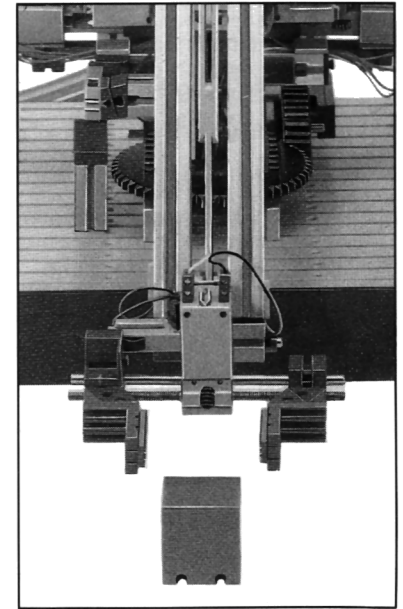


fischertechnik[®] COMPUTING

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 Gabellichtschränke	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 Zählverluste 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 Eigengewicht 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 Schaltdauern 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 „J“ 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.

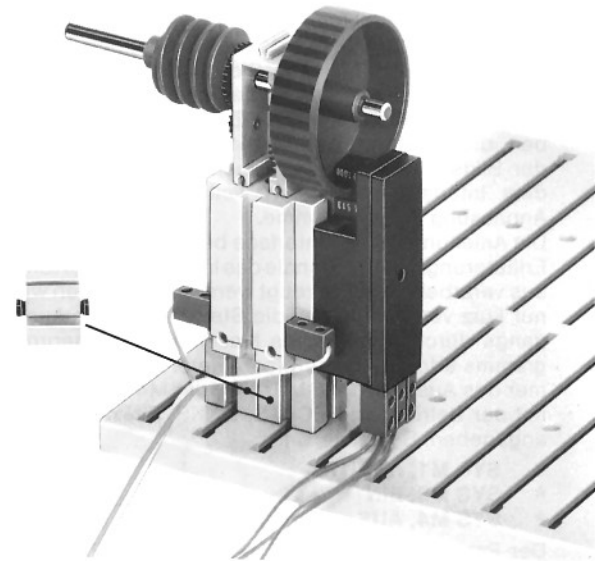
Wenn der Motor läuft, so erscheint an dem Ausgang der Lichtschranke eine Impulsfolge, die genau den dunklen und hellen Stellen des Rades entspricht. Nun sind wir in der Lage, den Roboter mit solch einem Aggregat präzise zu steuern. Wir messen nicht die Laufdauer des Motors, sondern zählen die Impulswechsel 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 beigegeführten 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 beigelegten 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 fischartechnik 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. fischartechnik
7244 Tümlingen/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(E2) USR(E3)
*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 205,96,120,32,247,193,201,0,67041
*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 9,10,44,1,221,16,2,9,72460
*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 237,5,221,32,145,179,88,96,82492
*200 DATA 162,0,192,194,240,18,162,2,83462
*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 207,240,3,189,208,207,157,224,100798
*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 205,240,3,76,30,207,88,96,130488
*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*8+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. Abschließend 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 $\neg$.

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 Spindel Mutter des Oberarm- und Unterarmantriebs 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

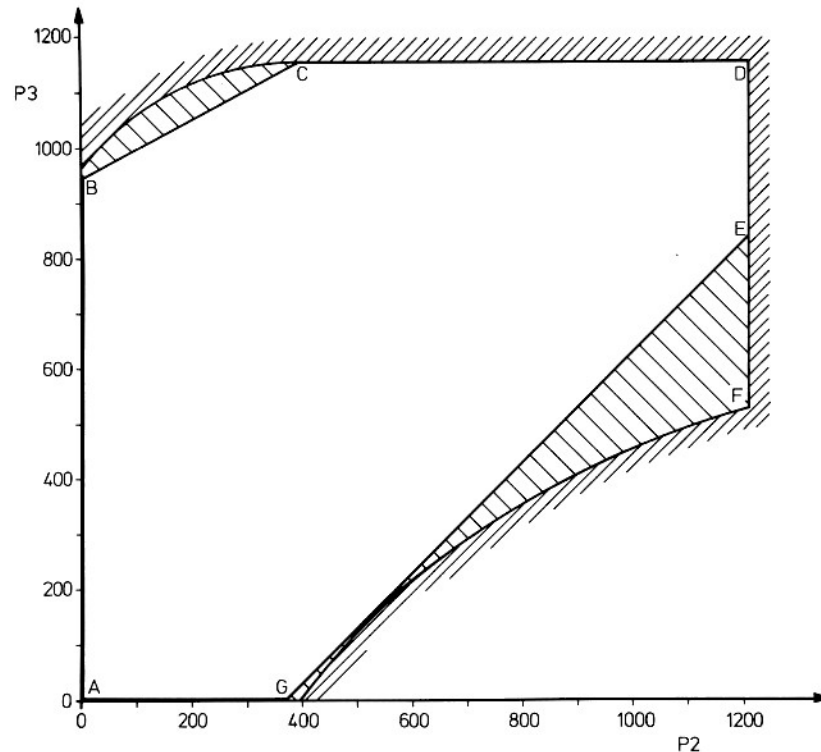
durchlaufen, so daß die Heimposition den Positionswert Null für alle drei Bewegungsachsen erhält.

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

Betätigungsnocks 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.

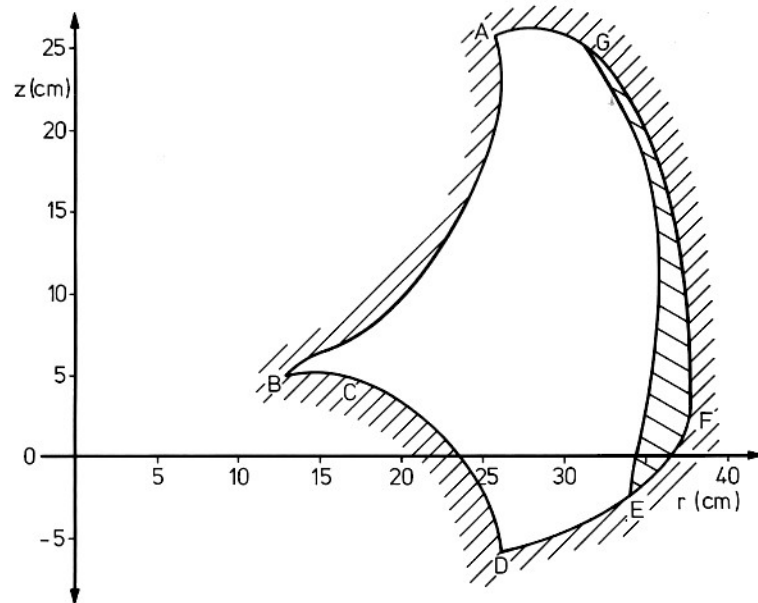
Da das Programm uns immer die Position der Bewegungsachsen auf dem Bildschirm angibt, können wir untersuchen, welche Zahlenkombinationen zulässig sind und welche nicht. Jede Zahlenposition ergibt einen Punkt in dem Achsenkreuz von Bild 2. Alle erlaubten Kombinationen bilden dann eine Fläche, die wir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

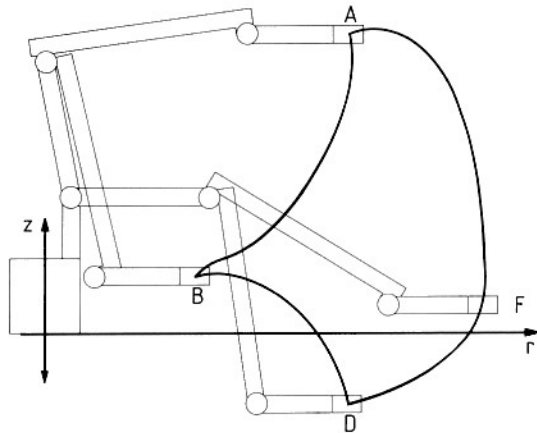
Bild 3



$$\begin{aligned}
 r &= 120 \cdot \cos \alpha + 180 \cdot \cos \beta + 110 \\
 z &= 120 \cdot \sin \alpha + 180 \cdot \sin \beta + 117,5 \\
 \alpha &= 126^\circ - \delta \\
 \beta &= 36^\circ - \varepsilon \\
 \delta &= \cos^{-1} [(14004 - (P2 \cdot 0,07363 + 60)^2) / 12240] \\
 \varepsilon &= \cos^{-1} [(14004 - (P3 \cdot 0,07363 + 60)^2) / 12240] \\
 &\text{(alle Maße in mm)}
 \end{aligned}$$

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.:

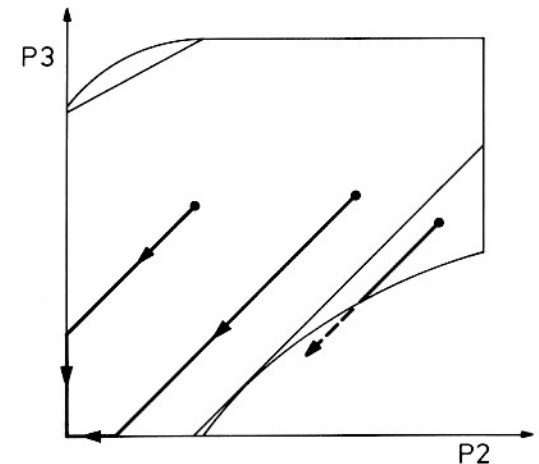
* SYS M1, RECHTS

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 Sight-system, 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 Lösch taste 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

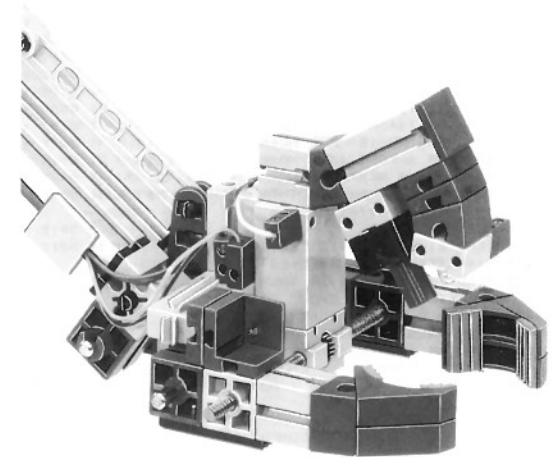
Zum Schluß noch einige Hinweise, wie Sie den Trainingsroboter auch noch einsetzen können. Hierzu sollen Ihnen die folgenden Abbildungen einige Anregungen vermitteln. So kann die Greifzange auch nach unten zeigend montiert werden. Diese Greiferhaltung ist z.B. zum Greifen von Schachfiguren geeignet. Sicherlich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





