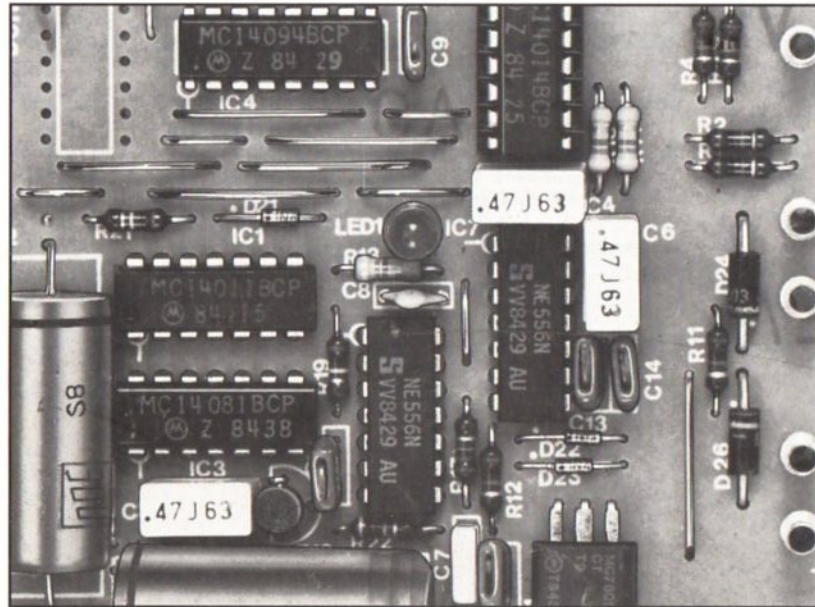
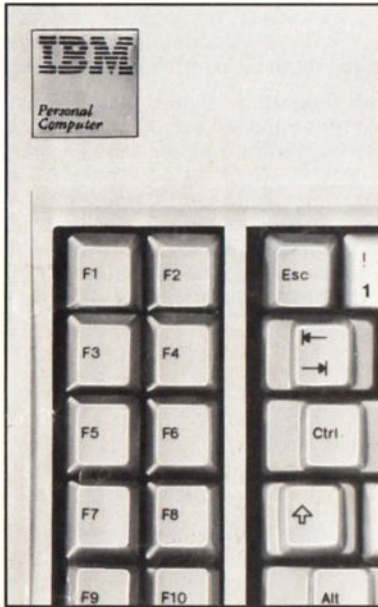


# fischertechnik<sup>®</sup> <sup>®</sup> COMPUTING

## Interface IBM Personal Computer



## Inhalt

Einführung	3
Anschluß des Interface	4
fischertechnik computing Software	5
Diagnoseprogramm	8
Microsoft BASIC	10
Checkliste	10
Technische Daten	10
Benutzung von fischertechnik Elektromechanik und Elektronik	11
Funktionsweise des Interface und des Interface-Treiberprogramms	12
Verdrahtungsplan	27

## Table of Contents

Introduction	15
Connecting the Interface	16
fischertechnik computing Software	17
The Diagnostic Program	20
Microsoft BASIC	22
Check List	22
Technical Data	22
Using the fischertechnik Electro- mechanics and Electronics Kits	23
Operation of the Interface and the Interface Driver Program	24
Wiring diagram of the interface inputs and outputs	27

## fischertechnik computing Interface

Lieber fischertechnik-Freund,

um mit einem Computer, in Erweiterung seiner Einsatzmöglichkeiten, auch technische Modelle ansteuern zu können, wurde fischertechnik computing entwickelt. Hierzu gehören sowohl der fischertechnik computing Baukasten und die fischertechnik computing Bausätze ebenso wie die fischertechnik computing Interfaces und die Software. Es ist jetzt möglich, technische Funktionen und Vorgänge zu simulieren, Aufgaben zu lösen und einfach viel Spaß an computergesteuerten Modellen zu haben.

Was braucht man zum Steuern der Modelle? Zunächst einmal das fischertechnik Modell zur Ausführung der Abläufe. Dann einen Personalcomputer, wie Sie ihn besitzen. Er dient der Steuerung und der Koordination. Und dann noch ein Interface als Bindeglied zwischen beiden.

Was Sie in den Händen halten, ist das fischertechnik computing Interface. Steuersignale, die von dem Computer kommen, z. B. „Motor einschalten!“, wer-

den von dem Interface in kräftige Ströme umgesetzt, die in der Lage sind, tatsächlich einen Motor zu bewegen. Wir sprechen in diesem Fall von einer Ausgabe. Die gedachte Blickrichtung verläuft von dem Computer nach außen. Aber auch der umgekehrte Weg ist denkbar und kommt vor. Die Modelle besitzen Taster, Potentiometer etc., um dem Computer Bericht zu erstatten, was an dem Modell draußen vorgeht. Auch hier greift das Interface wieder helfend ein und bereitet diese Signale dergestalt auf, daß sie eine für den Computer verständliche Eingabe darstellen.

Das fischertechnik Interface besitzt nun folgende Leistungsmerkmale:

- Mit ihm lassen sich vier fischertechnik Motoren, Lampen, Elektromagnete etc. steuern.
- Mit ihm kann man acht Taster oder Schalter abfragen.
- Darüber hinaus liefern zwei Eingänge die Werte von stufenlosen Signalgebern wie etwa Potentiometern.

Doch was würden alle elektrischen Verbindungen zwischen Computer und fischertechnik Modell mit Hilfe des Interface nutzen, wenn Sie keine Hilfsmittel hätten, jene zu aktivieren. Die Rede ist von der Software. Dieser Teil liegt in der Form einer Diskette vor. Auf ihr befindet sich ein Programm, das den Sprachschatz Ihres Computers derart erweitert, daß die Steuerung über das Interface tatsächlich erfolgen kann. Dieses Programm wird die Keimzelle Ihrer eigenen Programme sein. Doch damit nicht genug: Damit Sie den Einsatz dieser neuen Hilfsmittel studieren und lernen können, sind Beispielprogramme für alle fischertechnik computing Modelle auch noch untergebracht.

Sie sehen, es wartet eine ganze Menge von interessanten Aufgaben auf Sie. Ich wünsche Ihnen viel Spaß dabei. Ihr



## Anschluß des Interface

Das fischertechnik Interface IBM Personal Computer paßt sowohl an den IBM-PC, IBM-PPC, IBM-XT als auch an den IBM-AT. Diese Computer haben außerdem einen Standard gebildet, den Standard der IBM-Kompatibilität. Es sind daher eine Reihe weiterer Computer verfügbar, die diesen Computern in mancher Hinsicht ähneln. Kompatible Computer müssen und dürfen auch nicht hundertprozentig identisch mit den IBM-Modellen sein. Daher kann letztlich nicht für jeden Kompatiblen mit Gewißheit gesagt werden, daß der Betrieb von fischertechnik computing unverändert möglich ist. Im Einzelfall werden Sie dies prüfen müssen. Vielleicht gibt Ihnen Ihr Händler Gelegenheit dazu.

