
8050 INPUT" SIND SIE SICHER (J/N)";L$
8060 IF L$="N" THEN 1020
8070 IF L$<>"J" THEN GOTO 8050
* 8080 PRINT CHR$(147)
8090 END
9000 REM DISKETTENINHALT
* 9010 PRINT CHR$(147)
9020 PRINT"FISCHERTECHNIK"
9030 PRINT"COMPUTING"
9040 PRINT
* 9050 OPEN 1,8,0,"#0"
* 9060 GET#1,A$,B$
* 9070 GET#1,A$,B$
* 9080 GET#1,A$,B$
* 9090 C=0 : C$=""
* 9100 IF A$<>"*" THEN C=ASC(A$)
* 9110 IF B$<>"*" THEN C=C+ASC(B$)*256
* 9120 PRINT MID$(STR$(C),2);TAB(3);
* 9130 GET#1,B$:IF ST<>0 THEN 9190

```

```

* 9140 IF B$<>CHR$(34) THEN 9130
* 9150 GET#1,B$:IF B$<>CHR$(34) THEN C$=C$+B$:GOTO 9
150
* 9160 GET#1,B$:IF B$<>"*" THEN 9160
* 9170 PRINT C$
* 9180 IF ST=0 THEN 9070
9190 PRINT" BLOCKS FREE"
* 9200 CLOSE 1
9210 PRINT
9220 PRINT" > M < MENUE"
* 9230 GET Z$
9240 IF Z$<>"M" THEN 9230
* 9250 PRINT CHR$(147)
9260 GOTO 1020
10000 REM TITELMELDUNG
* 10010 PRINT CHR$(147)
* 10020 PRINT CHR$(28)" F I S C H E R"CHR$(31)"
T E C H N I K"CHR$(144)
10030 PRINT
10040 PRINT" C O M P U T I N G"
10050 PRINT:PRINT
10060 PRINT" D R E I A C H S I G E R"
10070 PRINT
10080 PRINT" T R A I N I N G S R O B O T E R"
10090 PRINT
10100 PRINT" T E A C H I N V E R F A H R E N"
10110 RETURN
11000 REM HEIMPOSITION ANFAHREN
11010 LET H1=1:H2=1:H3=1:H4=1
* 11020 IFUSR(E1)=1 AND H1=1 THEN SYS M1,RECHTS
* 11030 IFUSR(E3)=1 AND H2=1 THEN SYS M2,RECHTS
* 11040 IFUSR(E5)=1 AND H3=1 THEN SYS M3,RECHTS
* 11050 IFUSR(E7)=1 AND H4=1 THEN SYS M4,RECHTS
* 11060 IFUSR(E1)=0 THEN SYS M1,LINKS:H1=-1
* 11070 IFUSR(E3)=0 THEN SYS M2,LINKS:H2=-1
* 11080 IFUSR(E5)=0 THEN SYS M3,LINKS:H3=-1
* 11090 IFUSR(E7)=0 THEN SYS M4,AUS:H4=0
* 11100 IFUSR(E1)=1 AND H1=-1 THEN SYS M1,AUS:H1=0
* 11110 IFUSR(E3)=1 AND H2=-1 THEN SYS M2,AUS:H2=0
* 11120 IFUSR(E5)=1 AND H3=-1 THEN SYS M3,AUS:H3=0
11130 IF H1<>0 OR H2<>0 OR H3<>0 OR H4<>0 THEN GOT
0 11020
* 11140 SYS INIT
11150 LET ZA$="AUF"
11160 RETURN
12000 REM AKTUELLE POSITION AUSDRUCKEN
* 12010 PRINT CHR$(19);
12020 FOR I=1 TO 23
* 12030 PRINT CHR$(17);
12040 NEXT I
12050 PRINT"
"
* 12060 PRINT CHR$(145);
12070 PRINT ID;TAB(3);DR(ID);TAB(12);OA(ID);TAB(21
);UA(ID);TAB(31);ZA$(ID)
12080 RETURN

```

Funktionsweise des Interface und des Roboter-Systemprogramms

Wenn Sie die fischertechnik computing Software benutzen oder selbst Programme entsprechend der Hinweise in den vorigen Kapiteln erstellen, werden Sie kaum die nun folgende Information benötigen. Wenn Sie aber die Programme in anderen Sprachen als BASIC formulieren wollen, die Programme durch komplexe Abläufe in Maschinensprache beschleunigen wollen, die Funktionen des Interface erweitern wollen oder auch nur einfach einen Blick hinter die Kulissen werfen wollen, so wird Ihnen das Nachfolgende sicherlich hilfreich sein. Allerdings sollten Sie dann auch ein paar Kenntnisse der Maschinensprache und der Digitalelektronik mitbringen, denn hier geht es an die "bits and pieces". Das fischertechnik Interface erfüllt eine Reihe von Aufgaben, die wir anhand des Blockdiagramms besprechen wollen. Am linken Rand sind die Signale von und zu dem Computer aufgeführt. Es fällt auf, daß diese recht wenig mit den Ausgängen M1 bis M4 und Eingängen E1 bis E8 sowie EX und EY gemein haben. Der Grund ist darin zu suchen, daß am Computeranschluß wesentlich weniger Datenleitungen zur Verfügung stehen, als auf der Modellseite des Interface benötigt werden. Diese wenigen Datenleitungen müssen deshalb so eingesetzt werden, daß alle Signale auf der Modellseite gesteuert werden können. Das Konzept sieht eine Mehrfachverwendung der Datenleitungen mit Hilfe von Schieberegistern vor. Auf diese Weise werden z.B. nur drei Datenleitungen für die Steuerung der Ausgabe notwendig. Eine parallele Anschlußweise hätte acht Datenleitungen benötigt.

Schauen wir uns gleich die Ausgabe an den Anschlüssen M1 bis M4 genauer an. Die dafür benötigten Datenleitungen werden mit DATA-OUT, CLOCK und LOAD-OUT bezeichnet. Bei einer Ausgabe werden immer die Daten für alle vier Motoren übertragen, d.h. ein ganzes Byte (ein Byte deswegen, weil jeder der vier Motoren zwei Bits zur Steuerung der Drehrichtung benötigt). Die von dem Kommando

nicht betroffenen Motorausgänge erhalten somit den derzeitigen Stand, der im Computer als Ausgabewort zwischengespeichert ist, erneut eingeschrieben.

Bei der Ausgabe werden der Reihe nach die Bits des Ausgabeworts an die Leitung DATA-OUT angelegt, das höchstwertige zuerst. Mit einem Übergang von low nach high am Ausgang CLOCK wird das Bit in ein Schieberegister übernommen. Danach folgt das nächste Bit an DATA-OUT, das ebenfalls in das Schieberegister mit dem nächsten CLOCK-Impuls übernommen wird. Das vorangegangene Bit ist dabei aber auch um eine Position im Schieberegister nach rechts gerutscht, um dem nachfolgenden Platz zu machen. Nach insgesamt acht solchen Datenübertragungen ist das ganze Ausgabewort im Schieberegister abgelegt. Das zuerst übertragene Bit ist im Verlaufe des Datentransfers ganz nach rechts durchgeschoben worden. Von der Aktivität im Schieberegister ist aber bislang an seinen Ausgängen noch nichts spürbar. Die Ausgangsverstärker werden nicht direkt über das Schieberegister angesteuert, sondern über ein zwischengeschaltetes Speicherregister, das auch noch im Schieberegister-Baustein integriert ist. Erst mit dem Übergang von low nach high am Ausgang LOAD-OUT erfolgt die Übernahme in das Speicherregister. Die zeitliche Abfolge der Signale können Sie dem Impulsdigramm entnehmen.

Ob die Daten allerdings auch die Leistungsverstärker durchsteuern, hängt wiederum von der Freigabesteuerung des Speicherbausteins ab. Die Freigabesteuerung erfolgt durch ein Monoflop. Diese Schaltung erzeugt ein Freigabesignal von einer halben Sekunde Dauer, wenn ein Impuls auf der CLOCK-Leitung vorliegt. Wir können davon ausgehen, daß zunächst die Leistungsverstärker angesteuert werden, da zuvor gerade die Daten mit Hilfe der CLOCK-Leitung übertragen wurden. Sollte aber innerhalb der nächsten halben Sekunde kein weite-

rer Datentransfer erfolgen, so kippt das Monoflop wieder in seinen stabilen Zustand zurück und das Freigabesignal wird zurückgenommen. Das Monoflop ist übrigens nachtriggerbar, d.h. die Zeitdauer von einer halben Sekunde rechnet sich jeweils vom Zeitpunkt des letzten CLOCK-Impulses an.

Auch das Monoflop besitzt einen Freigabeeingang. Über jenen kann letztlich die Ausgabe an die Verstärker sofort unterbunden werden. Beim fischertechnik Interface erfolgt dies, wenn ein ungültiges Datenmuster am Ausgang des Speicherregisters anliegen würde, das einen angeschlossenen Motor quasi in Rechts- und Linkslauf gleichermaßen steuern würde.

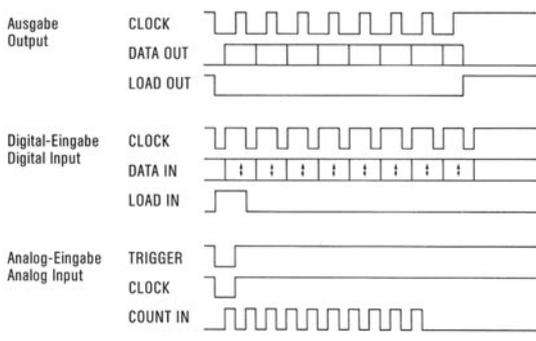
Nun zu der Übertragung der digitalen Signale an E1 bis E8. Im Prinzip findet bei der Eingabe eine Umkehrung des oben Beschriebenen statt. Durch das Ausgabe-Signal LOAD-IN werden die an den Eingängen anstehenden Signale in das Eingabeschieberegister übernommen. Dies erfolgt wiederum für alle acht Eingänge, auch wenn nur ein einziger abgefragt werden soll. In dem Schieberegister angelangt, bringt jeder Impuls auf der CLOCK-Leitung ein Bit auf der Eingabeleitung DATA-IN zum Vorschein, jenes von E8 zuerst und das von E1 zuletzt. Durch Testen dieser Leitung kann der Computer die Bits "aufsammeln" und wieder ein Datenwort bilden. Das gewünschte Bit wird anschließend herausgefiltert und dem BASIC-Programm übergeben.

Da zur Übertragung der Daten dieselbe CLOCK-Leitung wie bei der Ausgabe benutzt wird, wird auch bei der digitalen Eingabe das Monoflop aktiviert, das das Freigabesignal für die Ausgabedaten steuert. Eine Fehlfunktion des Ausgabeschieberegisters durch die Mehrfachfunktion der CLOCK-Leitung steht nicht zu befürchten, denn die aktuellen Ausgabedaten stehen ja nicht im Ausgabeschieberegister, sondern im Speicherregister. Ersteres wird zwar wohl durch die CLOCK-Impulse beeinflusst,

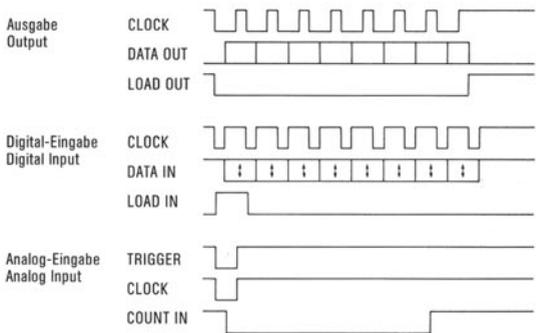
nicht aber letzteres, das ja nur auf das Signal LOAD-OUT reagiert.

Bleiben zum Schluß noch die Analogeingänge EX und EY. Potentiometer oder sonstige veränderlichen Widerstände dienen als zeitbestimmendes Bauelement in zwei weiteren Monoflop-Schaltungen. Ein niedriger Widerstandswert wird in einem Impuls kurzer Dauer, ein hoher Widerstandswert in einen Impuls langer Dauer umgesetzt. Der Impuls selbst wird durch Startsignal TRIGGER-X bzw. TRIGGER-Y (mit negativer Logik) ausgelöst und erscheint dann auf der Leitung COUNT-IN. Ein Maschinenprogramm stellt die Impulsdauer anhand der Zahl der Schleifendurchläufe fest, die während der Impulsdauer durchgeführt werden können. Diese Zahl wird in das aufrufende BASIC-Programm zurückgegeben. Sie sehen also, daß der Analogwert weder die Winkelstellung noch den Widerstandswert der Potentiometer darstellt. Dagegen geht die Arbeitsgeschwindigkeit des Prozessors ein. Dennoch besteht zwischen der letztlich ermittelten Zahl und dem Widerstandswert ein linearer Zusammenhang. Dieser muß gegebenenfalls im BASIC-Programm noch anhand einer Eichung in Winkelgrade oder Widerstandswerte umgerechnet werden.

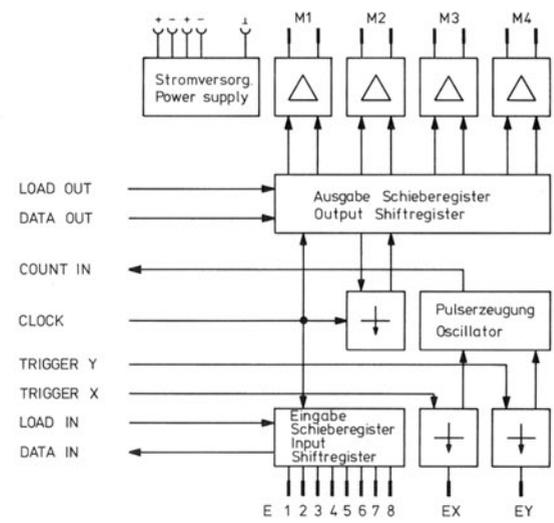
Auf den folgenden Seiten ist der Quelltext des Robotersystemprogramms angegeben. Aus Platzgründen können wir nicht alle Versionen des Roboter-Systemprogramms für die verschiedenen Computer abdrucken. Außerdem sind die Unterschiede nur geringfügig. Stellvertretend für Computer mit Mikroprozessoren der 6502-Familie geben wir hier das Roboter-Systemprogramm des Commodore 64 an. Ein anderer weitverbreiteter Mikroprozessor ist der Z80. Das Roboter-Systemprogramm des Schneider CPC steht stellvertretend für all jene Computer.



Impulsdiagramme des fischertechnik Interface
Pulse Diagrams of the fischertechnik Interface
C 64, VIC 20, Acorn



Impulsdiagramme des fischertechnik Interface
Pulse Diagrams of the fischertechnik Interface
IBM-PC, CPC, CBM 4/8 xxx



Prog. ROBOT.SYS (6502)

```

0010      ;PROGRAMM C64 INTERFACE
0020      ;COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
0030      ;VERSION 9 INKLUSIVE ROBOTERSTEUERUNG
0040      ;FILE C64IF9A
0050      ;MÄI 1985
0060      ;
0070      ;AUFRUF DES FISCHERTECHNIK INTERFACE
0080      ;VOM C64 DURCH KOMMANDOS
0090      ;SYS M1,EIN     SYS M1,AUS
0100      ;SYS M1,LINKS  SYS M1,RECHTS
0110      ;USR(E1)   USR(EX)   USR(EY)
0120      ;SPEZIELLE ROBOTERBEFEHLE:
0130      ;SYS P1,NNNN  SOLLPOSITION ABLEGEN
0140      ;USR(P1)  ISTPOSITION ABFRAGEN
0150      ;SYS ROBOT START DES ROBOTERS
0160      ;*****
0170      .OS      ;OBJECTCODE ERZEUGEN
0180      .BA #CDB0      ;PROGRAMM IM FREISPEICHER
0190      ;*****
0200 YFAC      .DE #B3A2      ;WANDLE Y IN FLIESSKOMMA
0210 AYFAC     .DE #B391      ;WANDLE A/Y IN FLIESSKOMMA
0220 CKCOM     .DE #AEFD      ;PRUEFE AUF KOMMA
0230 F16INT    .DE #B7F7      ;WANDLE FLIESSKOMMA IN INT
0240 GETBYTE   .DE #B79E      ;LIES EIN BYTE EIN
0250 GETWORD   .DE #AD8A      ;LIES EIN WORT EIN
0260 ROPOS     .DE #CE98      ;ROBOTER STEuern (TEIL B)
0270 ROPOSL    .DE #CFD0      ;ROBOTER TABELLE (TEIL B)
0280      ;*****
0290      ;EIN- AUSGABEREGISTER
0300      ;*****
0310 UP        .DE #DD01      ;USER PORT DATENREGISTER
0320 DRR       .DE #DD03      ;USER PORT DATENRICHTUNG
0330 TIL       .DE #DD04      ;TIMER LOW
0340 TIH       .DE #DD05      ;TIMER HIGH
0350 TIC       .DE #DD0E      ;TIMER CTRL REG
0360      ;*****
0370      ;VARIABLEN
0380      ;*****
CDB0- 00     0390 AVAR      .BY #00      ;AUSGABEVARIABLE
CDB1- 00     0400 MASK     .BY #00      ;MASKENVARIABLE
0410      ;*****
0420      ;EINSPRUNGLEISTE
CDB2- A9 00  0430 INIT     LDA #000      ;INITIALISIERUNG
CDB4- A2 17  0440      LDX #17      ;TABELLENZEIGER
CDB6- 9D D0 CF 0450 CLOOP   STA ROPOSL,X  ;TABELLEN DER
CDB9- CA      0460      DEX      ;ROBOTERPOSITIONEN
CDBA- 10 FA  0470      BPL CLOOP  ;LOESCHEN
CDBC- 30 2E  0480      BMI STVAR  ;BRANCH ALWAYS
CDBE- A9 03  0490 M1      LDA #00000011 ;MOTOR 1
CDC0- D0 0A  0500      BNE BOUT   ;
CDC2- A9 0C  0510 M2      LDA #00001100 ;MOTOR 2
CDC4- D0 05  0520      BNE BOUT   ;
CDC6- A9 30  0530 M3      LDA #00110000 ;MOTOR 3
CDC8- D0 02  0540      BNE BOUT   ;
CDCA- A9 C0  0550 M4      LDA #11000000 ;MOTOR 4
0560      ;*****
CDCC- 78     0570 BOUT     SEI      ;INTERRUPT SPERREN

CDCD- 9D B1 CD 0580
CDD0- 20 FD AE 0590
CDD3- AD B0 CD 0600
CDD6- 00 B1 CD 0610
CDD9- 9D B0 CD 0620
CDDC- 20 9E B7 0630
CDDF- 9A      0640
CDE0- 2D B1 CD 0650
CDE3- 8D B1 CD 0660
CDE6- AD B0 CD 0670
CDE9- 4D B1 CD 0680
COEC- 20 F1 CD 0690 STVAR
COEF- 58      0700
COF0- 60      0710
0720
0730
0740
0750
0760
0770
0780
COF1- 8D B0 CD 0790 SHOUT
COF4- 48      0800
COF5- A9 3F  0810
COF7- 9D 03 DD 0820
COFA- A2 08  0830
COFC- A9 30  0840 LOOP
COFE- 0E B0 CD 0850
CE01- 90 02  0860
CE03- 09 04  0870
CE05- 8D 01 DD 0880 NULL
CE08- 09 08  0890
CE0A- 8D 01 DD 0900
CE0D- CA      0910
CE0E- D0 EC  0920
CE10- A9 39  0930
CE12- 8D 01 DD 0940
CE15- 68      0950
CE16- 9D B0 CD 0960
CE19- 60      0970
0980
0990
1000
1010
CE1A- 78      1020 BINP
CE1B- 20 F7 B7 1030
CE1E- C9 00  1040
CE20- D0 76  1050
CE22- C0 A2  1060
CE24- F0 39  1070
CE26- C0 92  1080
CE28- F0 35  1090
CE2A- 8C B1 CD 1100
CE2D- 20 3D CE 1110
CE30- 2D B1 CD 1120
CE33- A8      1130
CE34- F0 02  1140

STA MASK      ;SPEICHERE BITMASKE AB
JSR CKCOM     ;PRUEFE AUF KOMMA
LDA AVAR      ;AUSGABEVARIABLE
ORA MASK      ;SETZE BIT
STA AVAR      ;ZURUECK
JSR GETBYTE   ;LIES 2. ARGUMENT
TXA
AND MASK      ;BLENDE MOTOR AUS
STA MASK      ;ABSPEICHERN
LDA AVAR      ;AUSGABEVARIABLE
EOR MASK      ;SETZE BIT
JSR SHOUT     ;AUSGABE AN INTERFACE
CLI           ;INTERRUPT FREIGEBEN
RTS           ;ZURUECK INS BASIC
;*****
;ROUTINE ZUR INTERFACESTEUERUNG
;AUSGABE
;AUSGABEMUSTER WIRD IM AKKU UEBERGEHEN
;SEITENEFFEKT: SPEICHERT AKKU IN AVAR AB
;BENUTZT AKKU UND X-REG
;*****
STA AVAR      ;AUSGABEWORT RETTEN
PHA          ;AKKU RETTEN
LDA #B3F      ;SETZE DATENRICHTUNG
STA DRR
LDX #008      ;SCHLEIFENZAehler
LDA #B30      ;RUHEBITMUSTER USERPORT
ASL AVAR      ;TESTE AUSGABEWORT
BCC NULL
ORA #B04      ;SETZE DATA OUT
STA UP        ;AUSGABE
ORA #B08      ;SETZE CLOCK
STA UP        ;SETZE GLOCK
DEX
BNE LOOP      ;SCHLEIFENENDE
LDA #B39      ;SETZE LOAD OUT
STA UP        ;AUSGABE
PLA          ;AKKU RESTAURIEREN
STA AVAR      ;RESTAURIERE AVAR
RTS          ;RUECKSPRUNG
;*****
;EINGABROUTINE
;EINSPRUNG UEBER USR
;*****
SEI           ;INTERRUPT SPERREN
JSR F16INT    ;ARG NACH INTEGER WANDELN
CMP #00      ;HIGH BYTE GESETZT
BNE ROPOS     ;ABFRAGE ROBOTERPOSITION
CPY #A2      ;POTIABFRAGE?
BEQ POT1     ;X
CPY #B2      ;POTIABFRAGE?
BEQ POT1     ;Y
STY MASK      ;INPUTMASKE ABSPEICHERN
JSR SHIN     ;EINGABE ROUTINE
AND MASK      ;BIT HERAUSMASKIEREN
TAY          ;WERT IN Y-REGISTER
BEQ CVAR

```

