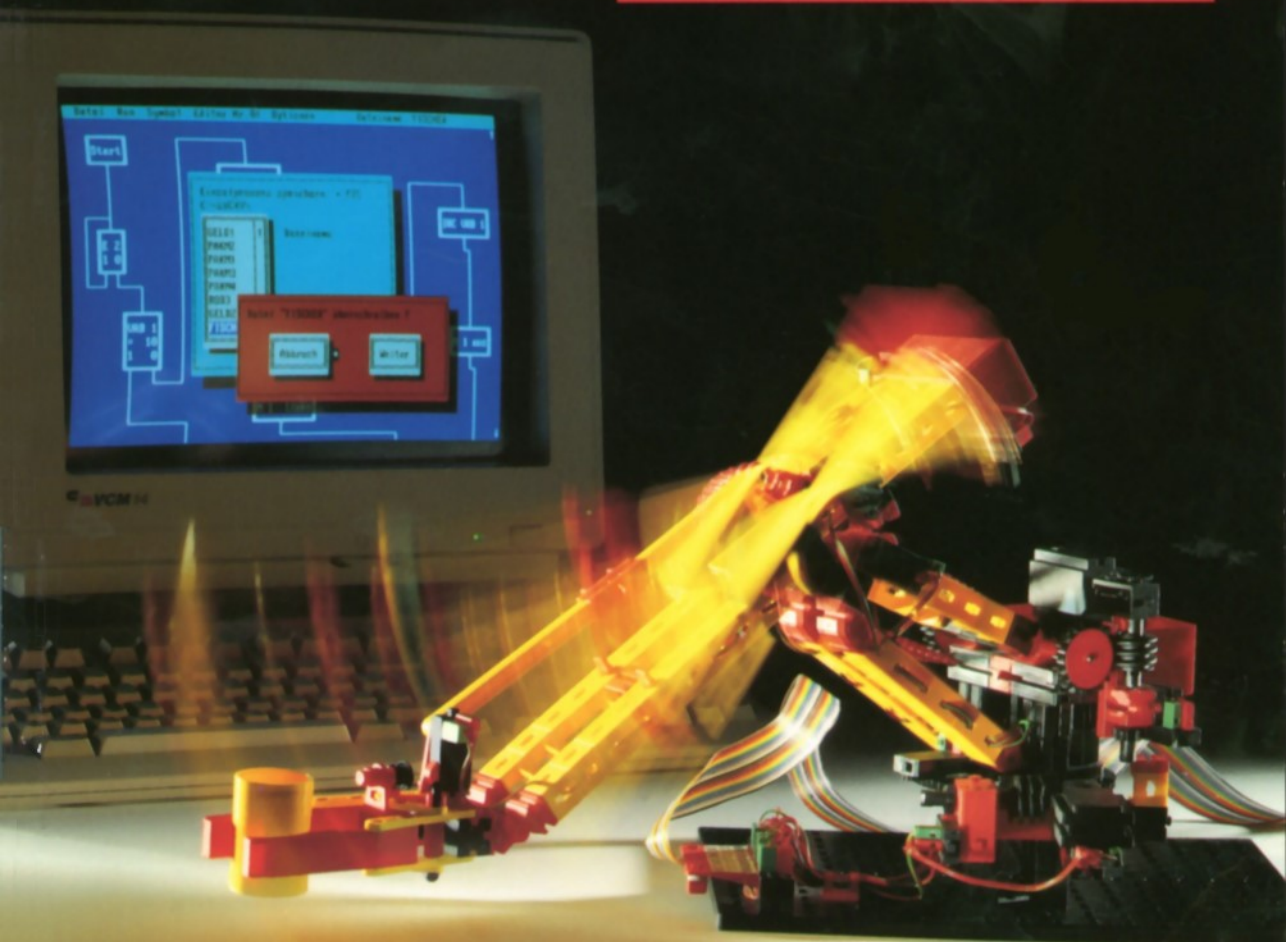


EXPERIMENTIERBUCH

PROFI COMPUTING



Experimentierbuch

PROFI COMPUTING

fischertechnik® 

Pflaum Verlag München

Bildquellen-Nachweis

Schmuckbild S. 33: Bilderdienst Süddeutscher Verlag München

S. 34/35: IWEKO Werbe GmbH, 5401 Halsenbach

S. 42/43: Bilderdienst Süddeutscher Verlag München

S. 52/53: H. Baer, Stahl- und Panzergeldschränke, Spieljochstr. 22a, München 82

S. 68/69: Deutsche Bundespost TELECOM, Bonn

S. 114/115: Fotostudio Otto, Wien

S. 126/127: H. H. Kropf, Königsbrunn-Zang

S. 138/139: KUKA Schweißanlagen + Roboter GmbH, Augsburg

ISBN 3-7905-0625-7

Copyright 1991 by Richard Pflaum Verlag GmbH & Co., KG, München · Bad Kissingen · Baden-Baden · Berlin · Düsseldorf · Heidelberg

Alle Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Funksendung, der Wiedergabe auf fotomechanischem oder ähnlichem Wege und der Speicherung in Datenverarbeitungsanlagen bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten.

© für die Abbildungen fischerwerke Arthur Fischer GmbH & Co. KG, Tümlingen

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, daß solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Satz: typozentral Ingrid Geithner, Erding

Druck: Pflaum Verlag München

Inhalt

Profi Computing	7
Bevor du anfängst ...	9
Was Du noch brauchst	9
Steckermontage	9
Kabel konfektionieren	9
Abisolieren und Stecker montieren	10
Papierdrahtband und Kabelklips	11
Netzgerät	11
Interface	12
Steuersoftware	12
Stromversorgung	12
Verbindung vom Computer zum Interface	13
Verbindung vom Interface zum Modell	13
Ein wenig Elektrotechnik	15
Der Strom fließt im Kreis	15
Elektrotechnik „nach Fahrplan“	16
Elektrische Bauteile	16
Montage der mechanischen Bauteile	19
Aufbau der Modelle	22
Ein paar Tips zur Software	22
Reaktionstester	29
Melodie-Spiel	35
Turtle (Schildkröte)	43
Tresor mit Code-Schloß	53
Geldautomat	59
Codekartenleser mit automatischem Einzug	69
CD-Player	79
Sortieranlage	93
Paket-Wendeanlage	103
Kurvenschreiber	115
Plotter	127
Roboter	139
Stückliste	156

Profi Computing

Lieber fischertechnik-Freund,

als das „Herz“ des Computers, der Mikroprozessor, entwickelt wurde, dachte zunächst niemand an die heute alltäglichen Programme wie Textverarbeitung, Buchhaltung, Grafik, Spiele, Programmiersprachen usw., sondern man wollte einen frei programmierbaren Schaltkreis für das Steuern von Geräten und Maschinen schaffen.

Sicher kennt jeder die Berichte aus Zeitung, Radio und Fernsehen über Computer, Roboter und die Automatisierung ganzer Fabriken. Aber auch daheim findet man oft kleine Computer zur Steuerung der Funktionen: im Aufzug, im Videocoder oder CD-Player, in der Waschmaschine, in der Mikrowelle, im Drucker der Computeranlage oder im Auto.

Natürlich gab es Steuerungen schon, als der Mikrocomputer noch nicht erfunden war. Diese Steuerungen waren aber fest verdrahtet und mußten bei Änderungen im Arbeitsablauf der jeweiligen Maschine mühsam neu verdrahtet werden — ein enormer Zeit- und Geldaufwand. Mit dem Mikroprozessor sind diese Änderungen problemlos durch Ändern der Steuerprogramme möglich; Löten und Umbauen ist nicht mehr notwendig. Damit war der Weg zur computergesteuerten Maschine frei. Speziell bei Drehbänken, Fräs- und Bohrautomaten werden kleine Computer zur Steuerung eingesetzt. Die Programmierung erfolgt direkt über eine Tastatur oder durch die Übertragung von Steuerdaten, die auf einem anderen Computer erzeugt werden.

Von solchen „numerisch gesteuerten Werkzeugmaschinen“ (numerisch = durch Zahlenangaben) kam dann der Schritt zu Robotern mit noch mehr Fähigkeiten. So ist es heute möglich, daß auf derselben Produktionsstraße ganz unterschiedliche Ausstattungs- und Modellvarianten eines Autos gebaut werden können — ob mit zwei oder vier Türen, Schiebedach oder Turbo-Motor, per Computer kann man die Herstellung aller Modelle exakt steuern.

Die Schwerarbeit wird dabei von Robotern erledigt. Der Name „Roboter“ ist noch nicht sehr alt. Vor etwa siebzig Jahren erfand ihn der tschechische Schriftsteller Karel Capek: Er schrieb über einen Wissenschaftler, der künstliche Menschen erfand — Puppen, die sich scheinbar selbstständig bewegen. Diese Maschinenmenschen nannte er „Roboter“, abgeleitet vom tschechischen Wort „robota“, das soviel bedeutet wie „Sklavenarbeit“. In hunderten von Science-Fiction-Romanen und -Filmen wurde diese Idee von anderen Autoren wieder aufgegriffen.

Die wirklichen Roboter unterscheiden sich wesentlich von den Erfindungen der Phantasie. Sie sind keine furchterregenden, superintelligente Wesen aus Metall, sondern Maschinen, die von Computern gesteuert werden und ganz bestimmte Arbeiten ausführen und die auch keine eigene Intelligenz besitzen. Roboter sind jedoch im Gegensatz zu „normalen“ Maschinen recht flexibel und lassen sich für die unterschiedlichsten Aufgaben einsetzen. Durch den Wechsel des Computerprogramms kann derselbe Roboter heu-

te ein Auto lackieren und nächste Woche Karosserieteile zusammenschweißen (die Farbspritzpistole wird dann natürlich durch ein Schweißgerät ersetzt).

Roboter läßt man meist dort arbeiten, wo es für den Menschen zu schwer, zu gefährlich oder zu ermüdend ist. In Fabriken werden sie auch deswegen eingesetzt, weil sie rationeller arbeiten können — sie brauchen weder Feierabend noch Ferien, weder Schlaf noch Essen. Und in Kernkraftwerken oder auf weit entfernten Planeten verrichten sie Arbeiten, die der Mensch überhaupt nicht ausführen kann. Die Intelligenz und Kreativität der Kombination aus Computer und Roboter kommt jedoch einzig und allein vom Programm — und das wird vom Menschen gemacht.

Um die Steuerung von Maschinen und Robotern geht es auch im fischertechnik Profi Computing Baukasten. Mit den ver-

schiedenen Modellen machen wir einen Ausflug in die industrielle Steuerung per Computer. Motoren werden zu „Armen und Händen“ des Computers, Fotozellen oder Taster bilden seine Sinnesorgane — Computer und Modell arbeiten dann Hand in Hand. Dieser Experimentierkasten demonstriert die Möglichkeiten von Computersteuerungen im Kleinen. Mit Anleitung und spezieller Steuerungssoftware von fischertechnik kommst Du auch schnell mit der Programmierung zurecht. Selbstverständlich kann fischertechnik Profi Computing mit allen anderen fischertechnik-Komponenten (z.B. Profi Sensoric) kombiniert werden, und Du kannst Deiner Phantasie freien Lauf lassen. Wenn Du mit den Modellen und der Software vertraut geworden bist, kannst Du versuchen, eigene Ideen zu verwirklichen oder einzelne Modelle zu kombinieren.

Bevor Du anfängst ...

... noch ein paar wichtige Arbeiten, Tips und Hinweise zur Vorbereitung des Baukastens.

Was Du noch brauchst

Da die Modelle mit dem Computer gesteuert werden sollen, brauchst Du außer dem Baukasten noch einige Dinge:

- natürlich einen Computer (IBM-PC/XT-AT-kompatibler Rechner, Atari-ST, Amiga 500/2000) mit Maus,
- ein Interface (das ist eine elektronische Schaltung, die den Computer mit den fischertechnik-Elementen koppelt),
- die fischertechnik-Steuerungs-Software LUCKY LOGIC (der „Treibstoff“ für das Ganze).

Software und Interface sind nicht im Baukasten enthalten. Zusammen mit der Software erhältst Du auch eine ausführliche Bedienungsanleitung. Dort finden sich auch erste Übungen zur Steuerung des fischertechnik-Interface. Manches von den dort geschilderten Informationen wiederholen sich auch in dieser Anleitung; hier aber wesentlich knapper und nur auf das jeweilige Modell bezogen.

Steckermontage

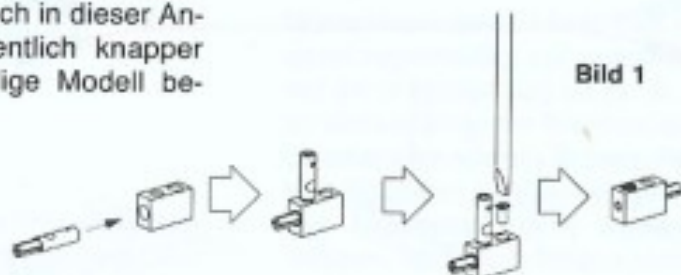
Alle Kabel werden mit Steckern versehen. Den Aufbau und die Montage der Stecker zeigt Dir das Bild 1.

Kabel konfektionieren

Für die Verbindung von Motoren, Lampen, Sensoren, Stromversorgung und Interface werden verschieden lange Kabel gebraucht. Dazu schneidest Du vom zweiadrigen Kabel im Baukasten folgende Kabellängen ab:

- 4 Stücke 20 cm lang
- 3 Stücke 25 cm lang
- 3 Stücke 30 cm lang
- 3 Stücke 35 cm lang
- 3 Stücke 40 cm lang
- 2 Stücke 45 cm lang

Der Rest des Kabels (ca. 90 cm) wird als Stromversorgungskabel vom Netzgerät zum Interface verwendet.



Abisolieren und Stecker montieren

An beiden Enden trennst Du die zwei Adern etwa 3 cm weit auf (**Bild 2**). Zum Abisolieren die Kunststoffhülle etwa 4 mm vom Kabelende mit einem Messer ringsherum einritzen. Die feinen Kupferadern dürfen dabei nicht verletzt werden! Anschließend die Kunststoffisolierung vom Kupferleiter abziehen. Vor dem Anschrauben des Steckers wird die Kupferlitze nach hinten umgebogen und dann der Stecker aufgeschraubt. Dazu löst Du die Schraube des Steckers, damit das Kabelende eingeschoben werden kann. Nun ziehst Du die Schraube sanft an, damit die Isolierung nicht zu stark gequetscht wird (**Bild 3**).

Bei einigen Schaltungen ist die Polung der Bauteile wichtig; es ist also nicht

egal, wie herum ein Bauteil angeschlossen wird. Damit es später beim Anschluß keine Verwechslung gibt, enthält der Baukasten rote und grüne Stecker, und wir legen jetzt gleich fest:

- Roter Stecker = (+)
- Grüner Stecker = (-)

An eine Kabelader kommen also an beide Enden Stecker mit der gleichen Farbe — an die rote Ader die roten Stecker und an die rotgrüne Ader die grünen Stecker (**Bild 4**).

Bei manchen Modellen werden längere Kabelstücke als die bereits vorbereiteten Kabellängen benötigt. Die erforderliche Kombination ist im Verkabelungsplan des Modells eingezeichnet. Um zwei Kabelstücke zu verbinden, mußt Du an einem Kabelende die Stecker durch Steckbuchsen austauschen. Damit beim



Bild 2

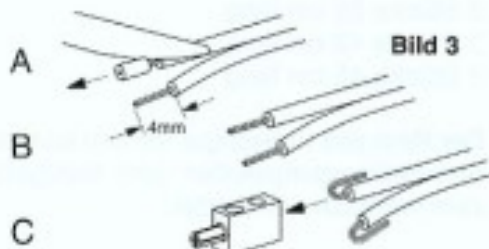


Bild 3

Befestigen des Kabels in der Steckbuchse das Kabelende nicht zu weit in die Buchse hineinrutscht, wird zuvor ein Stecker als Anschlag in die Buchse gesteckt (**Bild 5**).



Bild 4

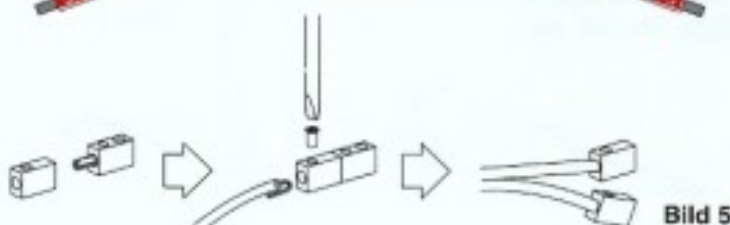


Bild 5

Papierdrahtband und Kabelklips

Mit dem Papierdrahtband können mehrere Kabel zu einem sogenannten „Kabelbaum“ zusammengefaßt werden (Bild 6). Du solltest es überall dort verwenden, wo Du Ordnung in die Verkabelung bringen mußt oder wo herabhängende Kabel den Funktionsablauf des Modells stören können, z. B. beim Plotter mit den langen, herabhängenden Kabeln.

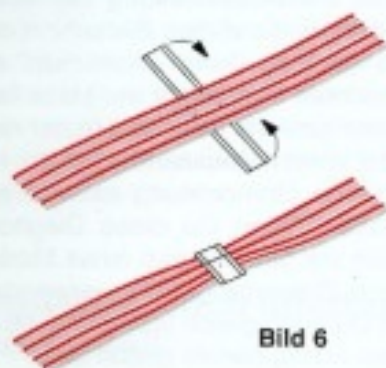


Bild 6

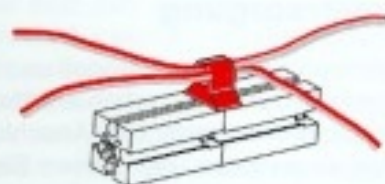


Bild 7

Um eines oder auch mehrere Kabel an einem Baustein oder auf der Experimentierplatte zu befestigen (damit sie z. B. nicht in ein Zahnrad geraten), verwendest Du den Kabelclips (Bild 7).

Netzgerät



Die Stromversorgung der Modelle erfolgt durch das fischertechnikNetzgerät Art. Nr. 30180. Es besitzt zwei Buchsenpaare

(also insgesamt vier Buchsen), die mit (+) und (–) beschriftet sind. Die beiden Plusbuchsen und die beiden Minusbuchsen sind im Netzteil miteinander verbunden, es spielt also keine Rolle, welche der beiden Buchsen Du verwendest. Der richtige Anschluß des Netzteils spielt eine wichtige Rolle, bei falscher Polung funktioniert das Modell nicht, es können sogar Bauteile beschädigt werden (Bild 8).

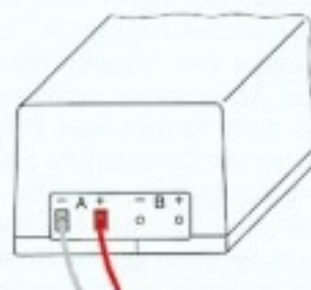
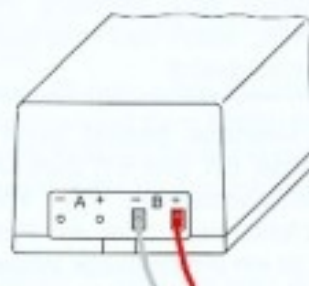


Bild 8



Wir weisen darauf hin, daß das Netzgerät regelmäßig auf mögliche Gefahren zu untersuchen ist (z. B. Schäden an der Leitung, am Stecker, am Gehäuse oder an anderen Teilen). Falls derartige Schäden festgestellt werden, darf das Netzgerät nicht weiter benutzt werden, bis der Schaden ordnungsgemäß behoben ist!

Interface

Das Interface koppelt die fischertechnik-Modelle und den Computer, denn die Anschlüsse des Computers können nicht genügend Strom liefern, um z.B. einen Motor zu versorgen. Zusätzlich schützt das Interface die Anschlüsse des Computers, wenn bei der Verdrahtung des Modells ein Fehler gemacht worden ist. Zusätzlich ist im Interface eine Zeitautomatik eingebaut, die den Strom der Motoren abschaltet, wenn vom Computer keine Steuerbefehle mehr kommen. Beim Testen der Steuersoftware wird man öfter Änderungen vornehmen. Damit man zum Ändern des Programms nicht immer die Stromzufuhr des Interface unterbrechen muß, sorgt die Automatik für den Stillstand des Modells. Wenn der Computer dann wieder Befehle sendet, geht es ganz normal weiter. Eines sollte man jedoch immer beachten:

WICHTIG:

Computer und Interface immer ausschalten, wenn das Interface an- oder abgesteckt wird.

Das Interface besitzt auch zwei Anschlüsse, EX und EY, zum Messen von Widerstandswerten. Hier können z. B. der Lichtsensor oder ein Potentiometer (das ist ein einstellbarer Widerstand) angeschlossen werden.

Das Wie und Wo beim Interface-Anschluß ist in der InterfaceAnleitung speziell für Deinen Computer beschrieben.

Steuersoftware

Zur Steuerung der Modelle gibt es eine spezielle Steuersoftware, LUCKY LOGIC, die sich sehr einfach bedienen läßt. Man kann die Bilder der einzelnen Taster, Lampen und Motoren auf dem Bild-

schirm platzieren und mit der Maus Verbindungen ziehen. Wenn das Steuerprogramm abläuft, wird über das Interface das Modell vom Computer gesteuert. Eine genaue Beschreibung der Steuerbefehle liegt der Software-Diskette bei.

Die fischertechnik-Steuersoftware ermöglicht auch, das Interface mit dem Verbindungskabel zu testen. Dazu werden Motoren oder Lampen an die Ausgänge (M1 bis M4) und Taster an die Eingänge (E1 bis E8) angeschlossen. Im Menü der Software wird „Interfacetest“ gewählt. Die Schaltstellung der einzelnen Taster wird auf dem Bildschirm dauernd angezeigt. Durch „Anklicken“ des gewünschten Motors mit der Maus kann der Motor getestet werden. Mit der rechten und linken Maustaste wird die entsprechende Drehrichtung ein- und ausgeschaltet. Wenn Du diese Diagnosefunktion vor dem Betrieb eines Modells mit dem Steuerprogramm verwendest, kannst Du auch gleich feststellen, ob die Motoren richtig herum gepolt sind.

Stromversorgung

Die Stromversorgung von Modell und Interface erfolgt durch das schon erwähnte fischertechnik-Netzgerät. Das Anschlußkabel hat einen roten und grünen Stecker. Am Interface gibt es zwei Buchsenpaare, die mit (+) und (-) bezeichnet sind. Die beiden (+)-Buchsen und die beiden (-)-Buchsen sind im Interface parallelgeschaltet. Es ist daher gleichgültig, welche der beiden Buchsen verwendet werden. Der richtige Anschluß des Netzteils an das Interface spielt eine wichtige Rolle, bei falscher Polung funktioniert es nicht:

- Roter Stecker in eine (+)-Buchse
- Grüner Stecker in eine (-)-Buchse

Verbindung vom Computer zum Interface

Zum Anschluß des Interfaces an die Druckerschnittstelle der Computer gibt es einen Adapter. Er besteht aus einer kleinen Leiterplatte mit zwei Steckverbindern. Der eine Steckverbinder paßt an den Computer, der andere Steckverbinder besteht aus zwei Reihen zu je 10 Stiften in einem kleinen Gehäuse. Auf diesen Steckverbinder wird der Stecker des grauen Verbindungskabels gesteckt. Der Stecker hat auf der Oberseite in der Mitte eine kleine Nase, die genau in die Aussparung des Steckverbinders auf der Leiterplatte paßt. Auf diese Weise wird verhindert, daß der Stecker falsch herum eingesteckt wird.

Verbindung vom Interface zum Modell

Nun muß das Interface mit dem Modell verbunden werden. Im Baukasten befindet sich dazu ein 20poliges, farbiges Flachbandkabel. An einem Ende des Kabels ist ein 20poliger Stecker angebracht, der am Interface eingesteckt wird. Auch dieser Stecker hat wieder eine Nase, die in die Aussparung am Interface-Gehäuse paßt. So kann auch hier der Stecker nicht verkehrt herum eingesteckt werden. Zur Sicherheit sind die Kabelfarben auf dem Gehäuse noch einmal aufgedruckt.

Doch zunächst muß das andere Ende des Kabels für den Anschluß der Modelle hergerichtet werden. Das geht fast genauso, wie Du es schon beim Montieren der Stecker gemacht hast.

Im Baukasten liegt eine 28polige Buchsenleiste, in die Du die fischertechnik-Stecker einstecken kannst. Zuerst wird

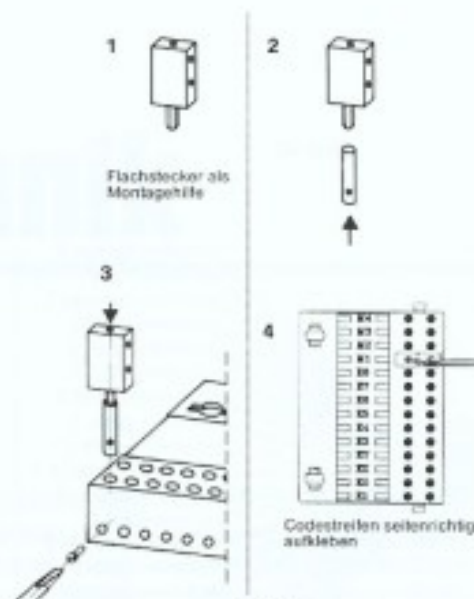


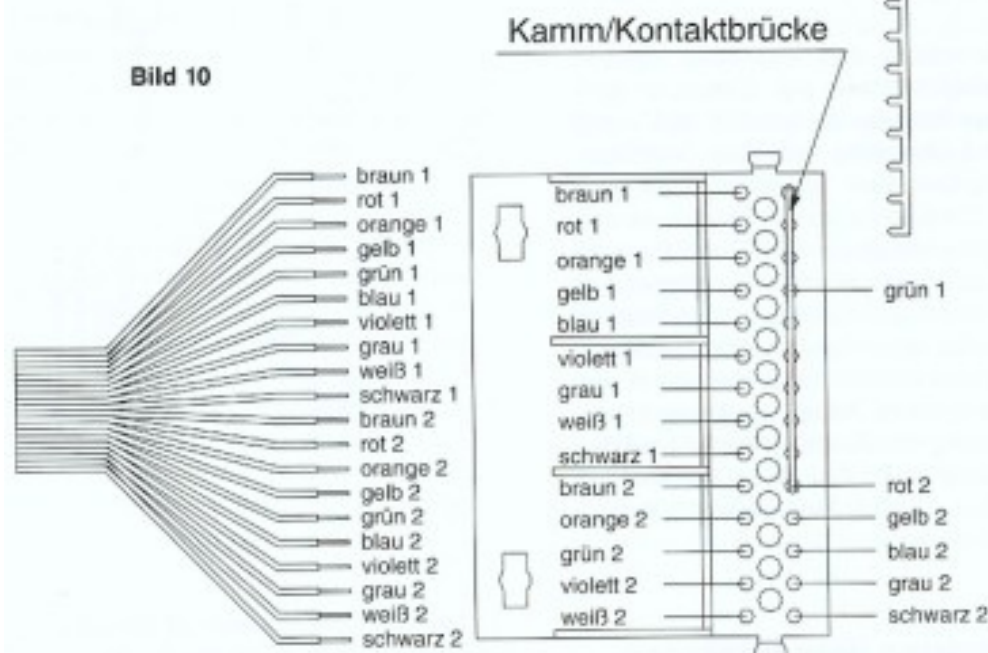
Bild 9

die Steckerleiste mit allen 28 Einzelbuchsen wie in **Bild 9** montiert. Im **Bild 9** siehst Du auch, wie ein Stecker als Montagehilfe verwendet wird. Vor dem Anschrauben des Kabels klebst Du noch den Codestreifen auf die Oberseite der Buchsenleiste.

Schraube die 20 Adern des Flachbandkabels zusammen mit dem Kamm an die Buchsenleiste. **Bild 10** (Seite 14) gibt genau an, welche Ader an welche Buchse kommt:

1. Kabelende ca. 3 bis 4 cm weit in Einzeladern auftrennen
2. Einzeladern ca. 4 mm weit vorsichtig abisolieren
3. Die blanken Kabelenden auf die Isolation umbiegen
4. Adern in die richtige Buchse der Buchsenleiste einstecken und die Schraube sanft festziehen
5. Auf der einen Seite wird gleichzeitig ein Metallkamm (Kontaktbrücke) befestigt, der mehrere Buchsen untereinander verbindet.

Bild 10



Laß Dir Zeit beim Anschließen der vielen Kabeladern und achte genau auf den richtigen Anschluß. (Hastig Zusammenbauen und nacher nach dem Fehler suchen dauert bestimmt länger.)

Nach Abschluß der Arbeiten wird zunächst noch einmal eine Sichtkontrolle durchgeführt. Kontrolliere genau, ob auch alle Kabel richtig angeschlossen sind (auch Superprofis werden manchmal abgelenkt und machen dann einen Fehler). Du sparst Dir so späteren Ärger oder gar eine Beschädigung des Inter-

face. Wenn erst einmal das Modell angeschlossen ist, wird die Fehlersuche wesentlich schwieriger. Erst wenn die Beschaltung wirklich stimmt, wird das Kabel an das Interface angeschlossen. Danach wird die Zugentlastung für das Kabel an der Buchsenleiste angebracht, wie Du es in **Bild 11** siehst.



Bild 11

Zugentlastung

Ein wenig Elektrotechnik

Der Strom fließt im Kreis

Nicht alle Stoffe leiten den Strom gleich gut. Besonders gut fließt er durch Metalle, wie z.B. die dünnen Kupferleitungen, die beim fischertechnik-Baukasten verwendet werden. Aber auch Messing, Eisen, Blei, Zinn oder die fischertechnik-Metall-Achsen sind gute Leiter. Berühren sich zwei Leiter, kann der Strom auch über die Kontaktstelle fließen (wir nutzen das z. B. bei Steckern und Buchsen aus). Andere Stoffe leiten den Strom schlecht oder gar nicht. Darum werden auch die Kupferadern der Kabel gegen zufällige Berührung durch Kunststoff geschützt, denn der Kunststoff ist ein ausgesprochener Nichtleiter oder Isolator. Auch Luft, Glas, trockenes Holz und die meisten nichtmetallischen Stoffe sind Nichtleiter.

Für den Betrieb von elektrischen Verbrauchern (Lampen, Elektromagnete, Motoren) braucht man eine Stromquelle, z. B. eine Batterie oder ein Netzteil. Die Stromquelle kann man sich wie eine Wasserpumpe vorstellen, die den Strom durch die Leitungen und Verbraucher drückt. Wie bei der Pumpe vom Aquarium ist ein geschlossener Kreislauf nötig, damit der Strom fließen kann. Wird der Stromkreis an irgendeiner Stelle unter-

brochen, kann kein Strom mehr fließen. So wie die Pumpe je nach Leistungsfähigkeit einen bestimmten Wasserdruck erzeugen kann, liefern die verschiedenen Stromquellen eine bestimmte Spannung (die in Volt gemessen wird). Das Netzteil liefert z. B. 8 Volt, aus der Steckdose kommen 220 Volt und für spezielle Zwecke sogar 380 Volt. Ab einer bestimmten Höhe (etwa 50 - 60 Volt) wird die Spannung für den Menschen lebensgefährlich. Die Bausteine aus dem fischertechnik-Baukasten arbeiten mit einer ungefährlichen Spannung von 6 - 9 Volt; sie dürfen auch nicht mit einer höheren Spannung betrieben werden, da sie sonst zerstört würden. Jetzt ist auch klar, wozu das Netzteil dient: Es macht aus den 220 Volt der Steckdose die passende und ungefährliche Spannung für die fischertechnik-Modelle.

Jeder Verbraucher benötigt einen bestimmten elektrischen Strom — so wie durch eine Leitung Wasser strömt. Und genauso, wie ein Wasserhahn dem Wasser einen Widerstand entgegensetzt, bildet auch der Verbraucher einen Widerstand für den elektrischen Strom. Je kleiner der Widerstand des Verbrauchers ist, desto größer wird der Strom und umgekehrt. Die Stärke des Stroms wird in der Elektrotechnik in „Ampere“ (abge-

kürzt „A“) angegeben, z. B. findest Du auf den Sicherungsautomaten in Deiner Wohnung meist die Angabe „10 A“ oder „16 A“.

Wenn eine Schaltung einmal nicht funktioniert, solltest Du erst alle Kontaktstellen überprüfen, z.B. locker sitzende Stecker oder lose Kabelanschlüsse in den Steckern. Auch wenn oft zu fest an einem Kabel gezogen wird, kann die Kupferseele brechen oder reißen. Das führt dann zu einem heimtückischen „Wackelkontakt“, bei dem es nur manchmal zu Störungen kommt — die sich aber umso schwerer finden lassen.

Elektrotechnik „nach Fahrplan“

So wie man einen Stadtplan braucht, um sich in einer fremden Stadt zurechtzufinden, so geht es auch bei der Elektronik nicht ohne „Schaltplan“. Und so, wie man beim Stadtplan kein Foto aus der Vogelschau mit Häusern und Bäumen zeigt (was den Plan nur unübersichtlich machen würde), zeigt man beim Schaltplan keine realen Abbildungen von Kabeln, Lampen, Tastern und Motoren. Die einzelnen Bauteile werden vielmehr durch einfache Symbole dargestellt und die Verbindungskabel durch Linien zwischen den Bauteilen. Damit man erkennt, ob zwei Kabel elektrisch verbunden sind, setzt man einen kräftigen Punkt auf die Verbindungen. Bei einer Linienkreuzung ohne Punkt gibt es auch keine elektrische Verbindung.

Ganz so weit wie die Ingenieure müssen wir die Vereinfachung des Schaltplans aber nicht treiben. Bei den fischertechnik-Schaltungen wirst Du immer noch auf einen Blick erkennen können, um welches Bauteil es sich handelt.

Elektrische Bauteile

1. Taster

Die Taster besitzen drei Anschlüsse: der mittlere Anschluß (1) ist beweglich und bildet im nicht betätigten Zustand eine Verbindung mit dem unteren Anschluß (2), dem Ruhekontakt. Wird der Taster gedrückt, springt der Kontakt um, und es ergibt sich eine Verbindung zwischen dem Mittenanschluß (1) und dem oberen Anschluß (3), dem Arbeitskontakt. Wird der Taster durch eine Impulsscheibe betätigt, schließt er bei jeder Umdrehung fünfmal (Bild 12).

Ein Taster wird immer mit einem E-Anschluß des Interface verbunden.

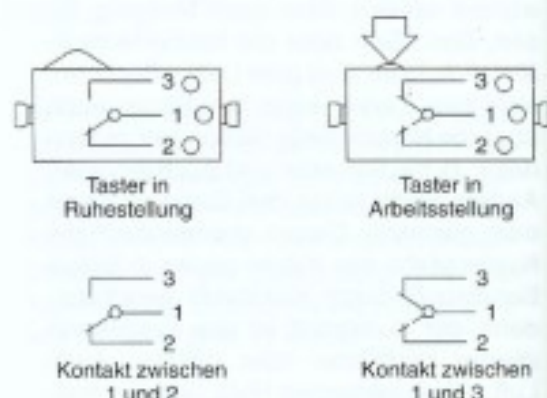
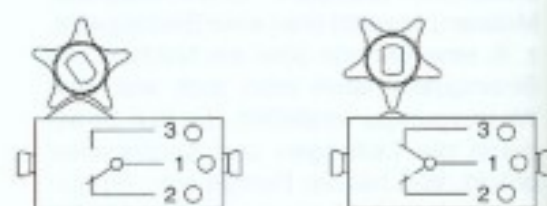


Bild 12



Taster mit Impulsscheibe

2. Lampen

Der Leuchtstein besteht aus einem Kunststoffsockel mit zwei Röhrchen, zwischen denen die Lampe eingesteckt wird

Bild 13



Kugellampe



Linsenlampe



Fototransistor



ben ziehen, sondern Du hebelst die Lampe mit dem Schraubendreher aus dem Leuchtstein.



(Bild 13). Du kannst die Stromversorgung also auf einer Seite oder auf beiden Seiten anschließen (Bild 14). Wenn aber beide Stecker vom Netzteil an das gleiche Röhrchen angeschlossen werden, gibt es einen Kurzschluß! Da die Röhrchen wie Buchsenverbinder funktionieren, kannst Du beim Anschluß von zwei Lampen Kabel sparen, wenn die Verbindung wie im Bild 14 unten erfolgt.

Achte bei den Modellen darauf, welche Lampe verwendet wird. So solltest Du die Linsenlampe nicht mit einer farbigen Leuchtkappe abdecken, denn durch die Wärmentwicklung kann sich nach einiger Betriebszeit die Leuchtkappe verformen (Brennglaseneffekt). Die Linsenlampe wird hauptsächlich zusammen mit dem Lichtsensor für Lichtschranken verwendet.

Zum Entfernen einer Lampe aus dem Leuchtstein darfst Du nicht am Glaskol-

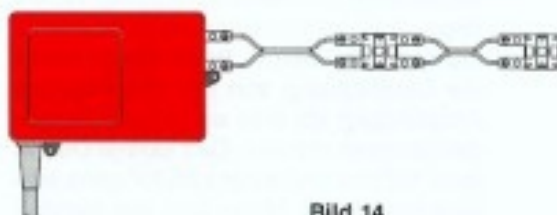
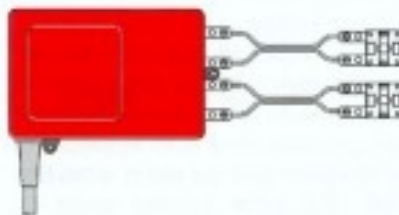
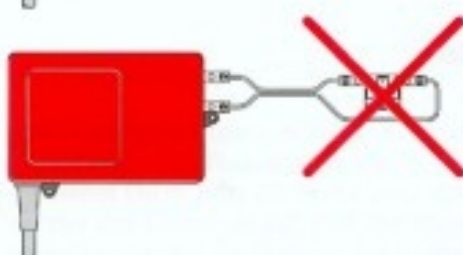
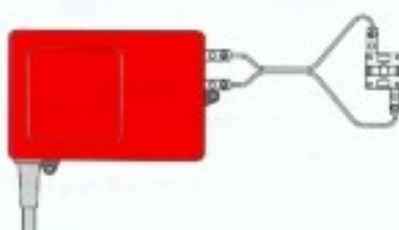


Bild 14

Eine Lampe wird entweder direkt mit der Stromversorgung oder einem M-Anschluß des Interface verbunden.

3. Motor

Gleichspannungs-Elektromotor mit drei Anschlußmöglichkeiten (Bild 15). Die Anschlußmöglichkeiten kannst Du auch kombinieren (z. B. rechts den Stecker von vorne und links von oben einstecken). Die beiden Anschlußstecker dürfen aber NIE auf der gleichen Seite des Motors eingesteckt werden, denn das gibt einen Kurzschluß.

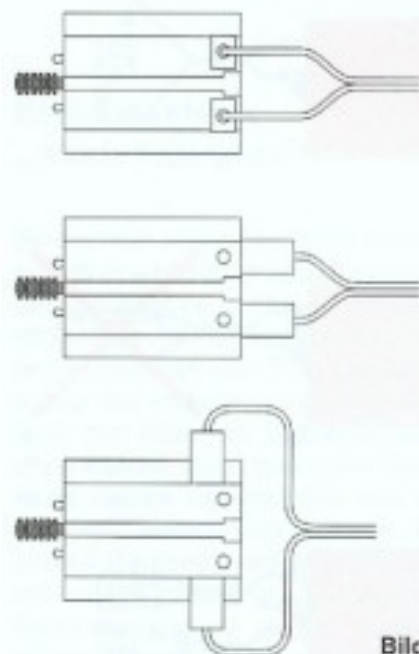


Bild 15

Bei den Motoren gibt es noch eine Besonderheit. Bei einer Lampe spielt es keine Rolle, „wie herum“ der Anschluß erfolgt; bei geschlossenem Stromkreis leuchtet die Lampe. Beim Motor hängt die Drehrichtung von der Polarität des Anschlusses ab, also wo (+) und (-) angeschlossen werden. Das kannst Du mit dem Netzteil und einem Motor ganz einfach testen. Der Motor wird ans Netzteil

angeschlossen. Er dreht dann in eine Richtung. Wenn Du die Anschlüsse am Motor vertauschst, dreht er in die entgegengesetzte Richtung. Wenn im Modell ein Motor in die falsche Richtung dreht, brauchst Du nur die Anschlußstecker zu vertauschen.

Ein Motor wird immer mit einem M-Anschluß des Interface verbunden.

4. Lichtsensor (Fototransistor)

Der Fototransistor leitet den Strom umso besser, je heller es ist.

Der Fototransistor ist das einzige Bauelement, bei dem Du auf die richtige Polung achten mußt.

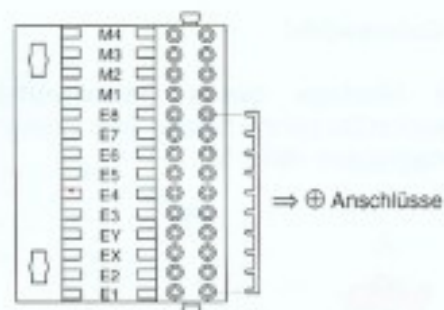
Er liegt dem Baukasten komplett montiert bei. Die (+)-Seite ist rot markiert. Beim Anschluß an die 28polige Buchse mußt Du unbedingt darauf achten, daß die (+)-Seite des Fototransistors an einer der Buchsen eingesteckt wird (Bild 16), die durch den langen Metallkamm miteinander verbunden sind. NIE an die M-Anschlüsse oder direkt ans Netzteil anschließen! Maximale Spannung 12 Volt. Der Lichtsensor wird immer mit einem E-Eingang des Interface verbunden. Er wird durch eine Lampe beleuchtet. Wenn der Lichtweg zwischen Lampe und Fototransistor unterbrochen wird, wirkt das für den Computer genauso, als wenn ein Taster mit Ruhekontakt geöffnet wurde — also wie eine Unterbrechung. Die Anordnung aus Fototransistor und Lampe, wie sie in Bild 16 unten gezeigt wird, nennt der Fachmann „Lichtschranke“.

Merkregel:

Wenn Licht auf den Fototransistor fällt, ist der Eingang des Interface auf „1“ geschaltet. Wird der Lichtstrahl unterbrochen, schaltet der Eingang auf „0“.



Anschluß Fototransistor



Anschluß Lichtschranke

Bild 16

Alle elektrischen Bauteile dürfen nur mit den fischertechnik-Stromversorgungen (Netzteil, Batterien) betrieben werden, damit sie nicht durch eine zu hohe Betriebsspannung zerstört werden.

Montage der mechanischen Bauteile

Zwei mechanische Bausteine, der Federgelenkstein und der Gelenkwürfel, müssen erst zusammengebaut werden, bevor Du sie verwenden kannst. Dann folgen noch einige Tips zum Zusammen-

bau von Bausteinen, die Du bei verschiedenen Modellen brauchst.

1. Federgelenkstein

Der Federgelenkstein besteht aus fünf Einzelteilen. Die Reihenfolge der Montage siehst Du in **Bild 17**. Achte beim Einsetzen der Feder darauf, daß die abgewinkelten Enden der Feder in die kleinen Bohrungen der Gelenksteinhälften eingesteckt sind. Der zusätzliche Baustein mit dem Verbindungsstück dient nur als Montagehilfe. Zum Schluß mit zwei Münzen verschrauben — aber nicht zu fest, das Gelenk muß sich noch gut bewegen lassen.

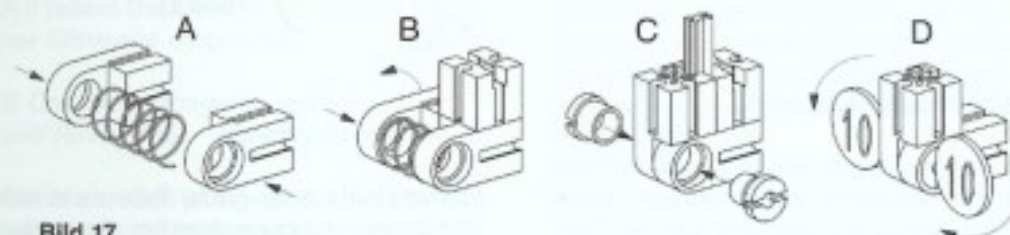


Bild 17

2. Gelenkwürfel

Zur Montage dieses Gelenkwürfels brauchst Du nur die beiden Teile ineinanderzustecken (Bild 18).

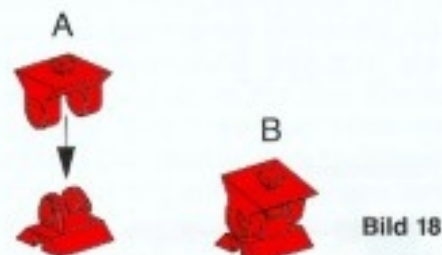


Bild 18

3. Flachnabe

Nabenzange und Nabenummer mit dem Zahnrad verschrauben. Dann auf die Achse stecken und Nabenummer gut festziehen (Bild 19).

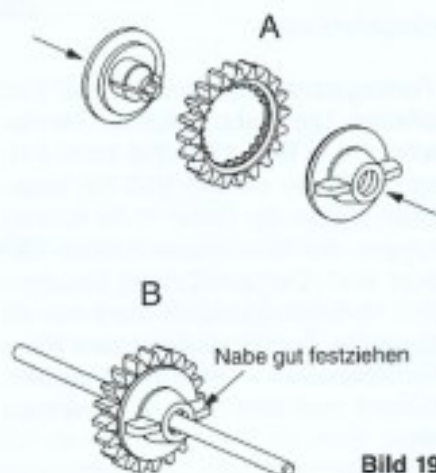


Bild 19

4. Klemmbare Schnecke

Die Zangenmutter einige Umdrehungen auf die Schnecke aufschrauben, dann auf die Achse stecken und nun die Zan-

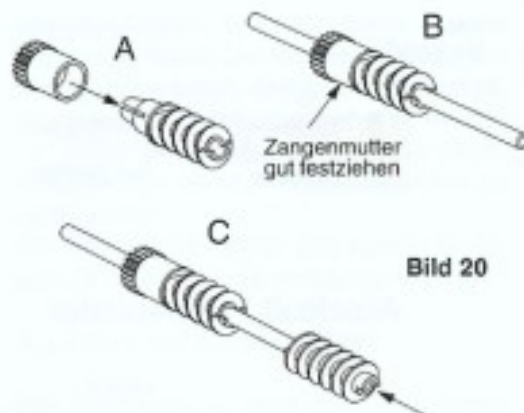


Bild 20

genmutter gut festziehen. Bei einigen Modellen wird die klemmbare Schnecke durch weitere Schneckenteile verlängert (Bild 20).

5. Kette und Raupenbeläge

Die Kette montierst Du durch Zusammenstecken der einzelnen Rastkettenlieder, bis die erforderliche Länge erreicht ist (in der Anleitung zu den Modellen steht immer, wie lang die Kette sein muß). Durch Aufstecken der Rast-raupenbeläge kann eine Raupenkette (oder ein Förderband) gebaut werden. Die Beläge werden durch die seitlichen Vierkanthasen festgehalten (Bild 21).

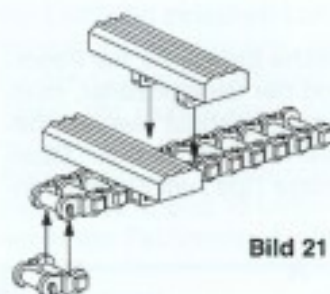


Bild 21

6. Roboter-Greifzange

Damit kleine oder glatte Teile nicht aus der Zange rutschen, werden die beiden

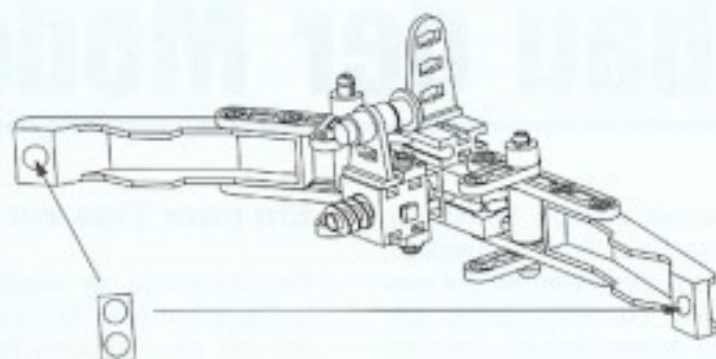


Bild 22



60°



30°



15°



7,5°

Bild 23

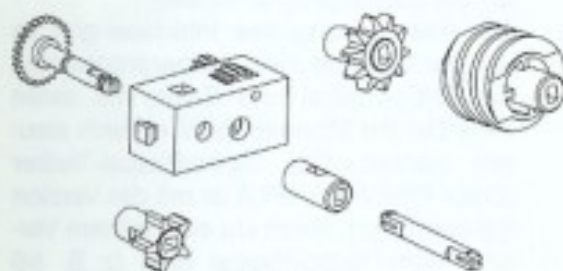


Bild 24

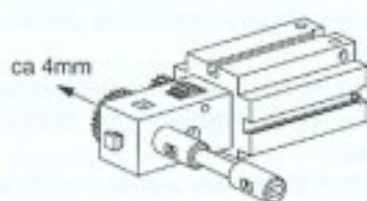


Bild 25

Haftpunkte auf die Flächen der Greifzange geklebt (**Bild 22**).