Wir werden der Vielzahl der Typen und Bezeichnungen wegen im folgenden Text den IBM oder kompatiblen Personal Computer kurz und bündig mit PC bezeichnen.

Der Anschluß des fischertechnik computing Interface erfolgt an die Parallel-Druckerschnittstelle des PC. An alle diese Computertypen paßt das Interface gleichermaßen, da sie sich in Bezug auf den Anschluß nicht unterscheiden.

Allen diesen Computern ist auch gemeinsam, daß sie zumindest das Betriebssystem MS-DOS bzw. PC-DOS ab Version 2.0 verstehen. Das fischertechnik Interface wird normalerweise unter dem Disketten-BASIC betrieben, das meist unter dem Namen BASIC aufgerufen wird. Sofern die Programme Bildschirmgrafik verwenden, ist einerseits eine Grafikkarte, andererseits das besser ausgebaute BASICA oder GWBASIC notwendig.

Doch zurück zur Hardware. Das fischertechnik Interface wird, wie bereits oben erwähnt, an den Druckeranschluss des PC angeschlossen. Hierzu gehen Sie folgendermaßen vor:

- Vergewissern Sie sich, daß der Computer abgeschaltet ist.
- Drehen Sie den Computer herum, so daß die rückseitigen Anschlußstecker vor Ihnen liegen.

Suchen Sie den Parallel-Druckeranschluß. Es muß ein 25poliger Buchsenstecker sein. Sie werden den Parallel-Druckeranschluß mit anderen 25poligen Schnittstellen, z.B. der Seriell-Schnittstelle, nicht verwechseln können, da diese Stifte und keine Buchsen besitzen.

- Durch Reibungselektrizität können Sie, ohne es zu merken und ohne daß es für Sie schädlich wäre, auf mehrere 1000 Volt aufgeladen sein. Diese Spannung ist jedoch schädlich für die Schaltkreise in dem Interface und dem Computer. Entladen Sie daher eine eventuell vorhandene elektrostatische Aufladung durch Berühren eines geerdeten Gegenstandes oder des Metallgehäuses des Computers.
- Legen Sie sich das Interface zurecht. An dem Interface ist ein mehradriges Flachbandkabel befestigt. An dem Flachbandkabel befindet sich wiederum ein zwanzigpoliger Stecker, den Sie mit dem Adapter verbinden. Der Adapter trägt nun den passenden Stecker zu dem Druckeranschluß Ihres PC. Der Stecker wird auf den Parallel-Druckeranschluß aufgesteckt; ein Verpolen ist wegen der Trapezform des Steckers nicht möglich.
- Schließen Sie das Interface an das Netzgerät an. Das Interface erwartet ungesieberte Gleichspannung zwischen 6 und 10 Volt. Beim Einsatz von stabilisierter Gleichspannung sind ca. 8 Volt (belastbar bis ca. 2 Ampere) einzustellen. Um ganz sicher zu gehen, sollten Sie jedoch das fischertechnik computing Netzgerät verwenden. Stecken Sie den Kabelausgang des Netzgeräts mit dem roten Flachstecker in die mit + gekennzeichnete Buchse des Interface und den Kabelausgang mit grünem Flachstecker in die mit - gekennzeichnete Buchse. Sie haben dabei die Auswahl zwischen zwei Buchsenpaaren; welches Sie verwenden, ist gleichgültig.

- Für Modelle mit bis zu zwei Motoren können auch die fischertechnik Netzgeräte mot4 eingesetzt werden, also insbesondere für die Modelle des fischertechnik computing Baukastens. Sollten Sie aber eigene Konstruktionen mit einer größeren Motorzahl oder die Bausätze Trainingsroboter oder Plotter/Scanner betreiben wollen, brauchen Sie Verstärkung. Entweder wechseln Sie dann doch zu dem kräftigeren fischertechnik computing Netzteil oder Sie speisen mit einem zweiten fischertechnik Netzgerät mot4 in die noch freien Anschlußbuchsen ein.
- Bei Einsatz der Netzgeräte mot4 verwenden Sie die den Baukästen beigelegte zweiadrige Leitung und Flachstecker zur Herstellung des Anschlußkabels.
- Verbinden Sie das fischertechnik computing Modell mit dem Interface. Hierzu dient das den Bausätzen und dem Baukasten fischertechnik computing beigefügte zwanzigadrige Flachbandkabel. Dieses Kabel ist auch als Einzelteil aus dem Service-Set fischertechnik erhältlich.
- Die Reihenfolge, in der Sie das Interface und den Computer nun einschalten, spielt keine Rolle. Wenn Sie das Interface mal nicht benutzen und mit anderen Programmen arbeiten, brauchen Sie das Interface dennoch nicht abkabeln. Lassen Sie in diesem Fall einfach das Interface ausgeschaltet.
- Nebeneffekte des Interface: Solange das Interface eingesteckt ist, können Sie selbstverständlich keinen Drucker benutzen, der auch an diesem Anschluß eingesteckt wird. Allerdings sieht das Betriebssystem des PC bis zu vier Druckeranschlüsse vor, so daß Sie gegebenenfalls ausweichen können.

Auch beim Arbeiten mit den Modellen sollten Sie sicherheitshalber immer zuerst eine eventuell vorhandene elektrostatische Aufladung ableiten, indem Sie einen geerdeten metallischen Gegen-

## fischertechnik computing Software

stand berühren, z.B. das Metallgehäuse des PC. Doch nun genug von der Hardware, im nächsten Abschnitt wollen wir uns die fischertechnik computing Software vornehmen.

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 der Ein- und Ausgabebausteine sowie der Maschinensprache für diese Aufgabe. Bislang stimmte diese Aussage, und dadurch wurde leider auch mancher von diesem interessanten Kapitel der Computerei abgehalten. Jetzt gibt es diese Schwierigkeit nicht mehr. In dem Lieferumfang des Interface sind die vorliegende Anleitung und Programme auf Diskette enthalten.

Um sicher zu sein, daß sie nicht bei einem verheerenden Löschen oder einer Beschädigung der Diskette ohne Software dastehen, sollten Sie sich von der fischertechnik computing Diskette eine Kopie anfertigen. Benutzen Sie hierfür das auf der MS-DOS Systemdiskette mitgelieferte Programm DISKCOPY. Verfahren Sie exakt nach den Angaben der MS-DOS Anleitung und des Programms. Wenn Sie die Kopie vorliegen haben, verstauen Sie die fischertechnik computing Diskette an einem sicheren Platz, an den keine natürlichen Feinde der Disketten, wie Sand, Hitze, Katzen oder Magnetfelder hinkommen können. Arbeiten Sie fortan mit der Kopie und benutzen Sie die fischertechnik computing Diskette nur, um gegebenenfalls eine weitere Kopie zu ziehen. Kopien sind vor allen Dingen dann nützlich, wenn Sie die Programme noch abändern wollen und die Originaldiskette bestehen bleiben soll. Das Kopieren beschränkt sich jedoch wohlge-merkt nur auf den eigenen Gebrauch.