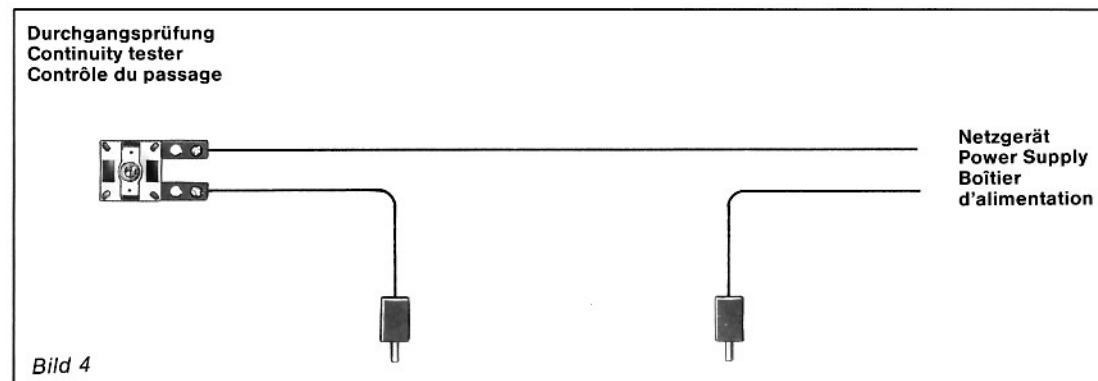
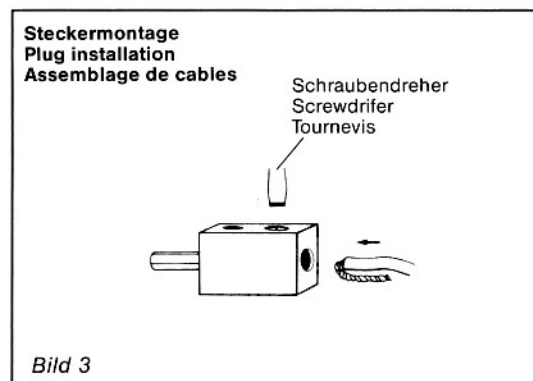
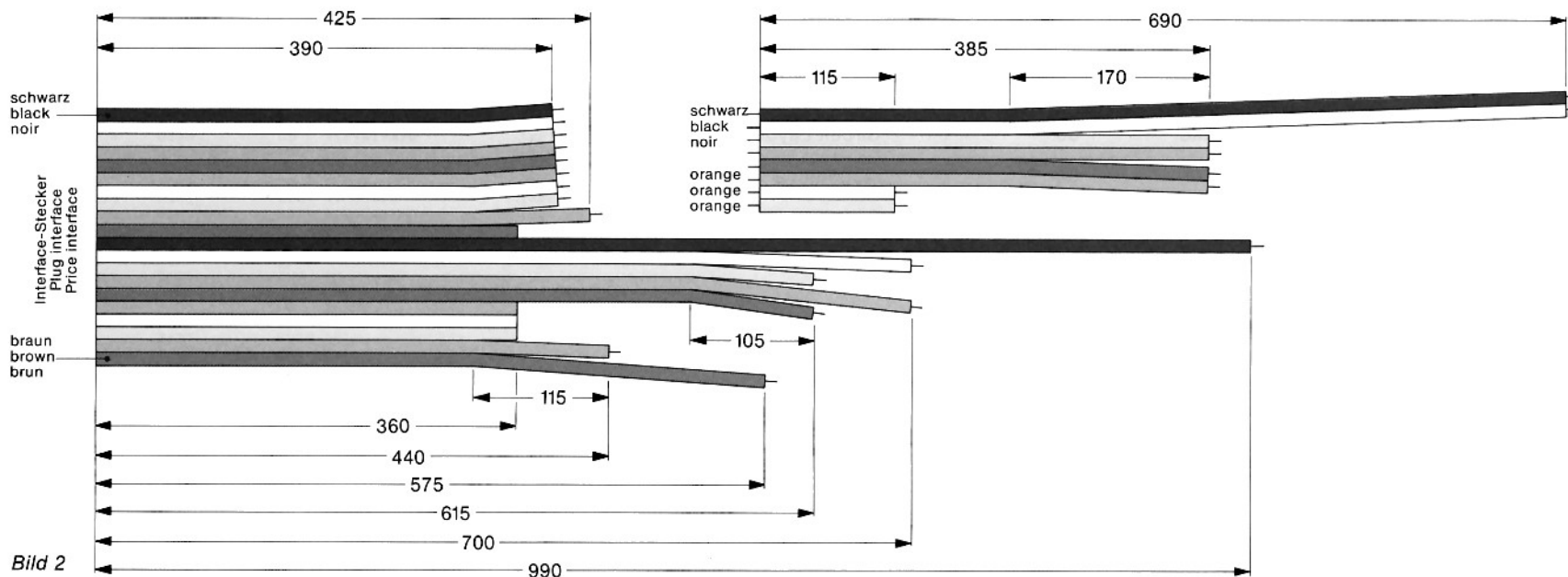
Bebilderte Bauanleitung

Kabelkonfektionierung	62
Verdrahtungsplan Gabellichtschranke	63
Mechanischer Aufbau	64
Verdrahtungsplan Trainingsroboter	91

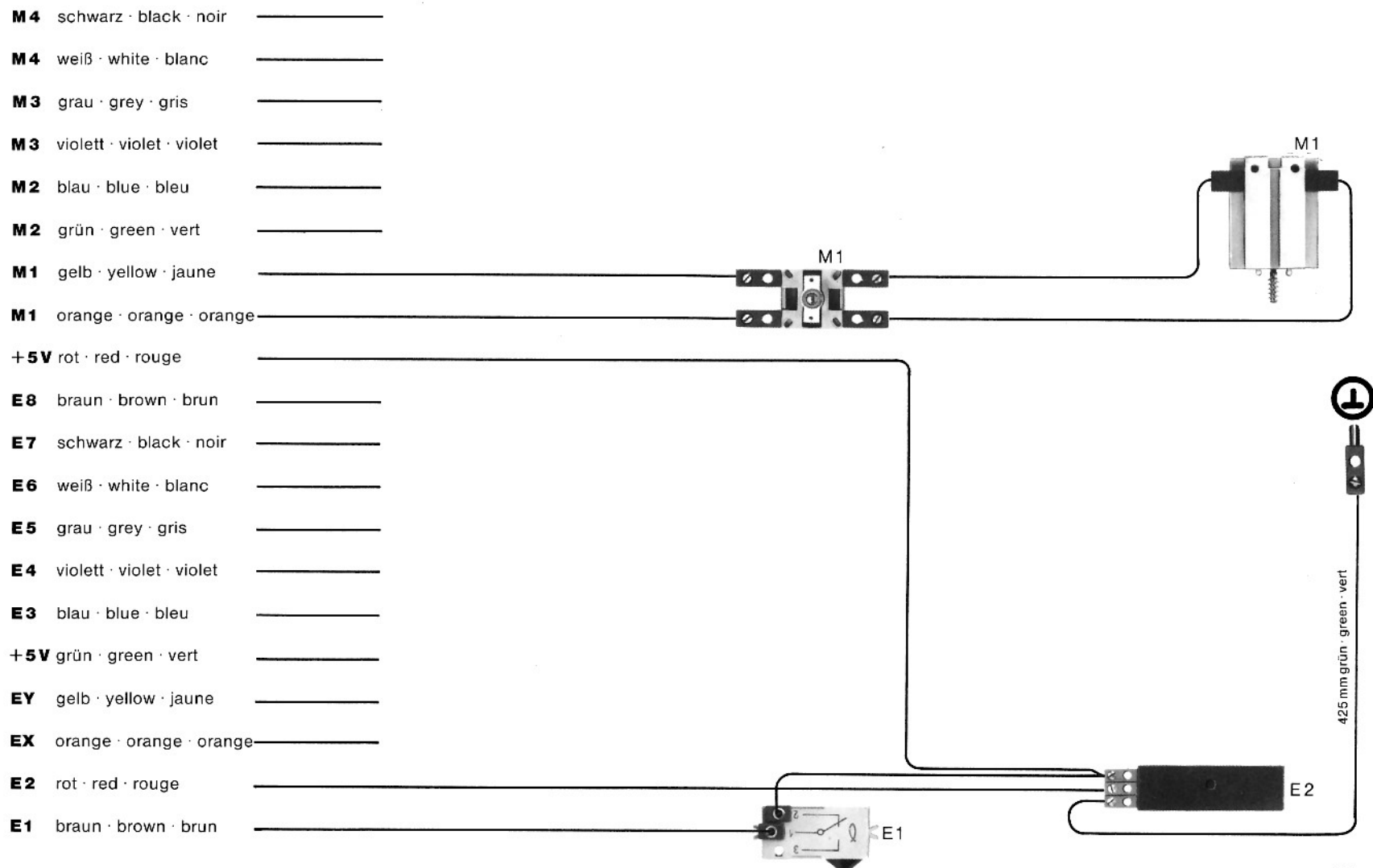
Illustrated Assembly Instructions

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

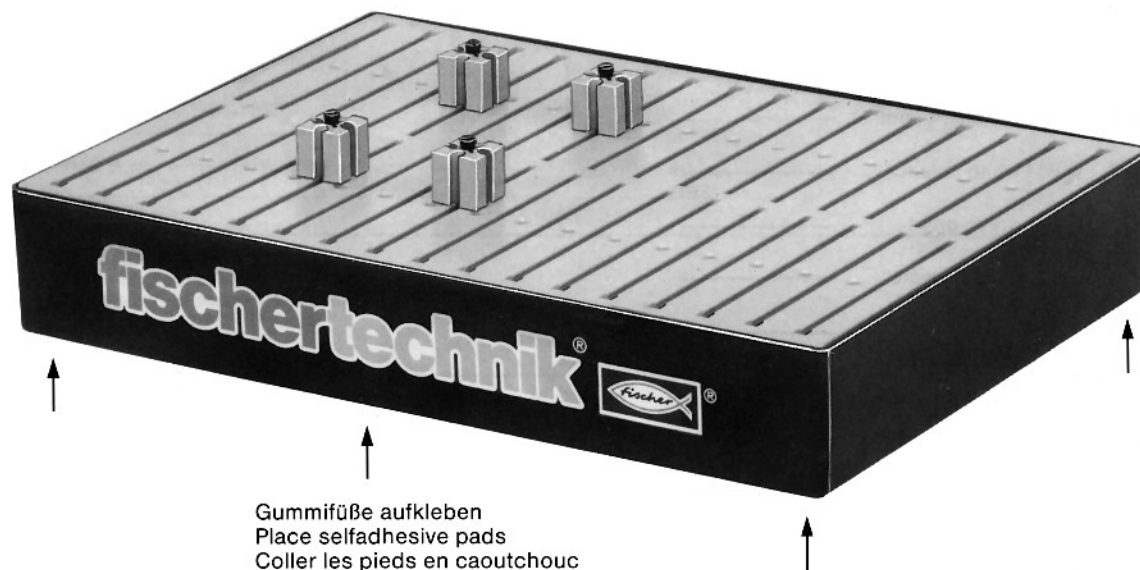
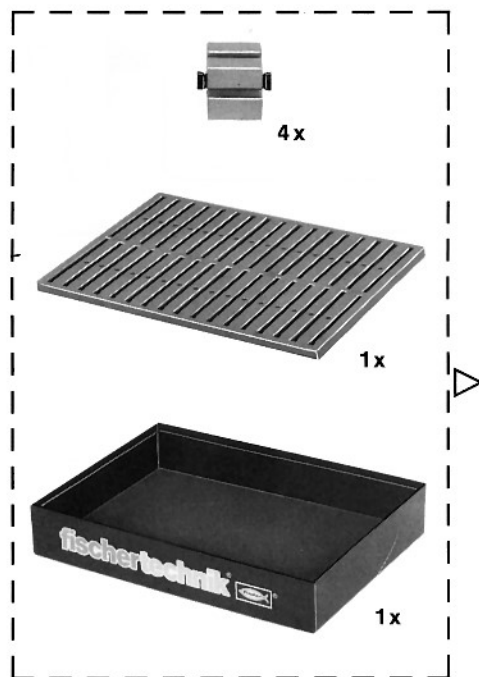
Kabelkonfektionierung · Ribbon cable configuration · Schéma de câblage