```

CE36- A0 01 1150 LDY #001 ;(<0 -> I 1720 .EN
CE38- 20 A2 B3 1160 CVAR JSR YFAC ;HANDLE Y IN FAC
CE3B- 58 1170 CLI ;INTERRUPT FREIGEBEN
CE3C- 60 1180 RTS ;ZURUECK INS BASIC
1190 ;*****
1200 ;INTERFACE STEUERUNG
1210 ;EINGABE
1220 ;BENUTZT WERDEN AKKU, X-REG. UND Y-REG.
1230 ;ARGUMENT BEI RUECKKEHR IM AKKU
1240 ;*****
CE3D- A9 32 1250 SHIN LDA #032 ;SETZE LOAD IN AVAR =CDB0 AYFAC =B391 BINP =CE1A
CE3F- 8D 01 DD 1260 STA UP ;AUSGABE BOUT =C0CC CKCOM =AEFD CLOOP =CDB6
CE42- 09 08 1270 ORA #008 ;SETZE CLOCK CVAR =CE38 DRR =DD03 F16INT =B7F7
CE44- 8D 01 DD 1280 STA UP ;AUSGABE GETBYTE =B79E GETWORD =AD8A INIT =CDB2
CE47- A2 08 1290 LDX #8 ;SCHLEIFENZAehler LOOP =C0FC LOOP2 =CE49 M1 =CDBE
CE49- 0A 1300 LOOP2 ASL A ;SCHIEBE AKKU HOCH M2 =C0C2 M3 =C0C6 M4 =C0CA
CE4A- 2C 01 DD 1310 BIT UP ;TESTE UP BIT7 MASK =CDB1 NULL =CE05 NULL2 =CE51
CE4D- 10 02 1320 BPL NULL2 POTT =CE5F ROPOS =CE98 ROPOSL =CFD0
CE4F- 09 01 1330 ORA #001 ;SETZE BIT SHIN =CE3D SHOUT =CDF1 STVAR =CDEC
CE51- A0 30 1340 NULL2 LDY #030 ;SETZE CLOCK TIC =DD0E TIH =DD05 TIL =DD04
CE53- 8C 01 DD 1350 STY UP ;AUSGABE TST =CE74 UP =DD01 VERZOEG =CE79
CE56- A0 38 1360 LDY #038 ;SETZE CLOCK YFAC =B3A2 //0000,CE98,CE98
CE58- 8C 01 DD 1370 STY UP ;AUSGABE
CE5B- CA 1380 DEX
CE5C- 00 EB 1390 BNE LOOP2 ;SCHLEIFENENDE
CE5E- 60 1400 RTS ;RUECKSPRUNG
1410 ;*****
1420 ;POTENTIOMETERABFRAGE
1430 ;HIERHER WIRD NACH USR VERZWEIGT
1440 ;WENN DAS ARGUMENT #92 ODER #A2 IST
1450 ;*****
CE5F- A9 FF 1460 POT1 LDA #0FF ;SETZE ZAEHLERREG. AUF #FF
CE61- 8D 04 DD 1470 STA TIL
CE64- 8D 05 DD 1480 STA TIH
CE67- A9 89 1490 LDA #089 ;SETZE TIMER CTRL REG
CE69- 8D 0E DD 1500 STA TIC
CE6C- 8C 01 DD 1510 STY UP ;MONOFLOP TRIGGERN
CE6F- A0 3A 1520 LDY #03A ;TRIGGER WEGNEHMEN
CE71- 8C 01 DD 1530 STY UP ;AUSGABE
CE74- A0 04 DD 1540 TST LDA TIL ;LADE ZAEHLER LOW BYTE
CE77- A2 03 1550 LDX #003 ;VERZOEGERUNGSSCHLEIFE
CE79- CA 1560 VERZOEG DEX
CE7A- D0 FD 1570 BNE VERZOEG
CE7C- 38 1580 SEC ;SUBTRAKTION
CE7D- ED 04 DD 1590 SBC TIL ;LAEUFT DER ZAEHLER NOCH
CE80- D0 F2 1600 BNE TST
CE82- A2 38 1610 LDX #038 ;SETZE CLOCK, LOESCHE LOAD
CE84- 8E 01 DD 1620 STX UP ;AUSGABE
CE87- 38 1630 SEC ;SUBTRAKTION VORBEREITEN
CE88- A9 FF 1640 LDA #0FF
CE8A- ED 04 DD 1650 SBC TIL ;BESTIMME DIFFERENZZEIT
CE8D- A8 1660 TAY
CE8E- A9 FF 1670 LDA #0FF
CE90- ED 05 DD 1680 SBC TIH ;HIGH BYTE
CE93- 20 91 B3 1690 JSR AYFAC ;HANDLE A/Y IN FAC
CE96- 58 1700 CLI ;INTERRUPT FREIGEBEN
CE97- 60 1710 RTS ;ZURUECK INS BASIC
0020 ;ROBOTER ROUTINEN
0030 ;
0040 ;FILE C64IF9B
0050 ;COPYRIGHT ARTUR FISCHER FORSCHUNG
0060 ;MAY 1985
0070 ;*****
0080 ;SPRUNGADRESSEN
0090 ;*****
0100 SHOUT .DE #CDF1 ;ADRESSE AUS TEIL A
0110 SHIN .DE #CE3D ;ADRESSE AUS TEIL A
0120 AYFAC .DE #B391 ;HANDLE A/Y IN REAL
0130 CKCOM .DE #AEFD ;PRUEFE AUF KOMMA
0140 GETWORD .DE #AD8A ;LIES 16 BIT INTEGER
0150 F16INT .DE #B7F7 ;HANDLE FAC IN 16 BIT INT.
0160 AVAR .DE #CDB0 ;ADRESSE AUS TEIL A
0170 MASK .DE #CDB1 ;ADRESSE AUS TEIL A
0180 ;*****
0190 ;ANFANGSADRESSE NACH TEIL A
0200 ;*****
0210 .OS ;OBJECTCODE ERZEUGEN
0220 .BA #CE98 ;ANFANGSADRESSE
0230 ;*****
0240 ;ROBOTERROUTINE
0250 ;
0260 ;ABFRAGE ISTWERT
0270 ;*****
CE98- A2 00 0280 ROPOS LDX #00 ;ZEIGER=0
CE9A- C0 C2 0290 CPY #L,P1 ;WELCHE POSITION
CE9C- F0 12 0300 BEQ LOP0S

```

```

CE9E- A2 02      0310      LDX #*02          ;ZEIGER=2
CEA0- C0 C6      0320      CPY #L,P2
CEA2- F0 0C      0330      BEQ LOPPOS
CEA4- A2 04      0340      LDX #*04          ;ZEIGER=4
CEA6- C0 CA      0350      CPY #L,P3
CEA8- F0 06      0360      BEQ LOPPOS
CEAA- A2 06      0370      LDX #*06          ;ZEIGER=6
CEAC- C0 CE      0380      CPY #L,P4
CEAE- D0 0B      0390      BNE SYNTAX
CEB0- BC D0 CF    0400      LDY ROPOSL,X      ;LADE LOW-BYTE
CEB2- 8D D1 CF    0410      LDA ROPOSH,X      ;LADE HIGH BYTE
CEB4- 20 91 B3    0420      JSR AYFAC         ;WANDLE IN REAL
CEB6- 58          0430      CLI              ;INTERRUPT FREIGEBEN
CEB8- 60          0440      RTS              ;ZURUECK INS BASIC
CEBB- 8C B0 CD    0450      STY AVAR
CEBE- 8D B1 CD    0460      STA MASK
CEC1- 60          0470      RTS
CEC2- 60          0480      ;*****
CEC3- 60          0490      ;ROBOTERROUTINE
CEC4- 60          0500      ;
CEC5- 60          0510      ;SETZE SOLLWERT FUER ROBOTERPOSITION
CEC6- 60          0520      ;*****
CEC7- 60          0530      ;*****
CEC8- 60          0540      ;*****
CEC9- 60          0550      ;*****
CECA- 60          0560      ;*****
CECB- 60          0570      ;*****
CECC- 60          0580      ;*****
CECD- 60          0590      ;*****
CECE- 60          0600      ;*****
CECF- 60          0610      ;*****
CEC0- 60          0620      ;*****
CEC1- 60          0630      ;*****
CEC2- 60          0640      ;*****
CEC3- 60          0650      ;*****
CEC4- 60          0660      ;*****
CEC5- 60          0670      ;*****
CEC6- 60          0680      ;*****
CEC7- 60          0690      ;*****
CEC8- 60          0700      ;*****
CEC9- 60          0710      ;*****
CECA- 60          0720      ;*****
CECB- 60          0730      ;*****
CECC- 60          0740      ;*****
CECD- 60          0750      ;*****
CECE- 60          0760      ;*****
CECF- 60          0770      ;*****
CEC0- 60          0780      ;*****
CEC1- 60          0790      ;*****
CEC2- 60          0800      ;*****
CEC3- 60          0810      ;*****
CEC4- 60          0820      ;*****
CEC5- 60          0830      ;*****
CEC6- 60          0840      ;*****
CEC7- 60          0850      ;*****
CEC8- 60          0860      ;*****
CEC9- 60          0870      ;*****
CEE7- A2 06      0860      LDX #6           ;SCHLEIFENZAehler
CEE9- 38          0870      SEC              ;SOLL-/ISTWERTVERGLEICH

CEEA- 8D D0 CF    0880      LDA ROPOSL,X     ; ISTWERT
CEED- FD D8 CF    0890      SBC ROSOLL,X     ; SOLLWERT
CEF0- 8D F0 CF    0900      STA SCRATCL      ; ZWISCHENSPEICHERN
CEF3- 8D D1 CF    0910      LDA ROPOSH,X     ; DTO.HIGHBYTE
CEF6- FD D9 CF    0920      SBC ROSOLH,X
CEF9- 8D F1 CF    0930      STA SCRATCH
CEFC- 10 05      0940      BPL PLUS
CEFE- 8D C9 CF    0950      LDA ML,X         ; MOTORENDEHRRICTUNG
CF01- D0 0B      0960      BNE NPOS         ; BRANCH ALWAYS
CF03- AD F0 CF    0970      LDA SCRATCL      ; UNTERSUCHE AUF NULL
CF06- 8D F1 CF    0980      ORA SCRATCH
CF09- F0 03      0990      BEQ NPOS
CF0B- 8D C8 CF    1000      LDA MR,X         ; MOTORDREHRRICTUNG
CF0E- 9D E0 CF    1010      STA AV,X         ; ABSPEICHERN
CF11- 9D E8 CF    1020      STA MD,X         ; MOTORRICTUNG ABSPEICHERN
CF14- CA         1030      DEX              ; SCHLEIFENZAehler
CF15- CA         1040      DEX
CF16- 10 D1      1050      BPL MDIR         ; SCHLEIFENENDE
CF18- 20 3D CE    1060      JSR SHIN         ; DIGITALEINGABE
CF1B- 8D F1 CF    1070      STA SCRATCH      ; PEGELANFANGSWERTE ABSP.
CF1C- 8D F1 CF    1080      ;*****
CF1D- 8D F1 CF    1090      ;EINLESEN DIGITALEINGABE
CF1E- 20 3D CE    1100      ;*****
CF1F- A8         1110      JSR SHIN         ; DIGITAL E.EINLESEN
CF21- A8         1120      TAY
CF22- 4D F1 CF    1130      EOR SCRATCH      ; DETEKTIERE FLANKEN
CF25- 8D F0 CF    1140      STA SCRATCL      ; SEKTORFLANKEN ZWISCHENSPEICHERN
CF28- 8C F1 CF    1150      STY SCRATCH      ; PEGEL ZWISCHENSPEICHERN
CF2B- A9 00      1160      LDA #0
CF2D- 8D B0 CD    1170      STA AVAR         ; AVAR VORBEREITEN
CF30- A2 06      1180      LDX #6           ; SCHLEIFE UEBER MOTOREN
CF31- 8D F1 CF    1190      ;*****
CF32- AD F1 CF    1200      ;SCHLEIFE UEBER ALLE MOTOREN
CF33- AD F1 CF    1210      ;*****
CF34- AD F1 CF    1220      ;DIGITALEINGABE
CF35- 3D C9 CF    1230      LDA SCRATCH      ; DIGITALEINGABE
CF36- D0 06      1240      ;*****
CF37- 9D E0 CF    1250      ;ENDTASTER TESTEN
CF38- D0 06      1260      ;*****
CF39- 9D E0 CF    1270      AND ML,X         ; ET HERAUSMASKIEREN
CF3A- 9D E0 CF    1280      BNE SEKTOR       ; ET NICHT AKTIV
CF3B- 9D E1 CF    1290      STA AV,X         ; MOTOR AUS
CF3C- 9D E1 CF    1300      STA NL,X         ; NACHLAUF AUF 0
CF3D- 9D E1 CF    1310      ;*****
CF3E- 9D E1 CF    1320      ;SEKTORENFLANKEN TESTEN
CF3F- 9D E1 CF    1330      ;*****
CF40- AD F0 CF    1340      LDA SCRATCL      ; DIGITALEINGABE
CF43- 3D C9 CF    1350      AND MR,X         ; SEKTOR HERAUSMASKIEREN
CF46- F0 4C      1360      BEQ NEXCOMP      ; SEKTOR KEINE FLANKE
CF48- 8D C8 CF    1370      LDA MD,X
CF4E- D0 14      1380      CMP MR,X         ; LINKSLAUF?
CF50- 38         1390      BNE INCPOS
CF51- 8D D0 CF    1400      SEC
CF54- E9 01      1410      LDA ROPOSL,X     ; ROBÖTER POSITION -1
CF56- 9D D0 CF    1420      SBC #1
CF59- 8D D1 CF    1430      STA ROPOSL,X
CF5C- E9 00      1440      LDA ROPOSH,X
CF5D- E9 00      1440      SBC #0

```

```

CF5E- 9D D1 CF 1450 STA ROPOSH,X
CF61- 38 1460 SEC
CF62- B0 11 1470 BCS NEXSK ;BRANCH ALWAYS
CF64- 18 1480 INCPOS CLC
CF65- B0 D0 CF 1490 LDA ROPOSL,X ;ROBOTER POSITION +1
CF68- 69 01 1500 ADC #1
CF6A- 9D D0 CF 1510 STA ROPOSL,X
CF6D- BD D1 CF 1520 LDA ROPOSH,X
CF70- 69 00 1530 ADC #0
CF72- 9D D1 CF 1540 STA ROPOSH,X
CF75- BD D0 CF 1550 LDA ROPOSL,X ;IST-/SOLLWERT VERGLEICHEN
CF78- DD D8 CF 1560 CMP ROSOLL,X
CF7B- D0 17 1570 BNE NEXCOMP
CF7D- BD D1 CF 1580 LDA ROPOSH,X
CF80- DD D9 CF 1590 CMP ROSOLH,X
CF83- D0 0F 1600 BNE NEXCOMP
CF85- A9 00 1610 LDA #0 ;MOTOR ABSCHALTEN
CF87- DD E0 CF 1620 CMP AV,X ;SOLL = IST DAS ERSTE MAL?
CF8A- F0 08 1630 BEQ NEXCOMP
CF8C- 9D E0 CF 1640 STA AV,X ;MOTOR AUS
CF8F- A9 FF 1650 LDA #FF
CF91- 9D E1 CF 1660 STA NL,X ;NACHLAUFZAEHLER SETZEN
CF94- A9 00 1670 NEXCOMP LDA #0 ;NACHLAUFZAEHLER TESTEN
CF96- DD E1 CF 1680 CMP NL,X
CF99- F0 03 1690 BEQ OUT ;NICHT DEKREMENTIEREN, WENN
CF9B- DE E1 CF 1700 DEC NL,X ;NACHLAUFZ. DEKR.
CF9E- AD B0 CD 1710 OUT LDA AVAR ;AVAR ZUSAMMENSETZEN
CFA1- 1D E0 CF 1720 ORA AV,X ;AUS AV DER EINZELNEN MOT.
CFA4- 8D B0 CD 1730 STA AVAR ;ZURUECKSPEICHERN
CFA7- CA 1740 DEX ;SCHLEIFENZAEBLER DEKR.
CFA8- CA 1750 DEX
CFA9- 10 87 1760 BPL LOOPHEAD ;SCHLEIFENENDE
CFAB- AD B0 CD 1770 LDA AVAR ;AUSGABE ANS INTERFACE
CFAE- 20 F1 CD 1780 JSR SHOUT
CFB1- F0 03 1790 BEQ NLTST ;ALLE MOTOREN AUS
CFB3- 4C 1E CF 1800 JMP DIGIN
CFB6- A2 06 1810 NLTST LDX #006 ;ALLE NACHLAUFZAEHLER
CFB8- 1D E1 CF 1820 TSTNL ORA NL,X ;NOCH TESTEN
CFBB- CA 1830 DEX ;SCHLEIFENZAEBLER
CFBC- CA 1840 DEX
CFBD- 10 F9 1850 BPL TSTNL
CFBF- C9 00 1860 CMP #000 ;ALLE ABGELAUFEN?
CFC1- F0 03 1870 BEQ END ;SCHON ALLE FERTIG
CFC3- 4C 1E CF 1880 JMP DIGIN
CFC6- 58 1890 END CLI ;INTERRUPT FREIGEBEN
CFC7- 60 1900 RTS ;ZURUECK INS BASIC
CFC8- 02 1910 MR .BY %00000010 ;MOTOR1 RECHTS
CFC9- 01 1920 ML .BY %00000001 ;MOTOR1 LINKS
CFCA- 00 1930 .BY %00001000 ;MOTOR2 RECHTS
CFCB- 04 1940 .BY %00000100 ;MOTOR2 LINKS
CFCC- 20 1950 .BY %01000000 ;MOTOR3 RECHTS
CFCD- 10 1960 .BY %00010000 ;MOTOR3 LINKS
CFCE- 80 1970 .BY %10000000 ;MOTOR4 RECHTS
CFCF- 40 1980 .BY %01000000 ;MOTOR4 LINKS
CFD0- 00 1990 ROPOSL .BY 0
CFD1- 00 2000 ROPOSH .BY 0
CFD2- 2010 .DS 6

```

```

CFD8- 00 2020 ROSOLL .BY 0
CFD9- 00 2030 ROSOLH .BY 0
CFDA- 00 2040 .DS 6
CFDE- 00 2050 AV .BY 0 ;AUSGABEVARIABLE
CFE1- 00 2060 NL .BY 0 ;NACHLAUFZAEHLER
CFE2- 00 2070 .DS 6 ;VERZAHNTE TABELLE
CFE8- 00 2080 MD .BY 0 ;MOTORRICHTUNG
CFE9- 00 2090 .BY 0
CFEA- 00 2100 .DS 6
CFF0- 00 2110 SCRATCL .BY 0
CFF1- 00 2120 SCRATCH .BY 0
2130 .EN

```

--- LABEL FILE: ---

```

AV =CFE0 AVAR =CDB0 AYFAC =B391
CKCOM =AEFD DIGIN =CF1E END =CFC6
F16INT =B7F7 GETWORD =AD8A INCPOS =CF64
LOOPHEAD =CF32 LOPOS =CEB0 MASK =CDB1
MD =CFE8 MDIR =CEE9 ML =CFC9
MR =CFC8 NEXCOMP =CF94 NEXSK =CF75
NL =CFE1 NLTST =CFB6 NPOS =CF0E
OUT =CF9E P1 =CEC2 P2 =CEC6
P3 =CECA P4 =CECE PLUS =CF03
ROBOT =CEE7 ROPOS =CE98 ROPOSH =CFD1
ROPOSL =CFD0 ROSOLH =CFD8 ROSOLL =CFD8
SCRATCH =CFF1 SCRATCL =CFF0 SEKTOR =CF40
SHIN =CE3D SHOUT =CDF1 STOPOS =CED0
SYNTAX =CEBB TSTNL =CFB8
//0000.CFF2,CFF2
I

```

Prog. ROBOT.SYS (Z80)

Pass 1 errors: 00