Einige fischertechnik Programme legen Daten auf Diskette ab. Wenn sie schon dabei sind, sollten Sie sich einige Disketten formatieren.

Wenn Sie das Interface noch nicht an Ihren Computer wie in dem vorigen Kapitel beschrieben angeschlossen haben, so sollten Sie es nun nachholen.

Schalten Sie den Computer ein. Starten Sie BASIC durch Eingabe von

### **BASIC**

Legen Sie nach dem Laden von BASIC die fischertechnik computing Diskette in das Diskettenlaufwerk A: ein. Wählen Sie das Unterverzeichnis mit den deutschsprachigen Programmen an:

### **CHDIR"A:DEUTSCH"**

Laden Sie das Programm FISCHER mit dem Kommando

### **RUN"A:FISCHER"**

In wenigen Sekunden erscheint eine Bildschirmmeldung, in der Sie gefragt werden, ob Sie die Software an Ihren PC anpassen wollen. Falls Sie das erstmal die Diskette benutzen und Ihr Computer kein IBM-PC ist, beantworten Sie die Frage mit

### **J**

für „ja“. Das Programm wird Sie anschließend nach der Arbeitsgeschwindigkeit Ihres Computers sowie nach der Nummer der verwendeten Druckerkarte fragen. Zum Abschluß wird das Programm INTERFAC.COM an die Arbeitsbedingungen Ihres PC angepaßt. Diesen Anpaßdurchlauf müssen Sie nicht jedesmal durchführen. Da die Anpassung auf die Diskette geschrieben wird, liegt bei jedem folgenden Start gleich die passende Software vor. Sollten Sie Ihre fischertechnik computing Diskette jedoch einmal auf einem anderen PC betreiben, werden Sie ggf. die Anpaßprozedur wieder durchlaufen müssen. Zum Abschluß druckt das Programm das Inhaltsverzeichnis der Diskette.

Laden Sie nun den Computer mit dem Grundprogramm GRUNDPR.BAS. Also:

### **RUN"A:GRUNDPR"**

Nach dem Laden des Programms wird das Disket-

tenlaufwerk nochmals kurz anlaufen, denn das Grundprogramm lädt das Maschinenprogramm INTERFAC.COM. Jenes enthält die oben erwähnten Detailkenntnisse über die Ein- und Ausgabebausteine des Computers. Anschließend werden Sie nach der Einstellung der Bildschirmfarben gefragt. Sie können an dieser Stelle mit Hilfe der Funktionstasten die Bildschirmfarben optimal auf Ihr Bildschirmgerät einstellen. Danach wird sich der PC wieder mit seinem Bereitzeichen melden.

Äußerlich scheint sich nichts geändert zu haben. Dennoch besitzt Ihr PC nun einige neue Befehle, die vorher in dieser Form nicht im BASIC enthalten waren. Diese Befehle sind genau auf Ihren PC und das fischertechnik computing Interface abgestimmt. Die Befehle benutzen das CALL-Kommando und die USR-Funktion. Das Grundprogramm selbst ist in BASIC geschrieben und erzeugt im Variablenspeicher des PC die Parameter jener Maschinenprogrammaufrufe.

Sie brauchen daher nur noch die folgend beschriebenen Befehle zu beherrschen:

Der Motorausgang M1 wird angesteuert mit:

**CALL M1(EIN)      CALL M1(AUS)**  
**CALL M1(RECHTS)    CALL M1(LINKS)**

Die Kommandoparameter bezeichnen den Motor, in diesem Fall Motor 1, und die Betriebsart.

Die entsprechenden Befehle mit M2, M3 und M4 steuern die übrigen drei Ausgänge. Außerdem sollten Sie sich merken, daß „EIN“ ebenfalls immer Rechtslauf bewirkt.

Die 10 Eingänge des Interface werden mit Hilfe der USR-Funktion eingelesen. Die Funktion

#### **USR(E1)**

ist 1, wenn der Eingang E1 des Interface mit +5 V verbunden ist. Sonst zeigt USR(E1) den Wert 0. Entspre-

chend erhält man mit USR(E2)...USR(E8) die Zustände der übrigen Digitaleingänge.

Die Analogeingänge EX und EY werden über je ein Potentiometer (4,7 kΩ) mit +5V verbunden. Die Funktionen

#### **USR(EX) USR(EY)**

führen dann einen Wert aus dem Bereich 20 bis 230, je nach Stellung der Potentiometer. Obige Grenzwerte können auch leicht anders ausfallen. Sie hängen von der Arbeitsgeschwindigkeit des Mikroprozessors im PC ab. Sollten Sie einen gänzlich anderen Wertebereich vorfinden, so müssen Sie das Programm FISCHER laufen lassen, um die Datei INTERFAC.COM anzupassen.

Die Analogeingänge dienen häufig der Positionsmessung. 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.

Wenn kein Potentiometer an den Eingang EX bzw. EY angeschlossen ist, wird sich ein Überlauf des internen Auswertezählers ergeben. Der Funktionswert beträgt dann entweder 255, 511 oder 1023.

Der letzte der neuen Befehle ist

#### **CALL 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.

Doch nun genug der langen Vorrede. Schließen Sie einen fischertechnik Motor über das zwanzigpolige farbcodierte Flachbandkabel an M1 an. Dies sind die

gelbe und orange Leitung in der oberen Hälfte des Flachbandkabels. Geben Sie ein:

#### **CALL M1(EIN)**

Der Motor wird kurz anlaufen und dann wieder stehenbleiben. Genießen Sie diesen Augenblick, er hat Ihnen das Gefühl gegeben, in kurzer Zeit die kompliziertesten fischertechnik Anlagen mit Ihrem PC zu steuern.

Doch zunächst interessiert uns auch die Frage, wieso der Motor wieder stehenblieb. Hatten wir ihn nicht eingeschaltet? Gibt es zum Ausschalten nicht, wie oben beschrieben, einen eigenen Befehl? Nun, der Motor ist zwar stehengeblieben, aber in dem Interface ist nach wie vor gespeichert, daß er eigentlich laufen sollte. Das Interface hat sich selbst „schlafen gelegt“, Dies tut es immer, wenn innerhalb einer halben Sekunde kein neuer Befehl kommt. Es geschieht aus Sicherheitsgründen. Stellen Sie sich vor, Sie erproben ein neues Programm. Die Wahrscheinlichkeit, daß noch irgendwo ein Fehler im Programm versteckt ist, grenzt an Gewißheit. Der Computer bleibt mit einer leidigen Meldung wie

#### **Syntax error in . . .**

stehen. Der Motor, der kurz vorher eingeschaltet wurde, bliebe jedoch nicht stehen und schicke sich an, das schöne Modell zu demolieren. Sie müßten zum Netzgerät hasten und schnell die Spannung abstellen.

