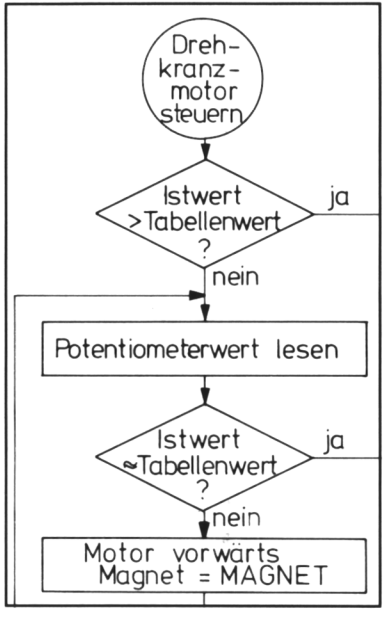
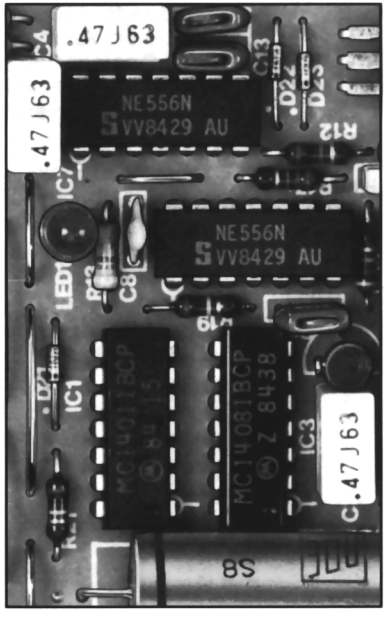
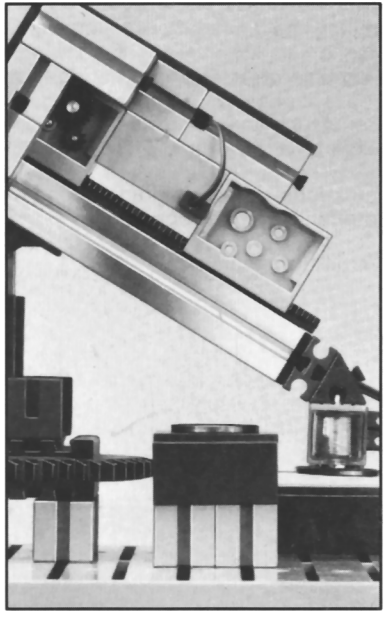
fischertechnik® COMPUTING

Programmieranleitung







Inhalt

Einführung	3
Kabelkonfektionierung	4
Wie wird programmiert?	6
Ampelanlage	8
Werkzeugmaschine	10
Aufzug	12
Ein stufenloser Positionsmelder	13
Antennenrotor	14
Sortieranlage	15
Turm von Hanoi	17
Teach-in Roboter	19
Grafiktablett	21
Plotter	24
Solarzellennachführung	26
Übersicht fischertechnik computing	29
Noch ein Roboter	30

fischertechnik computing

Lieber fischertechnik-Freund,

Um mit einem Computer, in Erweiterung seiner Einsatzmöglichkeiten, auch technische Modelle anzusteuern, wurde der fischertechnik computing Baukasten entwickelt. Es ist jetzt möglich, technische Funktionen und Vorgänge zu simulieren und entsprechende Aufgaben zu lösen.

Zwei Motoren und ein Elektromagnet sind in dem Baukasten enthalten – damit verleihen Sie Ihrem Computer „Arme, Beine und Hände". Blättern Sie das Anleitungsbuch durch; die beiden Robotermodelle verdeutlichen Ihnen, was damit gemeint ist. Drei Lampen zeigen Ihnen an, welche Aktionen der Computer unternimmt. Hinzu kommen als „Sinnesorgane" 8 Taster und zwei Potentiometer, ausreichend für die verschiedensten Rückmeldungen aus den Modellen. Die Taster können auch zum Aufbau von Kommandofeldern dienen.

Außer Robotern können auch noch eine Reihe weiterer Modelle gebaut werden.
Am Anfang stehen einfache Modelle, z.B. eine Ampelanlage, Werkzeugmaschine und ein Aufzug. Mit diesen Modellen erfolgt eine schrittweise Einführung in die Welt der Computersteuerungen. Einfache, klar gegliederte BASIC-Programme geben dem Anfänger eine Hilfe und dienen dem erfahrenen Programmierer als Ausgangspunkt für eigene Entwicklungen.
Doch auch die anspruchsvollen Modelle sind mit Beispielprogrammen versehen. So erlaubt z.B. der Plotter die Erstellung von Grafiken, wie sie bei Meßwerterfassungen anfallen. Den umgekehrten Weg können Sie mit Hilfe des Grafiktabletts beschreiten. Der Computer registriert Ihre Bewegungen des Griffels und übernimmt bei Tastendruck die Positionsdaten. Sie können mit einem geeigneten Programm auf dem Bildschirm malen, konstruieren oder aber, wie bei einer Menükarte, Programmstücke auswählen.

Zurück zu den Maschinen – die Solarzellennachführanlage bietet die Möglichkeit, eine Solarzelle frontal zu jeder Himmelsrichtung einzustellen. Ein entsprechendes Programm kann somit die Solarzelle auf optimale Energieeinstrahlung ausrichten.

Ebenfalls ein Modell aus der Technik ist die Sortieranlage. Sie unterscheidet Bausteine 30 und 15 und sortiert sie in zwei verschiedene Auffangbehälter.

Ein Teil der Modelle aus „fischertechnik computing" sind in Zusammenarbeit mit dem Norddeutschen Rundfunk für die Fernsehserie „Einführung in die Mikroelektronik" entstanden.

Selbstverständlich können Sie die Teile aus fischertechnik computing mit allen anderen fischertechnik-Komponenten kombinieren und Ihrer Phantasie freien Lauf lassen. Die Seite 31 soll Ihnen einen kleinen Denkanstoß geben.

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

Ihr

Kabelkonfektionierung

Halt – bevor Sie anfangen, das erste fischertechnik computing Modell aufzubauen, sollten Sie sich einige Dinge zurechtlegen. Insbesondere muß das beiliegende 20adrige Flachbandkabel konfektioniert werden.

Benutzen Sie das fischertechnik Interface? Wenn ja, gilt das folgend Gesagte. Ansonsten sollten Sie an dieser Stelle Ihre Interface-Anleitung aufschlagen und auf abweichende Anleitungen achten. In diesem Fall gilt das für Ihr Interface Gesagte.

Die Adern schwarz 2 (am unteren Rand, Abb. 1) bis gelb 1 werden von rechts um ca. 50 cm gekürzt. Ein Hinweis: das aufgeklappte Anleitungsbuch hat etwa eine Breite von 50 cm. Anschließend trennen Sie von rechts das Flachbandkabel in Einzeladern auf, so daß das linke Ende auf einer Strecke von 50 cm zusammenhängend bleibt. Die Adern schwarz 2 – weiß 2, grau 2 – violett 2, blau 2 – grün 2 und gelb 2 – orange 2 können Sie jeweils als Paar zusammenlassen.

Das abgeschnittene Kabelende bitte nicht fortwerfen. Hieraus werden je nach Modell kurze Verzweigungskabel angefertigt. Das große Kabelstück paßt jedoch für alle fischertechnik computing Modelle. Wenn das jeweilige Modell nicht alle Adern des Kabelbaums benötigt, so trennen Sie den Kabelbaum bitte nicht weiter auf und kürzen ihn auch nicht. Bei einem anderen Modell könnten Sie das Abgeschnittene benötigen.

Im nächsten Schritt sollten Sie die Kabeladern etwa 3–5 mm vorsichtig abisolieren, ohne die feinen Drähtchen der Litze zu verletzen.

Das gleiche tun Sie mit den kurzen Kabelenden, die an die Anschlußfahnen der Potentiometer angelötet sind. Auch das zweiadrige rotgrüne Kabel für den Netzgeräteanschluß wird so vorbereitet.

Das Flachbandkabel und das Netzgerätekabel werden mit Steckern, die Potentiometer mit Buchsen versehen. Dazu wird die Litze verdrillt und auf die Isolation umgebogen. Lösen Sie die Steckerschraube und schieben Sie das Kabelende in die Hülse ein. Schraube wieder anziehen, aber nicht so fest, daß das Kabel abgequetscht wird (Abb. 2).

Wie das computerseitige Ende des Kabelbaums aussehen muß, hängt von dem Computerausgang ab. Das Anschlußkabel ist mit einem zwanzigpoligen Steckverbinder versehen. Wenn Sie ein fischertechnik-Interface besitzen, brauchen Sie diesen Steckverbinder nur am Interface einzustecken und alle Leitungen sind korrekt nach Verdrahtungsplan angeschlossen.

Die Potentiometer werden in der Potentiometerhalterung mit Hilfe der Unterlegscheibe und Schraube montiert (Abb. 3).

Abschließend sollten Sie einen Funktionstest des Kabels und des Interface durchführen. Verwenden Sie dazu das mit dem fischertechnik Interface mitgelieferte Diagnoseprogramm. Dieses zeigt Ihnen den Zustand aller Eingabeleitungen an und gestattet, alle vier Motorkanäle anzusteuern. Führen Sie folgenden Test durch:

- Interface an Computer anschließen. (Computer vorher ausschalten!)
- Einen mini-Motor am Ausgang M1 anschließen.
- Taster zwischen Eingang E1 und +5V anschließen.
- Die beiden Potentiometer zwischen EX bzw. EY und +5V anschließen.
- Zur Spannungsversorgung Netzgerät anschließen (auf die Polarität achten).
- Diagnoseprogramm eingeben oder von Diskette laden.
- Diagnoseprogramm starten.

Flachbandkabelkonfektionierung

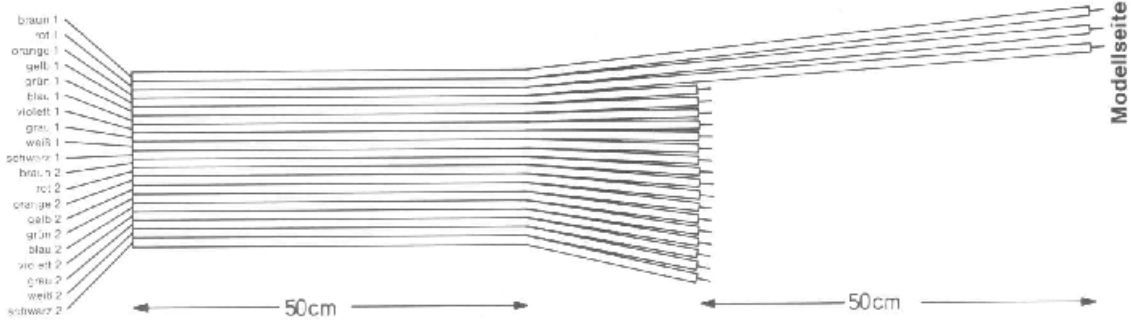


Abb. 1

Nun muß die rote Leuchtdiode am Interface brennen. Falls nicht: Polung des Netzgerätes überprüfen. Sitzt der Stecker im Computeranschluß richtig?
Auf dem Bildschirm werden nun die Zustände der 10 Eingänge (E1, .., E8, EX, EY) und der 4 Ausgänge (M1, ..., M4) angezeigt.
Bei Betätigen des Tasters muß sich der Zustand von E1 ändern. Entsprechend testet man nach Umstekken des Tasters die Eingänge E2, ..., E8.
Beim Verstellen der Potentiometer muß der Wert bei EX bzw. EY sich im Bereich von 0 bis 255 verändern. 0 und 255 werden möglicherweise nicht ganz erreicht.
Falls Fehler auftreten: Verdrahtung überprüfen, evtl. ist das Kabel defekt, versuchsweise Potentiometer austauschen.
Steuern Sie nun, wie von dem Programm gemeldet, den Motor rechts- und linksherum und schalten Sie ihn wieder aus.
Wenn auch dieser Testpunkt erfolgreich absolviert ist, stecken Sie den Motor nun an M2 und die weiteren Motorkanäle und wiederholen den Test.
Falls gar nichts läuft: Motor direkt ans Netzgerät anschließen und Funktion überprüfen; Verdrahtung überprüfen.

Grundsätzlich gilt: Wenn manche Eingänge funktionieren, andere nicht, oder der Rechtslauf geht, Linkslauf nicht, so besteht Grund zu der Annahme, daß das Interface defekt ist. Solange aber „gar nichts läuft", sollte man zunächst noch einmal seinen Aufbau gründlich überprüfen.
Abschließend sollten Sie alle weiteren Kabel auf Durchgang prüfen. Wenn Sie keinen Durchgangsprüfer oder Ohmmeter besitzen, können Sie wie in Abb. 4 einen Aufbau mit einer Lampe und einer Stromquelle benutzen.
Nun noch ein Hinweis zu den Schaltplänen: Soweit wie möglich werden immer die gleichen Kabeladern für die gleichen Funktionen benutzt. Dabei ist zu unterscheiden zwischen Funktionen, die für ein Modell unbedingt notwendig sind, wie z.B. der Magnet des Teach-in Roboters. Andere Funktionen sind wünschenswert, aber nicht unbedingt notwendig. So können Sie Kommandotasten sowohl mit den fischertechnik mini-Tastern als auch mit der Tastatur des Computers realisieren. Im zweiten Fall können die mini-Taster entfallen. Die Schaltpläne und die Bauphasen entsprechen jedoch immer dem Maximalausbau.

Steckermontage

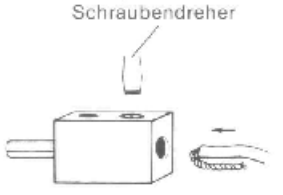


Abb. 2

Potimontage

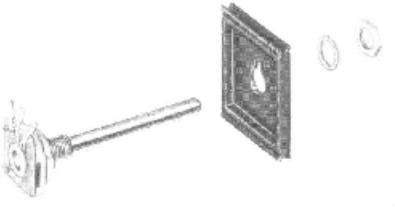


Abb. 3

Durchgangsprüfung

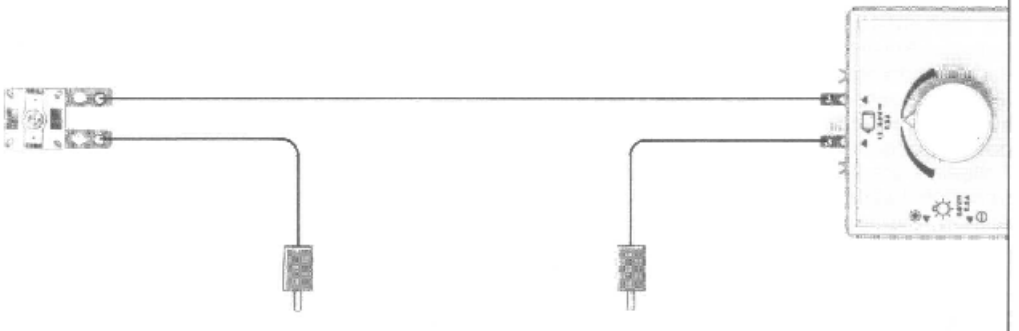


Abb. 4

Wie wird programmiert?

Wer sich schon einmal mit dem Gedanken befaßt hat, irgendwelche Geräte oder Modelle mit dem Computer zu steuern, wird aus eigener Erfahrung wissen oder von anderen Computerfreunden gehört haben, daß dies alles gar nicht so einfach sei. Man brauche eine genaue Kenntnis des Computers, des Mikroprozessors und Ein-/Ausgabebausteins sowie der Maschinensprache für diese Aufgabe.
Bislang hatten Sie damit recht, und dadurch wurde leider auch mancher von diesem interessanten Kapitel der Computerei abgehalten. Dies muß nun nicht mehr der Fall sein.

Dem fischertechnik Interface liegen eine ausführliche Anleitung und Programme auf Diskette bei. Davon ist ein Programm besonders wichtig, das Grundprogramm. Wenn Sie in der Anleitung des Interface oder im Inhaltsverzeichnis der Diskette nachschauen, so werden Sie es als GRUNDPR... vorfinden. Die Punkte stehen für eine Namenserweiterung, die den geeigneten Computertyp angibt, z.B. GRUNDPR .64 für den Commodore 64.

Schließen Sie gleich einmal das Interface, wie in der Anleitung beschrieben, an den Computer an. Wir wiederholen es noch einmal, weil es so wichtig ist: Der Computer muß dabei ausgeschaltet sein!
Schalten Sie den Computer wieder ein und laden Sie das Grundprogramm (Details in der Anleitung Interface). Wenn Sie das Kommando RUN eingeben, so wird sich nach einer kurzen Denkpause der Computer wieder mit seinem Bereitzeichen melden. Äußerlich scheint sich nichts geändert zu haben. Dennoch besitzt der Computer nach Ausführung des Grundprogramms einige neue Befehle.
Diese Befehle sind sowohl auf das fischertechnik Interface als auch den Computer abgestimmt. Deshalb gibt es für jeden Computertyp ein eigenes Grundprogramm. In dem Grundprogramm stecken nun die oben erwähnten Details über die Computerbausteine in Form von Maschinensprache.

Statt detaillierten Computerkenntnissen brauchen Sie nun nur noch die folgend beschriebenen BASIC-Befehle zu beherrschen:

Motorausgang M1 wird angesteuert mit:

SYS M1, EIN **SYS M1, AUS**
SYS M1, RECHTS **SYS M1, LINKS**
(**SYS M1, EIN** bewirkt ebenfalls immer Rechtslauf)

Die entsprechenden Befehle mit M2 ... M4 steuern die 3 anderen Ausgänge.

Die 10 Eingänge werden mit Hilfe der USR-Funktion des Basic-Interpreters abgefragt; die Funktion

USR (E1)

ist 1, wenn der Eingang E1 des Interface mit + verbunden ist, sonst zeigt USR (E1) den Wert 0, Entsprechend erhält man mit USR (E2) ... USR (E8) die Zustände der übrigen Digitaleingänge.