```

10 ;Pogramm Schneider CPC464 Interface
20 ;Copyright (C) Artur Fischer Forschung
30 ;Version 2 inklusive Robotersteuerung
40 ;File ROSYS.GEN
50 ;August 1985
60 ;
70 ;Aufruf des fischertechnik Interface
80 ;vom CPC durch Kommandos:
90 ;CALL m1,ein      CALL m1,aus
100 ;CALL m1,links  CALL m1,rechts
110 ;CALL in,@e1    CALL in,@ex
120 ;Statt m1 kann auch m2, m3 und m4 benutzt werden.
130 ;Statt e1 kann auch e2, e3 bis e8 benutzt werden.
140 ;Statt ex kann auch ey benutzt werden.
150 ;
160 ;Spezielle Roboter-Befehle:
170 ;CALL p1,nnnn   Sollposition ablegen
180 ;CALL in,@i1   Istposition abfragen
190 ;Statt p1 kann auch p2, p3 und p4 benutzt werden.
200 ;Statt i1 kann auch i2, i3 und i4 benutzt werden.
210 ;CALL robot    Start des Roboters
220 ;*****
A400 230      org #a400      ;Programmstart
240 ;*****
A400 250      INIT: CALL SYNT0      ;Syntax-Pruefung
A403 260      LD HL,ROPSL        ;Tabellenzeiger
A406 270      XOR A              ;Akku loeschen
A407 280      LD B,#17           ;Schleifenzaehler
A409 290      LOOP1: LD (HL),A   ;Robotertabellen loeschen
A40A 300      INC HL
A40B 310      DJNZ LOOP1        ;Schleifenende
A40D 320      LD A,#00          ;Ausgabevariable loeschen
A40F 330      JR STVAR
A411 340 M1:  LD B,#M03         ;Motor 1
A413 350      JR BOUT
A415 360 M2:  LD B,#M0C         ;Motor 2
A417 370      JR BOUT
A419 380 M3:  LD B,#M30         ;Motor 3
A41B 390      JR BOUT
A41D 400 M4:  LD B,#M0C         ;Motor 4
A41F 410      BOUT: CALL SYNT1   ;Syntax-Pruefung
420 ;*****
430 ;Einzelbit Ausgabe
440 ;*****
A422 450      LD A,(AVAR)       ;Ausgabevariable
A425 460      OR B              ;setze Bits
A426 470      LD C,A
A427 480      LD A,(IX+0)
A42A 490      AND B
A42B 47      LD B,A
A42C 79      LD A,C
A42D A8      XOR B              ;setze Drehrichtung
A42E 320EA5 530 STVAR: LD (AVAR),A ;setze Ausgabevariable
A431 CD36A4 540 CALL SHOUT        ;Ausgabe an Interface
A434 FB      550 EI              ;Interrupt freigeben
A435 C9      560 RET              ;Ruecksprung in BASIC

570 ;*****
580 ;Routine zur Interface-Steuerung
590 ;Ausgabe
600 ;Ausgabemuster wird im Akku uebergeben
610 ;benutzt A, BC, DE.
620 ;*****
A436 630      SHOUT: LD BC,#EF00 ;Zeiger in Drucker Port
A439 640      LD C,A            ;c ist Arbeitsregister
A43A 650      LD E,#08         ;Schleifenzaehler
A43C 660      LOOP: LD D,#30   ;Ruhepegel Interface
A43E 670      LD A,C
A43F 680      RLCA
A440 690      LD C,A
A441 3002    700 JR NC,NULL      ;=0?
A443 1634    710 LD D,#34       ;setze DATA OUT
A445 ED51    720 NULL: OUT (C),D ;Ausgabe
A447 7A      730 LD A,D
A448 F608    740 OR #08         ;setze CLOCK
A44A ED79    750 OUT (C),A      ;Ausgabe
A44C 1D      760 DEC E          ;Schleifenzaehler
A44D 20ED    770 JR NZ,LOOP     ;Schleifenende
A44F 1639    780 LD D,#39       ;setze LOAD OUT
A451 ED51    790 OUT (C),D      ;Ausgabe
A453 C9      800 RET

810 ;*****
820 ;Eingaberoutine
830 ;Kommando in,@en
840 ;*****
A454 CD04A5 850 inp: CALL SYNT1        ;Syntax-Pruefung
A457 2185AE 860 LD HL,#AE85         ;Variablen-Speicher
A45A 4E      870 LD C,(HL)        ;Zeiger in Variablen-Sp.
A45B 23      880 INC HL
A45C 46      890 LD B,(HL)
A45D 210500 900 LD HL,#0005
A460 09      910 ADD HL,BC          ;Adresse von E1
A461 DD5601 920 LD D,(IX+1)
A464 DD5E00 930 LD E,(IX+0)
A467 010100 940 LD BC,#0001
A46A 7C      950 VARTST: LD A,H ;Bit-Zaehler
A46B BA      960 CF D          ;vergleiche Adressen
A46C 2004    970 JR NZ,NEXTVAR ;high-Byte
A46E 7D      980 LD A,L
A46F BB      990 CF E          ;low-Byte
A470 2813    1000 JR Z,VARFOUND ;Variable gefunden
A472 C5      1010 NEXTVA: PUSH BC
A473 010700 1020 LD BC,#0007
A476 09      1030 ADD HL,BC
A477 C1      1040 POP BC
A478 79      1050 LD A,C
A479 17      1060 RLA
A47A 4F      1070 LD C,A
A47B 78      1080 LD A,B
A47C 17      1090 RLA
A47D 47      1100 LD B,A
A47E FE40    1110 CP #40
A480 CA08A5 1120 JP Z,SYNTAX   ;Variablenabelle zu Ende
A483 18E5    1130 JR VARTST    ;Syntax-Fehler
A485 78      1140 VARFOU: LD A,B ;weiter suchen
;Analogabfrage?

```

```

A486 FE04 1150 CP #04
A488 D20FA5 1160 JP NC,ROPOS
A48B FE01 1170 CP #01
A48D 2844 1180 JR Z,XPOTI ;Eingang EX
A48F FE02 1190 CP #02
A491 2944 1200 JR Z,YPOTI ;Eingang EY
A493 C5 1210 PUSH BC ;rette BC
A494 CD99A4 1220 CALL SHIN ;Digitaleingabe
A497 182F 1230 JR CONT
1240 ;*****
1250 ;Routine zur Interface-Steuerung
1260 ;Eingabe
1270 ;benutzt A, BC und DE.
1280 ;Eingabe wird im Akku uebergeben
1290 ;*****
A499 1632 1300 SHIN: LD D,#32 ;setze LOAD IN
A49B 0100EF 1310 LD BC,#EF00 ;Zeiger in Drucker-Port
A49E ED51 1320 OUT (C),D ;Ausgabe
A4A0 163A 1330 LD D,#3A ;setze CLOCK
A4A2 ED51 1340 OUT (C),D ;Ausgabe
A4A4 1E08 1350 LD E,#08 ;Schleifenzaeher
A4A6 17 1360 LOOP2: RLA ;Datenwort hochschieben
A4A7 E6FE 1370 AND #FE ;Bit 0 loeschen
A4A9 0100F5 1380 LD BC,#F500 ;Zeiger auf BUSY-Input
A4AC 4F 1390 LD C,A
A4AD ED78 1400 IN A,(C) ;einlesen
A4AF E640 1410 AND #40 ;Busy-Leitung maskieren
A4B1 17 1420 RLA ;und zum Testen in Carry
A4B2 17 1430 RLA
A4B3 79 1440 LD A,C
A4B4 3002 1450 JR NC,NULL2
A4B6 F601 1460 OR #01 ;teste DATA-IN
A4B8 0100EF 1470 NULL2: LD BC,#EF00 ;setze Bit 0
A4BB 1630 1480 LD D,#30 ;Zeiger in Drucker-Port
A4BD ED51 1490 OUT (C),D ;loesche CLOCK
A4BF 1638 1500 LD D,#38 ;Ausgabe
A4C1 ED51 1510 OUT (C),D ;setze CLOCK
A4C3 1D 1520 DEC E ;Ausgabe
A4C4 20E0 1530 JR NZ,LOOP2 ;Schleifenzaeher
A4C6 2F 1540 CPL ;Schleifenende
A4C7 C9 1550 RET ;negative Logik!
1560 ;*****
A4C8 C1 1570 CONT: POP BC ;BC restaurieren
A4C9 A1 1580 AND C
A4CA 2802 1590 JR Z,BASRET
A4CC 3E01 1600 LD A,#01 ;(<0 -> 1
A4CE 77 1610 BASRET: LD (HL),A ;in Variablentabelle
A4CF 23 1620 INC HL
A4D0 70 1630 LD (HL),B
A4D1 FB 1640 EI ;Interrupt freigeben
A4D2 C9 1650 RET ;zurueck in BASIC
1660 ;*****
1670 ;Analogeingabe
1680 ;hierher wird verzweigt, wenn ex oder ey
1690 ;abgefragt werden soll.
1700 ;*****
A4D3 16A0 1710 XPOTI: LD D,#A0 ;setze TRIGGER-X
A4D5 1802 1720 JR POTI
A4D7 1690 1730 YPOTI: LD D,#90 ;setze TRIGGER-Y
A4D9 0100EF 1740 POTI: LD BC,#EF00 ;Zeiger in Drucker-Port
A4DC ED51 1750 OUT (C),D ;Ausgabe
A4DE 1638 1760 LD D,#38 ;setze CLOCK
A4E0 ED51 1770 OUT (C),D ;Ausgabe
A4E2 0100F5 1780 LD BC,#F500 ;Zeiger auf BUSY-Input
A4E5 110000 1790 LD DE,#0000 ;Zaeher auf Null
A4E8 ED78 1800 LOOP3: IN A,(C) ;Zaeher COUNT IN
A4EA 17 1810 RLA ;lies COUNT IN
A4EB 17 1820 RLA ;in Carry schieben
A4EC 3804 1830 JR C,STOP ;zum Testen
A4EE 1C 1840 INC E ;Puls zu Ende?
A4EF 20F7 1850 JR NZ,LOOP3 ;Zaeher erhoehen
A4F1 1D 1860 DEC E ;weiterzaehlen
A4F2 73 1870 STOP: LD (HL),E ;Ueberlauf -> 255
A4F3 23 1880 INC HL ;in Variable ablegen
A4F4 72 1890 LD (HL),D
A4F5 0100EF 1900 LD BC,#EF00
A4F8 1638 1910 LD D,#38 ;setze CLOCK
A4FA ED51 1920 OUT (C),D ;Ausgabe
A4FC FB 1930 EI ;Interrupt freigeben
A4FD C9 1940 RET ;zurueck in BASIC
1950 ;*****
1960 ;Syntax Pruefroutine
1970 ;*****
A4FE FE00 1980 SYNT0: CP #00 ;CALL INIT, ROBOT 0 Arg.
A500 F3 1990 DI ;Interrupt sperren
A501 C8 2000 RET Z
A502 1804 2010 JR SYNTAX ;Fehlermeldung
A504 FE01 2020 SYNT1: CP #01 ;CALL Mn,Richt. ein Arg.
A506 F3 2030 DI ;Interrupt sperren
A507 C8 2040 RET Z
A508 CD00B9 2050 SYNTAX: CALL #B900 ;ROM einschalten
A50B C3C6DD 2060 JP #DDC6 ;Fehlermeldung drucken
A50E 00 2070 AVAR: DEFB #00 ;Ausgabewort
2080 ;*****
2090 ;Roboterroutinen
2100 ;
2110 ;Copyright (C) Artur Fischer Forschung
2120 ;August 1985
2130 ;
2140 ;Abfrage Istwert der Roboterposition
2150 ;*****
A50F EB 2160 ROPOS: EX DE,HL
A510 212CA6 2170 LD HL,ROPOS ;Zeiger Sollwerte
A513 1F 2180 RRA
A514 1F 2190 RRA
A515 1F 2200 LOOP4: RRA
A516 3804 2210 JR C,CONT1
A518 23 2220 INC HL ;Zeiger weiterschalten
A519 23 2230 INC HL
A51A 18F9 2240 JR LOOP4
A51C 7E 2250 CONT1: LD A,(HL) ;lade low-Byte
A51D 12 2260 LD (DE),A ;in Variable ablegen
A51E 23 2270 INC HL ;high-Byte
A51F 13 2280 INC DE
A520 7E 2290 LD A,(HL)
A521 12 2300 LD (DE),A

```

```

A522 FB      2310      EI          ;Interrupt freigeben
A523 C9      2320      RET         ;zurueck in BASIC
                2330      ;*****
                2340      ;Roboterroutine
                2350      ;
                2360      ;Setze Sollwerte der Roboterposition
                2370      ;*****
A524 OE00    2380 P1:   LD      C,#00      ;Zeiger=0
A526 180A    2390      JR      STOPS
A528 OE02    2400 P2:   LD      C,#02      ;Zeiger=2
A52A 1806    2410      JR      STOPS
A52C OE04    2420 P3:   LD      C,#04      ;Zeiger=4
A52E 1802    2430      JR      STOPS
A530 OE06    2440 P4:   LD      C,#06      ;Zeiger=6
A532 0600    2450 STOPS: LD      B,#00
A534 CD04A5  2460      CALL SYNT1      ;Syntax-Pruefung
A537 2134A6  2470      LD      HL,ROSOLL      ;Sollwert-Tabelle
A53A 09      2480      ADD     HL,BC          ;Zeiger addieren
A53B DD7E00  2490      LD      A,(IX)        ;Argument holen
A53E 77      2500      LD      (HL),A        ;abspeichern
A53F 23      2510      INC     HL          ;high-Byte
A540 DD7E01  2520      LD      A,(IX+#01)
A543 77      2530      LD      (HL),A
A544 FB      2540      EI          ;Interrupt freigeben
A545 C9      2550      RET         ;zurueck in BASIC
                2560      ;*****
                2570      ;Robotersteuerung
                2580      ;
                2590      ;Die Routine vergleicht die in ROSOL
                2600      ;abgespeicherten Werte mit jenen in
                2610      ;ROPOS. Aus der Differenz wird die
                2620      ;Motordrehrichtung bestimmt.
                2630      ;Die Motoren mit Differenz <0> werden
                2640      ;gestartet und die Impulse der Licht-
                2650      ;schranke gezaehlt. ROPOS wird
                2660      ;entsprechend weitergezaehlt.
                2670      ;Wenn ROPOS=ROSOL wird der Motor ab-
                2680      ;gestellt, die Impulse jedoch weiter-
                2690      ;gezaehlt. Wenn alle Achsen zur Ruhe
                2700      ;gekommen sind, wird in Basic zurueck-
                2710      ;gegeben.
                2720      ;*****
A546 0603    2730 ROBOT: LD      B,#03      ;Schleifenzaehler
A548 CDFEA4  2740      CALL SYNT0      ;Syntax-Pruefung
A54B DD2124A6 2750      LD      IX,MR
A54F DD6E08  2760 MDIR: LD      L,(IX+#08)      ;ROPOS (Istwert)
A552 DD6609  2770      LD      H,(IX+#09)
A555 DD5E10  2780      LD      E,(IX+#10)      ;ROSQL (Sollwert)
A558 DD5611  2790      LD      D,(IX+#11)
A55B A7      2800      AND     A          ;Carry loeschen
A55C ED52    2810      SBC     HL,DE        ;Differenz
A55E DD7528  2820      LD      (IX+#28),L      ;abspeichern (SCRATCH)
A561 DD7429  2830      LD      (IX+#29),H
A564 3005    2840      JR      NC,PLUS      ;Differenz>0
A566 DD7E01  2850      LD      A,(IX+#01)
A569 180B    2860      JR      NPOS
A56B DD7E28  2870 PLUS: LD      A,(IX+#28)      ;Differenz=0?
A56E DDB629  2880      OR      (IX+#29)
                2890
A571 2803    2890      JR      Z,NPOS
A573 DD7E00  2900      LD      A,(IX+#00)      ;Rechtslauf
A576 DD7718  2910 NPOS: LD      (IX+#18),A      ;Teil-Ausgabewort (AV)
A579 DD7720  2920      LD      (IX+#20),A      ;Drehrichtung abspeichern
A57C DD23    2930      INC     IX          ;Zeiger weiter
A57E DD23    2940      INC     IX
A580 10CD    2950      DJNZ  MDIR          ;Schleifenende
A582 DD2124A6 2960      LD      IX,MR
A586 214DA6  2970      LD      HL,SCRATCH      ;Zeiger Drehrichtungen
A589 CD99A4  2980      CALL SHIN          ;Zeiger auf SCRATCH
A58C 77      2990      LD      (HL),A        ;Digital-Eingabe
                3000      ;*****
                3010      ;Einlesen der Digitaleingange
                3020      ;*****
A58D CD99A4  3030 DIGIN: CALL SHIN          ;Digital-Eingabe
A590 4E      3040      LD      C,(HL)        ;Anfangswerte
A591 77      3050      LD      (HL),A        ;neues Bitmuster
A592 A9      3060      XOR     C          ;erkenne Flanken
A593 2B      3070      DEC     HL
A594 77      3080      LD      (HL),A
A595 23      3090      INC     HL
A596 AF      3100      XOR     A          ;Akku loeschen
A597 320EA5  3110      LD      (AVAR),A      ;Ausgabewort loeschen
A59A 0603    3120      LD      B,#03        ;Schleifenzaehler
A59C DD2124A6 3130      LD      IX,MR
                3140      ;*****
                3150      ;Schleife ueber alle Motoren
                3160      ;*****
A5A0 7E      3170 LOOPH: LD      A,(HL)      ;Schleifenkopf
                3180      ;*****
                3190      ;Endtaster testen
                3200      ;*****
A5A1 DDA601  3210      AND     (IX+#01)      ;Endtaster maskieren
A5A4 2006    3220      JR      NZ,SECTOR      ;betaetigt?
A5A6 DD7718  3230      LD      (IX+#18),A      ;Motor aus
A5A9 DD7719  3240      LD      (IX+#19),A      ;Nachlauf=0
                3250      ;*****
                3260      ;Sektorflanken testen
                3270      ;*****
A5AC 2B      3280 SECTOR: DEC     HL
A5AD 7E      3290      LD      A,(HL)        ;Sektor maskieren
A5AE 23      3300      INC     HL
A5AF DDA600  3310      AND     (IX+0)
A5B2 2831    3320      JR      Z,NEXCOM
A5B4 DD5E08  3330      LD      E,(IX+#08)
A5B7 DD5609  3340      LD      D,(IX+#09)
A5BA DD7E20  3350      LD      A,(IX+#20)
A5BD DDBE01  3360      CP      (IX+#01)
A5C0 2803    3370      JR      Z,INCPPOS
A5C2 1B      3380      DEC     DE          ;Roboter Position -1
A5C3 1801    3390      JR      NEXSK
A5C5 13      3400 INCPOS: INC     DE          ;Roboter Position +1
A5C6 DD7308  3410 NEXSK: LD      (IX+#08),E      ;Roboter Position absp.
A5C9 DD7209  3420      LD      (IX+#09),D
A5CC 7B      3430      LD      A,E
A5CD DDBE10  3440      CP      (IX+#10)
A5D0 2013    3450      JR      NZ,NEXCOM
A5D2 7A      3460      LD      A,D

```

```

A5D3  DDBE11  3470      CP  (IX+#11)
A5D4  200D    3480      JR  NZ,NEXCOM
A5D8  AF      3490      XOR  A
;Akku loeschen
A5D9  DDBE18  3500      CP  (IX+#18)
;Soll=Ist das 1. Mal?
A5DC  2807    3510      JR  Z,NEXCOM
A5DE  DD7718  3520      LD  (IX+#18),A
;Motor ausschalten
A5E1  3D      3530      DEC  A
;Akku=FF
A5E2  DD7719  3540      LD  (IX+#19),A
;Nachlaufzaehler setzen
A5E5  AF      3550  NEXCOM: XOR  A
;Akku loeschen
A5E6  DDBE19  3560      CP  (IX+#19)
;Nachlaufzaehler testen
A5E9  2803    3570      JR  Z,OUT
;Nicht dekr. wenn 0
A5EB  DD3519  3580      DEC  (IX+#19)
;Nachlaufzaehler dekr.
A5EE  3A0EA5  3590  OUT:  LD  A,(AVAR)
;Ausgabewort aus AV
A5F1  DDB618  3600      OR  (IX+#18)
;der einzelnen Motoren
A5F4  320EA5  3610      LD  (AVAR),A
;abspeichern
A5F7  DD23    3620      INC  IX
;Tabellenzeiger 2 Bytes
A5F9  DD23    3630      INC  IX
;weiter
A5FB  10A3    3640      DJNZ LOOPH
;Schleifenende
A5FD  DD2124A6 3650      LD  IX,MR
;Zeiger zuruecksetzen
A601  3A0EA5  3660      LD  A,(AVAR)
;Ausgabewort
A604  F5      3670      PUSH AF
;Register retten
A605  CD36A4  3680      CALL SHOUT
;Ausgabe ans Interface
A608  F1      3690      POP  AF
;Register restaurieren
A609  FE00    3700      CP  #00
;alle Motoren aus?
A60B  2803    3710      JR  Z,HLTST
A60D  C38DA5  3720      JP  DIGIN
;nein - weiter
A610  0603    3730  NLTST: LD  B,3
;Schleifenzaehler
A612  DDB619  3740  TSTNL: OR  (IX+#19)
;teste Nachlaufzaehler
A615  DD23    3750      INC  IX
;Zeiger 2 Bytes weiter
A617  DD23    3760      INC  IX
A619  10F7    3770      DJNZ TSTNL
;Schleifenende
A61B  FE00    3780      CP  #00
;alle abgelaufen?
A61D  2803    3790      JR  Z,END
;ja - Ende
A61F  C38DA5  3800      JP  DIGIN
;nein - weiter
A622  FB      3810  END:  EI
;Interrupt freigeben
A623  C9      3820      RET
;zurueck in BASIC
3830 ;*****
A624  02      3840  MR:  DEFB %00000010
;Motor1 rechts
A625  01      3850  ML:  DEFB %00000001
;Motor1 links
A626  08      3860      DEFB %00001000
;Motor2 rechts
A627  04      3870      DEFB %00000100
;Motor2 links
A628  20      3880      DEFB %00100000
;Motor3 rechts
A629  10      3890      DEFB %00010000
;Motor3 links
A62A  80      3900      DEFB %10000000
;Motor4 rechts
A62B  40      3910      DEFB %01000000
;Motor5 links
A62C  00      3920  ROPOSL: DEFB 0
;Roboter Istposition
A62D  00      3930  ROPOSH: DEFB 0
A62E  3940      DEFS 4
A634  00      3950  ROSOLL: DEFB 0
;Roboter Sollposition
A635  00      3960  ROSOLH: DEFB 0
A636  3970      DEFS 4
A63C  00      3980  AV:  DEFB 0
;Teil-Ausgabeworte
A63D  00      3990  NL:  DEFB 0
;Nachlaufzaehler
A63E  4000      DEFS 4
;verzahnte Tabelle
A644  00      4010  MD:  DEFB 0
;Motor-Drehrichtungen
A645  00      4020      DEFB 0
A646  4030      DEFS 4
A64C  00      4040      DEFB 0

```