7. Winkelsteine

Im Baukasten gibt es vier verschiedene Winkelsteine, die sich in den Baustufenzeichnungen nicht immer ganz leicht unterscheiden lassen. In **Bild 23** kannst Du sie durch ihre Lage gut unterscheiden. Auf jedem Baustein ist der Winkel auf einer Stirnseite eingepreßt.

8. Untersetzungsgetriebe 65:1 und Rastachse

Ohne das Getriebe würden die Modelle viel zu schnell laufen, und der Motor hät-

te auch nicht genügend Kraft. Auf dem **Bild 24** siehst Du, wie die Rastachse mit den anderen Rastelementen kombiniert werden kann.

9. Ausprobieren der Mechanik von Hand

Oft ist es sinnvoll, nach dem Zusammenbau der Modelle die mechanische Arbeitsweise der Modelle zuerst einmal mit der Hand auszuprobieren (z. B. die Roboterzange zu öffnen oder zu schließen). Dazu ziehst Du einfach das Zahnrad mit Rastachse etwa 4 mm aus dem Getriebe heraus (**Bild 25**).

Aufbau der Modelle

Die Modelle selbst werden Schritt für Schritt nach den Bauplänen aufgebaut. Bei jedem Bauabschnitt werden die neu hinzukommenden Bauteile farbig gezeigt. Wenn auf einem vorhergehenden aufgebaut wird, ist dieser Abschnitt weiß gezeichnet. Suche vor jedem Bauabschnitt zuerst die benötigten Bauteile heraus und baue diese anschließend ein. Der nächste Bauabschnitt wird erst in Angriff genommen, wenn alle Teile verarbeitet sind. Achte bei den Bausteinen auch auf die Einbaulage, damit Dir in späteren Abschnitten nicht der Weg verbaut ist.

Wichtig ist auch, daß die Achsverschraubungen (z.B. Naben) gut festgezogen werden.

Bei allen Modellen werden auch elektrische Bauteile verwendet: Taster, Motoren oder Lichtsensoren (Fototransistoren). Die Verbindung zum Computer-Interface wird immer über die 28polige Buchsenleiste hergestellt. In der Bauanleitung wird Dir gezeigt, wie die Bauteile an das Interface angeschlossen werden und welche Kabellängen Du dazu brauchst.

Anders als bei der Steuerung eines Modells von Hand, bei der Du sehen kannst, was das Modell macht und entsprechend reagieren kannst, ist der Computer „blind“. Da der Computer (genauer, das Steuerprogramm) Informationen über das Verhalten des Modells braucht (z. B. die Stellung eines Schiebers), werden in vielen Modellen Taster oder Lichtsensoren eingesetzt. Über diese „Rückmeldung“ kann das Programm den Zustand des Modells erkennen und entsprechend reagieren.

Ein paar Tips zur Software

Die Steuerung der Modelle per Computer kann auf zwei Arten erfolgen, entweder mit der fischertechnik-Steuerungssoftware LUCKY LOGIC oder direkt mit einer Programmiersprache wie Pascal oder Basic. Beide Möglichkeiten findest Du auch in diesem Handbuch. Die Programmierung mit LUCKY LOGIC wird im Handbuch dieser Software genau erklärt, daher richten sich diese Tips hier an die „Selbstprogrammierer“.

Zur Ansteuerung des Interface gibt es auf der Diskette zwei Treiberprogramme für Turbo-Pascal und Basic, mit deren Hilfe Du die Modelle recht einfach steuern kannst. Der Turbo-Pascal-Treiber (Datei FISCHER.TPU) ist mit der Version 6.0 compiliert, wenn Du eine frühere Version vom Turbo-Pascal hast (z. B. 5.0 oder 5.5), mußt Du das Quellprogramm FISCHER.PAS einfach noch einmal übersetzen.

Beim allerersten Programmstart braucht das Programm zwei Informationen über den Computer:

1. an welcher Druckerschnittstelle das Interface angeschlossen ist. Diese Frage mußt Du beantworten.
2. den Maximalwert, den der Analogeingang liefern kann. Das macht das Programm automatisch.

Diese beiden Informationen werden in der Datei FISCHER.DAT gespeichert und können dann von den Programmen immer abgerufen werden. Diese Datei wird übrigens auch von LUCKY LOGIC verwendet. Wenn Du das Interface auf

eine andere Schnittstelle steckst oder den Computer wechselst, löscht Du einfach die Datei FISCHER.DAT. Die Software fragt dann nach den neuen Werten.

Das folgende Programmlisting zeigt dieses Quellprogramm und aus den Kommentaren darin solltest Du alles Wissenswerte entnehmen können.

```
Unit FISCHER: { Turbo Pascal 5.0 - 6.0, Datei: FISCHER.PAS }
{.....}
{ Diese Unit ermöglicht den Zugriff auf das Fischertechnik-Interface aus }
{ Turbo-Pascal-Programmen heraus. Benötigt wird die Datei FISHPAS.OBJ, die }
{ die Maschinenprogrammroutinen zur ANsteuerung des Interface enthält }
{.....}

INTERFACE

Uses CRT;

CONST Install_Name :String[12]='fischer.dat'; { Datei mit Installationsdaten: Drucker- }
                                              { Port und Maximalwert für Analogeingänge }
                                              { Gleiche Werte, wie bei LUCKY LOGIC }

{ Typdefinitionen für die Ansteuerung des Interface }

TYPE Ausgang_Typ      =(Bit aus, Bit ein);      { Bei Lampe am Ausgang: Ein/Aus }
    Motor_Richtung_Typ =(ein, rechts, links, aus); { Bei Motor am Ausgang: Ein/Aus/Richtung }

    Poti_Typ           =(X_Poti,Y_Poti);         { Analogeingänge EX und EY }

    Digit_Array_Typ     =Array [1..16] of Boolean; { Eingangsdaten vom Interface als Feld }

    Digit_Rec_Typ       =Record                  { Eingangsdaten vom Interface als Record }
        E1, E2, E3, E4,                          { Die Taster lassen sich so per Namen an- }
        E5, E6, E7, E8,                          { sprechen }
        E9, E10, E11, E12,
        E13, E14, E15, E16: Boolean;
    End;

    Install_Feld_Typ    =Record                  { Installationsdaten (siehe oben) }
        Printadr: Word;
        PC_Wert : BYTE;
    End;

VAR Digit_Array :Digit_Array_Typ;              { Eingangsdaten als Feld und als Record }
    Digit_Rec   :Digit_Rec_Typ absolute Digit_Array; { belegen den gleichen Speicherplatz }
                                              { und können nach Bedarf verwendet werden }

    Install_Feld :Install_Feld_Typ;             { Installationsdaten }
    Install_File :File of Install_Feld_Typ;     { Datei mit Installationsdaten }

    Gutspei      :Byte;                         { Variablen für die Programminstallation }
    Printadr     :Word;
    PC_Wert      :Byte;
    Maus         :Boolean;

{.....}

PROCEDURE Motoren aus;
{ Schaltet alle Ausgänge aus }
{ Wird bei Initialisierung der Unit aufgerufen }

PROCEDURE Motor(Motor_Nr:Byte; Motor_Richtung:Motor_Richtung_Typ);
{ Steuerung eines Motors }
{ Motor_Nr: Nummer des anzusteuernenden Motors }
{ 1. Interf. 1 bis 4, 2. Interf. 5 bis 8 }
{ Motor_Richtung: Drehrichtung des Motors oder Stop }
```

```

FUNCTION Analog_Eingang(Podi:Poti Typ):Word;
{ Lesen eines Analogwertes an einem der Eingänge EX }
{ oder EY. Der Meßwert wird als WORD-Variable zu- }
{ rückgegeben. }
{ Podi: Analog-Eingang (X_Poti oder Y_Poti) }

PROCEDURE Eingang_Bitmuster(Var Digit_Array);
{ Lesen aller Eingangswerte (Taster, Lichtschranke, }
{ etc.). Als Parameter kann nach Belieben ein Feld }
{ (Digit_Array Typ) oder ein Record (Digit_Rec_Typ) }
{ verwendet werden. }
{ 1. Interf. 1 bis 8, 2. Interf. 9 bis 16 }

FUNCTION Eingang_Byte:Byte;
{ Lesen aller Eingangswerte des 1. Interface auf }
{ ein Byte - also in gepackter Form }

FUNCTION Eingang(Taster_Nummer:Byte):Boolean;
{ Lesen eines bestimmten Eingangs, der durch den }
{ Parameter Taster_Nummer festgelegt ist }

FUNCTION Taster(TasterNr : BYTE) : BOOLEAN;
{ Lesen eines bestimmten Eingangs, der durch den }
{ Parameter TasterNr festgelegt ist. In Pascal for- }
{ muliert - siehe unten }

PROCEDURE Ausgang(Motor_Nr:Byte; Wert:Ausgang_Typ);
{ Schalten eines Ausganges Ein/Aus }
{ Motor_Nr: Nummer des anzusteuernenden Ausganges }
{ 1. Interf. 1 bis 4, 2. Interf. 5 bis 8 }

FUNCTION Ermittle_PCWert(Podi:Poti Typ):Byte;
{ Interne Funktion: Kennwert des Analoginterface }
{ für den jeweiligen Computer ermitteln. }
{ (Abhängig von der Geschwindigkeit des Computers) }

PROCEDURE Interface_Installieren;
{ Parameterdatei erstellen }
{ Druckerschnittstelle abfragen, PC-Wert ermitteln }
{ (siehe unten) }

{.....}

IMPLEMENTATION

Var Taste:Char;

{ Die folgenden Grund-Prozeduren und -Funktionen befinden }
{ sich in der Datei FISCHPAS.OBJ }
{$L FISCHPAS }
Procedure Motoren_aus; external;
Procedure Motor; external;
Function Analog_Eingang; external;
Procedure Eingang_Bitmuster; external;
Function Eingang_Byte; external;
Function Eingang; external;
Procedure Ausgang; external;
Function Ermittle_PCWert; external;

{.....}

FUNCTION Taster(TasterNr : BYTE) : BOOLEAN;
{ Lesen eines bestimmten Eingangs, der durch den }
{ Parameter TasterNr festgelegt ist. }
BEGIN
    Eingang_Bitmuster(DIGIT_ARRAY);
    Taster := Digit_ARRAY[TasterNr];
END;

{.....}

```



```

Procedure Interface_Installieren;
  { Interface-Kennwerte abfragen und in die Datei FISCHER.DAT schreiben }
Var
  Drucker :Array [1..4] of Word absolute $40:8;           { Drucker-Schnittstellen- }
                                                         { Adressen im BIOS-Daten- }
                                                         { Bereich }
  Anzdru  :Byte;                                           { Anzahl Schnittstellen }
  zaehler :Word;                                           { Hilfsvariablen }
  Taste   :Byte;
  Zeichen :Char;

Begin
  If Drucker[2]=0 then Printadr:=Drucker[1]               { Anzahl Druckerschnittstellen }
  else Begin
    If Drucker[3]=0 then Anzdru:=2 else if Drucker[4]=0 then Anzdru:=3 else Anzdru:=4;
    Writeln('Welchen Druckerport wollen sie benutzen?');
    Writeln;
    Write(' 1 , ');
    For zaehler:=2 to Anzdru-1 do Write(zaehler:1, ' , ');
    Writeln(Anzdru:1, ' ');
    Repeat
      Taste:=ord(Readkey);                                { gewünschte Schnittstelle erfragen }
      If (Taste<49) or (Taste>52) then Taste:=9 else Taste:=Taste-48;
    Until Taste<=Anzdru;
    Printadr:=Drucker[Taste];                             { Schnittstellenadresse }

    Writeln;
    Writeln;
    Writeln;
  End;
  Zeichen:=#32;
  Writeln('Bitte Interface anschließen!');
  Writeln('Wenn bereit eine Taste drücken. ');
  Repeat Until Keypressed;
  PC_Wert:=Ermittle_PCWert(Y_Poti);                      { Init-Wert für Analogeingänge }
  Install_Feld.PC_Wert:=PC_Wert;
  Install_Feld.Printadr:=Printadr;
  Assign(Install_File,Install_name);                     { Datei mit Init-Werten erzeugen }
  ($I-)
  Rewrite(Install_File);
  ($I+)
  If IOResult=0 then Begin
    Write(Install_File,Install_Feld);                     { Werte schreiben }
    Close(Install_File);
  End
  else Writeln('Kann Datenfile nicht erstellen!');
End;

[.....]

BEGIN { Hp }
  { Initialisierungsteil der Unit FISCHER }
  Assign(Install_File,Install_name);                     { Init-Werte aus Datei FISCHER.DAT lesen }
  ($I-) Reset(Install_File); ($I+)
  If IOResult<<0
  THEN
    BEGIN
      TextColor(WHITE);                                   { Date nicht vorhanden - dann }
      TextBackGround(BLACK);                             { Werte erfragen }
      CLRSCR;
      Writeln('Konnte Interfacetreiber ',Install_Name,' nicht laden!');
      Writeln('Interface neu installieren ? (J/N)');
      Repeat
        Taste:=Ucase(Readkey);
        Until (Taste='J') or (Taste='N');
        If Taste='J' then Interface_Installieren
        else Halt;
      End
    End
  END

```

```

ELSE
  BEGIN
    Read(Install_File,Install_Feld);
    Close(Install_File);
    Printadr:=Install_Feld.Printadr;
    PC_Wert:=Install_Feld.PC_Wert;
    ClrScr;
  END;
  Maus:=False;
  Motoren aus;
END. { Hp }
{ alle Motoren aus }

```

Beim Abfragen der Taster genügt es oft nicht, nur abzufragen, ob ein Taster gedrückt ist — sondern das Programm muß warten, bis der Taster sich wieder in Ruhestellung befindet (z. B. bei Impulsscheiben). Es muß also zuerst gewartet werden, bis der Taster gedrückt ist:

```

REPEAT UNTIL Taster(1);
      [Beispiel Taster an E1]

```

Dann muß das Programm warten, bis die Impulsscheibe den Taster wieder freigegeben hat:

```

REPEAT UNTIL NOT Taster(1);

```

Keine Angst beim Experimentieren mit eigenen Programmen — sobald das Programm gestoppt wird, hält das Interface auch die Motoren an. Du kannst natürlich auch eine „NOT-AUS“-Funktion im Pro-

gramm einbauen, wenn Du an verschiedenen Stellen (vor allem innerhalb von Schleifen) die folgende Zeile einbaust:

```

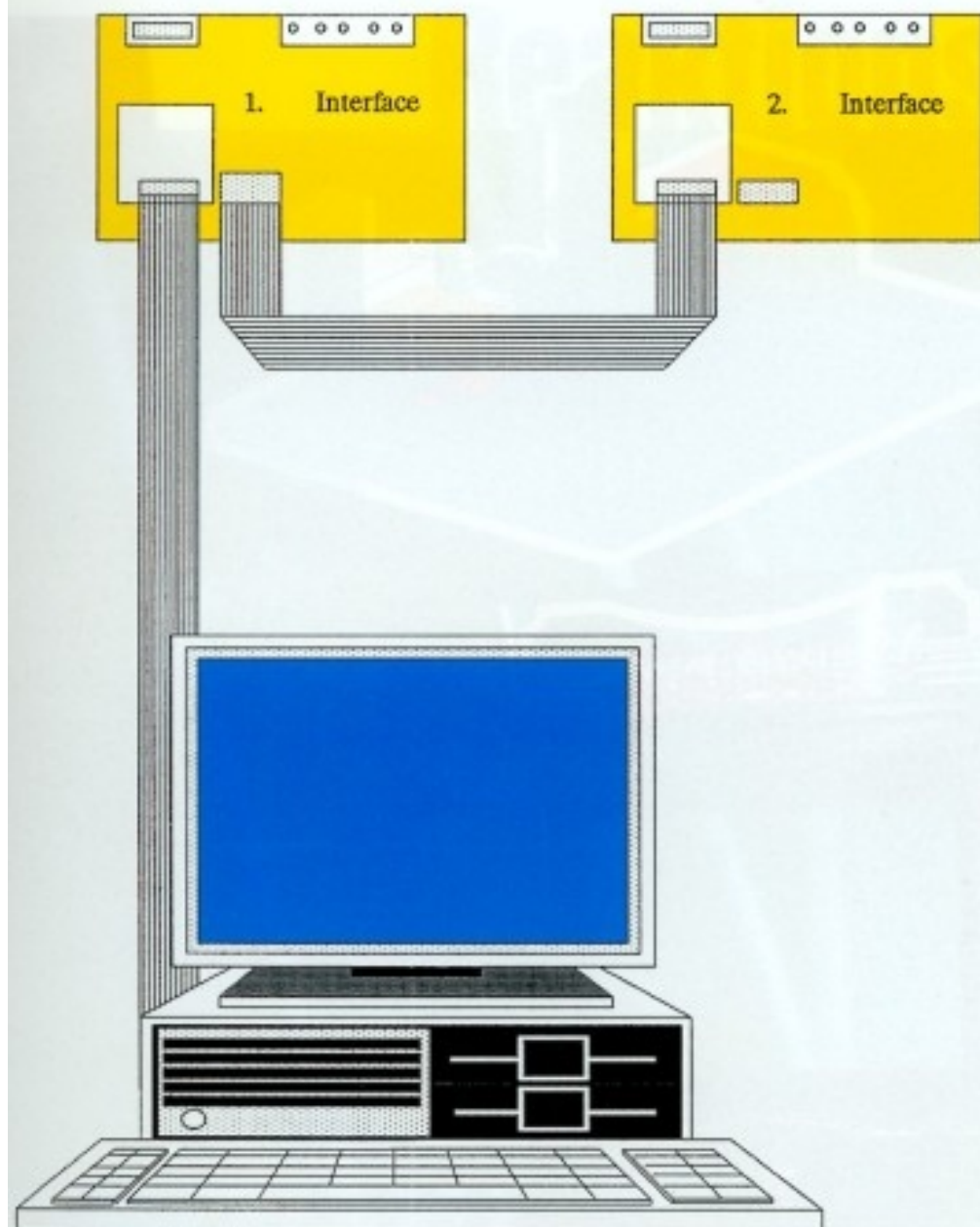
If KEYPRESSED then BEGIN
  Motoren__aus; Halt; END;

```

Sobald Du eine Taste am Computer drückst, bleibt das Modell stehen, und das Programm wird abgebrochen.

Wichtig ist nur, daß das Interface am Computer angeschlossen ist — sonst kann sich das Programm „aufhängen“, da keine Rückmeldung vom Interface kommt.

Die Vorbereitungen sind nun erledigt. Jetzt kannst Du die Anleitung ruhig einmal durchblättern und hier und da etwas schmökern — und natürlich das erste Modell aussuchen.





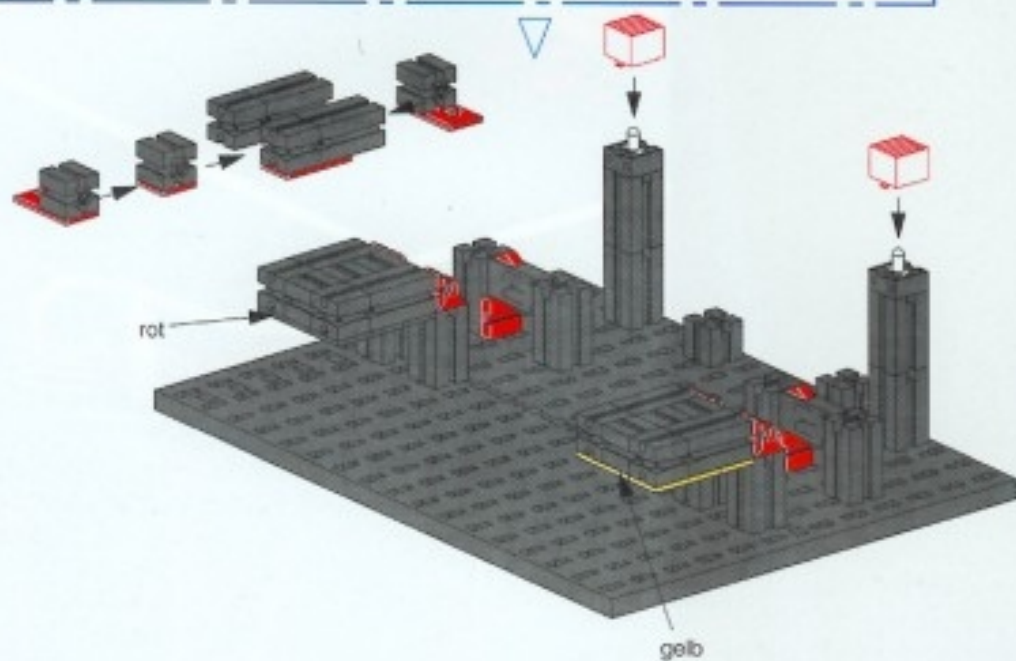
Reaktions- tester



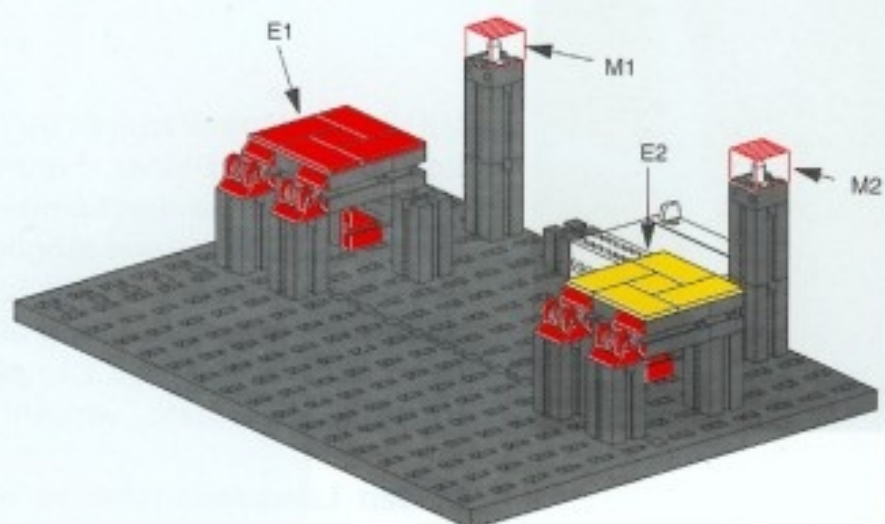
Mit diesem Modell kannst Du Deine Reaktion und die Deiner Freunde testen. Wenn eine der beiden Lampen aufleuchtet, muß so schnell wie möglich die Taste (rot oder gelb) vor der brennenden Lampe gedrückt werden. Die Zeit, die zwischen dem Aufleuchten der Lampe und dem Drücken der Taste vergeht, wird gemessen.

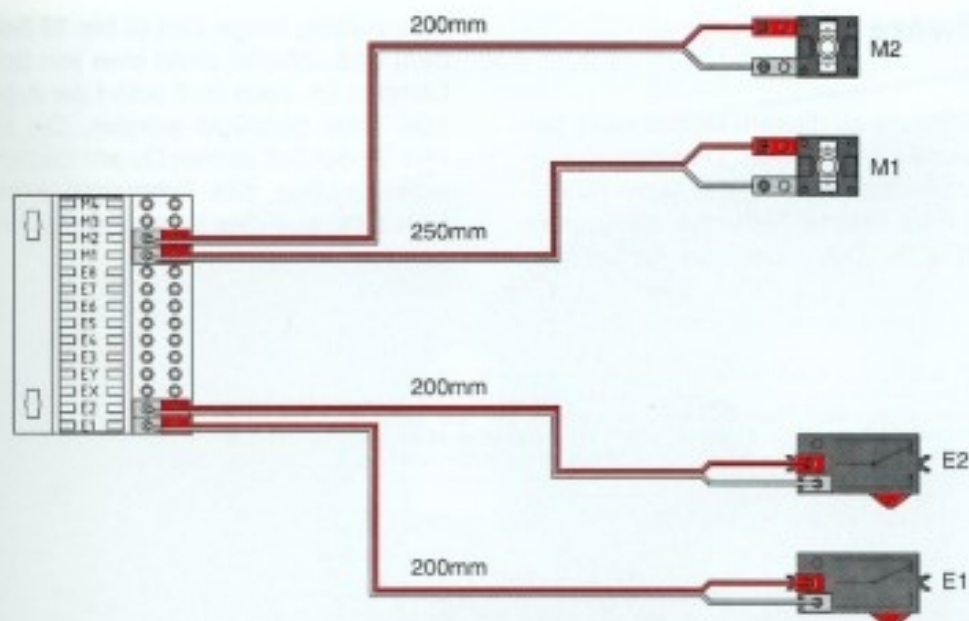
In den Lampenbausteinen müssen Kugellampen verwendet werden.

1a



1b





Verdrahtung

Die beiden Lampen werden an die Ausgänge M1 und M2 angeschlossen, die Taster an die Eingänge E1 und E2. Beide Taster verwenden den Arbeitskontakt. Zur Lampe an M1 gehört der Taster an E1, und zur Lampe an M2 gehört der Taster an E2. Achte auf die richtigen Kabellängen.

Starte nun das Programm LUCKY LOGIC und teste die Verdrahtung mit dem Menüpunkt OPTIONEN DIAGNOSE. Beim Betätigen der Taster müssen sich

die unten aufgeführten Eingangs-Anzeigen entsprechend verändern. Nun schalte die M-Ausgänge (Lampen) durch Anklicken mit der Maus, wie es in der Beschreibung zur Software erklärt ist. Wenn alles klappt, kann die Software geladen und gestartet werden.

Testplan:

E1	Taster bei Lampe 1
E2	Taster bei Lampe 2
M1	Lampe 1
M2	Lampe 2

Software

Die Software zu diesem Modell ist in Turbo-Pascal geschrieben. Das Pascal-Programm befindet sich in der Datei REAKTION.PAS. Starte nun das Programm REAKTION. Der Computer wartet nun

eine zufällig lange Zeit (0 bis 10 Sekunden) und schaltet dann eine von beiden Lampen an. Jetzt muß sofort die zugehörige Taste gedrückt werden. Die Start- und Stopp-Zeit kannst Du am Bildschirm ablesen. Wie das Programm arbeitet, siehst Du aus den Kommentaren im folgenden Listing.

```
PROGRAM Reaktion; { Turbo Pascal 5.0 - 6.0, Datei: REAKTION.PAS }
{ ..... }
{ Reaktionsspiel: Nach einer Wartezeit wird die rechte oder die linke Lampe }
{ eingeschaltet und die Zeit gemessen, bis die zugehörige Taste gedrückt ist }
{ Dauer der Wartezeit und Auswahl der Lampe erfolgen zufallsgesteuert }
{ ..... }

USES CRT,
      DOS,
      Fischer;

VAR Stunde_1, Minute_1, Sekunde_1, HndstSek_1,      { für Zeitmessung }
    Stunde_2, Minute_2, Sekunde_2, HndstSek_2 : WORD; { Start- und Stoppzeit }
    Zufallszeit : INTEGER; { Wartezeit zu Beginn }
    Inp_Outp_Nr : BYTE; { aktivierte Lampe }

{ ..... }

BEGIN { Programm Reaktion }
  CLRSKR;
  RANDOMIZE;
  Zufallszeit := RANDOM(10000); { Wartezeit 0 - 10 s }
  DELAY(Zufallszeit);
  Inp_Outp_Nr := RANDOM(3); { Lampe auswählen }
  IF Inp_Outp_Nr = 0 THEN Inp_Outp_Nr := 1;
  GETTIME(Stunde_1, Minute_1, Sekunde_1, HndstSek_1); { Startzeit speichern }
  REPEAT
    Motor(Inp_Outp_Nr, Links);
  UNTIL Taster(Inp_Outp_Nr); { Warten auf Tastendruck }
  GETTIME(Stunde_2, Minute_2, Sekunde_2, HndstSek_2); { Stoppzeit }
  Motor(Inp_Outp_Nr, Aus);
  GOTOXY(1,1);
  WRITELN('Startzeit: ', Stunde_1:2, ':', Minute_1:2, ':', Sekunde_1, ':', HndstSek_1:2);
  WRITELN('Endezeit: ', Stunde_2:2, ':', Minute_2:2, ':', Sekunde_2, ':', HndstSek_2:2);
  { Anmerkung: Der PC liefert die Zeit nur auf etwa 1/18 Sekunde genau }
END. { Programm Reaktion }
```



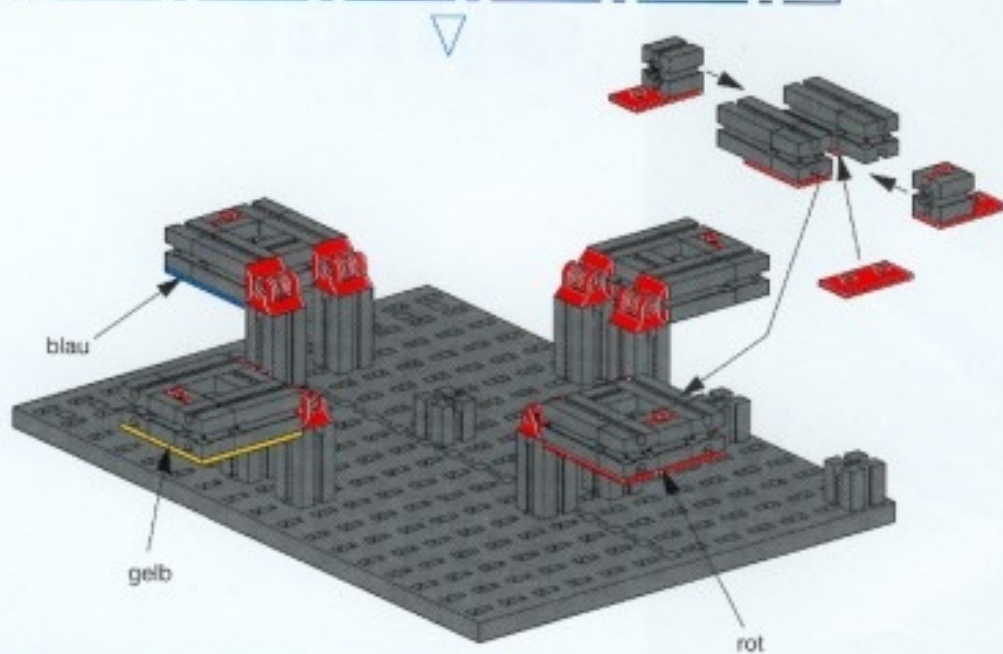
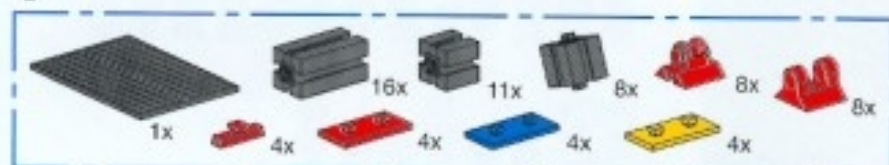


Melodie- Spiel

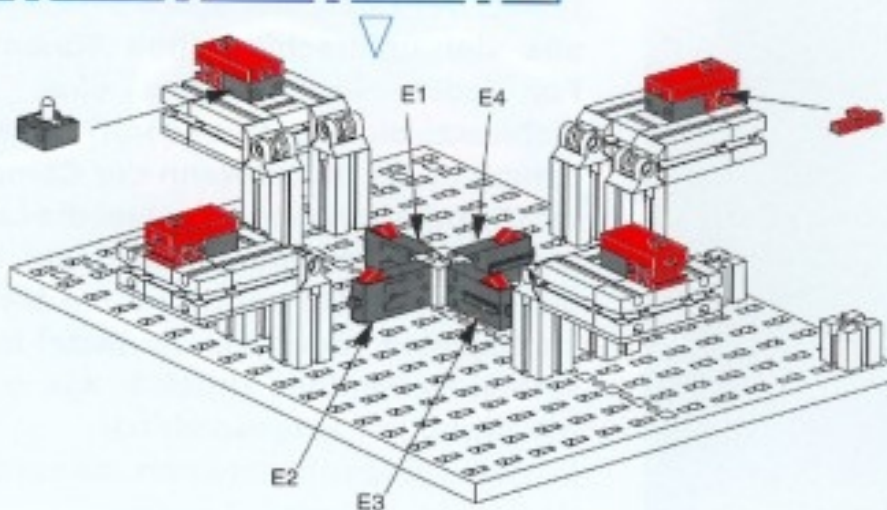


Hier wird Dein Erinnerungsvermögen getestet. Der Computer spielt eine Melodie aus vier unterschiedlichen Tönen vor. Für jeden Ton gibt es eine Taste (schwarz, blau, gelb und rot) mit einer Lampe in der Mitte. Wenn der Computer die Melodie vorspielt, leuchtet die Lampe in den entsprechenden Tastern auf. Dann bist Du dran. Du mußt nun die Melodie nachspielen, indem Du die Taster in derselben Reihenfolge drückst, wie sie Dir der Computer vorgespielt hat. In den Lampenbausteinen müssen Kugellampen verwendet werden.

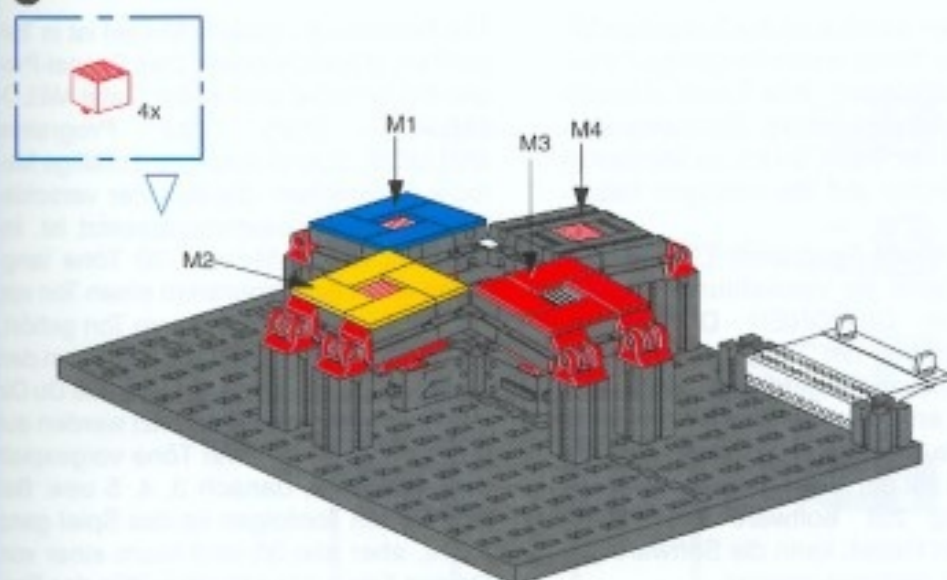
1



2



3



Verdrahtung

Die Lampen werden an die Ausgänge M1 bis M4, die Taster an die Eingänge E1 bis E4 angeschlossen. Alle Taster arbeiten mit dem Arbeitskontakt. Zur Lampe an M1 gehört der Taster an E1, zu M2 gehört E2 usw. Achte auf die richtigen Kabel-längen.

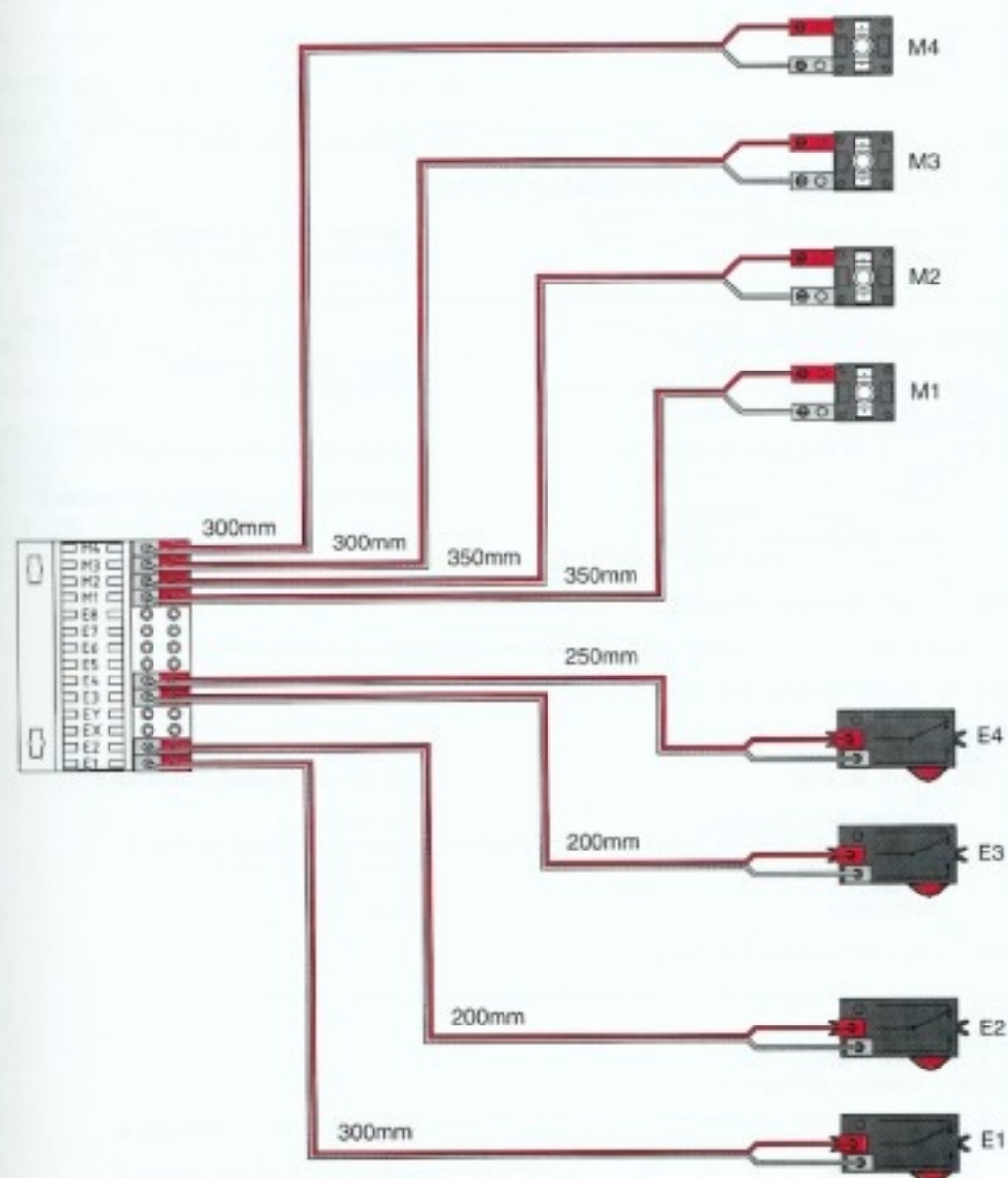
Starte nun das Programm LUCKY LOGIC und teste die Verdrahtung mit dem Menüpunkt OPTIONEN DIAGNOSE. Beim Betätigen der Taster müssen sich die im Testplan aufgeführten Eingangs-Anzeigen entsprechend verändern. Nun schalte die M-Ausgänge (Lampen) durch Anklicken mit der Maus, wie es in der Beschreibung zur Software erklärt ist. Wenn alles klappt, kann die Software geladen und gestartet werden.

Testplan:

E1	Taster an Lampe 1
E2	Taster an Lampe 2
E3	Taster an Lampe 3
E4	Taster an Lampe 4
M1	Lampe 1
M2	Lampe 2
M3	Lampe 3
M4	Lampe 4

Software

Die Software zu diesem Modell ist in Turbo-Pascal geschrieben. Das Pascal-Programm befindet sich in der Datei MELODIE.PAS. Starte das Programm MELODIE. Zuerst wird eine zufällige Melodie gespeichert, die aus vier verschiedenen Tönen zusammengesetzt ist. Insgesamt ist die Melodie 30 Töne lang. Dann spielt das Programm einen Ton vor und im Taster, der zu diesem Ton gehört, leuchtet die Lampe auf. Du mußt nun den Taster drücken, um zu zeigen, daß Du den Taster gemerkt hat. Jetzt werden auf die gleiche Weise zwei Töne vorgespielt und abgefragt; danach 3, 4, 5 usw. Bei den kurzen Tonfolgen ist das Spiel ganz leicht, aber alle 30 wird kaum einer vor Deinen Freunden schaffen. Wie das Programm arbeitet, siehst Du aus den Kommentaren im folgenden Listing.



Melodiespiel

```

PROGRAM Melodiespiel; { Turbo Pascal 5.0 - 6.0 }
{.....}
{ Erzeugt eine Zufallsmelodie von 30 Tönen (Auswahl aus jeweils 4 verschie- }
{ denen Tönen). Die Melodie wird dann in immer länger werdender Folge vorge- }
{ spielt und die zum Ton gehörige Taste angezeigt (Lampe) und muß dann nach- }
{ gespielt werden. Bei einem Fehler endet das Programm. }
{.....}

USES CRT,
      DOS,
      Fischer;

CONST Max_Zufallszahlen = 30; { Maximalzahl der Töne }

TYPE T_Melodie = ARRAY [1..Max_Zufallszahlen] OF BYTE;

VAR Melodie : T_Melodie; { Speicher für die Tonfolge }
    Fehler : BOOLEAN; { Beim Abfragen alles richtig? }

{.....}

PROCEDURE Zufallsmelodie_erzeugen;
VAR i : BYTE; { Zähler für die Tonfolge }
    Zufallszahl : BYTE; { Zufälliger Ton }
BEGIN { Zufallsmelodie_erzeugen }
    RANDOMIZE;
    FOR i := 1 TO Max_Zufallszahlen DO { zufällige Melodie speichern }
        BEGIN
            REPEAT { Zufallszahl 1,2,3 oder 4 erzeugen }
                Zufallszahl := RANDOM(5);
            UNTIL Zufallszahl <> 0;
            Melodie[i] := Zufallszahl;
        END;
    END; { Zufallsmelodie_erzeugen }

{.....}

PROCEDURE Spiele_Tonreihe_bis_Ton(i:BYTE);
VAR j, k : INTEGER;
BEGIN { Spiele_Tonreihe_bis_Ton }
    DELAY(1000); { 1 Sekunde Pause }
    FOR j := 1 TO i DO { bis zum Ton Nummer i spielen }
        BEGIN
            SOUND(Melodie[j]*1000); { Ton spielen }
            FOR k := 1 TO 6000 DO Motor(Melodie[j],Rechts); { Lampe eine gewisse Zeit an }
            Motor(Melodie[j],Aus); { Lampe und Ton aus }
            NOSOUND;
            DELAY(100); { kurze Pause }
        END;
    END; { Spiele_Tonreihe_bis_Ton }

{.....}

PROCEDURE Tonreihe_abfragen(i:BYTE);
VAR j,TasterNr : BYTE;
BEGIN { Tonreihe_abfragen }
    j := 0;
    REPEAT { bis zum Ton Nr. i abfragen }
        INC(j);
        TasterNr := 0;
        REPEAT { warten bis eine der Tasten gedrückt wurde }
            IF Taster(1) THEN TasterNr := 1;
            IF Taster(2) THEN TasterNr := 2;
            IF Taster(3) THEN TasterNr := 3;
            IF Taster(4) THEN TasterNr := 4;
        UNTIL TasterNr <> 0;
        Motor(TasterNr,Rechts); { Lampe und Ton einschalten }
    END;
END;

```

```

SOUND(TasterNr * 1000);
REPEAT UNTIL NOT Taster(TasterNr);           { solange die Taste gedrückt ist }
NOSOUND;                                       { Lampe und Ton wieder aus }
Motor(TasterNr,Aus);
Fehler := TasterNr <> Melodie[j];             { Wenn falsche Taste, dann Fehler }
UNTIL Fehler OR (i=j);                         { speichern und aufhören }
END; { Tonreihe_abfragen }

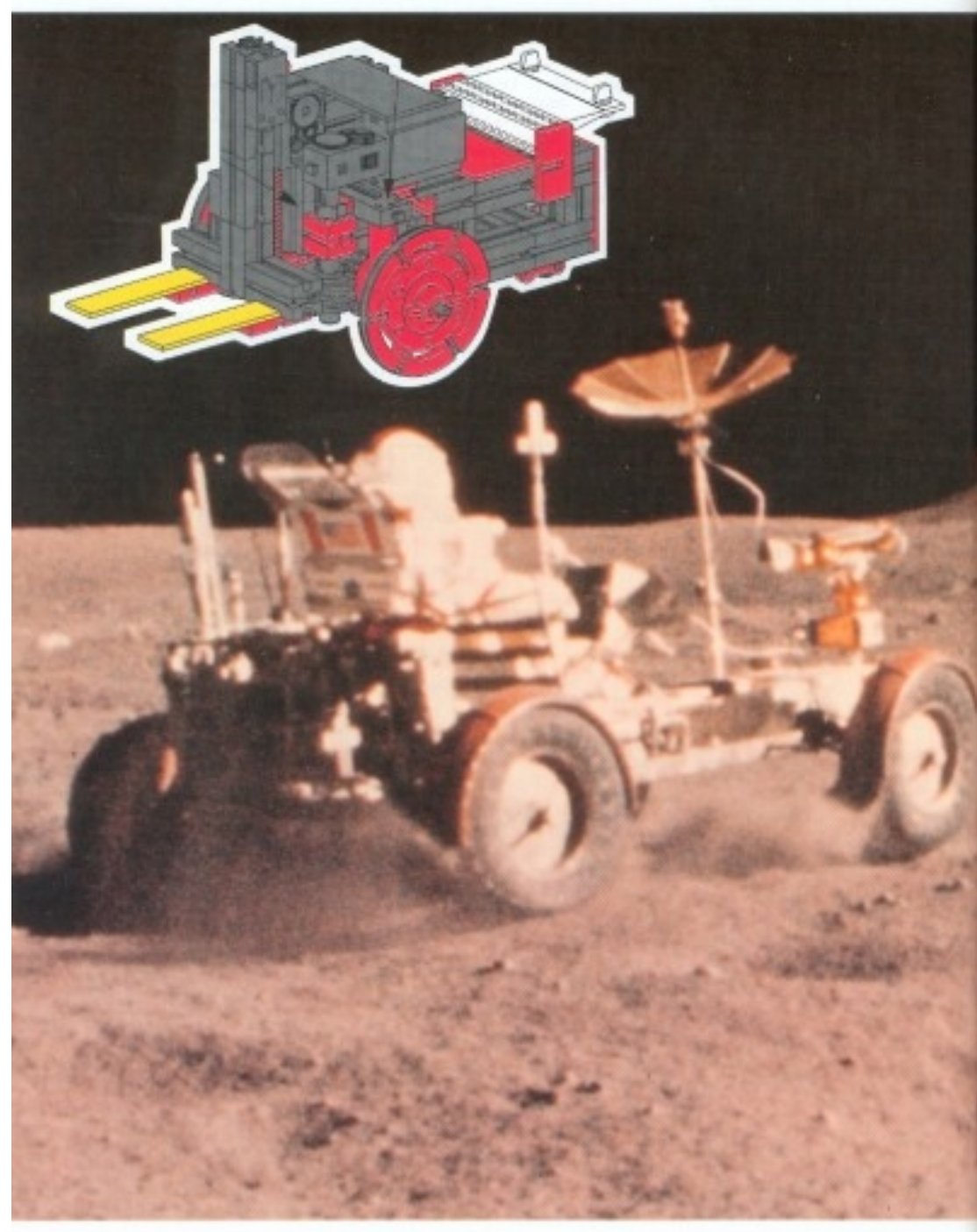
{.....}

PROCEDURE Zufallsmelodie_vorspielen_und_abfragen;
VAR i, j : BYTE;
BEGIN { Zufallsmelodie_vorspielen_und_abfragen }
  i := 0;
  REPEAT                                       { erst 1 Ton, dann 2, 3, usw. }
    INC(i);
    Spiele_Tonreihe_bis_Ton(i);               { Melodie vorspielen }
    Tonreihe_abfragen(i);                     { Jetzt nachspielen }
    CLRSOR;
    IF NOT Fehler THEN WRITE('Sehr gut !')
      ELSE WRITE('Fehler !');
  UNTIL Fehler OR (i = Max_Zufallszahlen);    { bis zu 30 Tönen oder Fehler }
END; { Zufallsmelodie_vorspielen_und_abfragen }

{.....}

BEGIN { Programm Melodie }
  CLRSOR;
  NOSOUND;                                    { Ton abschalten (zur Sicherheit) }
  Zufallsmelodie_erzeugen;
  Zufallsmelodie_vorspielen_und_abfragen;
  NOSOUND;                                    { Ton abschalten (zur Sicherheit) }
END. { Programm Melodie }

```



Turtle (Schild- kröte)



Die Turtle (zu deutsch: Schildkröte) ist ein kleiner Wagen mit zwei angetriebenen Vorderrädern, der vom Programm vorgegebene Wegstrecken abfahren kann. Jedes Antriebsrad besitzt einen eigenen Motor, es kann also unabhängig vorwärts und rückwärts drehen. Durch diesen Trick ist es der Turtle möglich, nicht nur vorwärts und rückwärts zu fahren, sondern sie kann auch beliebige Kurven fahren und sogar auf der Stelle drehen — je nachdem, wie die Motoren angesteuert werden.