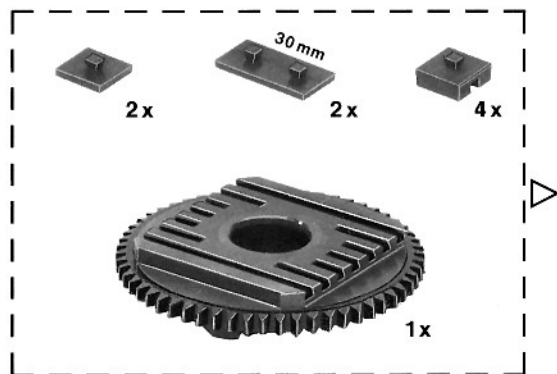
Verdrahtungsplan Gabellichtschranke · Circuit Layout Photo-interrupter · Plan de câblage d'interrupteur photo



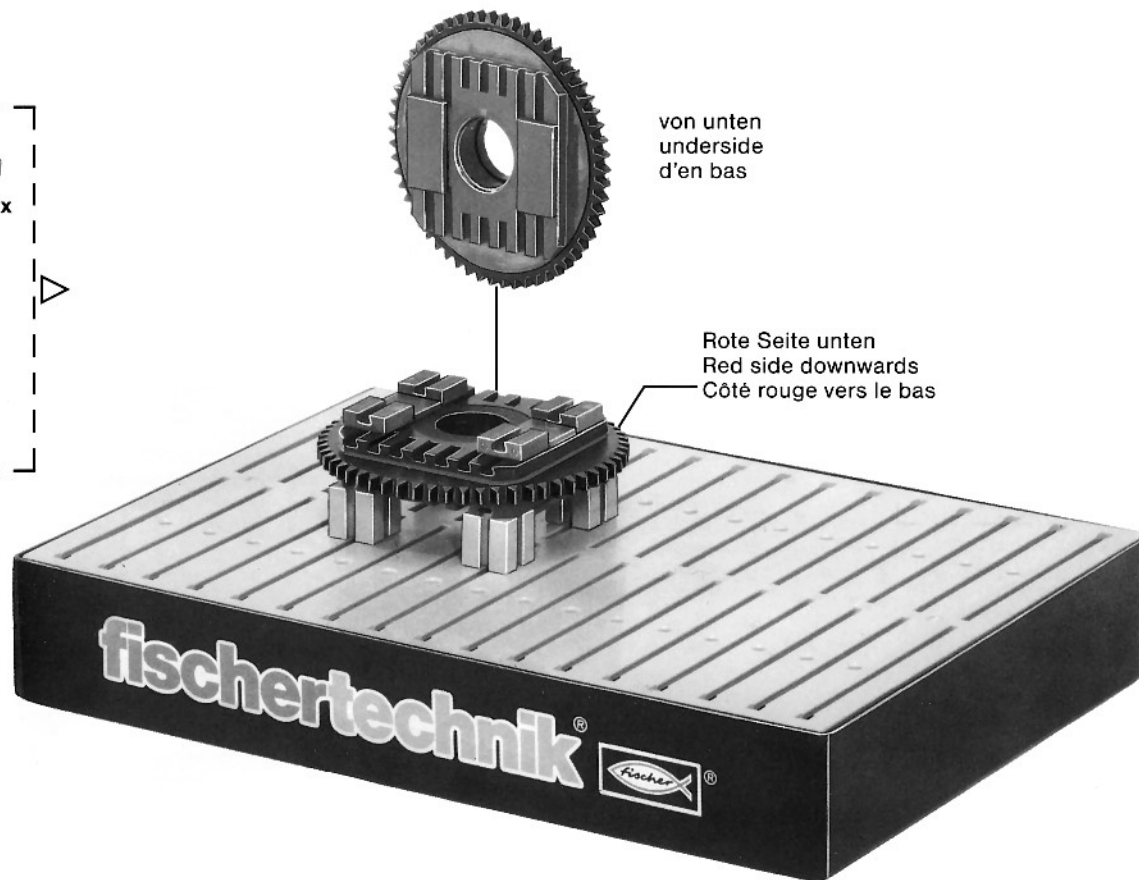
1



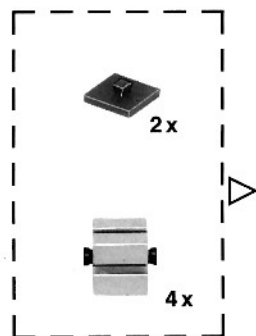
2



0 30 mm



3



4

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

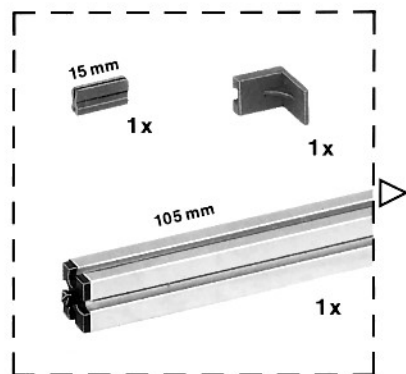
For technical reasons, the shearing forces of the structural metal rods are partially very high.

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

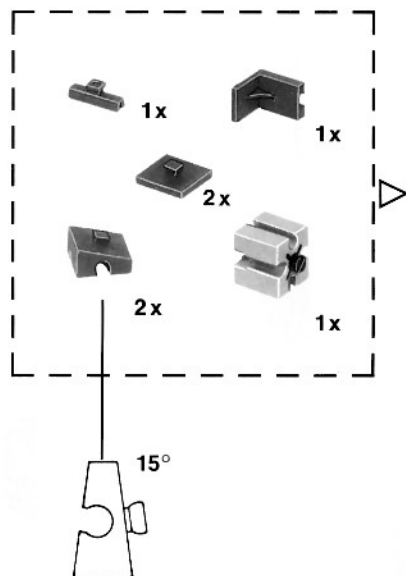


0 75 mm

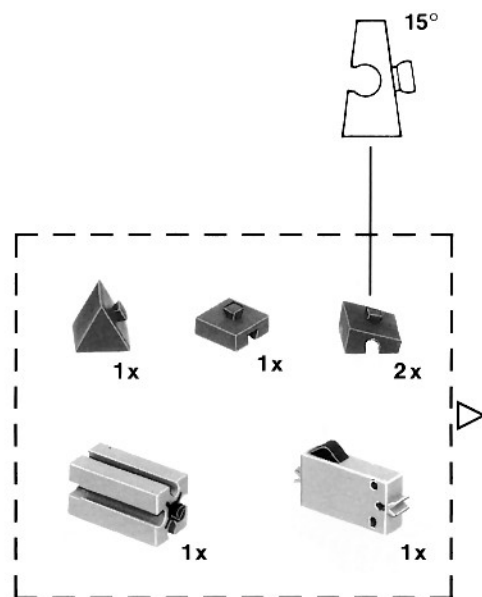
5

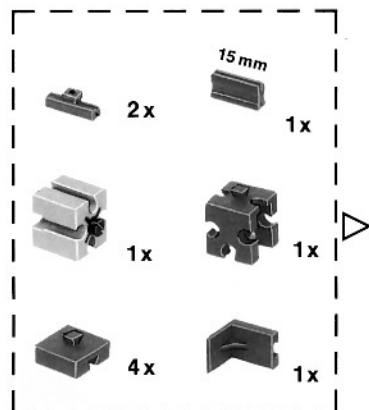


6



7

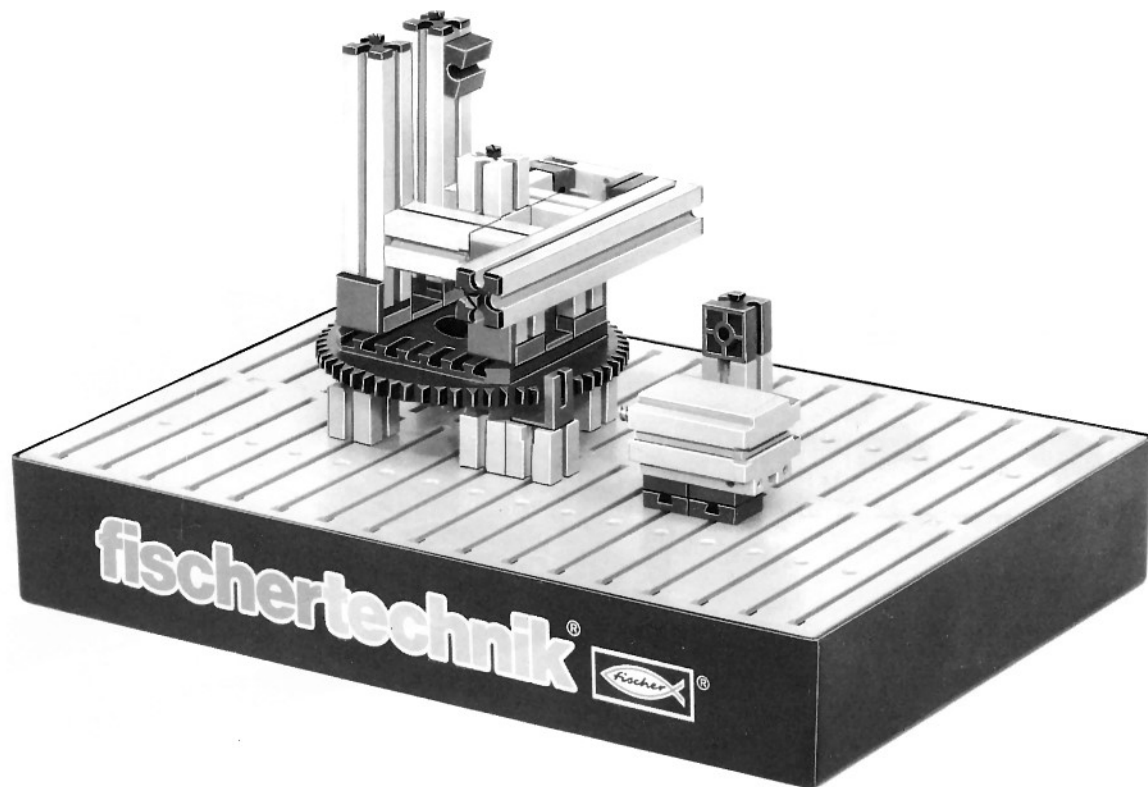
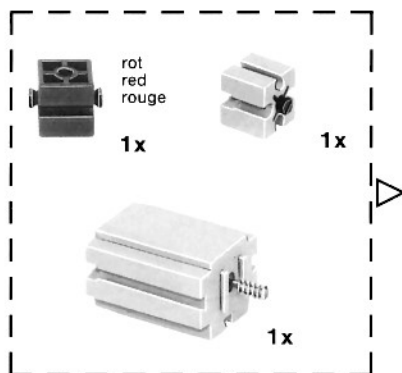
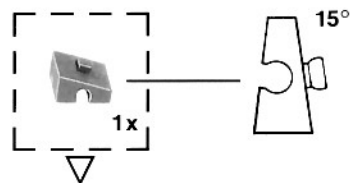


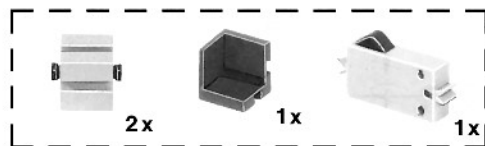
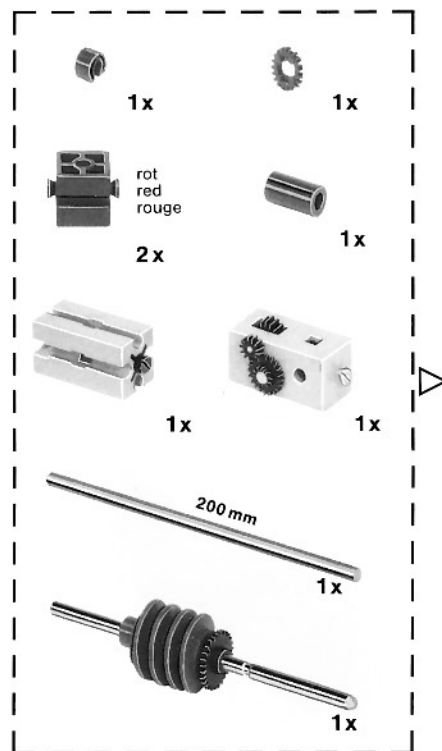