```

A64D  00      4050  SCRATC: DEFB 0
;Scratch
A64E  4060      DEFS 4
4070 ;***** Ende *****

```

Pass 2 errors: 00

Table used: 702 from 1408

Seite 31 ist eine leere Seite
Seiten 32 bis 60 sind wie 2 bis 30 nur
in Englisch. Daher wurde auf das Scannen
verzichtet.

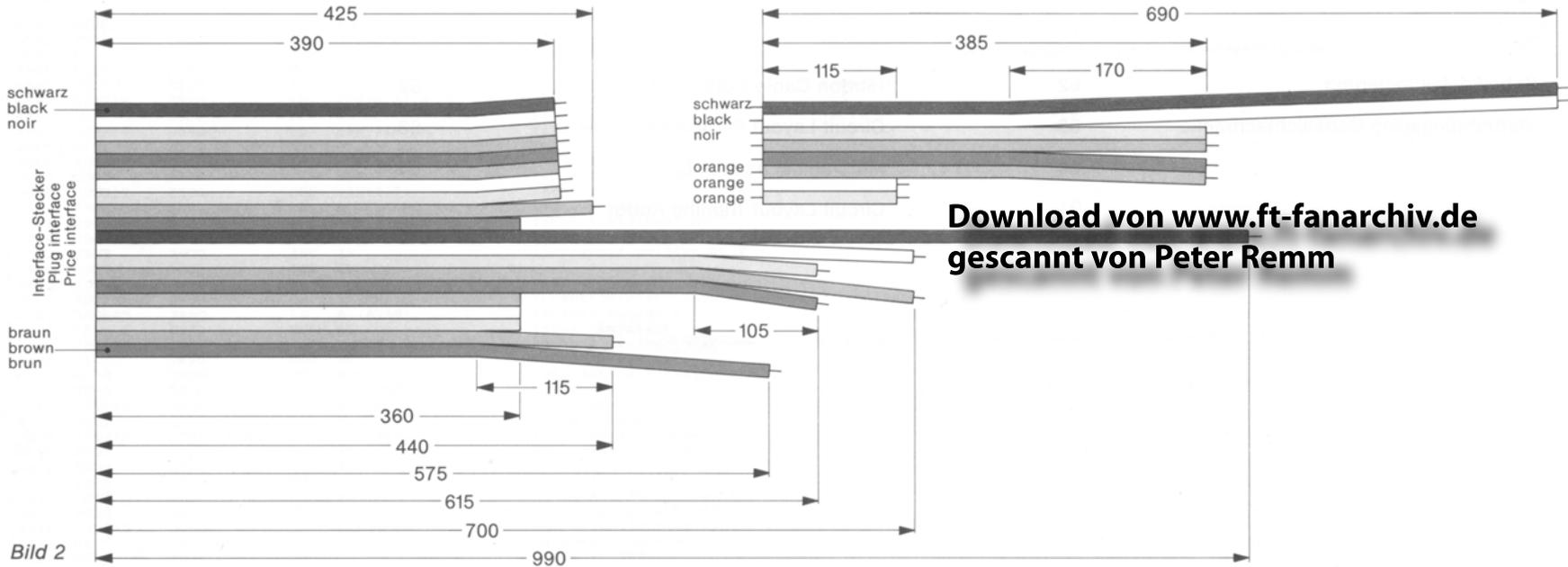
Bebilderte Bauanleitung

Kabelkonfektionierung	62
Verdrahtungsplan Gabellichtschränke	63
Mechanischer Aufbau	64
Verdrahtungsplan Trainingsroboter	91

Illustrated Assembly Instructions

Ribbon Cable Configuration	62
Circuit Layout Photo-Interrupter	63
Mechanical Assembly	64
Circuit Layout Training Robot	91

Kabelkonfektionierung · Ribbon cable configuration · Schéma de câblage



**Steckermontage
Plug installation
Assemblage de cables**

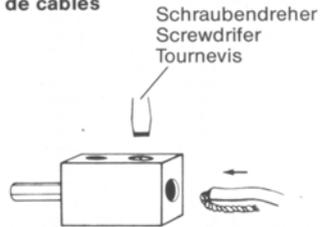


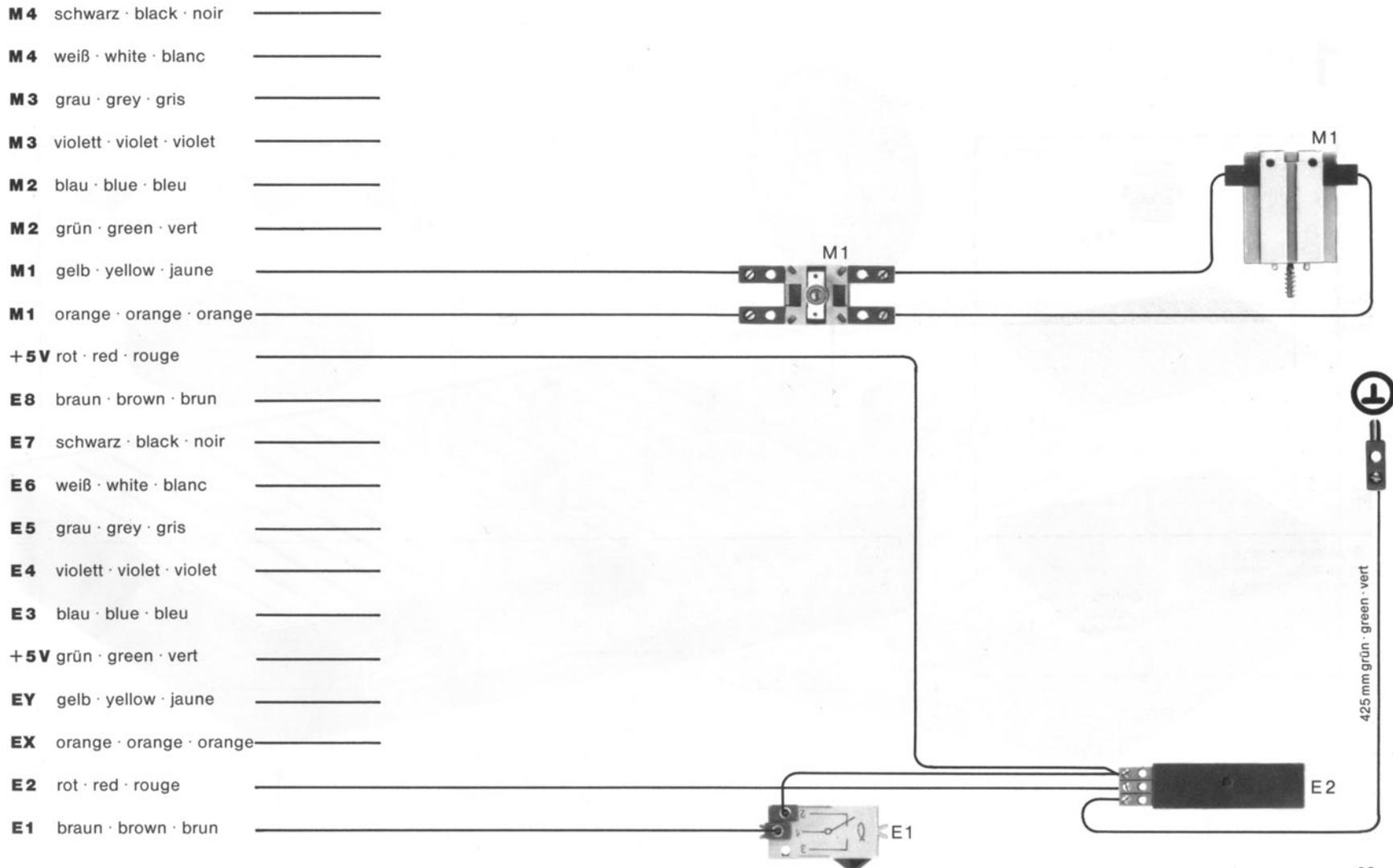
Bild 3

**Durchgangsprüfung
Continuity tester
Contrôle du passage**



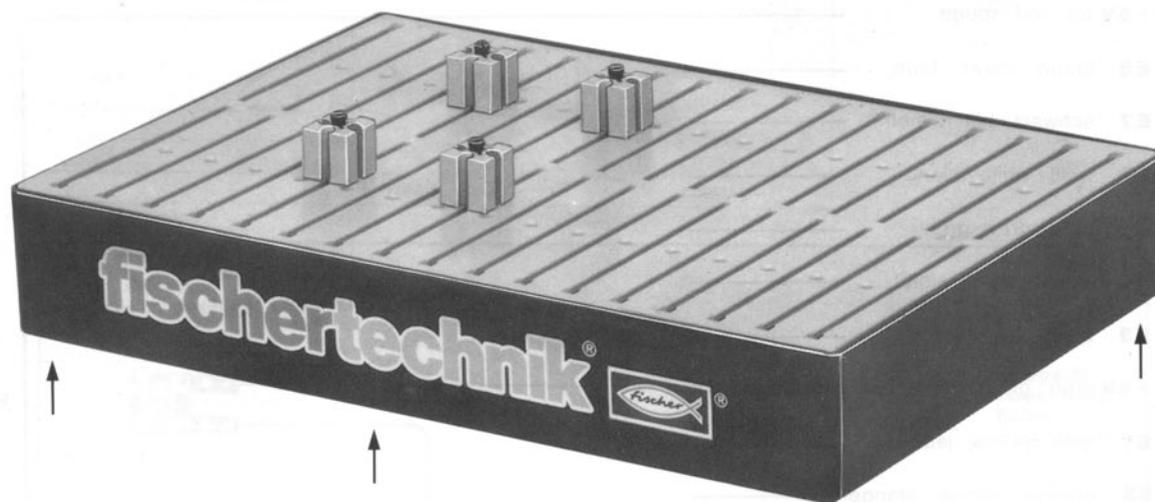
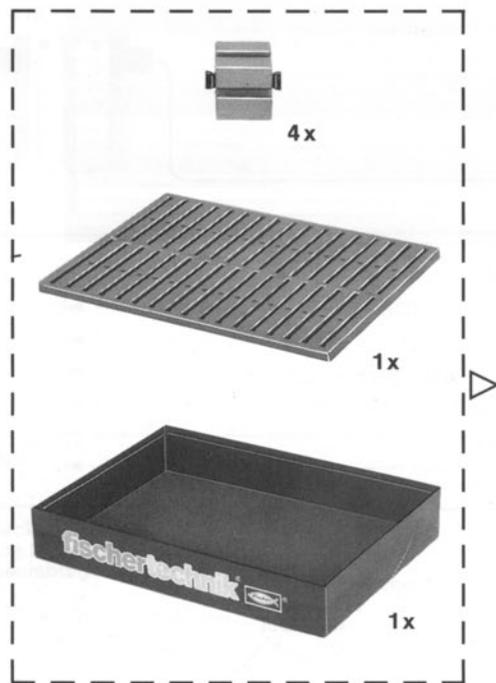
Bild 4