Die Analogeingänge EX und EY werden über je ein Potentiometer (4.7 k) mit + verbunden. Die Funktionen

USR (EX), USR (EY)

haben dann einen Wert zwischen 0 und 255, je nach Stellung der Potentiometer.

Wird z.B. ein Roboterarm von einem Motor angetrieben und synchron mit der Bewegung des Arms das Potentiometer EX verstellt, so kann das Programm, indem es immer wieder die Funktion

USR (EX)

aufruft, die Bewegung des Roboters genau verfolgen.

Der letzte der neuen Befehle ist

SYS INIT

Dieser wird benutzt, um das Interface in einen wohldefinierten Anfangszustand zu versetzen. Er kann auch benutzt werden, wenn alle Motorkanäle mit einem Male abgeschaltet werden sollen.
Bevor Sie die Befehle nun verwenden, empfiehlt sich wieder ein Blick in Ihre Interface-Anleitung. Statt des Befehlswortes SYS muß bei einigen Computern das Befehlswort CALL und statt der Funktion USR (...) die Funktion FN USR (...) verwendet werden.

Die hier beschriebenen Programme sind auf den weit verbreiteten Computer Commodore 64 angepaßt. Programmzeilen, die auf einem anderen Computer anders eingegeben werden müssen, haben wir im folgenden mit einem Sternchen * gekennzeichnet. Neben den schon erwähnten Befehlen ist dies auch der Befehl zum Löschen des Bildschirms.

Wir stellen hier kurz gegenüber:

Commodore : PRINT CHR$ (147)
IBM : CLS
Atari : CLS
Amiga : CLS

Schauen Sie in der Anleitung Ihres Computers nach, wenn Sie hierzu Fragen haben.

Doch nun genug der langen Vorrede: Schließen Sie an M1 einen Motor an und geben Sie ein:

*** SYS M1, EIN**
(natürlich ohne Sternchen! s.o.)

Der Motor wird kurz anlaufen und dann wieder stehenbleiben. Wie Sie aus der Interface-Anleitung wissen, überwacht das Interface den Datenfluß und schaltet bei Ausbleiben weiterer Befehle alle angeschlossenen Verbraucher ab.
Probieren Sie nun auch die anderen Ausgabebefehle und machen Sie sich mit Ihrem Gebrauch vertraut.

Die Eingabebefehle probieren Sie am besten mit

*** PRINT USR (E1)**
(wieder ohne Sternchen! Wir werden Sie im folgenden nicht mehr daran erinnern.)

für Eingang 1 aus. Je nachdem, ob der Taster zwischen E1 und +5 V bei der Betätigung der RETURN- bzw. ENTER-Taste gedrückt ist oder nicht, wird der Computer eine 1 oder 0 ausgeben.
Wenn an einen Ausgang noch ein Motor angeschlossen und eingeschaltet war, so wird sich dieser wieder rühren. Auch Eingabebefehle aktivieren das Interface wieder!
Sofern Sie die Diskette nicht benutzen oder auf Kassette umkopieren konnten und das Grundprogramm von Hand eingegeben haben, sollten Sie es jetzt auf Kassette abspeichern. Sie werden es immer wieder brauchen, weil *jedes* Programm, das mit dem fischertechnik Interface Modelle steuern soll, mit diesem Vorspann beginnt, der die neuen Befehle installiert.
Nach dieser Vorübung dürfte es ein Leichtes sein, das erste Projekt, eine computergesteuerte Ampel, anzugehen.

Ampelanlage . . .

Fangen wir zunächst mit dem einfachsten Modell aus dem Baukasten „fischertechnik computing" an: Eine Handvoll Bausteine, die Grundplatte und eine rote, gelbe und grüne Lampe – schon ist die Verkehrsampel aufgebaut. Wie Sie das Modell verkabeln, zeigt Ihnen die Abbildung aus der Bauanleitung.
Schließen Sie die grüne Lampe an M1, die gelbe an M2 und die rote an M3 an. Wenn Sie ein Diskettenlaufwerk besitzen, so laden Sie nun das Programm VERK.AMPEL... Die Punkte stehen wieder für die Namenserweiterung, die den Computertyp angibt. Haben Sie die Programme auf Kassette umkopiert, so verfahren Sie ähnlich.
Anders ist die Situation, wenn Sie die Programme nach der nachfolgend abgedruckten Programmliste eingeben wollen. In diesem Fall haben Sie von vorhin noch das Grundprogramm im Speicher oder laden es wieder ein. Das Ampelprogramm kommt dann zusätzlich in den Computerspeicher. Berührpunkt der beiden Programme ist die Zeile 500 mit dem Befehl

*** 500 SYS INIT**

Wenn das Programm komplett und ausgetestet ist, können Sie es dann natürlich als ein einziges Programm abspeichern. Die Vorgehensweise wird bei den anderen Programmen genauso sein.

Werfen wir nun einen Blick in das Programm hinein. Am Anfang steht nach der Titelmeldung die Belegung des Interface. Obwohl dies „nur" REM-Zeilen sind, wird empfohlen, sie trotzdem mit einzugeben. Mit einem Blick auf diese Zeilen, auf das Interface mit dem symbolischen Verdrahtungsplan und auf das Modell können Sie sofort die Richtigkeit Ihrer Verkabelung prüfen.
Anschließend folgen vier Blöcke Programmtext, für jede Ampelphase einer. In ihnen werden die Parameter der Ampelphase, der Schaltzustand der Lampen und die Zeitdauer, festgelegt. Das eigentliche Schalten erfolgt in dem Unterprogramm ab Zeile 2000. Sie sehen an diesem Beispiel, daß die Schaltzustände der Ausgänge nicht unbedingt direkt im SYS-Befehl verwendet werden müssen. Sie können auch über andere Variablen, ROT, GRUEN und GELB in diesem Fall, übergeben werden.

```
*500 SYS INIT
 600 REM
 610 REM FISCHERTECHNIK COMPUTING
 620 REM
 630 REM AMPELANLAGE
 640 REM
 650 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1994
 660 REM
 730 REM BELEGUNG DES INTERFACE
 740 REM
 750 REM GRUEN=M1
 760 REM GELB=M2
 770 REM ROT=M3
 780 REM
 900 REM FUNKTION
 910 REM EINFACHE AMPEL MIT ZYKLISCHEM DURCHLAUF
 920 REM
*900 PRINT CHR$(147)
 910 PRINT"FISCHERTECHNIK"
 920 PRINT"COMPUTING"
 930 PRINT
 940 PRINT"VERKEHRSAMPEL"
 950 PRINT
1000 REM
1010 REM GRUEN PHASE
1020 LET GRUEN=EIN
1030 LET GELB=AUS
1040 LET ROT=AUS
1050 LET ZEIT=1000
1060 PRINT"GRUEN PHASE"
1070 GOSUB 2000
1080 REM
1090 REM GELB PHASE
1100 LET GRUEN=AUS
1110 LET GELB=EIN
1120 LET ROT= AUS
1130 LET ZEIT=200
1140 PRINT"GELB PHASE"
1150 GOSUB 2000
1160 REM
1170 REM ROT PHASE
1180 LET GRUEN=AUS
1190 LET GELB=AUS
1200 LET ROT=EIN
1210 LET ZEIT=1000
1220 PRINT"ROT PHASE"
1230 GOSUB 2000
1240 REM
1250 REM ROT-GELB PHASE
1260 LET GRUEN=AUS
1270 LET GELB=EIN
1280 LET ROT=EIN
1290 LET ZEIT=200
1300 PRINT"ROT-GELB PHASE"
1310 GOSUB 2000
1320 REM
1330 GOTO 1020
2000 REM ZEITSCHLEIFE
2010 FOR Z=0 TO ZEIT
*2020 SYS M1,ROT
*2030 SYS M2,GELB
*2040 SYS M3,GRUEN
2050 NEXT Z
2060 RETURN
```

. . . mit Fußgängertaste

Mit dem Computer Lampen ansteuern zu können, ist gut und schön. Allerdings fängt die Steuerungstechnik erst dann richtig an, wenn Ein- und Ausgaben einander abwechseln. Denn, mal ehrlich, für die Ampel wie vorhin hätten wir keinen Computer gebraucht. So ist es auch in der Praxis – computergesteuerte Ampeln werden da eingesetzt, wo der Betrieb mit Hilfe von Eingabesignalen an den Verkehrsfluß angepaßt werden muß. Dies können Induktionsschleifen in der Fahrbahn oder auch Drucktaster für Fußgänger sein.

Mit wenigen Handgriffen können wir die Ampel um einen solchen Drucktaster bereichern. Wir schließen den Taster zwischen der Leitung E8 und der gemeinsamen +5V Leitung an.

Die Aufgabenstellung des Programmierers ist es nun, bei Vorliegen eines Tastendrucks von der Grünphase für den Fahrzeugverkehr umzuschalten auf Gelb und Rot. Dieses Umschalten sollte aber nicht schlagartig erfolgen, sondern mit einer kleinen Verzögerung.

In dem Musterprogramm wurde die Aufgabe so gelöst, daß der Laufindex der Verzögerungsschleife auf (Endwert −50) verstellt wird, sobald die Taste betätigt ist. Damit wird die Schleife in kurzer Zeit beendet.

Allerdings darf diese Verstellung nur während der Grünphase erlaubt sein. Die Ampel würde ansonsten ein kurioses Verhalten an den Tag legen.

Die Abfrage des Tastendrucks selbst erfolgt mit der USR-Funktion. Sie sehen aus dem Beispiel, daß sie wie jede andere Funktion verwendet werden kann; in Berechnungen, Abfragen und Drucklisten.

Ihr Wert ist 1, wenn der dazugehörige Eingang mit der gemeinsamen +5V-Leitung verbunden ist, 0 in allen anderen Fällen.

```
*500 SYS INIT
 600 REM
 610 REM FISCHERTECHNIK COMPUTING
 620 REM
 630 REM FUSSGAENGERAMPEL
 640 REM
 650 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
 660 REM
 730 REM BELEGUNG DES INTERFACE
 740 REM
 750 REM GRUEN=M1
 760 REM GELB=M2
 770 REM ROT=M3
 780 REM
 790 REM E8=FUSSGAENGERTASTE
 800 REM
 810 REM FUNKTION
 820 REM FUSSGAENGERAMPEL MIT FUSSGAENGERTASTE
 830 REM
*900 PRINT CHR$(147)
 910 PRINT"FISCHERTECHNIK"
 920 PRINT"COMPUTING"
 930 PRINT
 940 PRINT"FUSSGAENGERAMPEL"
1000 REM
1010 REM GRUEN PHASE
1020 LET GRUEN=EIN
1030 LET GELB=AUS
1040 LET ROT=AUS
1050 LET ZEIT=1000
1060 PRINT"GRUEN PHASE"
1070 GOSUB 2000
1080 REM
1090 REM GELB PHASE
1100 LET GRUEN=AUS
1110 LET GELB=EIN
1120 LET ROT= AUS
1130 LET ZEIT=200
1140 PRINT"GELB PHASE"
1150 GOSUB 2000
1160 REM
1170 REM ROT PHASE
1180 LET GRUEN=AUS
1190 LET GELB=AUS
1200 LET ROT=EIN
1210 LET ZEIT=1000
1220 PRINT"ROT PHASE"
1230 GOSUB 2000
1240 REM
1250 REM ROT-GELB PHASE
1260 LET GRUEN=AUS
1270 LET GELB=EIN
1280 LET ROT=EIN
1290 LET ZEIT=200
1300 PRINT"ROT-GELB PHASE"
1310 GOSUB 2000
1320 REM
1330 GOTO 1020
1340 END
2000 REM ZEITSCHLEIFE
2010 FOR Z=0 TO ZEIT
*2020 SYS M1,ROT
*2030 SYS M2,GELB
*2040 SYS M3,GRUEN
 2050 IF GRUEN <>EIN THEN GOTO 2090
 2060 REM FALLS NICHT GRUEN,TASTE NICHT ABFRAGEN
*2070 IF USR(E8)=1 THEN LET Z=ZEIT-10
 2080 REM FALLS TASTE GEDRUECKT,GRUENPHASE KUERZEN
 2090 NEXT Z
 2100 RETURN
```

Werkzeugmaschine

Nun wollen wir Bewegung in unseren Aufbau bringen. Die kleine Werkzeugmaschine bietet sich dafür an.
Zunächst ein paar Worte dazu, was eine Werkzeugmaschine eigentlich ist. Bohren, Fräsen, Drehen, Stanzen und Prägen sind einige Arbeitsgänge in der Metallverarbeitung. Diese Arbeitsgänge lassen sich in der Produktion automatisieren. Meistens stehen mehrere Automaten um einen runden Drehtisch. An einer Stelle des Tischumfangs werden die Werkstücke eingeführt. Vielleicht mit Hilfe eines Roboters, wie er weiter hinten abgebildet und beschrieben ist. Der Tisch dreht sich immer im Takt um einen festen Drehwinkel und führt so das Teil den einzelnen Automaten zu. Diese bearbeiten das Teil, und anschließend geht es im nächsten Takt um eine Station weiter. Zum Schluß kommt das fertig bearbeitete Werkstück an der Entnahmestelle an, wo es vom Rundtisch genommen wird.

Wir haben in dem Modell dieses Prinzip etwas vereinfacht. Aufnahme und Entnahme finden nicht statt, die vier Bausteine, die die Werkstücke darstellen sollen, kreisen immer herum. Wir haben auch nur einen Automaten, ein kleines Modell einer Bohrmaschine.
Der Rundtisch wird gedreht, bis der Taster von einem Flügel des Werkstücks betätigt wird. Dann fährt der Bohrer herab, wobei die untere Endstellung ebenfalls von einem Taster vermeldet wird. Der Bohrer bleibt stehen; der Bohrvorgang wird durch eine gelbe Lampe, die man noch an dem Modell unterbringen kann, angezeigt. Danach fährt der Bohrer wieder nach oben, bis die obere Endposition durch den zweiten Taster an dem Bohrständer vermeldet wird. Anschließend beginnt das ganze aufs Neue.
Allerdings wird man den Zyklus nicht so wie hier beschrieben beginnen lassen. Der erste Schritt sollte das Hochfahren der Bohrmaschine sein. Es ist bei sorgfältiger Justage der Baugruppen des Modells nämlich durchaus möglich, daß der Bohrer in die seitliche Nut des Bausteins eintauchen kann. Wir können nicht ausschließen, daß die Werkzeugmaschine sich in dieser Ausgangsstellung befindet, wenn das Programm startet. Daher muß die Bohrmaschine immer erst das Werkstück freigeben, ehe der Drehtisch sich dreht.

Um Motoren zu steuern, verwenden wir die Kommandos

```
* SYS M1, RECHTS
* SYS M1, LINKS    und
* SYS M1, AUS
```

für Rechts- und Linkslauf sowie zum Ausschalten. Übrigens,

```
* SYS M1, EIN
```

bewirkt ebenfalls immer Rechtslauf.

Was nun tatsächlich Rechts- oder Linkslauf am Modell bewirkt, läßt sich gar nicht so leicht beantworten. Dies kann von Motor zu Motor unterschiedlich sein. Außerdem hängt es von der Betrachtungsrichtung und dem Getriebeaufbau ab.
Daher sollten Sie die gewünschte Drehrichtung am aufgebauten Modell festlegen. Geben Sie im Direktmodus ein:

```
* SYS M2, RECHTS
```

Wenn die Bohrmaschine jetzt ein kleines Stück nach oben gefahren ist, stimmt die Verdrahtung.
Ist sie dagegen nach unten gefahren, so müssen Sie die beiden Anschlußstecker am Motor tauschen und noch einmal ausprobieren.
Hat sich gar nichts gerührt oder ist der andere Motor angelaufen, so überprüfen Sie bitte noch einmal Ihre Verkabelung. Nach Abschluß des Tests geben Sie wieder im Direktmodus ein:

```
* SYS INIT
```

Eine gute Hilfestellung beim Austesten der Verkabelung leistet auch das Diagnoseprogramm. Da haben Sie ständig Überblick über alle Taster und die beiden Potentiometer und können alle Motoren über Tastendruck von Hand steuern.
Von dem abgedruckten Programm schauen Sie sich einmal Zeile 1010 etwas genauer an: Der Computer prüft dort den Endtaster der Hubbewegung. Solange dieser nicht gedrückt ist, tritt der Computer sozusagen auf der Stelle. Da auch durch die Abfrage der Eingangsleitung das Interface aktiv bleibt, läuft der Motor, der in Zeile 1000 eingeschaltet wurde, nach wie vor weiter.
Wird der Taster gedrückt, so schreitet der Computer in Zeile 1020 weiter. Dort wird der Motor ausgeschaltet.
Derartige Konstruktionen finden Sie noch an zwei weiteren Stellen im Programm, bei Zeile 1070 und 1100.
Zeilen 1130 bis 1160 dienen dazu, eine Bearbeitungspause zu erzeugen. Dabei wird übrigens M3 eingeschaltet, um den Verarbeitungsvorgang zu signalisieren. Findige fischertechnik-Freunde benutzen diese Leitung, um einen dritten mini-Motor aus dem Fundus als Modell eines Bohrers laufen zu lassen oder zumindest eine weitere Lampe einzuschalten. Die FOR-NEXT-Schleifen-Konstruktion der Zeilen 1040–1060 hat jedoch eine andere Bedeutung. Wenn der Computer feststellen soll, ob der Drehteller die neue Lage erreicht hat, so darf er mit dem Testen nicht gleich anfangen. Eine kurze Zeit muß verstreichen, damit der Nocken des Drehtellers erst einmal von der bisherigen Position wegläuft.