Besser als jeder Text zeigt die folgende Tabelle alle Möglichkeiten der Turtle-Bewegung:

Bewegung der Turtle	Motor M1 rechts	Motor M2 links
Stillstand	aus	aus
vorwärts	vorwärts	vorwärts
rückwärts	rückwärts	rückwärts
Rechtskurve vorwärts	aus	vorwärts
Linkskurve vorwärts	vorwärts	aus
Rechtskurve rückwärts	rückwärts	aus
Linkskurve rückwärts	aus	rückwärts
rechtsdrehen	rückwärts	vorwärts
linksdrehen	vorwärts	rückwärts

Damit der Computer feststellen kann, wie weit die Turtle schon gefahren ist, sind bei beiden Motoren Taster befestigt, die über Impulsscheiben betätigt werden. Bei jeder Umdrehung der Antriebs-Welle schließt der Taster fünfmal. Der Taster an E1 gehört zum Motor an M1 und der Taster an E2 zum Motor an M2.

Vorne an der Turtle ist eine Transportgabel angebracht, die über Motor M3 nach oben und unten bewegt werden kann (wie bei einem Gabelstapler). Damit kann die Turtle eine Palette aufnehmen und an einem anderen Platz wieder absetzen. Auch hier sind wieder Taster nötig, damit das Computerprogramm feststellen kann, wann die Gabel die obere oder untere Endposition erreicht hat. Sie sind richtig eingestellt, wenn Du sie so einbaust, wie es die Baustufen zeigen. Der Taster an E3 stellt die untere Endstellung fest, der Taster an E4 die obere Endstellung.

Die Turtle fährt am besten auf einem glatten, ebenen Untergrund. Teppichboden ist nicht geeignet. Das Verbindungskabel zum Interface sollte von oben kommen, damit sich die Turtle nicht darin verfangen kann.

Verdrahtung

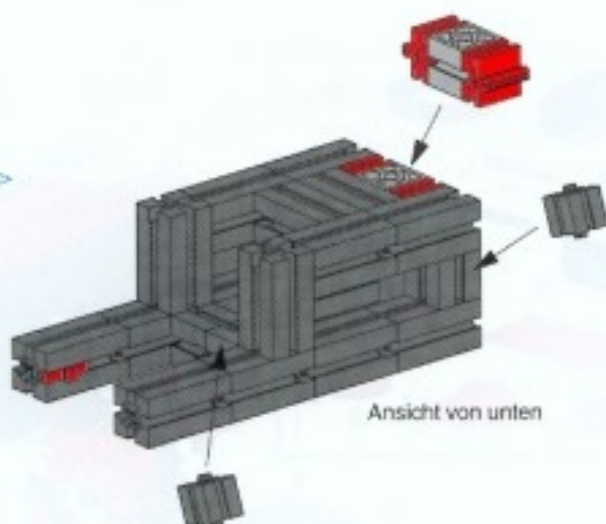
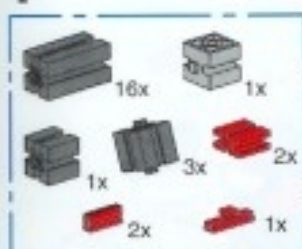
Die Taster und Motoren werden so verdrahtet, wie Du es im Plan siehst. Der Taster an E1 gehört zum Motor an M1 und der Taster an E2 zum Motor an M2. M3 bewegt die Gabel. Der Taster an E3 stellt die obere Endstellung fest, der Taster an E4 die untere Endstellung. Achte auf die richtigen Kabellängen.

Starte nun das Programm LUCKY LOGIC und teste die Verdrahtung mit dem Menüpunkt OPTIONEN DIAGNOSE. Beim Betätigen der Taster müssen sich die im Testplan aufgeführten Eingangs-Anzeigen entsprechend verändern. Nun schalte die M-Ausgänge (Motoren) durch Anklicken mit der Maus, wie es in der Beschreibung zur Software erklärt ist. Wenn alles klappt, kann die Software geladen und gestartet werden.

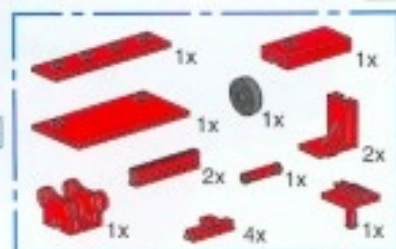
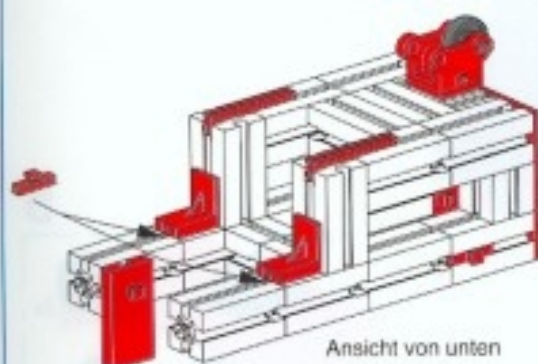
Testplan:

E1	Schrittzähler für rechten Antriebsmotor; Signaländerung nur, wenn der Motor läuft
E2	Schrittzähler für linken Antriebsmotor; Signaländerung nur, wenn der Motor läuft
E3	Endstellung der Gabel oben
E4	Endstellung der Gabel unten
M1	Rechter Antriebsmotor
M2	Linker Antriebsmotor
M3	Antriebsmotor für die Gabel

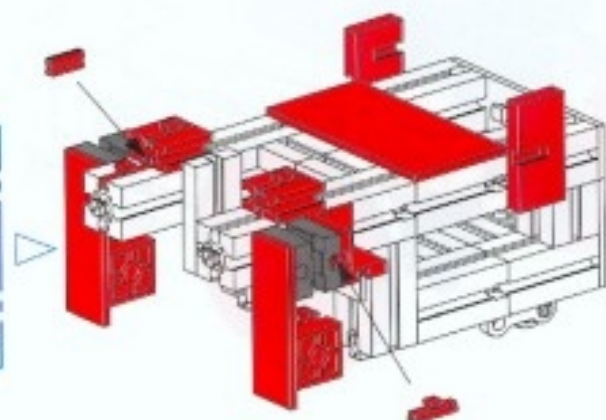
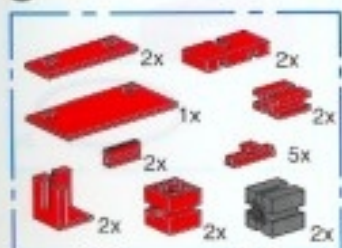
1



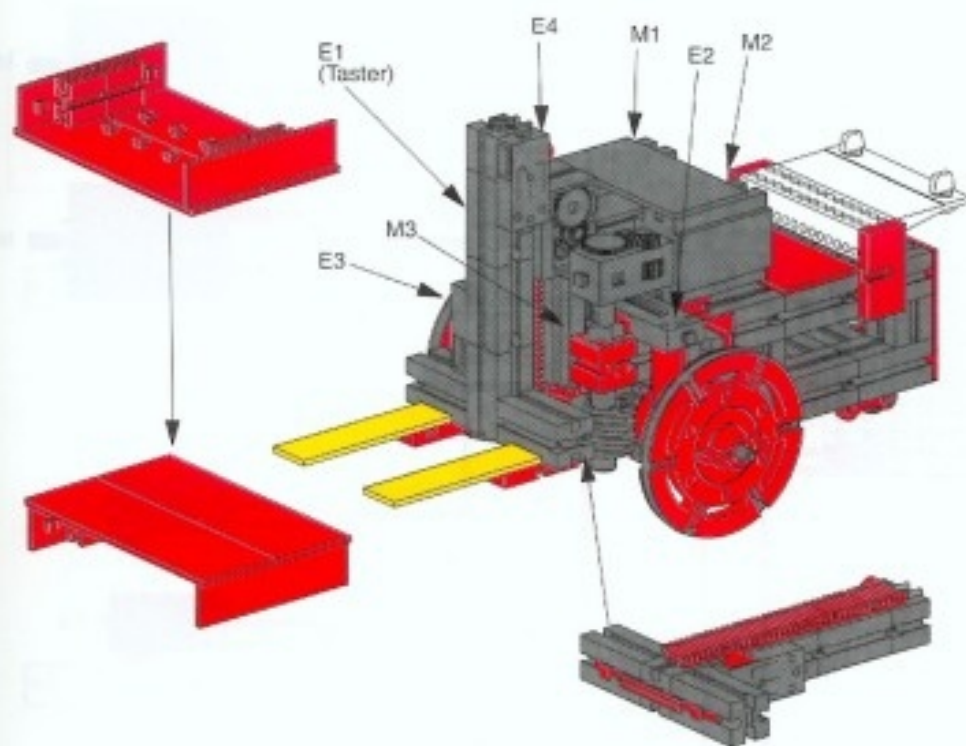
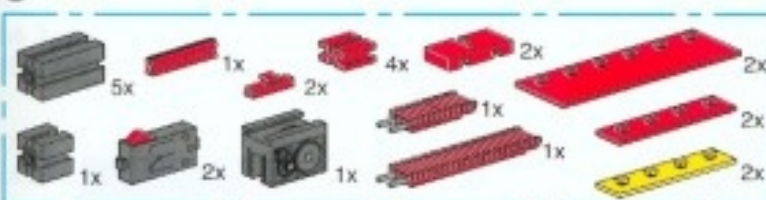
2

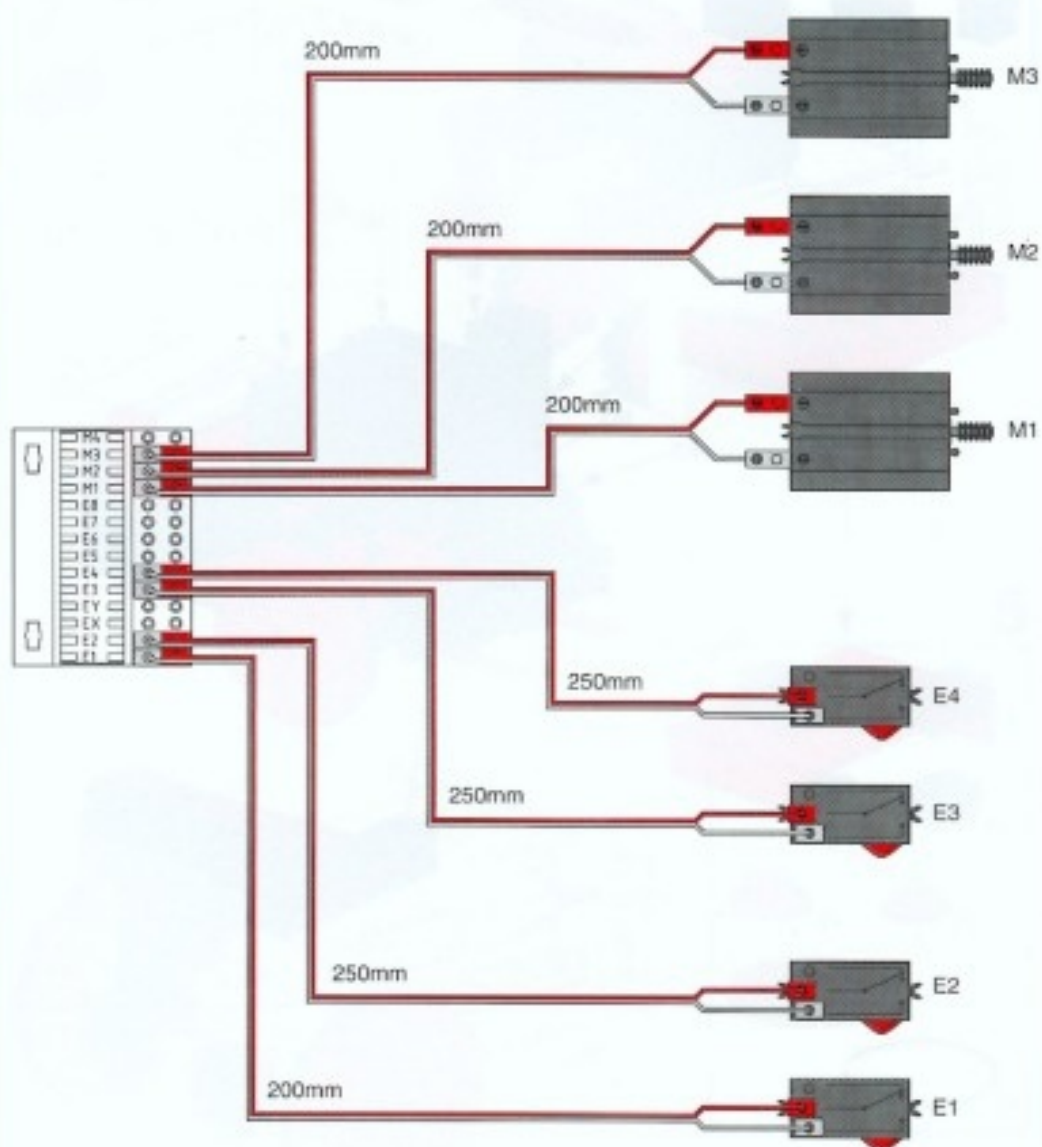


3



6





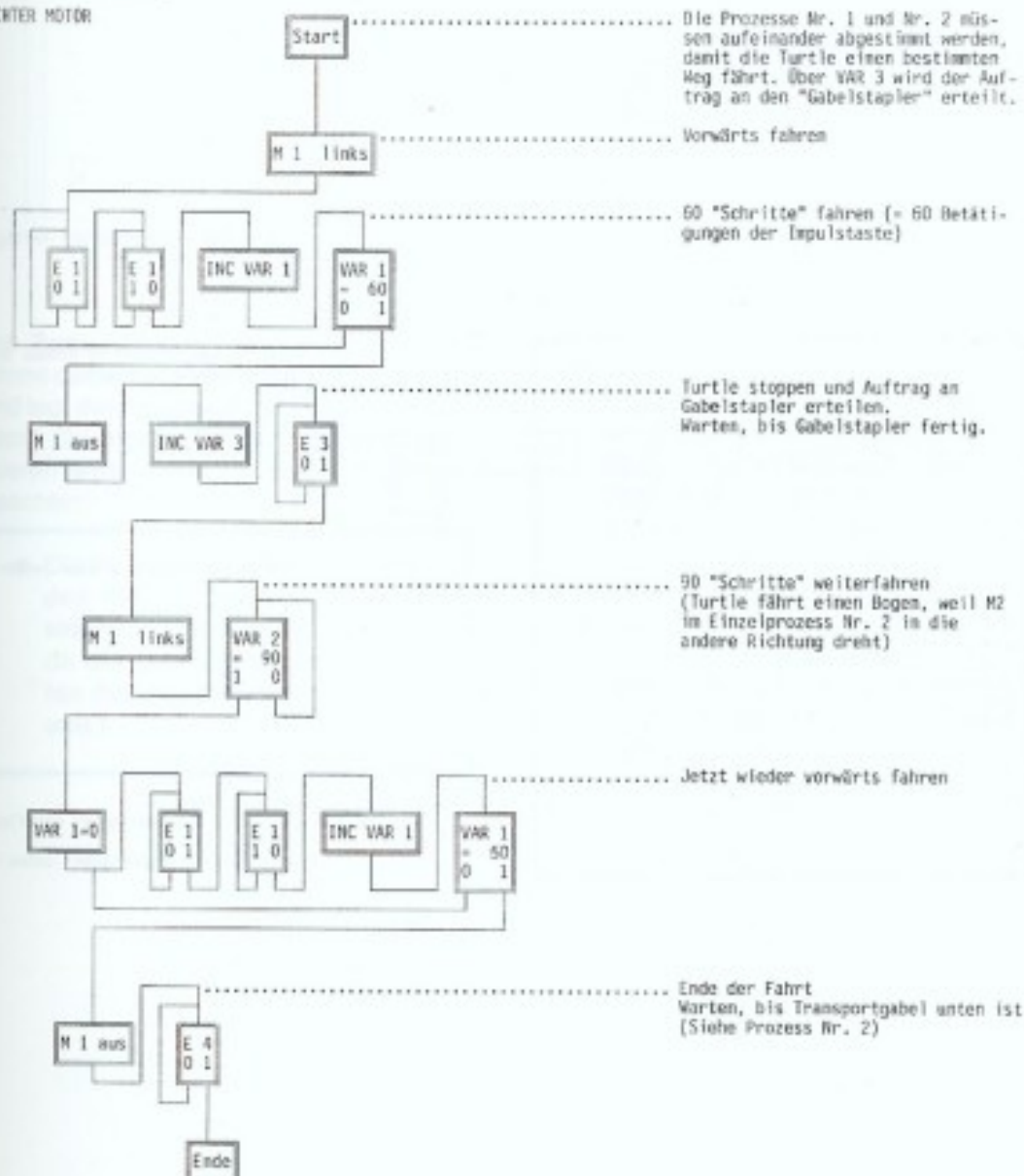
Software

Das Turtle-Programm ist mit LUCKY LOGIC geschrieben. Es besteht aus den

Dateien TURTLE.FTG, TURTLE1.FTL, TURTLE2.FTL und TURTLE3.FTL. Lade den Gesamtprozeß TURTLE und starte das Programm mit dem Menüpunkt RUN GESAMTPROZESS.

Turtles Prozess Nr. 1

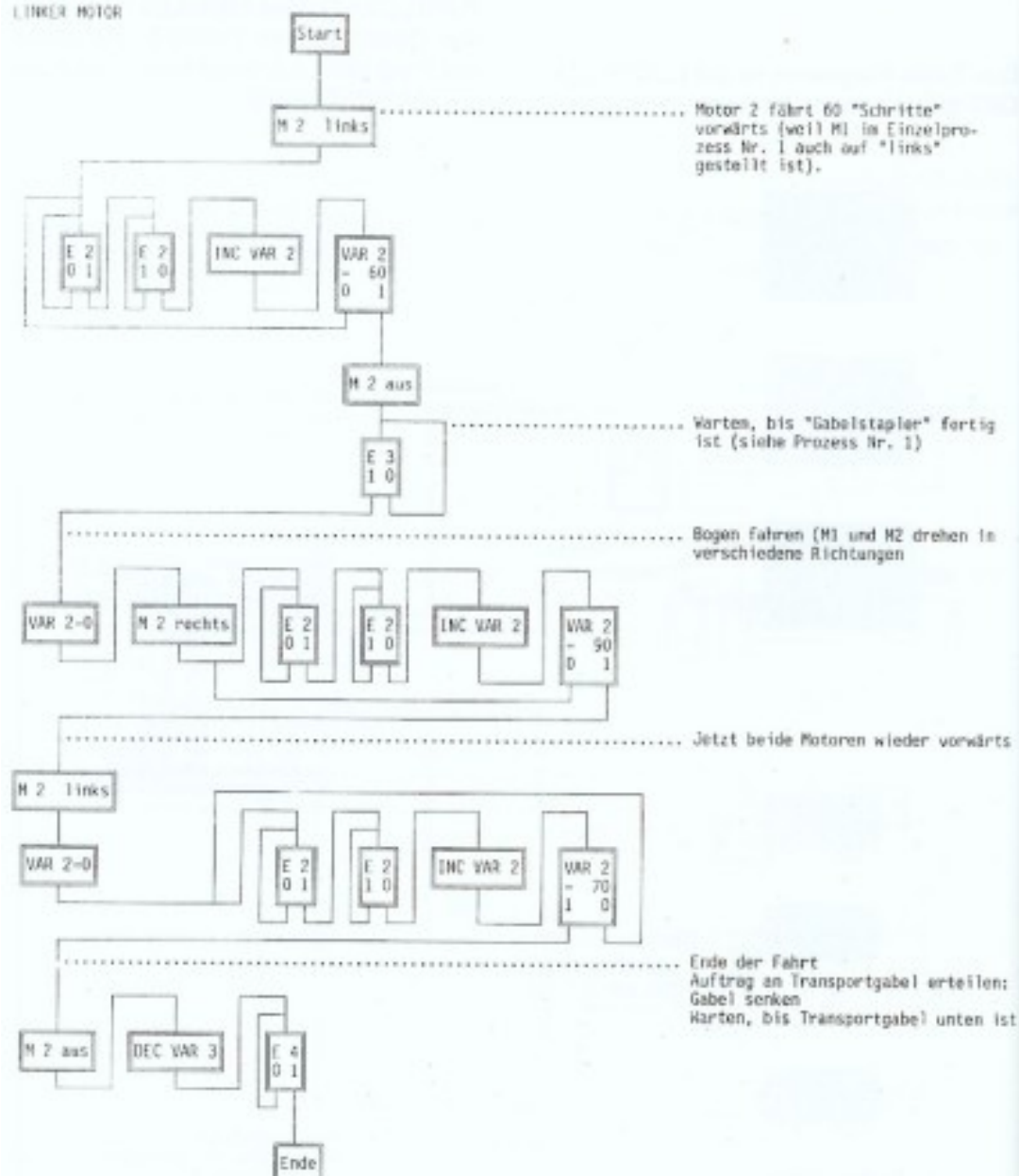
RECHTER MOTOR



Turtle (Schildkröte)

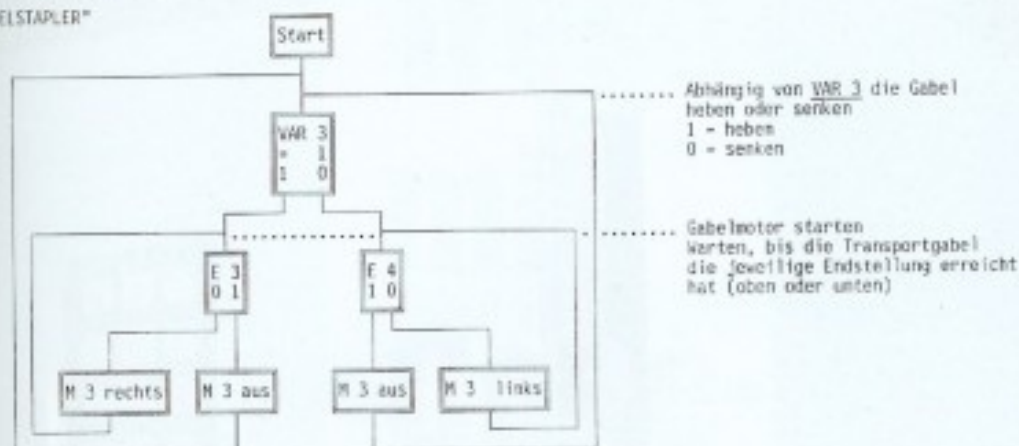
Turtle: Prozess Nr. 2

LINKER MOTOR



Turtle: Prozess Nr. 3

"GABELSTAPLER"



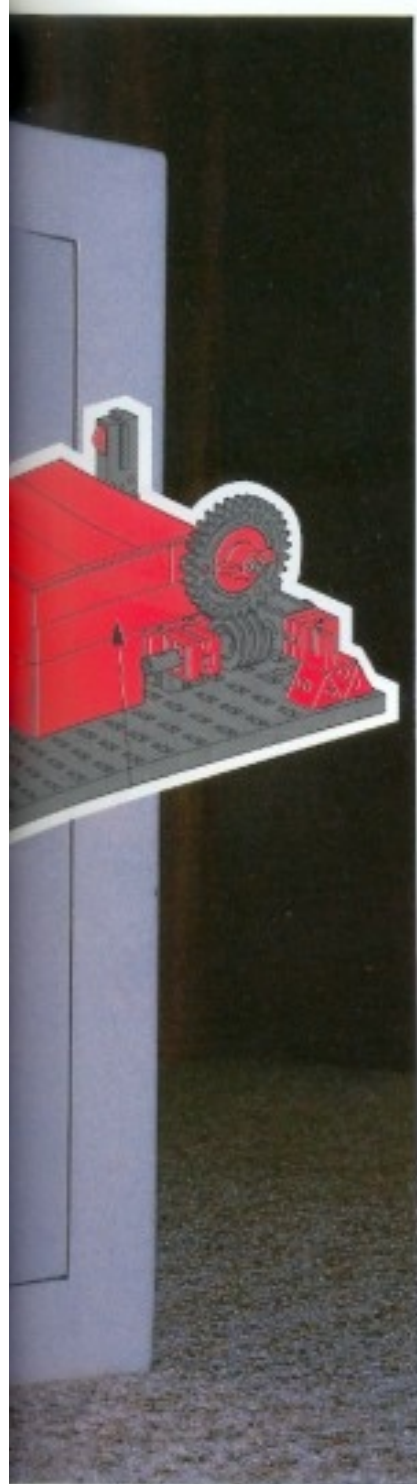
Die Turtle fährt ein Stück geradeaus, nimmt dann die „Palette“ auf, fährt weiter und legt die Palette an anderer Stelle ab. Wenn Du eigene Fahrstreken programmieren willst, mußt Du folgende Dinge beachten:

- Die Prozesse 1 und 2 für die beiden Fahrmotoren müssen aufeinander abgestimmt werden, da die Drehrichtung der Motoren den Weg bestimmt (Kurven usw.).

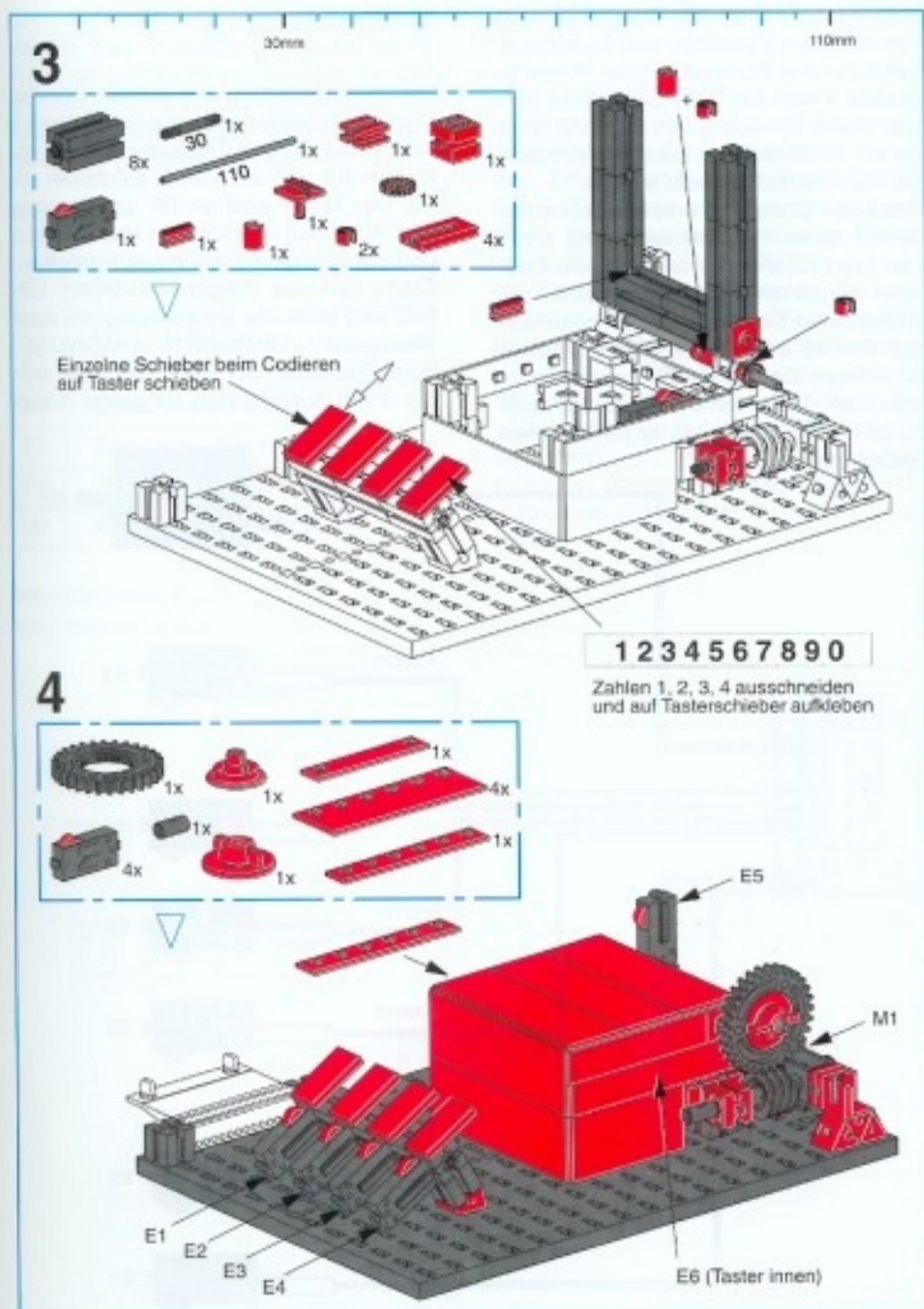
- Der Prozeß 3 für die Transportgabel wird über VAR 3 gesteuert. Bevor die Turtle weiterfährt, muß die Gabel erst „fertig“ sein. Das kann durch Abfragen von E3 oder E4 festgestellt werden (je nachdem, ob die Gabel oben oder unten stehen soll).
- Wenn die Laufzeiten von Prozeß 1 und 2 unterschiedlich lang sind (z. B. beim Kurvenfahren) sollte immer der Prozeß den Auftrag an die Gabel erteilen, der länger braucht. In dem beschriebenen Beispiel kannst Du das gut sehen.



Tresor mit Code- Schloß



Vielleicht hast Du schon Türen gesehen, die nicht mit einem Schlüssel geöffnet werden können, sondern bei denen man auf einer Tastatur eine Zahl eingeben muß. So ähnlich arbeitet auch der Tresor mit Codeschloß. Der Deckel öffnet sich automatisch, wenn der richtige Code an den vier Schaltern an E1 bis E4 eingetaetet wird. Dabei kannst Du den Code über das Programm selbst bestimmen (im Beispiel gilt: 1 — 3 — 2).

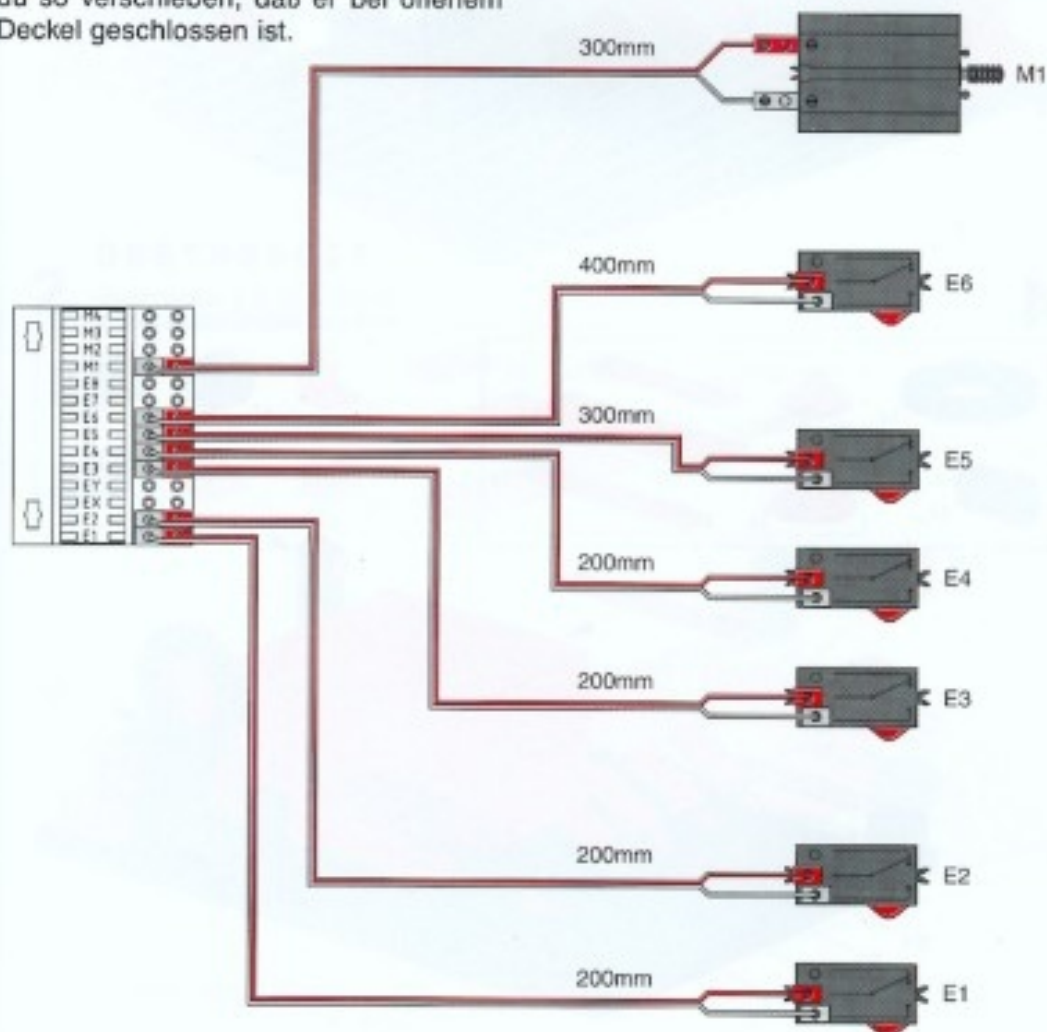


Zum Ausprobieren der Mechanik und zum richtigen Einstellen des Endtasters ziehst Du das Zahnrad mit der Rastachse etwa 4 mm aus dem U-Getriebe heraus. Wenn Du das Zahnrad nach links drehst, öffnet sich der Deckel, beim Drehen nach rechts schließt er sich.

Damit der Computer erkennt, wann der Deckel seine Endstellung (ganz offen oder ganz zu) erreicht hat, sind zwei Endtaster angebracht. Der Endtaster für geschlossenen Deckel an E6 ist innerhalb des Tresors angebracht. Der Endtaster für offenen Deckel an E5 befindet sich außen am Scharnier. Diesen Taster mußst du so verschieben, daß er bei offenem Deckel geschlossen ist.

Verdrahtung

Die Verdrahtung ist nicht schwer. Die vier Codetaster kommen an die Eingänge E1 bis E4, der äußere Endtaster (Deckel offen) an E5 und der innere Endtaster an E6. Der Motor wird an M1 angeschlossen. Achte auf die richtigen Kabellängen und die richtige Funktion der Endtaster. Starte nun das Programm LUCKY LOGIC und teste die Verdrahtung mit dem Menüpunkt OPTIONEN DIAGNOSE. Beim Betätigen der Taster müssen sich die unten aufgeführten Eingangs-Anzei-



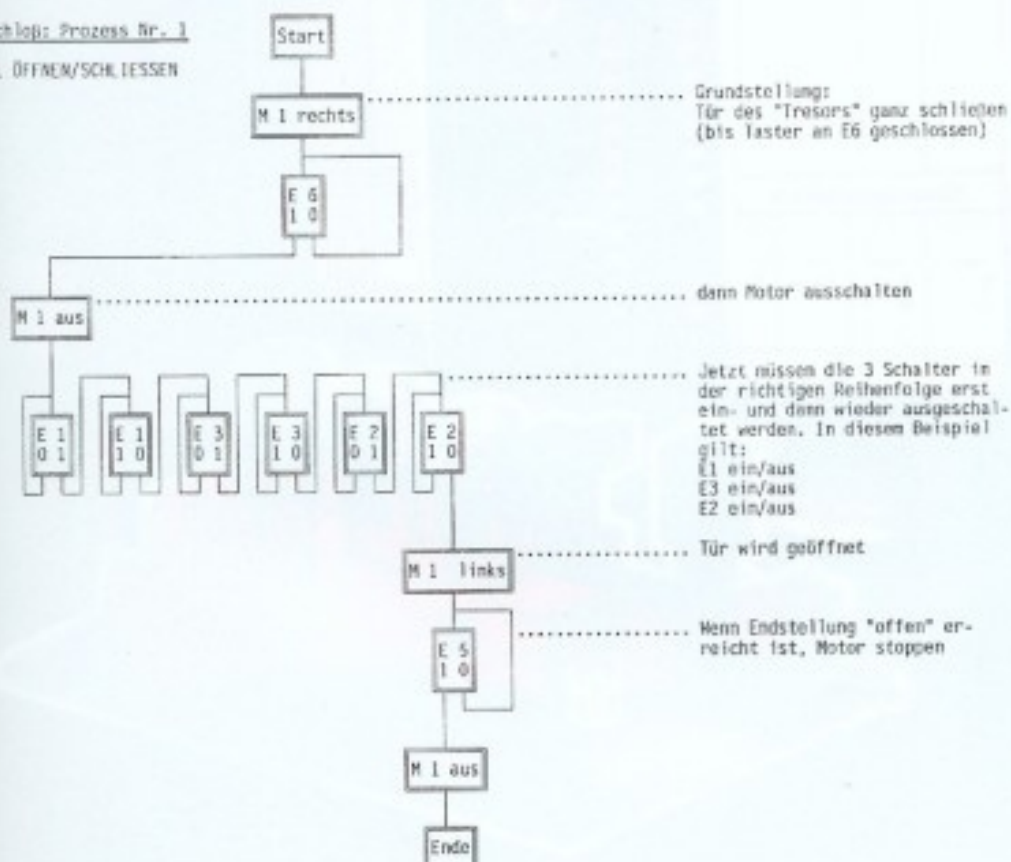
gen entsprechend verändern. Nun schalte den M-Ausgang (Motor) durch Anklicken mit der Maus, wie es in der Beschreibung zur Software erklärt ist. Wenn alles klappt, kann die Software geladen und gestartet werden.

Testplan:

E1	Codeschalter 1
E2	Codeschalter 2
E3	Codeschalter 3
E4	Codeschalter 4
E5	Endtaster „Deckel offen“
E6	Endtaster „Deckel geschlossen“
M1	Antriebsmotor für Deckel

Codeschloß: Prozess Nr. 1

DECKEL ÖFFNEN/SCHLIESSEN



Software

Das Codeschloß ist mit LUCKY LOGIC programmiert. Da nur ein Motor angesteuert werden muß, gibt es auch nur einen Einzelprozeß in der Datei CODESCHL.FTL. Lade den Einzelprozeß und starte das Programm mit dem Menüpunkt RUN EINZELPROZESS.

Im Programmbeispiel werden nur drei Schalter verwendet. Zum Öffnen der Box müssen die Schalter in der Reihenfolge E1 — E2 — E3 nacheinander ein- und wieder ausgeschaltet werden. Wenn Du willst, kannst Du ja noch den Schalter E4 programmieren, indem Du die Verbindung zur Steueranweisung „M1 LINKS“ aufträgst und dort den Schalter E4 einbaust.



Geld- automat



Bei vielen Fahrkarten- und Warenautomaten bekommt man Wechselgeld zurück, wenn man zuviel Geld eingeworfen hat (weil vielleicht nicht das passende Kleingeld parat war). Einige Automaten können sogar Geldscheine wechseln. So einem Geldautomaten kannst Du auch als fischertechnik-Modell bauen.

Durch das Programm gesteuert kann der gewünschte Betrag aus drei mit Münzen gefüllten Röhren ausgezahlt werden. Die Münzen dürfen aber nicht dünner als der Schieber sein (also keine Pfennige nehmen).

Beim Einschieben der Röhren darauf achten, daß sie nicht zu tief sitzen — der Schieber wird sonst eingeklemmt, und Dein Geldautomat funktioniert dann nicht richtig.

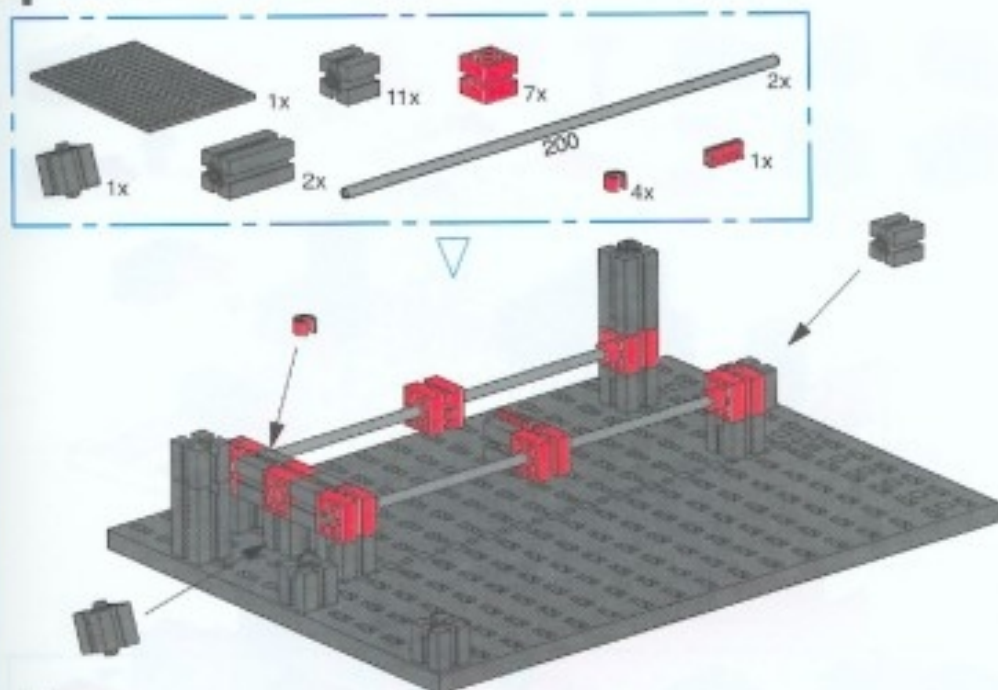
Wenn der Schlitten, auf dem die Schieber-Mechanik befestigt ist, auf den beiden Metallachsen schwer läuft, hilft ein kleiner Tropfen Haushalts- oder Nähmaschinenöl auf den Führungsachsen. Achte auch darauf, daß die Zangenmut-

ter auf der klemmbaren Schnecke und die Spannzange gut festgezogen sind, da sonst die Achsen durchdrehen.

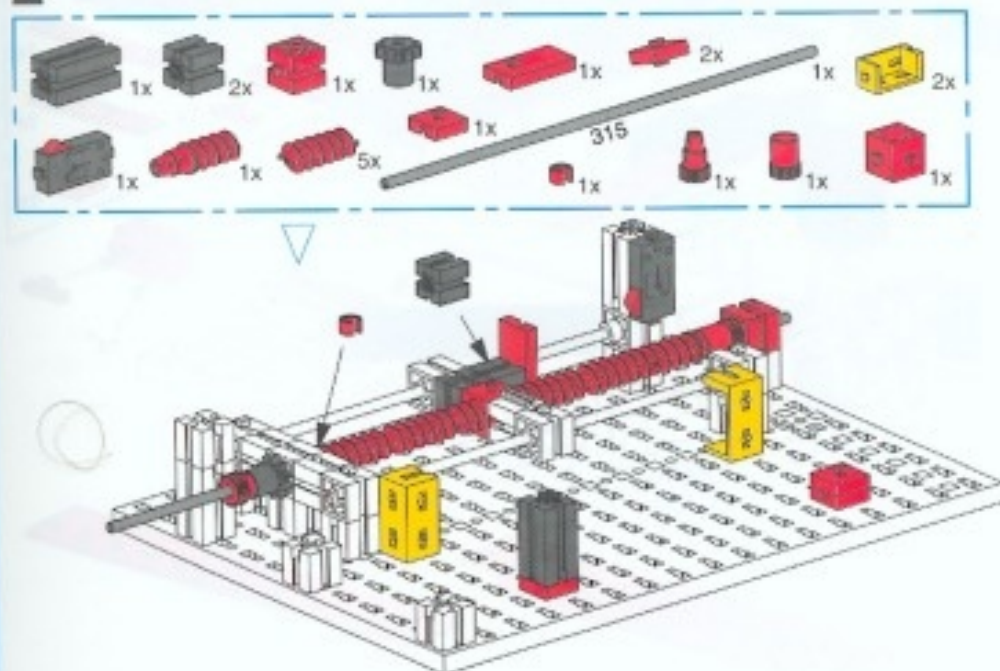
Wenn Du zum Ausprobieren der Mechanik den Schlitten mit der Hand bewegen willst, ziehst Du das Zahnrad mit der Rastachse etwa 4 mm aus dem Getriebe heraus. Nun kannst Du die Achse mit der Schnecke frei drehen.

Alle Taster arbeiten hier für den Computer, denn bei diesem Modell muß nicht nur die Position des Schiebers, sondern auch der vom Schlitten zurückgelegte Weg festgestellt werden. Der Taster an E4 wird von einer Impulsscheibe betätigt, die ihn bei jeder Umdrehung der Achse mit der Schnecke fünfmal betätigt.

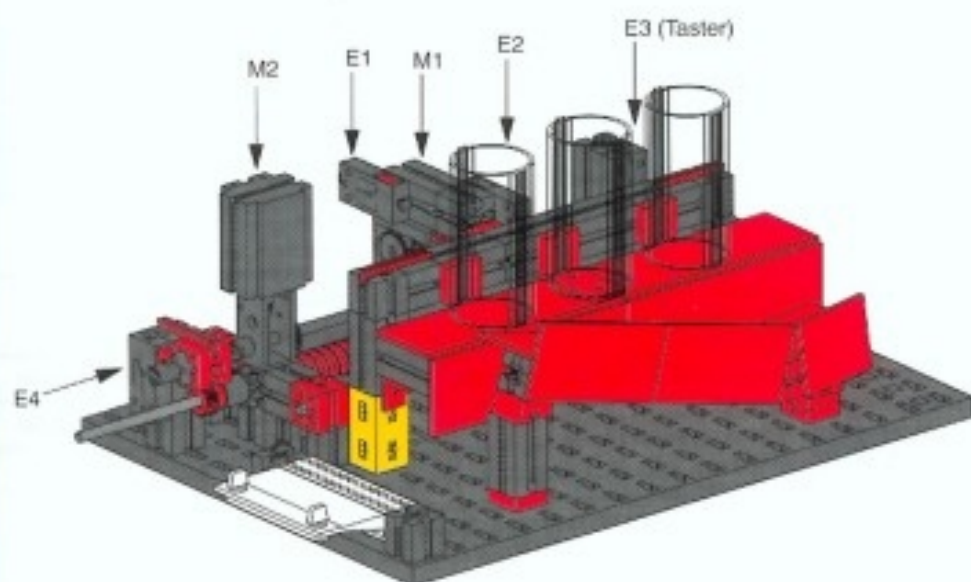
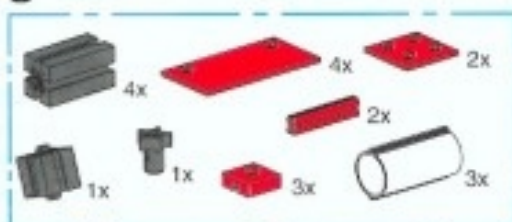
1



2



5



Verdrahtung

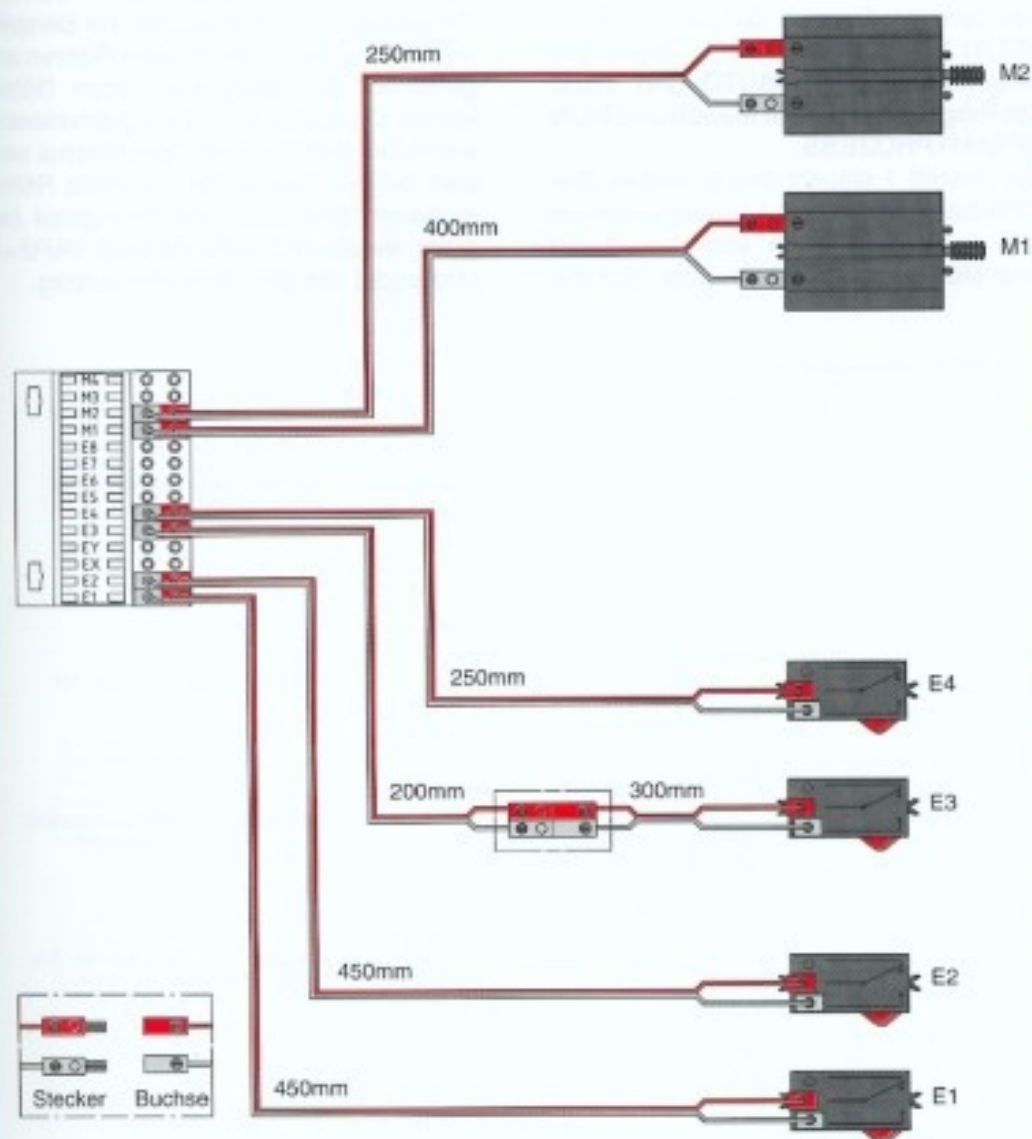
Der Motor an M1 steuert den Schieber, der Motor an M2 die horizontale Bewegung des Schlittens. Die Taster an E1 und an E2 melden dem Computer die beiden Endstellungen des Schiebers (E1 = Schieber hinten, E2 = Schieber vorne). Diese beiden Schalter mußt Du nach einem ersten Probelauf eventuell noch etwas verschieben. Der Taster an E3 meldet die rechte Endstellung des Schlittens und der Taster E4 zählt die Umdrehungen der Welle mit der Schnecke. Achte auf die richtigen Kabellängen; bei der Verbindung zum Taster an E3 mußt Du das Kabel aus zwei Teilen (20 cm und 30 cm) zusammensetzen. Dazu sind an einem Ende eines der beiden Kabel die Stecker gegen Steckbuchsen auszutauschen. Wie das geht, kannst Du am Anfang des Buchs nachlesen.