Verdrahtungsplan Gabellichtschränke · Circuit Layout Photo-interrupter · Plan de câblage d'interrupteur photo



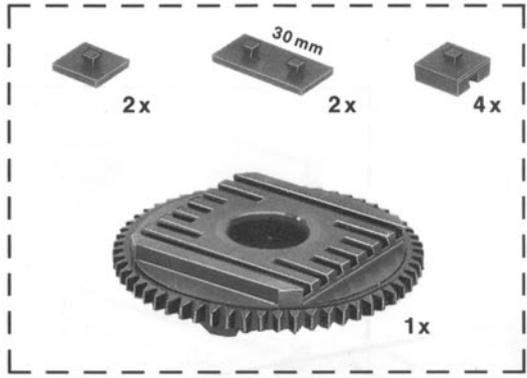
Mechanischer Aufbau · Mechanical Assembly · Assemblage des parts mécanique

1

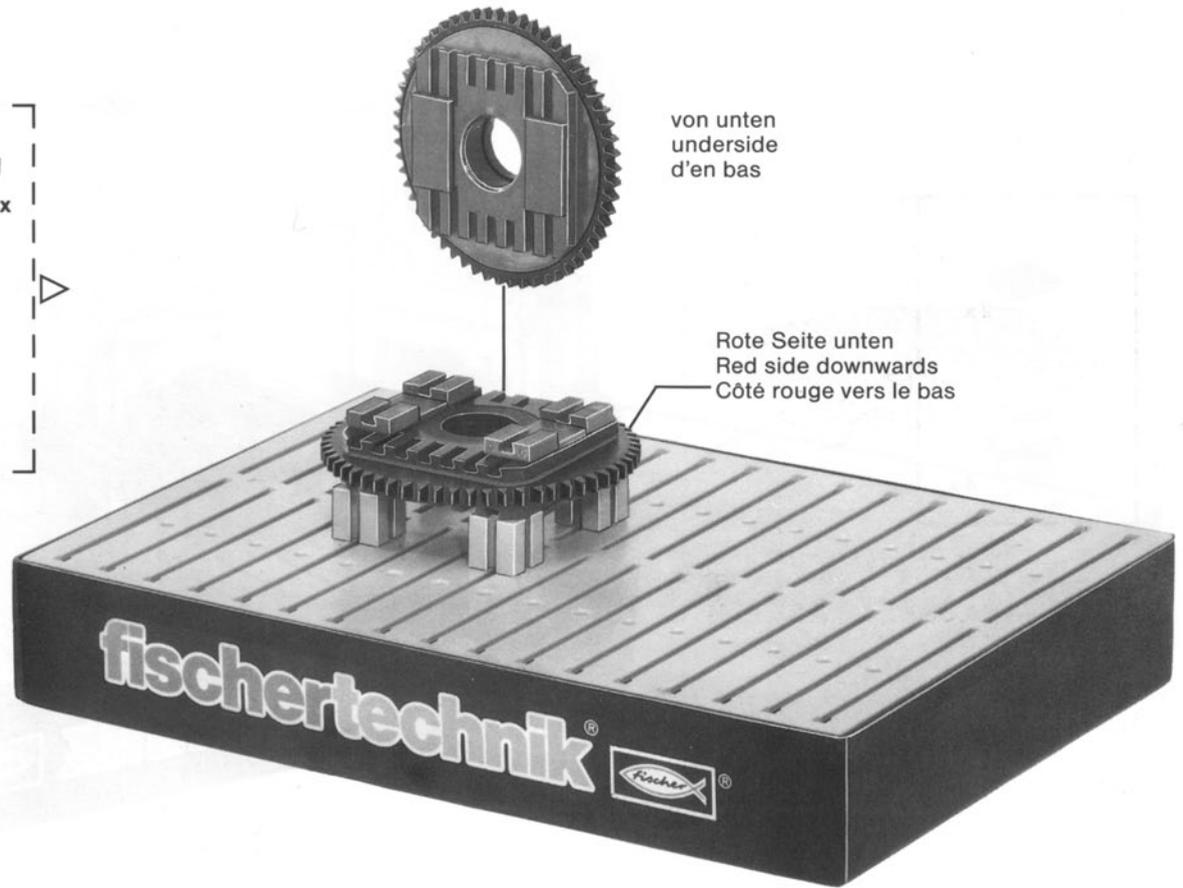


GummifüÙe aufkleben
Place selfadhesive pads
Coller les pieds en caoutchouc

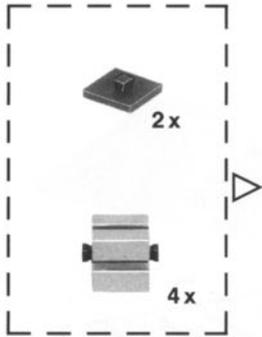
2



0 30 mm



3

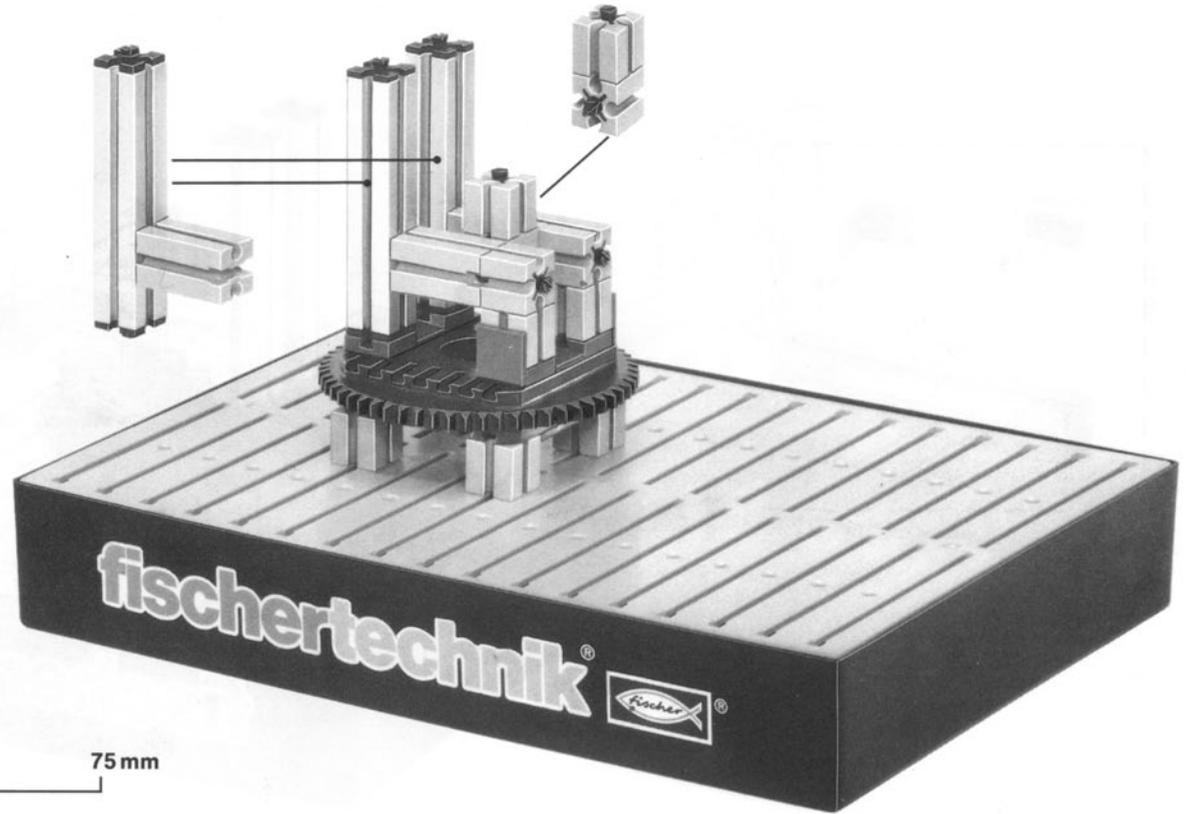
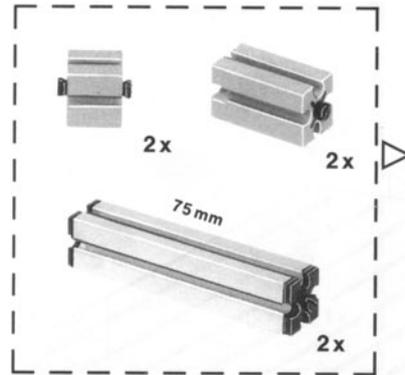


4

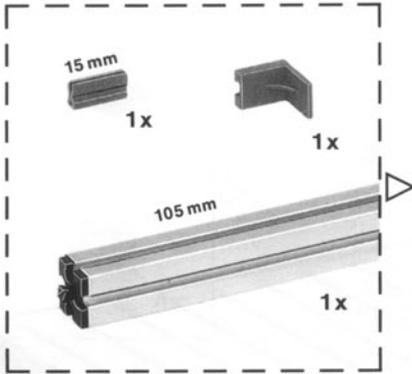
Aus technischen Gründen sind die Schiebekräfte der Metallbaustäbe teilweise sehr hoch.

For technical reasons, the shearing forces of the structural metal rods are partially very high.

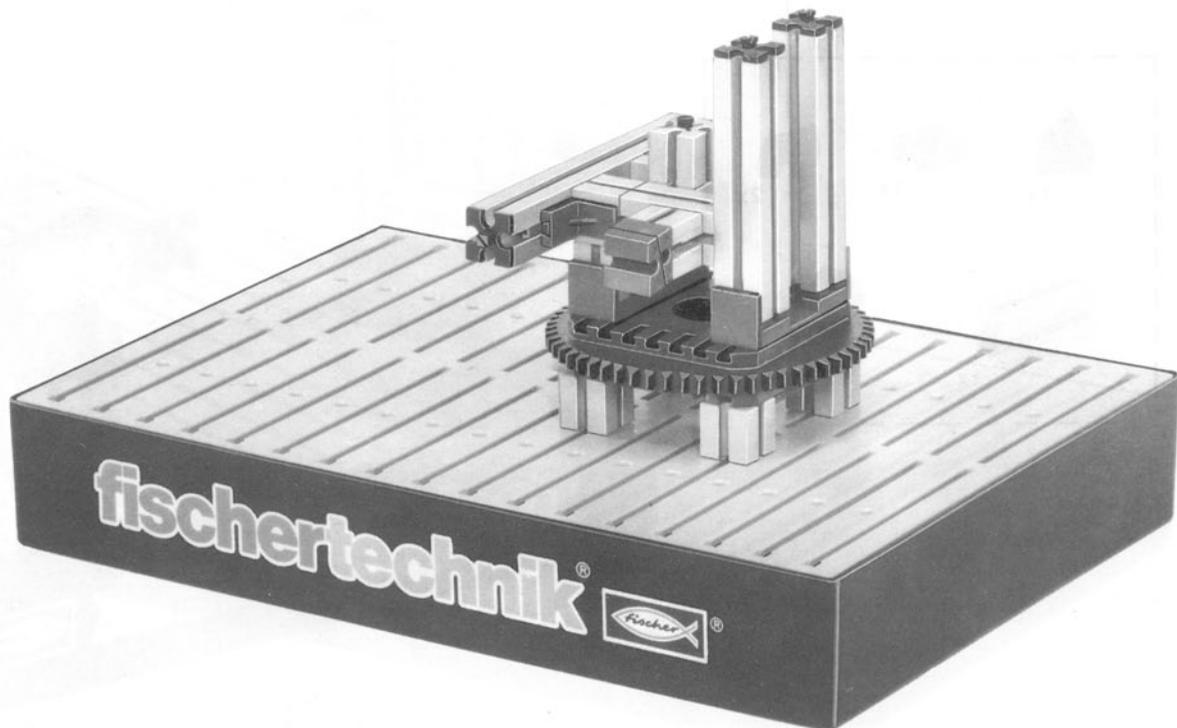
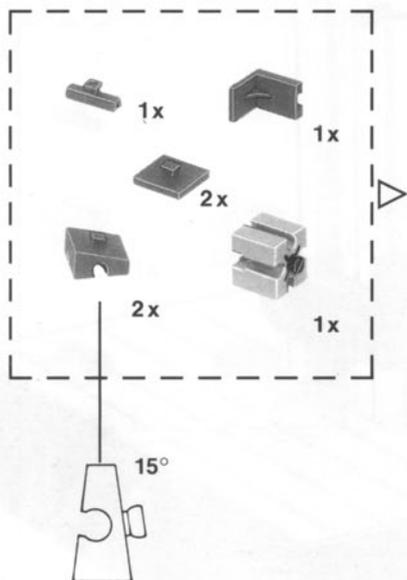
Pour des raisons techniques les forces transversales des barres métalliques de construction sont parfois très importantes.



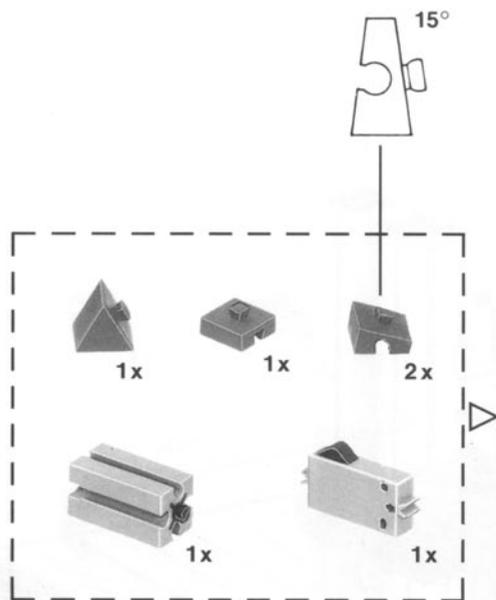
5



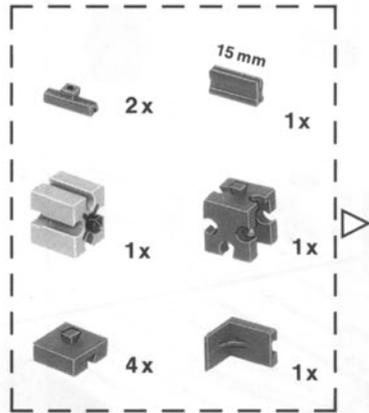
6



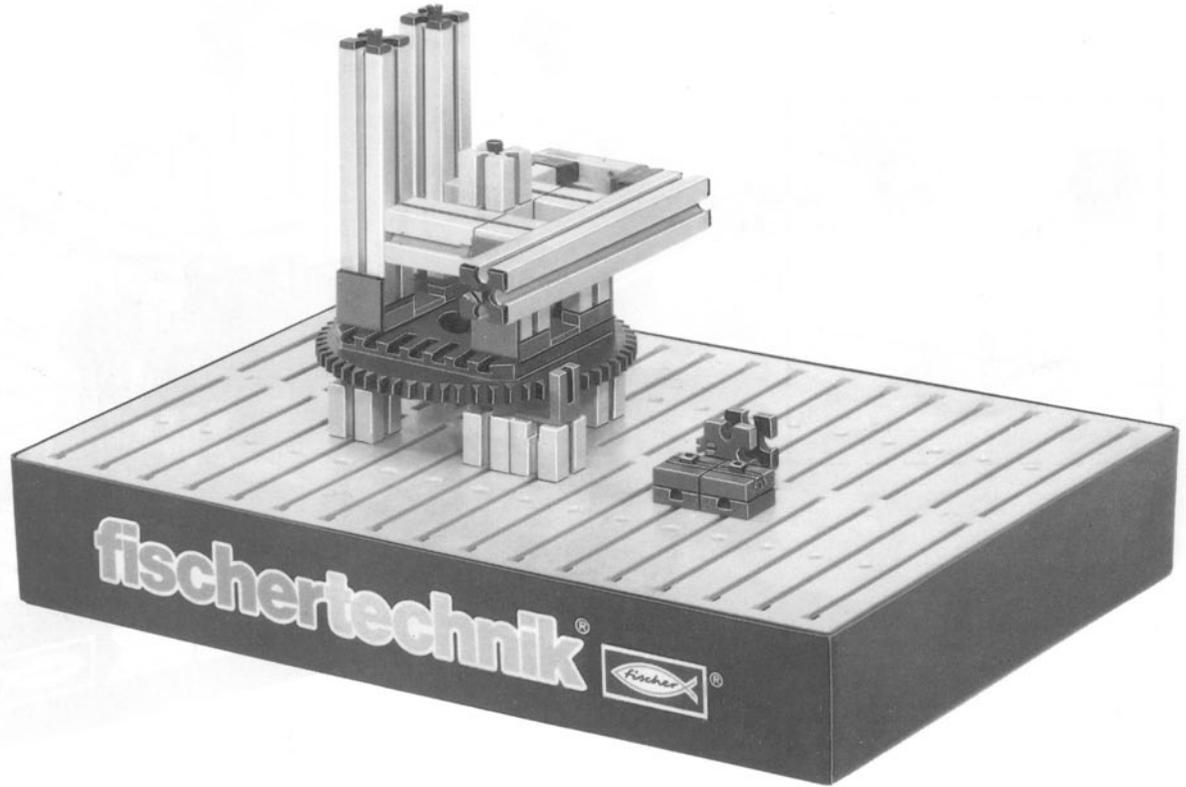