Wie beruhigend ist es da zu wissen, daß der Motor von alleine stehenbleiben wird. Auch dann, wenn Sie mit dem Tastendruck CTRL-BREAK den Programmablauf unterbrechen. Und wenn es wieder weitergeht, so wird mit dem ersten Befehl das Interface wieder „aufgeweckt“ und hat keinen der Motoren vergessen. Der Ablauf kann fortgesetzt werden, als sei nichts geschehen.

Daß das Interface mit dem Abschalten nicht sofort zur Hand ist, wurde mit Bedacht gewählt. Zwischen

den Ein- und Ausgabebefehlen an das Interface werden sich immer wieder Pausen aufgrund von Berechnungen ergeben, die es zu überbrücken gilt. Ob das Interface durch Ein- oder Ausgabebefehle aktiviert wird, können Sie auch durch einen Blick auf die Leuchtdiode des Interface sagen. Sie dient nicht nur der Spannungsanzeige, sondern auch der Betriebsanzeige.

Nun wollen wir noch einen kurzen Blick auf die Eingabebefehle werfen. Schließen Sie zwischen E1 (der braunen Leitung am unteren Rand) und +5V (der roten Leitung in der Mitte des Flachbandkabels) einen Taster an.

Probieren Sie aus:

#### **PRINT USR(E1)**

Je nachdem, ob der Taster zwischen E1 und +5V bei der Betätigung der Return-Taste des Computers gedrückt war oder nicht, wird auf dem Bildschirm eine 1 oder eine 0 ausgegeben.

Wenn an dem Ausgang noch von vorhin der Motor angeschlossen war, wird er sich wieder rühren. Auch Eingabebefehle aktivieren wieder die Ausgänge des Interface! Nun schließen Sie bitte ein Potentiometer 4,7 kΩ zwischen EX und +5V an. Stellen Sie den Schleifer in eine mittlere Stellung und geben Sie ein

#### **PRINT USR(EX)**

Die Zahl, die jetzt auf dem Bildschirm erschienen ist, muß zwischen 0 und 255 liegen.

Das Grundprogramm, das Sie eben benutzt haben, werden Sie immer wieder brauchen. Jedes Programm, das mit dem fischertechnik computing Interface Modelle steuern soll, beginnt mit diesem Vorspann, der die neuen Befehle installiert. Bei den Beispielprogrammen auf der Diskette ist das Grundprogramm jeweils schon enthalten. Bei Programmen, die Sie selbst schreiben, beginnen Sie mit dem Grundprogramm und schreiben bei den anschließenden Zeilennummern weiter.

Damit Sie das Potentiometer leichter beobachten können, wollen wir nun auf diese Weise das erste fischertechnik computing Programm schreiben. Das Grundprogramm befindet sich in dem Computer und belegt die Zeilennummer 1 bis 500. Geben Sie daher ein:

```
510 PRINT USR(EX)  
520 GOTO 510  
RUN
```

Es dauert einen ggf. kurzen Moment, bis das Grundprogramm mit der Datei INTERFAC.COM geladen ist. Es erfolgt wieder die Frage nach der Einstellung der Bildschirmfarben. Und dann geht es los. Im Nu wird der Bildschirm mit Zahlen gefüllt, die ständig nach oben hinausgeschoben werden. Wenn Sie jetzt das Potentiometer in die Hand nehmen und den Schleifer drehen, werden Sie die Veränderung der Zahlen beobachten. Drehen Sie von einem Anschlag zum andern. Die eingelesenen Zahlen sollten zwischen 0 und 255 liegen. Die 0 und die 255 werden jedoch nicht ganz erreicht, ein Wertebereich von 20 bis 230 dürfte sich in der Praxis einstellen. Zum Beenden des Programms müssen Sie CTRL-BREAK drücken.

Für diejenigen, die etwas genauer die Abläufe verstehen wollen und nicht nur die auf der Diskette vorliegenden Programme benutzen wollen, halten wir hier nun noch Detailinformationen bereit. Die Funktion des Grundprogramms besteht darin, in einen Speicherbereich des PC ein kurzes Maschinenprogramm einzuschreiben. Dieses liegt als Datei INTERFAC.COM auf der Diskette vor. Das Maschinenprogramm belegt den Speicherbereich von FF00 bis FFFF (Angaben in hexadezimal). Der für BASIC verfügbare Speicherbereich wird somit um 256 Bytes gekürzt, was aber in aller Regel keine spürbare Beschränkung bedeutet.

Neben dem Maschinenprogramm selbst werden auch noch die genormten Parameter M1, M2, M3, M4, E1, E2, E3, E4, E5, E6, E7, E8, EX, EY, EIN,

RECHTS, LINKS und AUS gesetzt. Außerdem sind die Variablen, die mit den Buchstaben A, E, L oder R anfangen, als Ganzzahlvariablen (INTEGER) definiert.

Für selbstgeschriebene BASIC-Programme sind daher unbedingt folgende Einschränkungen zu beachten:

Ähnlich wie Sie auch nicht die reservierten BASIC-Schlüsselwörter wie PRINT oder STOP als Variablen verwenden dürfen, ist auch der Gebrauch der obigen Parameter als Variable verboten. Die Einschränkung betrifft jedoch nicht Variablen anderen Typs (Strings, Real), so daß

**AUSS\$, M1!, E1! . . .**

durchaus vorkommen dürfen.

Die USR-Funktion 0 ist durch das Grundprogramm belegt. Allerdings können Sie noch die weiteren USR-Funktionen verwenden.

Der obengenannte Speicherbereich an FF00 bis FFFF kann nicht mehr für andere Zwecke verwendet werden.

Wer noch weiterforschen will, weil er z.B. die elektronischen Abläufe im Interface verstehen will oder eine Adaption an andere Programmiersprachen vornehmen will, wird auf die Beschreibung des Interface und des Quelltextes verwiesen.