Wenn Sie sich bis dahin durch das Programm hindurchgearbeitet haben, werden Sie schon ein Gefühl für „Echtzeitprogrammierung" (so der Fachausdruck) gewonnen haben. Viel mehr als Berechnungen steht hier immer wieder die zeitliche Synchronisation zwischen Modell und Computerabläufen im Vordergrund. Dies bedeutet, daß in

unseren Programmen der Computer die meiste Zeit auf ein Eingabesignal wartet, um anschließend zur nächsten Ausgabeaktion fortzuschreiten.
In der kleinen Grafik (Abb. 5) ist dies übersichtlich dargestellt. Nehmen Sie ein kleines Kegelmännchen von einem Mensch-ärgere-dich-nicht-Spiel und setzen Sie es auf die Startposition. Und nun stellen Sie sich vor, das Männchen wäre der Computer. Auf jedem Feld tut der Computer das, was darin angeschrieben ist. Wenn die am Ausgang angeschriebene Bedingung erfüllt ist, zieht er auf die nächste Position weiter.
Solche grafischen Hilfsmittel sind bei der Entwicklung eigener Programme recht hilfreich. Wenn Sie immer wieder einen Seitenblick auf die Programmliste werfen, stellen Sie schnell fest, wie das Spielfeld in BASIC übersetzt wird. Gestandene Informatiker nennen so eine Grafik übrigens Zustandsdiagramm.

Werkzeugmaschine

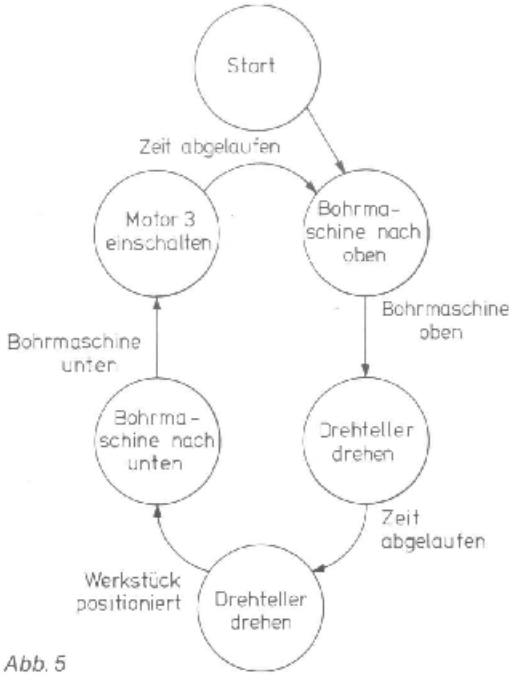


Abb. 5

```
*500 SYS INIT
 600 REM
 610 REM FISCHERTECHNIK COMPUTING
 620 REM
 630 REM WERKZEUGMASCHINE
 640 REM
 650 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
 660 REM
 700 REM BELEGUNG DES INTERFACE
 740 REM
 750 REM E3= DREHTELLERPOSITIONSTASTER
 760 REM E4= ENDTASTER BOHRMASCHINE UNTEN
 770 REM E5= ENDTASTER BOHRMASCHINE OBEN
 780 REM
 800 REM AUSGABEN
 810 REM
 820 REM M1=DREHTELLERMOTOR
 830 REM M2=BOHRMASCHINENMOTOR
 840 REM
*850 PRINT CHR$(147)
 900 PRINT"FISCHERTECHNIK"
 910 PRINT"COMPUTING"
 920 PRINT
 930 PRINT"WERKZEUGMASCHINE"
 940 REM
*1000 SYS M2,RECHTS:REM BOHRER NACH OBEN
*1010 IF USR(E5)=0 THEN GOTO 1010
*1020 SYS M2,AUS:REM BOHRER AUSSCHALTEN
 1030 REM ZEITSCHLEIFE DREHTELLER
 1040 FOR I=1 TO 200
*1050 SYS M1,RECHTS
 1060 NEXT I
*1070 IF USR(E3)=0 THEN GOTO 1070
*1080 SYS M1,AUS:REM DREHTELLERMOTOR AUSSCHALTEN
*1090 SYS M2,LINKS:REM BOHRER NACH UNTEN
*1100 IF USR(E4)=0 THEN GOTO 1100
*1110 SYS M2,AUS:REM BOHRER AUSSCHALTEN
 1120 REM PAUSE FUER BEARBEITUNG
 1130 FOR I=1TO500
*1140 SYS M3,EIN:REM BEARBEITUNGSLAMPE EIN
 1150 NEXT I
*1160 SYS M3,AUS:REM BEARBEITUNGSLAMPE AUS
 1170 GOTO 1000
```

Aufzug

Das Aufzugsmodell aus dem fischertechnik computing-Baukasten bietet ein weites Programmierfeld. Von einer ganz einfachen, „dummen" Aufzugssteuerung über eine Steuerung mit Ruftastenspeicher bis hin zum Programm mit Vorrangregelung – all dies läßt sich an dem kleinen Aufzug studieren.
Die Verdrahtung geht aus der Abbildung in der Bauanleitung hervor. Drei Taster vermelden die Position des Fahrkorbs. Drei weitere Taster sind als Ruftaster in den einzelnen Geschossen angebracht. Ein letzter Taster (er ist in den Abbildungen nicht dargestellt) kann im mittleren Stockwerk rechts von der „Aufzugstür" angebracht werden. Wir kommen auf diesen zweiten Taster weiter unten noch zurück.
Der Aufzug in seiner einfachsten Steuerung ist nicht schwer zu programmieren. Aus den Kommandotasten wird eine *Sollposition* gewonnen. Der Wert der Variablen SP – 0, 1 oder 2 – zeigt an, in welches Stockwerk der Aufzug fahren soll. Dabei steht 0 für Erdgeschoß, 1 und 2 für 1. und 2. Obergeschoß. Die Variable SP wird beim Durchlauf der Abfrageschleife entsprechend den Tastendrucken gesetzt. Dies geschieht in den Zeilen 1110, 1120 und 1130.
Die nächsten drei Zeilen setzen die Variable IP. Hier sind die Positionstaster, die durch den Fahrkorb betätigt werden, Auslöser. Da uns IP angibt, wo der Aufzug gerade ist, wird es *Istposition* genannt. Die Codierung der Stockwerke erfolgt mit den gleichen Zahlen wie bei der Sollposition.

Der Rest ist nun einfach: Ist die Sollposition kleiner als die Istposition, so muß der Aufzug nach unten fahren. Im umgekehrten Fall muß der Fahrkorb nach oben.
Stimmen Soll- und Istposition jedoch überein, so wird der Aufzugsmotor abgestellt. In diesem Augenblick wird auch die Stockwerksmeldung auf den Bildschirm gebracht.
Zu Beginn des Programms wird der Aufzug in das Erdgeschoß gefahren. Dies dient der Herstellung eindeutiger Anfangsbedingung. Stünde der Fahrkorb zwischen zwei Stockwerken, so könnte der Computer die Position nicht erfahren, da keiner der Positionstaster gedrückt ist.
Diese Anfangsbewegung können wir uns wieder zunutze machen, um die Drehrichtung des mini-Motors festzulegen. Geben Sie im Direktmodus ein:

*** SYS M1, LINKS**

Fährt der Fahrkorb abwärts, ist alles in Ordnung.

Wenn Programm und Fahrstuhl korrekt arbeiten, sollten Sie mal Ihre eigene Programmierkünste ins Spiel bringen.

Lösen Sie mal die Frage, was passiert, wenn folgende Zeilen geändert werden:

*** 1230 GOTO 1110**

*** 1270 GOTO 1110**

Als Programmvariante schlagen wir Ihnen einen Aufzug mit Ruftastenspeicher vor. Nun muß man nicht mehr warten, bis der Aufzug steht, um einen neuen Fahrwunsch einzugeben.
Oder erweitern Sie den Aufzug mit einer Richtungsprioritätssteuerung. Hier tritt der zweite Taster im mittleren Geschoß in Aktion. Der Taster auf der linken Seite wird nun als Ruftaster nach oben, der rechte als Ruftaster nach unten ausgewertet. Sie können sich dies leicht merken, da bei dem linken Taster der Betätigungsnocken oben liegt. Bei dem rechten Taster zeigt der Betätigungsnocken jedoch nach unten.
Das Programm soll nun den Fahrkorb so steuern, daß er nicht im mittleren Stockwerk anhält, wenn die gerade vorliegende Fahrtrichtung nicht mit der Wunschrichtung übereinstimmt.
Und wer ein noch weiteres Betätigungsfeld sucht, kann die Ruftasten, die normalerweise innen im Fahrkorb angebracht sind, durch Tasten der Computertastatur ergänzen.

Dann fehlt noch der Alarmknopf und die Abschaltung der Außensteuerung für den Servicetechniker und...
Sie sehen, da bleibt viel Freiraum für Ihre Phantasie.

```
*  500 SYS INIT
   600 REM
   610 REM FISCHERTECHNIK COMPUTING
   620 REM
   630 REM AUFZUG
   640 REM
   650 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
   660 REM
   670 REM BELEGUNG DES INTERFACE
   680 REM
   690 REM EINGABEN
   700 REM E3= POSITIONSTASTE OBEM
   710 REM E4= POSITIONSTASTE MITTE
   720 REM E5= POSITIONSTASTE UNTEN
   730 REM E6= RUFTASTE OBEN
   740 REM E7= RUFTASTE MITTE
   750 REM E8= RUFTASTE UNTEN
   760 REM
   770 REM AUSGABEN
   780 REM M1=AUFZUGSMOTOR
   790 REM M2=LAMPE AUFWAERTS
   800 REM M3=LAMPE ABWAERTS
   810 REM
   820 REM FUNKTION
   830 REM AUFZUG MIT 3 STOCKWERKEN
   840 REM DURCH TASTENDRUCK IM JEWEILIGEN STOCK-
   850 REM WERK WIRD DER AUFZUG POSITIONIERT
*  860 PRINT CHR$(147)
   870 PRINT"FISCHERTECHNIK"
   880 PRINT"COMPUTING"
   890 PRINT
   900 PRINT"AUFZUG"
   910 PRINT
   920 REM
  1000 LET SP=0 :REM SOLLPOS.=0
  1010 LET IP=0 :REM ISTPOS. =0
  1020 LET AZ=0 :REM STOCKWERKSANZEIGE
* 1030 SYS M1,LINKS :REM AUFZUG NACH UNTEN
* 1040 IF USR(E5)=0 GOTO 1030:REM IST AUFZUG UNTEN
* 1050 SYS INIT :REM ALLES ABSCHALTEN
  1060 IF AZ=1 THEN GOTO 1110
  1070 IF IP=0 THEN PRINT"ERDGESCHOSS"
  1080 IF IP=1 THEN PRINT"1.OBERGESCHOSS"
  1090 IF IP=2 THEN PRINT"2.OBERGESCHOSS"
  1100 LET AZ=1
```

Ein stufenloser Positionsmelder

```
*1110 IF USR(E6)=1 THEN SP=2 :REM 2.OG ANFAHREN
*1120 IF USR(F7)=1 THEN SP=1 :REM 1.OG ANFAHREN
*1130 IF USR(E8)=1 THEN SP=0 :REM EG ANFAHREN
*1140 IF USR(E3)=1 THEN IP=2 :REM AUFZUG IM 2.OG
*1150 IF USR(E4)=1 THEN IP=1 :REM AUFZUG IM 1.OG
*1160 IF USR(E5)=1 THEN IP=0 :REM AUFZUG IM EG
 1170 IF SP<IP THEN 1200:REM AUFZUG NACH UNTEN
 1180 IF SP>IP THEN 1240:REM AUFZUG NACH OBEN
 1190 GOTO 1050
*1200 SYS M1,LINKS :REM AUFZUG NACH UNTEN
*1210 SYS M2,EIN :REM LAMPE ABWAERTS EIN
 1220 LET AZ=0
 1230 GOTO 1140
*1240 SYS M1,RECHTS :REM AUFZUG NACH OBEN
*1250 SYS M3,EIN :REM LAMPE AUFWAERTS EIN
 1260 LET AZ=0
 1270 GOTO 1140
```

In fischertechnik computing sind zwei Potentiometer enthalten, die der stufenlosen Positionsmeldung dienen. Dies hat gegenüber dem Einsatz von Tastern den Vorteil, daß sich Bewegungen viel feiner abstufen lassen.

Ein Potentiometer besteht aus einer Widerstandsbahn, auf der durch Drehen einer Achse ein Schleifer entlangbewegt wird. An den Endpunkten der Widerstandsbahn und dem Schleifer sind Lötkontakte angebracht. Abb. 6 zeigt ein geöffnetes Potentiometer.

Je nach Stellung der Drehachse ergeben sich zwischen den Endpunkten des Potentiometers und dem Schleifer zwei Widerstandswerte, deren Summe dem Widerstandswert der gesamten Widerstandsbahn entspricht.

Aus den Herstellerangaben entnehmen wir, daß der gesamte Drehwinkel der Achse, von Endpunkt zu Endpunkt, 270° beträgt. Unter der Annahme, daß die Widerstandsbahn gleichmäßig ist, kann aus den Widerstandswerten des Potentiometers die Winkelstellung der Achse ermittelt werden.

Die Widerstandswerte in Zahlenwerte des Computers umzuwandeln, ist eine der Aufgaben des Interface. Die Potentiometer werden zwischen die Leitungen EX bzw. EY und +5 V als gemeinsamem Pol angeschlossen. Die Abfrage der Potentiometerstellung geschieht mit der USR-Funktion.

Obwohl bei den verschiedenen Computern, für die es ein fischertechnik Interface gibt, immer wieder der gleiche programmtechnische und elektronische Ablauf benutzt wird, verhält sich die USR-Funktion doch unterschiedlich. Insbesondere sollten Sie vermeiden, eine offene Potentiometerleitung abzufragen. Die Effekte reichen vom langsamen Blinken der Leuchtdiode des Interface (C64) über einen harmlosen Überlauf mit Wert 255 (Apple II, CBM) bis zum „Aufhängen" des Computers (VC20, Acorn).

Auch die zurückgelieferten Zahlenwerte können leicht unterschiedlich ausfallen, liegen aber immer im Bereich 0 bis 255. Bei einigen Interface-Varianten wird die 0 erreicht, bei anderen ist ca. 20 der kleinste Wert.

Damit wir üben, mit diesem wichtigen Bauteil von fischertechnik computing umzugehen, wollen wir uns an das nächste Modell heranwagen.

Potentiometer

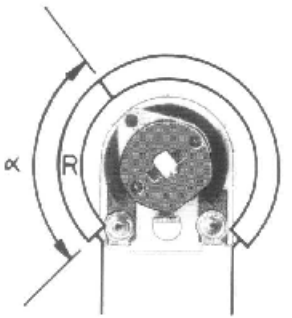


Abb. 6

Antennenrotor

Der Antennenrotor hat die Aufgabe, die Peilantenne entsprechend einer Vorgabe in die richtige Himmelsrichtung zu stellen. Damit der Computer weiß, in welche Richtung die Antenne zeigt, ist in ihrem Fuß ein Potentiometer eingebaut. Der Drehkranz, auf dem die Antenne ruht, wird durch einen Motor angetrieben. Die vorgegebene Richtung ergibt sich aus der Stellung des zweiten Potentiometers. Ein Drehrad mit Markierungsstift zeigt die Sollrichtung an.
Wenn Sie schon das Aufzugsmodell gebaut und programmiert haben, werden Sie eine große Ähnlichkeit in der Aufgabenstellung erkennen. Wieder gibt es Ist- und Sollpositionen, nur diesmal fein abgestuft über einen größeren Zahlenbereich. Wieder ergibt sich die Drehrichtung des Motors aus dem Unterschied zwischen Ist- und Sollposition. Bei Gleichheit von beiden wird der Motor abgestellt.
Schauen Sie sich das nachstehend aufgeführte Programm an: Die Verwandtschaft mit dem Aufzugsprogramm ist unverkennbar – zumindest bis Zeile 1050. Danach folgt aber etwas Neues, das einiger Worte der Erläuterung bedarf.

Stellen Sie sich vor, Sie würden bei Gleichheit von Ist- und Sollposition einfach den Rotorantrieb abschalten und erneut an den Anfang der Abfrageschleife (Zeile 1000) springen. Der Antriebsmotor bleibt nicht sofort stehen, sondern läuft noch aus. Schon ist das Potentiometer verstellt und der Motor wird nun in Gegenrichtung angesteuert. Unter Umständen kommt der Motor aus einem solchen Hin und Her gar nicht mehr heraus. Eine Regelschwingung hat sich aufgebaut, würde der Fachmann sagen.
Daher wurde der Motor so programmiert, daß er sich regelrecht an sein Ziel heranpirscht. Ist er noch weiter als 10 Einheiten entfernt, läuft er mit Vollgas. Ist er näher, läuft er mit reduzierter Kraft.
Da wir die Motorspannung nicht verstellen können, schalten wir den Motor in kurzen Abständen ein und aus. Die Einschaltzeit ergibt sich aus der Rechendauer der FOR-NEXT-Schleife bei 1070 und 1080. Die Ausschaltdauer besteht aus dem Rest der Abfrageschleife.
Ein kleiner Leckerbissen ergibt sich aus der Verwendung des absoluten Differenzwertes als Laufindex der Schleife. Die Anschaltdauer verkürzt sich bei Annäherung an das Ziel immer mehr.
Sie sollten das Programm ausprobieren, um mit dem Einsatz der Potentiometer vertraut zu werden. Als erstes prüfen Sie wieder die Drehrichtung des Motors. Ist er verpolt angeschlossen, so bewegt er den Rotor nicht auf das Ziel hinzu, sondern weg. Der Rotor hat die Tendenz, gegen den Endanschlag des Potentiometers zu laufen. Mit Hilfe des Bildschirmausdrucks ist dies leicht zu erkennen. Stoppen Sie in diesem Fall sofort das Programm und vertauschen Sie die Anschlüsse des Motors.
Merkwürdige Effekte ergeben sich auch in den Extremstellungen der Potentiometer. Aufgrund der Wertestreuung der Potentiometer und der übrigen Bauelemente des Interface müssen die Wertebereiche der beiden Potentiometer nicht übereinstimmen. Dann ist es natürlich nutzlos, den Motor gegen den Endanschlag des Potentiometers arbeiten zu lassen – für einen Wert, der niemals erreicht werden kann.
Sie können nun versuchen, die optimale Lageregelung für Ihre Zwecke zu finden. Entfernen Sie probehalber die Ausdruckzeile 1020. Ersetzen Sie Zeile 1070 durch Varianten wie

```
1070 FOR I = 0 TO 2 * D
```

Verändern Sie die Bremsphase (10 in unserem Beispiel) und beobachten Sie die Auswirkungen.

```
*500 SYS INIT
 600 REM
 610 REM FISCHERTECHNIK COMPUTING
 620 REM
 630 REM ANTENNENROTOR
 640 REM
 650 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
 660 REM
 730 REM BELEGUNG DES INTERFACE
 740 REM
 750 REM EINGABEN
 760 REM EX= POTENTIOMETER GEBER
 770 REM EY= POTENTIOMETER ROTOR
 780 REM
 790 REM AUSGABEN
 900 REM M1=ROTORMOTOR
 910 REM
 920 REM FUNKTION
 930 REM ROTOR WIRD NACH DER VORGABE
 940 REM AM DREHRAD POSITIONIERT
 950 REM
*900 PRINT CHR$(147)
 910 PRINT"FISCHERTECHNIK"
 920 PRINT"COMPUTING"
 930 PRINT
 940 PRINT"ANTENNENROTOR"
 950 REM
*1000 LET X=USR(EX) :REM POTISTELLUNG GEBER
*1010 LET Y=USR(EY) :REM POTISTELLUNG ROTOR
 1020 PRINT"SOLL=";X;"    IST=";Y
 1030 LET D=ABS(X-Y)
*1040 IF Y>X THEN SYS M1,RECHTS
*1050 IF Y<X THEN SYS M1,LINKS
 1060 IF D>10 THEN 1000
 1070 FOR I=0 TO D
 1080 NEXT I
*1090 SYS M1,AUS
 1100 GOTO 1000
```

Sortieranlage

Messen und Sortieren – eine Aufgabe, die immer wieder auftritt, ob nun Hühnereier nach Gewichtsklassen eingeteilt werden oder Ausschuß aus einer Produktion ausgesondert werden soll. Wir wollen diese Aufgabe im Kern erfassen und eine Maschine aufbauen, die Bausteine 30 und Bausteine 15 unterscheiden kann. Diese Leistung mag zwar den einen oder anderen nicht sonderlich beeindrucken, jedoch geht es uns darum, das Prinzip zu studieren. Alles weitere ist dann „nur" Fleißarbeit, bis hin zum kompletten fischertechnik-Aufräumautomaten.