7



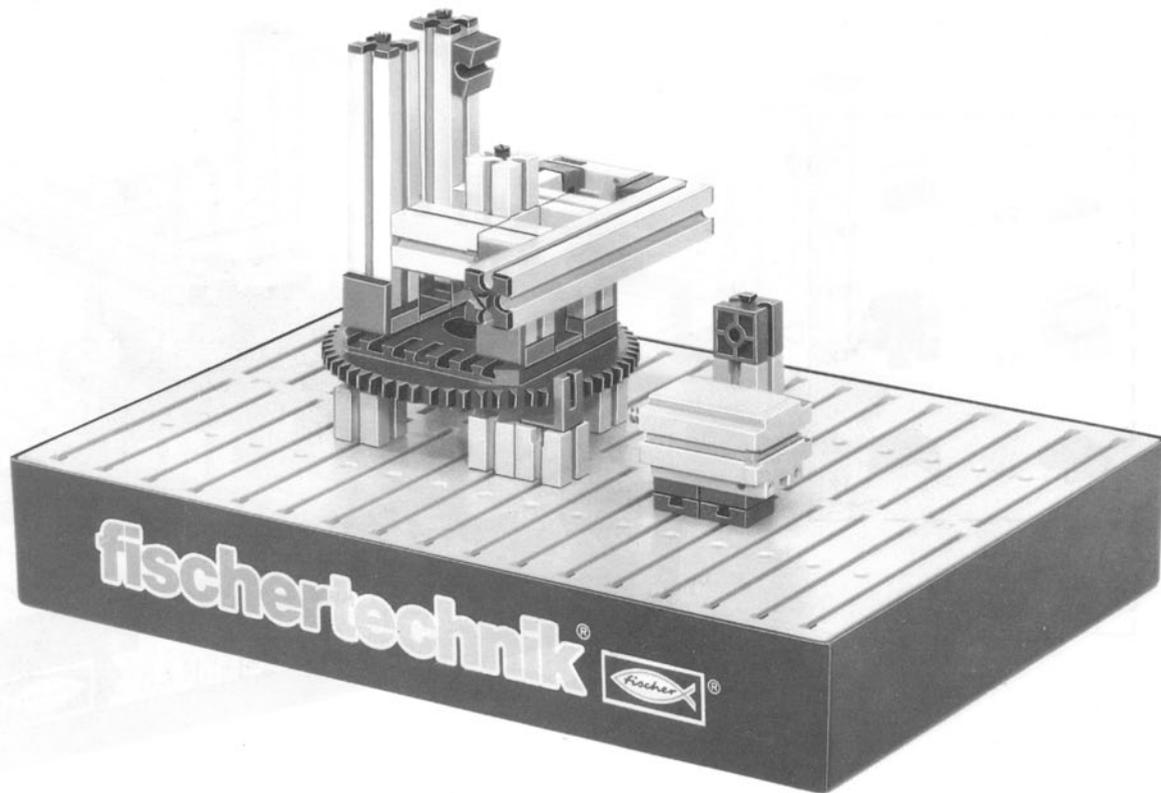
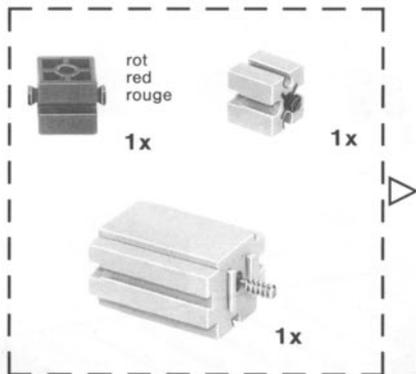
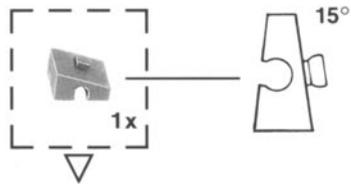
8



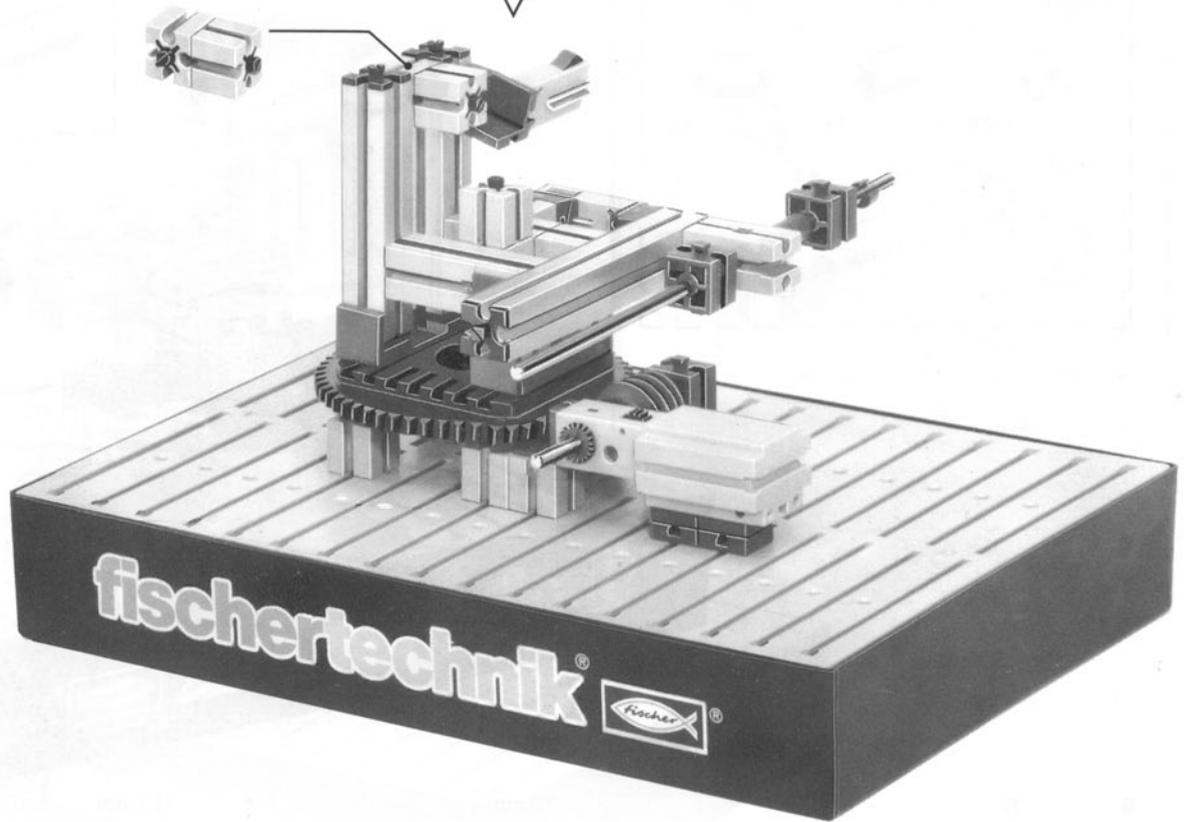
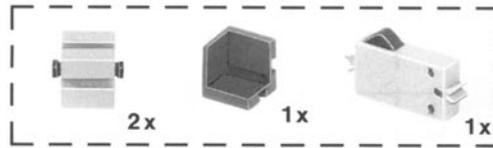
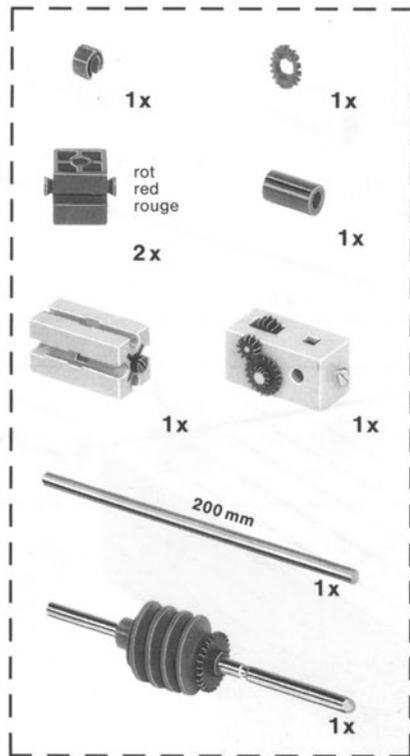
0 15 mm



9



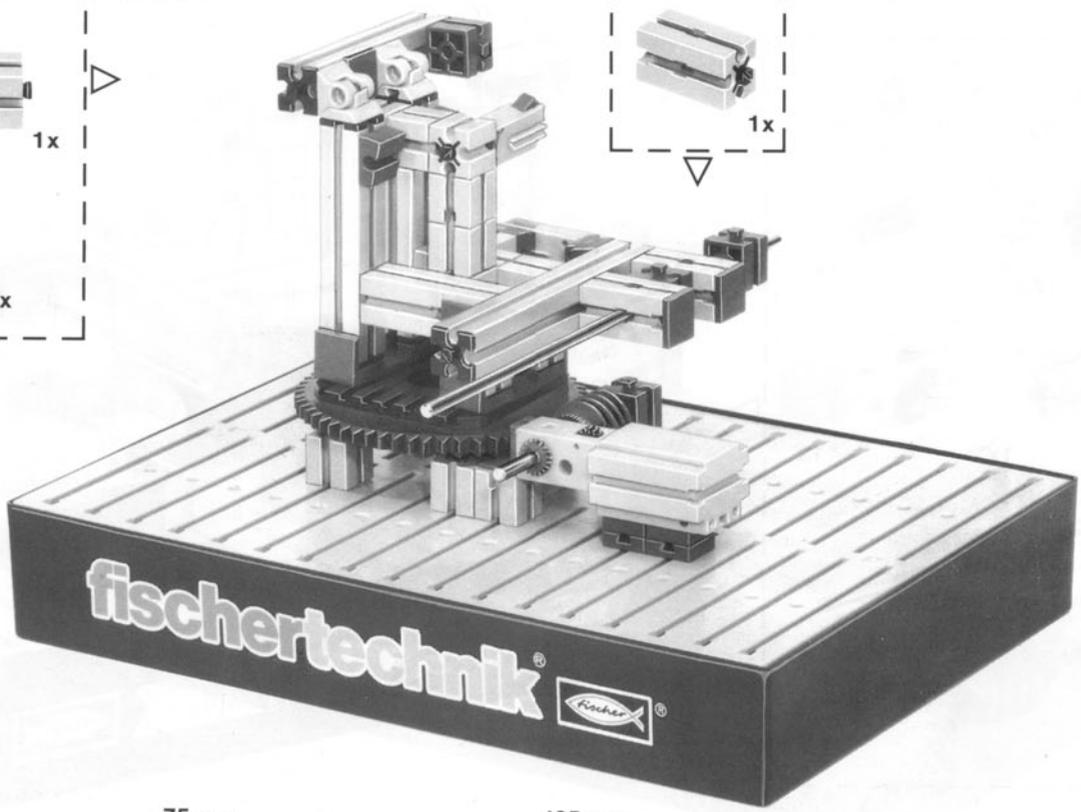
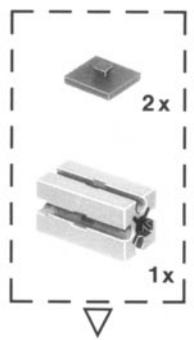
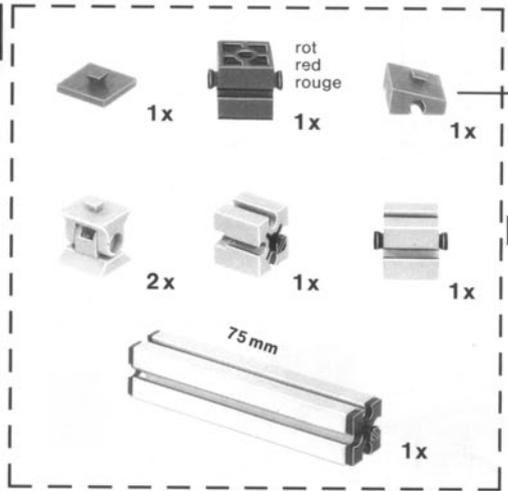
10



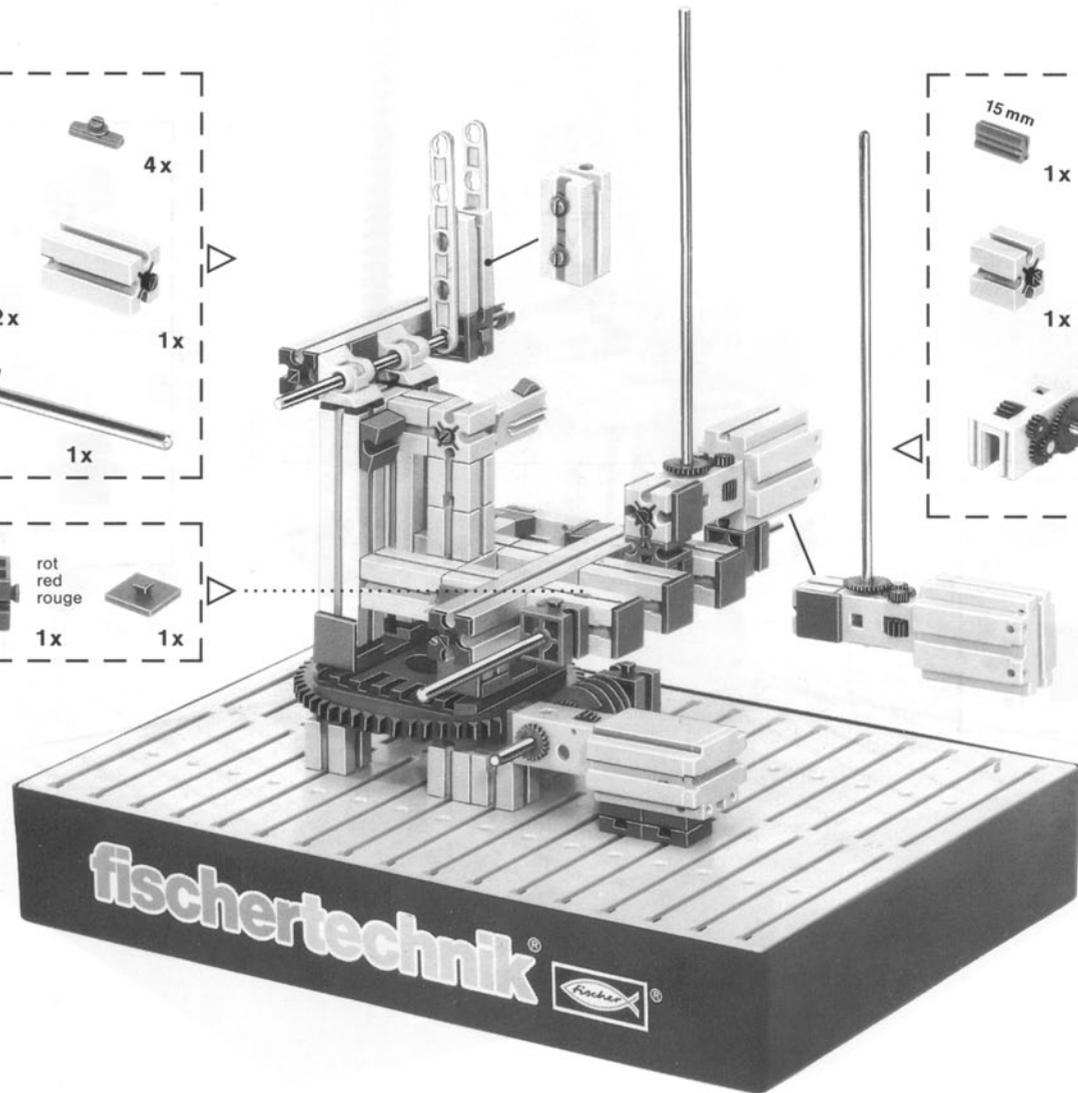
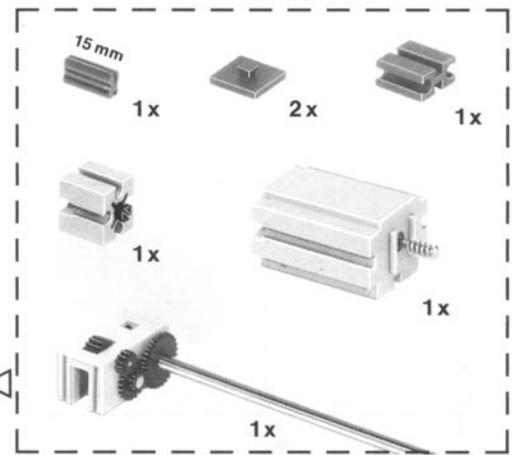
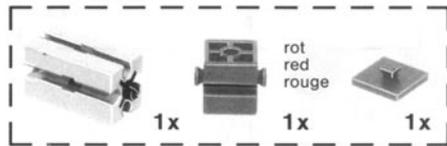
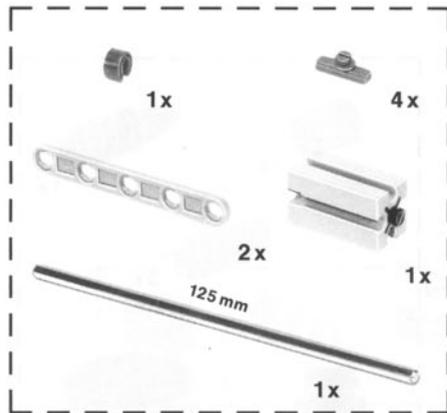
0

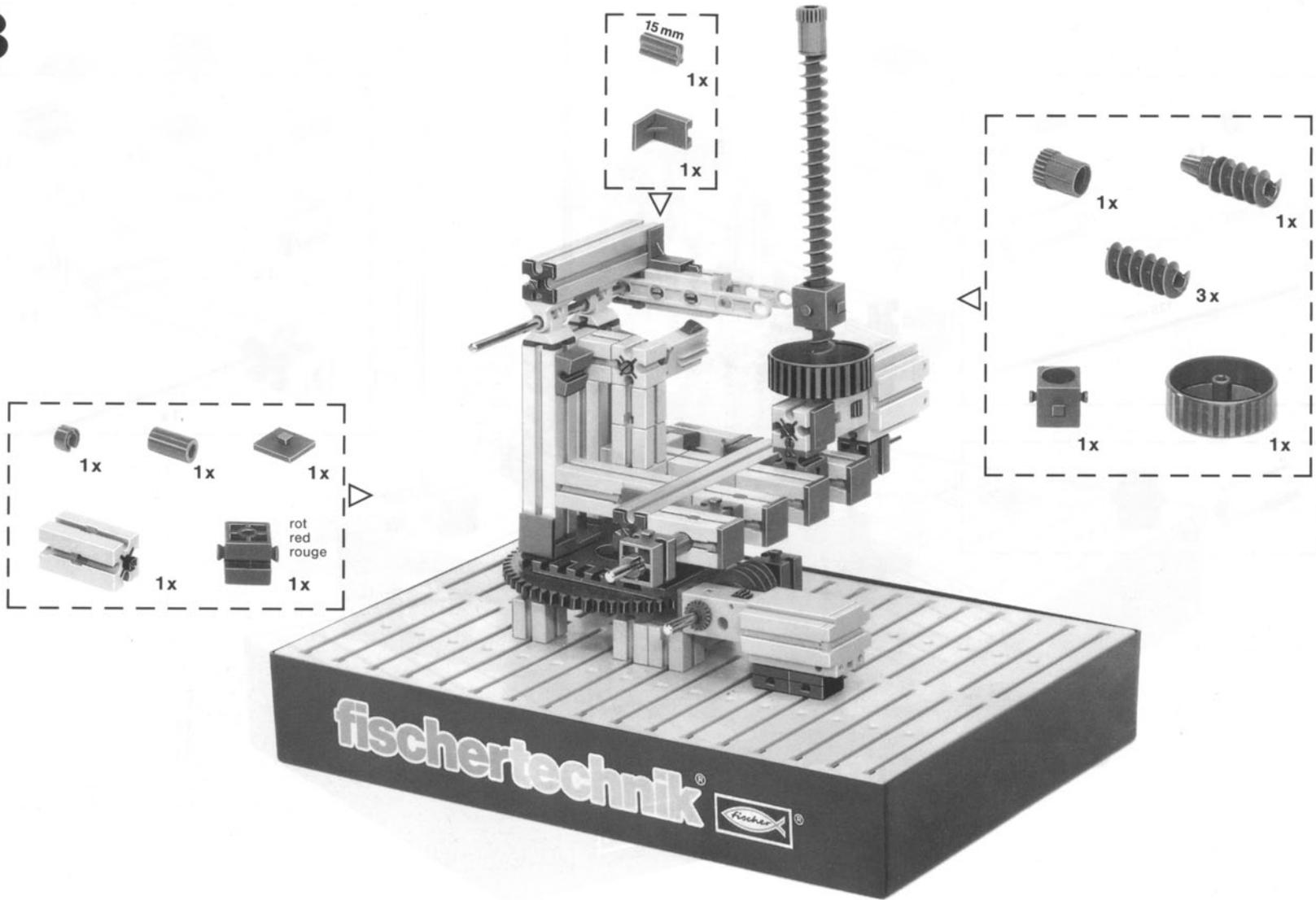
200 mm

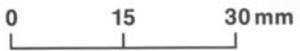
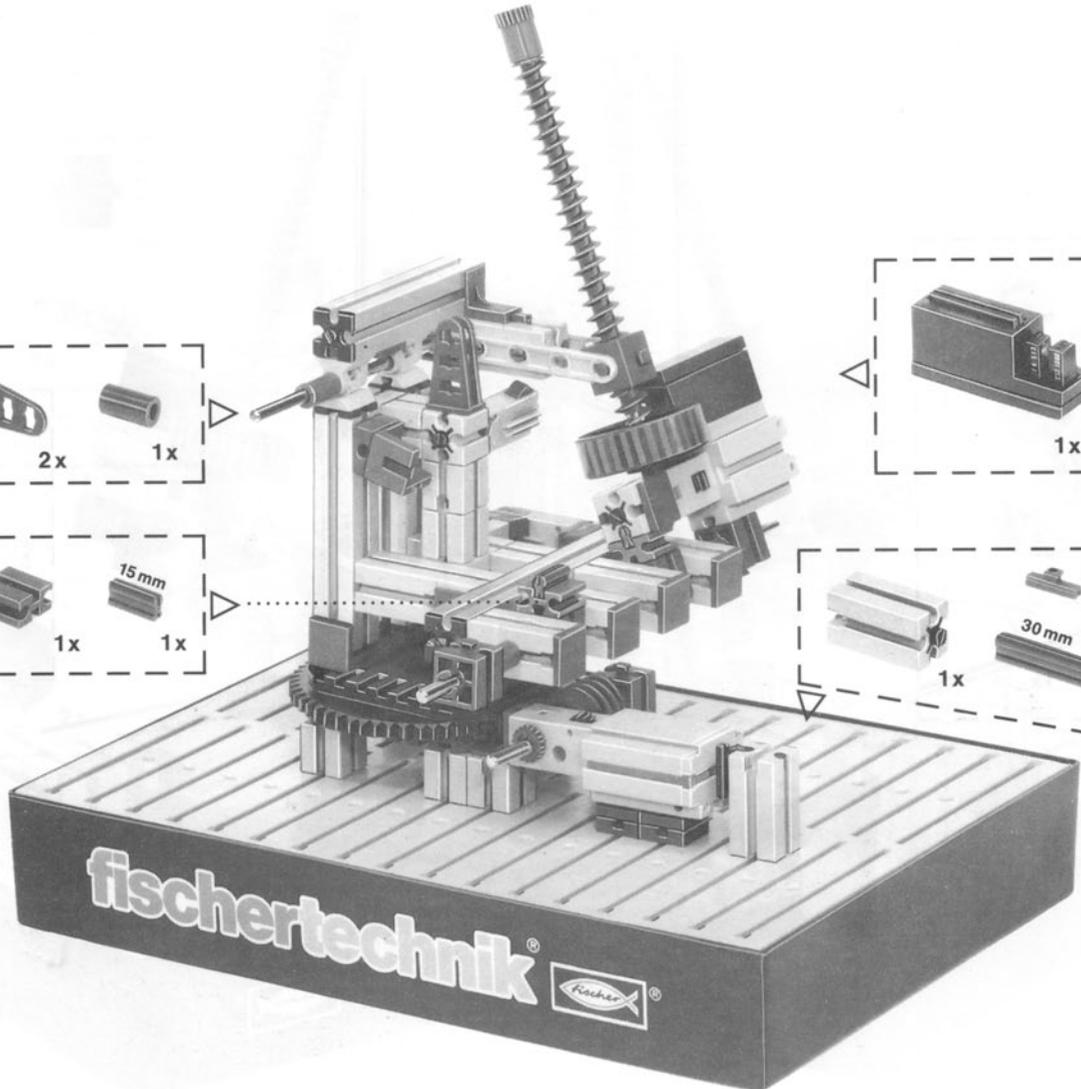
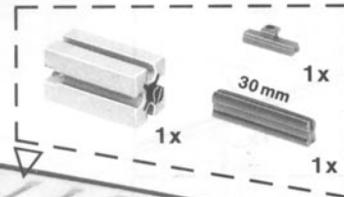
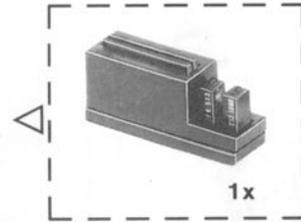
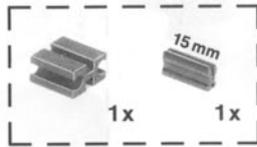
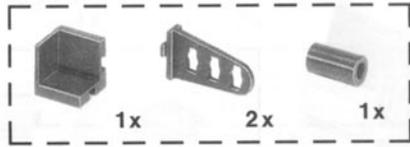
11



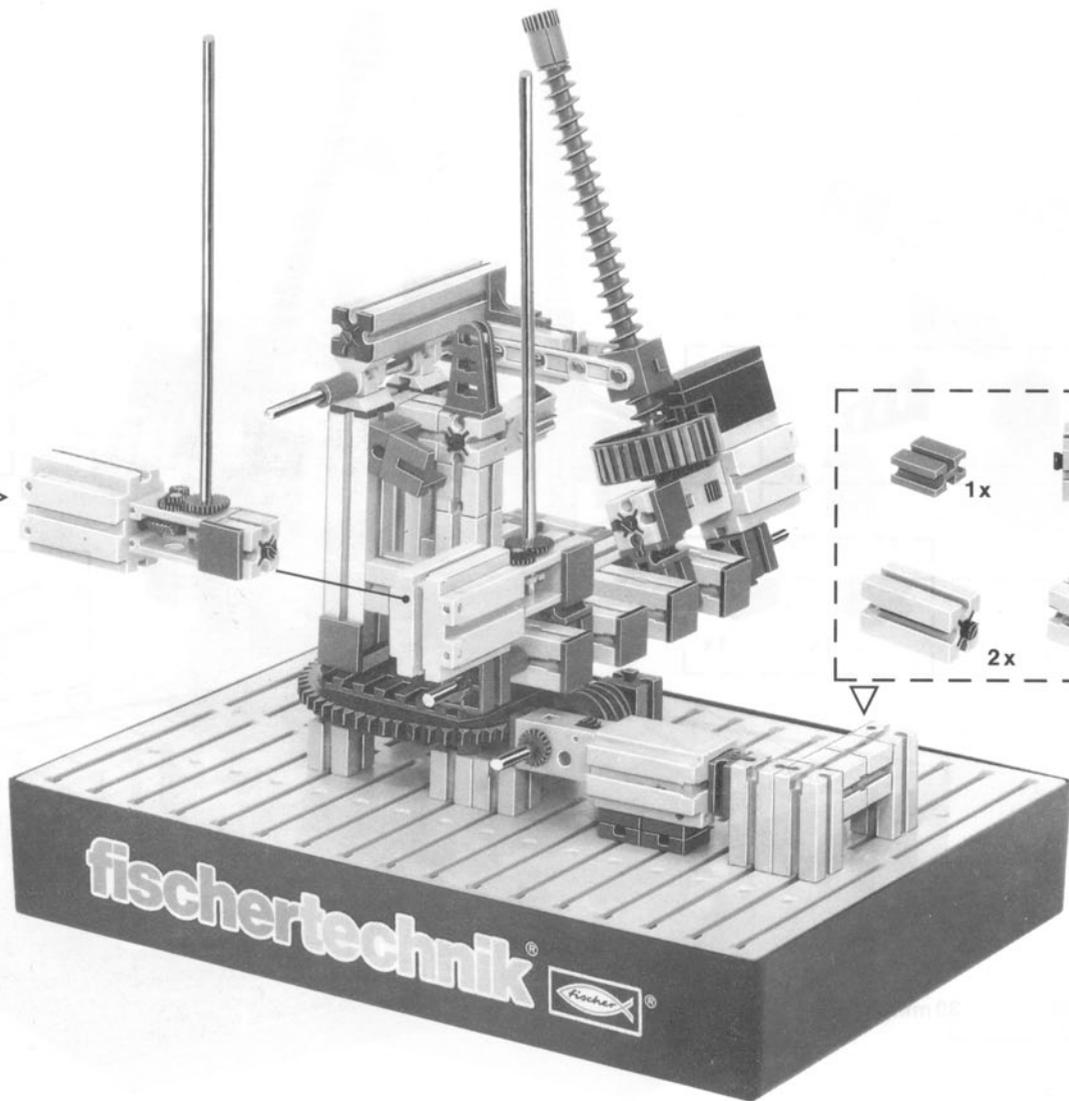
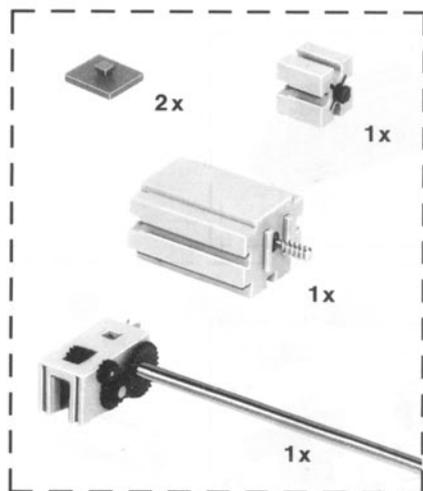
12



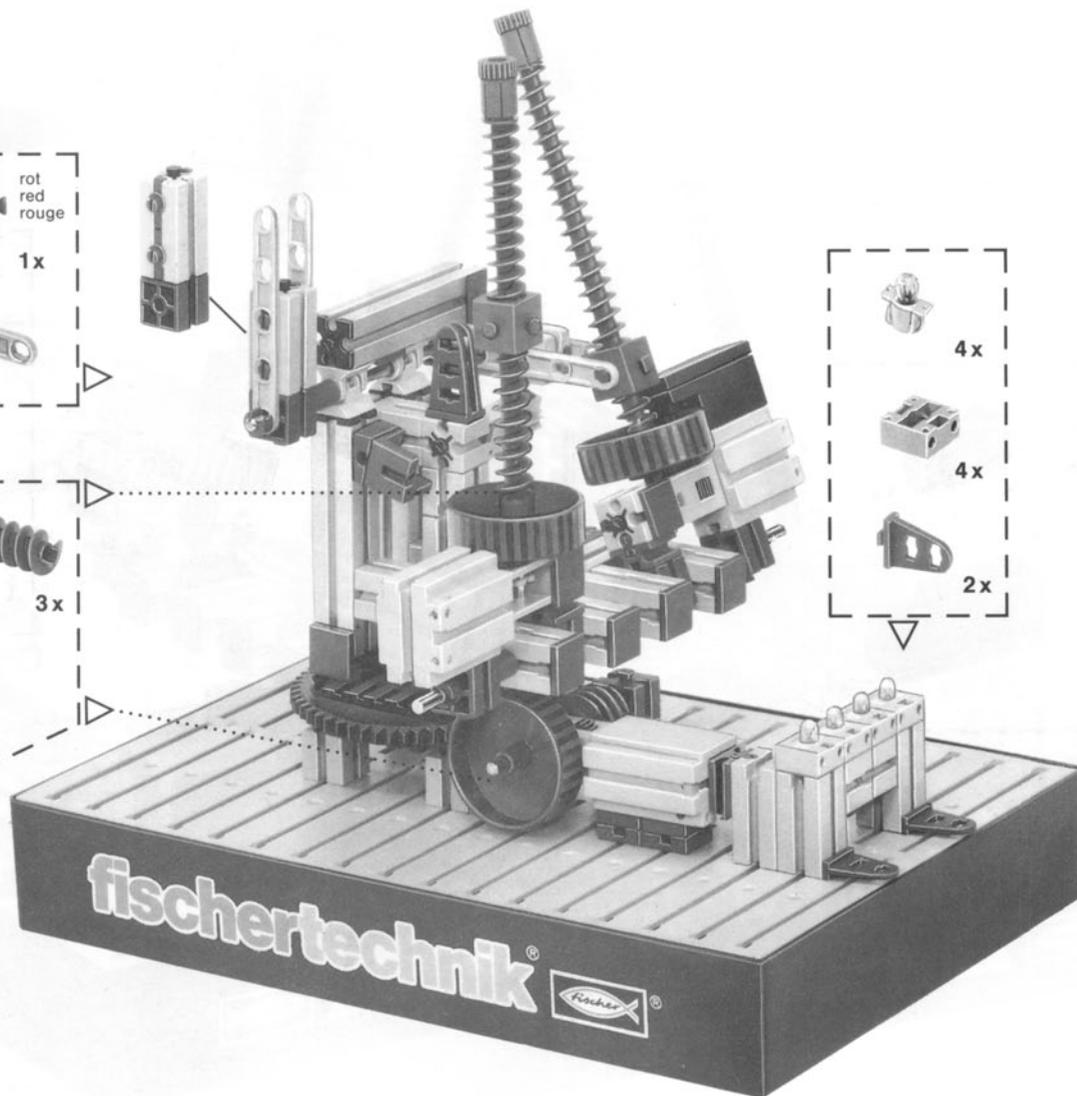
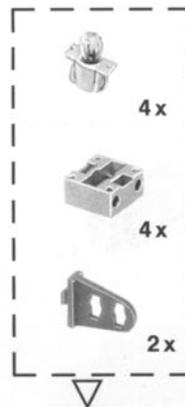