0 15 mm

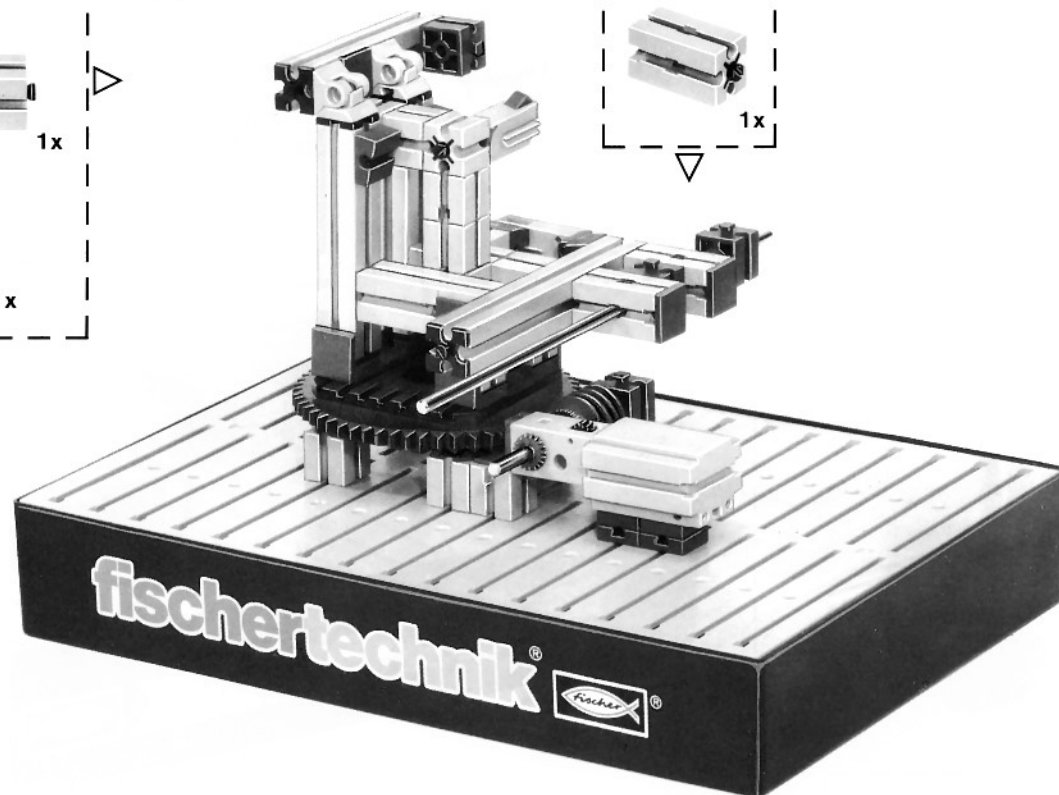
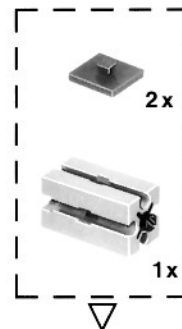
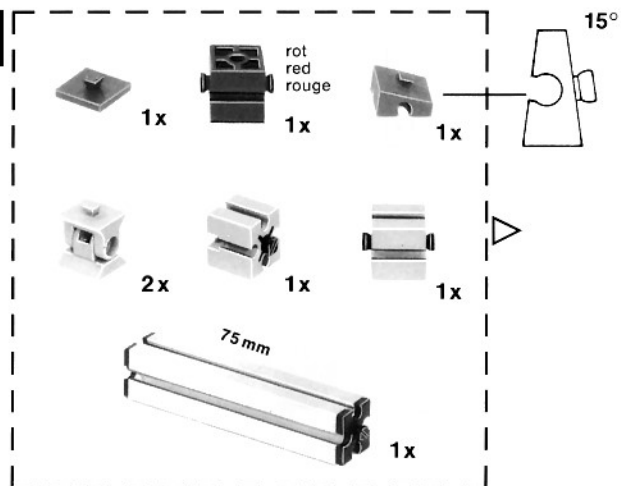