Die letzte Funktion des Grundprogramms besteht in dem Einschalten des Interface und dem Ausschalten aller angeschlossenen Verbraucher. Dies wird durch den Befehl

**500 CALL INIT**

bewirkt. Damit wird das Interface sozusagen in betriebsbereitem Zustand an ein hier anschließendes Benutzerprogramm übergeben. Auch alle Beispielprogramme der Diskette sind nach diesem Muster aufgebaut. Studieren Sie diese, wenn Sie sich Anregungen holen wollen.

```

10 REM Interface Treiberprogramm für
20 REM IBM Personal Computer
30 REM Copyright (C) fischerwerke 1986
40 REM Version 2.1
50 CLEAR ,&HFF00
60 DEF SEG
70 DEF USR=&HFF50
80 BLOAD "A:INTERFAC.COM",&HFF00
90 DEFINT E,R,L,A
100 LET S=7 : H=0 : R=0
110 SCREEN 0,1 : KEY OFF : WIDTH 40
120 COLOR S,H,R : CLS
130 LOCATE 2,1,0
140 PRINT " fischertechnik"
150 PRINT " computing"
160 PRINT
170 PRINT " Bildschirmfarben einstellen? (j/n)"
180 J$=INKEY$
190 IF J$="n" OR J$="N" THEN GOTO 420
200 IF J$("<"j" AND J$("<"J" THEN GOTO 180
210 KEY 1,"r" : KEY 3,"h" : KEY 5,"s" : KEY 7,"p"
220 COLOR S,H,R
230 CLS
240 LOCATE 2,1
250 PRINT " fischertechnik"
260 PRINT " computing"
270 PRINT
280 PRINT " F1 - Rahmen"
290 PRINT " F3 - Hintergrund"
300 PRINT " F5 - Schrift"
310 PRINT " F7 - Programmstart"
320 F$=INKEY$
330 IF F$="" THEN GOTO 320
340 IF F$="p" THEN GOTO 400
350 IF F$="r" THEN LET R=R+1 AND 15
360 IF F$="h" THEN LET H=H+1 AND 7
370 IF F$="s" THEN LET S=S+1 AND 15
380 GOTO 220
390 REM
400 F$=INKEY$:KEY 1,"LIST ":KEY 3,"LOAD"+CHR$(34)
410 KEY 5,"CONT"+CHR$(13):KEY 7,"TRON"+CHR$(13)
420 INIT=&HFF00:M1=INIT+6:M2=M1+4:M3=M2+4:M4=M3+4
430 E1=1:E2=2:E3=4:E4=8:E5=16
440 E6=32:E7=64:E8=128
450 EX=160:EY=144
460 EIN=85:RECHTS=85:LINKS=170:AUS=255
500 CALL INIT

```

## Das Diagnoseprogramm

Wenn Sie ein fischertechnik computing Modell aufgebaut haben, werden Sie vielleicht die Erfahrung machen, daß es nicht so läuft, wie Sie sich das vorgestellt haben. Wen wundert das bei dieser großen Zahl von Leitungen, die zwischen Modell und Interface hin- und herlaufen. Und wenn nur ein Taster vertauscht wäre, die verblüffendsten Effekte könnte dies zur Folge haben. Doppelt schwierig wird die Situation, wenn die Programme selbst geschrieben sind. Wo soll man da mit der Suche anfangen? In der Hardware oder der Software?

Damit Sie die Hardware eindeutig und komfortabel testen können, wurde das Diagnoseprogramm entwickelt. Es liegt auf der fischertechnik computing Diskette als DIAGNOSE.BAS vor. Laden Sie dieses Programm immer zum Austesten eines Modells. Sie können mit ihm sämtliche Eingänge beobachten und feststellen, ob ihr Verhalten mit Ihren Vorstellungen übereinstimmt.

Mit den Zahlentasten suchen Sie einen Steuerausgang aus. Er wird auf dem Bildschirm invers angezeigt. Diesen angewählten Ausgang können Sie nun mit den Funktionstasten einschalten (Rechts- und Linkslauf). Damit stellen Sie also nicht nur fest, ob ein Motor überhaupt läuft, sondern auch, ob er in der gewünschten Drehrichtung anläuft. Sollte dies nicht der Fall sein, so vertauschen Sie bitte die beiden Motoranschlüsse.

Betätigen Sie die Leertaste, um den angewählten Motor auszuschalten. Alle weiteren Tasten schalten sämtliche Motoren aus. Benutzen Sie die Taste X, um das Programm zu beenden.



```

500 CALL INIT
510 REM
520 REM fischertechnik computing
530 REM
540 REM Diagnose
550 REM
560 REM Copyright (C) fischerwerke 1986
570 REM
580 REM Funktion
590 REM Mit diesem Programm werden sämtliche
600 REM Funktionen der Modelle überprüft.
610 REM Alle Eingänge werden angezeigt.
620 REM Alle Ausgänge werden über die
630 REM Computertastatur angesteuert.
1000 REM
1010 REM *****
1020 REM *** Steuerung vorbereiten ***
1030 REM *****
1040 REM
1050 DIM STA(4),STA$(2): REM Status der 4 Motoren
1060 DIM M(4): REM Adressen für Motor 1-4
1070 DEFINT Z
1080 GOSUB 4000: REM Bildaufbau
1090 REM Alle Motoren aus
1100 FOR I=1 TO 4
1110 LET STA(I)=AUS
1120 NEXT I
1130 REM Steuerzeichen
1140 LET SP$(32)=CHR$(32)
1150 KEY 1,CHR$(224): LET F1$=CHR$(224)
1160 KEY 3,CHR$(225): LET F3$=CHR$(225)
1170 KEY 5,CHR$(226): LET F5$=CHR$(226)
1180 KEY 7,CHR$(227): LET F7$=CHR$(227)
1190 REM Texte für Statusanzeige
1200 STA$(0)="re."
1210 STA$(1)="11."
1220 STA$(2)="aus"
1230 LET M(1)=M1:M(2)=M2:M(3)=M3:M(4)=M4
1240 LET MOT=1: REM"Start mit Motor 1
2000 REM
2010 REM *****
2020 REM *** Anzeige der Eingabe ***
2030 REM *****
2040 REM
2050 GOSUB 5050 :REM Motoranzeige löschen
2060 LOCATE 20,2
2070 PRINT USR(E1);
2080 PRINT TAB(5);USR(E2);
2090 PRINT TAB(8);USR(E3);
2100 PRINT TAB(11);USR(E4);
2110 PRINT TAB(14);USR(E5);
2120 PRINT TAB(17);USR(E6);
2130 PRINT TAB(20);USR(E7);
2140 PRINT TAB(23);USR(E8);
2150 PRINT TAB(26);USING"####";USR(EX);
2160 PRINT TAB(31);USING"####";USR(EY)
3000 REM
3010 REM *****
3020 REM *** Steuerung der Ausgabe ***
3030 REM *****
3040 REM
3050 K$=INKEY$
3060 LET F=0 :REM Flagge für gültiges Kommando
3070 M=VAL(K$) :REM Zahlentaste
3080 IF M>1 AND M<=4 THEN LET MOT=M:LET F=1
      GOSUB 5050
3090 LET ZU=STA(MOT)
3100 IF K$=SP$ THEN LET ZU=AUS: STA(MOT)=AUS
      LET F=1
3110 IF K$=F1$ THEN LET ZU=LINKS:LET F=1
3120 IF K$=F3$ THEN LET ZU=LINKS:STA(MOT)=LINKS
      LET F=1
3130 IF K$=F5$ THEN LET ZU=RECHTS:STA(MOT)=RECHTS
      LET F=1
3140 IF K$=F7$ THEN LET ZU=RECHTS:LET F=1
3150 IF K$<>"x" THEN GOTO 3230
3160 CALL INIT
3170 WIDTH 80
3180 KEY 1,"LIST"
3190 KEY 3,"LOAD"+CHR$(34)
3200 KEY 5,"CONT"+CHR$(13)
3210 KEY 7,"TRON"+CHR$(13)
3220 END
3230 IF K$="" OR F=1 THEN GOTO 3320
3240 REM Alle anderen Tasten -> Not-Aus
3250 CALL INIT
3260 FOR I=1 TO 4
3270 LET STA(I)=AUS
3280 NEXT I
3290 LET ZU=AUS
3300 GOSUB 5050
3310 REM Ausgabe an gewählten Motor
3320 M=M(MOT)
3330 CALL M(ZU)
3340 LOCATE 22,(5*MOT-2):COLOR H,S:PRINT"M";MOT
3350 S$=STA$(INT(ZU/100))
3360 LOCATE 24,(5*MOT-2):PRINT S$;:COLOR S,H
3370 GOTO 2060
4000 REM
4010 REM *****
4020 REM *** Bildaufbau ***
4030 REM *****
4040 REM
4050 CLS
4055 PRINT
4060 PRINT" fischertechnik"
4070 PRINT" computing"
4080 PRINT
4090 COLOR H,S
4100 PRINT"
4110 PRINT" Diagnoseprogramm
4120 PRINT"
4130 COLOR S,H
4140 LOCATE 9,2
4150 PRINT"Motoranwahl : Tasten 1-4"
4160 PRINT" Gewählter Motor links (Puls) :F1"
4170 PRINT" Gewählter Motor links (Dauer):F3"
4180 PRINT" Gewählter Motor rechts (Puls) :F5"
4190 PRINT" Gewählter Motor rechts (Dauer):F7"
4200 PRINT" Gewählter Motor aus : Leertaste
4210 PRINT" Alle Motoren aus : Beliebige Taste"
4220 PRINT" Programmende : x
4230 PRINT
4240 PRINT" E1 E2 E3 E4 E5 E6 E7 E8 EX EY"
4250 PRINT
4260 RETURN
5000 REM
5010 REM *****
5020 REM *** Motoranzeige loeschen ***
5030 REM *****
5040 REM
5050 LOCATE 22,3
5060 PRINT"M 1 M 2 M 3 M 4 "
5070 PRINT
5080 FOR I=1 TO 4
5090 S$=STA$(INT(STA(I)/100))
5100 PRINT" ";S$;
5110 NEXT
5120 RETURN