Wie in den ersten Worten gesagt, besteht die Aufgabe aus zwei Teilen: Messen – in diesem Fall die Länge des Bausteins, und Sortieren – in zwei Auffangbehälter, die rechts und links der Teilzufuhr angebracht sind.
Der Baustein wird über die schräge Rutsche vorn am Modell in die Meßstrecke befördert. Anschließend wird er von Hand nach links in den Meßkanal geschoben, wobei zwangsläufig der Starttaster betätigt wird.
Messen heißt Vergleichen, hier vergleichen wir die unbekannte Länge des Bausteins mit dem Abstand zwischen dem Starttaster und dem Meßtaster. Werden beide mini-Taster gleichzeitig durch den Baustein betätigt, so muß es sich um den längeren Baustein 30 handeln. Der Baustein 15 kann nur einen mini-Taster betätigen.

Nachdem erkannt ist, um welchen Baustein es sich handelt, erfolgt der Sortierschritt. In einem Fall läuft der Schlitten, der den Baustein schon in den Meßkanal transportiert hat, weiter und wirft den Baustein am linken Ende aus. Im anderen Fall wird der Motor des Schlittens umgepolt, so daß der Baustein nach rechts mitgenommen wird. Drei Taster, die längs der Bahn des Schlittens angeordnet sind, vermelden dem Computer das Erreichen der Endlagen und der Ausgangsstellung in der Mitte, zu der der Schlitten zuletzt wieder zurückkehrt.

Es wäre schön, wenn Sie noch die rote und die grüne Anzeigelampe benutzen würden, um dem „Bedienungspersonal" das Ergebnis des Sortiervorgangs anzuzeigen.

Abb. 7

Die Zusammenhänge des Programms stellen wir wieder als Zustandsdiagramm dar. Wie schon bei der Werkzeugmaschine können Sie die Wege im Zustandsdiagramm verfolgen. Eine neue Situation ergibt sich unten in der Mitte. Hier gilt es, einen Blick auf zwei verschiedene Ausgangsbedingungen zu haben. Je nachdem welche eintrifft, verfolgen Sie den weiteren Weg.

Ganz klar, der Weg rechts herum gehört zu dem Baustein 30, der Weg links herum zu dem Baustein 15. Ein Blick in das BASIC-Programm lehrt uns, daß die Entscheidung, ob kurz oder lang, in den Zeilen 1070 und 1090 fällt.
Mittlerweile haben wir schon ganz schön Routine bei solchen Steuerungsaufgaben. Auch die Bewegungsrichtung des Sortierschlittens finden wir schnell heraus.

*** SYS M1, LINKS** und
*** SYS M1, RECHTS**

sollen den Schlitten in die angegebene Richtung bewegen.

```
*500 SYS INIT
  600 REM
  610 REM FISCHERTECHNIK COMPUTING
  620 REM
  630 REM SORTIERANLAGE
  640 REM
  650 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
  660 REM
  730 REM BELEGUNG DES INTERFACE
  740 REM
  750 REM EINGABEN
  760 REM E1=STARTTASTE
  770 REM E2=MESSTASTE
  780 REM E3=AUSWURF KURZER BAUSTEIN
  790 REM E4=STARTPOSITION
  800 REM E5=AUSWURF LANGER BAUSTEIN
  810 REM
  820 REM AUSGABEN
  830 REM M1=WAGEN
  840 REM M2=LAMPE KURZER BAUSTEIN
  850 REM M3=LAMPE LANGER BAUSTEIN
  860 REM
  870 REM FUNKTION
  880 REM ANLAGE SORTIERT NACH KURZEN
  890 REM UND LANGEN BAUSTEINEN
  900 REM
*910 PRINT CHR$(147)
  920 PRINT"FISCHERTECHNIK"
  930 PRINT"COMPUTING"
  940 PRINT
  950 PRINT"SORTIERANLAGE"
  960 PRINT
```

```
  970 PRINT"MASCHINE BEREIT"
  980 PRINT
* 1000 SYS INIT
* 1010 IF USR(E1)=1 THEN GOTO 1010
  1020 REM WARTE AUF STARTTASTE
  1030 REM WARTESCHLEIFE
  1040 FOR I=1 TO 500
  1050 NEXT I
* 1060 SYS M1,LINKS :REM BAUSTEINE ZU MESSUNG
* 1070 IF USR(E1)=0 AND USR(E2)=0 THEN GOTO 1180
  1080 REM MESSUNG LANGER BAUSTEIN
* 1090 IF USR(E1)=1 AND USR(E2)=1 THEN GOTO 1120
  1100 REM MESSUNG KURZER BAUSTEIN
  1110 GOTO 1070
  1120 PRINT"KURZER BAUSTEIN"
* 1130 SYS M2,EIN :REM KONTROLLE FUER KURZ
* 1140 IF USR(E3)=0 THEN GOTO 1140
* 1150 SYS M1,RECHTS :REM WAGENRUECKLAUF
* 1160 IF USR(E4)=0 THEN GOTO 1160
  1170 GOTO 1020
  1180 PRINT"LANGER BAUSTEIN"
* 1190 SYS M1,RECHTS :REM AUSSORTIEREN
* 1200 SYS M3,EIN: REM KONTROLLE LANG
* 1210 IF USR(E5)=0 THEN GOTO 1210
* 1220 SYS M1,LINKS: REM WAGENRUECKLAUF
* 1230 IF USR(E4)=0 THEN GOTO 1230
  1240 GOTO 1000
```

Turm von Hanoi

Nun endlich kommen wir zu dem ersten Roboter. Dieser Geselle hier kann sich um seine senkrechte Achse drehen und seinen Arm auf- und abfahren. Mit dem Magneten als Greifhand kann er Eisenteile, die vor ihm auf einem Kreisbogen angeordnet sind, auflesen und an anderer Stelle ablegen. Hierzu eignen sich z.B. Fünfpfennigstücke und Groschen oder aber auch die dem Kasten beigefügten Blechscheiben.
Wir haben uns für den Roboter einen Bewegungsablauf ausgesucht, dem eine alte Denksportaufgabe zugrunde liegt.
Der Überlieferung nach sind in einem Buddhistenkloster Mönche mit einer Aufgabe beschäftigt, die ein außerordentliches Maß an Geduld erfordert. Auf einem Brett stehen drei Pfähle, aus Kupfer, Silber und Gold. Irgendwann vor langen Zeiten staken auf dem Kupferstab hundert Lochscheiben, alle mit verschiedenem Durchmesser und der Größe nach geordnet. Die Aufgabe ist nun, den Stapel auf den goldenen Pfahl umzuschichten. Dabei gelten folgende Regeln:

1. Es darf jeweils nur eine Scheibe bewegt werden.
2. Es darf nie eine größere Scheibe auf eine kleinere gelegt werden.
3. Zur Ablage dürfen nur die drei Pfähle benutzt werden.

Die Überlieferung berichtet weiter, daß bei Erreichen des Ziels die Welt untergeht.

In unserem Beispiel haben wir auf die kupfernen, silbernen und goldenen Pfähle verzichtet und höchstens 19 Scheiben verwendet, um den Weltuntergang nicht mutwillig herbeizuführen. Für die Stapelplätze befinden sich je eine Position am vorderen und hinteren Rand sowie in der Mitte der Bauplatte. Der Ausleger des Roboters ist übrigens in der Länge leicht verstellbar, so daß der Magnetgreifer verschiedene Radien überstreichen kann.

Es ist ziemlich hoffnungslos, wenn Sie versuchen sollten, z.B. mit fünf Scheiben sofort die Lösung zu erhalten. Besser ist es, folgende Beobachtung nachzuvollziehen: Um die unterste Scheibe von dem „kupfernen Pfahl" auf den „goldenen Pfahl" zu bringen, müssen Sie den ganzen darüberliegenden Stapel zunächst irgendwie auf den „silbernen Pfahl" räumen. Damit ist der Weg frei und die unterste Scheibe kann umgesetzt werden. Anschließend wird nur wieder der Reststapel vom „silbernen Pfahl" auf den „goldenen Pfahl" umgeräumt.

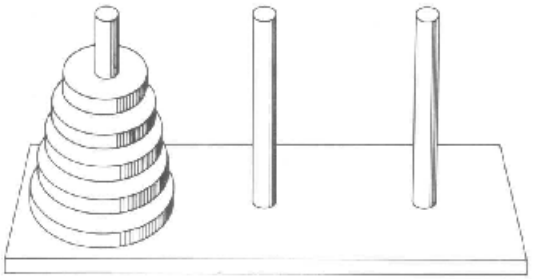


Abb. 8

Fühlen Sie sich betrogen? Vielleicht ja, denn wie soll das Umräumen des Reststapels geschehen? Doch halt, jetzt kommt der entscheidende Moment, und uns trennt nur noch ein Hauch von der vollständigen Lösung. Nehmen wir an, wir hätten fünf Scheiben. Dann wissen wir ja, wie eben diese letzte, fünfte Scheibe umzuordnen ist. Betrachten wir aber den verflixten Reststapel mit vier Scheiben, so fällt uns auf, daß wir genauso die unterste, vierte Scheibe wie erforderlich umsetzen können. Wir bringen den Reststapel von drei Scheiben zwischenzeitlich auf einem freien Pfahl unter. So schieben wir das Problem vor uns her, von einem Vierer- auf einen Dreier- und Zweierstapel. Zuletzt sind wir in der Überlegung bei einer einzigen Scheibe angelangt. Und von der wissen wir ja, wie sie umgesetzt wird.
Den Schluß können wir nun wieder zurückverfolgen und also behaupten, daß wir das Spiel „Turm von Hanoi" für beliebige Scheibenzahl beherrschen. Genau nach dieser Überlegung arbeitet das Programm, das nachstehend abgedruckt ist.
Zum Schluß noch ein Wort zur Laufdauer des Programms. Bei Durchsicht der Rekursion (so nennt man ein sich selbst aufrufendes Programm) erhalten wir 2^n-1 Schritte bei n Scheiben. Bei fünf Scheiben muß der Roboter somit 31 Schritte absolvieren. Bei hundert Scheiben ergeben sich rund

126.765.060.000.000.000.000.000.000.000

Schritte. Selbst der schnellste Mönch (1 Sekunde je Zug) müßte etwa 40 Tausend Millionen mal Billionen Jahre arbeiten, um den Weltuntergang herbeizuführen.

Das BASIC-Programm selbst ist klar gegliedert. Der erste Teil, von Zeile 1000 bis Zeile 1850 enthält das Rahmenprogramm mit den Anfangsbedingungen und insbesondere die Rekursionsroutine ab Zeile 1690. Dieser Teil kann auch für sich alleine verwendet werden, da nur über die Zeile 1780 der gesamte zweite Teil des Programms mit der Steuerung des Roboters aufgerufen wird.

Ein solches Vorgehen wird auch Rucksackprogrammierung genannt, da die Roboterroutinen wie ein Rucksack an ein schon vorhandenes Programm angehängt werden. Mit dieser Methode können Sie bestimmt einige andere bekannte Spiele wie z.B. Nim auf einen Roboter als Spielpartner umstellen.
In der Roboterroutine treffen wir eine Reihe von alten Bekannten wieder, so z.B. die Lagerregelungsroutine ab Zeile 2200. Sie ist für die Drehung des

Roboters zuständig. Die Größe SOLLWERT leitet sich letztlich von den in den Zeilen 1010–1040 festgelegten Säulenpositionen ab. Diese sollten Sie vorher mit Hilfe des Diagnoseprogramms festlegen.
Das gleiche Programm verwenden Sie auch, um die zuverlässige und leichtgängige Funktion der Endabschalter für die senkrechte Bewegung zu testen.
Eine kleine Besonderheit weist das Unterprogramm zum Ausschalten des Magnets auf. Bevor der Magnet abgeschaltet wird, wird er noch einmal kurz in *umgekehrter* Stromrichtung betrieben. Eine Restmagnetisierung in den Blechscheiben wird dadurch vermindert.
Dennoch muß die Stärke des Elektromagneten durch aufgeklebte Abziehfolien über den Polschuhen so eingestellt werden, daß die Teile zuverlässig gegriffen und losgelassen werden.
Wenn Sie nun auch noch die richtige Drehrichtung der beiden Antriebsmotoren ermittelt haben, kann es mit dem Umsortieren losgehen.

```
*500 SYS INIT
 600 REM
 610 REM FISCHERTECHNIK COMPUTING
 620 REM
 630 REM TURM VON HANOI
 640 REM
 650 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
 660 REM
 730 REM BELEGUNG DES INTERFACE
 740 REM
 750 REM EINGABEN
 760 REM E1=UNTEN
 770 REM E2=OBEN
 780 REM EY=WINKELSTELLUNG
 790 REM
 800 REM AUSGABEN
 810 REM M1=TURM
 820 REM M2=ARM
 830 REM M3=MAGNET
 840 REM
 850 REM FUNKTION
 860 REM SORTIERUNG DES SCHEIBENSTAPELS
 870 REM NACH DER BUDDHISTISCHEN UEBERLIEFERUNG
 880 REM
 1000 DIM X(31),Y(31),Z(31)
 1010 REM SAEULENPOSITIONEN (EMPIRISCH ERMITTELT)
 1020 LET SAEULE(1)=85
 1030 LET SAEULE(2)=110
 1040 LET SAEULE(3)=135
*1500 PRINT CHR$(147)
 1510 PRINT"FISCHERTECHNIK"
 1520 PRINT"COMPUTING"
 1530 PRINT
 1540 PRINT"TURM VON HANOI"
 1550 PRINT
 1600 INPUT"WIEVIELE SCHEIBEN HAT DER TURM";N
 1610 IF N<1 THEN END
 1620 LET M=0
 1630 LET X(M)=N:LET Y(M)=1:LET Z(M)=3
 1640 GOSUB 1700
 1650 END
 1660 REM PSEUDOREKURSIVE ROUTINE
 1700 IF  X(M)=0 THEN RETURN
 1710 LET M=M+1
 1720 LET X(M)=X(M-1)-1
 1730 LET Y(M)=Y(M-1)
 1740 LET Z(M)=6-Y(M-1)-Z(M-1)
 1750 GOSUB 1700:REM REKURSIVAUFRUF
 1760 LET M=M-1
 1770 PRINT"SCHEIBE";X(M);"VON SAEULE";Y(M);
      "NACH SAEULE";Z(M)
 1780 GOSUB 2000 :REM ROBOTER ROUTINE
 1790 LET M=M+1
 1800 LET X(M)=X(M-1)-1
 1810 LET Y(M)=6-Y(M-1)-Z(M-1)
 1820 LET Z(M)=Z(M-1)
 1830 GOSUB 1700:REM REKURSIVAUFRUF
 1840 LET M=M-1
 1850 RETURN
 2000 REM ROBOTER ROUTINE
 2010 LET SOLLWERT=SAEULE(Y(M))
 2020 GOSUB 2200 :REM DREHE TURM
 2030 GOSUB 2310 :REM SENKE ARM
*2040 SYS M3,EIN :REM MAGNET EINSCHALTEN
 2050 GOSUB 2350 :REM HEBE ARM
 2060 LET SOLLWERT=SAEULE(Z(M))
 2070 GOSUB 2200 :REM DREHE TURM
 2080 GOSUB 2310 :REM SENKE ARM
 2090 GOSUB 2120 :REM MAGNET AUS
 2100 GOSUB 2350 :REM HEBE ARM
 2110 RETURN
 2120 REM MAGNET AUS ROUTINE
*2130 SYS M3,LINKS
 2140 REM MAGNET UMGEPOLT ZUR ENTMAGNETISIERUNG
*2150 SYS M3,AUS :REM MAGNET AUSGESCHALTET
 2160 RETURN
 2200 REM LAGEREGELUNGS ROUTINE
*2210 LET D=USR(EY)-SOLLWERT
*2220 IF D>0 THEN SYS M1,RECHTS
*2230 IF D<0 THEN SYS M1,LINKS
 2240 IF D=0 THEN RETURN
 2250 LET D=ABS(D)
 2260 IF D >5 THEN GOTO 2210
 2270 FOR I=0 TO D
 2280 NEXT
*2290 SYS M1,AUS
 2300 GOTO 2210
*2310 SYS M2,LINKS :REM ARM ABSENKEN
*2320 IF USR(E1)=0 THEN 2310
*2330 SYS M2,AUS
 2340 RETURN
*2350 SYS M2,RECHTS :REM ARM ANHEBEN
*2360 IF USR(E2)=0THEN 2350
*2370 SYS M2,AUS
 2380 RETURN
```

Teach-in Roboter

Industrieroboter werden eingesetzt, wo es darauf ankommt, eine Bewegungsabfolge immer wieder unermüdlich und mit nicht nachlassender Genauigkeit auszuführen. Wenn dann noch hinzukommt, daß die Arbeit in einer dem Menschen unzuträglichen Umgebung, z.B. radioaktive Strahlung, Hitze, Staub, Farbsprühnebel usw. ausgeführt werden muß, ist der Roboter das geeignete Arbeitsmittel. Eine weitere Vorbedingung für den Robotereinsatz ist, daß die Aufgaben dennoch von Zeit zu Zeit wechseln. Sonst wäre es sicherlich nutzbringender, eine speziell angelegte Maschine zu bauen. Und dies bringt uns zu dem Kernpunkt der Aufgabenstellung: Ein Roboter kann in seiner Bewegungsabfolge frei programmiert werden.