9

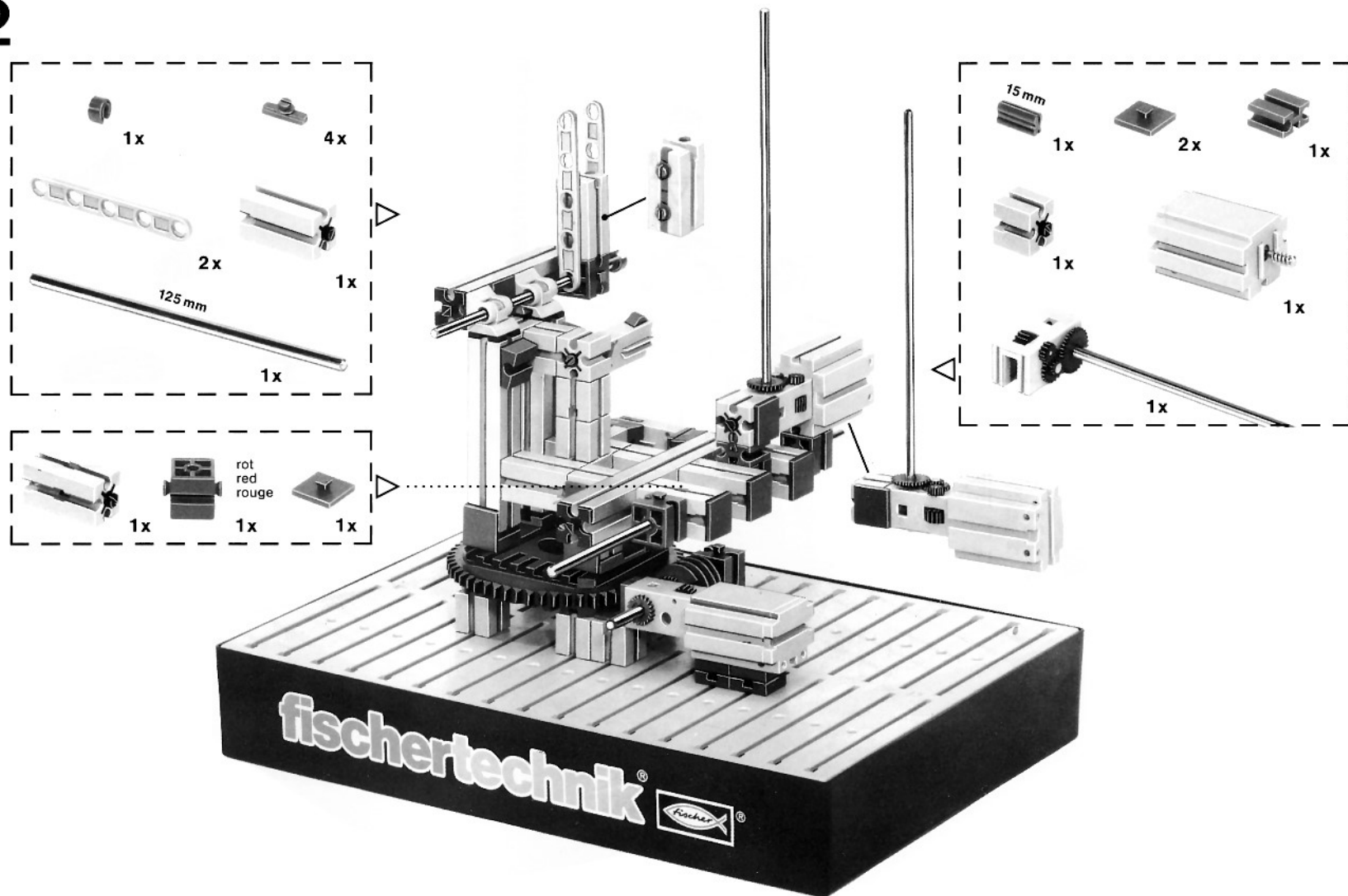


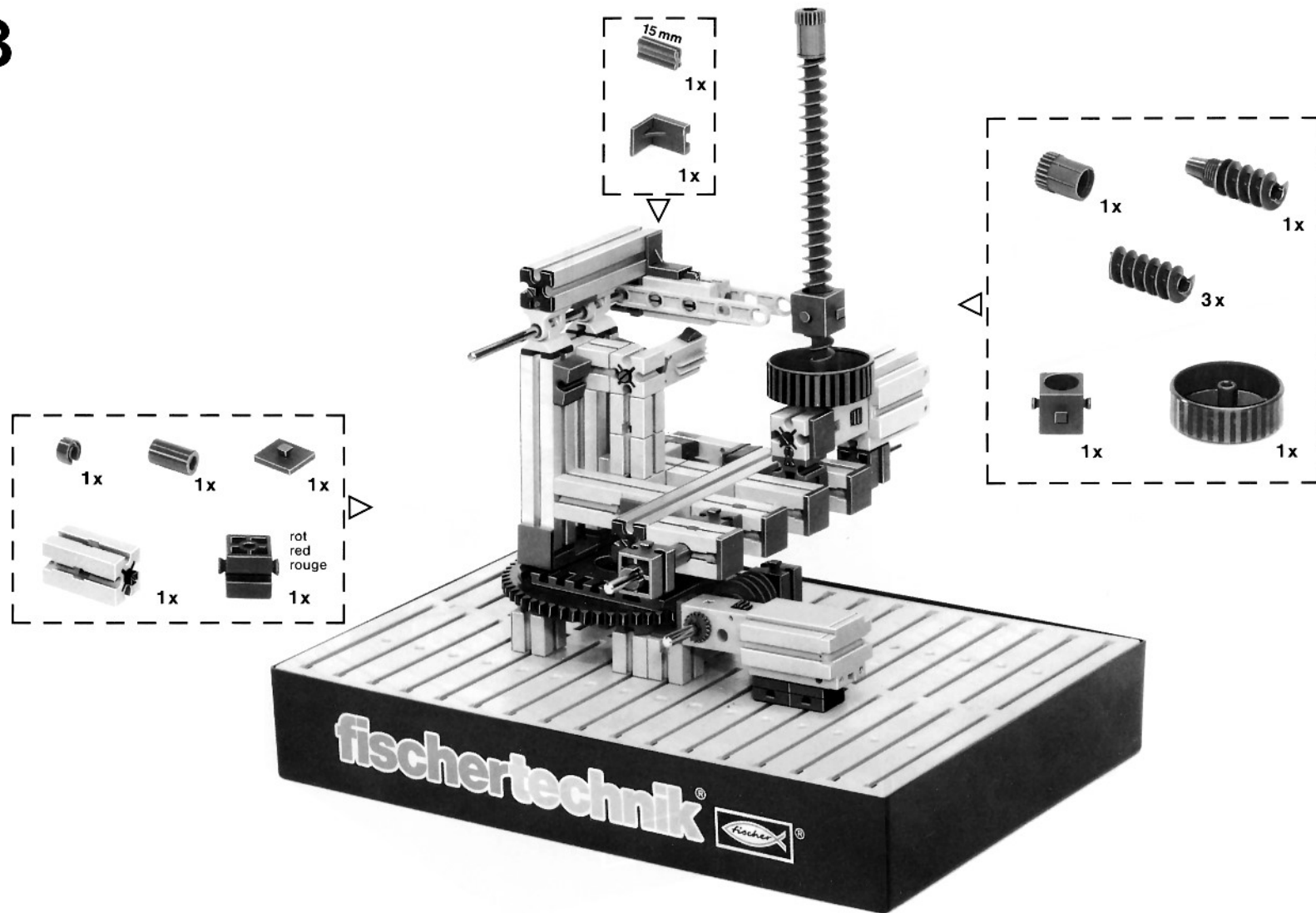


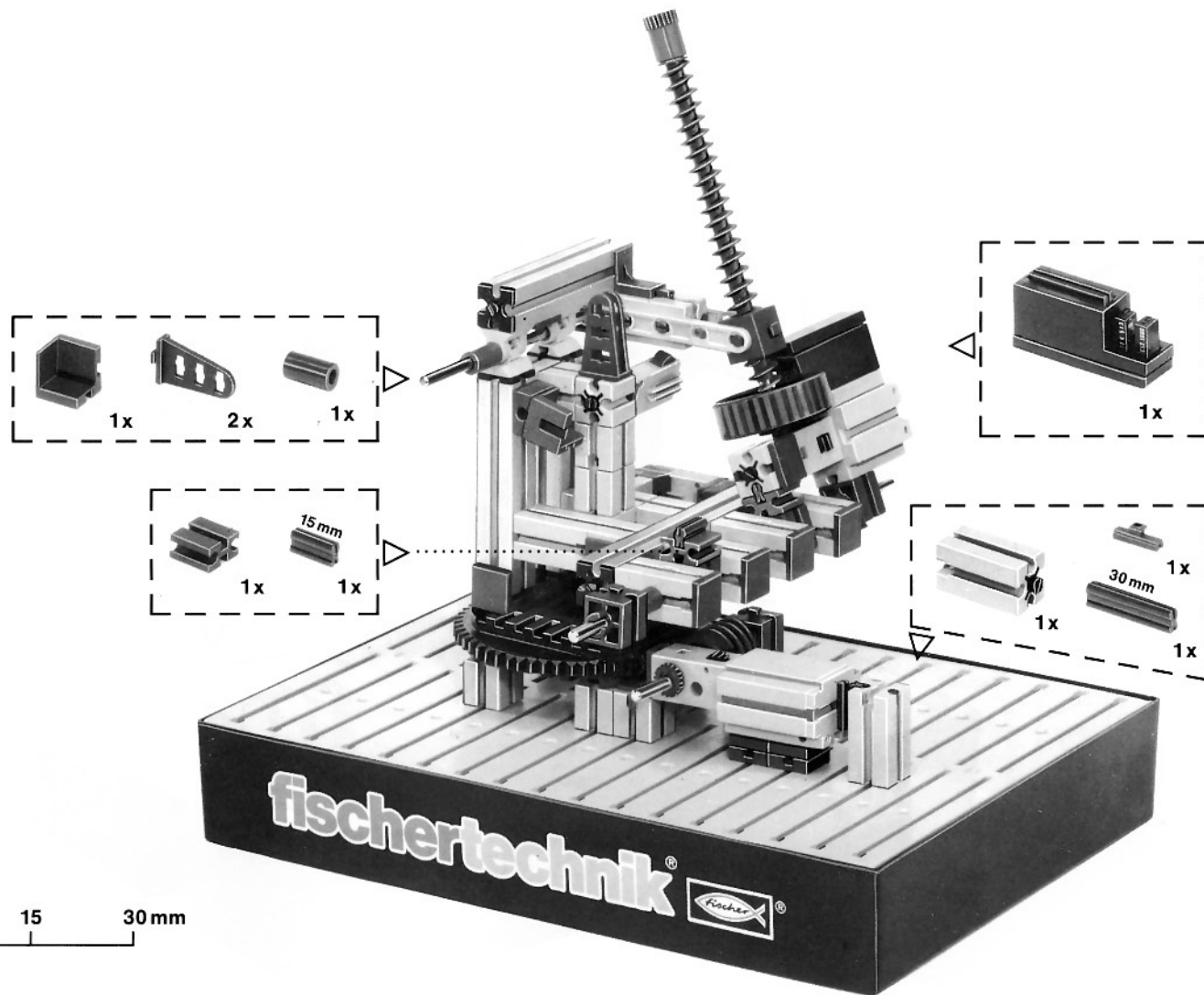
11

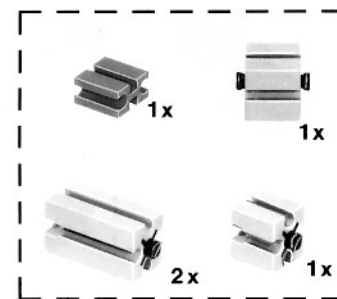
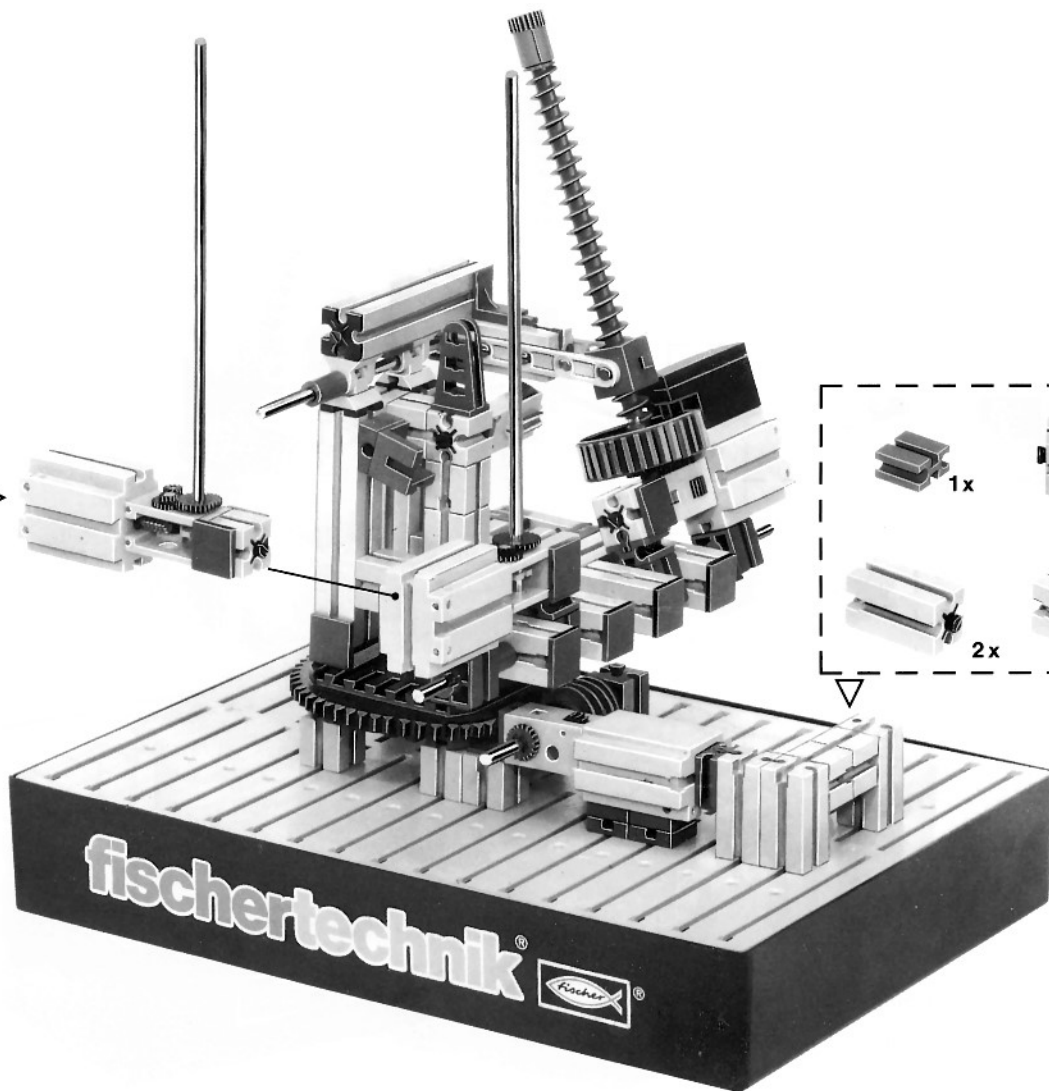
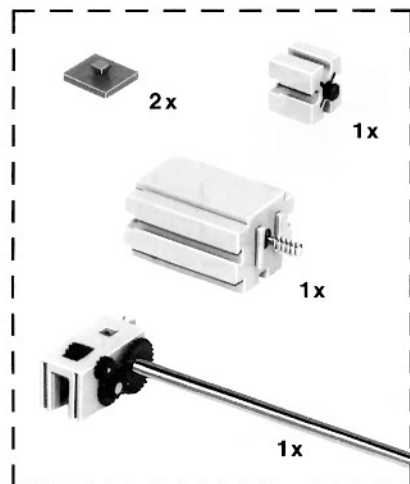


12

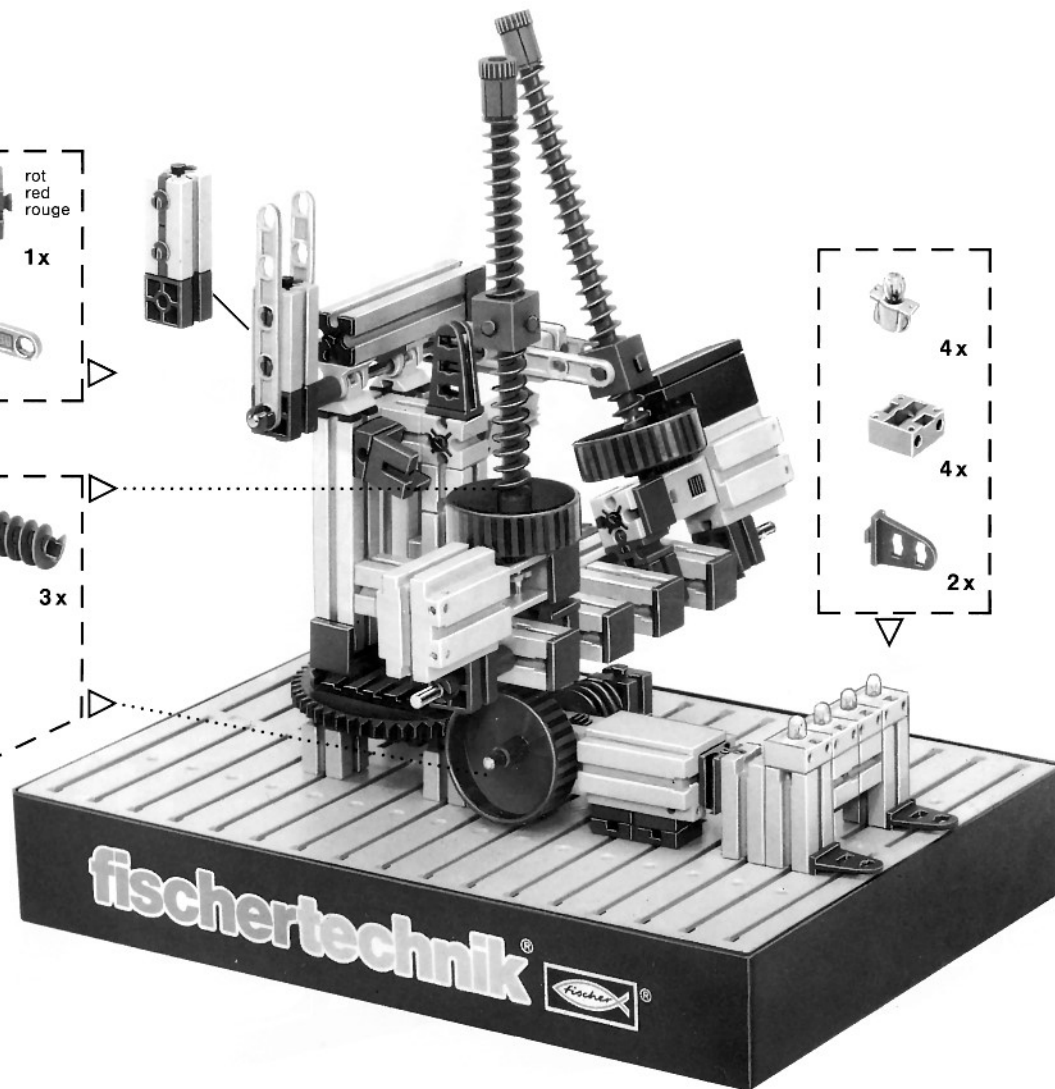
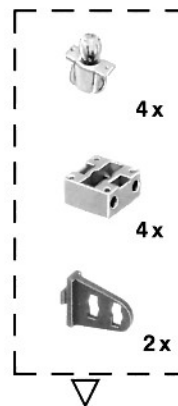
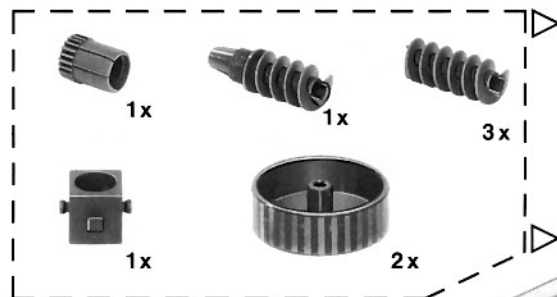
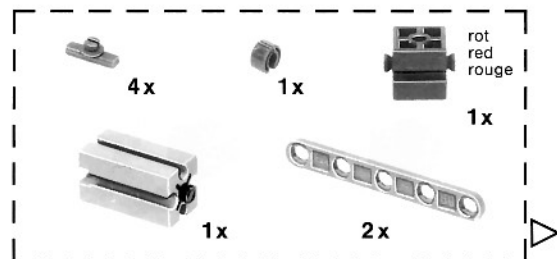