```

## Microsoft BASIC

Die fischertechnik computing Programme der beigefügten Diskette sind in dem Microsoft BASIC geschrieben. Dabei wird, bis auf wenige Ausnahmen mit Befehlen für Bildschirmgrafik, nur die einfachste BASIC-Version vorausgesetzt. Die in den fischertechnik computing Programmieranleitungen dokumentierten Programme sind jedoch in dem BASIC eines anderen Computers formuliert. Zwar wurde darauf geachtet, daß möglichst wenige computerspezifische Merkmale einfließen. Ganz konnte es jedoch nicht vermieden werden, daß Sie die abgedruckten Programme nach Modifikationen durchführen müssen. Die wichtigsten Zeilen mit Änderungen haben wir durch ein Sternchen vor der Zeilennummer gekennzeichnet. Wir stellen hier noch einmal kurz die Unterschiede zusammen:

fischertechnik  
computing  
Programmieranleitung

Microsoft  
BASIC  
**CALL M1(EIN)**  
**CALL INIT**  
**CLS**  
**A\$=INKEY\$**

**SYS M1, EIN**  
**SYS INIT**  
**PRINT CHR\$(147)**  
**GET A\$**

Cursorsteuern vereinfachen sich mit Microsoft-BASIC, da das Kommando LOCATE zur Verfügung steht. Die Ansteuerung von inversen Schriftzeilen kann nicht im PRINT-Kommando erfolgen, sondern erfordert jeweils den Aufruf des COLOR-Kommandos. Andererseits vereinfachen sich einige PRINT-Kommandos durch die Benutzung des PRINT-Kommandos mit Formatangabe (PRINT USING).

Infolge der komfortablen Ausstattung des Microsoft BASIC mit verschiedensten Kommandos kann es zu einigen Konflikten bei der Namensgebung der Variablen kommen. In diesen Fällen müssen die Variablenamen gegenüber der Programmieranleitung abgeändert werden.

Alle notwendigen Änderungen sind jedoch in den Programmen auf der Diskette bereits durchgeführt. Auch können die Diskettenprogramme noch in anderen Details von der gedruckten Dokumentation abweichen, wo sich Vorteile ergaben.

## Checkliste

Sollte das fischertechnik computing Interface sich einmal widerwillig zeigen und nicht so arbeiten, wie Sie es erwarten, so überprüfen Sie bitte folgende Punkte mit dem Diagnoseprogramm:

Das Diagnoseprogramm zeigt bei E1 bis E8 überall 1 an, obwohl kein Modell angeschlossen ist. – Das Interface ist nicht an den Computer oder nicht an das Netzgerät angeschlossen.

Einer der Eingänge E1 bis E8 zeigt bei Betätigung des Tasters gerade das umgekehrte Ergebnis. – Öffner- und Schließfunktion des Tasters sind vertauscht.

Einer der Eingänge E1 bis E8 zeigt immer das Ergebnis 0, obwohl er angeschlossen ist und betätigt wird. – Prüfen Sie die Verkabelung.

Einer der Eingänge E1 bis E8 zeigt immer das Ergebnis 1, selbst wenn kein Modell angeschlossen ist. – Vermutlich das Eingangsgatter IC 4014 durch Überspannung (elektrostatische Aufladung) beschädigt. Ein Motorausgang arbeitet nicht. – Bitte Verkabelung überprüfen.

Ein Motorausgang arbeitet nur in einer Richtung. – Leistungsstufe des Motors defekt.

Motor läuft sehr langsam oder setzt aus. – Entweder Netzgerät durch zu viele Motoren überlastet (zwei Netzgeräte mot4 oder das stärkere computing Netzgerät verwenden) oder Netzgerät bei Verwendung des regelbaren Ausgangs nicht voll aufgedreht.

Bei Defekten schicken Sie das Gerät bitte an die fischerwerke, Abt. Service, ein.

## Technische Daten

fischertechnik computing Interface IBM Personal Computer für IBM Personal Computer und kompatible Computer, Art.-Nr. 30567.

4 Ausgänge zum Anschluß von Motoren, Lampen, Elektromagneten.. (M1 bis M4).

Polarität des Ausgangs steuerbar.

Belastbarkeit: 1A Dauerstrom, 1,5A Spitzenstrom.

8 Eingänge für digitale Signale (E1 bis E8).

Durch interne Beschaltung sowohl Anschluß von elektromagnetischen Artikeln (Taster, Schalter, Relais) in positiver Logik als auch Anschluß von TTL-Ausgängen möglich. Schutz gegen Überspannung eingebaut.