Zwei Programmiermethoden für Roboter sind heute üblich. Zum einen können wir uns an den Schreibtisch setzen und das Programm von Anfang an planen. Wenn das Programm erstellt ist, wird es in den Steuerungscomputer des Roboters eingegeben und mit dem Roboter getestet. Gegebenenfalls muß das Programm dann korrigiert und verbessert werden, bis letztlich das gewünschte Resultat erzielt wird. Dieses Verfahren ist zwar sehr zeitaufwendig, wird aber immer dann eingesetzt werden, wenn die Roboterbewegungen von irgendwelchen Rechenergebnissen abhängen.

Viel häufiger jedoch wird heute das sogenannte „Teach-in"-Verfahren angewandt. In diesem Fall wird der Roboter von dem Bediener von Hand über eine Kommandotastatur gesteuert. Das in dem Computer installierte Programm zeichnet die Bewegungen auf. Nach Abschluß der Aufgabe kann der Computer die abgespeicherte Bewegung von dem Roboter beliebig oft wiederholen lassen.

Der Vorteil des Verfahrens ist offensichtlich die unmittelbare Programmierung des Roboters, die sogar ohne Kenntnisse des Computers und seiner Programmiersprache durchgeführt werden kann. Der Nachteil liegt in der Beschränkung auf einfache, unverzweigte Bewegungsabfolgen.
Ein solches Programm wollen wir im folgenden vorstellen.

Der Roboter wird dabei über sechs Tasten der Bedienkonsole gesteuert. Die Tasten haben im einzelnen die Bedeutung:

Arm auf	(E1)
Arm ab	(E2)
Drehe links	(E3)
Drehe rechts	(E4)
Magnet ein	(E5)
Magnet aus	(E6)

Zwei Tasten haben eine Sonderfunktion. Die Taste

Lerne (E7)

bewirkt, daß der Computer die Roboterposition abspeichert. Die letzte Taste startet dann den Roboter:

Start/Stop (E8)

Sie dient gleichzeitig dazu, den endlos wiederholenden Bewegungsablauf des Roboters zu stoppen.

Wie aus dieser einleitenden Beschreibung hervorgeht, zerfällt das Programm in zwei Teile: das Lernmodul zur Aufzeichnung der Bewegung und das Ausführmodul zur Wiederholung der Bewegung.
Das Lernmodul erstreckt sich über den ersten Teil des Programms, von Zeile 1000 bis 1680. Dabei prüft das Programm immer wieder alle acht Tasten des Bedienpults. Ist eine der Motorsteuertasten gedrückt, so wird der entsprechende Motor in der verlangten Drehrichtung betrieben. In diesem Fall wird die Taste immer wieder geprüft, bis sie losgelassen wird. Damit kann der Roboter feinfühlig positioniert werden und eine Mehrfachbetätigung der Tasten wird unterdrückt.

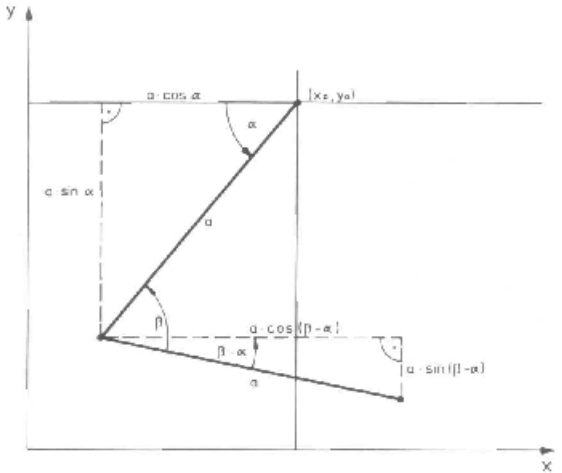
Ein kleiner Unterschied tritt bei der Steuerung des als Greifhand dienenden Elektromagneten auf. Der Tastendruck wird hier in der Variablen MG$ abgespeichert. Der Magnet hat eine Dauerfunktion.
Die Lerntaste löst die Aufzeichnung der gerade vorliegenden Position aus. Die beiden Potentiometerstellungen sowie der Schaltzustand des Magneten werden in einer Tabelle abgespeichert (Zeile 1620–1660), ausgedruckt und der Zeiger auf die nächste Tabellenposition vorgerückt.
Das Programm führt jedoch erst die Abfrage durch, wenn die Lerne-Taste losgelassen wird. Warum? Wenn der Computer nicht auf das Ende der Tastenbetätigung wartet, sondern gleich wieder in die Abfrageschleife verzweigt, würde die gleiche Position mehrfach abgespeichert werden.
Andererseits können Sie durch mehrmaliges Drükken der Lerne-Taste auch Pausen im Bewegungsablauf erzeugen.
Die Betätigung der Start-Taste verzweigt in den zweiten Teil des Programms, das Ausführmodul. Zunächst wird auch wieder das Loslassen der Taste überprüft, da die Taste gleichzeitig Stoptaste ist.
Der Tabellenzeiger wird auf die erste Tabellenposition gesetzt. Die Tabellenwerte dienen nun als Sollwerte für zwei Lageregelungsroutinen, wie wir sie schon kennen. Eine steuert die Auf/Ab-Bewegung, die andere die Drehbewegung. Im dritten Teil des Ausführmoduls wird der Magnet geschaltet.
Bei Erreichen des Tabellenendes wird wieder auf den Anfang der Tabelle zurückgestellt.

```
*500 SYS INIT
 510 REM
 520 REM FISCHERTECHNIK COMPUTING
 530 REM
 540 REM TEACH IN ROBOTER
 550 REM
 560 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
 570 REM
 580 REM TEACH TABELLE
 590 DIM HOEHE (100),WINKEL(100),MAGNET$(100)
 600 REM KOSTANTEN
 610 LET GX=30 :REM BREMSPHASE ARM
 620 LET GY=4 :REM BREMSPHASE WINKEL
 630 LET ID=-1 :REM ZEIGER IN DIE TEACH TABELLE
 640 LET MG$="AUS" :REM START MIT MAGNET AUS
 650 REM BELEGUNG DES INTERFACE
 660 REM
 670 REM EINGABEN
 680 REM E1=AUF
 690 REM E2=AB
 700 REM E3=RECHTS
 710 REM E4=LINKS
 720 REM E5=MAGNET EIN
 730 REM E6=MAGNET AUS
 740 REM E7=LERNEN
 750 REM E8=STARTEN
 760 REM EX=ARMPOSITION
 770 REM EY=WINKELPOSITION
 780 REM
 790 REM AUSGABEN
 800 REM M1=DREHEN
 810 REM M2=AUF/AB
 820 REM M3=MAGNET
 830 REM
 840 REM FUNKTION
 850 REM DAS PROGRAMM BEFINDET SICH
 860 REM ZUNAECHST IN DER LERNPHASE.
 870 REM DER ROBOTER WIRD UEBER DIE
 880 REM TASTER E1-E6 GESTEUERT.
 890 REM MIT E7 WIRD DER STATUS DES
 900 REM ROBOTERS ABGESPEICHERT.
 910 REM MIT E8 WIRD DER AUTOMATISCHE
 920 REM ABLAUF GESTARTET.
 930 REM DER ABLAUF WIRD ZYKLISCH WIEDERHOLT,
 940 REM BIS DIE TASTE E8 WIEDER GEDRUECKT WIRD.
 950 REM
 960 GOSUB2800
 1000 REM ABFRAGE SCHLEIFE
 1100 REM TASTE AUF
*1110 IF USR(E1)=0 THEN GOTO 1140
*1120 SYS M2,RECHTS
 1130 GOTO 1100
*1140 SYS M2,AUS
 1200 REM TASTE AB
*1210 IF USR(E2)=0 THEN GOTO 1240
*1220 SYS M2,LINKS
 1230 GOTO 1200
*1240 SYS M2,AUS
 1300 REM TASTE RECHTS
*1310 IF USR(E3)=0 THEN GOTO 1340
*1320 SYS M1,RECHTS
 1330 GOTO 1300
*1340 SYS M1,AUS
 1400 REM TASTE LINKS
*1410 IF USR(E4)=0 THEN GOTO 1440
*1420 SYS M1,LINKS
 1430 GOTO 1400
*1440 SYS M1,AUS
 1500 REM TASTE MAGNET EIN
*1510 IF USR(E5)=0 THEN GOTO 1540
*1520 SYS M3,EIN
 1530 LET MG$="EIN"
 1540 REM TASTE MAGNET AUS
*1550 IF USR(E6)=0 THEN GOTO 1580
*1560 SYS M3,AUS
 1570 LET MG$="AUS"
 1580 REM TASTE LERNEN
*1590 IF USR(E7)=0 THEN GOTO 1670
 1600 REM WARTE BIS TASTE LOSGELASSEN
*1610 IF USR(E7)=1 THEN GOTO 1610
 1620 LET ID = ID +1
 1630 LET HOEHE(ID)=USR(EX)
 1640 LET WINKEL(ID)=USR(EY)
 1650 LET MAGNET$(ID)=MG$
 1660 PRINT ID;"  ";HOEHE(ID);"  ";WINKEL(ID);"
      ";MAGNET$(ID)
 1670 REM AUSFUEHREN
*1680 IF USR(E8)=0 THEN GOTO 1100
 1690 GOSUB 2900
 1700 IF USR(E0)=1 THEN GOTO 1700
 1710 LET ID=ID+1
 1720 LET HOEHE(ID)=-1
 2000 REM AUSFUEHRUNGSSCHLEIFE
 2010 LET ID = -1
 2100 LET ID =ID + 1
 2110 PRINT ID ;"  ";HOEHE(ID);"  ";WINKEL(ID);"
      ";MAGNET$(ID)
 2120 IF HOEHE(ID)=-1 THEN GOTO 2000
*2150 IF USR(E8)=1 THEN GOTO 4000
*2200 D=USR(EX)-HOEHE(ID)
*2210 IF D>0 THEN SYS M2,RECHTS
*2220 IF D<0 THEN SYS M2,LINKS
 2230 IF D=0 THEN GOTO 2400
 2240 LET D=ABS(D)
 2250 IF D>GX THEN GOTO 2150
*2300 SYS M2,AUS
 2330 GOTO 2120
 2400 REM WINKEL EINSTELLEN
*2410 IF USR(E8)=1 THEN GOTO 4000
*2420 D=USR(EY)-WINKEL(ID)
*2430 IF D>0 THEN SYS M1,RECHTS
*2440 IF D<0 THEN SYS M1,LINKS
 2450 IF D=0 THEN GOTO 2600
 2460 LET D=ABS(D)
 2470 IF D>GY THEN GOTO 2410
*2520 SYS M1,AUS
 2530 GOTO 2410
 2600 REM MAGNET SCHALTEN
*2610 IF MAGNET$(ID)="EIN" THEN SYS M3,EIN
 2620 FOR I=0TO 100
*2630 IF USR(E8)=1 THEN 4000
 2640 NEXT
 2650 IF MAGNET$(ID)="AUS"THEN GOSUB 2800
 2700 GOTO 2100
 2800 REM MAGNET AUS ROUTINE
*2810 SYS M3,LINKS
 2820 REM MAGNET UMGEPOLT ZUR ENTMAGNETISIERUNG
*2030 SYS M3,AUS :REM MAGNET AUSGESCHALTET
 2840 RETURN
 2900 REM TITELMELDUNG
*2810 PRINT CHR$(147)
 2920 PRINT"FISCHERTECHNIK"
 2930 PRINT"COMPUTING"
 2940 PRINT
 2950 PRINT"TEACH IN ROBOTER"
 2960 PRINT
 2970 PRINT"TEACH IN TABELLE"
 2980 PRINT
 2990 PRINT"NR. HOEHE  WINKEL  MAGNET"
 3000 RETURN
 4000 REM ROBOTER ABSTELLEN
*4010 SYS INIT
```

Grafiktablett

CAD – Computer aided design – ist eines der neuen Schlagworte der Computertechnologie. Dabei wird das Reißbrett durch Bildschirm und Plotter ersetzt. Die Konstruktion der Zeichnung erfolgt durch die Bewegung eines Griffels auf einem sogenannten Grafiktablett. Dabei werden die unterschiedlichsten technischen Lösungen eingesetzt. Wir verwenden hier eine einfache Lösung mit einem Knickarm, dessen Endpunkt über die Zeichenfläche geführt wird (Abb. 9).

In den Angelpunkten des Arms sind jeweils Potentiometer angebracht.
Anhand von Abb. 1 wollen wir uns die geometrischen Zusammenhänge veranschaulichen. Zur Abspeicherung im Computer und zur Erstellung von Zeichnungen auf dem Bildschirm benötigen wir die Koordinaten des Bildpunktes in waagerechter und senkrechter Richtung, x und y. Der Ursprung dieses sogenannten cartesischen Koordinatensystems liegt in der linken unteren Ecke der Zeichenfläche.



$x = x_0 - a \cdot \cos\alpha + a \cdot \cos(\beta - \alpha)$

$y = y_0 - a \cdot \sin\alpha - a \cdot \sin(\beta - \alpha)$

Abb. 9

Mit (x_2, y_0) haben wir die Position des ersten Drehpunktes, nämlich den Mittelpunkt des Drehkranzes, bezeichnet. Von da aus folgen wir in Gedanken der ersten Hälfte des Armes und kommen zu dem zweiten Drehpunkt. Wer die Grundkenntnisse der Trigonometrie parat hat, erkennt leicht, daß die neuen Koordinaten um die Achsabschnitte $a \cdot \cos\alpha$ und $a \cdot \sin\alpha$ verschoben liegen. Dabei bezeichnet a die Länge des „Oberarms" und α den Winkel des Potentiometers.
Weiter geht es entlang des „Unterarmes" zum Griffel. Damit die Berechnung vereinfacht wird, hat dieses Teilstück gleichfalls die Länge a. Wohl dem, der sich der Schulmathematik über Winkel an Parallelen und Geraden erinnert. Er wird erkennen, daß der Winkel zur Waagrechten sich aus der Differenz der beiden Potentiometerwinkel α und β ergibt. Nehmen wir auch noch die zu diesem Winkel gehörigen Achsabschnitte hinzu, so erhalten wir die Endformeln, wie sie in Abb. 1 angegeben sind.
Setzen wir in die Formel α und β in Winkelgraden und a, x_0 und y_0 in Millimeter ein, so erhalten wir die korrekte Lage des Griffels auf dem Grafiktablett. Es gilt nun, diese Position auf dem Bildschirm des Computers wiederzufinden.

Zunächst benötigen Sie hierzu ein Grafiksystem in Ihrem Computer. Die meisten Computer haben ein solches eingebaut, bei den anderen läßt es sich im Rahmen einer BASIC-Befehlserweiterung nachrüsten. Die hier vorliegende Dokumentation geht von der Commodore BASIC-Erweiterung SIMON's BASIC aus. Aus Gründen des Urheberrechts ist SIMON's BASIC *nicht* auf der fischertechnik computing Diskette enthalten.

Im nächsten Schritt müssen Sie feststellen, wo der Ursprung des Graphikschirms sitzt. Nicht alle Grafiksysteme halten sich an die in der Mathematik übliche Konvention, sondern legen den Ursprung in die linke *obere* Ecke.
Außerdem spielt die Grafikauflösung eine Rolle. Wenn Sie sich das Leben vereinfachen wollen, nehmen Sie eine leichte Verzerrung in Kauf. Strecken Sie das Seitenverhältnis des Grafiktabletts auf jenes des Bildschirms. Hierzu ermitteln Sie die Strekkungsfaktoren in x- und y-Richtung.

f = (Anzahl der Bildpunkte in x-Richtung)/260 mm
g = (Anzahl der Bildpunkte in y-Richtung)/187 mm

Die Bildschirmkoordinaten X und Y ergeben sich dann als

X = f * x
Y = g * y bzw.
Y = (Anzahl der Bildpunkte in y-Richtung) – g * y

Als letztes fehlt noch die Umrechnung der Funktionswerte von USR (EX) und USR (EY) in Winkelgrade.
Bislang hatte sich diese Frage nicht gestellt, da die Werte empirisch ermittelt wurden.
Am besten wir eichen das Grafiktablett jeweils zu Beginn des Programms. Dazu führen wir den Griffel in die linke obere Ecke der Zeichenfläche und übernehmen die Zahlenwerte A1 und B1 der beiden Potentiometer. Anschließend wiederholen wir dies für die rechte untere Ecke der Zeichenfläche und notieren A2 und B2.
A1 entspricht dem Winkel α_1, B1 dem Winkel β_1 usw., so daß wir folgende Umrechnung zwischen beliebigen Zahlenwerten AN, BN und den Winkeln α_n, β_n erhalten:

$$\alpha_n = \alpha_1 + \left(\frac{\alpha_1 - \alpha_2}{A1 - A2}\right) * (AN - A1)$$

$$\beta_n = \beta_1 + \left(\frac{\beta_1 - \beta_2}{B1 - B2}\right) * (BN - B1)$$

Um die Koordinatenumrechnung programmieren zu können, setzen wir die Konstanten aus der Tabelle in die Formel ein. Sie ergeben sich aus den Abmessungen des Grafiktabletts.