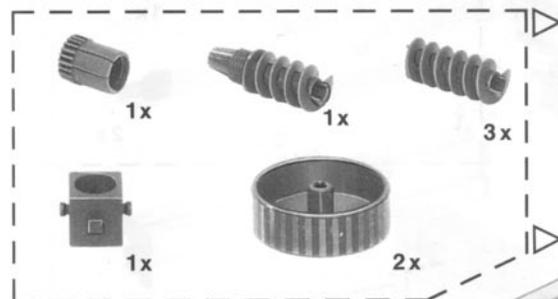
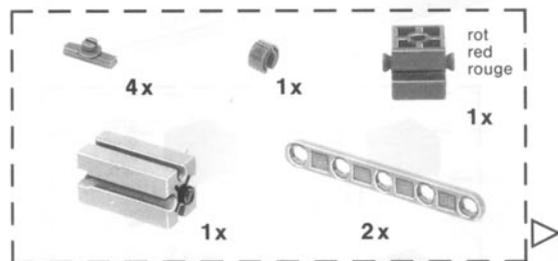


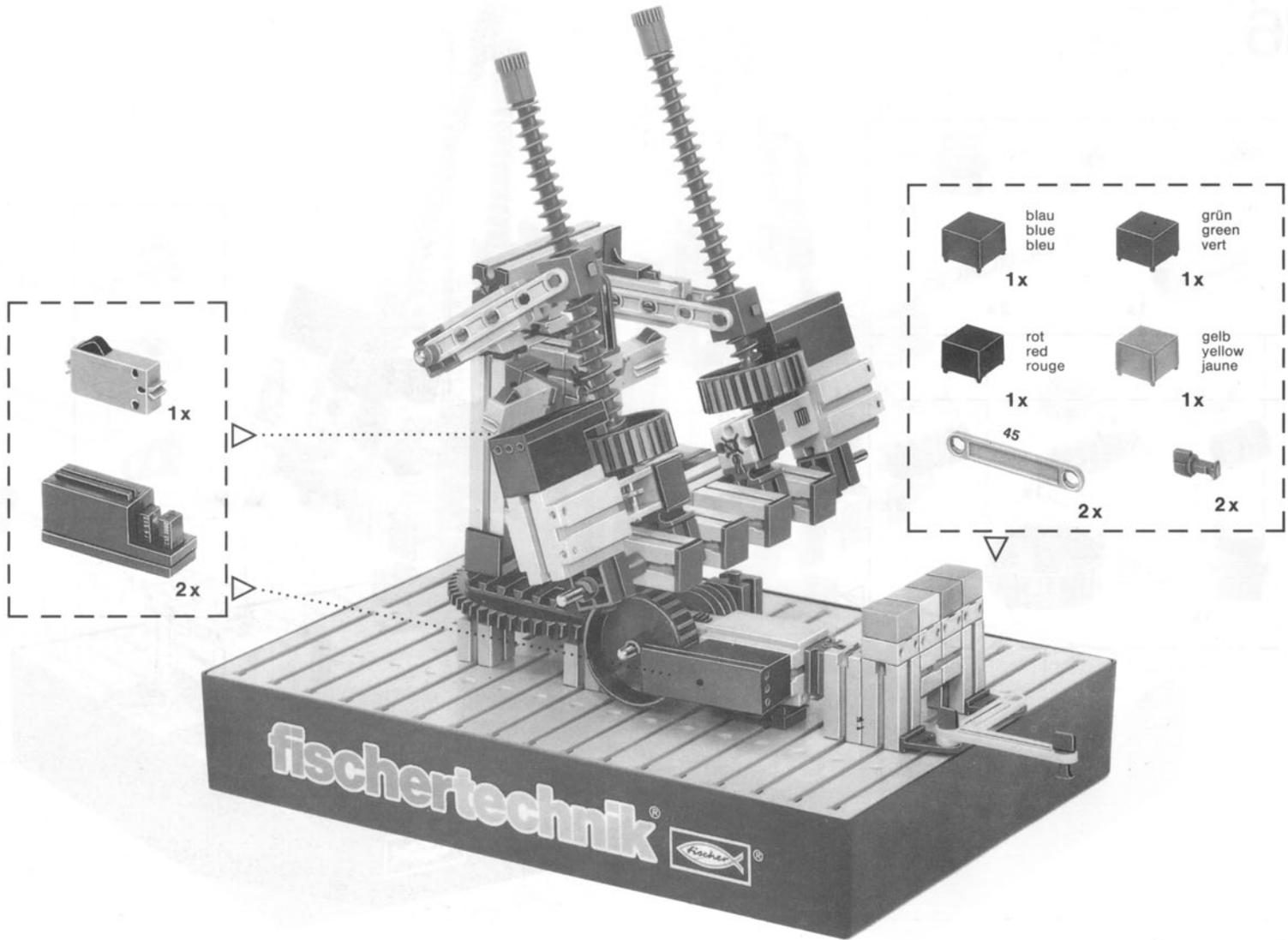


15



16



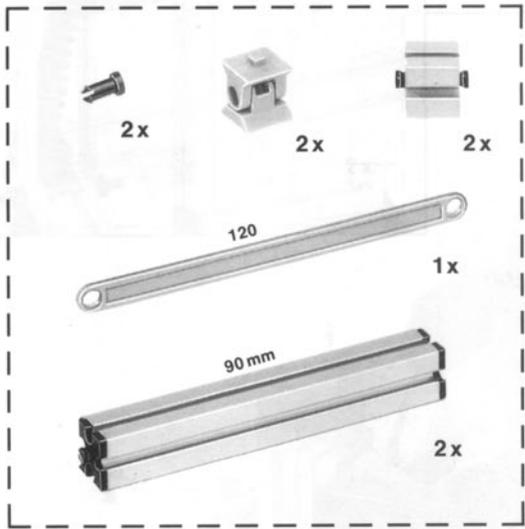
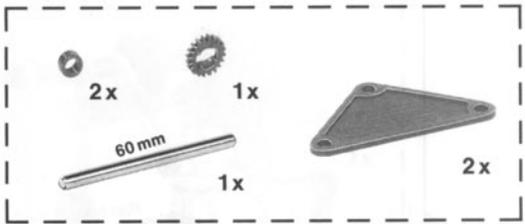


1x

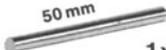
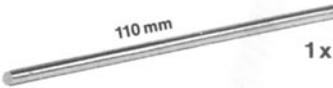
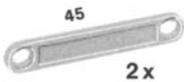
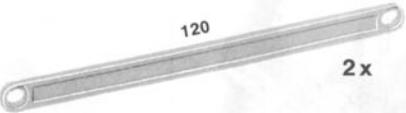
2x

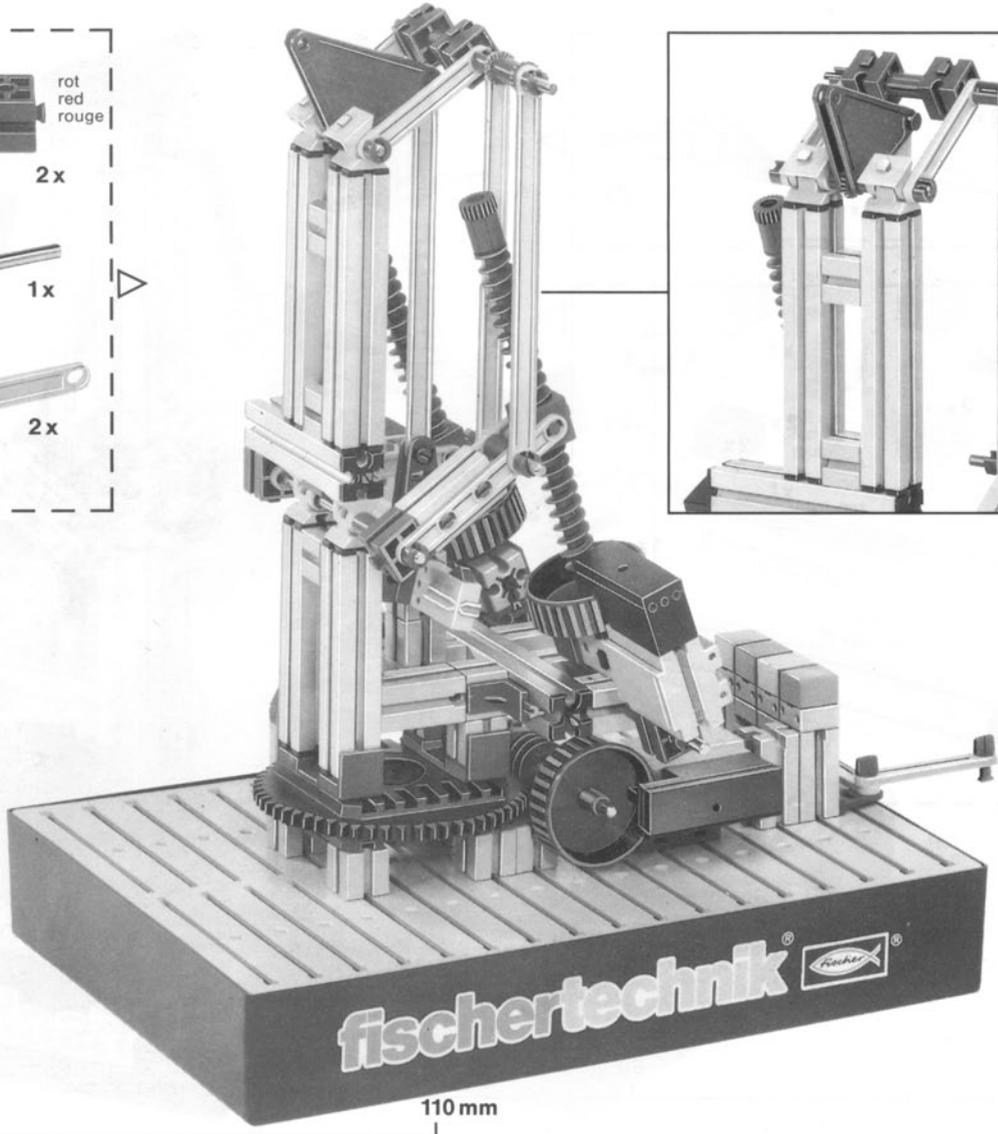
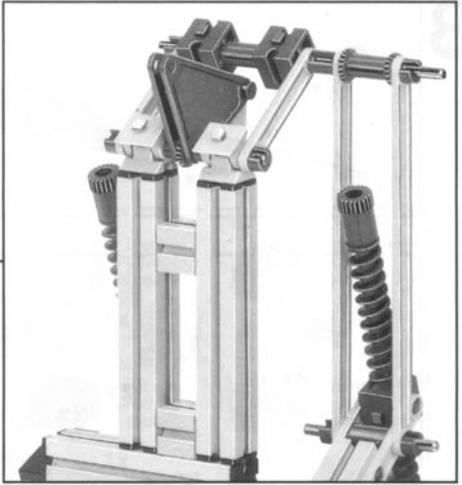
	blau blue bleu	1x		grün green vert	1x
	rot red rouge	1x		gelb yellow jaune	1x
		45			
		2x			2x

18



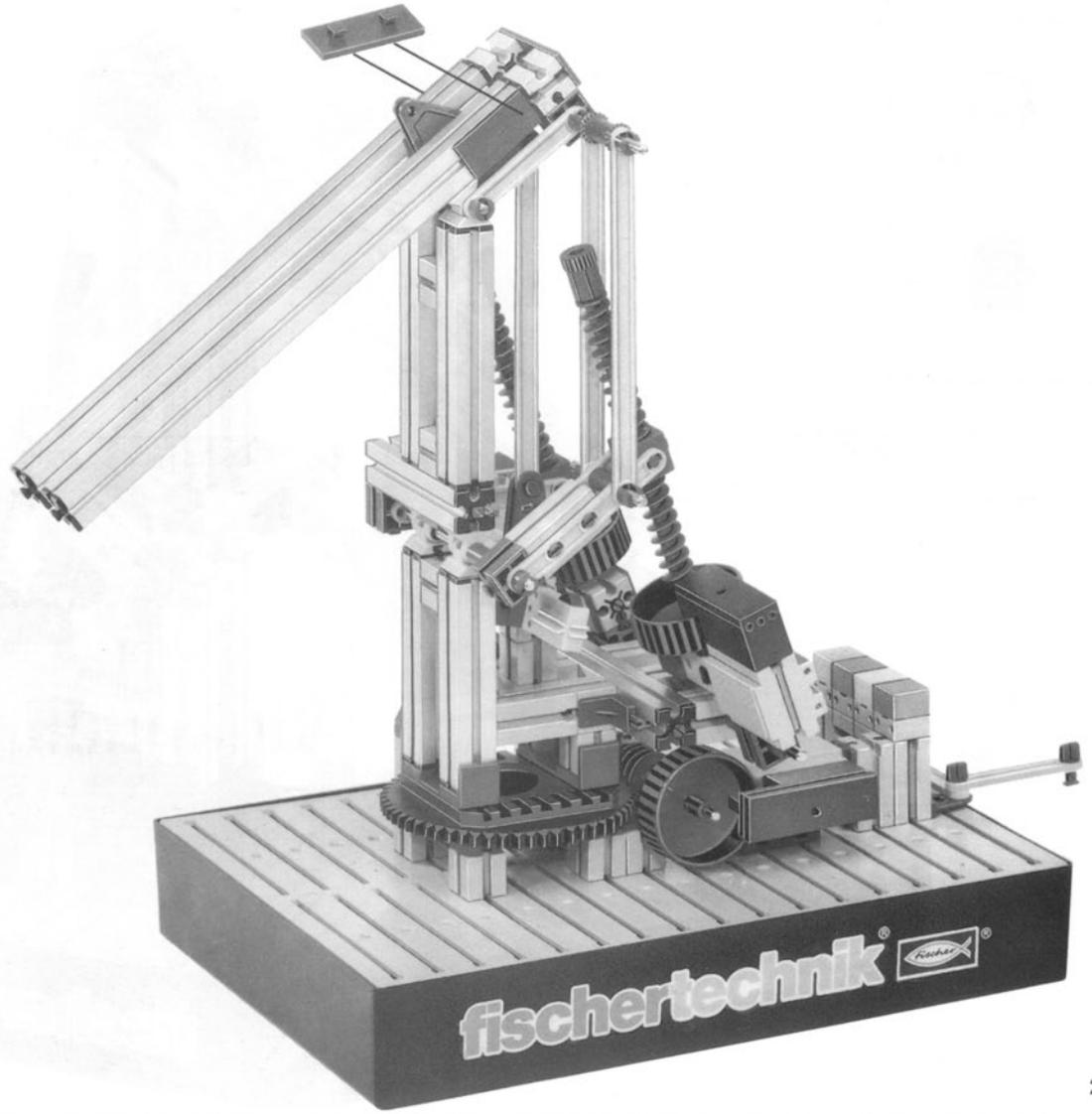
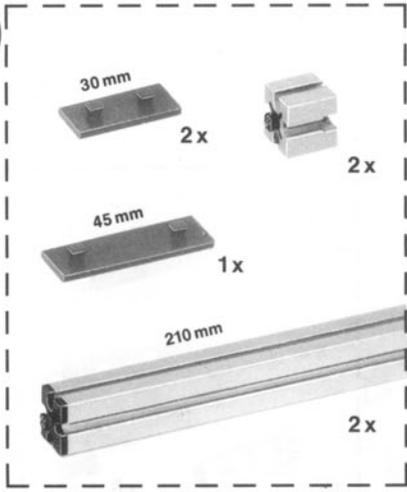
19

					rot red rouge
2x	6x	1x	3x	2x	
					
50 mm	110 mm				
1x	1x				
					
45	120				
2x	2x				

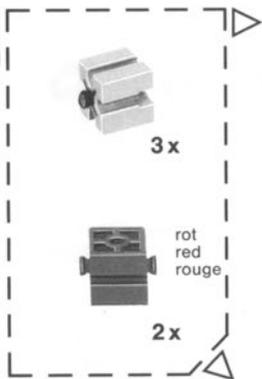


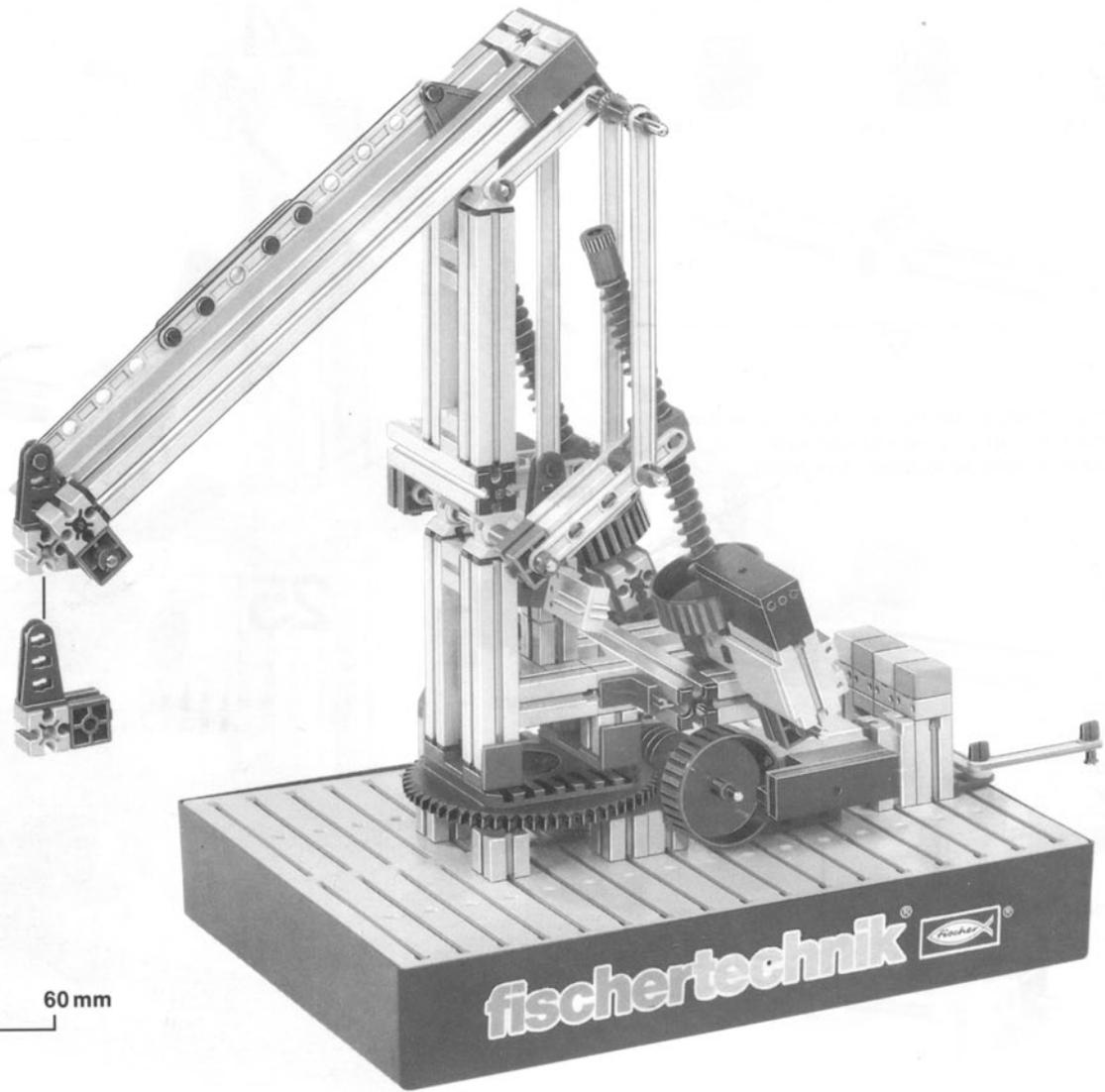
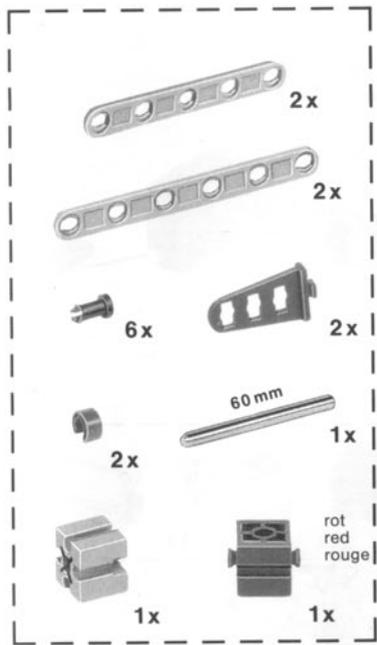
0 50 110 mm

20

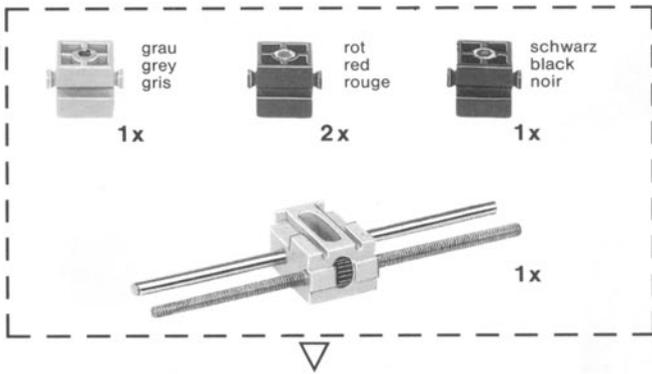


21

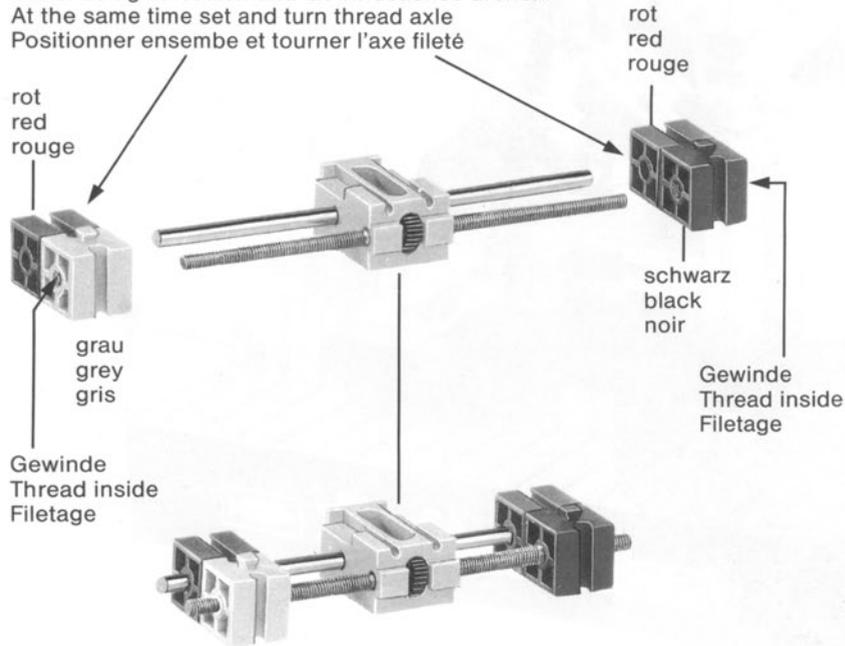