2 Eingänge für analoge Signale (EX und EY).

Anschließbar sind Geber mit Widerstandswerten zwischen 0 und 5 k $\Omega$ , z. B. Potentiometer, Fotowiderstände...

Überwachungsschaltung des Datenstroms. Bei Ausbleiben von Datensignalen des PC schaltet das Interface nach 0,5 Sekunden alle Ausgänge inaktiv. Die Signale bleiben jedoch gespeichert.

Überwachungsschaltung der Software. Bei gravierenden Syntaxfehlern spricht ebenfalls die Überwachungsschaltung, jedoch ohne Verzögerung, an. Die Überwachungsschaltung reagiert auch auf Unterversorgung des Interface, sei es durch Überlastung oder zu niedrige Spannung des Netzgeräts. Software mit Interfacesteuerungsbefehlen und Beispielprogrammen für die fischertechnik computing Modelle im Lieferumfang enthalten.

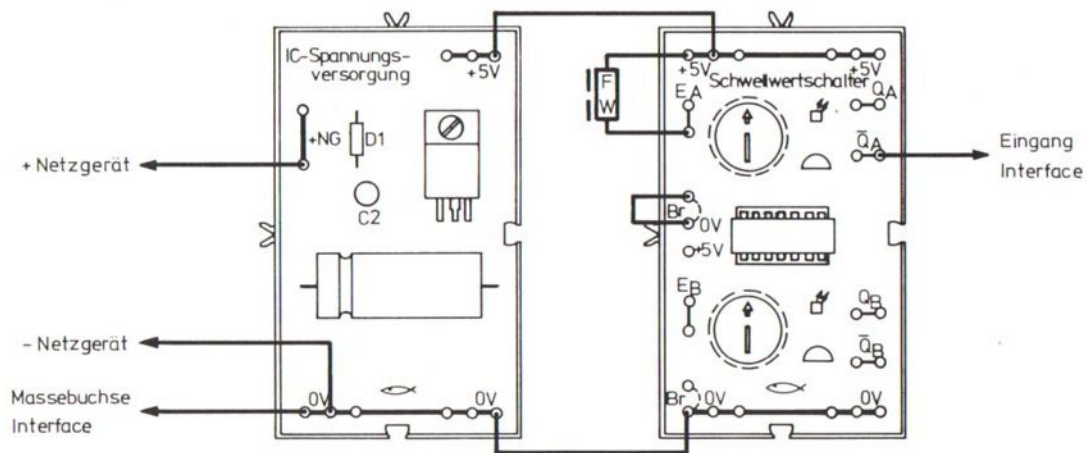
## Benutzung von fischertechnik Elektromechanik und Elektronik

Das fischertechnik computing Interface ist kompatibel zu den Bauteilen und Elektronikbausteinen der obengenannten Baukästen. Anstelle der bei den fischertechnik computing Modellen verwendeten mini-Taster können Sie genauso gut Taster und Schalter anderer Bauart anschließen. Z.B. den großen Taster oder den Polwendeschalter, aber auch den Reedkontakt oder den Schaltkontakt eines Relais. Aufpassen müssen Sie jedoch bei der Verwendung von selbstgebauten Tastern und Schaltern aus Gelenkbausteinen und Federn. Hier könnten eventuell Prellerscheinungen auftreten. Wir empfehlen, in diesen Fällen den Taster mehrmals abzufragen und den Wert nur dann als gültig zu erachten, wenn zweimal hintereinander der gleiche Wert erschien.

Die Analogeingänge des Interface können mit Sensoren beschaltet werden, die einen Widerstandswert zwischen 0 und 5 k $\Omega$  als Ausgang liefern. Zunächst bieten sich die Potentiometer aus dem Baukasten fischertechnik computing an. Genauso können aber auch andere Bauelemente, wie z.B. der Fotowiderstand, verwendet werden.

Die Motorausgänge des Interface sind kräftig belastbar. Nicht nur die mini-Motoren, auch der S-Motor und der N-Motor lassen sich mit dem Interface ansteuern, wobei noch eine Lampe zur Funktionsanzeige parallelgeschaltet sein darf. Außer Motoren eignen sich noch der Elektromagnet und das Relais RBII.

Die Signale der Elektronikbausteine mit integriertem Schaltkreis aus der TTL-Familie (z.B. Schwellwertschalter) können ebenfalls in die Eingänge des Interface eingespeist werden. Als gemeinsamer Bezugspunkt ist jedoch zuvor die Massebuchse des Elektronikbausteins mit der Massebuchse des Interface zu verbinden. In der Abbildung zeigen wir, wie eine Lichtschranke aufgebaut wird. Der Schwellwertschalter dient dazu, die Ansprechschwelle der Lichtschranke einzustellen.



## Funktionsweise des Interface und des Interface-Treiberprogramms

Wenn Sie die fischertechnik computing Software benutzen oder selbst Programme entsprechend der Hinweise in den vorigen Kapiteln erstellen, werden Sie kaum die nun folgende Information benötigen. Wenn Sie aber die Programme in anderen Sprachen als BASIC formulieren wollen, die Programme durch komplexe Abläufe in Maschinensprache beschleunigen wollen, die Funktionen des Interface erweitern wollen oder auch nur einfach einen Blick hinter die Kulissen werfen wollen, so wird Ihnen das Nachfolgende sicherlich hilfreich sein. Allerdings sollten Sie dann auch ein paar Kenntnisse der Maschinensprache und der Digitalelektronik mitbringen, denn hier geht es an die „bits and pieces“.

Das fischertechnik Interface erfüllt eine Reihe von Aufgaben, die wir anhand des Blockdiagramms besprechen wollen. Am linken Rand sind die Signale von und zu dem Computer aufgeführt. Es fällt auf, daß diese recht wenig mit den Ausgängen M1 bis M4 und Eingängen E1 bis E8 sowie EX und EY gemein haben. Der Grund ist darin zu suchen, daß am Computeranschluß wesentlich weniger Datenleitungen zur Verfügung stehen, als auf der Modellseite des Interface benötigt werden. Diese wenigen Datenleitungen müssen deshalb so eingesetzt werden, daß alle Signale auf der Modellseite gesteuert werden können. Das Konzept sieht eine Mehrfachverwendung der Datenleitungen mit Hilfe von Schieberegistern vor. Auf diese Weise werden z.B. nur drei Datenleitungen für die Steuerung der Ausgabe notwendig. Eine parallele Anschlußweise hätte acht Datenleitungen benötigt.

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

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

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

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

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

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

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

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

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

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

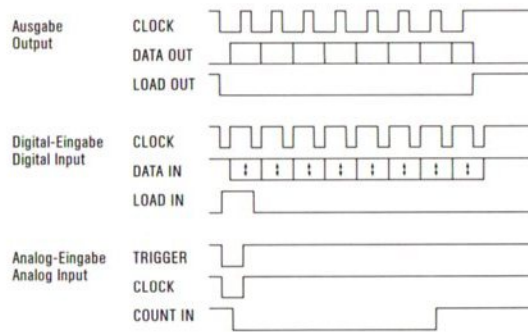
Wir wollen jetzt noch kurz zusammenfassen, wie das Interface mit dem PC verbunden wird. Es wird, wie Sie ja schon von dem Kapitel über den Anschluß des Interface wissen, die Parallel-Druckerschnittstelle verwendet. Von deren acht Datenleitungen werden die untersten sechs für die oben besprochenen Ausgabesignale verwendet, siehe auch die nachstehende Tabelle.