Starte nun das Programm LUCKY LOGIC und teste die Verdrahtung mit dem Menüpunkt OPTIONEN DIAGNOSE.

Beim Betätigen der Taster müssen sich die im Testplan aufgeführten Eingangs-Anzeigen entsprechend verändern. Nun schalte die M-Ausgänge (Motoren) durch Anklicken mit der Maus, wie es in der Beschreibung zur Software erklärt ist. Achtung beim Motor M1: Hier mußt Du rechtzeitig stoppen, damit der Schieber nicht zu weit nach vorne oder hinten läuft. Wenn alles klappt, kann die Software geladen und gestartet werden.

Testplan:

E1	Schieber hinten
E2	Schieber vorne
E3	rechte Endstellung des Schlittens
E4	Impulsscheibe des Schlittenantriebs
M1	Antriebsmotor des Schiebers
M2	Antriebsmotor des Schlittens



Software

Die Software für den Geldautomaten ist mit LUCKY LOGIC erstellt. Sie besteht aus den drei Dateien GELDAUTO.FTG, GELD1.FTL und GELD2.FTL. Lade den Gesamtprozeß GELDAUTO und starte das Programm mit dem Menüpunkt RUN GESAMTPROZESS.

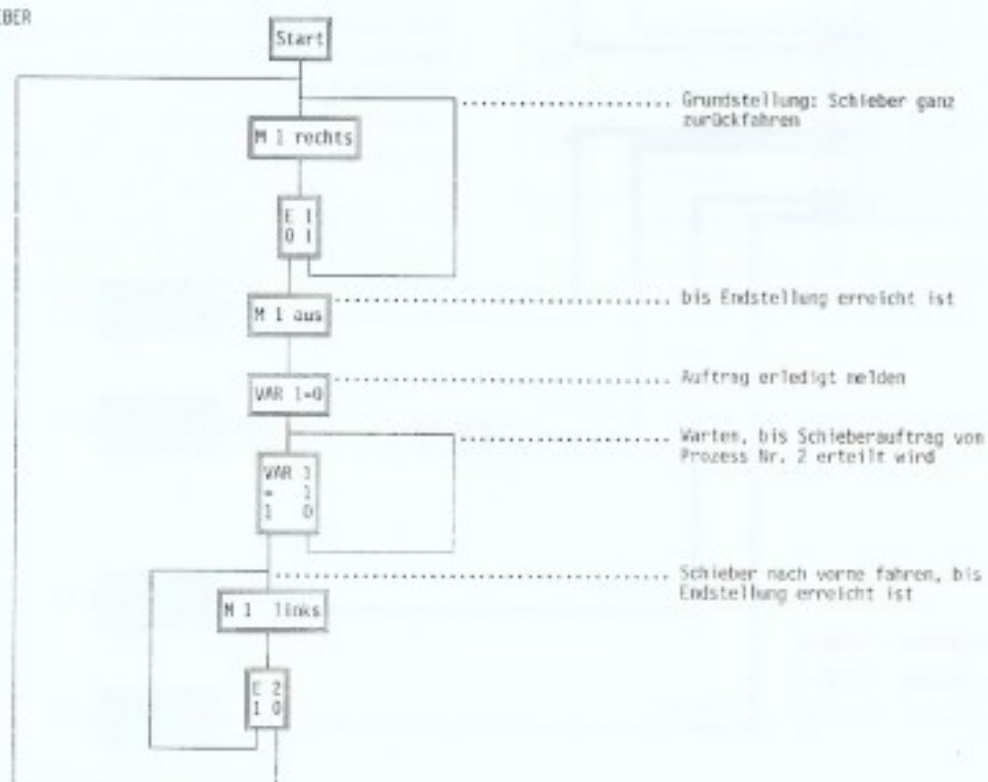
Der Prozeß 1 steuert den Schieber. Der Schieber wird erst ganz zurück gefahren und dann ganz nach vorne, wodurch eine Münze ausgegeben wird. Für die

Ausgabe mehrerer Münzen muß der Schieberauftrag entsprechend oft erteilt werden (im Beispiel wird immer nur eine Münze ausgegeben).

Der zweite Prozeß fährt den Schlitten vor die gewünschte Münzröhre. Im Beispiel werden nur die ersten beiden Röhren angefahren, den Weg zur dritten Röhre kannst Du sicher selbst programmieren, wenn Du das Ende-Symbol löschst und dort die Steuerung für die dritte Röhre einbaust. Der neue Steuerungsteil beginnt wieder mit dem Symbol VAR2=0 und endet mit dem Schieberauftrag.

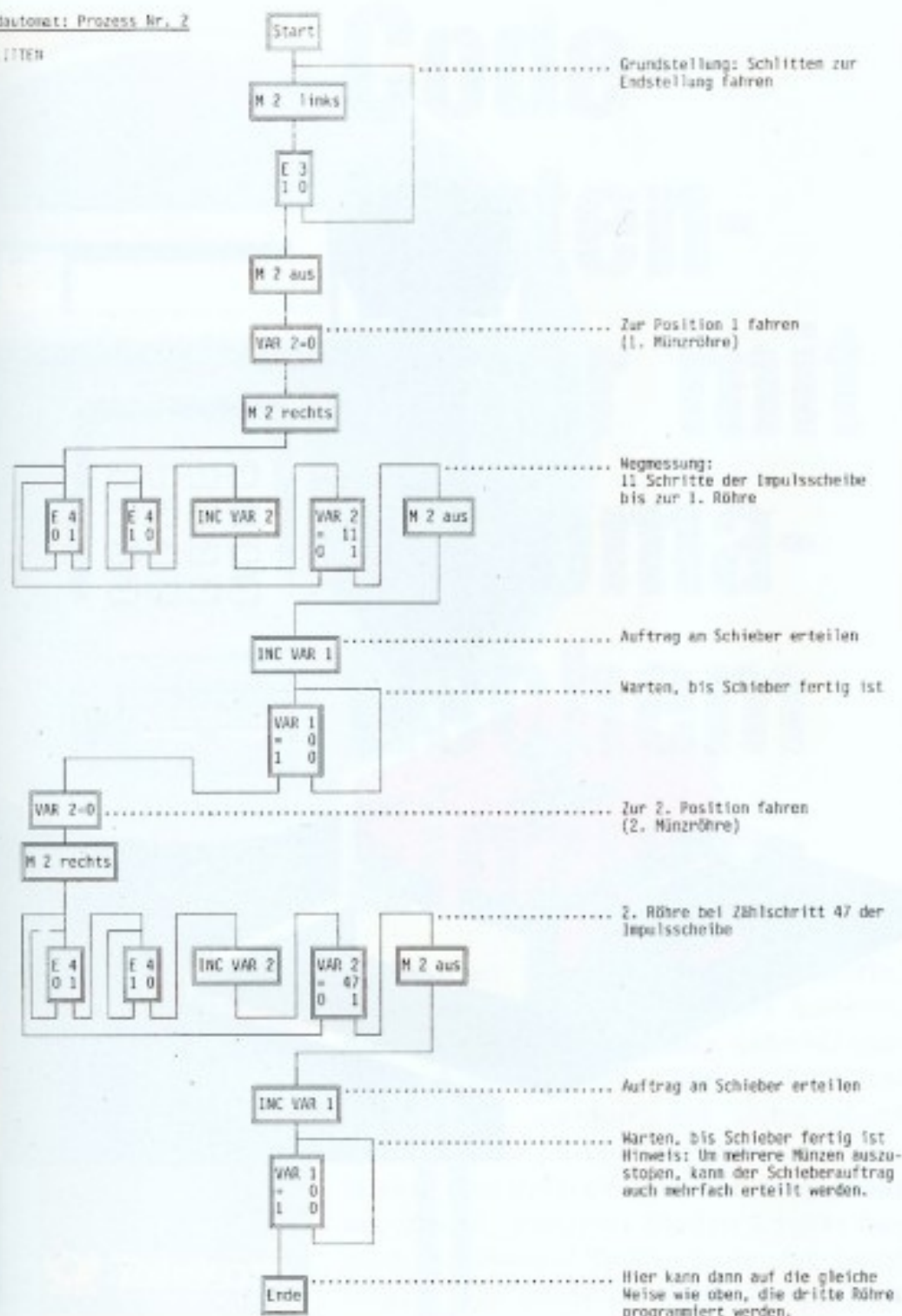
Geldautomat: Prozess Nr. 1

SCHIEBER



Geldautomat: Prozess Nr. 2

SCHITTEN





Code- karten- leser mit automa- tischem Einzug



Codekarten, wie z. B. die Scheckkarte, haben zwar alle die gleichen äußeren Abmessungen, haben aber unterschiedliche Arbeitsweisen. Die einfachsten Codekarten werden mit Infrarotlicht durchleuchtet, wobei im Inneren der Karte eine undurchsichtige Folie eingepreßt ist, die an manchen Stellen Löcher besitzt. An diesen Stellen kann das Licht ungehindert durch die Karte dringen und einen Schaltvorgang auslösen.

Die Codierung erfolgt durch die Lage und Anzahl der Löcher. Solche Karten findet man z. B. als Firmenausweis, der dann auch gleich zum Feststellen der Arbeitszeit dient.

Statt Infrarotlicht verwendet unser Codekartenleser sichtbares Licht. Der Vorteil ist, daß Du die Codekarten selber machen kannst. Die Codekarte ist so groß wie eine Scheckkarte (Bild 26), und sie wird automatisch eingezogen. Dabei wird der Code von zwei Lochreihen in den Computer eingelesen und ausgewertet. Zum Einlesen des Codes verwendet das Modell zwei Lichtschranken, für jede Lochreihe eine. Wie die Codekarte aussehen muß, kannst Du auf Seite 74 nachlesen.

Wenn Du die Karte so hältst, daß der Pfeil darauf von Dir weg weist, dann ist die linke Lochreihe für den „Takt“ und die rechte für die Daten zuständig. Die „Takt“-

Lochreihe ist nötig, damit der Codekartenleser erkennen kann, wann er in der rechten Reihe ein Loch abtasten kann (ob dann wirklich ein Loch da ist, hängt von der Codierung ab). In der linken Lochreihe werden also immer alle 5 Löcher gelocht. In der rechten Lochreihe erfolgt die Codierung (Loch oder kein Loch — je nach Programm).

Der Taster beim Schlitz des Kartenlesers (dort, wo die Karte hineingesteckt wird) wird so eingestellt, daß er beim Einführen der Karte betätigt wird. Der Motor zieht die Karte dann automatisch ein und gibt sie auch wieder aus.

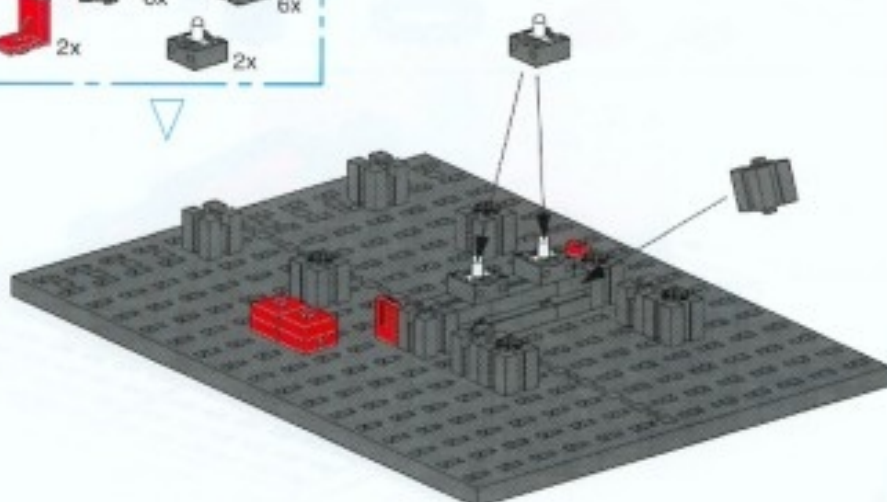
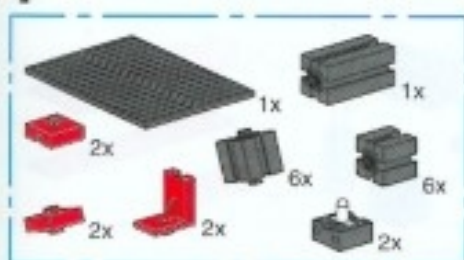
Für die Lichtschranken werden Kugellampen verwendet, da der Abstand zwischen Lichtsensor und Lampe recht gering ist. Bei Linsenlampen würde der Karton das Licht eventuell nicht mehr ganz abhalten — und so könnte der Code nicht mehr sicher erkannt werden.

modell 92

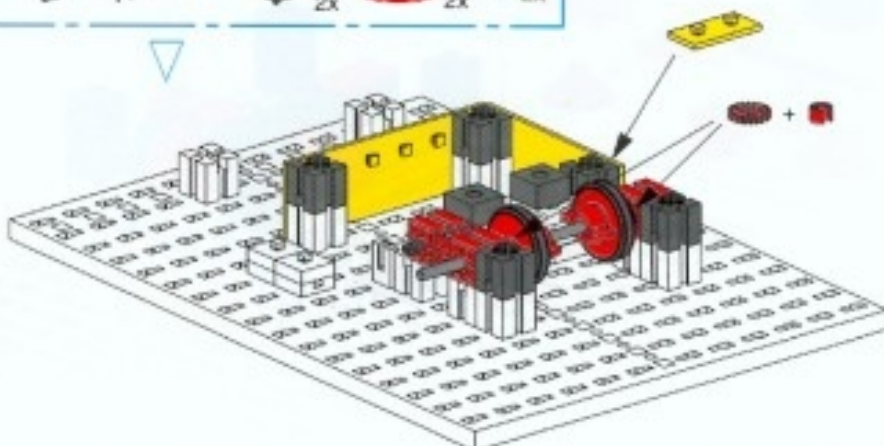
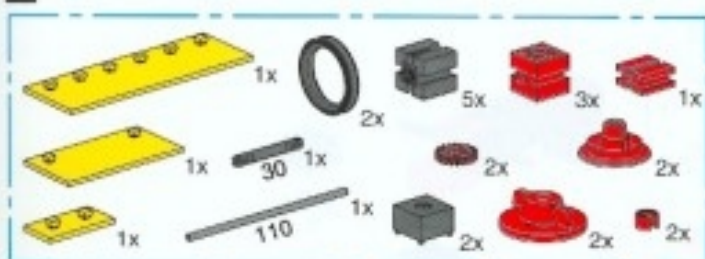
1

30mm

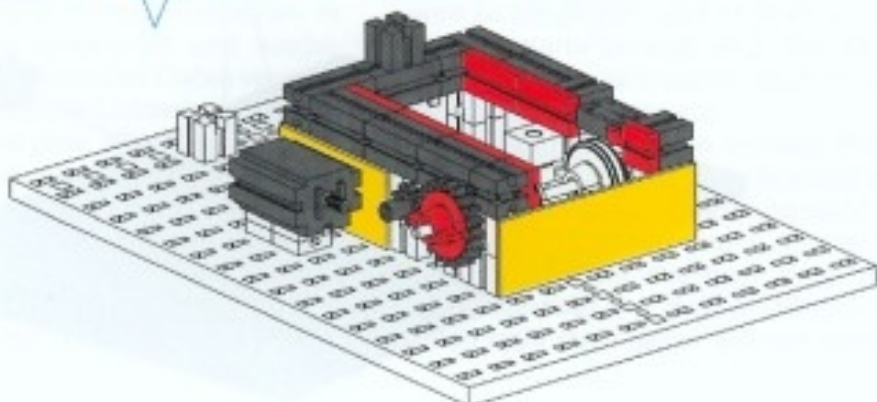
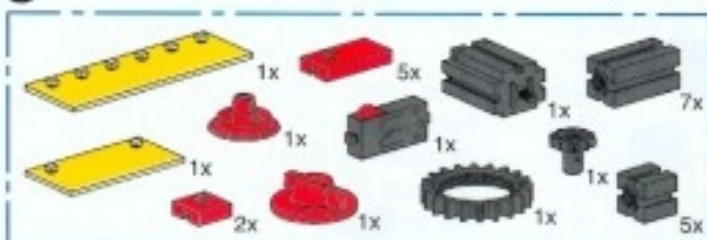
110mm



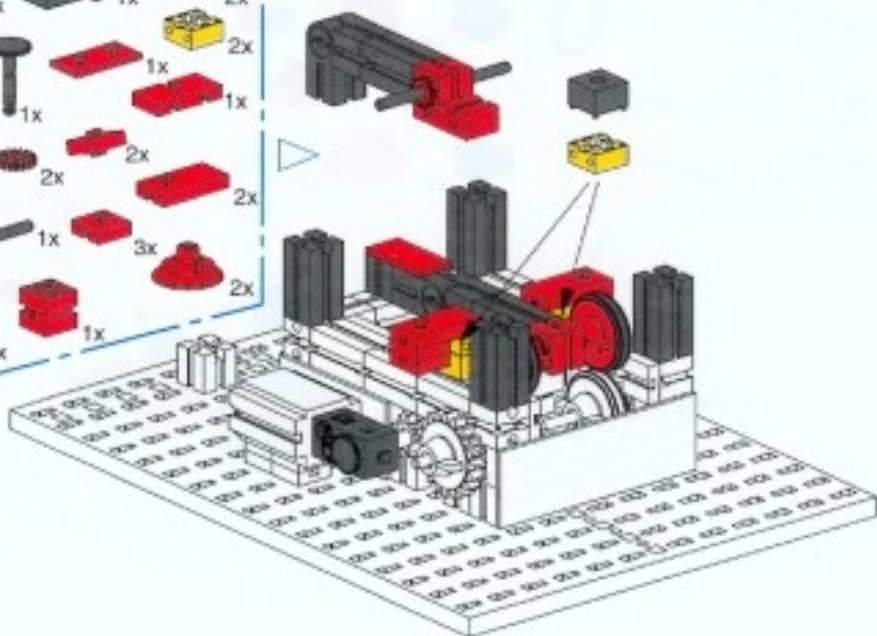
2



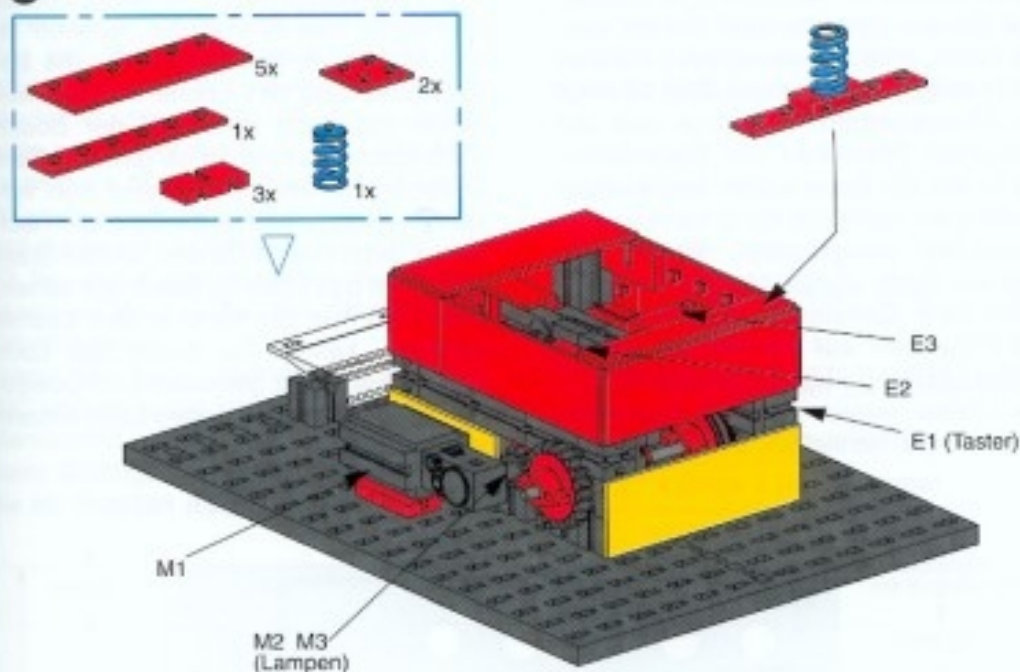
3



4



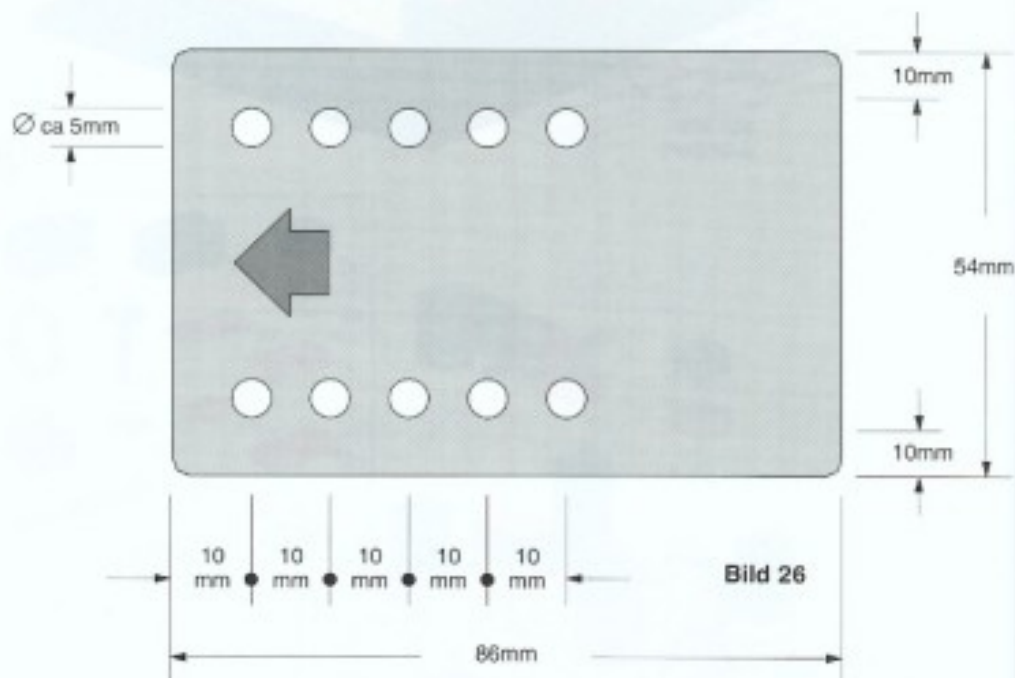
5



Die Codekarte

Die Codekarte für das Lesegerät schneidest Du aus mittelstarkem Karton aus. Der Karton muß fest genug sein, um den Taster betätigen zu können. **Bild 26** zeigt die Abmessungen der Karte und die möglichen Positionen der Codelöcher. Die Ecken der Karte sollten leicht abgerundet sein, damit die Karte nicht an den Bausteinen hängenbleibt. Welche Löcher als Code ausgeschnitten werden, bleibt Dein Geheimnis. Natürlich muß das Programm auf die von Dir ausgesuchte Lochung abgestimmt werden. Die Löcher lassen sich mit folgenden Trick leichter herstellen: Zuerst wird die

Codekarte ausgeschnitten und die Löcher mit schwarzem Filzstift aufgemalt. Dann nimmst Du einen normalen Locher und leerst erst einmal die „Konfetti“ in den Mülleimer. Nun kannst Du die Bodenabdeckung des Lochers abnehmen (wenn das nicht geht und der Boden auch nicht durchsichtig ist, geht der Trick leider nicht). Der Locher wird nun verkehrt herum gehalten (Bodenseite nach oben). Jetzt kannst Du den runden Stanzer durch das Loch im Blech gut sehen. Wenn Du nun die Karte in den Locher schiebst, kannst Du durch das Loch auch die schwarz gemalten Lochpositionen erkennen und so jedes Loch einzeln stanzen.



Verdrahtung

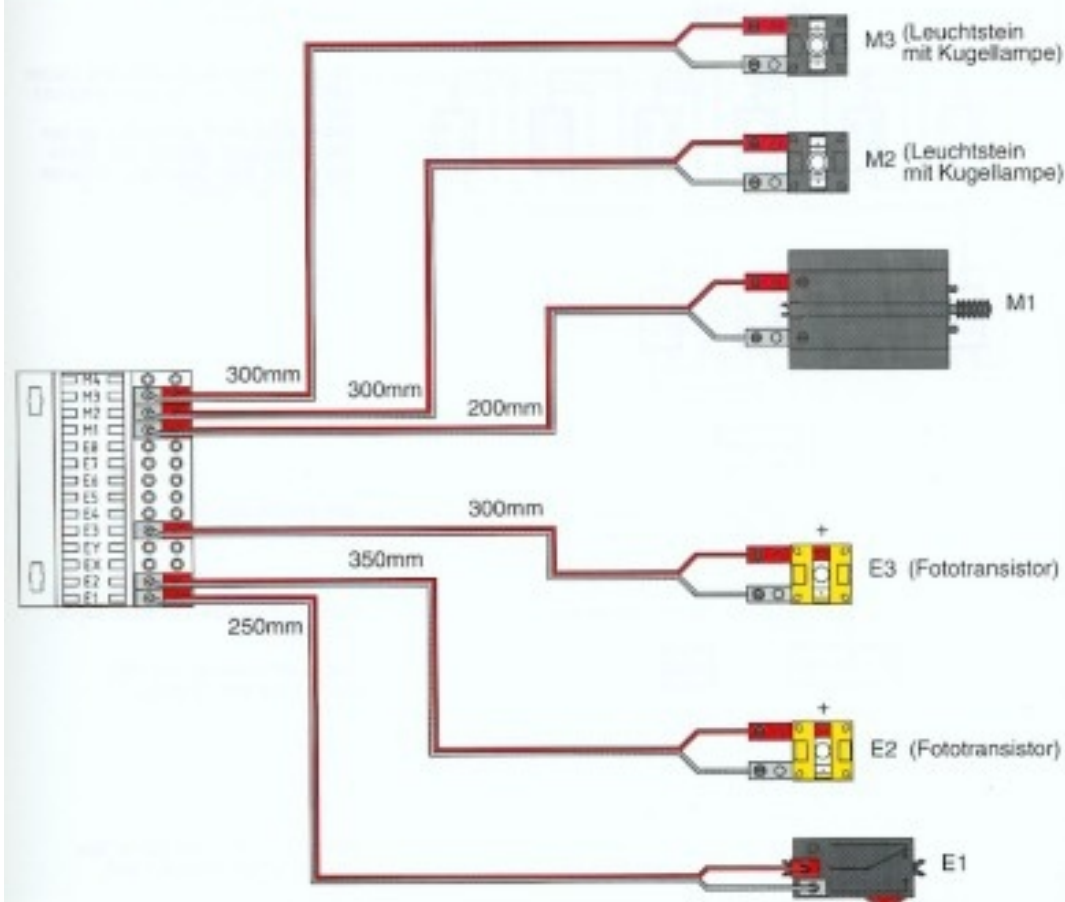
Bei der Verdrahtung mußt Du darauf achten, daß die Kabel nicht im Kartenschacht liegen. Am besten führst Du außer der Motorzuleitung alle Kabel an der Seite zu, auf der der Taster liegt. Der Motor wird mit M1, die Lampen mit M2 und M3 verbunden. Der Taster kommt an E1 und die beiden Lichtsensoren an E2 und E3. Bei den Fototransistoren die Polung beachten! Achte auch auf die richtigen Kabellängen.

Starte nun das Programm LUCKY LOGIC und teste die Verdrahtung mit dem Menüpunkt OPTIONEN DIAGNOSE. Beim Betätigen der Taster müssen sich die im Testplan aufgeführten Eingangs-

Anzeigen entsprechend verändern. Nun schalte die M-Ausgänge (Motoren oder Lampen) durch Anklicken mit der Maus, wie es in der Beschreibung zur Software erklärt ist. Wenn alles klappt, kann die Software geladen und gestartet werden.

Testplan:

E1	Taster am Einleseschlitz
E2	Lichtsensor Lochreihe 1 (rechts)
E3	Lichtsensor Lochreihe 2 (links)
M1	Antriebsmotor für Karteneinzug
M2	Lampe 1 (Kugellampe)
M3	Lampe 2 (Kugellampe)



Software

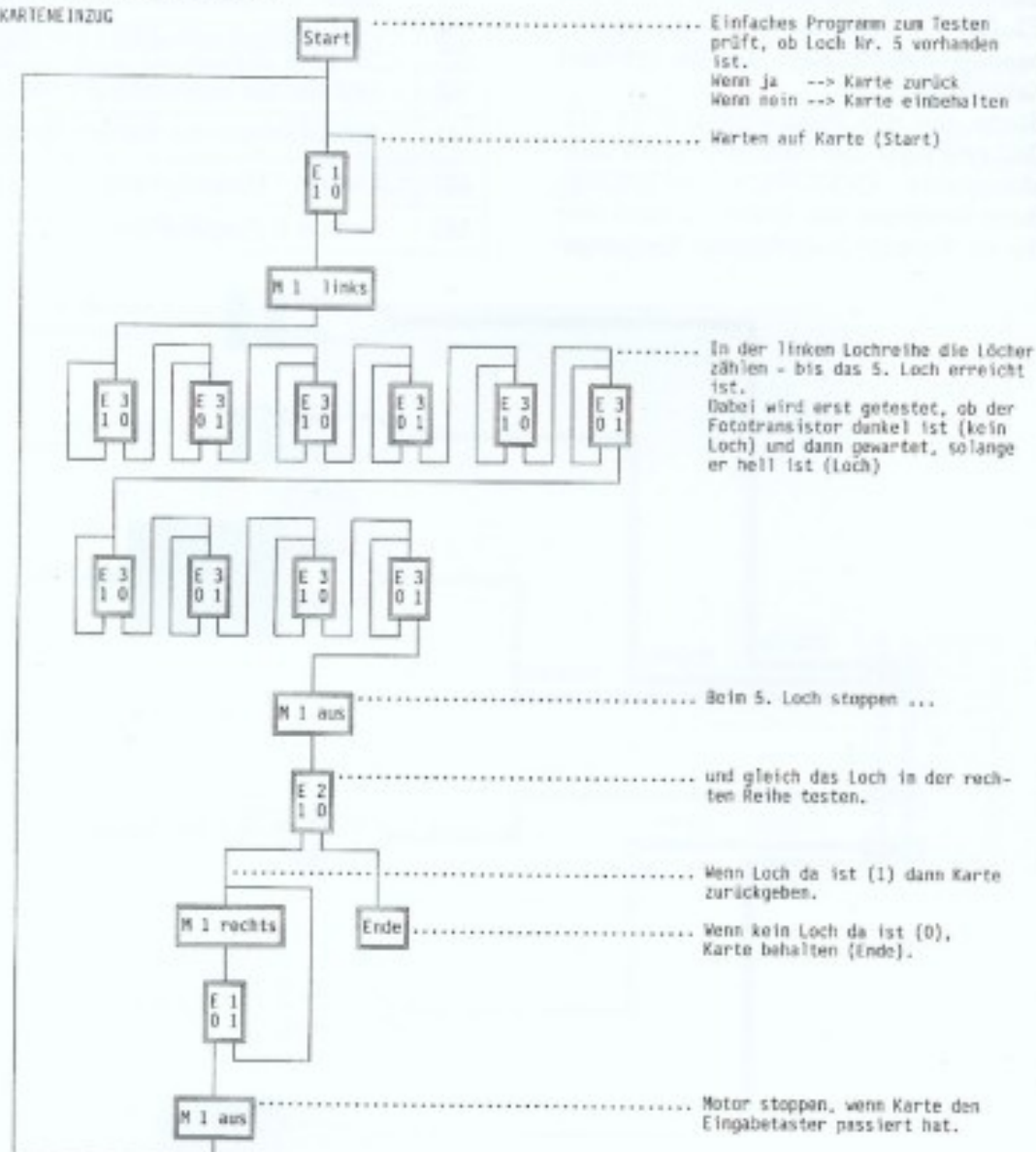
Die Software für den Codekartenleser ist mit LUCKY LOGIC erstellt. Sie besteht aus den Dateien CODE.FTG, CODE1.FTL,

CODE2.FTL und CODE3.FTL. Lade den Gesamtprozeß CODE und starte das Programm mit dem Menüpunkt RUN GESAMTPROZESS.

Im Beispielpogramm wird nur das fünfte Loch ausgewertet (damit es nicht zu un-

Codekartenleser: Prozess Nr. 1

KARTENEINZUG



übersichtlich wird). Wenn alle Löcher getestet werden sollen, muß immer dann die rechte Reihe (E2) geprüft werden, wenn in der linken Reihe (E3) ein Loch da ist. Wenn der richtige Code erkannt wurde, gibt der Leser die Karte zurück.

Stimmt der Code auf der Karte nicht mit dem Steuerprogramm überein, wird die Karte einbehalten. Du kannst die Karte dann nur über den Diagnoseteil von LUCKY LOGIC „herausfahren“.

Codekartenleser: Prozess Nr.2

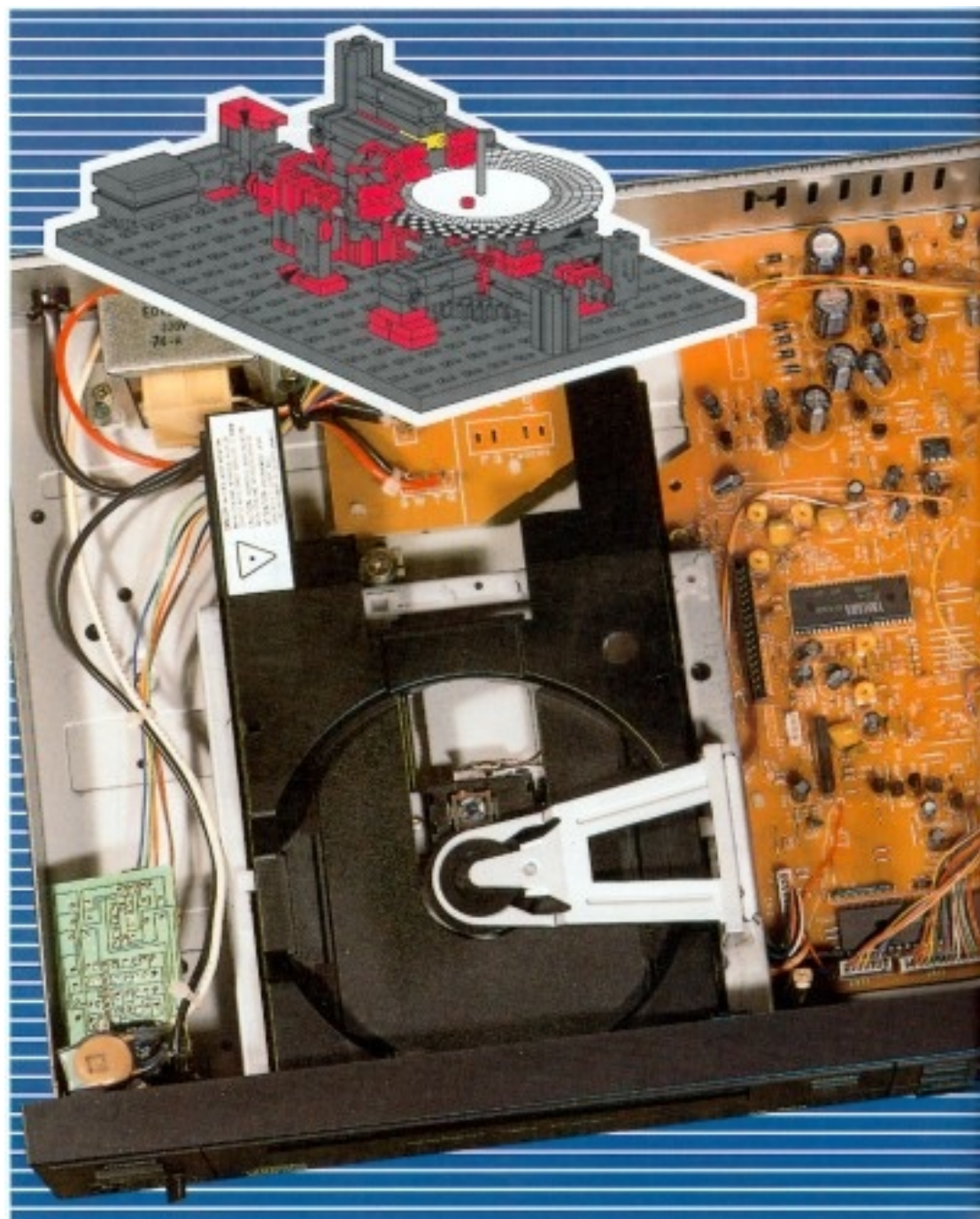
LAMPPE



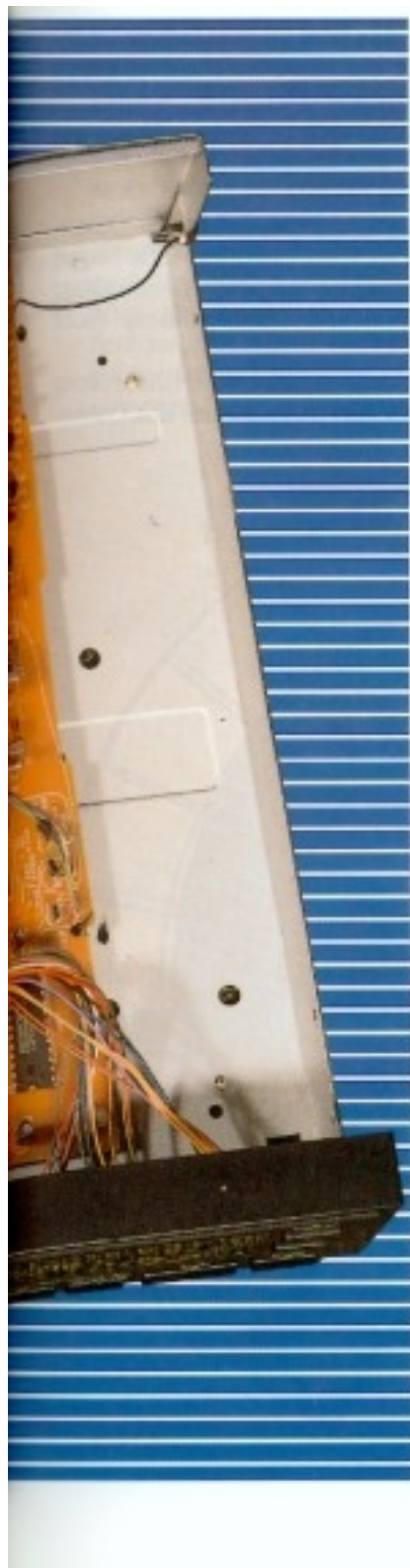
Codekartenleser: Prozess Nr.3

LAMPPE





CD- Player



Der CD-Player kann Daten lesen, die Du als helle und dunkle Felder in einem vorgegebenen Muster auf einer Papierscheibe eintragen kannst. Die Papierscheibe hat drei verschiedene Spuren. Weiter hinten ist die Vorlage der Scheibe abgebildet. Damit Du das Buch nicht zerschneiden mußt, kannst Du die Vorlage fotokopieren oder mit Zirkel und Lineal auf ein weißes Papier übertragen. Die Datenfelder werden sorgfältig mit einem schwarzen Filzstift ausgemalt (Vorsicht! Nicht über den Rand hinausmalen!). Dann schneidest Du die beiden Löcher für die Drehachse (in der Mitte, 4 mm Durchmesser) und die Rastachse (17 mm seitlich) aus.

Achte darauf, daß das Loch für die Drehachse genau in der Mitte liegt, sonst funktioniert das Lesen nicht richtig.

Zum Einlegen der Datenscheibe wird das Oberteil des Lesekopfs aufgeklappt und die Datenscheibe auf die Achse mit dem Zahnrad Z 30 gesteckt. Mit der Rastachse wird die Scheibe in einem der Löcher des Zahnrades fixiert. Diese Rastachse dient auch als Schaltknocken für den Referenztaster.

Bild 27 zeigt eine Codescheibe zum Fotokopieren (oder abzeichnen) und selbst

ausmalen, das **Bild 28** eine fertige Codescheibe für die Tonreihe 0 bis 15 (auch zum Kopieren).

Beim Aufbau mußt Du einige Dinge beachten:

- Das Zahnrad Z 30 unter der Datenscheibe muß auf gleicher Höhe liegen wie das Unterteil des Lesekopfes; sonst läuft die Datenscheibe nicht richtig.
- Der Referenztaster an E1, der vom Schaltknocken (Rastachse) der Daten-

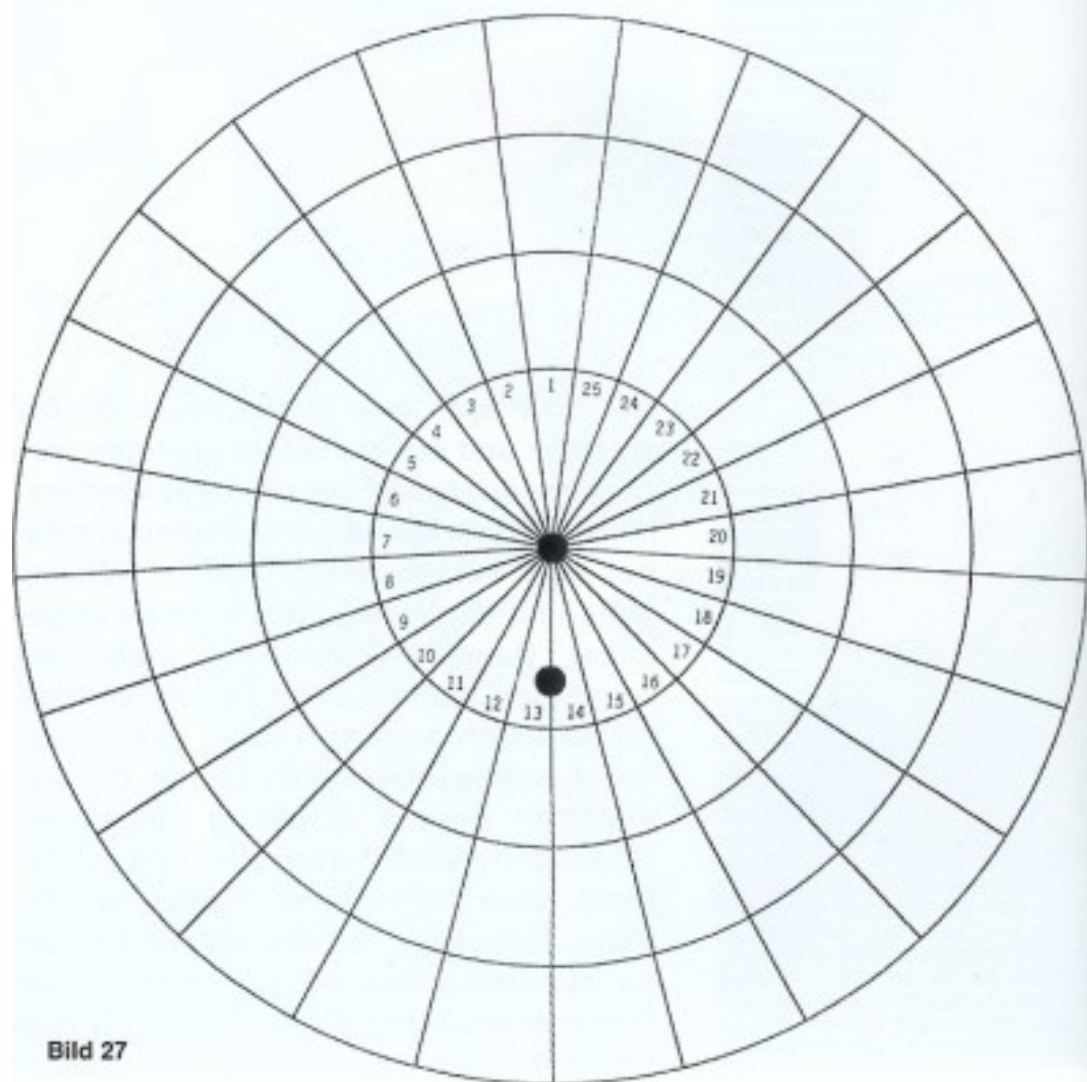


Bild 27

scheibe betätigt wird, muß so verschoben werden, daß er einwandfrei betätigt wird (Rastachse zwischen den Segmenten 13 und 14).

Von der Lichtdurchlässigkeit der Datenscheibe hängt es ab, ob die Leseeinrichtung mit einer Kugellampe oder einer Linse Lampe bestückt werden muß. Bei normalem Kopierpapier genügt die Kugellampe. Bei weniger lichtdurchlässigem Papier (z. B. Recyclingpapier) muß die Linse Lampe eingesetzt und unter den Leuchtstein

zusätzlich ein Baustein 5 eingebaut werden.

- Der Schlitten mit dem Lesekopf muß sich frei bewegen können. Das testest Du, indem Du das Zahnrad mit Rastachse etwa 4 mm herausziehst. Jetzt kannst Du die Schnecke mit der Hand drehen. Die Kabel bitte so verlegen, daß sie keine beweglichen Teile des Modells berühren oder zwischen Schnecke und Zahnräder geraten können.