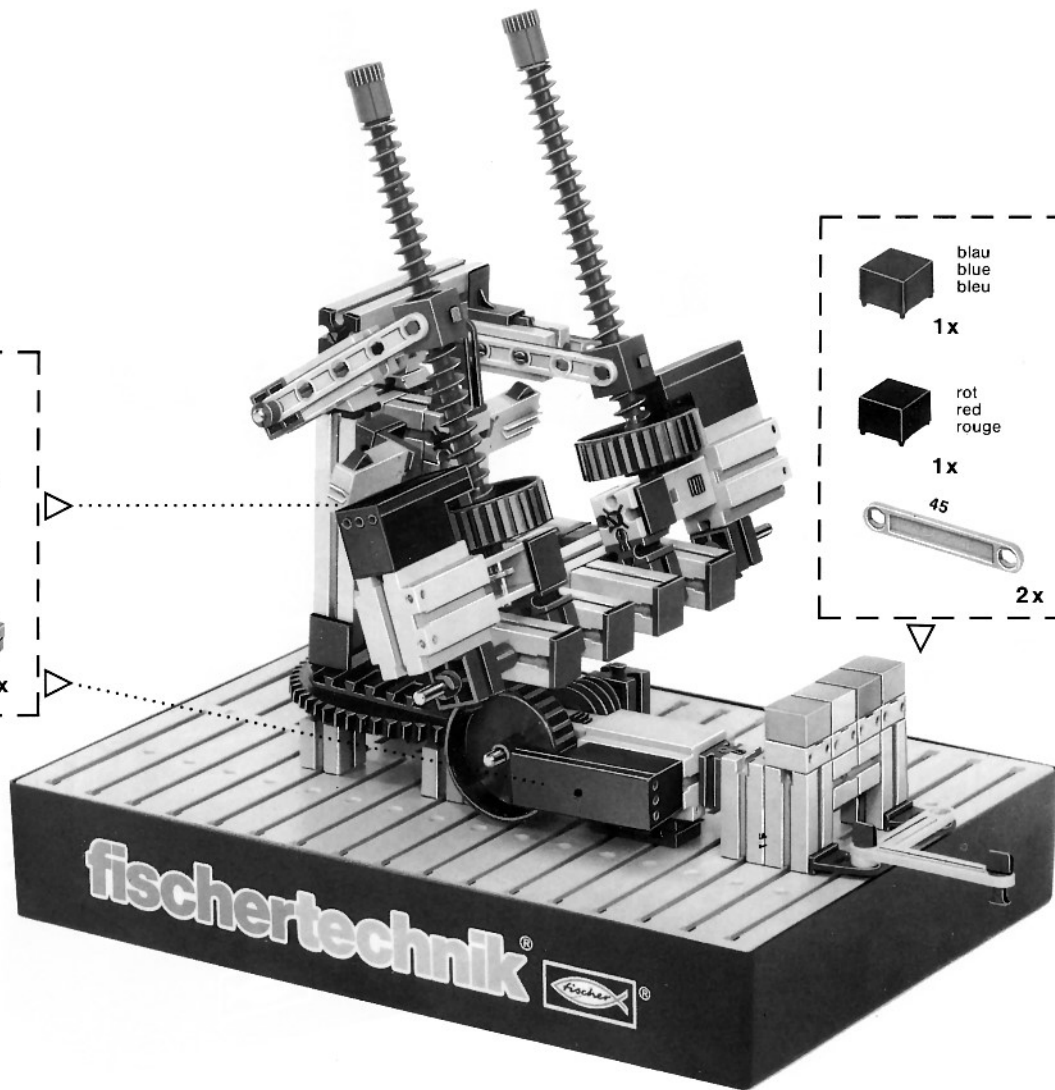
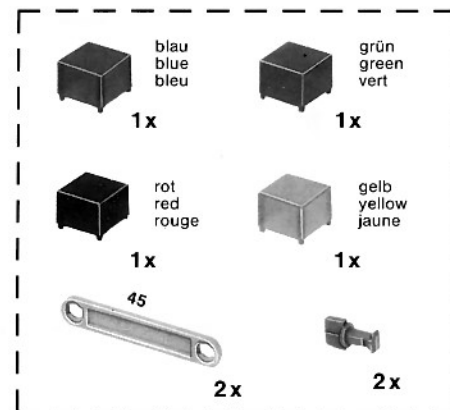
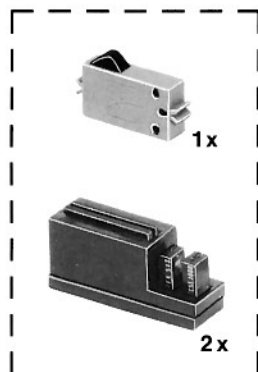




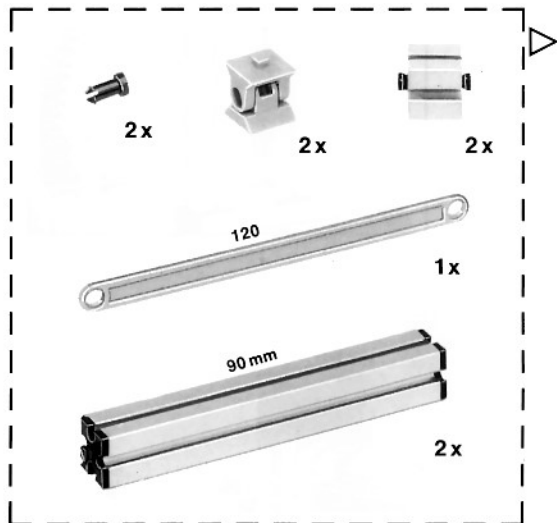
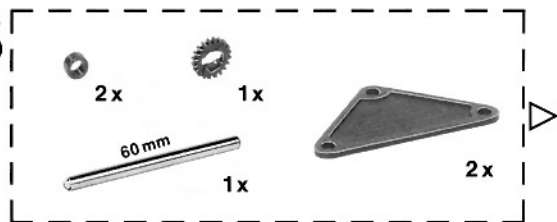


16



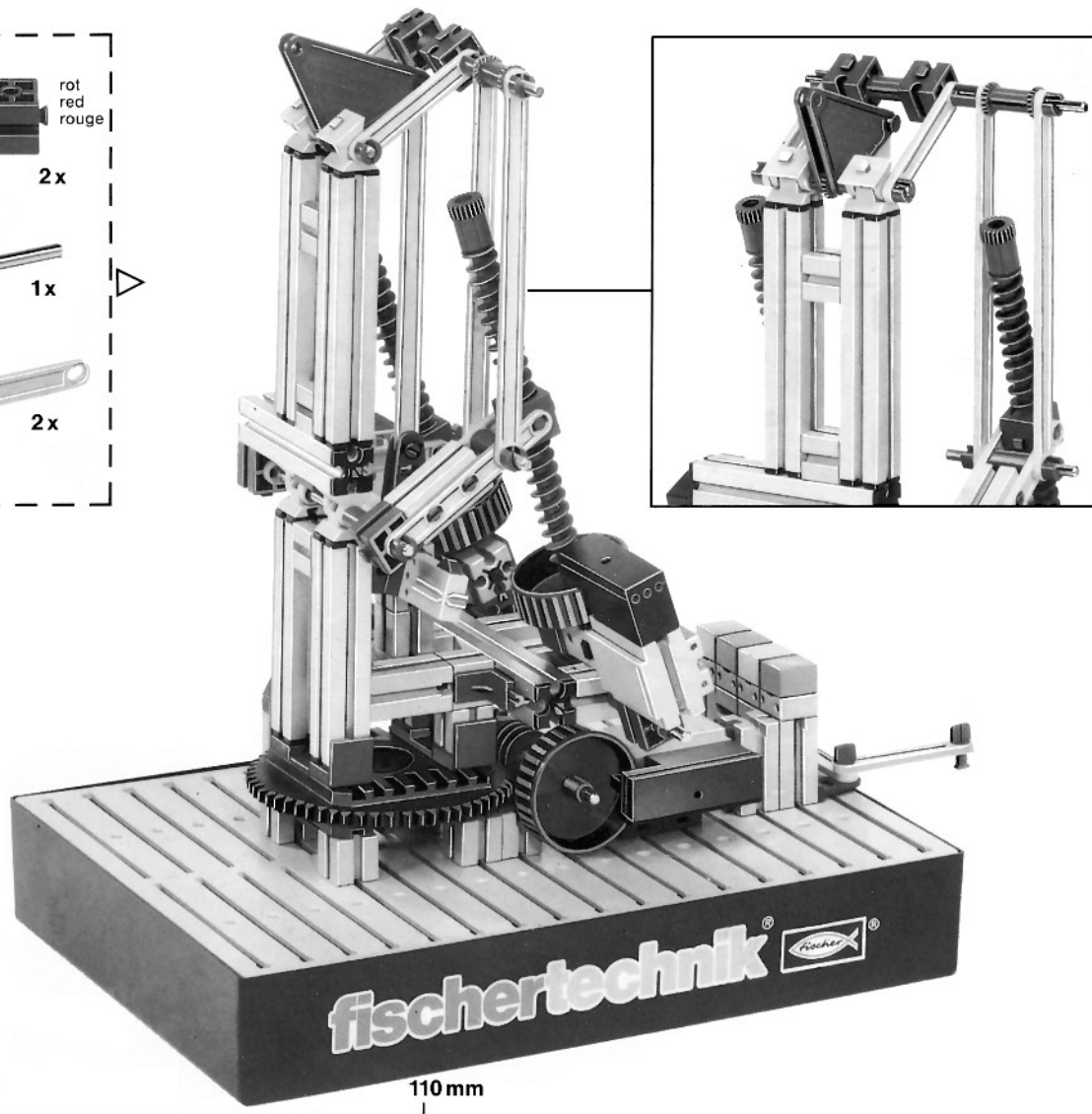
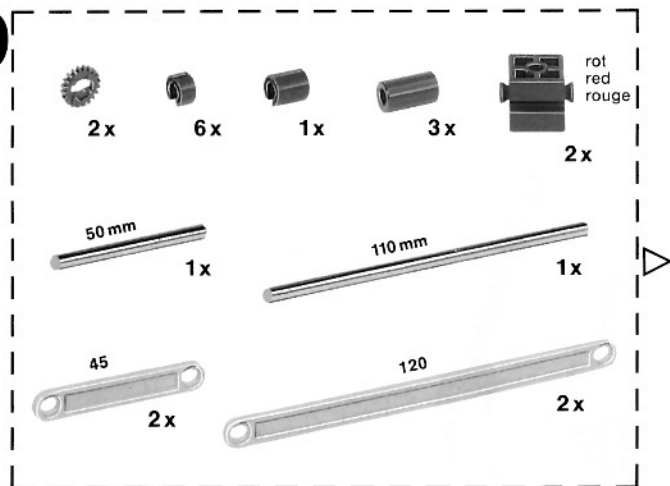