Bei der Programmierung berücksichtigen wir noch, daß die Taste am rechten Rand der Zeichenfläche (E1) verwendet wird, um dem Computer mitzuteilen, wann der Griffel positioniert ist. Die weiteren Taster steuern die Grafikfunktionen. Um eine genügend große Zahl von Funktionen realisieren zu können, sind die Tasten E3 bis E8 doppelt belegt. Zur Umschaltung dient die Taste E2, ähnlich wie die „Shift"-Taste an der Computertastatur. Folgende Grafikfunktionen sind in dem anschließend aufgelisteten Beispielprogramm eingebaut:

E3	–	Zeichne Punkt
E4	–	Zeichne Linie
E5	–	Zeichne Dreieck
E6	–	Zeichne Rechteck
E7	–	Zeichne Kreis
E8	–	Lösche Schirm
E3 + E2	–	Übertrage Buchstabe
E4 + E2	–	Zeichne beliebige Linie
E5 + E2	–	Zeichne ausgefülltes Dreieck
E6 + E2	–	Zeichne ausgefülltes Rechteck
E7 + E2	–	Zeichne ausgefüllten Kreis
E8 + E2	–	Male Gebiet aus

Zum Schluß noch eine Anregung zur weiteren Nutzung des Grafiktabletts. Verwenden Sie als Vorlage eine aufgezeichnete Klaviertastatur, um (in der Waagrechten) die Frequenz des Tongenerators ihres Computers durch Anwahl der Taste zu finden. In der Senkrechten ist die Tonlänge aufgetragen. Entwickeln Sie ein Programm, mit dem Sie Klavier spielen oder komponieren können. Und nun wünschen wir viel Spaß beim Zeichnen und Musizieren.

Tabelle der geometrischen Daten
des Grafiktabletts

$\alpha_1 = -39°$ $\beta_1 = 53{,}7°$ $x_0 = 137$ mm $a = 166$ mm
$\alpha_2 = 82{,}5°$ $\beta_2 = 112{,}2°$ $y_0 = 247$ mm

```
*500 SYS INIT
 510 REM
 520 REM FISCHERTECHNIK COMPUTING
 530 REM
 540 REM GRAFIKTABLETT
 550 REM
 560 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
 570 REM
 580 REM BELEGUNG DES INTERFACE
 590 REM
 600 REM EINGABEN
 610 REM E1=POSITIONSTASTE
 620 REM E2=SHIFT-TASTE
 630 REM E3=PUNKT/ZEICHEN
 640 REM E4=STRECKE/LINIE
 650 REM E5=DREIECK
 660 REM E6=RECHTECK
 670 REM E7=KREIS
 680 REM E8=SCHIRM LOESCHEN/AUSMALEN
 690 REM EX=BETA
 700 REM EY=ALPHA
 710 REM
 720 REM AUSGABEN
 730 REM M1=GRAPHIK AKTIV
 740 REM M2=EINGABE BEREIT
 750 REM
 760 REM FUNKTION
 770 REM UEBERTRAGUNG EINER GRAFISCHEN
 780 REM VORLAGE IN DEN BILDSPEICHER DES
 790 REM COMPUTERS
 800 REM
 810 REM GEOMETRISCHE DATEN
 820 LET WA=-0.68068 :REM WINKEL ALPHA LINKS OBEN
 830 LET WB=0.93724 :REM WINKEL BETA LINKS OBEN
 840 LET A=166 :REM ARMLAENGE
 850 LET X0=137 :REM POSITION DREHKRANZ
 860 LET Y0=61
 870 LET SX=1.2364
 880 REM SKALENFAKTOR X-ACHSE BILDPUNKTE/MM
 890 LET SY=1.0699
 900 REM SKALENFAKTOR Y-ACHSE BILDPUNKTE/MM
 910 REM
 920 PRINT CHR$(147)
 930 PRINT"FISCHERTECHNIK"
 940 PRINT"COMPUTING"
 950 PRINT
 960 PRINT"GRAFIKTABLETT"
 970 PRINT
 980 PRINT"DIESES PROGRAMM ERFORDERT EINE BASIC"
 990 PRINT"ERWEITERUNG UM GRAFIKBEFEHLE"
1000 PRINT
1010 PRINT"BITTE EINEICHEN"
1020 PRINT
1030 PRINT"GRIFFEL IN LINKE"
1040 PRINT"OBERE ECKE!"
1050 PRINT"POS.TASTE DRUECKEN!"
*1060 IF USR(E1)=0 THEN GOTO 1060
1070 REM WARTE AUF POS.TASTE
*1080 LET A1=USR(EY)
*1090 LET B1=USR(EX) :REM GRIFFELPOSITION
*1100 IF USR(E1)=1 THEN GOTO 1100
1110 REM WARTE BIS TASTE LOSGELASSEN
1120 PRINT
1130 PRINT"GRIFFEL IN RECHTE"
1140 PRINT"UNTERE ECKE!"
1150 PRINT"POS.TASTE DRUECKEN!"
*1160 IF USR(E1)=0 THEN GOTO 1160
1170 REM WARTE AUF POS.TASTE
*1180 LET A2=USR(EY)
*1190 LET B2=USR(EX) :REM GRIFFELPOSITION
*1200 IF USR(E1)=1 THEN GOTO 1200
1210 REM WARTE BIS TASTE LOSGELASSEN
1220 LET AF=2.12858/(A2-A1)
1230 LET BF=1.02102/(B2-B1)
*1240 HIRES 7,6 :REM HOCHAUFLOESENDE GRAFIK (GE/BL)
*1250 SYS M1,AUS :REM AKTIVANZEIGE AUS
*1260 SYS M2,EIN :REM BEREITANZEIGE EIN
1270 FOR I=2 TO 7 :REM SCHLEIFE KOMMANDOTASTEN
*1280 S=USR(E2)*7 :REM SHIFTTASTE
1290 LET EJ=2+I
*1300 IF USR(EJ)=0 THEN GOTO 1330
1310 LET S=S+I-1
1320 LET I=7 :REM SCHLEIFE BEENDEN
1330 NEXT I
```

```
 1340 IF(S=0)OR(S=7) THEN GOTO 1270
*1350 SYS M2,AUS :REM BEREITANZEIGE AUS
*1360 SYS M1,EIN :REM AKTIVANZEIGE EIN
 1370 ON S GOTO 1380,1410,1470,1490,1570,1240,1250,
      1590,1650,1720,1770,1830,1860
 1380 GOSUB 3000 :REM KOORDINATEN
*1390 PLOT X,Y,1 :REM PUNKT ZEICHNEN
 1400 GOTO 1250
 1410 GOSUB 3000 :REM KOORDINATEN
 1420 LET X1=X
 1430 LET Y1=Y :REM UMSPEICHERN
 1440 GOSUB 3000
*1450 LINE X1,Y1,X,Y,1 :REM STRECKE ZEICHNEN
 1460 GOTO 1250
 1470 GOSUB 5010:REM DREIECK ZEICHNEN
 1480 GOTO 1250
 1490 GOSUB 3000 :REM KOORDINATEN
 1500 LET X1=X
 1510 LET Y1=Y :REM UMSPEICHERN
 1520 GOSUB 3000 :REM KOORDINATEN
 1530 LET H=Y-Y1 :REM HOEHE
 1540 LET B=X-X1 :REM BREITE
*1550 REC X1,Y1,B,H,1 :REM RECHTECK ZEICHNEN
 1560 GOTO 1250
 1570 GOSUB 6010:REM KREIS ZEICHNEN
 1580 GOTO 1250
 1590 GOSUB 3000 :REM KOORDINATEN
 1600 GET C$ :REM LIES ZEICHEN
 1610 IF C$="" THEN GOTO 1600
*1620 C=ASC(C$)-64 :REM BILDSCHIRMCODE
*1630 CHAR X,Y,C,1,1 :REM ZEICHEN DARSTELLEN
 1640 GOTO 1250
 1650 GOSUB 3000 :REM STARTSIGNAL
*1660 IF USR(E1)=1 THEN GOTO 1250
 1670 GOSUB 4000:REM KOORDINATEN
*1680 LINE X1,Y1,X,Y,1
 1690 LET X=X1 :REM UMSPEICHERN
 1700 LET Y=Y1
 1710 GOTO 1660
 1720 GOSUB 5010
 1730 LET XM=(X1+X2+X)/3
 1740 LET YM=(Y1+Y2+Y)/3
*1750 PAINT XM,YM,1 :REM FLAECHE AUSMALEN
 1760 GOTO 1250
 1770 GOSUB 3000 :REM KOORDINATEN
 1780 LET X1=X
 1790 LET Y1=Y :REM UMSPEICHERN
 1800 GOSUB 3000 :REM KOORDINATEN
*1810 BLOCK X1,Y1,X,Y,1 :REM BLOCK ZEICHNEN
 1820 GOTO 1250
 1830 GOSUB 6010:REM KREIS ZEICHNEN
*1840 PAINT X1,Y1,1 :REM KREIS AUSMALEN
 1850 GOTO 1250
 1860 GOSUB 3000 :REM KOORDINATEN
*1870 PAINT X,Y,1 :REM FLAECHE AUSMALEN
 1880 GOTO 1250
 3000 REM KOORDINATEN EINLESEN
*3010 IF USR(E1)=0 THEN GOTO 3010
*3020 LET ALPHA=(USR(EY)-A1)*AF+WA
*3030 LET BETA=(USR(EX)-B1)*BF+WB
 3040 REM WINKEL EINLESEN
 3050 LET X=INT(SX*(X0-A*COS(ALPHA)+A*COS(BETA-ALPHA)))
 3060 LET Y=INT(SY*(A*SIN(ALPHA)+A*SIN(BETA-ALPHA)-Y0))
 3070 IF X<0 THEN LET X=0
 3080 IF X>319 THEN LET X=319
 3090 IF Y<0 THEN LET Y=0
 3100 IF Y>199 THEN LET Y=199
*3110 IF USR(E1)=1 THEN GOTO 3110
 3120 RETURN
 4000 REM KOORDINATEN EINLESEN OHNE ZU WARTEN
*4010 LET ALPHA=(USR(EY)-A1)*AF+WA
*4020 LET BETA=(USR(EX)-B1)*BF+WB
 4030 REM WINKEL EINLESEN
 4040 LET X1=INT(SX*(X0-A*COS(ALPHA)+A*COS(BETA-ALPHA)))
 4050 LET Y1=INT(SY*(A*SIN(ALPHA)+A*SIN(BETA-ALPHA)-Y0))
 4060 IF X1<0 THEN LET X1=0
 4070 IF X1>319 THEN LET X1=319
 4080 IF Y1<0 THEN LET Y1=0
 4090 IF Y1>199 THEN LET Y1=199
 4100 RETURN
 5000 REM UNTERPROGRAMM DREIECK
 5010 GOSUB 3000 :REM KOORDINATEN
 5020 LET X1=X
 5030 LET Y1=Y :REM UMSPEICHERN
 5040 GOSUB 3000 :REM KOORDINATEN
 5050 LET X2=X
 5060 LET Y2=Y :REM UMSPEICHERN
 5070 GOSUB 3000 :REM KOORDINATEN
*5080 LINE X1,Y1,X2,Y2,1 :REM DREIECK ZEICHNEN
*5090 LINE X2,Y2,X,Y,1
*5100 LINE X,Y,X1,Y1,1
 5110 RETURN
 6000 REM UNTERPROGRAMM KREIS
 6010 GOSUB 3000 :REM KOORDINATEN
 6020 LET X1=X
 6030 LET Y1=Y :REM UMSPEICHERN
 6040 GOSUB 3000 :REM KOORDINATEN
 6050 LET R=SQR((X1-X)↑2+(Y1-Y)↑2)
*6060 CIRCLE X1,Y1,R,R,1 :REM KREIS ZEICHNEN
 6070 RETURN
```

Plotter

Zum Schluß wollen wir Ihnen noch ein paar Anregungen geben, was Sie mit den Bauteilen aus fischertechnik computing noch anfangen können. Ein Gerät, das sehr häufig an Computern zu finden sein wird, ist ein Plotter. Ein Plotter dient dazu, Ergebnisse des Computers in einer grafischen Form zu Papier zu bringen. Wir wollen jetzt nicht behaupten, daß mit fischertechnik computing ein vollwertiger Plotter gebaut werden kann – dafür gibt es einen eigenen Bausatz im fischertechnik computing-Programm. Aber für ein paar kleine Grafiken reicht die Maschine schon, die in der Bauanleitung dargestellt ist. Unter Zuhilfenahme der Bauelemente, die in fischertechnik computing vorhanden sind, können wir einen Polarkoordinatenplotter bauen. Bei einem solchen Gerät wird die Position über dem Papierblatt durch die Drehbewegung des Drehkranzes und durch das Vor- und Zurückfahren des Schreibkopfes längs des Arms bewirkt. Das Schreiben selbst wird über den Magneten gesteuert. Der eingeschaltete Magnet zieht den Schreibstift nach unten auf das Papier.

Wenn wir den Plotter programmieren, müssen wir vor allen Dingen die Koordinatenumrechnung beachten. Normalerweise denken wir in rechtwinkligen Koordinaten: z.B. Länge und Breite, in der Mathematik meist als x und y bezeichnet. Ein solches Koordinatensystem wird auch als cartesisches Koordinatensystem (nach dem französischen Mathematiker Descartes) bezeichnet. Unser Plotter arbeitet jedoch entsprechend seiner Mechanik in einem Polarkoordinatensystem mit den charakteristischen Koordinaten Drehung und Radius. Diese werden in der Regel als φ und r bezeichnet. Zwischen beiden Koordinatensystemen kann umgerechnet werden:

$$x = r * \cos(\varphi)$$
$$y = r * \sin(\varphi)$$

Als Beispielprogramm haben wir die Erstellung einer sog. Kuchengrafik aufgegriffen (bei der übrigens die vorgenannte Umrechnung nicht nötig ist). Solche Grafiken haben Sie sicher schon öfters gesehen: Ob Sitzverteilung in einem Parlament oder Marktanteil von Produkten – oft wird in diesen Fällen ein Kreissegment in „Kuchenstücke" aufgeteilt. Unser Programm kann dies auch. Nachdem der Arbeitsbereich des Plotters festgelegt ist, frägt das Programm nach der Anzahl der Segmente und deren jeweiligem Anteil. Dabei müssen die Anteile nicht vorher prozentual ausgerechnet werden, eine solche Kleinigkeit kann natürlich der Computer für uns erledigen. Wenn die Eingabe komplett ist, werden zunächst der Rahmen und anschließend die Segmentgrenzen gezeichnet.