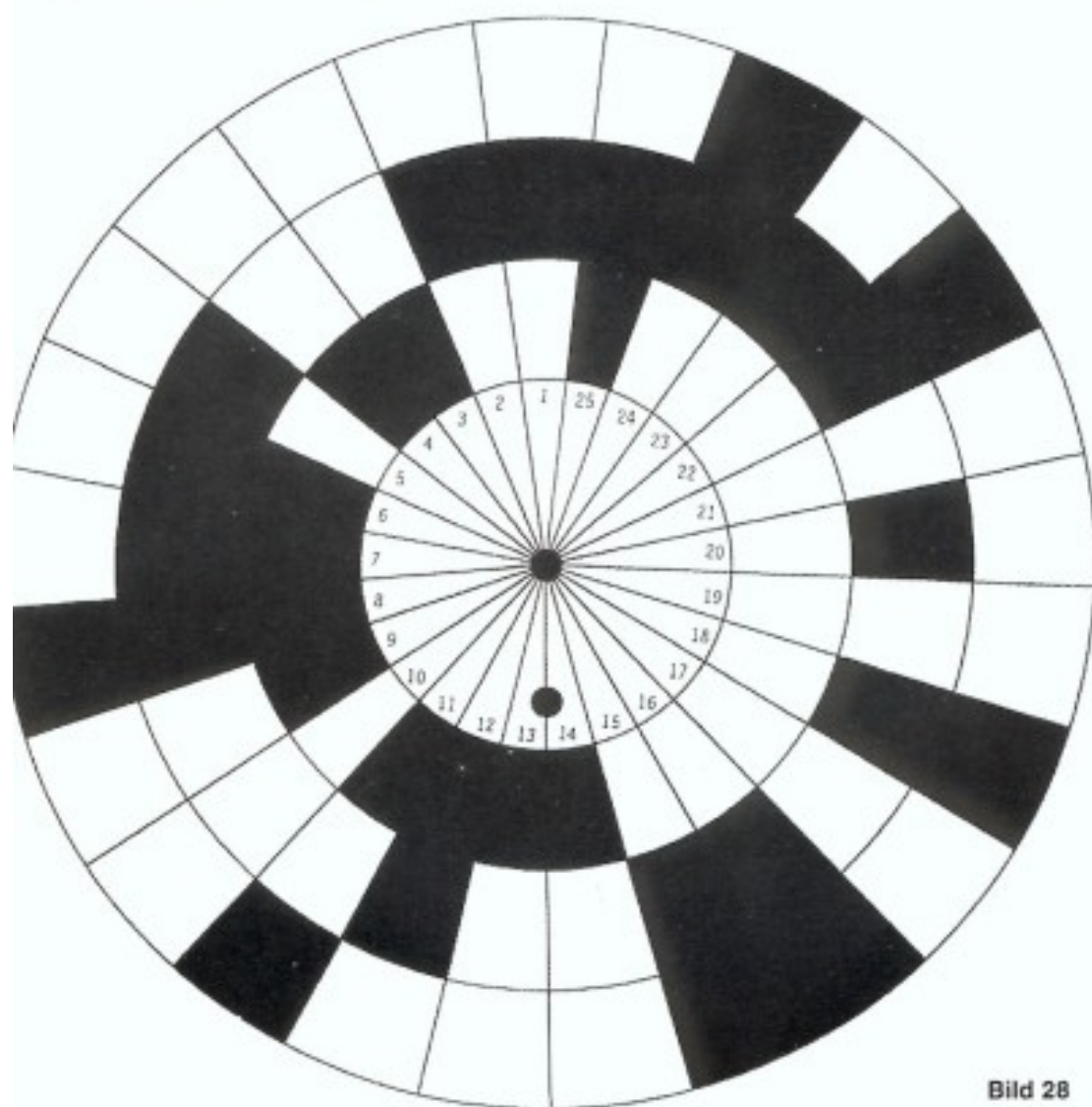
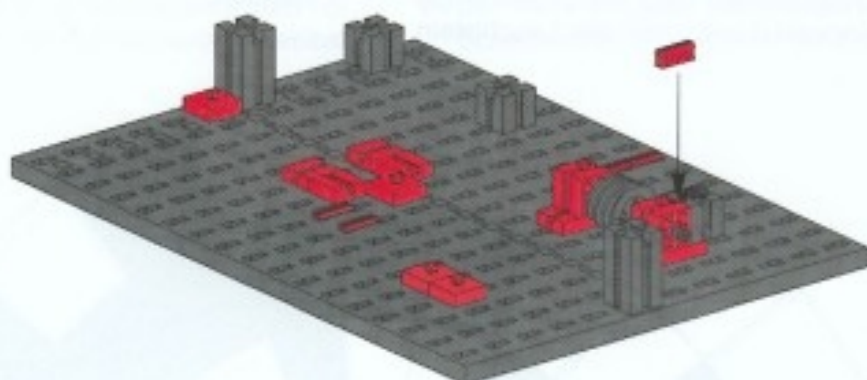
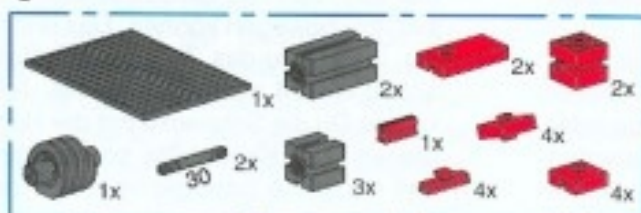
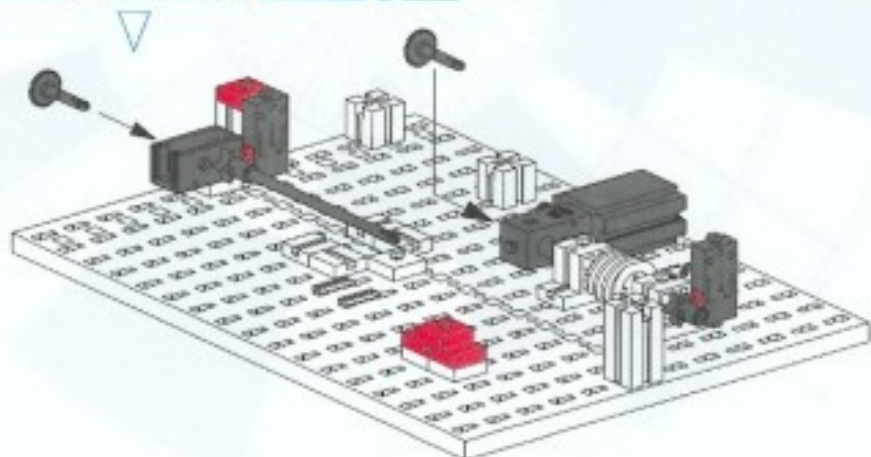
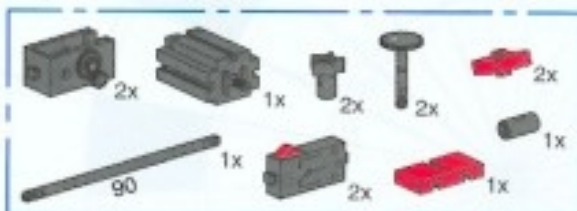


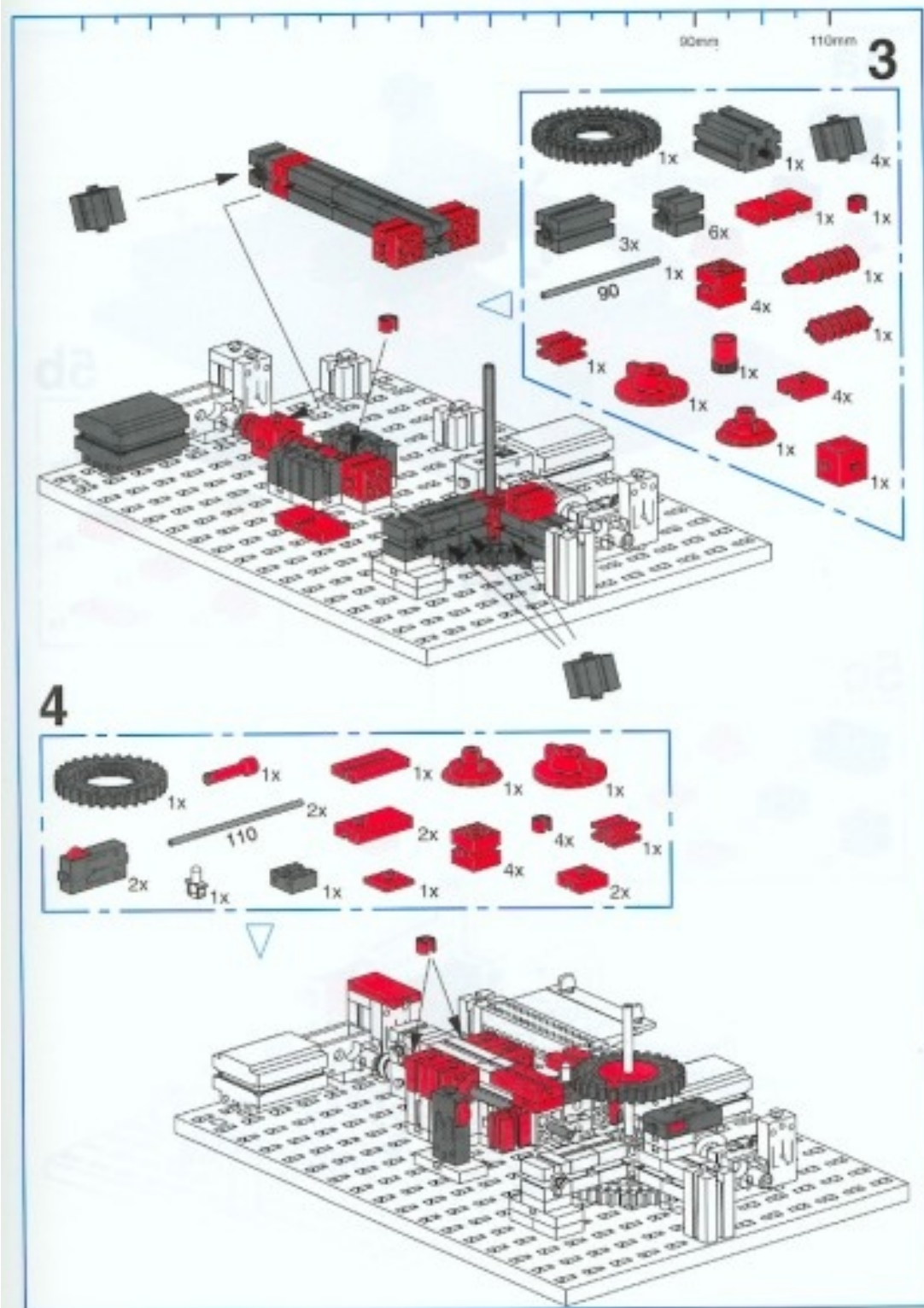
Bild 28

1

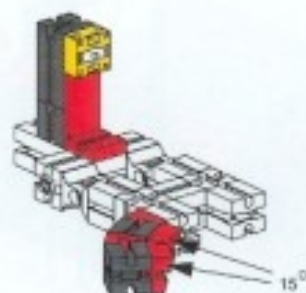
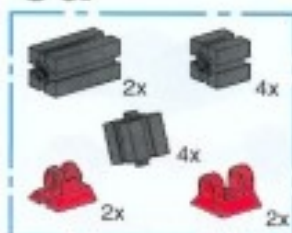


2

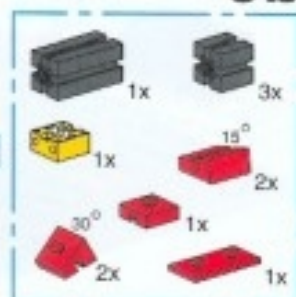




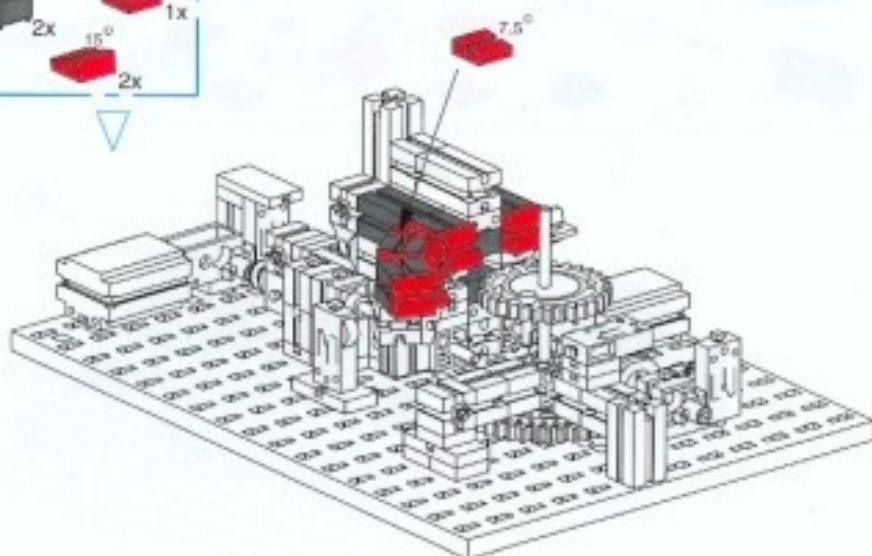
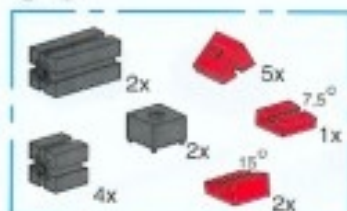
5a

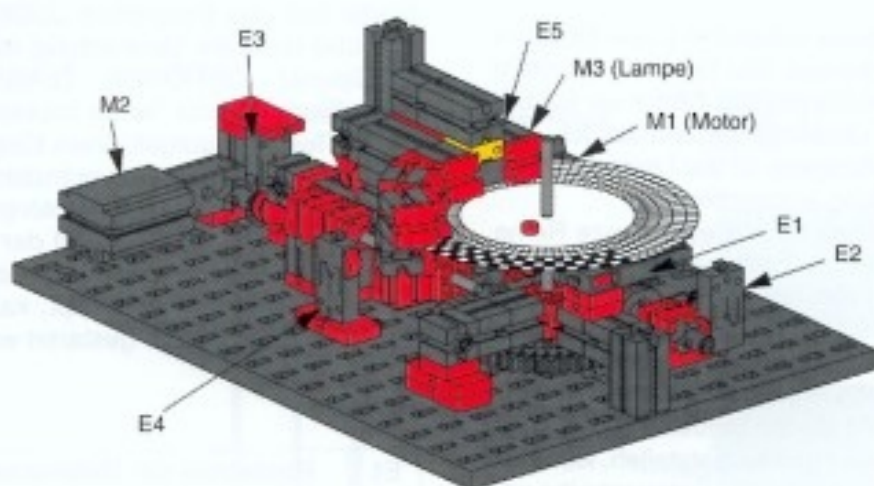


5b



5c





Verdrahtung

Der CD-Player arbeitet mit zwei Motoren und einer Lampe. Der Motor an M1 dreht die Datenscheibe, der Motor an E2 bewegt den Lesekopf vor und zurück. Am Interface-Ausgang ist die Lampe der Leseeinrichtung angeschlossen.

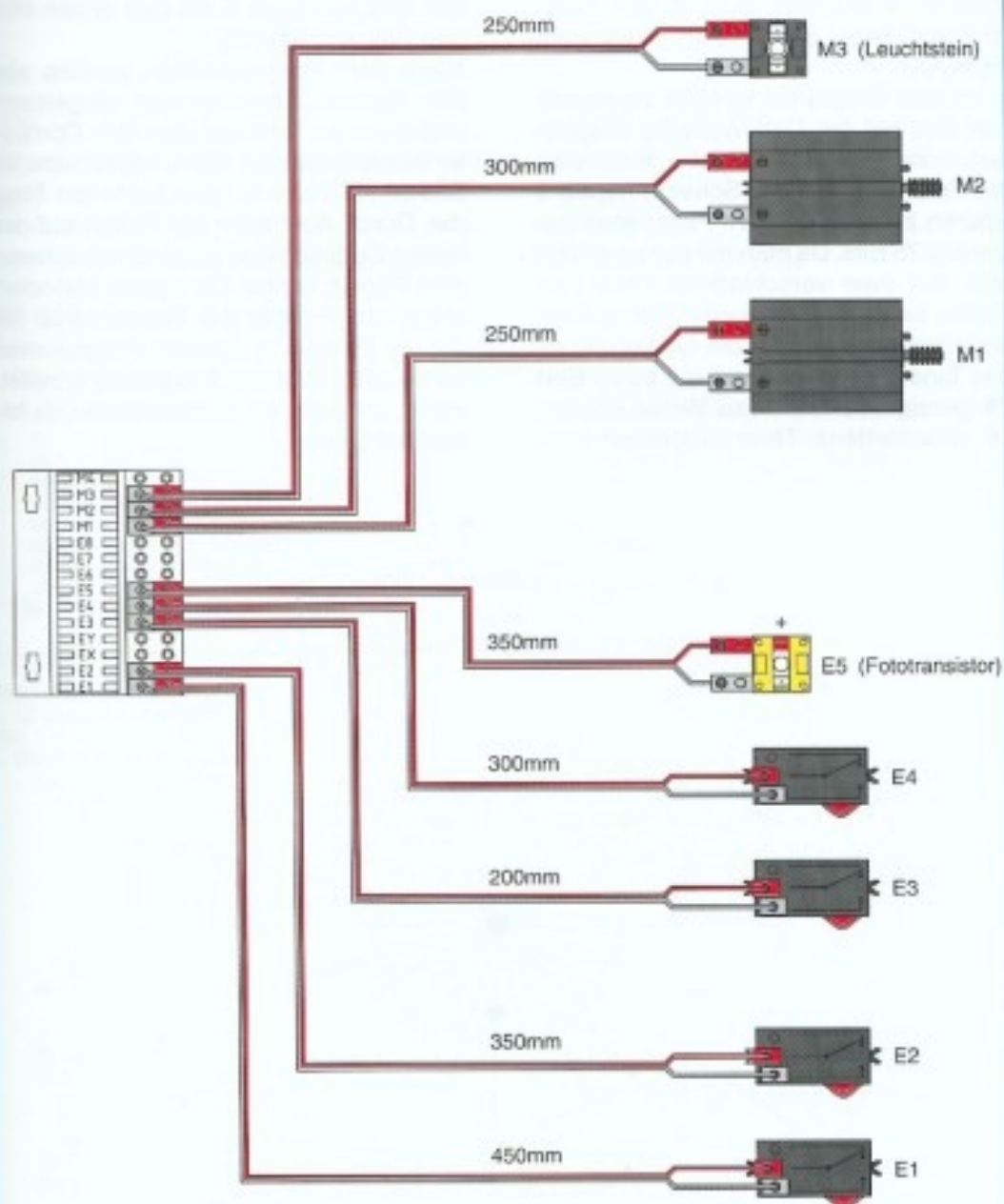
Der CD-Player besitzt eine ganze Reihe von Tastern als Rückmelder. Der Taster an E1 wird vom Schaltnocken der Datenscheibe betätigt. Der Taster an E2 wird von der Impulsscheibe des Datenscheiben-Antriebs regelmäßig ein- und ausgeschaltet. Mit diesen beiden Tastern kann der Computer genau feststellen, welches Feld auf einer Spur der Datenscheibe gerade gelesen wird. Das Feld 1 wird durch den Taster an E1 festgestellt. Danach wird bei jeder Betätigung des Tasters an E2 um eins weitergezählt. Der Taster an E3 wird durch die Impulsscheibe am Lesekopf-Motor betätigt und ermöglicht es dem Computer, die Spur auf der Datenscheibe auszuwählen. Der Taster an E4 ermittelt die Endstellung des Lesekopfes (ganz herausgefahren). Beim Programmstart wird daher der Lesekopf zuerst in die Endstellung gefahren. An E5 ist schließlich der Lichtsensor für das Lesen der Daten angeschlossen. Beim Fo-

totransistor die Polung beachten! Achte auch auf die richtigen Kabellängen.

Starte nun das Programm LUCKY LOGIC und teste die Verdrahtung mit dem Menüpunkt OPTIONEN DIAGNOSE. Beim Betätigen der Taster müssen sich die im Testplan aufgeführten Eingangs-Anzeigen entsprechend verändern. Nun schalte die M-Ausgänge (Motoren oder Lampen) durch Anklicken mit der Maus, wie es in der Beschreibung zur Software erklärt ist. Wenn alles klappt, kann die Software geladen und gestartet werden.

Testplan:

E1	Rastachse der Datenscheibe (Feld 0)
E2	Impulsscheibe am Datenscheiben-Antrieb
E3	Impulsscheibe am Lesekopf-Antrieb
E4	Endstellung Lesekopf (ganz außen)
E5	Lichtsensor (Lesedaten)
M1	Datenscheiben-Antrieb
M2	Lesekopf-Antrieb
M3	Lampe der Leseeinrichtung



Software

Die Software zu diesem Modell ist in Turbo-Pascal geschrieben. Dieses Pascal-Programm befindet sich in der Datei READCD.PAS. Starte das Programm READCD.

In diesem Programm werden die einzelnen Bits auf der Datenscheibe eingelesen, wobei gilt: weißes Feld = 0, schwarzes Feld = 1. Auf der Scheibe haben 3 Spuren zu je 25 Feldern Platz, also insgesamt 75 Bits. Da man mit nur einem Bit aber nur zwei verschiedene Töne darstellen kann, werden immer vier aufeinanderfolgende Bits für die Codierung eines Tons verwendet — so wie es im **Bild 29** gezeigt ist. Auf diese Weise können 16 verschiedene Töne dargestellt wer-

den. Die Codierung setzt sich über die Spuren fort (das letzte Bit von Spur 1 ergibt mit den ersten 3 Bits der zweiten Spur einen Ton; ebenso die letzten beiden Bits von Spur 2 mit den ersten beiden Bits von Spur 3).

Nach dem Programmstart werden alle drei Spuren nacheinander eingelesen und dann als Tonfolge über den Computer wiedergegeben. Die Codescheibe im Bild 28 stellt alle 16 verschiedenen Töne dar. Durch Ausmalen der Felder auf der leeren Codescheibe (Kopie!) mit schwarzem Filzstift kannst Du eigene Melodien speichern — oder die Codescheibe für andere Zwecke in eigenen Programmen verwenden. Wie das Programm arbeitet, siehst Du aus den Kommentaren im folgenden Listing.

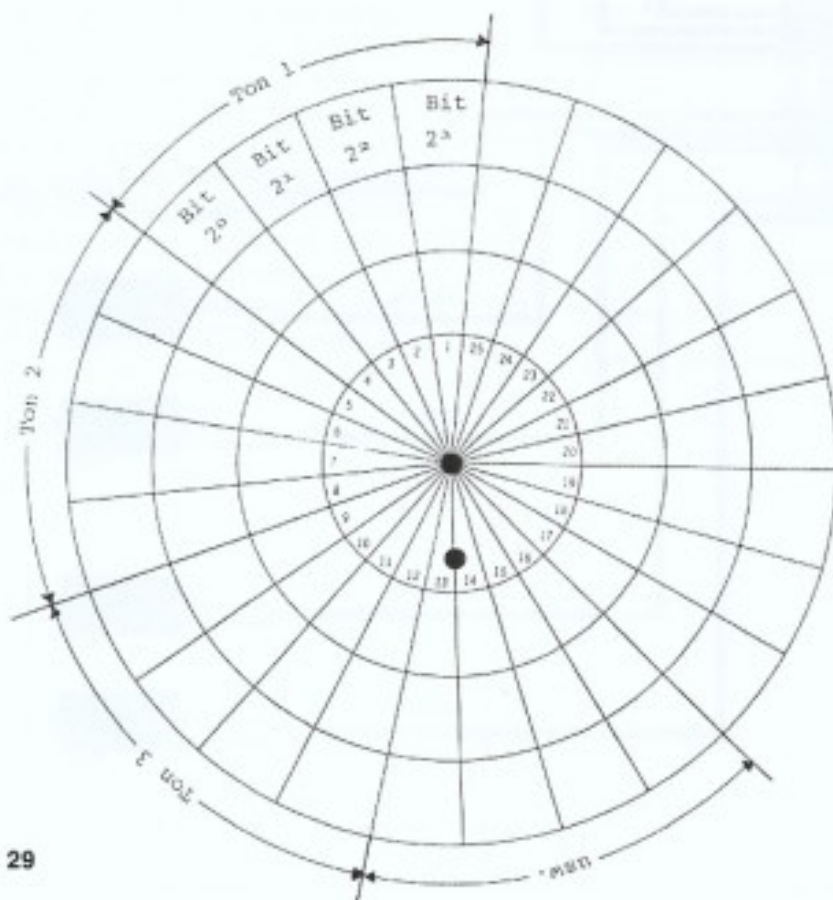


Bild 29

```

PROGRAM READCD; { Turbo Pascal 5.0 - 6.0, Datei: READCD.PAS }
{.....}
{ Programm für das Modell "CD-PLAYER" }
{ Es gibt 3 Datenspuren zu je 25 Segmenten (Feldern) = 75 Datenbits }
{ Zuerst werden alle Daten eingelesen und dann aus jeweils 4 aufeinanderfol- }
{ genden Bits ein Ton zusammengesetzt. Es gibt also 16 verschiedene Töne. }
{ Zum Schluß werden die Töne am Computer "abgespielt" }
{.....}

USES CRT, FISCHER;

CONST Max_Felder = 75; { Anzahl Datenbits }

TYPE t_Datenfeld = ARRAY [1..Max_Felder] OF BOOLEAN; { Typ für Datenfeld }

VAR i : INTEGER; { Zähler }
    Datenfeld : t_Datenfeld; { Datenfeld }

{.....}

PROCEDURE CD_Player_initialisieren;
{ CD-Player in den Grundzustand bringen und starten }
VAR i : BYTE;
BEGIN { CD-Player_initialisieren }
    Motor(3,Rechts); { Lampe für Leser einschalten }
    Motor(1,Rechts); { Datenscheiben-Antrieb einschalten }
    Motor(2,Links); { Lesekopf zur Endstellung fahren: }
    REPEAT UNTIL NOT Taster(4); { erst herausfahren, bis Taster frei ist, }
    Motor(2,Rechts); { dann zurück, bis Taster gerade schließt }
    REPEAT UNTIL Taster(4);
    Motor(2,Aus);
    REPEAT UNTIL NOT Taster(1); { Warten auf Feld 0 (Rastachse) }
    REPEAT UNTIL Taster(1);
    FOR i := 1 TO Max_Felder DO Datenfeld[i] := FALSE; { Datenfeld löschen }
END; { CD_Player_initialisieren }

{.....}

PROCEDURE Scheibe_einlesen;
{ CD komplett einlesen }
VAR i,C,Y : BYTE;
BEGIN { Scheibe einlesen }
    FOR i := 1 TO 25 DO { 1. Spur einlesen }
        BEGIN
            Datenfeld[i] := NOT Taster(5); { dunkles Feld = TRUE }
            C := 0;
            REPEAT { bis zum nächsten Feld 8 Impulse }
                REPEAT UNTIL Taster(2); DELAY(2); { am Datenscheibenantrieb abwarten }
                REPEAT UNTIL NOT Taster(2); DELAY(2);
            INC(C);
            UNTIL C = 8;
        END;
        Motor(1,Aus); { Scheibe nach dem letzten Feld stoppen }

        Y := 0; { Auf die zweite Spur positionieren }
        Motor(2,Links); { Lesekopf-Antrieb }
        REPEAT
            REPEAT UNTIL Taster(3); { 26 Impulse am Lesekopf-Antrieb }
            REPEAT UNTIL NOT Taster(3);
            INC(Y);
        UNTIL Y = 26;
        Motor(2,Aus); { 2. Spur erreicht }

        Motor(1,Rechts); { Datenscheibe wieder starten - erstes Feld }
        FOR i := 26 TO 50 DO { 2. Spur einlesen (genauso, wie bei 1. Spur) }
            BEGIN
                Datenfeld[i] := NOT Taster(5);
                C := 0;
            END;
    END;

```

```

    REPEAT
      REPEAT UNTIL Taster(2); DELAY(2);
      REPEAT UNTIL NOT Taster(2); DELAY(2);
      INC(C);
    UNTIL C = 8;
  END;
  Motor(1,Aus); { Datenscheibe wieder stoppen }

  Y := 0; { Lesekopf auf 3. Spur positionieren }
  Motor(2,Links); { wie oben }
  REPEAT
    REPEAT UNTIL Taster(3);
    REPEAT UNTIL NOT Taster(3);
    INC(Y);
  UNTIL Y = 26;
  Motor(2,Aus);

  Motor(1,Rechts); { 3. Spur einlesen (wie oben) }
  FOR i := 51 TO 75 DO
    BEGIN
      Datenfeld[i] := NOT Taster(5);
      C := 0;
      REPEAT
        REPEAT UNTIL Taster(2); DELAY(2);
        REPEAT UNTIL NOT Taster(2); DELAY(2);
        INC(C);
      UNTIL C = 8;
    END;
    Motor(1,Aus);
    Motor(3,Aus);
    Motor(2,Rechts);
    REPEAT UNTIL Taster(4);
    Motor(2,Aus);
  END; { Scheibe_einlesen }

  { ..... }

PROCEDURE Tonfolge_erzeugen;
{ Aus den Daten Töne produzieren }
VAR i;
    Frequenz : INTEGER;
BEGIN { Tonfolge_erzeugen }
  i := 1;
  REPEAT
    Frequenz := 0;
    IF Datenfeld[i] THEN Frequenz := Frequenz + 2*2*2; { Tonhöhe aus 4 Bits zusammensetzen }
    IF Datenfeld[i+1] THEN Frequenz := Frequenz + 2*2; { 2^2 }
    IF Datenfeld[i+2] THEN Frequenz := Frequenz + 2; { 2^1 }
    IF Datenfeld[i+3] THEN Frequenz := Frequenz + 1; { 2^0 }
    SOUND(Frequenz * 100); { Ton spielen }
    DELAY(600); { 0,6 Sekunden lang }
    NOSOUND;
    INC(i,4); { weiter mit nächstem Ton }
  UNTIL i > 72;
END; { Tonfolge_erzeugen }

  { ..... }

BEGIN { Programm READCD }
  CLRSCR;
  CD Player initialisieren;
  Scheibe_einlesen;
  Tonfolge_erzeugen;
END. { Programm READCD }

```


So funktioniert die CD

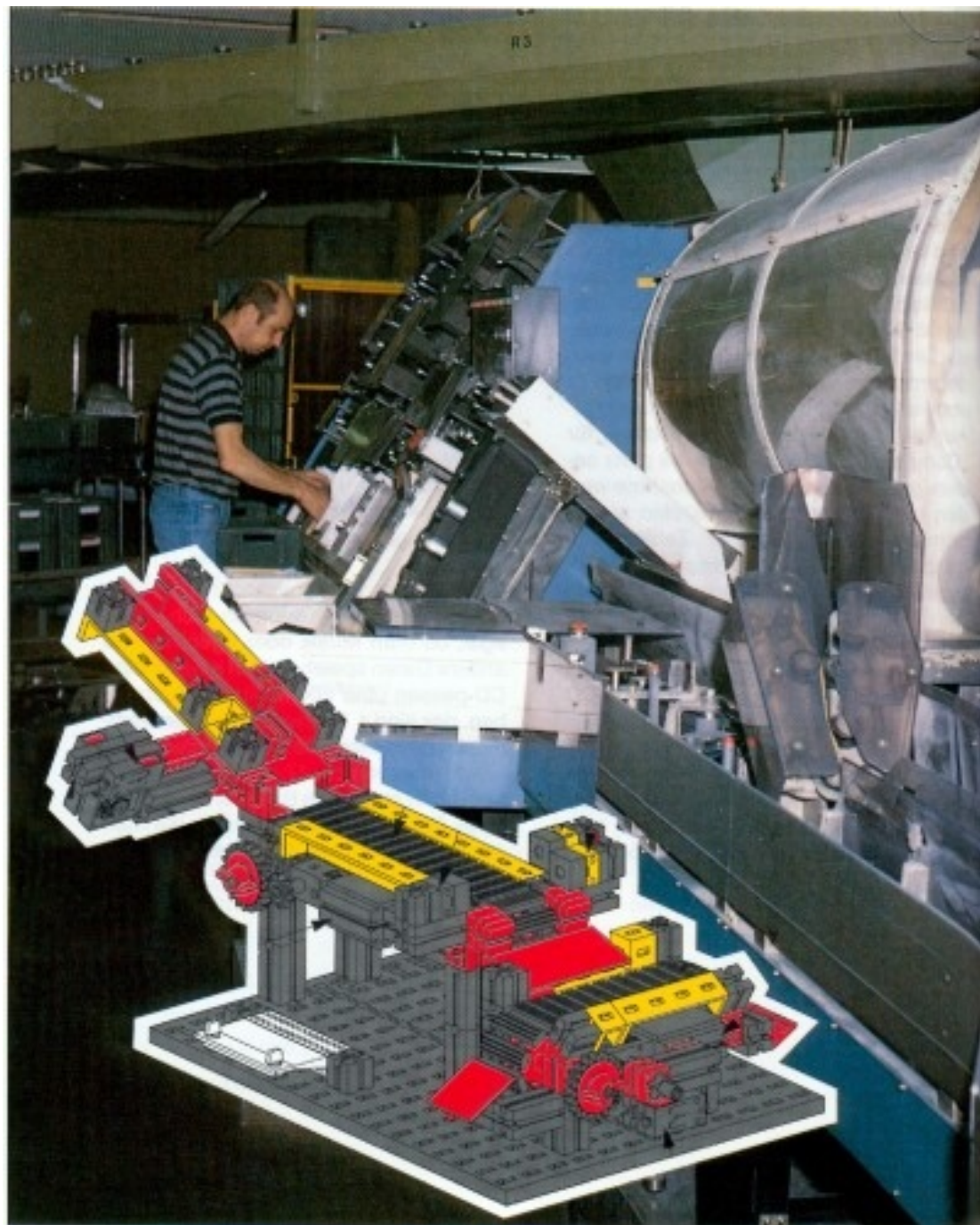
Sicher kennst Du schon die glänzenden Musik-CDs, bei denen die Musik ganz anders aufgezeichnet ist als bei den herkömmlichen Schallplatten. Bei der Schallplatte hat die Rille winzige „Berge“ und „Täler“, die von der Nadel abgetastet und in elektrische Stromschwankungen umgewandelt werden. Die Stromschwankungen werden elektrisch verstärkt und im Lautsprecher dann wieder dadurch hörbar, daß sich die Lautsprechermembran im Rhythmus der Stromschwankungen vor und zurück bewegt. Auf der CD („CD“ ist die Abkürzung für „Compact Disk“) wird die Musik ganz anders gespeichert. Bei der Aufnahme werden die Töne, die vom Mikrophon aufgenommen werden, als Zahlenwerte gespeichert. Genauer gesagt, es wird eigentlich nur die Lautstärke als Zahlenwert gespeichert — das aber viele tausend mal in einer Sekunde. Die so gewonnenen Zahlenkolonnen geben ganz genau die Musik wieder. Auf der CD werden diese Zahlen dann als schier endlose Reihe von mikroskopisch kleinen hellen und dunklen Punkten gespeichert. Um diese Punkte unterscheiden zu können, kann man nicht mehr eine normale Lampe nehmen, denn bei die-

ser wäre der Lichtstrahl viel zu breit. Im CD-Player wird daher ein Laserstrahl benutzt.

Laser ist eine besondere Methode, einen Lichtstrahl mit ganz besonderen Eigenschaften zu erzeugen. Das Licht hat nur eine einzige Farbe, wogegen das normale weiße Licht aus verschiedenen Farben zusammengesetzt ist. Gewöhnliches Licht breitet sich nach allen Seiten aus; Laserstrahlen verlaufen dagegen fast parallel zueinander. Die Lichtwellen eines Laserstrahls schwingen auch im Gegensatz zu normalem Licht alle im Gleichtakt. Daher ist das Licht sehr hell und intensiv.

Beim CD-Player wird ein ganz feiner Laserstrahl auf die Platte gelenkt, und dadurch kann die Information eines jeden Punktes abgetastet werden. Die Wiedergabe erfolgt dann umgekehrt wie die Aufnahme.

Auch in der Computertechnik werden CDs verwendet — denn an sich ist es egal, ob man Musik oder irgendwelche andere Daten speichert. Auf eine einzige CD passen über 600 Millionen Buchstaben, das sind z. B. etwa 350 000 Schreibmaschinenseiten, die aufeinandergelegt einen Turm von ca. 17 m Höhe ergeben. Man kann aber auch Bilder und Töne auf einer Computer-CD speichern, die man übrigens als „CD-ROM“ bezeichnet.





Sortier- anlage

Sortieren und Vereinzeln sind zwei sehr wichtige Aktionen bei der automatischen Herstellung von Waren oder Geräten aller Art. (Beim „Vereinzeln“ werden einzelne Teile, die in einer größeren Menge zur Verarbeitung bereit stehen, Stück für Stück weitergegeben.) Mit der Sortieranlage können verschiedene Teile erkannt und getrennt voneinander abgelegt werden.

Mit diesem Modell werden verschieden lange fischertechnik-Bausteine sortiert. Diese werden mit dem Zapfen nach oben auf die Rutsche gestellt. Sie werden zunächst durch den Schieber festgehalten. Sobald Du die Anlage startest, vereinzelt sie die Bausteine; es wird jeweils ein Baustein weitergeschoben. Der Baustein fällt dann auf das Transportband. Am Ende des Transportbandes befinden sich zwei Lichtschranken, die aus zwei Lichtsensoren und einer Linsenlampe bestehen.

Wird ein langer Baustein an den Lichtschranken vorbei transportiert, sind für einen Moment beide Lichtschranken unterbrochen. Der Baustein wird als langes Teil erkannt, und das quer zum Transportband angeordnete Sortierband transportiert ihn zur Ablage für die langen Bausteine. Wird ein kurzer Baustein an den Lichtschranken vorbei transportiert, ist immer nur eine Lichtschranke unterbrochen, und das Sortierband läuft entgegengesetzt wie vorher. Über eine Impulsscheibe kann das Programm den Weg des Sortierbandes messen und es anhalten, wenn der Baustein das Band verlassen hat.

Einstellung der Taster am Schieber

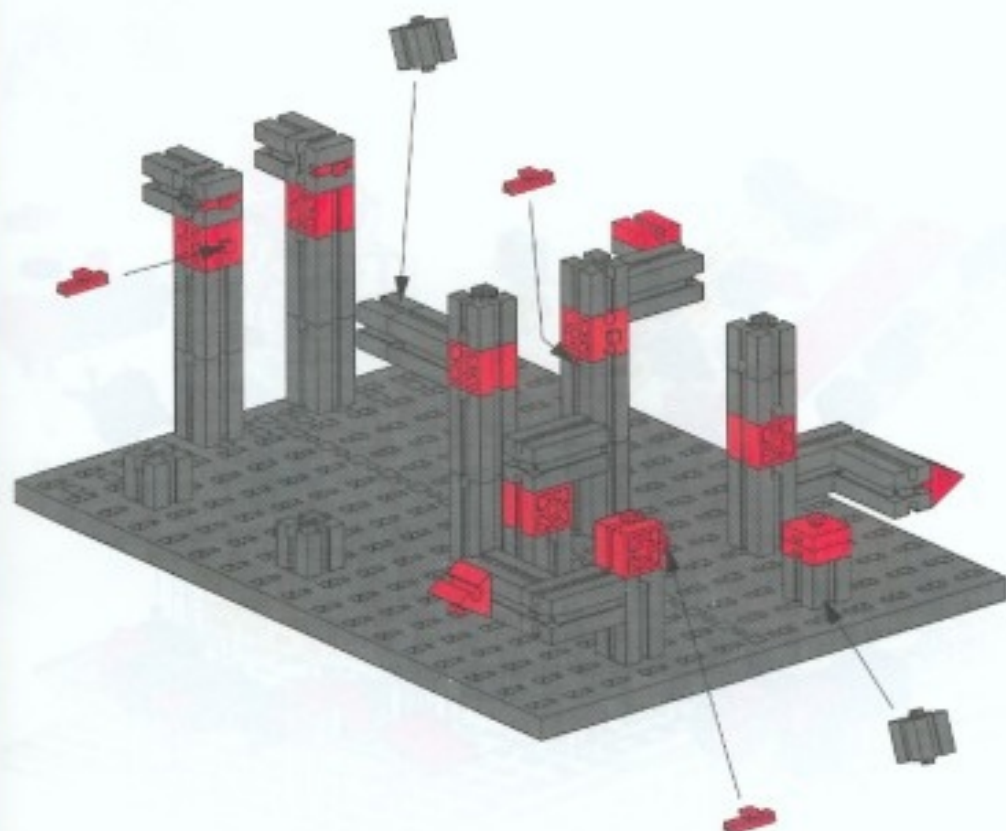
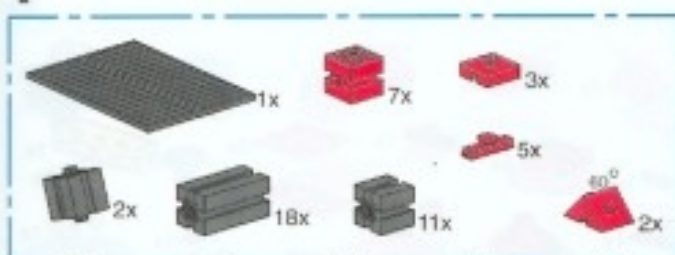
Stelle das fertige Modell so vor Dich hin, daß sich die Zuführung links befindet und das Sortierband rechts ist. Am Schieber befinden sich zwei Taster als Rückmelder für die Schieberstellung. Um die Taster einzustellen, ziehst Du das U-Getriebe ca. 2 mm vom Motor ab, damit Du den Schieber mit der Hand bewegen kannst. Vor den Schieber stellst Du nun einen Baustein aufrecht in die Zuführungsrinne (Zapfen nach oben). Jetzt den Schieber bis zum Anschlag nach

vorne schieben und den rechten Taster so verschieben, daß er knapp an der Getriebekante öffnet. Nun ziehst Du den Schieber soweit zurück, daß die Bausteine ungehindert nachrutschen können. Den linken Taster nun so verschieben, daß er in dieser Schieberstellung eben öffnet. Zum Schluß schiebst Du das U-Getriebe wieder in die alte Position.

Verdrahtung

Der Motor an M1 treibt das Sortierband an und der Motor an M2 das Transportband. Der Motor an M3 ist für die Schieberbewegung zuständig, und am Ausgang M4 wird die Lampe angeschlossen. Die Taster an E1 und E2 melden die Endstellungen des Schiebers (E1 linker Taster, E2 rechter Taster). An E3 wird der (in Laufrichtung des Transportbandes) erste Fototransistor, an E4 der zweite Fototransistor angeschlossen. Der Taster, der von der Impulsscheibe des Sortierbandes betätigt wird, kommt an E5. Bei den Fototransistoren Polung beachten! Achte auch auf die richtigen Kabellängen. Starte nun das Programm LUCKY LOGIC und teste die Verdrahtung mit dem Menüpunkt OPTIONEN DIAGNOSE. Beim Betätigen der Taster müssen sich die im Testplan (Seite 98) aufgeführten Eingangs-Anzeigen entsprechend verändern. Nun schalte die M-Ausgänge (Motoren oder Lampen) durch Anklicken mit der Maus, wie es in der Beschreibung zur Software erklärt ist. Wenn alles klappt, kann die Software geladen und gestartet werden.