18

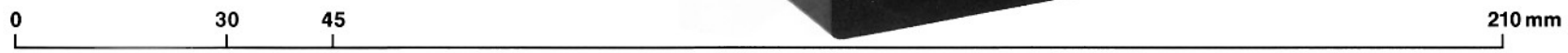
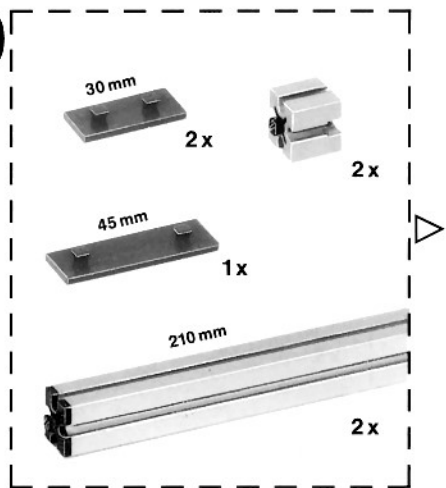


0 60 90 mm

19



20



21



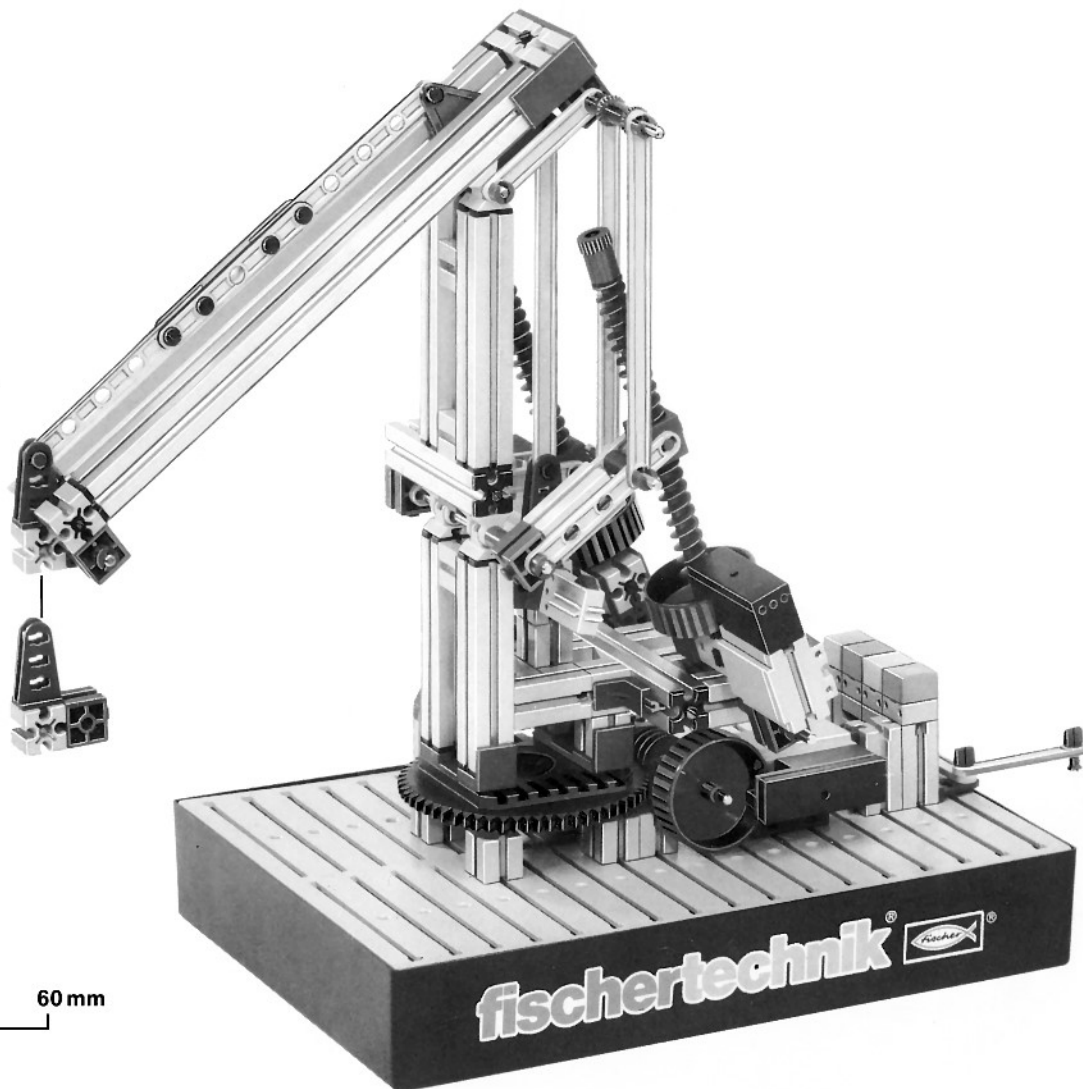
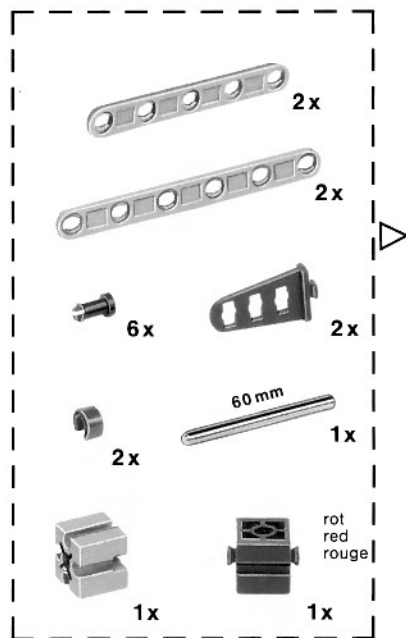
3x



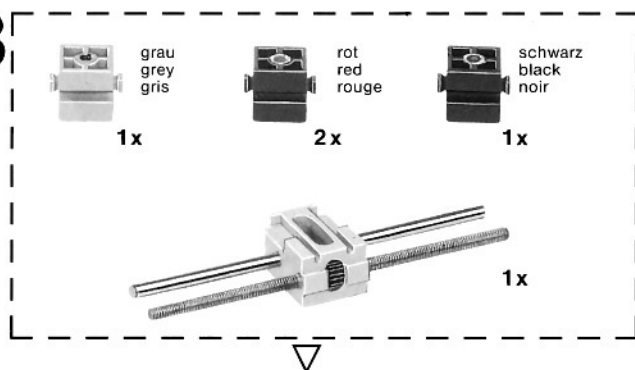
rot
red
rouge

2x

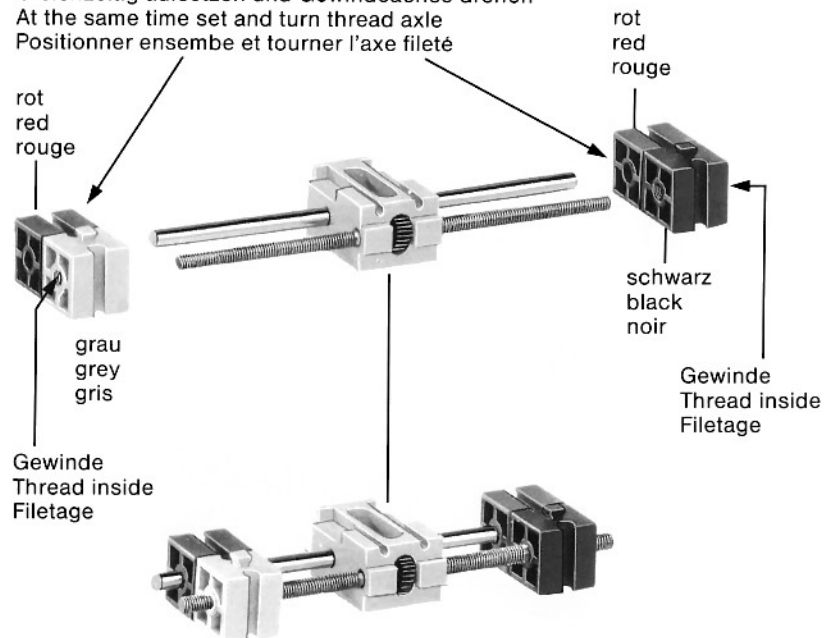




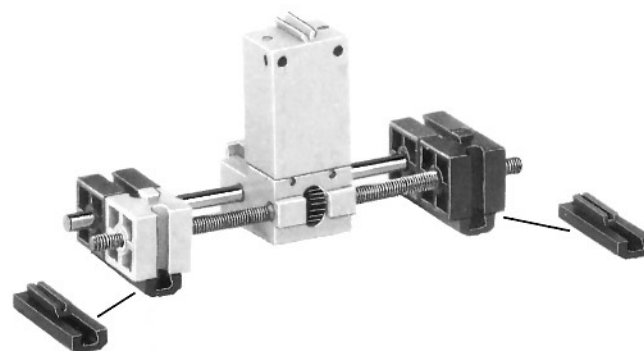
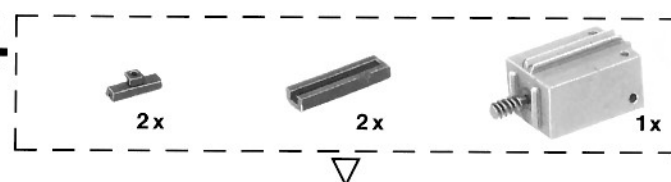
23



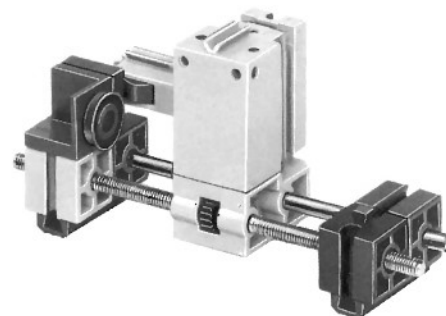
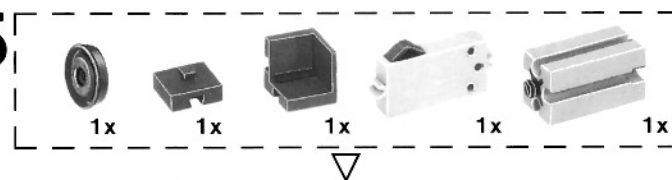
Gleichzeitig aufsetzen und Gewindeachse drehen
At the same time set and turn thread axle
Positionner ensemble et tourner l'axe fileté