Der Plotter wäre nicht von fischertechnik computing, ließe er sich nicht auch noch für andere Zwecke einsetzen. Wir überlassen es Ihnen als geschicktem fischertechnik-Freund, den Schreibkopf so umzubauen, daß die Magnetpolschuhe mit geringem Abstand über der Grundplatte geführt werden. Schon haben wir einen Portalroboter, der Münzen oder die beigefügten Blechscheiben transportieren kann. Zwar hat er keine so große Bewegungsfreiheit nach oben wie die beiden anderen Roboter, dafür kann er jedoch sein Arbeitsfeld in verschiedenen Radien überstreichen. Probieren Sie mal, einen eigenen Spielroboter z.B. mit dem Nim-Spiel zu programmieren!

```
*500 SYS INIT
 510 REM
 520 REM FISCHERTECHNIK COMPUTING
 530 REM
 540 REM PLOTTER
 550 REM
 560 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
 570 REM
 580 REM SEGMENT TABELLE
 590 DIM SEG(9)
 600 REM KONSTANTEN
 610 LET GR=30 :REM BREMSPHASE R
 620 LET GA=5 :REM BREMSPHASE A
 630 REM BELEGUNG DES INTERFACE
 640 REM
 650 REM EINGABEN
 660 REM E3=UEBERNAHME
 670 REM E4=RECHTS
 680 REM E5=LINKS
 690 REM E6=VORWAERTS
 700 REM E7=RUECKWAERTS
 710 REM EX RADIALPOSITION
 720 REM EY WINKELPOSITION
 730 REM
 740 REM AUSGABEN
 750 REM M1=DREHBEWEGUNG
 760 REM M2=RADIALBEWEGUNG
 770 REM M3=MAGNET
 780 REM
 790 REM FUNKTION
 800 REM DAS PROGRAMM STEUERT DEN PLOTTER,
 810 REM DER EINE KREISSEGMENTGRAFIK AUSGIBT.
 820 REM NACH DEM PROGRAMMSTART WIRD DER PLOTTER
 830 REM ZUNACHST EINGEEICHT.
 840 REM DANACH WERDEN DIE ZAHLENWERTE EINGEGEBEN
 850 REM UND DIE GRAFIK ERSTELLT.
 860 REM
*870 PRINT CHR$(147)
 880 PRINT"FISCHERTECHNIK"
 890 PRINT"COMPUTING"
 900 PRINT
 910 PRINT"PLOTTER"
1000 REM
1010 LET A$="PLOTTER BITTE UEBER DIE TASTER NACH"
1020 PRINT
1030 PRINT A$
1040 PRINT"RECHTS AUSSEN STEUERN"
1050 GOSUB 3000
1060 LET RM=R
1070 LET AM=A
1080 PRINT A$
1090 PRINT"LINKS INNEN STEUERN"
1100 GOSUB 3000
1110 LET R0=R
1120 LET A0=A
1130 REM DATENEINGABE
1140 PRINT
1150 INPUT"ANZAHL DER SEGMENTE";SG
1160 IF SG=0 THEN END
1170 LET SS=0
1180 FOR S=1 TO SG
1190 PRINT"ANTEIL NR.";S;
1200 INPUT SEG(S-1)
1210 LET SS=SS+SEG(S-1)
1220 NEXT S
1230 REM NORMIEREN
1240 FOR S=1 TO SG
1250 LET SEG(S-1)=SEG(S-1)/SS
```

```
 1260 NEXT S
 1270 PRINT"BITTE PAPIER EINLEGEN!"
 1280 PRINT"TASTE E3, WENN FERTIG!"
*1290 IF USR(E3)=0 THEN GOTO 1290
 1300 REM PLOTTEN
 1310 REM 1. RAHMEN
 1320 LET R=R0
 1330 LET PEN=AUS
 1340 GOSUB 4000 :REM R-BEWEGUNG
 1350 LET A=A0
 1360 LET PEN=AUS
 1370 GOSUB 5000:REM A-BEWEGUNG
 1380 LET A=AM
 1390 GOSUB 6000 :REM STIFT EIN
 1400 GOSUB 5000
 1410 LET R=RM
 1420 LET PEN=EIN
 1430 GOSUB 4000 :REM R-BEWEGUNG
 1440 LET A=A0
 1450 GOSUB 6000 :REM STIFT EIN
 1460 GOSUB 5000:REM A-BEWEGUNG
 1470 LET R=R0
 1480 GOSUB 6000 :REM STIFT EIN
 1490 GOSUB 4000 :REM R-BEWEGUNG
 1500 REM 2.SEGMENTGRENZEN PLOTTEN
 1510 LET SS=0
 1520 FOR S=1 TO SG-1
 1530 LET SS=SS+SEG(S-1)
 1540 LET A=A0+INT((AM-A0)*SS+.5)
 1550 LET PEN=AUS
 1560 GOSUB 5000:REM A-BEWEGUNG
 1570 LET R=RM
 1580 GOSUB 6000 :REM STIFT EIN
 1590 GOSUB 4000 :REM R-BEWEGUNG
 1600 LET R=R0
 1610 LET PEN=AUS
 1620 GOSUB 4000 :REM R-BEWEGUNG
 1630 NEXT S
 1640 GOTO 1140
 1650 END
 3000 REM UEBERNAHME DER POSITION
*3010 SYS INIT :REM ALLE MOTOREN AUS
*3020 IF USR(E4)=0 THEN GOTO 3050
*3030 SYS M1,RECHTS
 3040 GOTO 3020
*3050 SYS M1,AUS
*3060 IF USR(E5)=0 THEN GOTO 3090
*3070 SYS M1,LINKS
 3080 GOTO 3060
*3090 SYS M1,AUS
*3100 IF USR(E6)=0 THEN GOTO 3130
*3110 SYS M2,RECHTS
 3120 GOTO 3100
*3130 SYS M2,AUS
*3140 IF USR(E7)=0 THEN GOTO 3170
*3150 SYS M2,LINKS
 3160 GOTO 3140
*3170 SYS M2,AUS
*3180 IF USR(E3)=0 THEN GOTO 3010
*3190 LET R=USR(EX)
*3200 LET A=USR(EY)
*3210 IF USR(E3)=1 THEN GOTO 3210
 3220 REM WARTE, BIS TASTE LOSGELASSEN
 3230 RETURN
 4000 REM R-BEWEGUNG
*4010 SYS M3,PEN
*4020 LET D=USR(EX)-R
*4030 IF D>0 THEN SYS M2,RECHTS
*4040 IF D<0 THEN SYS M2,LINKS
 4050 IF D=0 THEN RETURN
 4060 LET D=ABS(D)
 4070 IF D>GR THEN GOTO 4020
 4080 FOR I=0 TO D
 4090 NEXT I
*4100 SYS M2,AUS
 4110 GOTO 4020
 5000 REM A-BEWEGUNG
*5010 SYS M3,PEN
*5020 LET D=USR(EY)-A
*5030 IF D>0 THEN SYS M1,RECHTS
*5040 IF D<0 THEN SYS M1,LINKS
 5050 IF D=0 THEN RETURN
 5060 LET D=ABS(D)
 5070 IF D>GA THEN GOTO 5020
 5080 FOR I=0 TO D
 5090 NEXT I
*5100 SYS M1,AUS
 5110 GOTO 5020
 6000 REM STIFT EINSCHALTEN
 6010 LET PEN=EIN
 6020 FOR I=0 TO 100
*6030 SYS M3,PEN :REM VERZOEGERUNGSSCHLEIFE
 6040 NEXT
 6050 RETURN
```

Solarzellennachführung

Solarzellen können in bestimmten Regionen der Erde die Energieversorgung anstelle von Verbrennungs- oder Kernkraftwerken übernehmen. Wichtig dabei ist, daß die Solarzelle möglichst viel der einfallenden Strahlung in elektrische Energie umsetzt. Und das tut sie genau dann, wenn sie senkrecht zur Einfallsrichtung steht.
Damit kommen wir auf unsere Programmieraufgabe zurück, nämlich eine Himmelsrichtung fest einzuhalten, obwohl die Erde sich im Laufe eines Tages mitsamt des „Fundaments" der Solarzellenanlage einmal um ihre Achse dreht und dazu noch im Verlaufe eines Jahres in Schräglage um die Sonne reist (Abb. 10).

Aus beiden Bewegungen ergeben sich der Tagesverlauf der Sonne mit seinen charakteristischen jahreszeitlichen Abhängigkeiten. Und damit hängt der Sonnenstand nicht nur von Uhrzeit und Datum, sondern auch von der geographischen Breite ab. Denken Sie an die Polarnacht, die an den Polen ein halbes Jahr dauert, genauso wie dort im Sommer die Sonne nie untergeht. Die Polarkreise begrenzen die Polkappen, wo mindestens an einem Sommertag die Sonne nicht untergeht bzw. an einem Wintertag die Sonne nicht aufgeht. Dies sind dann der 21. Juni und der 21. Dezember.
Wichtige Daten sind auch der Frühlingsanfang (20. März) und Herbstanfang (23. September). An diesen Tagen ist Tag und Nacht überall auf der Erde gleichlang. Am Äquator geht dann die Sonne senkrecht auf, über den Zenith und senkrecht unter.
Sie können sich vorstellen, daß die mathematischen Formeln, die diese Bewegung beschreiben, recht kompliziert sind. In der Tat beruhen die Gleichungen auf den Formeln der sphärischen Trigonometrie.
Doch zunächst müssen wir uns mit einem näherliegenden Problem befassen. Wieviel Uhr ist es? Kein Problem, werden Sie denken und geben, wenn es sein muß, die Uhrzeit quarzgenau auf die Sekunde an. Doch dies ficht eine Sonnenuhr wenig an. Was sie angibt, ist die wahre Sonnenzeit, die sich von unserer üblichen Zeit in einigen Punkten unterscheidet. So berücksichtigt sie, daß wir jedes Jahr um einen Vierteltag betrogen werden, den wir dann im Schaltjahr nachfeiern. Weiter ist unsere Zeit immer in Zeitzonen einheitlich, z.B. Westeuropäische Zeit, Mitteleuropäische Zeit, Osteuropäische Zeit. In Amerika gibt es gar sieben Zeitzonen, die sich vom Atlantik bis Hawaii erstrecken. Die wahre Sonnenzeit ist jedoch als astronomische Größe von solchen mehr oder weniger willkürlichen Zoneneinteilungen unabhängig und errechnet sich aus der geographischen Länge des Beobachtungsortes. Da die Erde nicht in einer Kreis-, sondern einer Ellipsenbahn um die Sonne läuft, ergeben sich noch weitere feine Abweichungen.
All dies ist in dem nachstehenden BASIC-Programm berücksichtigt, darüber hinaus noch die über Jahre hinweg erfolgende langsame Änderung des Zeitmaßstabes, die auf Störungen der Erdbahn durch andere Planeten sowie die Gezeitenreibung zurückgeht. Die Genauigkeit des Programms in den Bahnpositionen ist besser als 1°. Dies ist zwar sicherlich genauer, als die mechanische Präzision der Solarzellennachführung und die Winkelbestimmung per Potentiometer. Andererseits mögen Sie als Hobbyastronom vielleicht einmal Verwendung für dieses Programm haben.

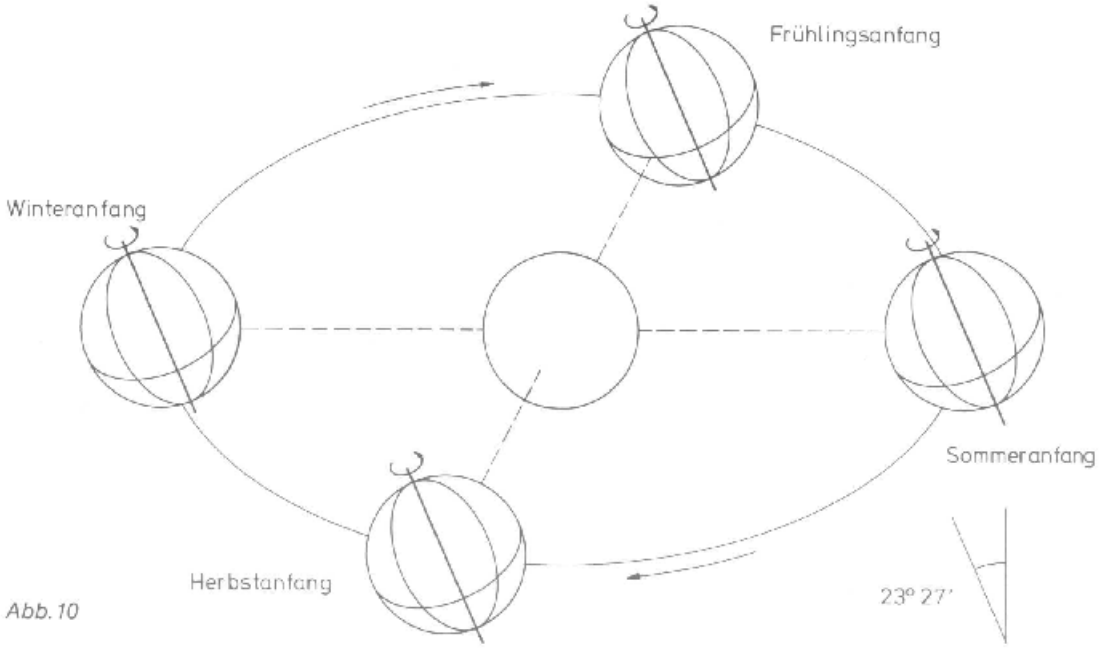


Abb. 10

Fassen wir zusammen, was das Programm als Eingabe benötigt:
Die Zeitangabe als Jahr, Tag im Jahr, Stunde, Minute und Sekunde sowie die Zeitzone, auf die sich diese Angabe bezieht. Berücksichtigen Sie dabei eventuell die Sommerzeit!
Die Ortsangabe in Form der geographischen Breite und Länge. Diese erhalten Sie aus einer Karte oder einem Atlas für Ihren Standort.
Das Programm liefert Ihnen die Winkel der beiden Bewegungsachsen der Solarzellennachführung. Dies ist zum einen der Azimuth, der die Abweichung zur Südrichtung angibt und zum anderen die Höhe, die die Neigung relativ zur Horizontlinie beschreibt. Wie zuvor bei dem Grafiktablett beschrieben, werden die Potentiometerwerte durch Eichung in die Winkelgrade umgerechnet. Wer es genauer haben möchte, führt diese Eichung in mehreren Bahnpunkten durch. Die Ergebnisse werden dann zeichnerisch aufgetragen: Potentiometerwert gegen Winkeleinstellung. Was Sie vor sich haben, wenn Sie die Punkte untereinander verbinden, ist die individuelle Linearitätskurve Ihres Potentiometers, die Sie sich als Datenblatt aufbewahren sollten. In der Literatur sind hinreichend Methoden beschrieben, wie Sie eine solche Kurve zur Umrechnung heranziehen.
Die Steuerung der Motoren wird ähnlich den Robotermodellen vorgenommen. Allerdings empfiehlt es sich, die Motoren kurz vor Erreichen der Sollwertvorgabe nur noch mit ganz kurzen Impulsen zu versorgen, damit der Motor nicht über das Ziel hinausschießt. Schließlich haben Sie hier ja Zeit; es ist allemal ausreichend, die Nachführanlage alle zehn Minuten zu positionieren. Bei einem gerafften Zeitablauf 1 : 720 zu Demonstrationszwecken (1 Tag = 2 Minuten) dürfte allerdings dann die Leistungsgrenze der üblichen BASIC-Interpreter erreicht sein.

```
*500 SYS INIT
 510 REM
 520 REM FISCHERTECHNIK COMPUTING
 530 REM
 540 REM SOLARZELLENNACHFUEHRUNG
 550 REM
 560 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
 570 REM
 580 REM BELEGUNG DES INTERFACE
 590 REM
 600 REM EINGABEN
 610 REM E0=HANDSTEUERUNG AUF/RECHTS
 620 REM E4=HANDSTEUERUNG AB/LINKS
 630 REM E5=MOTORAUSWAHL
 640 REM E6=UEBERNAHMESIGNAL
 650 REM EX=HORIZONTHOEHE
 660 REM EY=AZIMUTH
 670 REM
 680 REM AUSGABEN
 690 REM M1=AZIMUTH-MOTOR
 700 REM M2=HOEHENEINSTELLUNG
 710 REM
 720 REM FUNKTION
 730 REM DAS NACHFOLGENDE PROGRAMM ERRECHNET DEN
 740 REM SONNENSTAND UND STEUERT DIE
       SOLARZELLENNACHFUEHRANLAGE.
 750 REM ALS EINGABEPARAMETER WERDEN VERLANGT:
 760 REM DIE JAHRESZAHL
 770 REM DER TAG IM JAHR, BEGINNEND MIT 1 AM 1.JANUAR
 780 REM AUSNAHME: IN SCHALTJAHREN BEGINNT DIE
 790 REM ZAEHLUNG BEI 0.
 800 REM STUNDE, MINUTE UND SEKUNDE DER LOKALEN ZEITZONE
 810 REM DIE ZEITZONE RELATIV ZU GREENWICH UNIVERSAL
 820 REM TIME, DABEI WIRD NACH WESTEN POSITIV UND
 830 REM NACH OSTEN NEGATIV GEZAEHLT.
 840 REM DIE LOKALE BREITE (NACH NORDEN POSITIV)
 850 REM DIE LOKALE LAENGE IN GRAD WESTLICH GREENWICH
 860 REM QUELLE: R.WALRAVEN IN SOLAR ENERGY, VOL.20
 870 REM (1978), S.393-397
 880 REM
 890 REM FUNKTIONSDEFINITION ARCUS SINUS
 900 DEF FNAS(X)=ATN(X/SQR(1-X*X))
 910 REM
 920 REM ZAHLENKONSTANTEN 2** UND 4/180
 930 DATA 6.2831853,0.017453293
 940 READ ZWEIPI,RD
 950 LET AL=20 :REM GRENZWERT POTENTIOMETER A LINKS
 960 LET AR=200 :REM GRENZWERT POTENTIOMETER A RECHTS
 970 LET EM=20 :REM GRENZWERT POTENTIOMETER E LINKS
 980 LET ER=200 :REM GRENZWERT POTENTIOMETER E RECHTS
 990 LET EA=EM
1000 LET GA=5 :REM BREMSPHASE AZIMUTH
1010 REM DAS PROGRAMM BENOETIGT EINE ZEITFUNKTION TI$.
1020 REM DER ABLAUF WIRD BEI EINGABE EINER STARTZEIT
       VON 0.00 UHR
1030 REM ODER BEI NICHTVORHANDENSEIN DER FUNKTION TI$
1040 REM GERAFFT DURCHGEFUEHRT.

*1050 PRINT CHR$(147)
 1060 PRINT"FISCHERTECHNIK"
 1070 PRINT"COMPUTING"
 1080 PRINT
 1090 PRINT"SOLARZELLEN"
 1100 PRINT"NACHFUEHRUNG"
 1110 PRINT
 1120 INPUT"JAHR";JAHR
 1130 INPUT"TAG IM JAHR";TAG
 1140 INPUT"ZEITZONE";ZZ
 1150 INPUT"BREITE";BREITE
 1160 INPUT"LAENGE";LAENGE
 1170 REM JUSTAGE
 1180 PRINT"SOLARZELLE BITTE EINJUSTIEREN"
 1190 LET A$="SOLARZELLE EXAKT NACH "
 1200 PRINT A$+" OSTEN"
 1210 PRINT"BLICKRICHTUNG AUF DIE HORIZONTLINIE"
 1220 GOSUB 1990
 1230 LET Y0=Y
 1240 LET E0=X
 1250 PRINT A$+" WESTEN"
 1260 PRINT "BLICKRICHTUNG SENKRECHT NACH OBEN"
 1270 GOSUB 1990
 1280 LET AA=(Y+Y0)/2
 1290 LET FA=(Y0-Y)/180
 1300 LET FE=(X-E0)/90
 1310 PRINT"BITTE DIE UHRZEIT EINGEBEN"
 1320 INPUT"STUNDE";H$
 1330 LET H$=RIGHT$("00"+H$,2)
 1340 INPUT"MINUTE";M$
 1350 LET M$=RIGHT$("00"+M$,2)
 1360 INPUT"SEKUNDE";S$
 1370 LET S$=RIGHT$("00"+S$,2)
*1380 LET TI$=H$+M$+S$
 1390 LET ZR=0
*1400 IF TI$="000000" THEN LET ZR=1
 1410 FOR I=0 TO 2000
 1420 REM WARTE MINDESTENS EINE SEKUNDE
 1430 NEXT
*1440 IF TI$=H$+M$+S$ THEN LET ZR=1
 1450 REM WENN TI$ UNVERAENDERT, EXISTIERT KEINE UHR
 1460 LET DF=JAHR-1980
 1470 LET SCHALTJAHR=INT(DF/4)
 1480 LET UHR=VAL(H$)
 1490 LET MIN=VAL(M$)
 1500 LET SK=VAL(S$)
 1510 LET TZEIT=UHR+(MIN+SK/60)/60+ZZ
 1520 LET ZEIT=DF*365+SCHALTJAHR+TAG-1+TZEIT/24
 1530 IF DF-4*SCHALTJAHR THEN LET ZEIT=ZEIT-1
 1540 IF (DF<0) AND (DF<>4*SCHALTJAHR) THEN LET ZEIT=ZEIT
 1550 LET THETA=(ZWEIPI*ZEIT/365.25)
 1560 LET G=-0.031271-(4.53963E-7)*ZEIT+THETA
 1570 LET EL=4.889866+(3.67474E-7)*ZEIT
 1580 LET EL=EL+(0.033494-(2.3E-9)*ZEIT)*SIN(G)
 1590 LET EL=EL+(9.49E-4)*SIN(2*G)+THETA
 1600 REM LAENGE DER SONNE
 1610 LET SEL=SIN(EL)
```

```
1615 LET CL=COS(EL)
1620 LET EPS=0.40914-(6.2149E-9)*ZEIT
1625 REM EKLIPTIK CA. 23.5 GRAD
1630 LET CP=COS(EPS)
1631 LET RA=ATN(SEL*CP/CL)
1632 LET RL=0
1634 IF SEL*CP<0THEN LET RL=ZWEIPI/2
1635 IF SEL*CP=CL<0 THEN LET RL=RL+ZWEIPI/4
1636 LET RH=RL+ZWEIPI/4
1638 IF RA<RL THEN LET RA=RA+ZWEIPI/4:GOTO 1638
1640 IF RA>RH THEN LET RA=RA-ZWEIPI/4:GOTO 1640
1660 LET DEKL=FNAS(SEL*SIN(EPS))
1670 REM DEKLINATION
1680 LET SID=1.759335+ZWEIPI*(ZEIT/365.25-DF)
     +(3.694E-7)*ZEIT
1690 IF SID>=ZWEIPI THEN LET SID=SID-ZWEIPI
1700 REM SIDERIALE ZEIT
1710 LET S=SID+(TZEIT*15-LAENGE)*RD
1720 IF S>=ZWEIPI THEN LET S=S-ZWEIPI
1730 REM LOKALE SONNENZEIT
1740 LET H=RA-S
1750 REM STUNDENWINKEL
1760 LET PHI=BREITE*RD
1770 LET E=FNAS(SIN(PHI)*SIN(DEKL)+COS(PHI)
     *COS(DEKL)*COS(H))
1780 LET A=FNAS(COS(DEKL)*SIN(H)/COS(E))/RD
1790 IF SIN(E)=SIN(DEKL)/SIN(PHI) THEN GOTO 1820
1800 IF A<0 THEN A=A+360
1810 LET A=180-A
1820 LET E=E/RD
1830 PRINT H#;":";M#;":";S#
1840 PRINT "A=";A;"GRAD  E=";E;"GRAD"
1850 GOSUB 3000:REM STEUERUNGSROUTINE
1860 LET MIN=MIN+10
1870 IF MIN<60 THEN GOTO 1900
1880 LET MIN=0
1890 LET UHR=UHR+1
1900 IF UHR<24 THEN GOTO 1930
1910 LET UHR=0
1920 LET TAG=TAG+1
1930 LET M#=STR$(MIN)
1940 LET H#=STR$(UHR)
1950 IF ZR=1 THEN GOTO 1510
1960 LET T=100*(MIN+100*UHR)
1970 IF VAL(TI$)<T THEN GOTO 1970
1980 GOTO 1510
1990 REM EICHWERTE UEBERNEHMEN
2000 PRINT"WENN DIE AUSRICHTUNG STIMMT"
2010 PRINT"TASTE E8 BETAETIGEN"
2020 M=M1
2030 IF USR(E5)=1 THEN M=M2
2040 IF USR(E3)=0 THEN GOTO 2070
2050 SYS M,RECHTS :REM AUF/RECHTS
2060 GOTO 2040
2070 SYS M1,AUS
2080 SYS M2,AUS
2090 IF USR(E4)=0 THEN GOTO 2120
2100 SYS M,LINKS :REM AB/LINKS
2110 GOTO 2090
2120 SYS M1,AUS
2130 SYS M2,AUS
2140 IF USR(E6)=0 THEN GOTO 2020
2150 LET X=USR(EK) :REM NEIGUNGSWINKEL
2160 LET Y=USR(EY) :REM AZIMUTH
2170 IF USR(E6)=1 THEN GOTO 2170
2180 RETURN
3000 REM STEUERUNGSROUTINE
3005 IF E<0 THEN RETURN
3010 LET AS=INT(A*FA+A0+0.5)
3020 LET ES=INT(E*FE+E0+0.5)
3030 IF AS<AL THEN LET AS=AL
3040 IF AS>AR THEN LET AS=AR
3050 IF ES<EM THEN LET ES=EM
3060 IF ES>ER THEN LET ES=ER
3070 IF ES=EA THEN GOTO 3160
3080 REM HOEHENSTEUERUNG
3090 LET D=USR(EX)-ES
3100 IF D>0 THEN SYS M2,RECHTS
3120 IF D<0 THEN SYS M2,LINKS
3140 IF D=0 THEN GOTO 3160
3145 SYS M2,AUS
3150 GOTO 3090
3160 LET EA=ES
3170 IF AS=AA THEN GOTO 3270
3180 REM AZIMUTHSTEUERUNG
3190 LET D=USR(EY)-AS
3200 IF D>0 THEN SYS M1,RECHTS
3210 IF D<0 THEN SYS M1,LINKS
3220 IF D=0 THEN GOTO 3270
3230 LET D=ABS(D)
3240 IF D>GA THEN GOTO 3190
3250 SYS M1,AUS
3260 GOTO 3190
3270 LET AA=AS
3280 RETURN
```