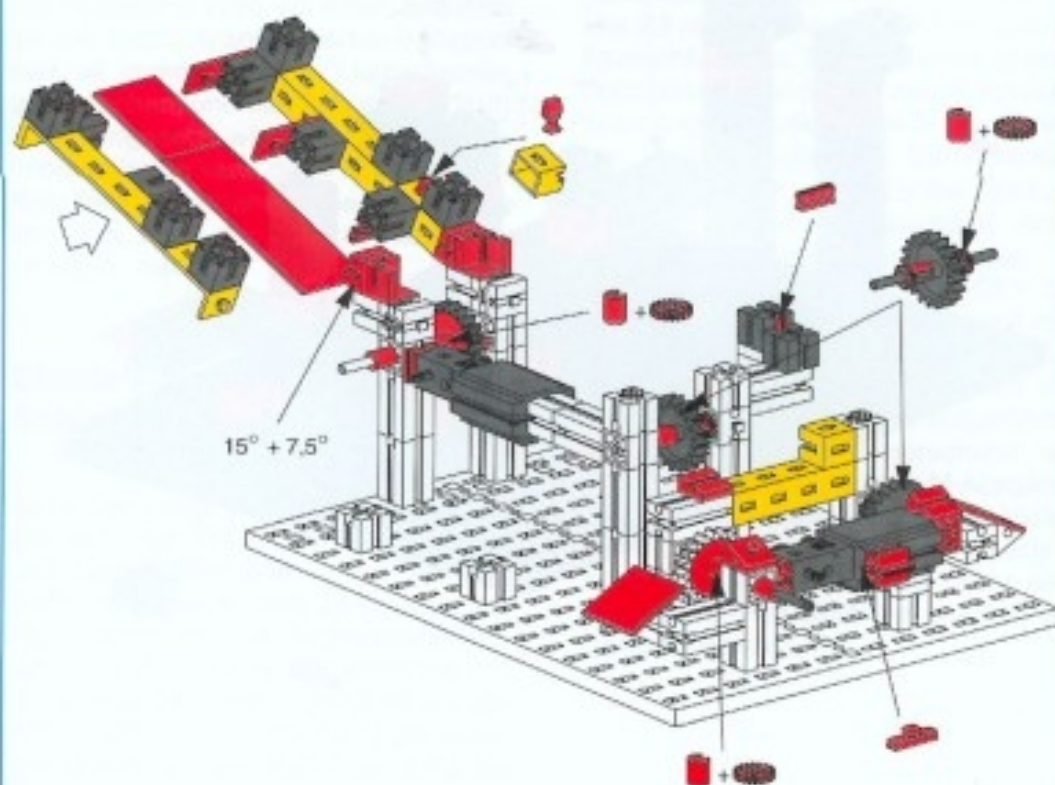
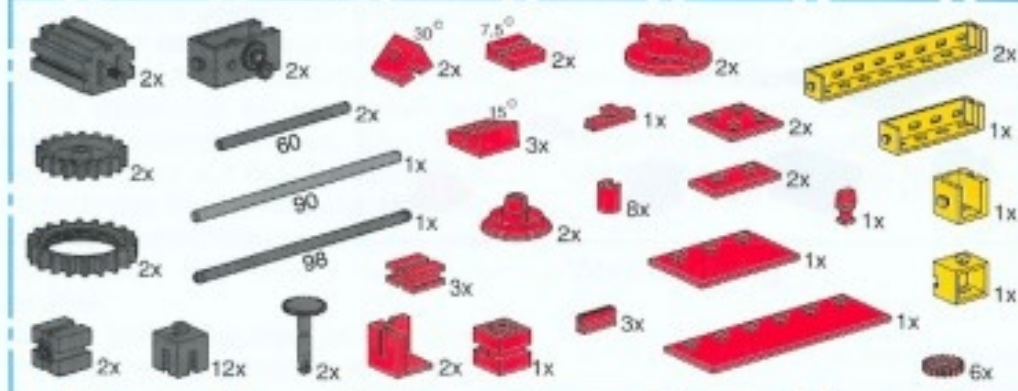
1



2

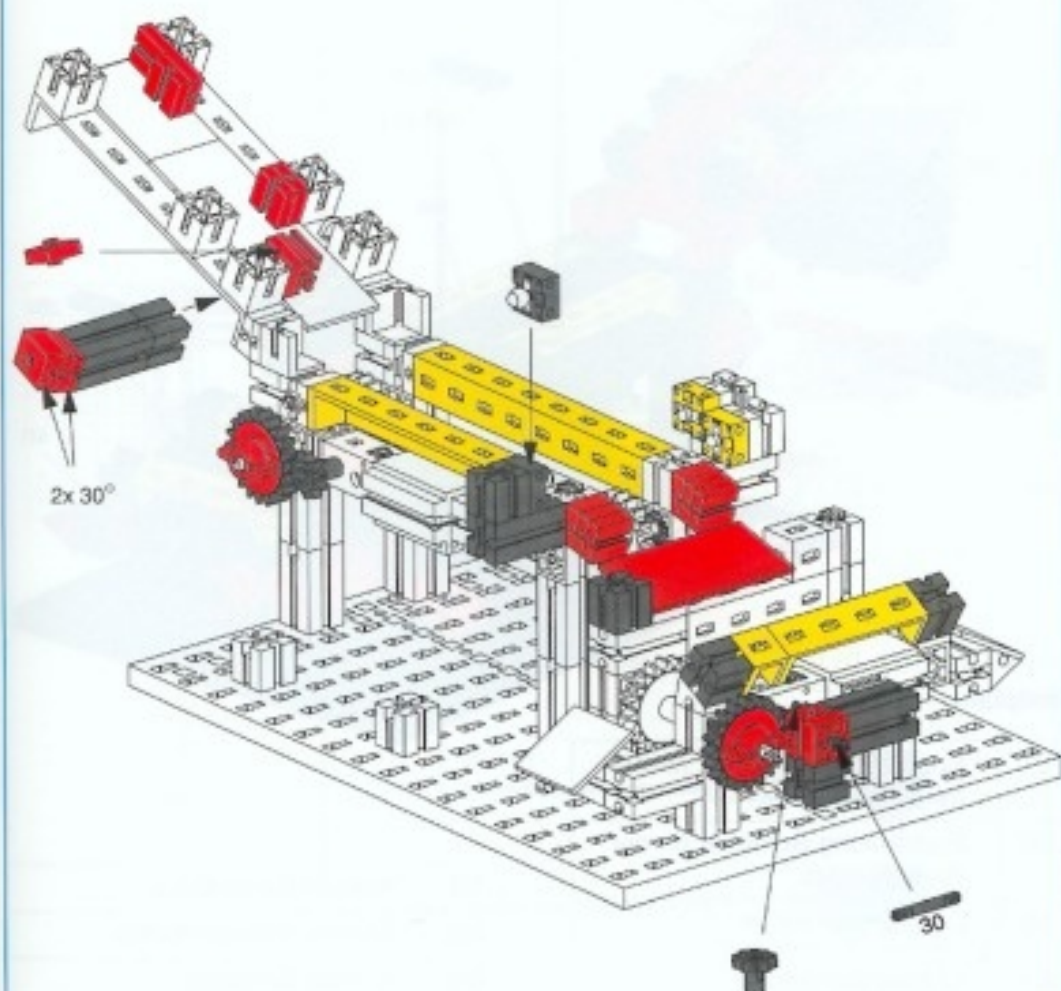
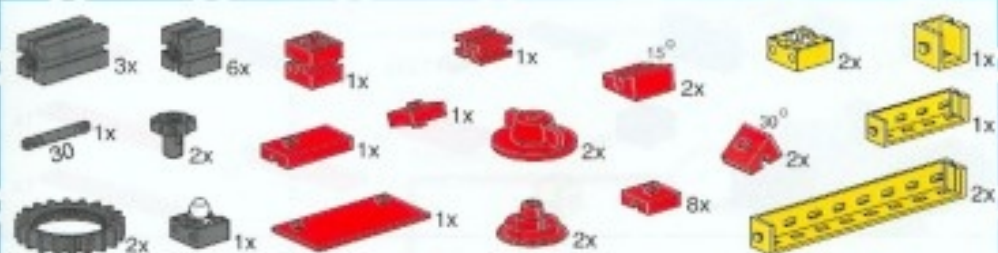
60mm

90mm 96mm

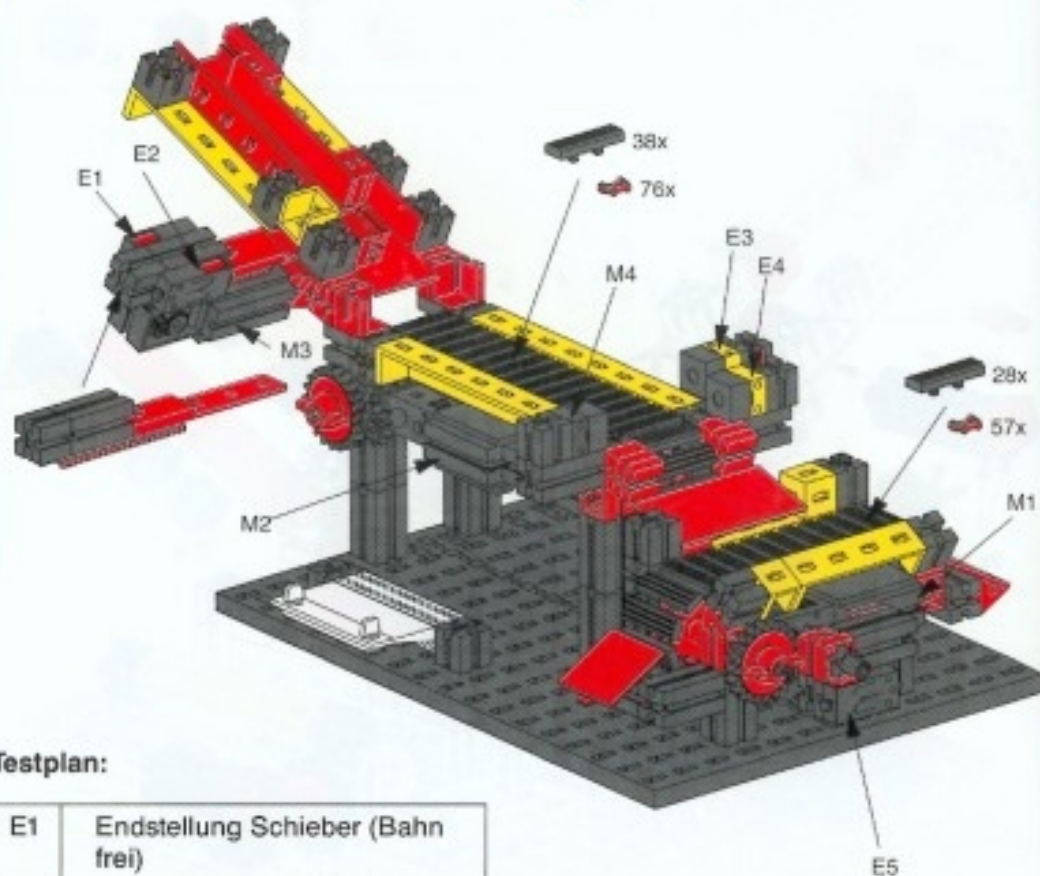


3

30mm



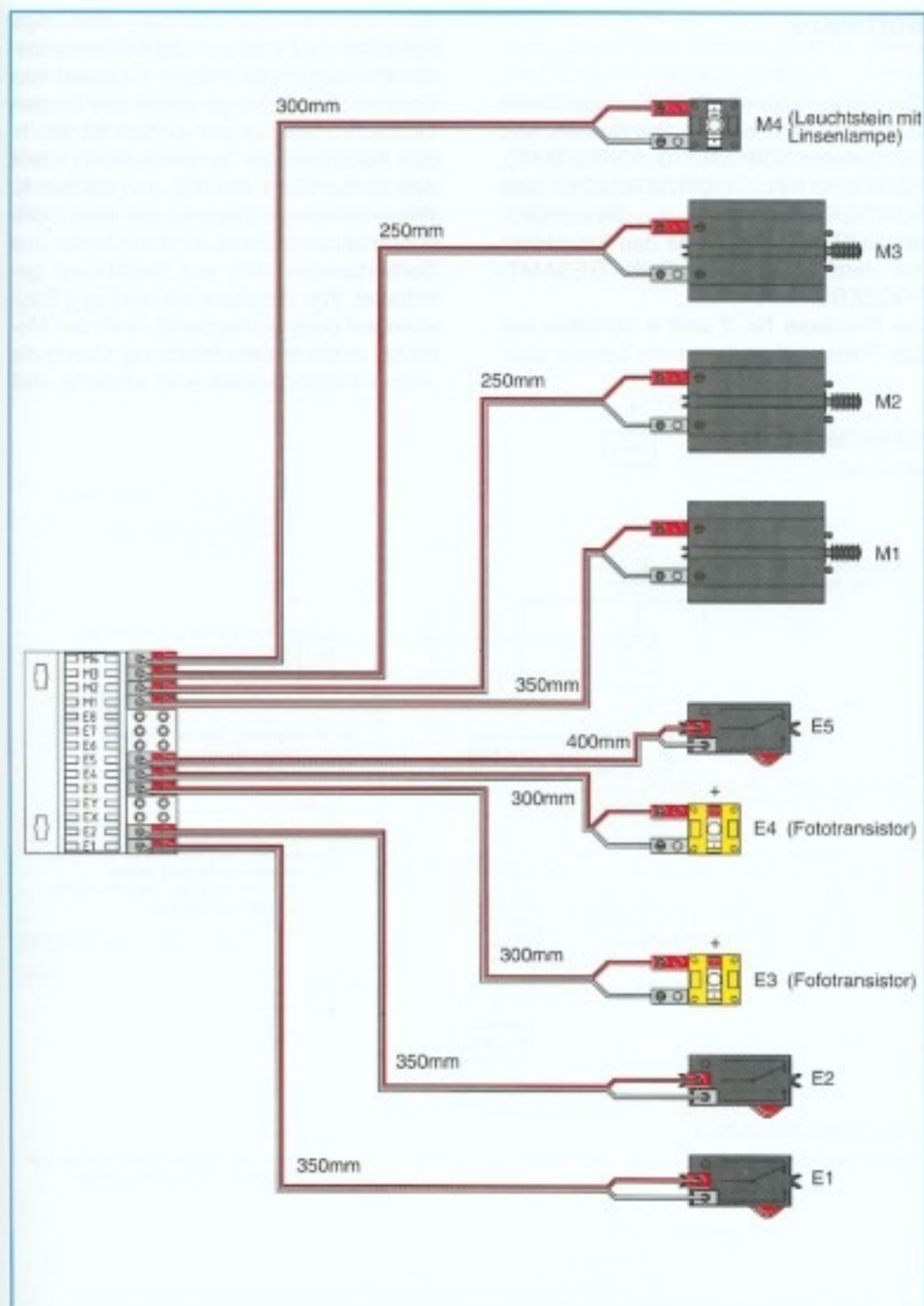
4



Testplan:

E1	Endstellung Schieber (Bahn frei)
E2	Endstellung Schieber (in der Rinne)
E3	Lichtsensor vorne
E4	Lichtsensor hinten
E5	Impulsscheibe Sortierband

M1	Antrieb Sortierband
M2	Antrieb Transportband
M3	Antrieb Schieber
M4	Linse Lampe



Software

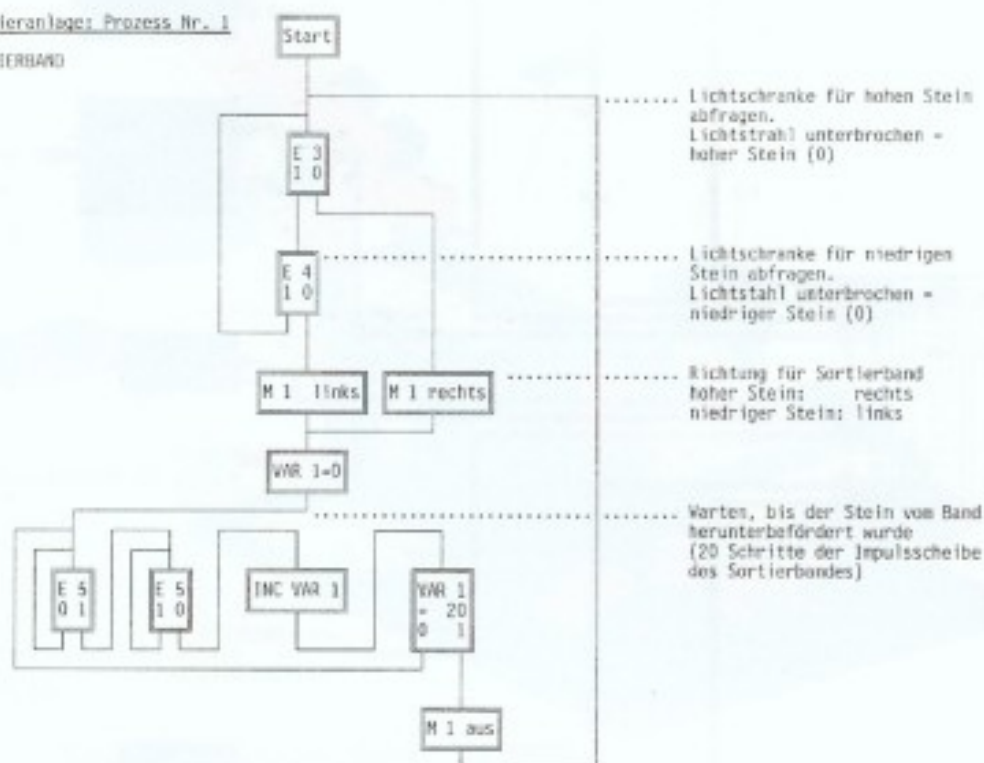
Die Software für die Sortieranlage ist mit LUCKY LOGIC erstellt. Sie besteht aus den Dateien SORTIER.FTG, SORTIER1.FTL, SORTIER2.FTL, SORTIER3.FTL und SORTIER4.FTL. Lade den Gesamtprozeß SORTIER und starte das Programm mit dem Menüpunkt RUN GESAMT-PROZESS.

Die Prozesse Nr. 2 und 4 schalten nur das Transportband und die Lampe dau-

ernd ein, interessant für die vollständige Funktion sind also nur die beiden anderen Prozesse. Der Prozeß 1 steuert das Sortierband. Solange keine der beiden Lichtschranken unterbrochen ist (d. h. der Fototransistor ist verdunkelt), steht das Sortierband still (E3 und E4 auf 1). Wenn ein langer Baustein die eine Lichtschranke unterbricht, wird der Motor des Sortierbandes (M1) auf Rechtslauf geschaltet. War dagegen ein niedriger Baustein auf dem Sortierband, läuft der Motor M1 in die andere Richtung. Durch die zweite Lichtschranke wird erreicht, daß

Sortieranlage: Prozess Nr. 1

SORTIERBAND



Sortieranlage: Prozess Nr. 2

TRANSPORTBAND



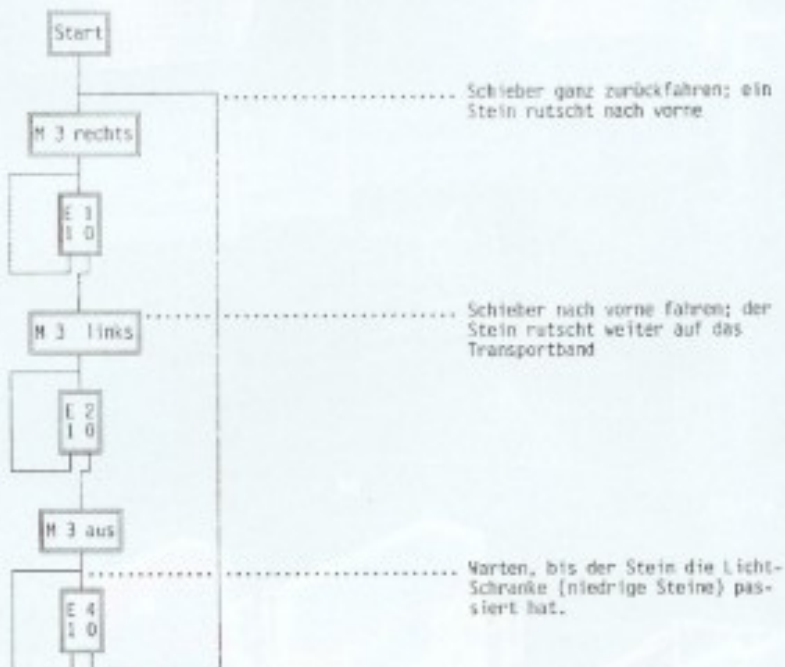
das Sortierband nur läuft, wenn ein Stein auf dem Transportband liegt. Nach einer gewissen Zeit (20 Impuls der Impuls-scheibe am Sortierband) bleibt das Band wieder stehen.

Der Prozeß 3 steuert den Schieber zum Vereinzeln der Bausteine. Der Schieber fährt ganz zurück (Stein rutscht vor den

Schieber) und dann ganz vor (Stein rutscht auf das Transportband). Bevor der Schieber wieder in Aktion tritt, wartet der Prozeß, bis der Baustein die Licht-schranke passiert hat — das ist wichtig, damit zwischen den einzelnen Bausteinen ein genügend großer Abstand eingehalten wird.

Sortieranlage: Prozess Nr. 3

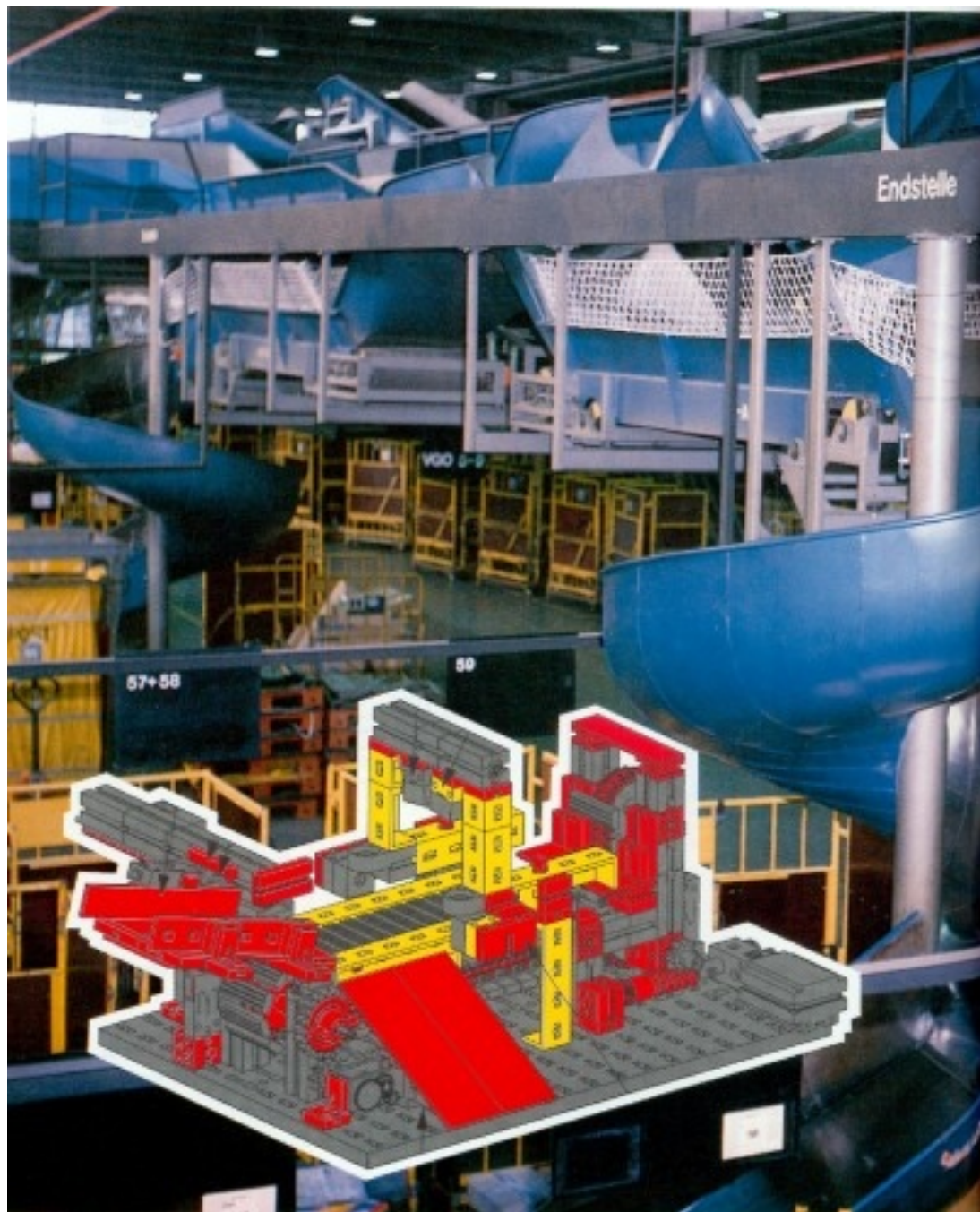
SCHIEBER



Sortieranlage: Prozess Nr. 4

LAMPE







Paket- Wende- anlage

Zur Weiterverteilung müssen Pakete so gedreht werden, daß die Anschrift oben liegt, damit beim Verteilen der Pakete auf die Zielpostämter alles „ruck-zuck“ gehen kann. Mit der Paketwendeanlage wird ein, über die Rutsche zugeführtes, Paket so oft gedreht, bis sich die schwarze Seite oben befindet. (Bei der echten Paketwendeanlage ist dies die Seite mit dem Adreßaufkleber.) Übrigens werden auch Briefe in den Verteil-Postämtern automatisch gedreht und gewendet, bis die Briefmarke rechts oben liegt, damit sie automatisch abgestempelt werden kann (dazu ist die Marke mit einer Farbe bedruckt, die bei Bestrahlung mit Ultraviolettlicht aufleuchtet).

Das Paket wird quer über die Rutsche in Richtung zum Band geschoben. Dabei wird der Taster an der Rutsche betätigt und so die Anlage automatisch eingeschaltet. Das Transportband bewegt das Paket unter die Reflexlichtschranke und bleibt dann stehen (über die waagrecht angeordnete, zweite Lichtschranke ist das leicht festzustellen). Über die unterschiedliche Reflexion der gelben und schwarzen Fläche kann das Programm feststellen, ob die schwarze Seite schon oben ist.

Erkennt die Reflexlichtschranke eine gelbe Fläche, muß das Paket gewendet werden. Das Transportband läuft wieder an und transportiert das Paket weiter zur Wendestation. Mit der vertikalen Kette und den daran befestigten Raupenbelägen wird das Paket angehoben und durch das laufende Transportband gekippt. Damit das Paket nur einmal gekippt wird, befindet sich auf der Rückseite der Wendestation ein Taster, der von den Raupenbelägen betätigt wird (dann schaltet der Computer die senkrechte Kette wieder ab). Nun wird das Paket wieder zurück zur Lichtschranke transportiert. Der Weg des Transportbandes wird über einen Taster mit Impulsscheibe zum Computer gemeldet. Das Ganze wird wiederholt, bis die schwarze Seite oben liegt.

Erkennt die Reflexlichtschranke die schwarze Fläche, liegt das Paket richtig.

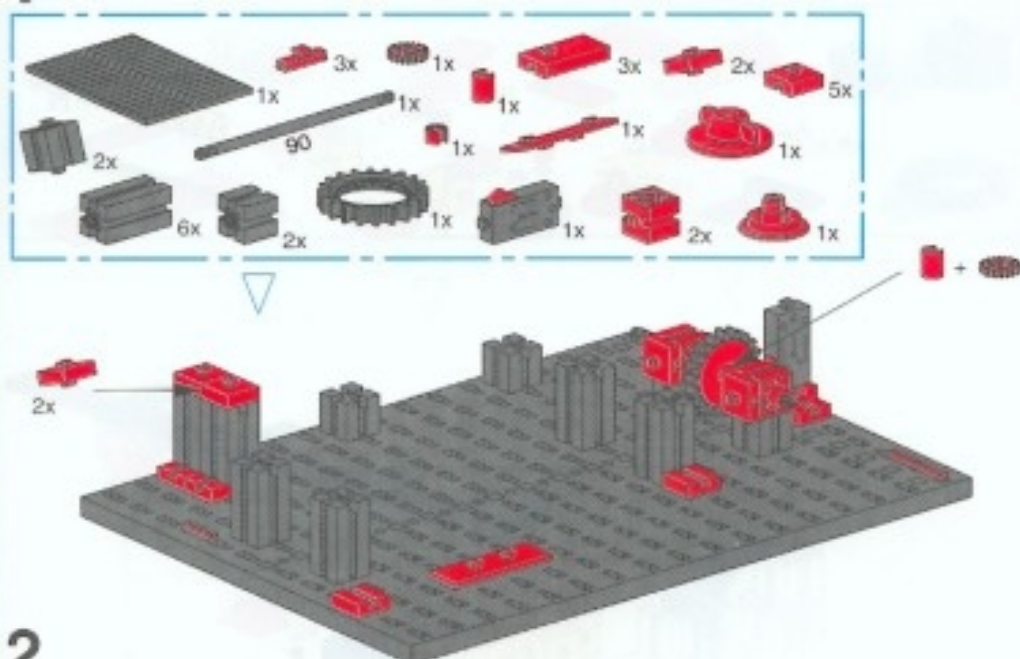
Das Transportband läuft von der Lichtschranke zurück, und das Paket wird dann mit dem Schieber vom Transportband auf die Ausgaberutsche befördert. Nun ist die Anlage bereit für das nächste Paket.

Die beiden Lichtschranken sollten möglichst mit Linsenlampen bestückt werden. Die Zangenmutter der klemmbaren Schnecke und die Nabennuttern der Zahnräder gut festziehen!

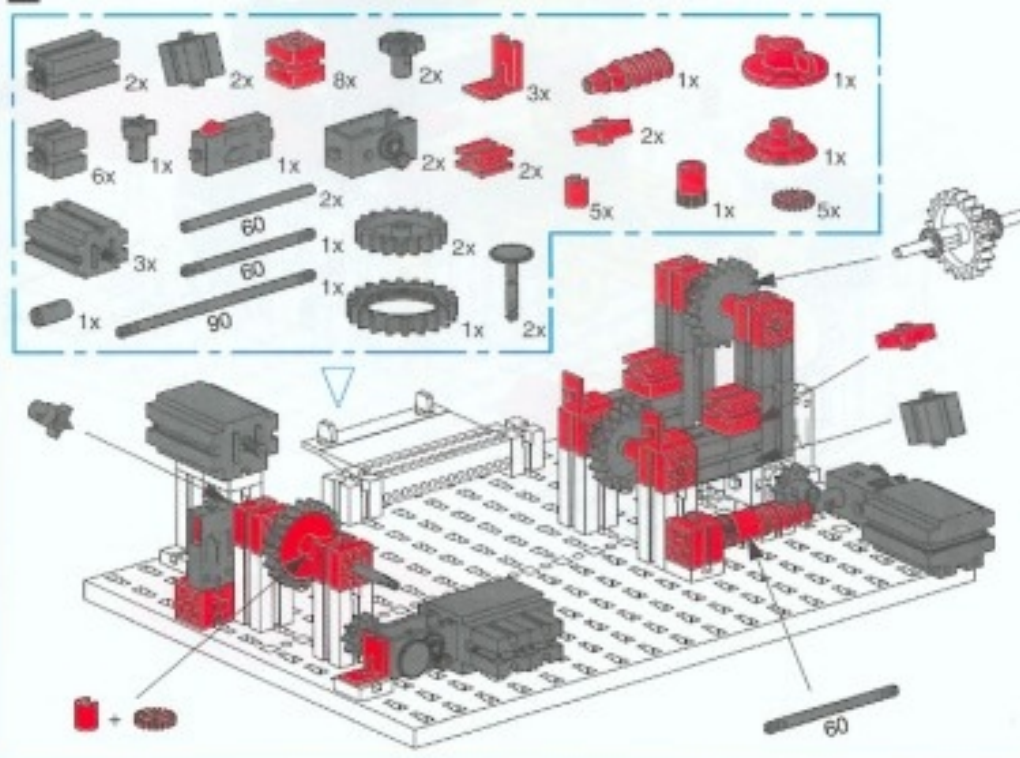
Einstellen der Taster am Schieber

Stelle das fertige Modell so vor Dich hin, daß die 28polige Steckerleiste zur Dir hin weist. Um die Taster einzustellen, ziehst Du das Hub-Getriebe ca. 2 mm vom Motor ab, damit Du den Schieber mit der Hand bewegen kannst. Jetzt bewege den Schieber nach vorne über das Transportband hinweg bis zur Ausgaberutsche. Verschiebe den linken Taster so, daß er gerade an der Kante des U-Getriebes schaltet (Schieber mit der anderen Hand festhalten). Dann den Schieber zurückziehen, bis das Transportband ganz frei ist. Nun die rote Bauplatte so verschieben, bis der rechte Taster gerade schließt (Schieber wieder festhalten). Danach das U-Getriebe wieder an die alte Stelle schieben.

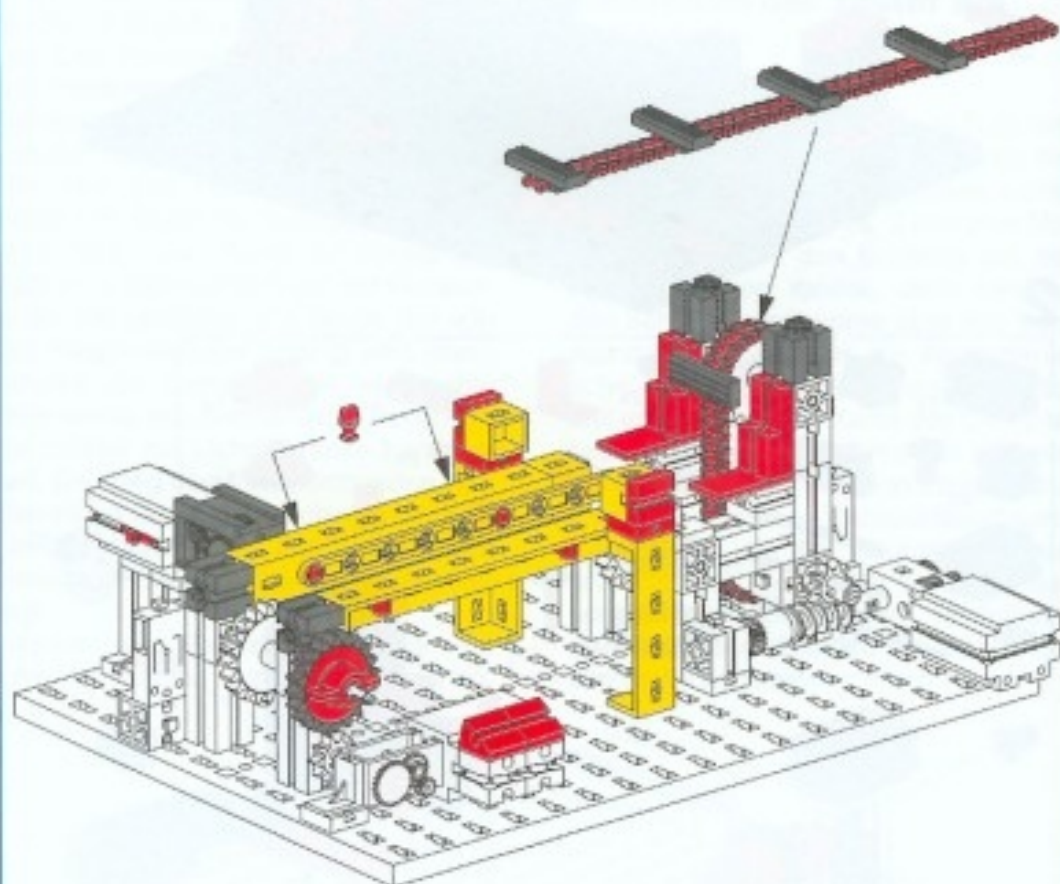
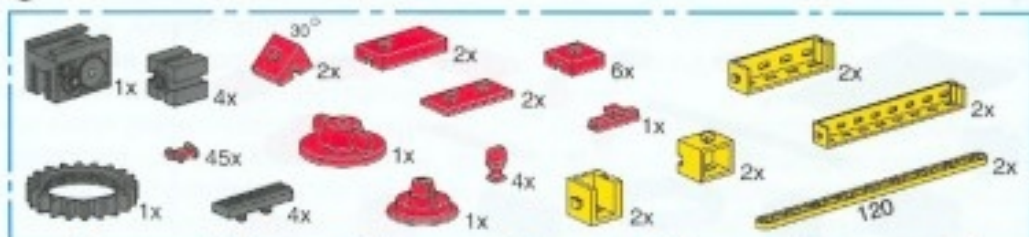
1



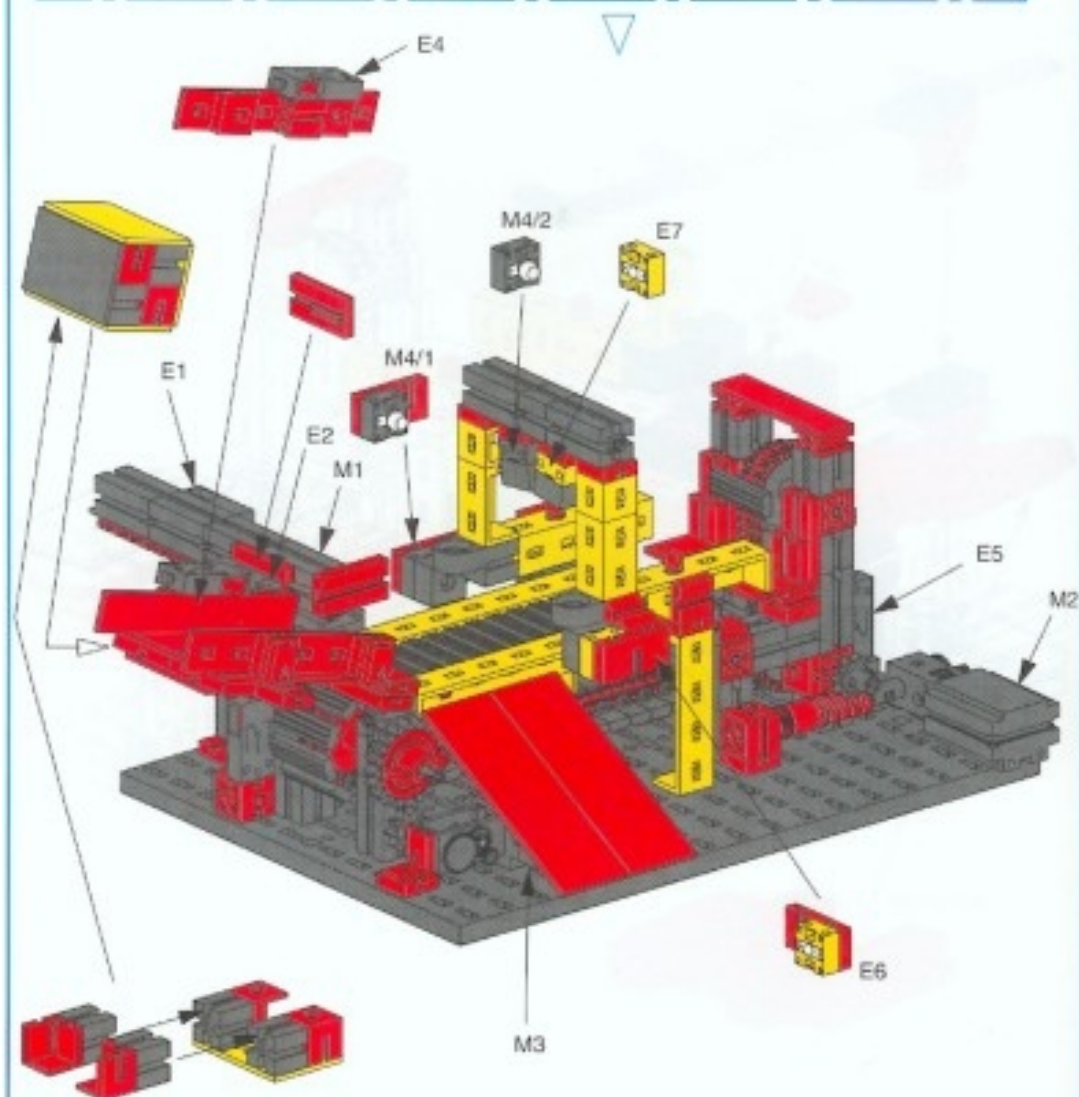
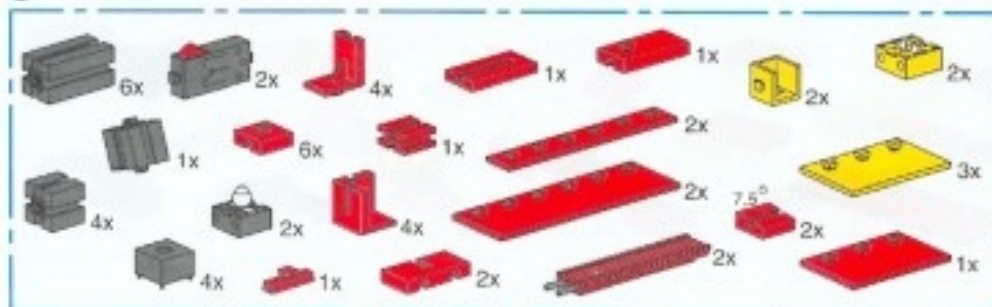
2



3



5



Verdrahtung

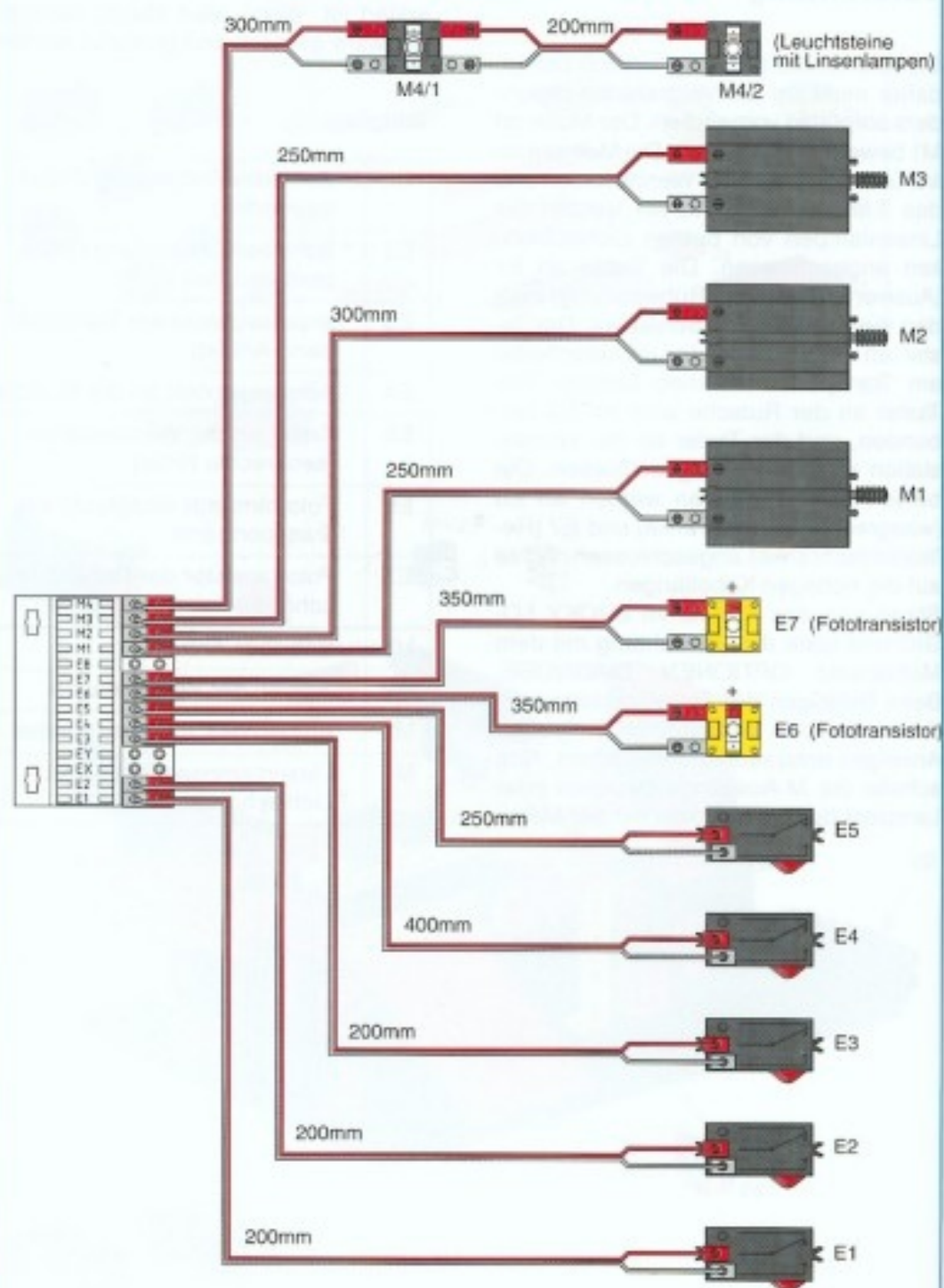
Diesmal ist das Interface fast voll belegt, daher mußt Du die Verdrahtung besonders sorgfältig vornehmen. Der Motor an M1 bewegt den Schieber. Die Motoren an M2 und M3 treiben die Wendestation und das Transportband. An M4 werden die Linsenlampen von beiden Lichtschranken angeschlossen. Die Taster an E1 (Auswerfen) und E2 (Ruhestellung) melden die Stellung des Schiebers. Der Taster an E3 wird von der Impulsscheibe am Transportbandantrieb betätigt. Der Taster an der Rutsche wird mit E4 verbunden, und der Taster an der Wendestation wird an E5 angeschlossen. Die beiden Fototransistoren werden an E6 (waagrechte Lichtschranke) und E7 (Reflexlichtschranke) angeschlossen. Achte auf die richtigen Kabellängen.

Starte nun das Programm LUCKY LOGIC und teste die Verdrahtung mit dem Menüpunkt OPTIONEN DIAGNOSE. Beim Betätigen der Taster müssen sich die im Testplan aufgeführten Eingangs-Anzeigen entsprechend verändern. Nun schalte die M-Ausgänge (Motoren oder Lampen) durch Anklicken mit der Maus,

wie es in der Beschreibung zur Software erklärt ist. Wenn alles klappt, kann die Software geladen und gestartet werden.

Testplan:

E1	Schieber-Endstellung (Paket auswerfen)
E2	Schieber-Endstellung (Transportband frei)
E3	Impulsscheibe am Transportband-Antrieb
E4	Auslösekontakt an der Rutsche
E5	Taster an der Wendestation (senkrechte Kette)
E6	Fototransistor waagrecht am Transportband
E7	Fototransistor der Reflexlichtschranke
M1	Schieber-Antrieb
M2	Antrieb der Wendestation
M3	Antrieb des Transportbandes
M4	Linsenlampen von beiden Lichtschranken



Software

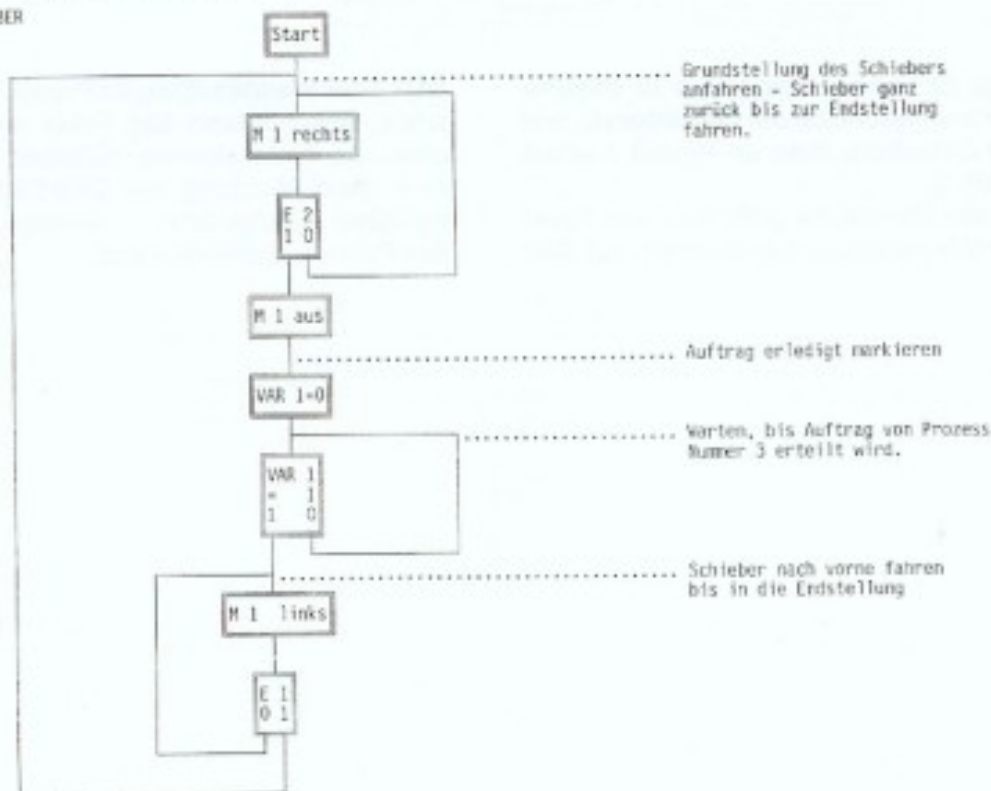
Die Software für die Paketwendeanlage ist mit LUCKY LOGIC erstellt. Sie besteht aus den fünf Dateien PAKET.FTG, PAKET1.FTL, PAKET2.FTL, PAKET3.FTL und PAKET4.FTL. Lade den Gesamtprozeß PAKET und starte das Programm mit dem Menüpunkt RUN GESAMTPROZESS.

Der erste Prozeß steuert den Schieber für das Auswerfen des Pakets. Der Schieber wird zunächst in die Grundstellung (ganz zurück) gefahren. Dann wartet dieser Prozeß, bis über VAR 1 der Auftrag vom Prozeß 3 erteilt wird, das Paket auszuwerfen (Schieber nach vorne fahren und wieder zurück).

Prozeß Nr. 2 steuert das Wendeband. Am Anfang läuft das Wendeband, bis der Taster E5 durch einen Raupenstein geschlossen wird. Dann hält das Wendeband an und wartet auf den Auftrag vom Prozeß 3 (VAR 2). Zum Wenden des Pakets läuft das Wendeband bis zur nächsten Betätigung vom Taster an E5. Der dritte Prozeß steuert das Förderband und erteilt die Aufträge an Prozeß Nr. 1 (Auswerfen) und Prozeß Nr. 2 (Wenden). Wenn ein Paket eingelegt wurde (Taster an E4 meldet das Paket), wird das Paket bis unter die Lichtschranke gefahren. Die Lichtschranke an E6 stellt fest, wann das Paket an der richtigen Stelle liegt. Danach wird die Helligkeit der Oberfläche von der Lichtschranke an E7 gemessen. Ist die Oberseite schwarz, wird das Paket

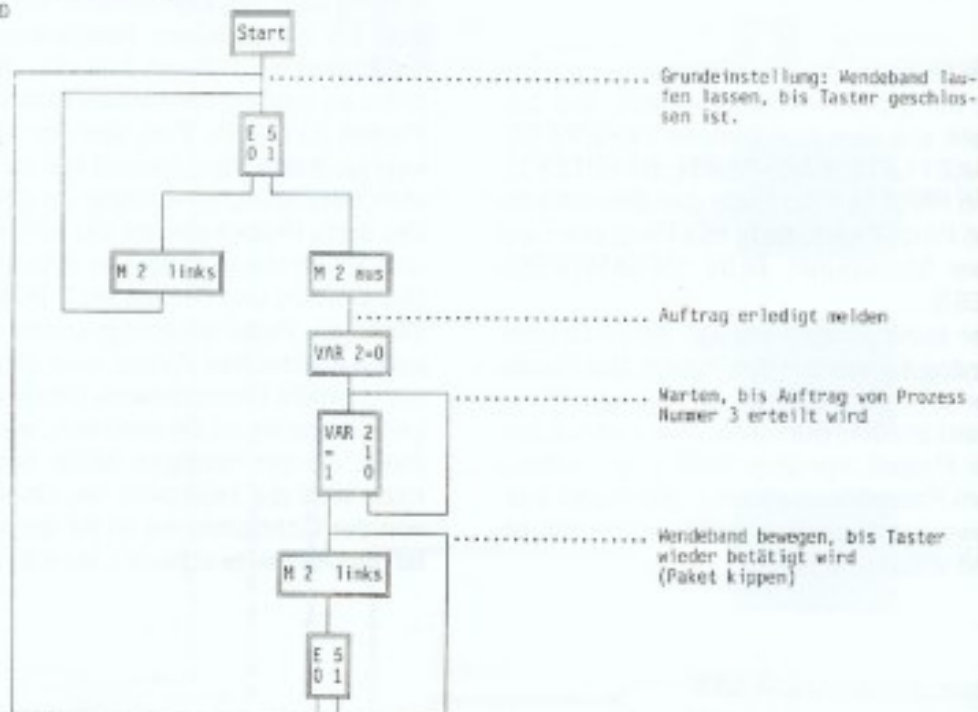
Paketwendeanlage: Prozess Nr. 1

SCHIEBER



Paketwendeanlage: Prozess Nr. 2

WENDEBAND



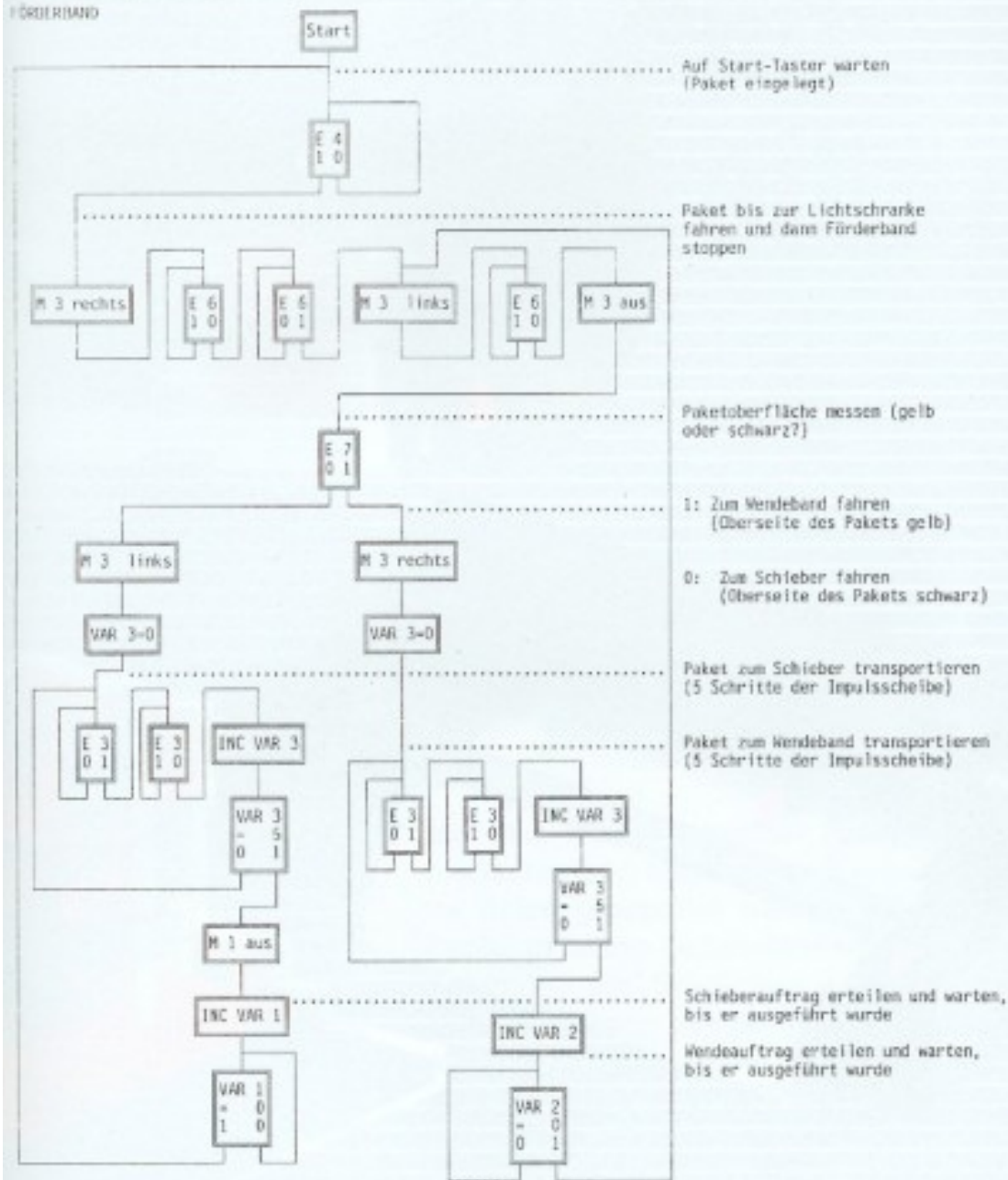
zum Schieber transportiert (5 Impulse der Impulsscheibe am Förderband), und der Schieberauftrag an Prozeß 1 erteilt (VAR 1).