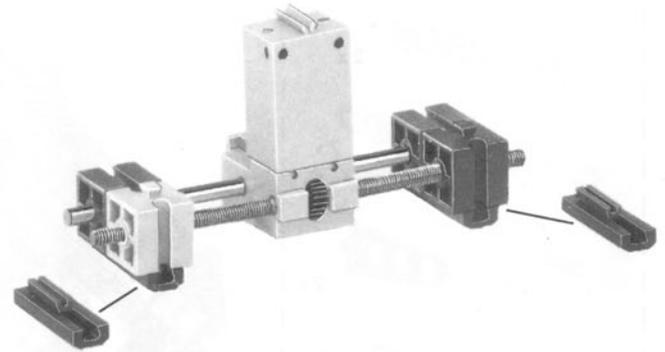
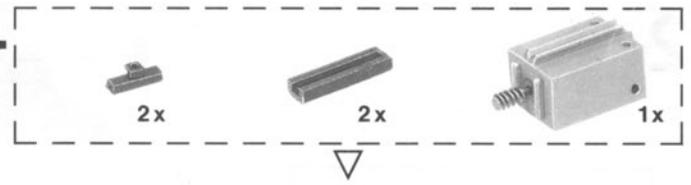
23



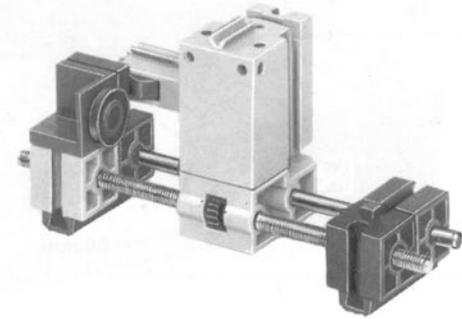
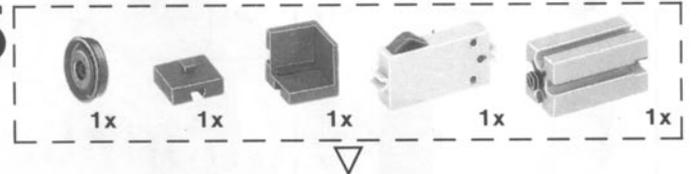
Gleichzeitig aufsetzen und Gewindeachse drehen
At the same time set and turn thread axle
Positionner ensemble et tourner l'axe fileté

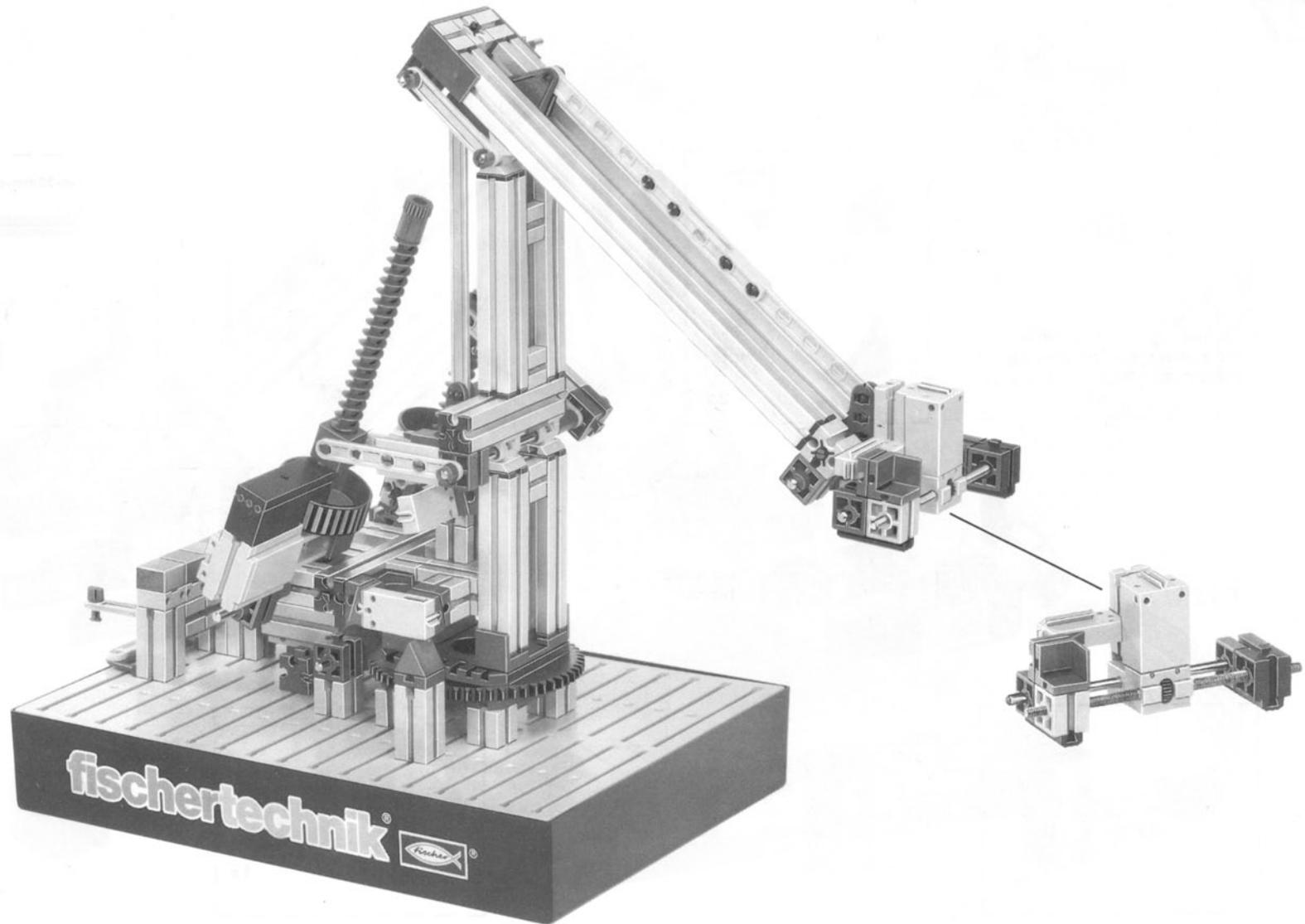


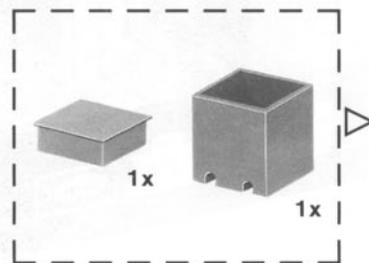
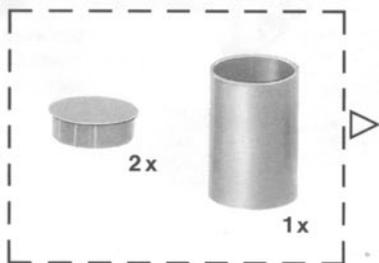
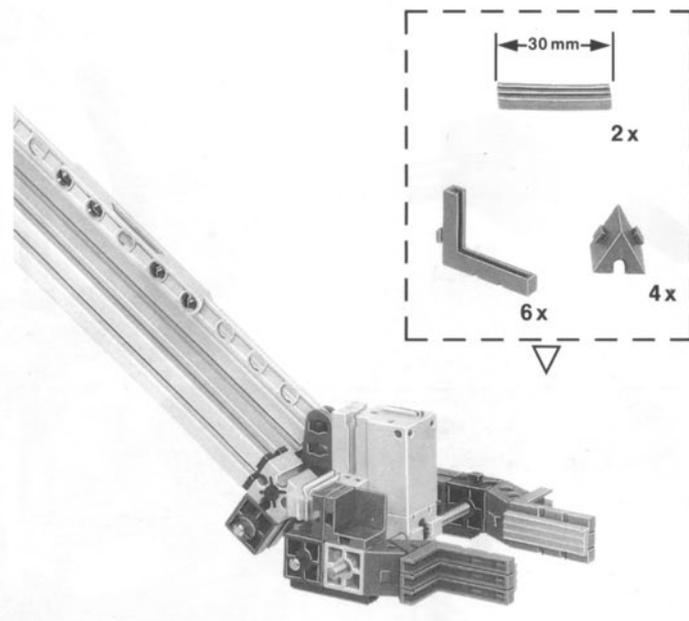
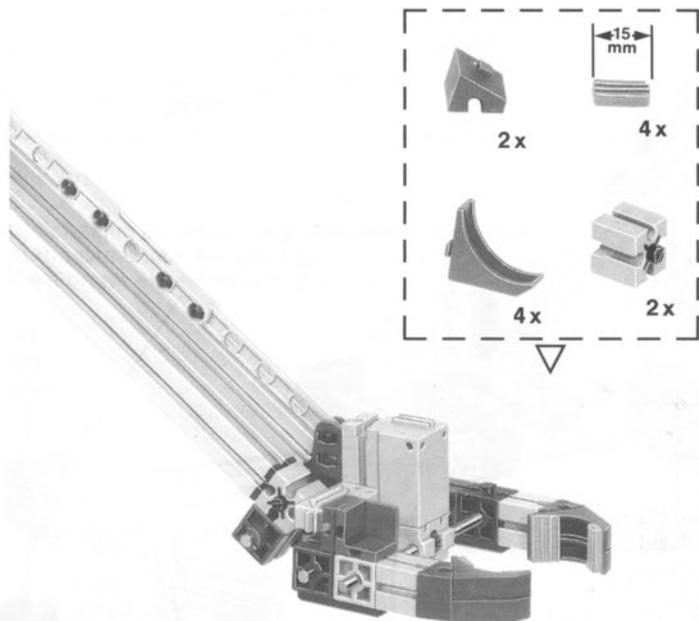
24

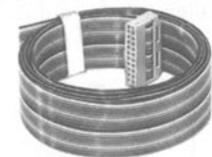
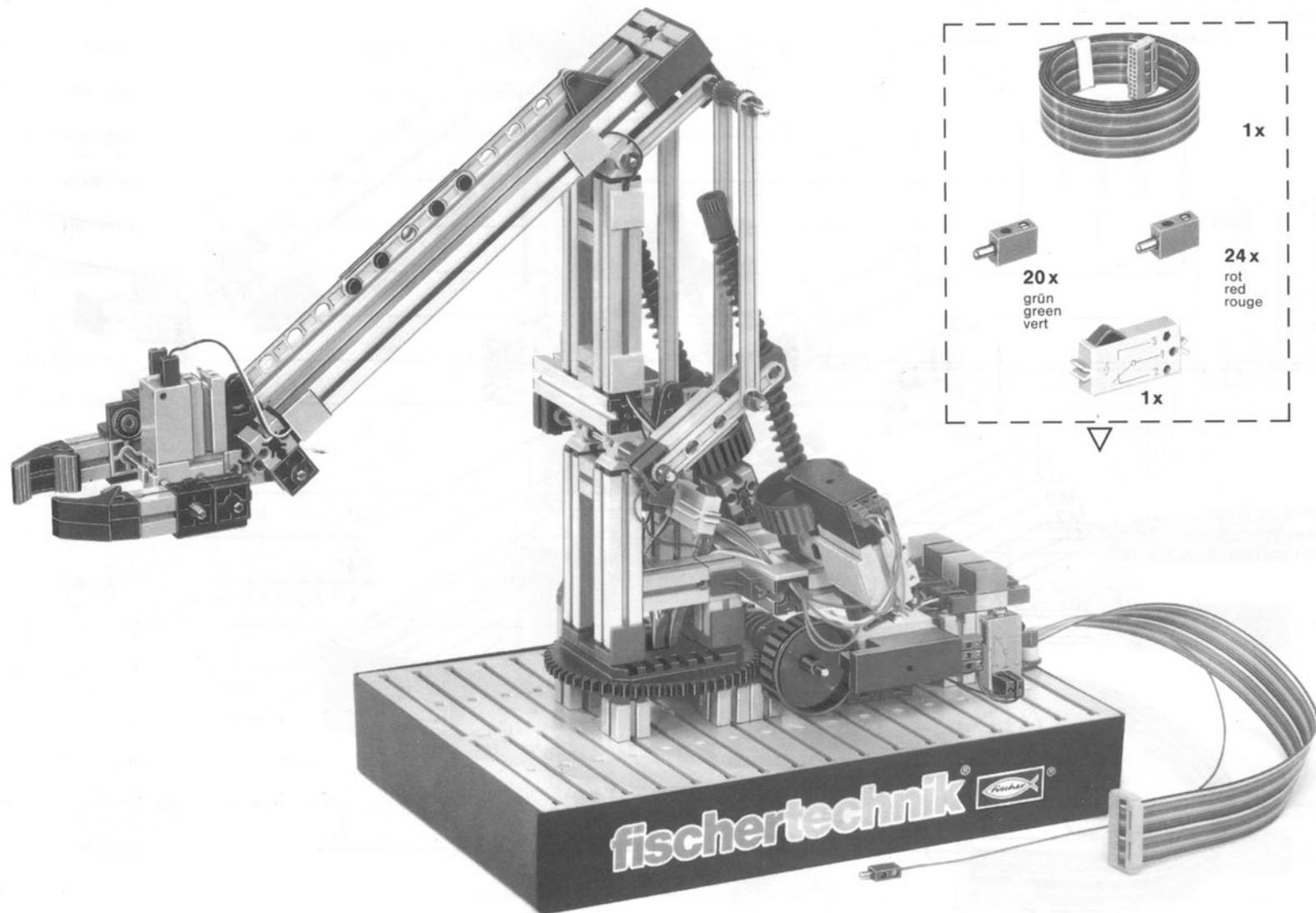


25









1x



20 x
grün
green
vert



24 x
rot
red
rouge

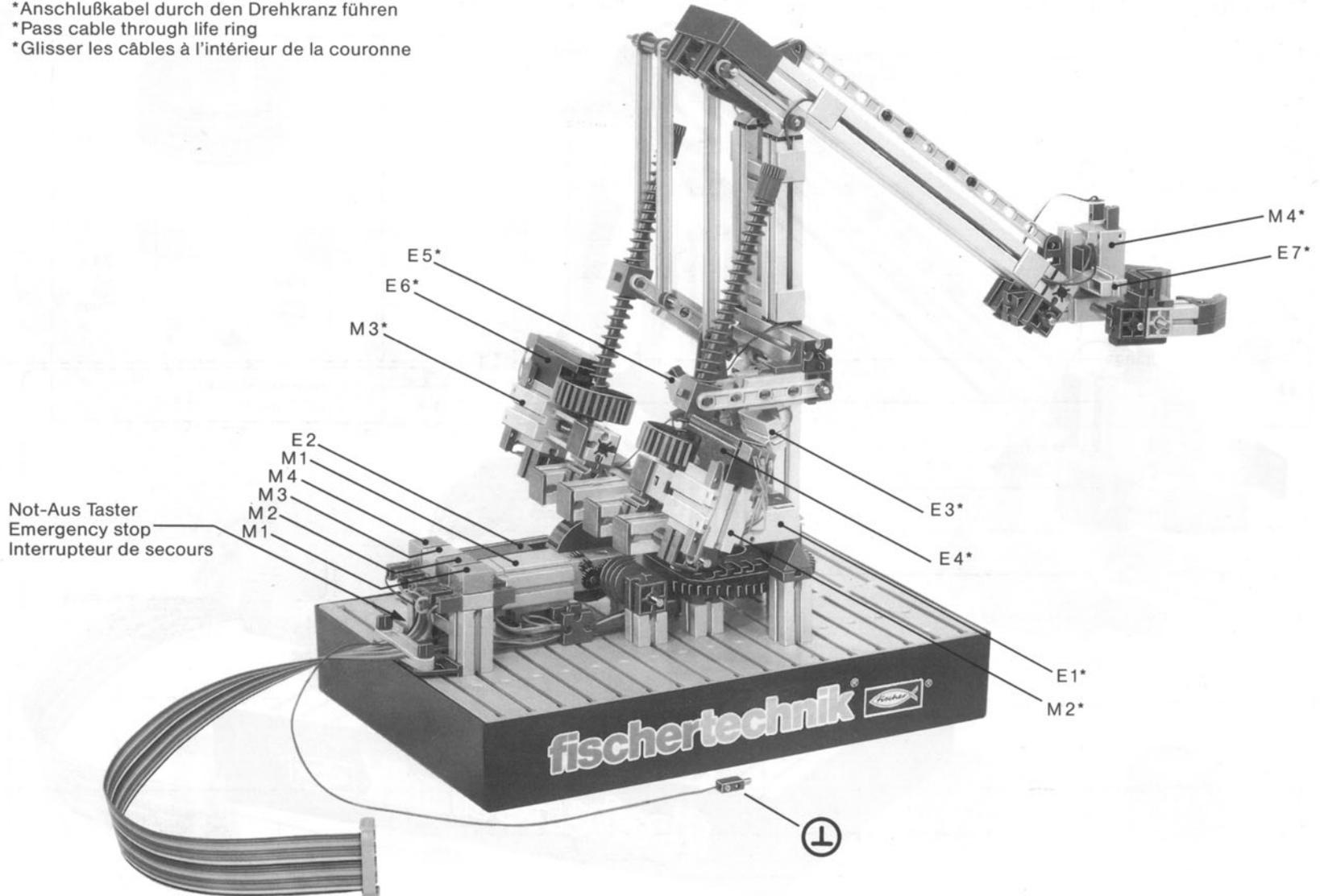


1x

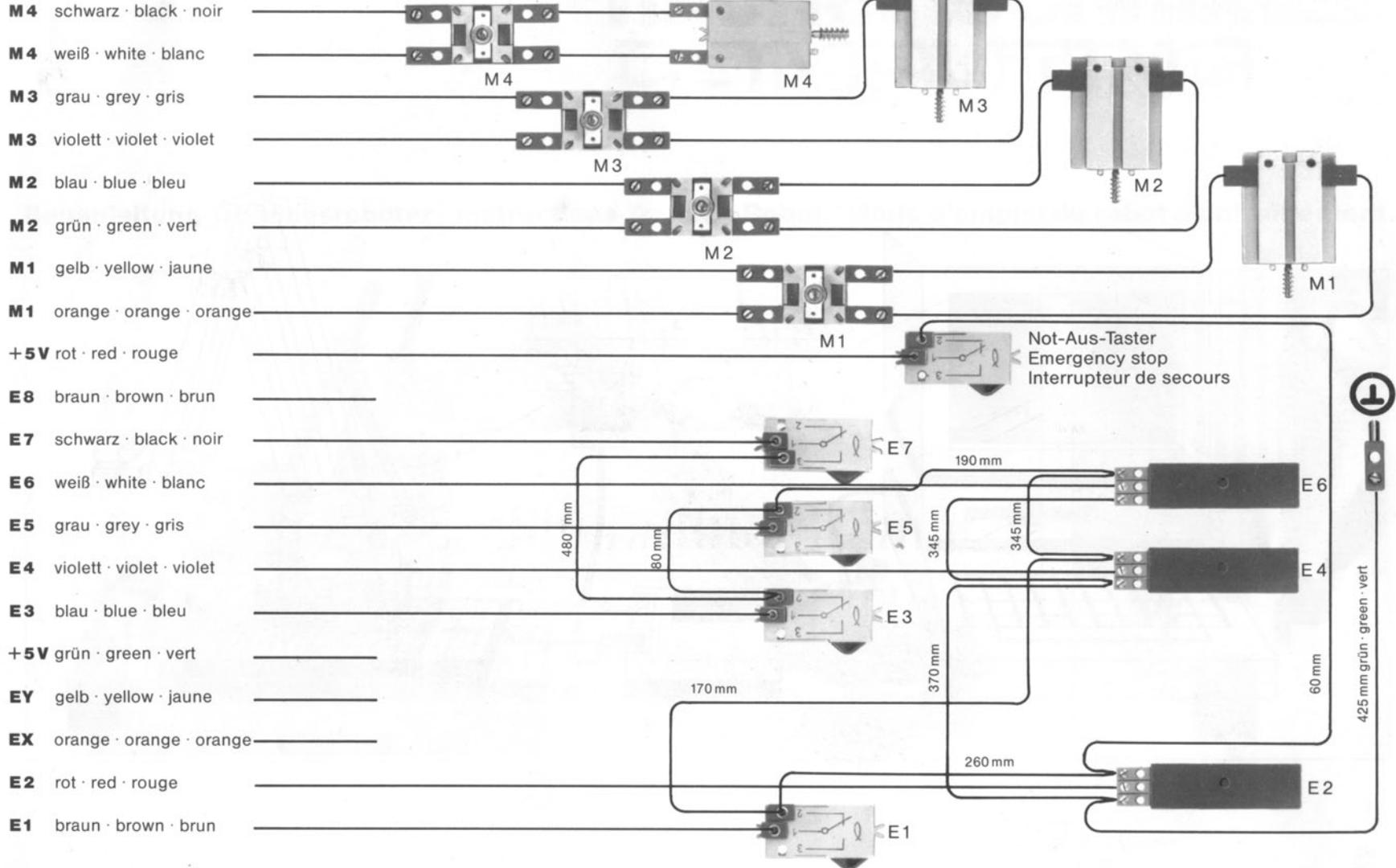


fischertechnik 

- *Anschlußkabel durch den Drehkranz führen
- *Pass cable through life ring
- *Glisser les câbles à l'intérieur de la couronne



Verdrahtungsplan Trainingsroboter · Circuit layout Training Robot · Plan de câblage du robot d'entraînement



fischertechnik computing System