Bitte kontrollieren Sie die Vollständigkeit Ihres fischertechnik COMPUTING Systems.

Lieber fischertechnik-Freund, bevor Sie mit den ersten Computer-Experimenten mit fischertechnik COMPUTING beginnen, kontrollieren Sie bitte die Vollständigkeit Ihrer Ausstattung:

Für den ...	benötigen Sie neben dem Bausatz noch ...
fischertechnik COMPUTING Baukasten	• das fischertechnik Interface • den Interface-Adapter (Gutschein) • die Diskette COMPUTING (Gutschein) • das fischertechnik Netzgerät COMPUTING
fischertechnik COMPUTING Trainings-roboter	• das fischertechnik Interface • den Interface-Adapter (Gutschein) • die Diskette Trainingsroboter (Gutschein) • das fischertechnik Netzgerät COMPUTING
fischertechnik COMPUTING Plotter/Scanner	• das fischertechnik Interface • den Interface-Adapter (Gutschein) • die Diskette Plotter/Scanner (Gutschein) • das fischertechnik Netzgerät COMPUTING
fischertechnik COMPUTING EXPERIMENTAL	• den Interface-Adapter (Gutschein) • die Diskette EXPERIMENTAL (Gutschein)

Der Gutschein für den Adapter liegt beim Interface bei. Der Gutschein für die Diskette liegt entweder beim Bausatz oder Interface bei. Senden Sie die Gutscheine an die fischerwerke ein, um die benötigten Teile kostenlos zu erhalten. Bei künftigen Bausätzen fragen Sie Ihren Händler oder die fischerwerke.

fischerwerke · Abt. fischertechnik · D-7244 Tumlingen/Waldachtal

Please check the completeness of your fischertechnik COMPUTING system.

Dear friend of fischertechnik, before starting right with your first computer control experiments using fischertechnik COMPUTING please check the completeness of your equipment:

Using the ...	you will require additionally ...
fischertechnik COMPUTING robotics kit	• the fischertechnik interface • the Interface-adapter (coupon) • the diskette COMPUTING (coupon) • an adequate power supply (see text)
fischertechnik COMPUTING Training Robot	• the fischertechnik interface • the Interface-adapter (coupon) • the diskette Training Robot (coupon) • an adequate power supply (see text)
fischertechnik COMPUTING Plotter/Scanner	• the fischertechnik interface • the Interface-adapter (coupon) • the diskette Plotter/Scanner (coupon) • an adequate power supply (see text)
fischertechnik COMPUTING EXPERIMENTAL	• the Interface-adapter (coupon) • the diskette EXPERIMENTAL (coupon)

The coupon for the adapter is part of the interface package. The coupon for the diskette is either part of the construction kit or the interface package. Mail the coupons to the nearest address as given on the coupon to obtain the required items free of charge. Concerning future kits ask your dealer.

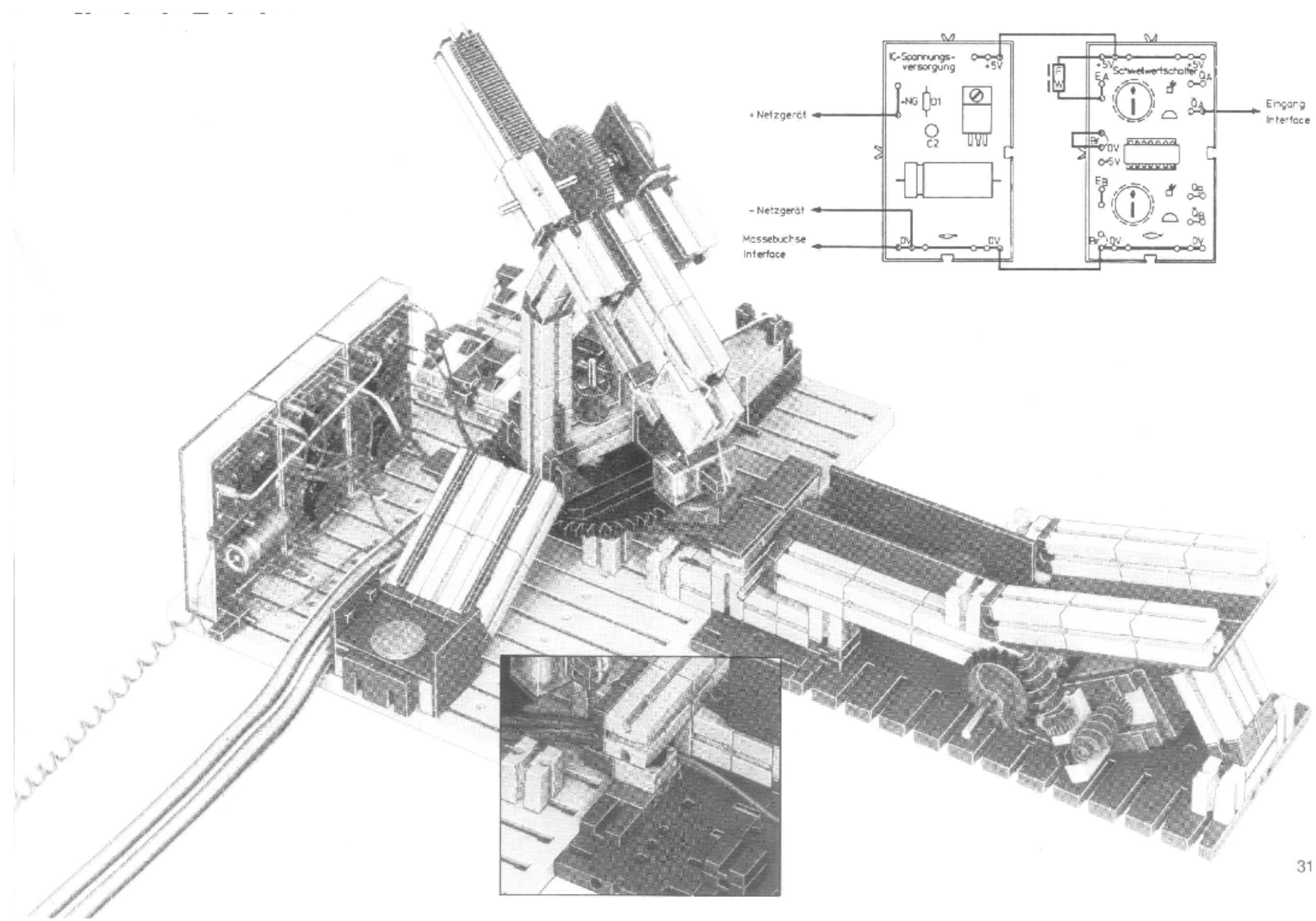
Noch ein Roboter

Erinnern Sie sich noch an den Teach-in Roboter von S. 19 dieser Anleitung? Auf Seite 31 ist er noch einmal dargestellt. Leichte Umbauten hat er sich gefallen lassen müssen. Aus Start 100 und einigen grauen Bausteinen aus dem Service-Set sind ein Förderband und eine Rutsche entstanden. Angetrieben wird das Förderband mit einem fischertechnik-Motor aus dem Baukasten Motor und Getriebe. Damit kann eine Münze, die auf das Förderband rutscht, vor zu dem Roboter transportiert werden. Dort angekommen, würde sie einfach vom Band fallen – hätten wir nicht einen Fotowiderstand eingebaut. Dieser vermeldet die Abdunklung durch die Münze und die Bausteine aus dem Baukasten Elektronik, an denen die Ansprechschwelle eingestellt wird, bereiten das Signal computergerecht auf. Die Schaltung entspricht den Schaltungen 29–3 bzw. 30–7 aus dem Anleitungsbuch Elektronik.

Hierzu wird die Masseschiene der Elektronikbausteine mit der freien Massebuchse des Interface verbunden. Die Ausgänge der Elektronikbausteine können dann mit den Eingängen des Interface verbunden werden. Hier startet die Abdunklung des Fotowiderstands den Bewegungsablauf des Roboters.

Mit diesem Beispiel wollen wir Sie ermuntern, fischertechnik computing mit Bauteilen aus dem übrigen fischertechnik-Programm zu kombinieren. Alles paßt zueinander – die Bausteine ebenso gut wie die elektrischen Signale.

```
*500 SYS INIT
 510 REM
 520 REM FISCHERTECHNIK COMPUTING
 530 REM
 540 REM PICK & PLACE
 550 REM
 560 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
 570 REM
 580 REM BELEGUNG DES INTERFACE
 590 REM
 600 REM EINGABEN
 610 REM E1=FOTOZELLE
 620 REM E2=START
 630 REM EX=ARMPOSITION
 640 REM EY=WINKELPOSITION
 650 REM
 700 REM AUSGABEN
 710 REM M1=DREHEN
 720 REM M2=ARM
 730 REM M3=MAGNET
 740 REM M4=FOERDERBAND
 750 REM
 800 REM FUNKTION
 810 REM WENN STARTTASTE GEDRUECKT IST
 820 REM BEFOERDERT DAS FOERDERBAND DAS
 830 REM GELDSTUECK ZUM ROBOTARM,WELCHER
 840 REM ES ZUR RUTSCHE TRANSPORTIERT.
 900 REM DREHPOSITIONEN (EMPIRISCH ERMITTELT)
 910 DIM DREH(2),ARM(2)
 920 LET DREH(1)=113
 930 LET DREH(2)=35
 950 LET ARM(1)= 120
 960 LET ARM(2)= 30
*1000 PRINT CHR$(147)
 1010 PRINT"FISCHERTECHNIK"
 1020 PRINT"COMPUTING"
 1030 PRINT
 1040 PRINT"PICK & PLACE"
 1050 PRINT
 1060 PRINT"BITTE STARTTASTE BETAETIGEN
 1070 REM START
*1080 IF USR(E2)=0 THEN GOTO 1080
*1090 SYS M4,EIN
*1100 IF USR(E1)=0 THEN GOTO 1100
*1110 SYS M4,AUS
 1120 LET SOLLWERT=DREH(1)
 1130 GOSUB 1300
 1140 LET SOLLWERT=ARM(1)
 1150 GOSUB 1400
 1160 REM MAGNET EIN
 1170 FORI=0TO100
*1180 SYS M3,EIN
 1190 NEXTI
 1200 LET SOLLWERT=ARM(2)
 1210 GOSUB 1400
 1220 LET SOLLWERT=DREH(2)
 1230 GOSUB 1300
 1240 LET SOLLWERT=ARM(1)
 1250 GOSUB 1400
*1260 SYS M3,AUS :REM MAGNET AUS
 1270 LET SOLLWERT=ARM(2)
 1280 GOSUB 1400
 1290 GOTO 1260
 1300 REM LAGEREGELUNGSROUTINE DREHUNG
*1310 LET D=USR(EY)-SOLLWERT
*1320 IF D>0 THEN SYS M1,RECHTS
*1330 IF D<0 THEN SYS M1,LINKS
 1340 IF D=0 THEN RETURN
 1350 LET D=ABS(D)
 1360 IF D>2 THEN GOTO 1310
*1370 SYS M1,AUS
 1380 GOTO 1310
 1390 RETURN
 1400 REM LAGEREGELUNGSROUTINE ARM
*1410 LET D=USR(EX)-SOLLWERT
*1420 IF D>0THEN SYS M2,RECHTS
*1430 IF D<0 THEN SYS M2,LINKS
 1440 IF D=0 THEN RETURN
 1450 LET D=ABS(D)
 1460 IF D>20 THEN GOTO 1400
*1470 SYS M2,AUS
 1480 GOTO 1400
 1490 RETURN
```


IC-Spannungs-
versorgung
+5V
+NG
D1
C2
+ Netzgerät
- Netzgerät
Massebuchse
Interface
0V
0V
F
W
+5V
Schwellwertschalter
+5V
EA
QA
Q̄A
Eingang
Interface
Br
0V
+5V
EB
QB
Q̄B
Br
0V
0V

39497 · 0490 WA · Printed in Germany · Technische Änderungen vorbehalten