Ist die Oberfläche gelb, wird das Paket zur Wendeanlage transportiert und über

VAR 2 der Wendeauftrag an Prozeß Nr. 2 erteilt, anschließend das Paket wieder unter die Lichtschranke befördert und eine neue Messung der Oberflächen-Helligkeit durchgeführt — solange, bis das Paket ausgeworfen wird.

Paketwendeanlage: Prozess Nr. 3

FÖRDERBAND



Paketwendeanlage: Prozess Nr. 4

LICHT





Kurven- schreiber



Mit dem Kurvenschreiber (oder Kennlinienschreiber) kann auf zwei verschiedene Arten gearbeitet werden. Wenn das Papier in festen Zeitabständen (z. B. jede Sekunde) ein Stückchen vorwärts bewegt wird, kann der zeitliche Verlauf der aufgezeichneten Daten dargestellt werden. Wenn Du z. B. im Programm bei jedem Papier-Transportschritt den Stift nach rechts bewegst, bis er ganz an der rechten Seite angekommen ist und dann den Stift bei jedem Schritt nach links bewegst, bis er ganz links angekommen ist, zeichnet der Schreiber eine Schlangenlinie.

Der Fachmann nennt so einen Schreiber „X-t-Schreiber“ („t“ ist in der Technik die Abkürzung für die Zeit (lateinisch „tempora“) und „X“ steht für die waagrechte Bewegung des Stiftes).

Wenn das Papier vom Programm vorwärts und rückwärts bewegt wird, kann der Stift auf dem ganzen Papier zeichnen. Dazu sagt der Fachmann „X-Y-Schreiber“ („X“ und „Y“ stehen für die beiden Bewegungsrichtungen von Papier und Schreiber). Bewegt sich nur der Stift oder nur das Papier, werden waagrechte oder senkrechte Linien gezeichnet.

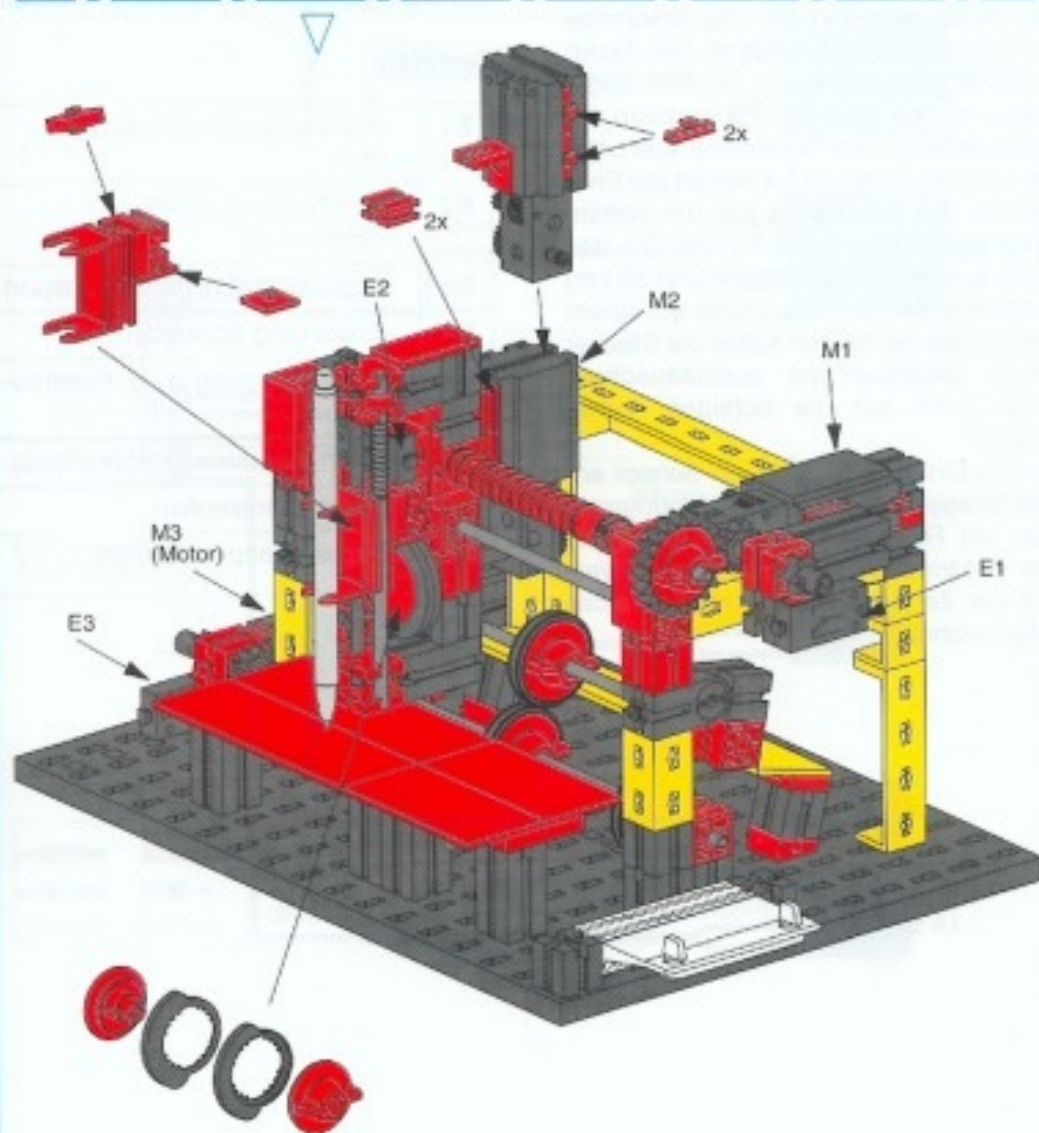
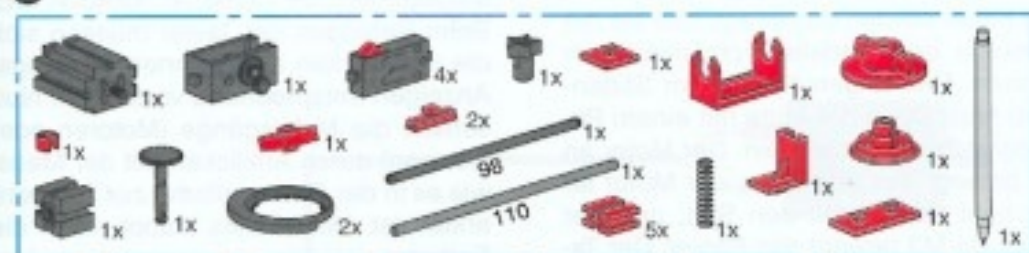
Das Zeichenpapier sollte 14 cm breit sein, die Länge des Papiers ist beliebig. Zum Einlegen des Papiers schiebst Du es von der Schreiberseite aus soweit hinein, bis es von den Transportrollen erfaßt

wird. Um das Papier in die Grundposition zu bringen, kannst Du den Transportmotor über die Diagnosefunktion von LUCKY LOGIC steuern.

Der Schreibstift wird so eingestellt, daß sich seine Spitze in angehobener Position etwa 3 bis 4 mm über dem Papier befindet. In dieser Stellung ist auch der Taster für die Stiftposition betätigt. Bei längeren Zeichenpausen (oder, wenn Du den Schreiber nicht benutzt) die Schutzkappe wieder auf den Stift setzen, sonst trocknet der Stift aus.

Der Schreiber bewegt sich leichter, wenn auf die Führungsstange unter der Antriebsschnecke ein Tropfen Haushaltsöl gegeben wird. Achte auch darauf, daß die Nabenmutter von Zahn- und Antriebsrädern und die Zangenmutter der Schnecken gut festgezogen sind.

5



Verdrahtung

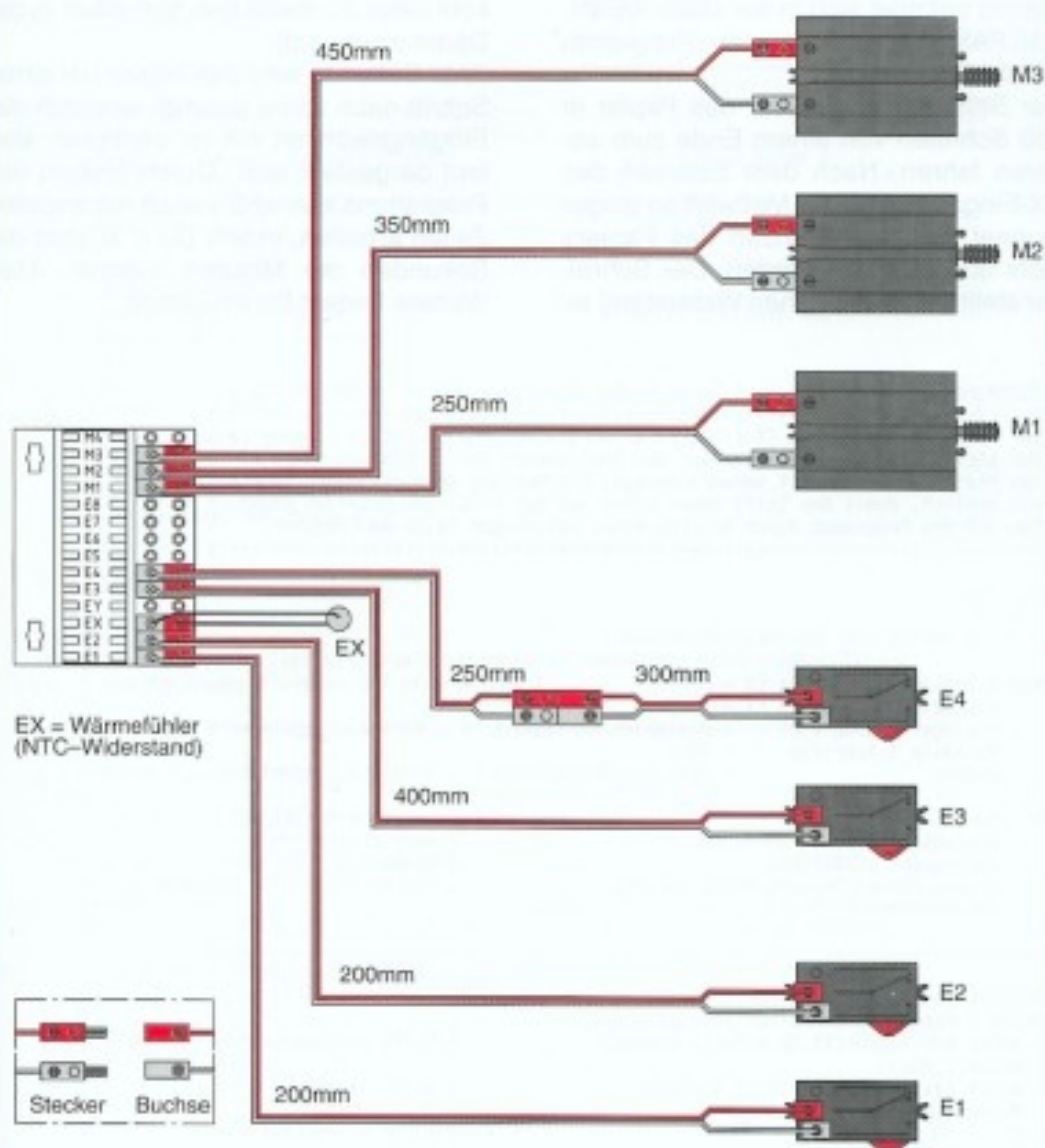
Die Kabel werden so verlegt, daß sie die Funktion des Modells nicht behindern können. Die langen Kabel zum Stiftantrieb hältst Du in der Mitte mit einem Papierdrahtband zusammen. Der Motor an M1 bewegt den Schreiber, der Motor an M2 hebt oder senkt den Stift, und der Motor an M3 bewegt das Papier. Der Taster an E1 wird von der Impulsscheibe am Schreiberantrieb betätigt. Der Taster an E2 ist gedrückt, wenn der Stift angehoben ist. Der Taster an E3 wird von der Impulsscheibe des Papier-Antriebs betätigt, und der Taster an E4 meldet die Endstellung des Schreibers. Bei der Verbindung zum Taster an E4 mußst Du das Kabel aus zwei Teilen (25 cm und 30 cm) zusammensetzen. Dazu sind an einem Ende eines der beiden Kabel die Stecker gegen Steckbuchsen auszutauschen. Achte auch auf die richtigen Kabellängen.

An den Eingang EX wird ein Sensor angeschlossen (z. B. ein Potentiometer oder ein Fototransistor). Die Änderung des elektrischen Widerstandes im Verlauf der Zeit wird dann vom Schreiber aufgezeichnet.

Starte nun das Programm LUCKY LOGIC und teste die Verdrahtung mit dem Menüpunkt **OPTIONEN DIAGNOSE**. Beim Betätigen der Taster müssen sich die im Testplan aufgeführten Eingangs-Anzeigen entsprechend verändern. Nun schalte die M-Ausgänge (Motoren oder Lampen) durch Anklicken mit der Maus, wie es in der Beschreibung zur Software erklärt ist. Wenn alles klappt, kann die Software geladen und gestartet werden.

Testplan:

E1	Impulsscheibe Schreiberantrieb
E2	Stift-Stellung (oben = 1, unten = 0)
E3	Impulsscheibe Papiertransport
E4	Einstellung Schreiber
EX	Sensoreingang (z. B. Potentiometer)
M1	Schreiberantrieb (rechts/links)
M2	Stift heben/senken
M3	Papiertransport-Antrieb



Software

Die Software zu diesem Modell ist in Turbo-Pascal geschrieben. Das Pascal-Programm befindet sich in der Datei KENNLINI.PAS. Starte nun das Programm KENNLINI.

Der Stift kann quer über das Papier in 360 Schritten von einem Ende zum anderen fahren. Nach dem Einlesen des EX-Eingangs wird der Meßwert so umgerechnet, daß die Grenzen des Papiers nicht überschritten werden. Der Schreiber stellt den elektrischen Widerstand ei-

nes Sensors am Eingang EX dar. Du kannst zum Ausprobieren z. B. einen Fototransistor aus dem Baukasten nehmen (Beim Anschließen auf die Polung achten!) und diesen dann langsam verdunkeln (oder du stellst den Schreiber in der Dämmerung auf).

Jede Sekunde wird das Papier um einen Schritt nach vorne bewegt, wodurch der Eingangswert an EX im zeitlichen Verlauf dargestellt wird. Durch ändern des Programms kannst Du auch mit anderen Zeiten arbeiten, indem Du z. B. statt der Sekunden die Minuten nimmst. Alles Weitere findest Du im Listing.

```
PROGRAM Kennlinienschreiber; { Turbo Pascal 5.0 - 6.0, Datei: KENNLINI.PAS }
{.....}
{ Der Kennlinienschreiber (X-t-Schreiber) mißt einen Analogwert am Eingang EX }
{ und stellt den zeitlichen Verlauf des Analogwerts dar - jede Sekunde wird }
{ das Papier einen Schritt vorwärtsbewegt. Die Messung erfolgt jedoch so oft }
{ wie möglich, damit der Stift immer schon auf der richtigen Position steht. }
{ Abbruch des Programms durch Drücken einer beliebigen Taste am Computer. }
{.....}

USES CRT,
      DOS,
      Fischer;

CONST Minimaler_Analogwert_EX = 70;
      Maximaler_Analogwert_EX = 450;
      Analoger_Messbereich = Maximaler_Analogwert_EX - Minimaler_Analogwert_EX;
      Maximale_X_Schritte = 360;
      Faktor = Analoger_Messbereich DIV Maximale_X_Schritte;

VAR X,                               { letzte X-Position }
     Zielposition X,                 { neue X-Position }
     Analogwert : INTEGER;           { Meßwert }
     Stunde,Minute,Sekunde,Hndst_Sekunde, { für Vorschub alle Sekunden }
     Sekundenmerker : WORD;

{.....}

PROCEDURE Kennlinienschreiber_initialisieren;
BEGIN { Kennlinienschreiber_initialisieren }
  WHILE NOT Taster(4) DO Motor(1,Rechts); { Stift zur Endstellung fahren }
  Motor(1,Aus);
  WHILE Taster(2) DO Motor(2,Rechts); { Stift senken }
  Motor(2,Aus);
  X := 0; { Werte Nullsetzen }
  Sekundenmerker := 0;
END; { Kennlinienschreiber_initialisieren }

{.....}

PROCEDURE Zeilenvorschub;
BEGIN { Zeilenvorschub }
  GETTIME(Stunde,Minute,Sekunde,Hndst_Sekunde); { Uhrzeit holen }
  IF Sekunde <> Sekundenmerker { Wenn eine Sekunde vergangen }
  THEN { ist, dann Papier einen Schritt }
```

```

BEGIN                                     { nach vorne fahren           }
  Motor(3,Rechts);
  REPEAT UNTIL Taster(3);                { Warten Impulstaster betätigt }
  REPEAT UNTIL NOT Taster(3);            { Warten Impulstaster frei }
  Motor(3,Aus);
  Sekundenmerker := Sekunde;              { letzte Zeit merken }
END;

END; { Zeilenvorschub }

{.....}

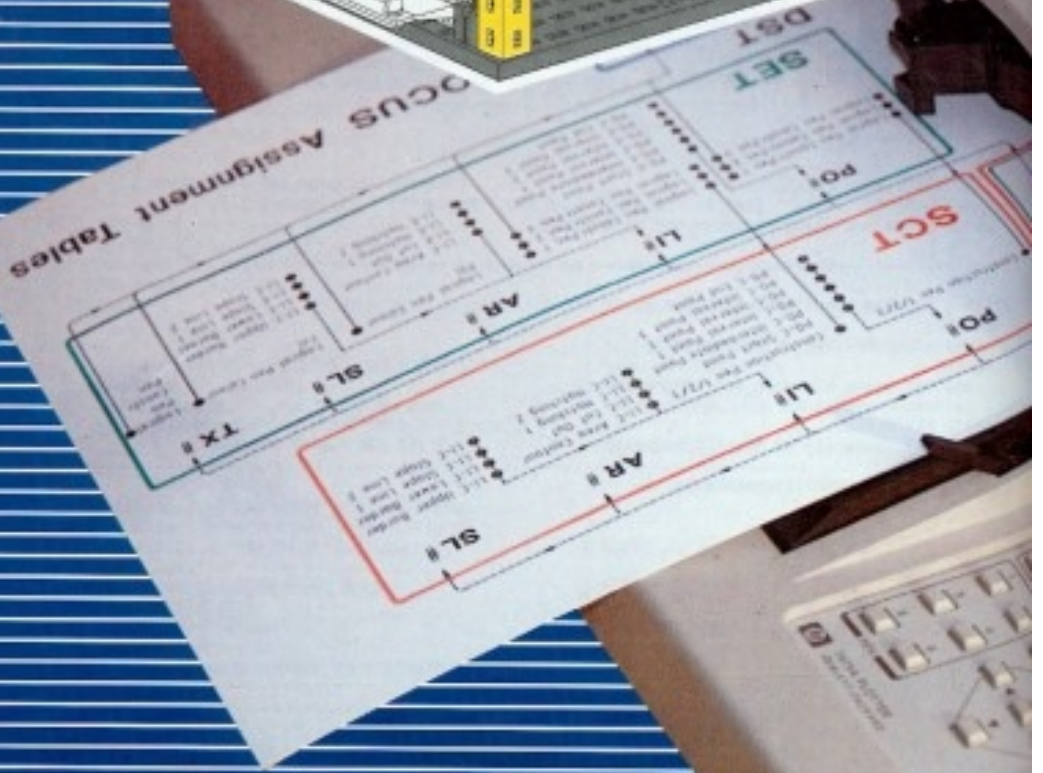
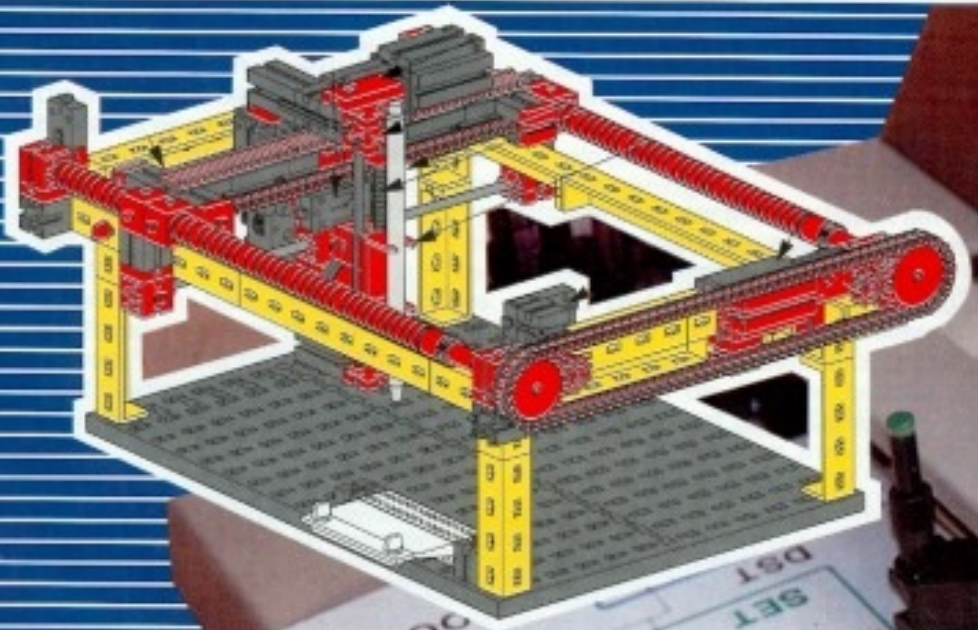
PROCEDURE Analogeingang_einlesen_und_verarbeiten;
BEGIN { Analogeingang_einlesen_und_verarbeiten }
  Analogwert := Analog_Eingang(X_PÖt);    { Wert von EX holen }
  { Wert auf die Papierbreite umrechnen }
  Zielposition_X := (Analogwert - Minimaler_Analogwert EX) DIV Faktor;
  IF Zielposition_X > X                    { Stift einen Schritt nach links }
  THEN
    BEGIN
      Motor(1,Links);
      REPEAT UNTIL Taster(1);              { Impuls Stiftantrieb }
      REPEAT UNTIL NOT Taster(1);
      INC(X);
    END;
  IF Zielposition_X < X                    { Stift einen Schritt nach rechts }
  THEN
    BEGIN
      Motor(1,Rechts);
      REPEAT UNTIL Taster(1);
      REPEAT UNTIL NOT Taster(1);
      DEC(X);
    END;
  Motor(1,Aus);

  GOTOXY(1,1);                            { Kontrollausgabe auf dem Schirm }
  WRITELN('X: ',X:3,' Ziel_X: ',Zielposition_X:3,' Messwert: ',Analogwert:3);
END; { Analogeingang_einlesen_und_verarbeiten }

{.....}

BEGIN { PROGRAM Kennlinienschreiber }
  CLRSCL;
  Kennlinienschreiber_initialisieren;
  REPEAT                                     { Schreiber läuft }
    Analogeingang_einlesen_und_verarbeiten;
    Zeilenvorschub;
  UNTIL KEYPRESSED;                         { bis eine Taste gedrückt wird }
  WHILE NOT Taster(2) DO Motor(2,Rechts);  { Stift vom Papier abheben }
  Motor(2,Aus);
END. { PROGRAM Kennlinienschreiber }

```

Plotter



„Plotter“ sind Zeichenmaschinen, die sich vom Computer aus steuern lassen. Der Stift des Plotters kann in zwei Richtungen bewegt werden, wodurch sich programmgesteuert beliebige Zeichnungen erzeugen lassen. Das Zeichenpapier ist ca. 15 x 21 cm groß (Format DIN A5 = Schulheft). Das Blatt klebst Du mit Klebefilmstreifen auf der Experimentierplatte fest. Wenn das Papier sehr dünn ist, kannst Du einen Karton darunter legen, oder Du legst einfach mehrere Blätter aufeinander.

Der Schreibstift wird so eingestellt, daß sich seine Spitze in angehobener Position etwa 3 bis 4 mm über dem Papier befindet. In dieser Stellung ist auch der Taster für die Stiftposition betätigt. Bei längeren Zeichenpausen (oder, wenn Du den Schreiber nicht benutzt) die Schutzkappe wieder auf den Stift setzen, sonst trocknet der Stift aus.

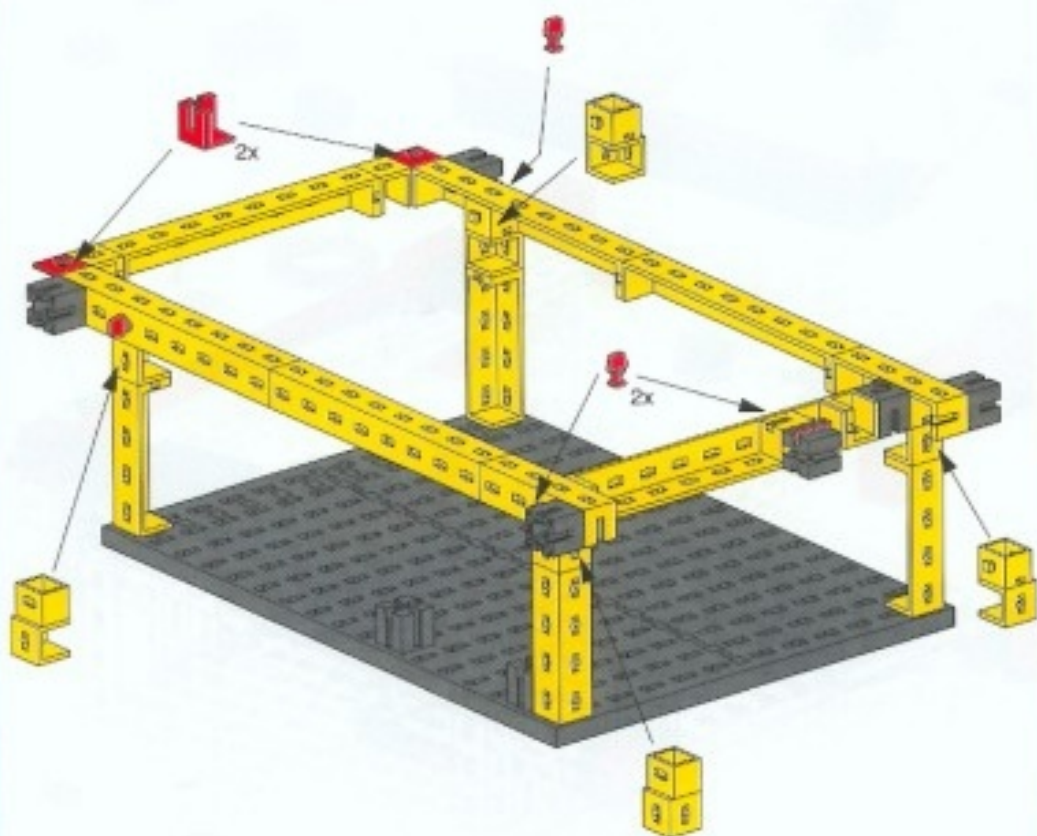
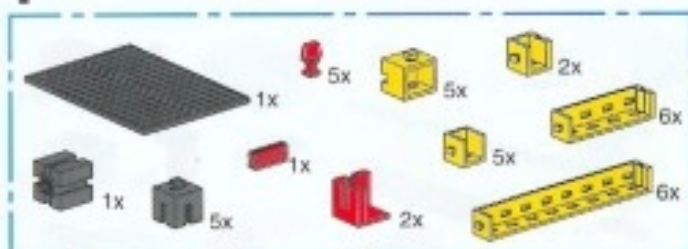
In seiner Arbeitsweise ähnelt der Plotter dem X-t-Schreiber. Die Bewegung des Stiftes in der Längsrichtung (der „X-Achse“) erfolgt durch das bewegen des Schlittens über die beiden mit einem Kettenantrieb verbundenen Schneckenantriebe. Zur Rückmeldung der Schlittenposition dient ein Taster mit Impulscheibe. Ein Strich in Längsrichtung wird so in Teilstücken von ca. 0,5 mm Länge meßbar. Auf dem Schlitten wird der Stift in Querrichtung (der „Y-Achse“) bewegt. Hier wird die Strichlänge durch Referenzimpulse von einer Lichtschranke über der Kette bestimmt. Immer, wenn der Lichtstrahl durch ein Kettenglied un-

terbrochen wird, liefert die Lichtschranke eine „0“, an den freien Stellen eine „1“. Ein Impuls entspricht einer Strichlänge von ca. 4,5 mm. Den Schlitten mußt Du sehr sorgfältig zusammenbauen, damit der Stiftträger leicht und störungsfrei über die Bausteinreihe mit der Zahnstange läuft. Ein Tropfen Haushaltsöl auf die Führungsschnecke ist zu empfehlen.

Wenn der Plotter eingeschaltet wird, steht der Stift irgendwo über der Zeichenfläche. Damit das Programm eine feste Ausgangsstellung für das Zählen der Schritte hat, sind für beide Bewegungsrichtungen Referenztaster angebracht. Das Programm fährt den Stift in Längs- und Querrichtung solange zurück, bis beide Referenztaster geschlossen sind. Diese Position ist dann der Nullpunkt der Zeichnung.

Achte auch darauf, daß alle Nabenmutter der Zahnräder und die Zangenmutter der Schnecken gut festgezogen sind.

1



Verdrahtung

Um die Bewegung des Schlittens nicht zu behindern, müssen die Kabel sorgfältig verlegt und mit Papierdrahtband an mehreren Stellen zusammengebunden werden.

Der Motor an M1 bewegt über die Schnecken den Schlitten in der X-Richtung, der Motor an M2 bewegt den Schreiber in Y-Richtung. Der Motor an M3 hebt und senkt den Stift, und an M4 wird die Lampe der Lichtschranke angeschlossen. Der Taster an E1 wird von der Impulsscheibe an der X-Achse betätigt. Die Taster an E2 (X-Richtung) und E3 (Y-Richtung) melden die Endstellung des Plotters. Der Taster an E4 wird betätigt, wenn der Stift abgehoben ist, und an E5 ist der Fototransistor angeschlossen.

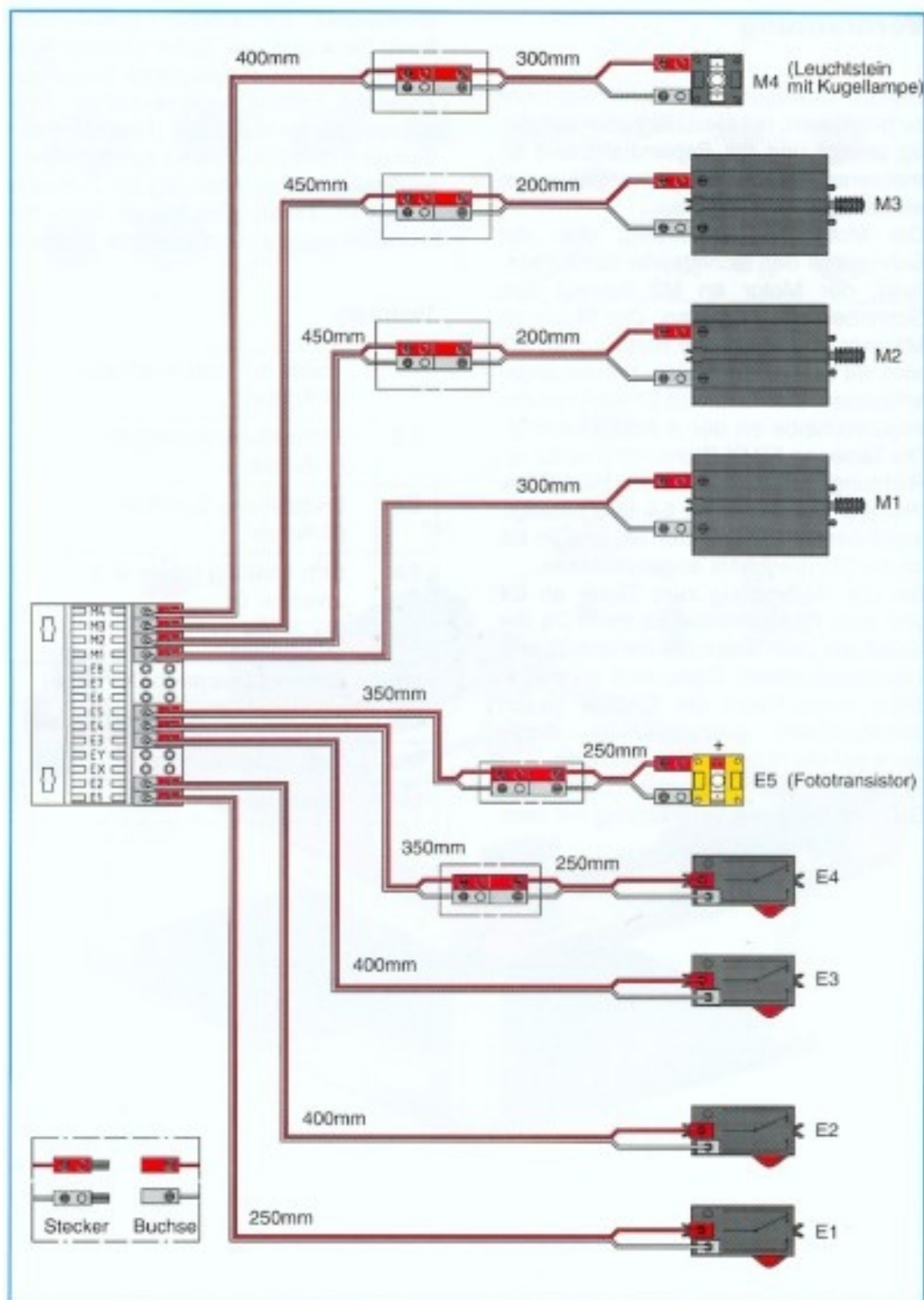
Bei der Verbindung zum Taster an E4 und zum Fototransistor E5 muß Du die Kabel aus zwei Teilen (25 cm und 35 cm) zusammensetzen. Dazu sind an einem Ende eines Kabel die Stecker gegen Steckbuchsen auszutauschen. Achte auch auf die richtigen Kabellängen.

Starte nun das Programm LUCKY LOGIC und teste die Verdrahtung mit dem

Menüpunkt OPTIONEN DIAGNOSE. Beim Betätigen der Taster müssen sich die im Testplan aufgeführten Eingangs-Anzeigen entsprechend verändern. Nun schalte die M-Ausgänge (Motoren oder Lampen) durch Anklicken mit der Maus, wie es in der Beschreibung zur Software erklärt ist. Wenn alles klappt, kann die Software geladen und gestartet werden.

Testplan:

E1	Taster der Impulsscheibe (X-Achse)
E2	Endstellung Schreiber (Y-Achse)
E3	Endstellung Schlitten (X-Achse)
E4	Stift-Stellung (oben = 1, unten = 0)
E5	Fototransistor
M1	Schneckenantrieb (X-Achse)
M2	Zahnstangenantrieb (Y-Achse)
M3	Stift heben/senken
M4	Linienlampe



Software

Die Software zu diesem Modell ist in Turbo-Pascal geschrieben. Das Pascal-Programm befindet sich in der Datei PLOTTER.PAS. Starte nun das Programm PLOTTER.

Die Plottersoftware enthält die wichtigsten Prozeduren zur Steuerung des Plot-

ters: Plotter-Initialisierung, Stift heben/senken und das Anfahren einer beliebigen Position auf dem Papier. Im folgenden Beispiel wird auch eine ganz einfache Zeichnung ausgeführt. Für eigene Zeichnungen brauchst Du nur die Anweisungen im Hauptprogramm zu ändern. Wie der Plotter genau funktioniert, kannst Du aus den Kommentaren im Listing ersehen.

```
PROGRAM Plotter; { Turbo Pascal 5.0 - 6.0, Datei: PLOTTER.PAS }
{.....}
{ Einfache Steuerung des Plotter-Modells. Die Grundprozeduren erlauben Heben }
{ und Senken des Stifts und das Anfahren einer beliebigen Stelle der Zeichen- }
{ fläche. Mit diesen Prozeduren muß dann die gewünschte Zeichnung im Haupt- }
{ programm programmiert werden }
{.....}

USES CRT,
      Fischer;

VAR  X_Pos, Y_Pos,           { Letzte Position des Stifts }
      PC_X, PC_Y             { Merker für Zeichenprozedur }
      X_Richtung,           { Richtung der Stiftbewegung }
      Y_Richtung : Motor_Richtung_Typ;

{.....}

PROCEDURE Stift_nach_Unten;
BEGIN { Stift nach Unten }
  Motor(3,Links);           { Stift bis zur End- }
  REPEAT UNTIL NOT Taster(4); { Stellung absenken }
  Motor(3,Aus);
END; { Stift_nach_Unten }

{.....}

PROCEDURE Stift_nach_Oben;
BEGIN { Stift nach Oben }
  Motor(3,Rechts);          { Stift bis zur End- }
  REPEAT UNTIL Taster(4);   { Stellung anheben }
  Motor(3,Aus);
END; { Stift_nach_Oben }

{.....}

PROCEDURE Plotter_initialisieren;
BEGIN { Plotter initialisieren }
  Motor(4,Links);           { Lampe einschalten }
  Stift_nach_Oben;           { Endstellung X anfahen }
  Motor(1,Rechts);
  REPEAT UNTIL Taster(3);
  Motor(1,Aus);
  Motor(2,Rechts);          { Endstellung Y anfahen }
  REPEAT UNTIL Taster(2);
  Motor(2,Aus);
```

```

DELAY(200);
X_Pos := 0; { Werte initialisieren }
Y_Pos := 0;
PC_X := 1;
PC_Y := 1;
END; { Plotter_initialisieren }

(.....)

PROCEDURE Fahre_nach_XY(X,Y:INTEGER);
{ Eine bestimmte Position auf dem Zeichenblatt anfahren }
{ mit gesenktem Stift = Zeichnen, mit gehobenem Stift = Positionieren }

PROCEDURE X_Position_anfahren; { 1 Schritt in X-Richtung }
BEGIN { X_Position_anfahren }
CASE PC_X OF
1 : BEGIN { Richtung bestimmen }
IF X > X_Pos THEN BEGIN X_Richtung := Links; Motor(2,Links); END
ELSE BEGIN X_Richtung := Rechts; Motor(2,Rechts); END;
INC(PC_X);
END;
2 : IF NOT Taster(5) THEN INC(PC_X); { Ende des Schritts testen }
3 : IF Taster(5) { (Lichtschanke) }
THEN
BEGIN
IF X_Richtung = Links THEN INC(X_Pos)
ELSE DEC(X_Pos);
IF X = X_Pos { Endposition erreicht }
THEN
BEGIN
Motor(2,Aus);
PC_X := 1; { PC_X = 1: Fertig! }
END
ELSE PC_X := 2; { PC_X = 2: noch nicht fertig }
END;
END; { CASE PC_X OF }
END; { X_Position_anfahren }

PROCEDURE Y_Position_anfahren; { 1 Schritt in Y-Richtung }
BEGIN { Y_Position_anfahren }
CASE PC_Y OF
1 : BEGIN { Richtung festlegen }
IF Y > Y_Pos THEN BEGIN Y_Richtung := Links; Motor(1,Links); END
ELSE BEGIN Y_Richtung := Rechts; Motor(1,Rechts); END;
INC(PC_Y);
END;
2 : IF NOT Taster(1) THEN INC(PC_Y); { Ende des Schritts testen }
3 : IF Taster(1) { (Impuls-Taster) }
THEN
BEGIN
IF Y_Richtung = Links THEN INC(Y_Pos)
ELSE DEC(Y_Pos);
IF Y = Y_Pos
THEN
BEGIN
Motor(1,Aus);
PC_Y := 1; { PC_Y = 1: Fertig! }
END
ELSE PC_Y := 2; { PC_Y = 2: noch nicht fertig }
END;
END; { CASE PC_Y OF }
END; { Y_Position_anfahren }

```

```

BEGIN { Fahre_nach_XY }
  REPEAT
    IF X <> X_Pos THEN X_Position_anfahren;      { X-Schritte, bis neue Pos. erreicht }
    IF Y <> Y_Pos THEN Y_Position_anfahren;      { Y-Schritte, bis neue Pos. erreicht }
  UNTIL (X=X_Pos) AND (Y=Y_Pos);                { bis X- und Y-Position erreicht ist }
END; { Fahre_nach_XY }

{.....}

BEGIN { PROGRAM Plotter }
  CLRSCR;
  Plotter_initialisieren;

  { ganz einfache Zeichnung: }
  Fahre_nach_XY(10,10);      { erst ab Pos. (10,10) zeichnen }
  Stift_nach_Unten;

  Fahre_nach_XY(15,10);
  Fahre_nach_XY(15,100);
  Fahre_nach_XY( 5,100);
  Fahre_nach_XY( 5, 10);

  Plotter_initialisieren;      { Plotter in Grundstellung }
END. { PROGRAM Plotter }      { Papier kann entnommen werden }

```



Roboter



Roboter werden vor allem in Fabriken eingesetzt, wo sie oft auch mit anderen automatischen Maschinen zusammenarbeiten. Die meisten von ihnen sind am Boden festgeschraubt und verrichten ihre Arbeit von diesem festen Standort aus. Daß Roboter nicht wie künstliche Menschen aussehen, wurde in der Einleitung schon erwähnt. Sie sind vielmehr in der Bauart der Aufgabe angepaßt: Ein Roboter, der elektronische Schaltungen verdrahtet, ist klein und feingliedrig — ganz anders als ein tonnenschweres Ungetüm, das Automobilkarosserien auf das Fahrwerk setzt.

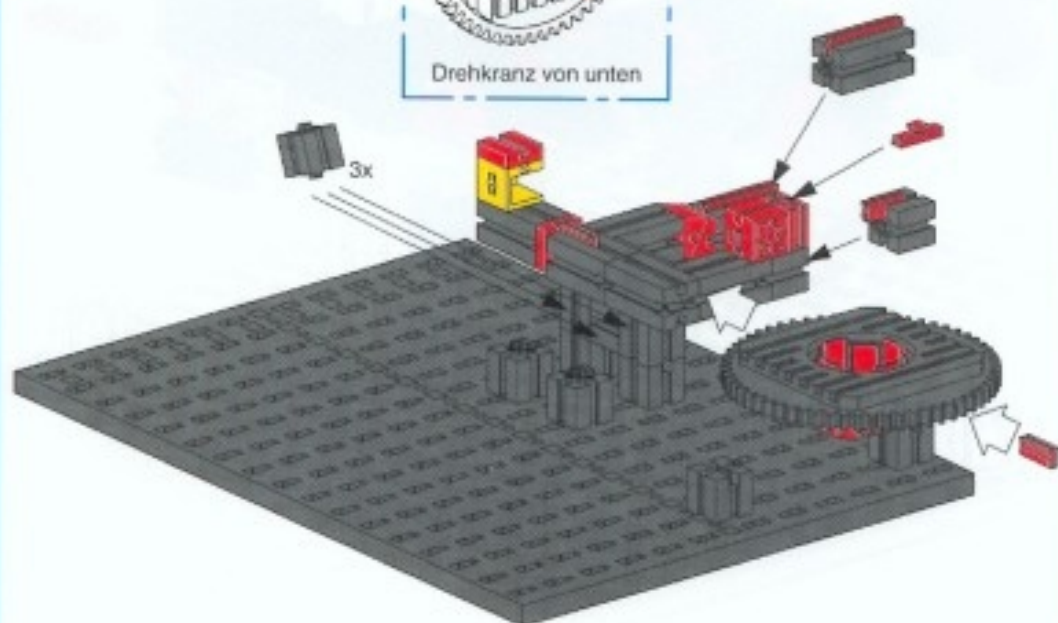
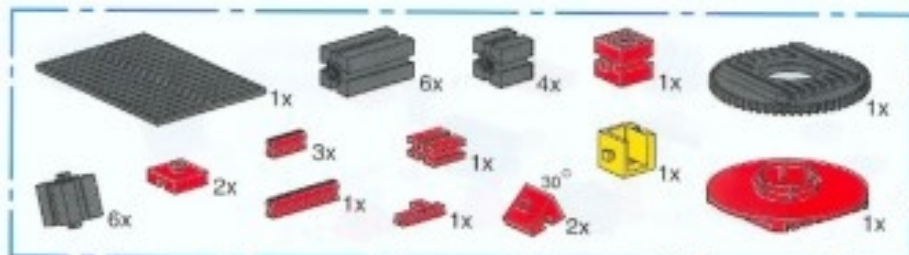
Der fischertechnik-Roboter hat zwei Bewegungsrichtungen, er kann sich drehen und den „Arm“ heben oder senken. Vorne an seinem Arm ist eine Zange, die unterschiedliche Gegenstände ergreifen kann. Im Baukasten befindet sich eine gelbe Rohrhülse mit Boden, die der Greifer gut handhaben kann. Damit die Zange kleine Teile mit glatter Oberfläche besser greifen kann, klebst Du die beiliegenden Haftpunkte auf die Fläche der Zange. Der Arbeitsraum des Greifers ist kreisförmig, er hat einen Radius von etwa 30 cm.

Damit der Computer die Stellung des Drehtellers und des Arms erkennen kann, werden wie in den vorhergehenden Modellen wieder Endstellungstaster und Taster mit Impulsscheiben verwen-

det. Es gibt eine Stellung des gesamten Roboters, in der alle Endstellungstaster (= Referenztaster) betätigt sind. Diese Position muß beim Start des Programms zuerst angefahren werden. Danach kann der Computer die Impulse zählen, die ihm von den Impulstastern an den drei Antriebsmotoren geliefert werden.

