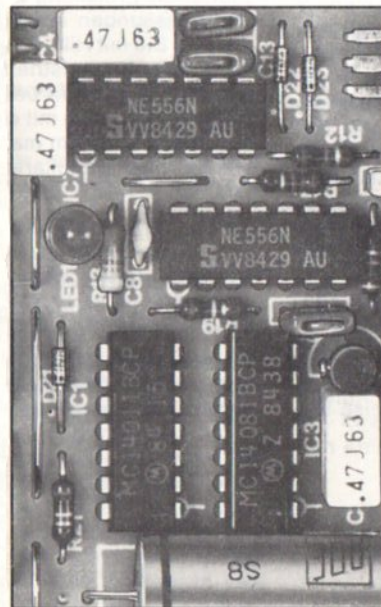
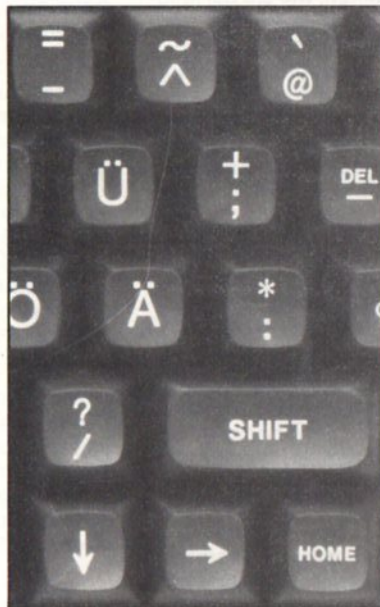
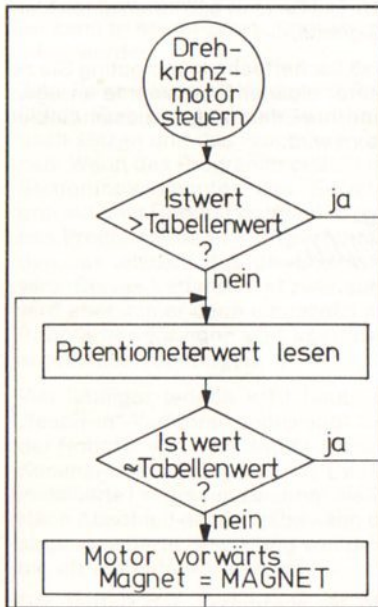




COMPUTING
COMPUTING
COMPUTING
COMPUTING



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 Ein-

führung 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 Grafiktablets beschreiten. Der Computer registriert Ihre Bewegungen des Grifels 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.

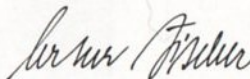
Ebenfalls ein Modell aus der Technik ist die Sortieranlage. Sie unterscheidet Bausteine 30 und 15 und sortiert sie in zwei verschiedene Auffangbehälter.

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



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 Namensverlängerung, 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. Außerlich 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)
ACORN	: CLS
Sinclair	: CLS
Apple II	: HOME

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.)

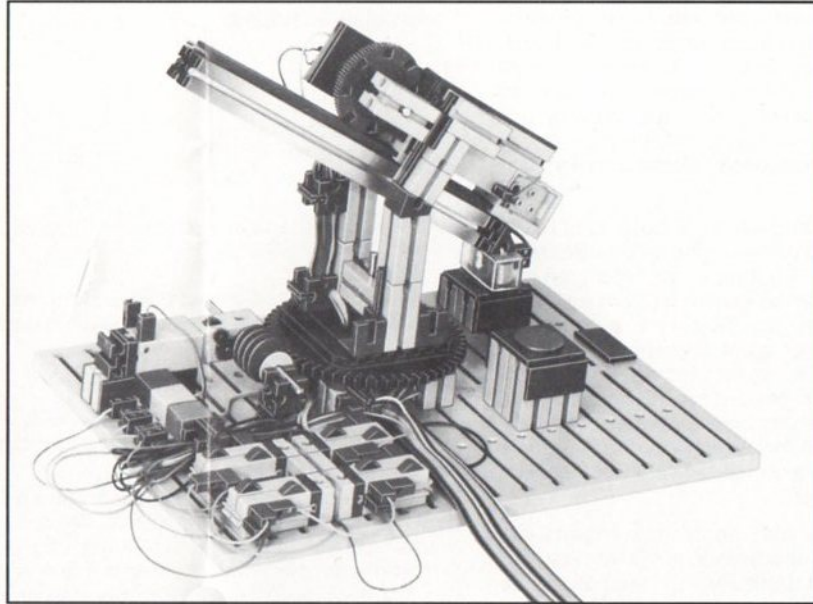
für Eingang 1 aus. Je nachdem, ob der Taster zwischen E1 und +5V bei der Betätigung der RETURN- bzw. ENTER-Taste gedrückt ist oder nicht, wird der Computer eine 1 oder 0 ausgeben.

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.

Als Beispiel für eine konkrete Programmierung wird der „Teach-in Roboter“ vorgestellt:



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 unzutraglichen 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	(E 1)
Arm ab	(E 2)
Drehe links	(E 3)
Drehe rechts	(E 4)
Magnet ein	(E 5)
Magnet aus	(E 6)

Zwei Tasten haben eine Sonderfunktion. Die Taste

Lerne	(E 7)
-------	-------

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

Start/Stop	(E 8)
------------	-------

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 Lern-taste 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 Drücken 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 E9=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-EG 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 GOSUB2900
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 WART E 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(E8)=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)
2260 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$(ID)="EIN" THEN SYS M3,EIN
2620 FOR I=0 TO 100
*2630 IF USR(E8)=1 THEN 4000
2640 NEXT
2650 IF MAGNET$(ID)="AUS" THEN GOSUB 2800
2700 GOTO 2100
2800 REM MAGNET AUS ROUTINE
*2810 SYS M3,LINKS
2820 REM MAGNET UMGEPOLT ZUR ENTMAGNETISIERUNG
*2830 SYS M3,AUS :REM MAGNET AUSGESCHALTET
2840 RETURN
2900 REM TITELMELDUNG
*2910 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

```