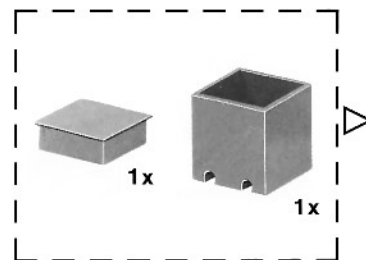
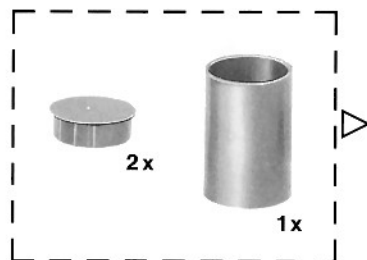
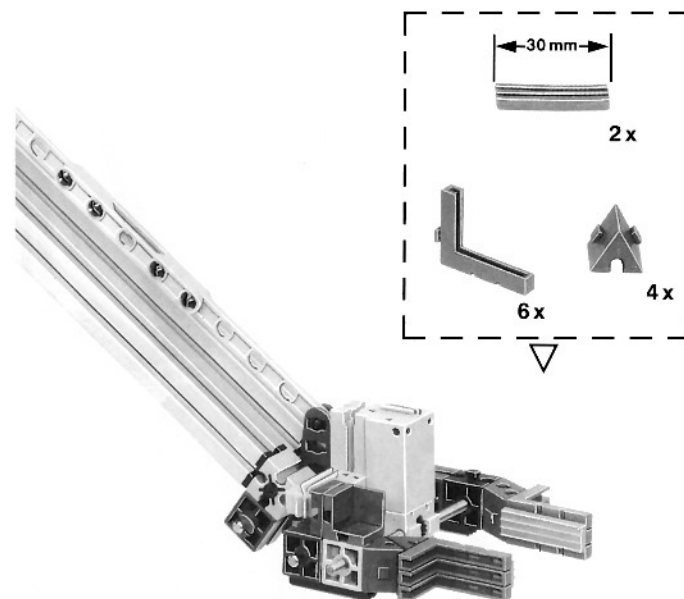
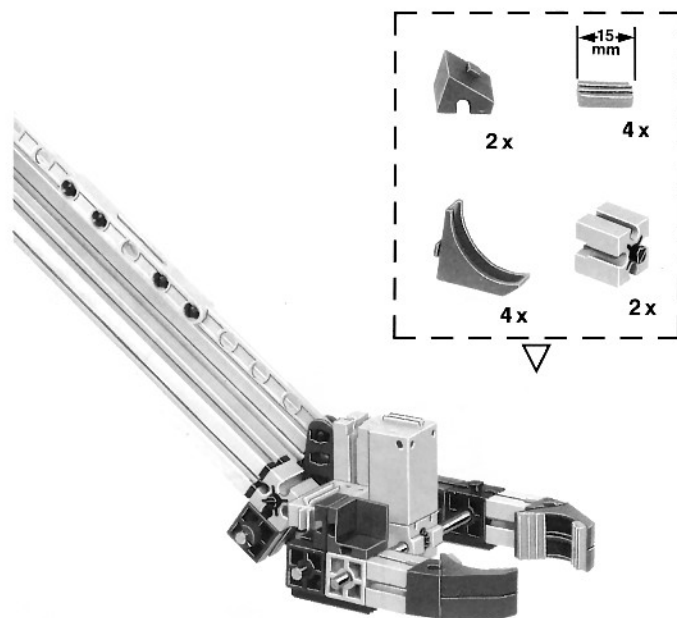
24

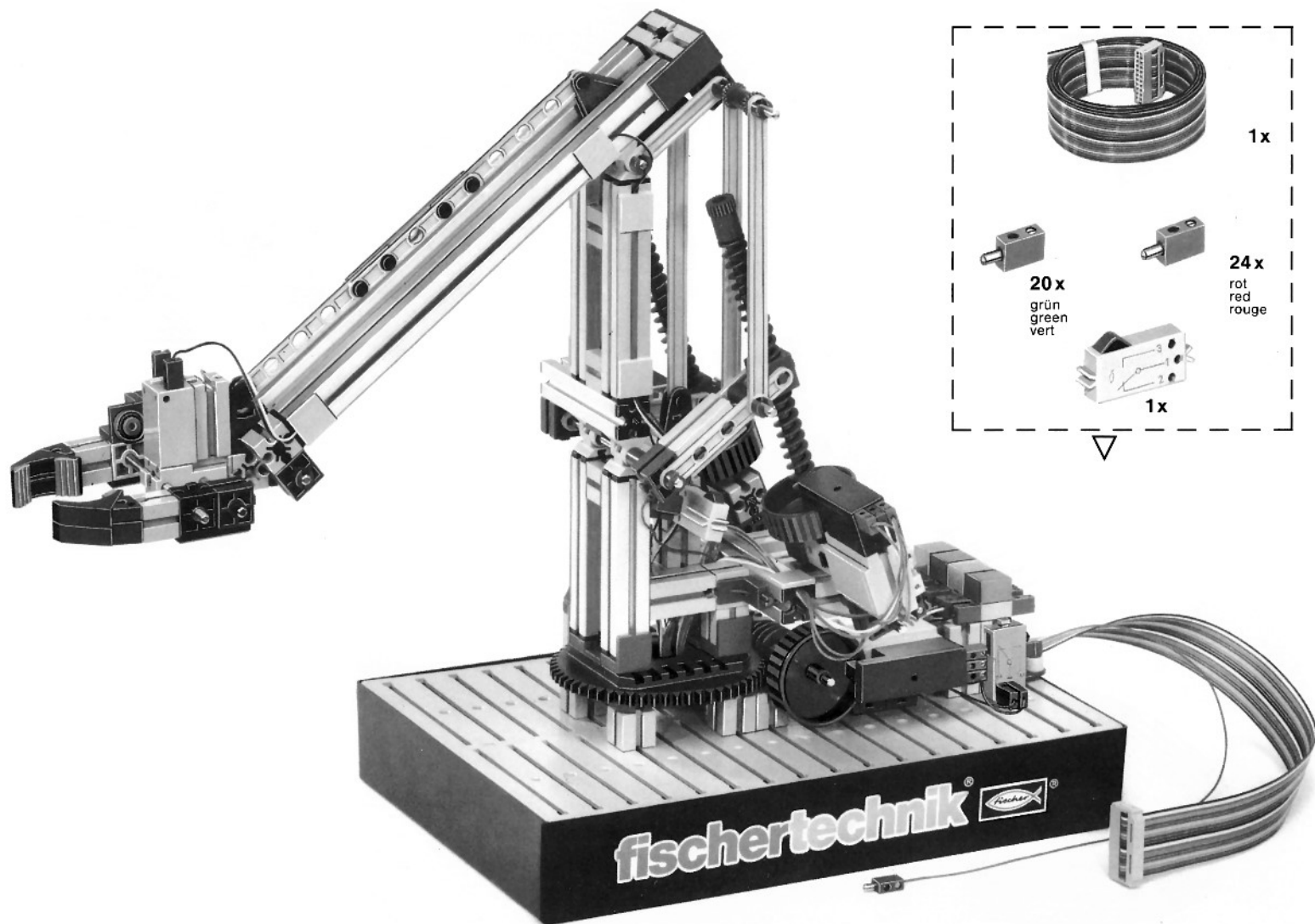


25

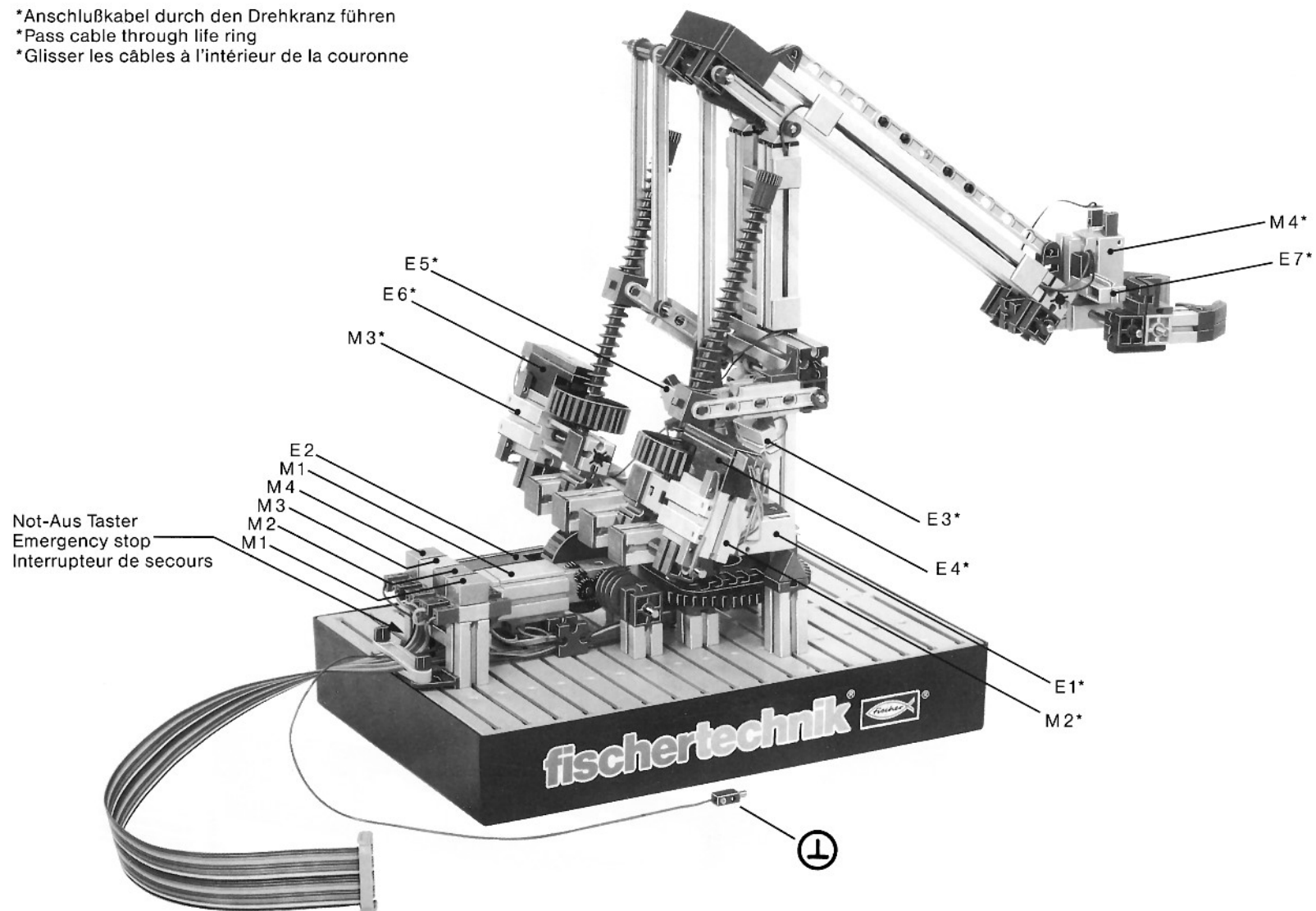








- *Anschlußkabel durch den Drehkranz führen
- *Pass cable through life ring
- *Glisser les câbles à l'intérieur de la couronne



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