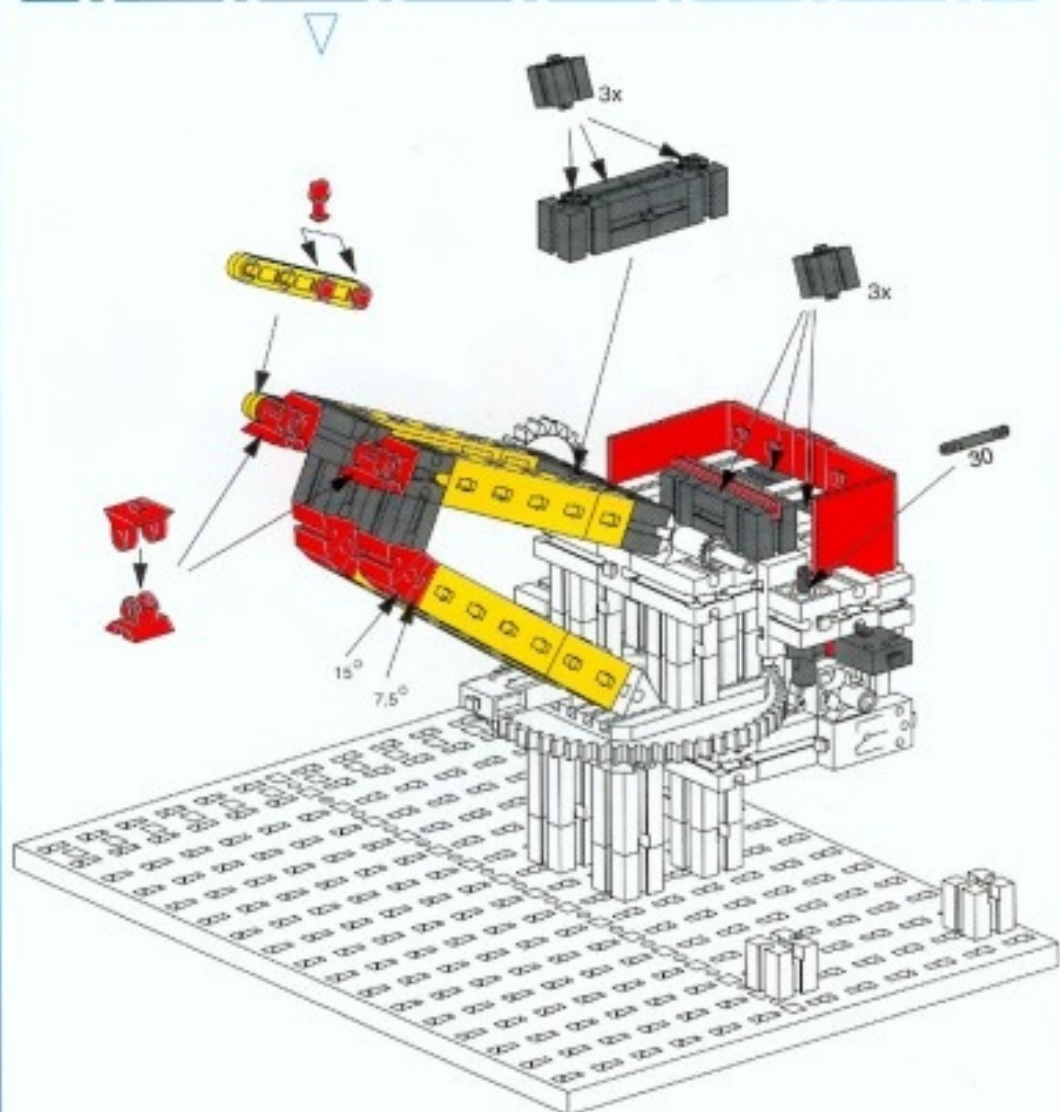
Beim Aufbau ist es besonders wichtig, daß die Naben der Zahnräder gut festgezogen werden. Besonders der über die Kette angetriebene Greifarm ist recht schwer. Damit das Gewicht der Zange etwas ausgeglichen wird, ist der Antriebsmotor für die Zange auch hinten am Greifarm angebaut; die Zange wird dann über eine Welle mit Kreuzgelenk betätigt.

1



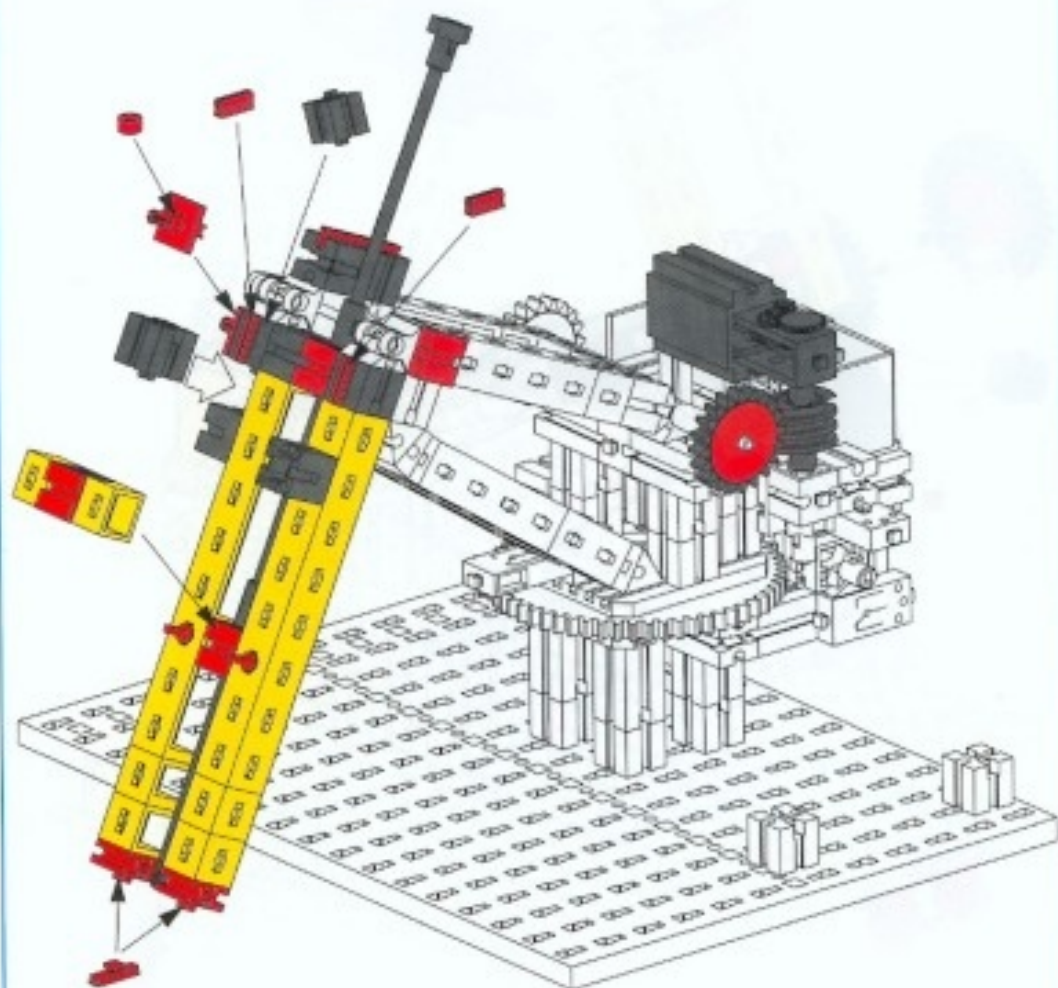
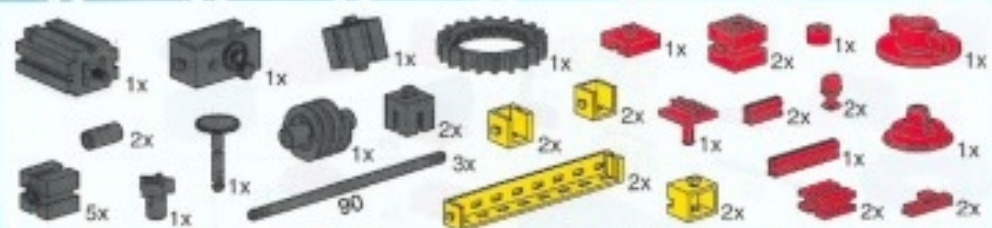
4

340mm



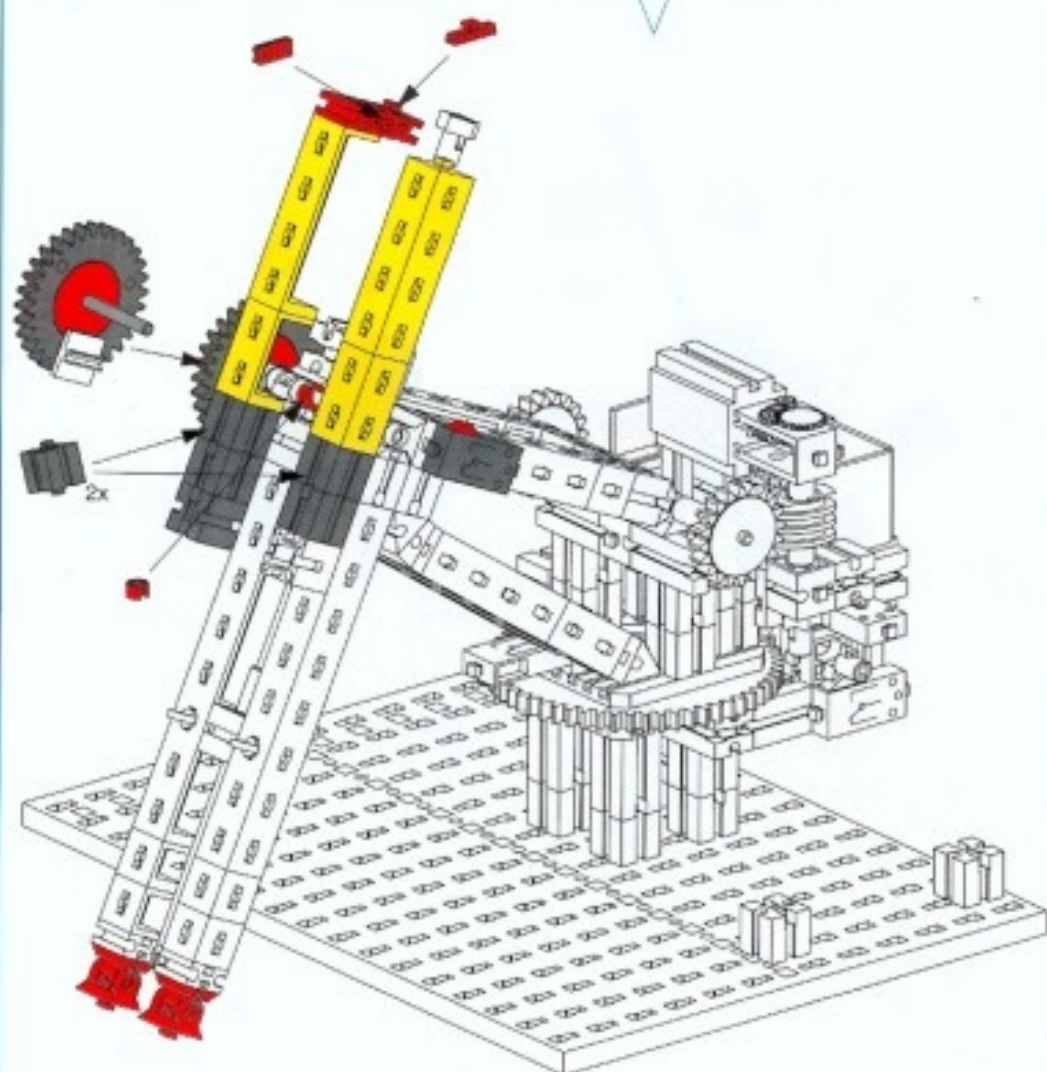
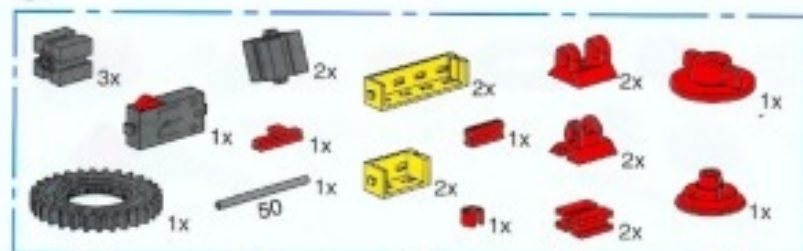
5

2007/01/10

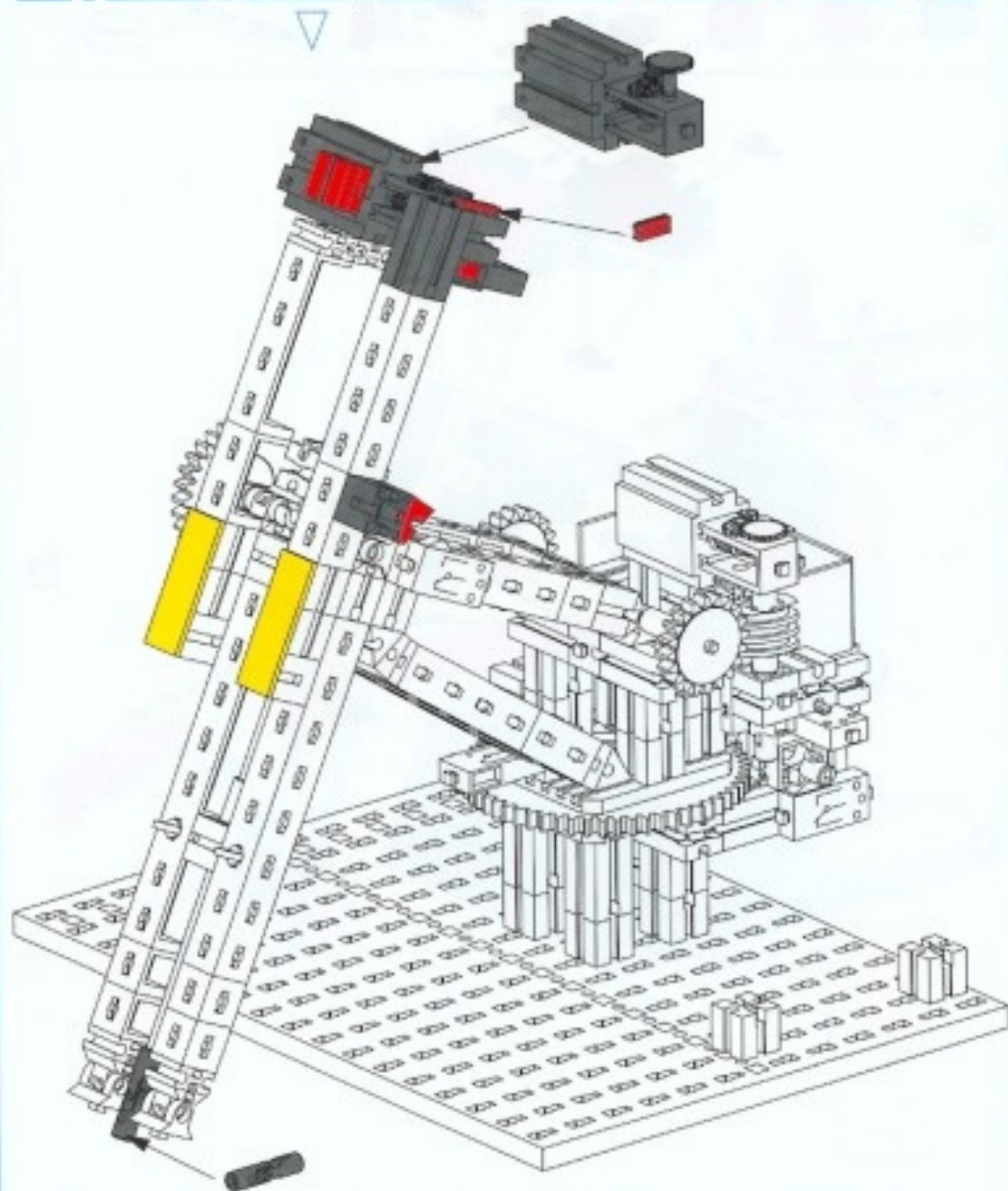
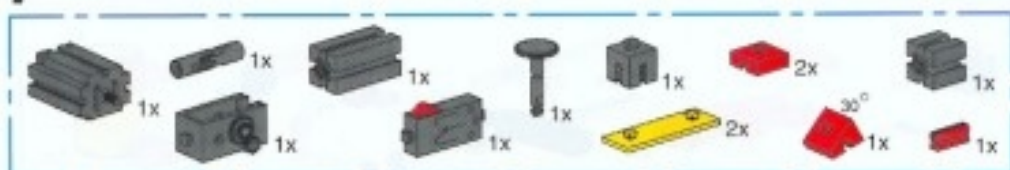


6

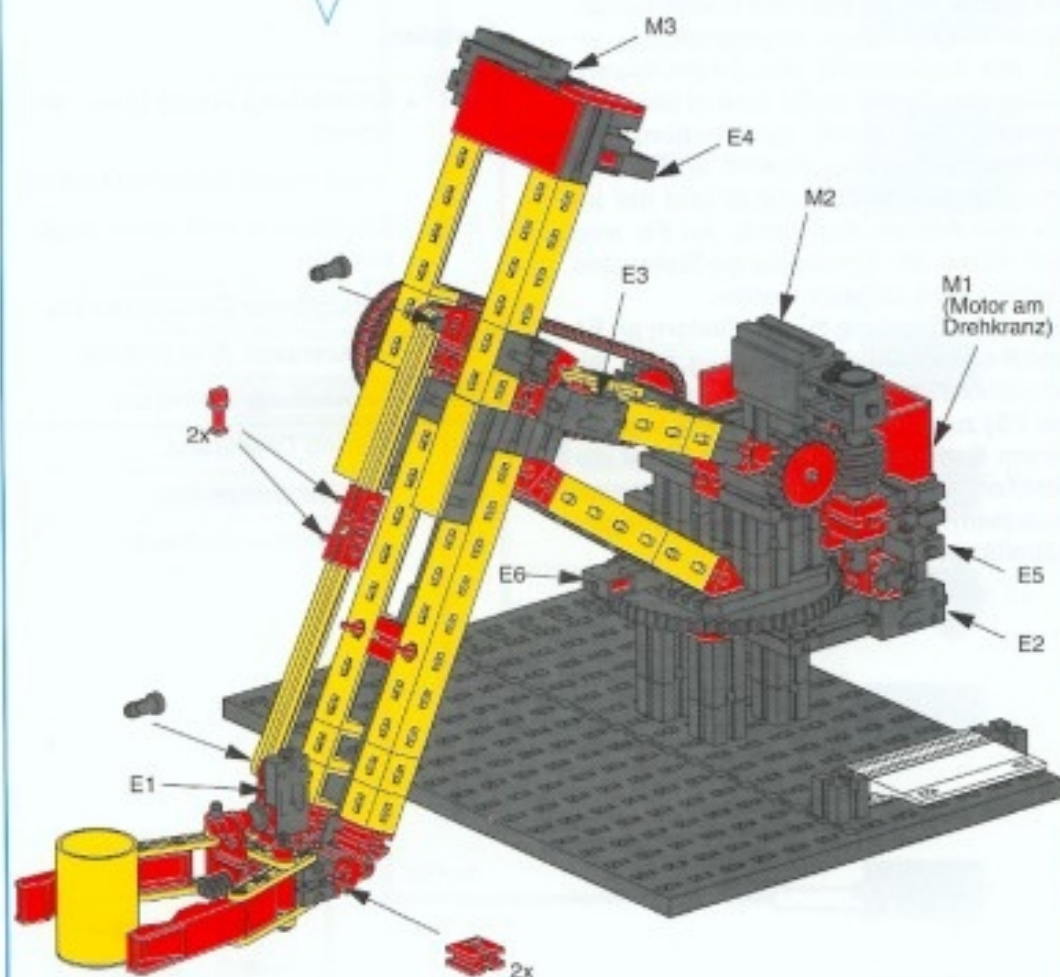
50mm



7



9



Verdrahtung

Damit die Bewegung des Roboters nicht behindert wird, werden bis auf die Zuleitungen zum Drehkranzmotor und zum Impulstaster alle Kabel durch die Mittenöffnung des Drehkranzes geführt.

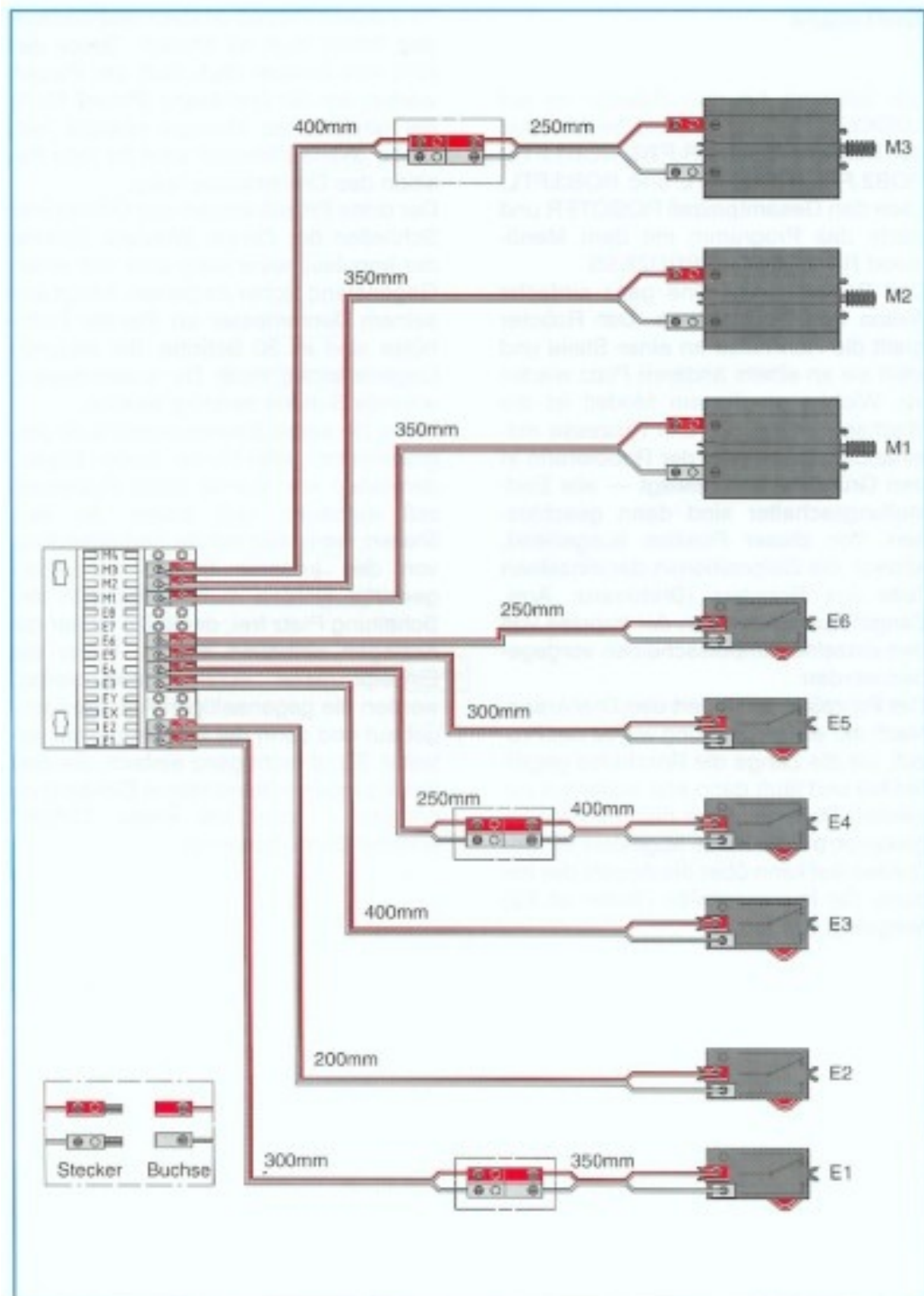
Der Motor an M1 dreht den Roboter, der Motor an M2 hebt und senkt den Arm, und der Motor an M3 öffnet und schließt die Zange. An E1 wird der Endstellungstaster für die Zange angeschlossen, an E2 der Impulstaster des Drehkranzantriebs. Der Taster an E3 wird in der Endstellung des Arms (angehoben) geschlossen. Der Impulstaster an E4 wird vom Zangenantrieb betätigt und der an E5 vom Antrieb des Arms. An E6 wird schließlich der Endstellungs-Taster des Drehkranzes angeschlossen.

Bei der Verbindung zu den Tastern an E1 und E4 muß Du die Kabel aus zwei Teilen (30 cm/35 cm bei E1 und 25 cm/40 cm bei E5) zusammensetzen. Dazu sind an einem Ende eines der beiden Kabel die Stecker gegen Steckbuchsen auszutauschen. Achte auch auf die richtigen Kabellängen.

Starte nun das Programm LUCKY LOGIC und teste die Verdrahtung mit dem Menüpunkt OPTIONEN DIAGNOSE. Beim Betätigen der Taster müssen sich die im Testplan aufgeführten Eingangs-Anzeigen entsprechend verändern. Nun schalte die M-Ausgänge (Motoren oder Lampen) durch Anklicken mit der Maus, wie es in der Beschreibung zur Software erklärt ist. Wenn alles klappt, kann die Software geladen und gestartet werden.

Testplan:

E1	Endstellung Zange (ganz geöffnet)
E2	Impulstaster Drehkranzantrieb
E3	Endstellung Arm (ganz angehoben)
E4	Impulstaster Zangenantrieb
E5	Impulstaster Arm-Antrieb
E6	Endstellung Drehkranz
M1	Antrieb Drehkranz
M2	Arm heben/senken
M3	Zange öffnen/schließen



Software

Die Software für den Roboter ist mit LUCKY LOGIC erstellt. Sie besteht aus den Dateien ROBTER.FTG, ROB1.FTL, ROB2.FTL, ROB2.FTL und ROB3.FTL. Lade den Gesamtprozeß ROBTER und starte das Programm mit dem Menüpunkt RUN GESAMTPROZESS.

Das Beispiel zeigt eine ganz einfache Aktion des Roboterarms. Der Roboter greift die Rohrhülse an einer Stelle und stellt sie an einem anderen Platz wieder ab. Wichtig an diesem Modell ist die Wechselwirkung der drei Prozesse miteinander. Zuerst wird der Roboterarm in den Grundzustand bewegt — alle Endstellungsschalter sind dann geschlossen. Von dieser Position ausgehend, können die Zielpositionen der einzelnen Teile des Roboters (Drehkranz, Arm, Zange) durch Abzählen der Impulse von den einzelnen Impulsscheiben vorgegeben werden.

Der Prozeß Nr. 1 steuert den Drehkranz. Nach der ersten Drehung wartet der Prozeß, bis die Zange die Rohrhülse gegriffen hat und läuft dann erst weiter bis zur zweiten Position. Auch hier wird wieder gewartet, bis die Hülse abgestellt ist. Der Drehwinkel kann über die Anzahl der Impulse der Impulsscheibe (Taster an E2) festgelegt werden.

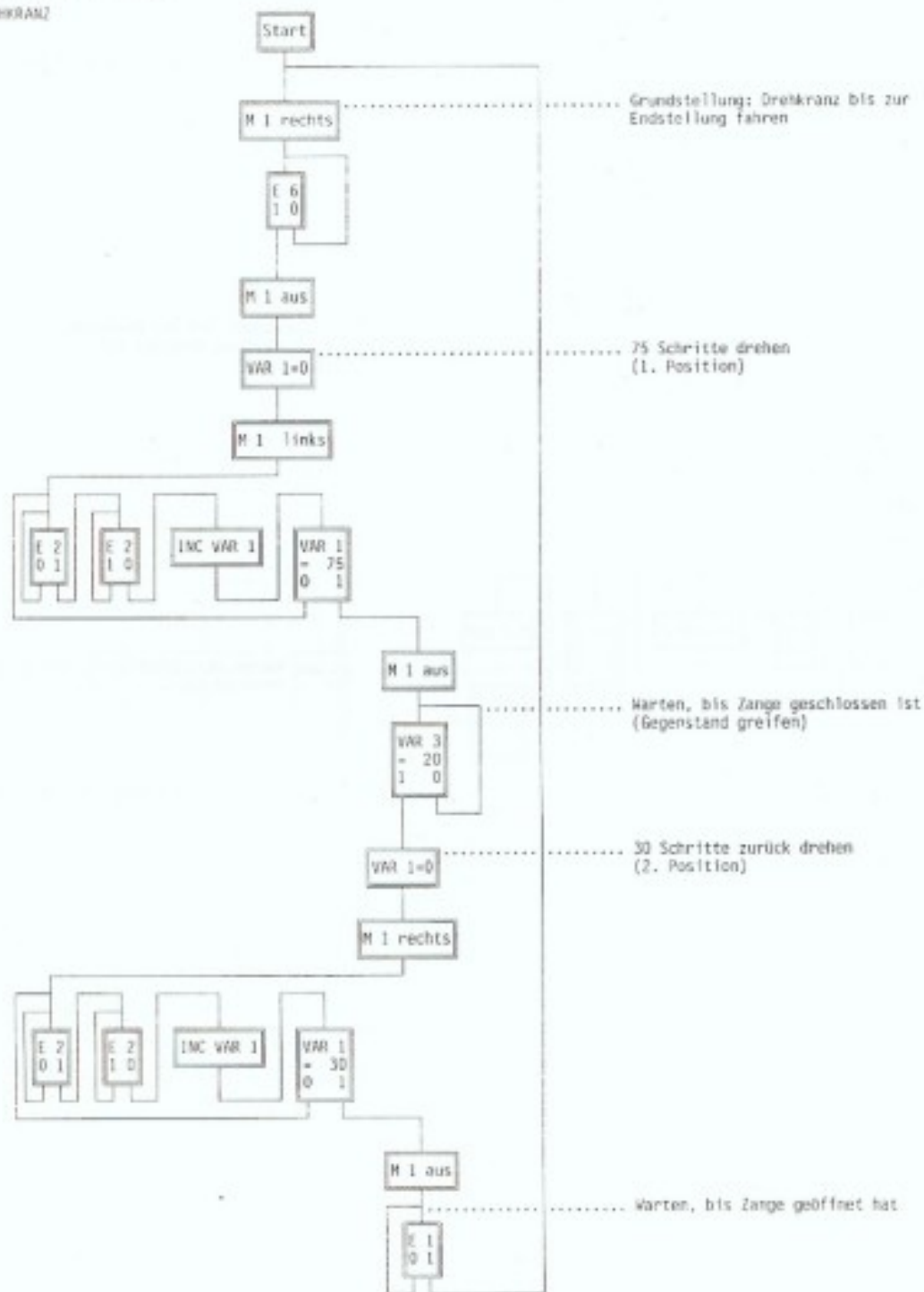
Im zweiten Prozeß (Heben und Senken des Arms) läuft es ähnlich. Bevor der Arm sich senken darf, muß der Prozeß warten, bis der Drehkranz (Prozeß Nr. 1) die gewünschte Position erreicht hat. Diese „Warteschleifen“ sind für jede Position des Drehkranzes nötig.

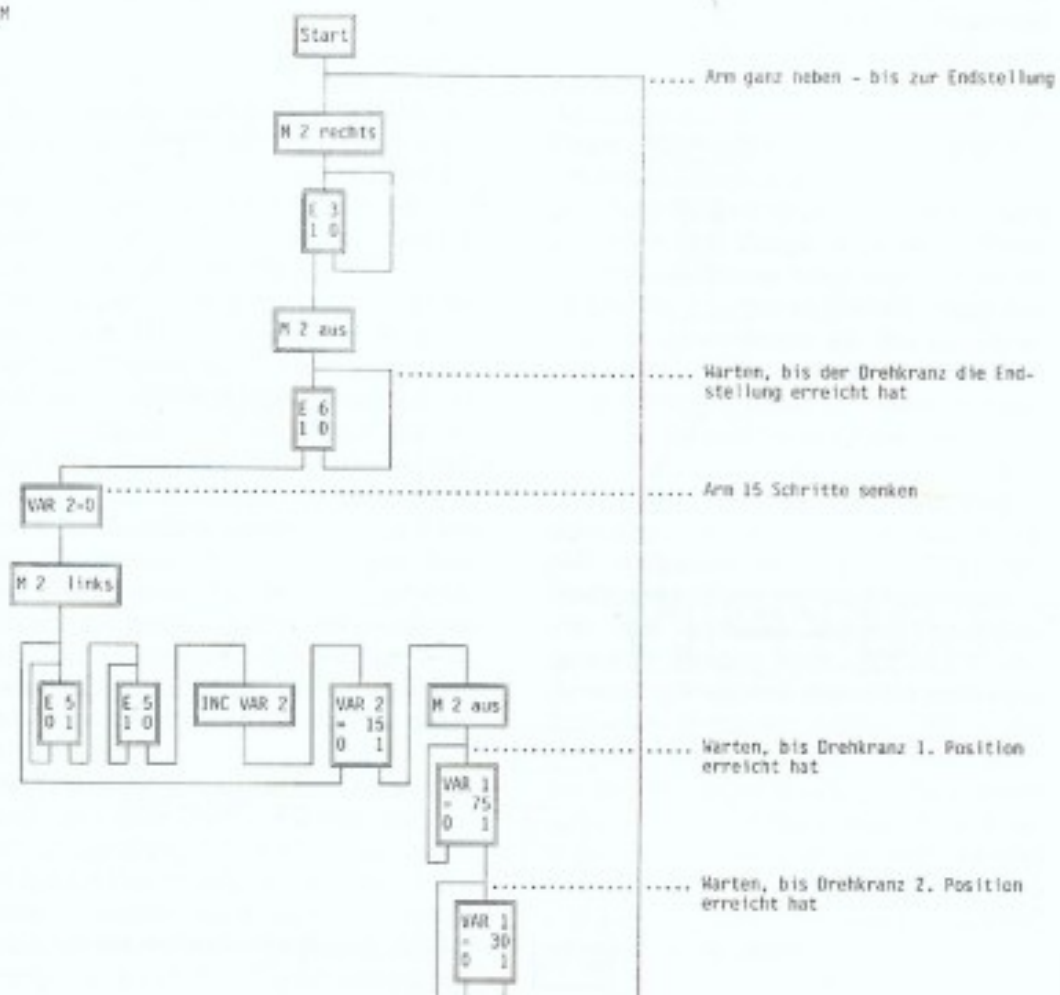
Der dritte Prozeß steuert das Öffnen und Schließen der Zange. Wieviele Schritte der Impulsscheibe nötig sind, um einen Gegenstand sicher zu greifen, hängt von seinem Durchmesser ab. Bei der Rohrhülse sind es 30 Schritte. Bei anderen Gegenständen mußt Du ausprobieren, wieviele Schritte benötigt werden.

Wenn Du selbst Bewegungsabläufe programmierst, gehst Du am besten folgendermaßen vor: Zuerst jeden Einzelprozeß aufbauen und testen. An den Stellen, wo später auf die Fertigmeldung von den anderen beiden Prozessen gewartet werden muß, läßt Du in der Schaltung Platz frei, damit Du später die Abfragen einbauen kannst. Wenn die Einzelprozesse richtig funktionieren, werden die gegenseitigen Abfragen eingebaut und dann der Gesamtprozeß getestet. Es ist nicht ganz einfach, die drei Bewegungen miteinander in Einklang zu bringen — aber mit etwas „Tüfteln“ schaffst Du es bestimmt.

Roboter: Prozess Nr. 1

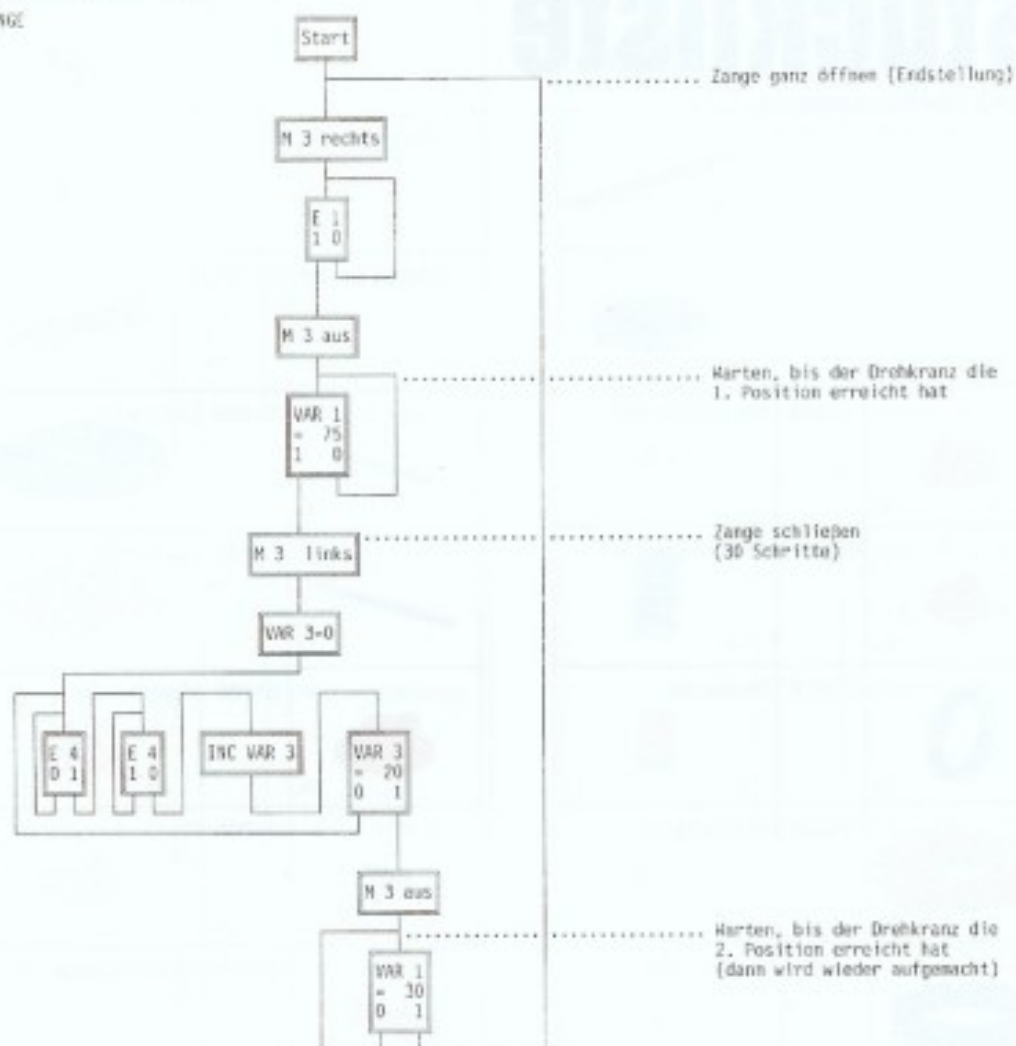
GREIFKRANZ





Roboter: Prozess Nr. 3

ZANGE



Stückliste

(Inhalt des Baukastens)

 4x	31010	Winkelstein 60°
 7x	31011	Winkelstein 7,5°
 6x	31017	Reifen 30
 2x	31019	Drehscheibe
 4x	31021	Zahnrad Z20
 1x	31022	Zahnrad Z40/32
 8x	31023	Klemmbuchse 10
 2x	31031	Achse 110

 1x	31033	Achse 50
 1x	31040	Achse 90
 9x	31058	Nabenmutter
 8x	31060	Verbindungsstück 15
 3x	31061	Verbindungsstück 30
 3x	31078	U-Getriebe
 3x	31082	Zahnrad Z28 m=0,5 mit Rastachse
 1x	31124	Aufnahmeachse






	31253	Flachbuchse grün
5x		
	31254	Flachbuchse rot
5x		
	31330	Verbindungsdübel 45
1x		
	31390	Drehkranz-Oberteil
1x		
	31391	Drehkranz-Unterteil
1x		
	31426	Gelenkwürfelzunge
8x		
	31436	Gelenkwürfelklaue
8x		
	31597	Abstandsring
1x		
	31667	Lasche 15
2x		
	31690	Rastachse
1x		
	31707	Deckel
1x		

	31762	Rast-Raupenbelag
67x		
	31765	I-Strebe 90
2x		
	31779	Kettenrad
2x		
	31848	Strebenadapter
4x		
	31892	Kunststoff-Feder 26
1x		
	31915	Zangenmutter
2x		
	31981	Winkelstein 15°
5x		
	31982	Federnocken
13x		
	31983	Hülse 15
3x		
	32064	Baustein 15 mit Bohrung
12x		
	32071	Winkelstein 7,5°
4x		

 1x	32085	Rollenbock
 1x	32119	Flachbandkabel mit Stecker 20 adrig
 1x	32121	Schreibstift Fasermine
 2x	32200	I-Strebe 75
 3x	32293	S-Motor 6-9V=
 2x	32316	Verbindungsstopfen
 1x	32321	Baustein 15 mit Ansenkung
 4x	32330	Bauplatte 15x30x3,75 mit Nut
 2x	32355	Achse 315
 1x	32657	Steckerbuchsen- Gehäuse 28-polig
ohne Abbildung	32793	Teilebeutel zu 28-poligem Stecker

 12x	32850	Riegelstein 15x15
 2x	32870	Clipsachse
 22x	32879	Baustein 30
 27x	32881	Baustein 15
 21x	32882	Baustein 15 mit 2 Zapfen
 1x	32985	Bauplatte Computing
 4x	33582	Tesa Klebepunkt
 2x	35018	Gelenkschraube
 2x	35019	Gelenkmutter
 9x	35031	Flachnabenzange
 10x	35049	Baustein 15x30x5

	35053	Winkelträger 15
5x		
	35060	I-Strebe 120 mit Loch
2x		
	35063	Rastachse 30
3x		
	35065	Rastachse 60
1x		
	35066	Rastachse 90
3x		
	35069	Seilwinden-Gestell
1x		
	35072	Rastschnecke
2x		
	35073	Rastkupplung
2x		
	35076	Statik-Stein
5x		
	35078	Rastleuchtkappe orange
4x		
	35112	Ritzel Z10 M1,5
1x		

	35113	Spannzange
1x		
	35116	Flachsteckerhülse rot
22x		
	35117	Flachsteckerhülse grün
22x		
	35389	Rohrhülse
1x		
	35404	V-Achse 4x17
1x		
	35414	V-Achse 4x98
1x		
	35678	O-Ring 54x3
2x		
	35696	Achse 170
2x		
	35697	Achse 200
2x		
	35796	Druckfeder 30x5x0,3
1x		
	35945	Rast-Ritzel Z10 m=1,5
2x		

	35969	Reed-Kontakthalter
7x		
	35970	Schiebeteil
1x		
	35971	Kardangel
2x		
	35972	Kardanwürfel
1x		
	35973	Schneckenmutter Modul 1
1x		
	35975	Statikadapter
2x		
	35977	Rastschnecke Modul 1
1x		
	35978	Zangenteil
2x		
	35984	Rohrhülse klarsichtig
3x		
	35995	Impulszahnrad
3x		
	36050	Gelenkstein-Feder
2x		

	36067	Litze 2-adrig rot/grün
1x		
	36132	Puffer
1x		
	36134	Fototransistor mit Sockel
2x		
	36223	Feder-Gelenkstein
6x		
	36248	Rastkettenglied
134x		
	36264	Zahnrad Z30
1x		
	35294	Winkelträger
6x		
	36297	Winkelträger 60
7x		
	36298	Winkelträger 15 mit 2 Zapfen
5x		
	36299	Winkelträger30
4x		
	36323	S-Riegel 4mm
7x		





 4x	36324	S-Riegel 6mm
 6x	36334	Riegelscheibe
 1x	36437	NTC-Widerstand 1,5 KOhm 0,75
 1x	36443	Schraubendreher
 2x	36457	S-Riegel 8mm
 4x	36532	Störlichtkappe Bohrung 6,0
 1x	36573	Rad 14
 19x	37237	Baustein 5
 6x	37238	Baustein 5 mit 2 Zapfen
 1x	37272	ZG-Getriebe
 3x	37351	Hub-Zahnstange 60 mit Rändelstift

 1x	37457	Hub-Zahnstange 30 mit Rändelstift
 9x	37468	Baustein 7,5
 2x	37636	Rollenlager
 6x	37679	Klemmbuchse 5
 2x	37727	Schaltscheibe
 6x	37783	Mini-Taster
 2x	37858	Klemmbare Schnecke
 4x	37869	Kugelstecklampe
 2x	37875	Linsenstecklampe
 4x	37925	Schnecken-Mutter
 14x	37926	Schneckenteil

	38213	Steckerstift
44x		
	38214	Schraube
44x		
	38216	Leuchstein mit Steckfassung
2x		
	38236	Bauplatte 15x15 mit Zapfen
1x		
	38237	Bauplatte 30x45 Z3
1x		
	38238	Bauplatte 30x45 Z3
3x		
	38240	Baustein V15 Eck
4x		
	38241	Bauplatte 15x30 mit Zapfen
4x		
	38242	Bauplatte 15x45 Z2
2x		
	38243	Bauplatte 15x45 Z2
2x		
	38244	Bauplatte 15x75
1x		

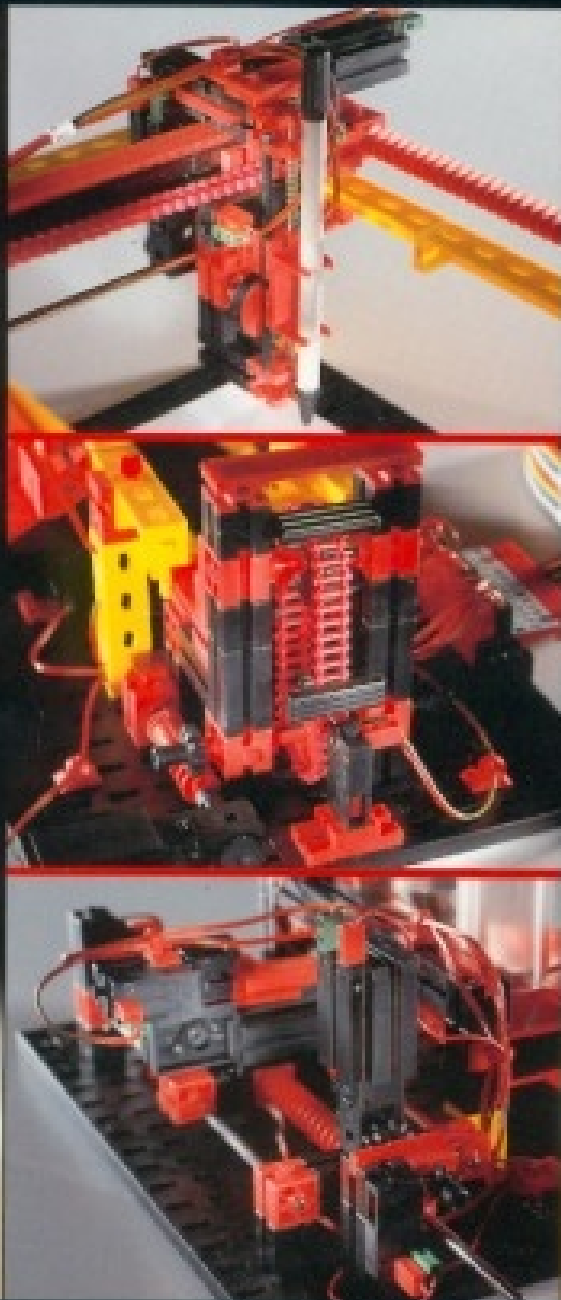
	38245	Bauplatte 15x90
3x		
	38246	Bauplate 15x15
1x		
	38248	Bauplatte 30x45
1x		
	38249	Bauplatte V30x60 Z2
4x		
	38251	Bauplatte 30x90
5x		
	38253	Kupplungsstück 2
2x		
	38259	Bauplatte 30x90
3x		
	38260	Kupplungsstück 30
2x		
	38267	Bauplatte 15x30
4x		
	38268	Bauplatte 15x30
4x		
	38277	Bauplatte 15x45 mit 2x2 Zapfen
1x		

 2x	38416	K-Achse
 7x	38423	Winkelstein 10x15x15
 4x	38428	Bauplatte 15x30x5 mit 3 Nuten
 2x	38432	Bauplatte 30x60
 3x	38464	Bauplatte 15x60

 2x	38474	Bauplatte 15x60
 2x	38485	Bauplatte 30x90
 6x	38541	I-Strebe 45 mit Loch
 4x	38544	Strebe 15

Technische Änderungen vorbehalten

fischertechnik®



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

Hinweise und Ergänzungen zum Experimentierbuch PROFI computing

Software

Die fischertechnik-Software beinhaltet:

1. Lucky Logic Programme
2. Turbo Pascal Programme
3. Basic Treiber

Spezifische Programmhinweise befinden sich im jeweiligen Modellkapitel des Experimentierbuches.

Sortieranlage

Mit der Sortieranlage können Bausteine 7,5 und Bausteine 15 sortiert werden. Entsprechend der unterschiedlichen Bausteinhöhe müssen die beiden Fototransistoren justiert werden.

Stromversorgung

Wir empfehlen den fischertechnik Power Supply Art.-Nr. 30 180 oder das Computing Netzgerät Art.-Nr. 30 579.

Paketwendeanlage

Anstatt der gelben Bauplatten 30 x 45 müssen für das „Paket“ die drei weißen Bauplatten 30 x 45 und die schwarze Bauplatte 30 x 45 verwendet werden.

Plotter

In der auf Seite 132 dargestellten Bauphase werden die beiden Taster E1 und E2 eingebaut.