Ein kleines Problem gilt es für die Eingabeleitungen zu lösen. Die Parallel-Druckerschnittstelle beinhal-

tet nur eine für alle PC verbindliche Eingabeleitung, das Aktiv-Signal des Druckers, BUSY. Es entsteht jedoch kein Konflikt, wenn alle Eingabeleitungen mit Hilfe einer ODER-Schaltung zusammengefaßt werden. Da das Maschinenprogramm ja „weiß“, welche Eingabefunktion es angefordert hat, kann es die Signale an dieser einzigen Eingabeleitung auch korrekt interpretieren.

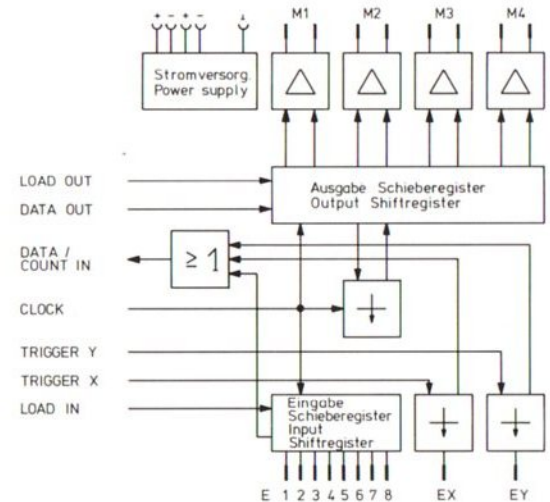
### Anschluß des fischertechnik Interface an die Parallel-Druckerschnittstelle

Interface-Signal	Drucker-Signal	Stiftnummer
<b>LOAD-OUT</b>	<b>Datenbit 1</b>	<b>2</b>
<b>LOAD-IN</b>	<b>Datenbit 2</b>	<b>3</b>
<b>DATA-OUT</b>	<b>Datenbit 3</b>	<b>4</b>
<b>CLOCK</b>	<b>Datenbit 4</b>	<b>5</b>
<b>TRIGGER-X</b>	<b>Datenbit 5</b>	<b>6</b>
<b>TRIGGER-Y</b>	<b>Datenbit 6</b>	<b>7</b>
<b>DATA/COUNT-IN</b>	<b>Busy</b>	<b>11</b>



Impulsdigramme des fischertechnik Interface  
Pulse Diagrams of the fischertechnik Interface

Das nachstehende Maschinenprogramm steuert die zuvor beschriebenen Abläufe. Zu der Übergabe der Parameter in das Maschinenprogramm und wieder zurück studieren Sie die Beschreibung des CALL-Befehls bzw. der USR-Funktion in der BASIC-Anleitung Ihres PC. In dem vorliegenden Quelltext wird von einer Druckeradresse 03BC (hexadezimal) ausgegangen, die bei FFFA und den nachfolgenden Speicherstellen abgelegt ist. Das Programm FISCHER modifiziert ggf. diese Stelle entsprechend den Gegebenheiten Ihres Computers. Die am Ende der Routine für die Analogeingabe vorzufindenden NOP-(no operation)-Befehle dienen als Platzhalter. Bei dem Einsatz des fischertechnik-Interface an PC mit schnelleren Mikroprozessoren werden hier, ebenfalls durch das Programm FISCHER, die entsprechende Zahl von Rechtsschiebe-Befehlen eingesetzt.



```

*****
* IBM-PC Interface Treiber
*
* Version 1.2
* File INTERFAC.COM
*
* Copyright (C) fischerwerke,1986
* Steuerung des fischertechnik Interface
* für Aus- und Eingabe mithilfe des CALL-
* Kommandos und der USRO-Funktion.
*****
* Dieses Programm benötigt eine
* Centronics-Parallele Drucherschnittstelle
*
* Das Maschinenprogramm steht oberhalb
* des Basic Ramtop und wird somit
* durch einfache Befehle wie z.B.
* NEW, LOAD nicht gelöscht.
* Das Programm nimmt 256 Bytes vom Basicspeicher
* ein.
*
* Ausgabe Steuerung:
*
* CALL Motornummer(Betriebsart)
* Motornummern sind M1, M2, M3 und M4.
* Betriebsart ist RECHTS, LINKS, EIN und AUS.
* EIN entspricht RECHTS.
*
* CALL INIT
* Initialisiert das Interface und schaltet alle
* Motoren ab.
*
* Eingabe-Steuerung:
*
* Digital-Eingabe Kommandos:
* USR(Digital-Eingänge)
* Digital-Eingänge sind E1, E2, E3, E4, E5, E6,
* E7 und E8.
*
* Analog-Eingabe Kommandos:
* USR(Analog-Eingang)
* Analog-Eingänge sind EX und EY.
*
* Anschluß des Interface an den Drucker-Port:
* Interface Drucker-Port Leitung
* OV GND 17-25
* LOAD-OUT Data 1 2
* LOAD-IN Data 2 3
* DATA-OUT Data 3 4
* CLOCK Data 4 5
* TRIGGER-X Data 5 6
* TRIGGER-Y Data 6 7
* DATA/COUNT-IN -Busy 11
*
* Routine zur Ausgabe-Steuerung
* Einsprung mit dem CALL-Kommando
*****
:FF00 BB0001 MOV BX,0100 ;INIT
:FF03 FA CLI ;Interrupt sperren
:FF04 EB24 JMP FF2A
:FF06 B003 MOV AL,03 ;M1
:FF08 E80A JMP FF14
:FF0A B00C MOV AL,0C ;M2
:FF0C EB06 JMP FF14
:FF0E B030 MOV AL,30 ;M3
:FF10 EB02 JMP FF14
:FF12 B0C0 MOV AL,C0 ;M4
:FF14 FA CLI ;Interrupt sperren
:FF15 8A1EBEFF MOV BL,[FFBE] ;alten Wert holen
:FF19 08C3 OR BL,AL
:FF1B 55 PDSH
:FF1C 89E5 MOV BP,SP
:FF1E 8B7606 MOV SI,[BP+06] ; Betriebsart holen

:FF21 8A24 MOV AH,[S1]
:FF23 5D POP BP
:FF24 B700 MOV BH,00
:FF26 20C4 AND AH,AL
:FF28 30E3 XOR BL,AH ;Motorbit ausrechnen
:FF2A 8B1EBEFF MOV [FFBE],BL ;neuen Wert speichern
*****
* Routine zur Interface-Steuerung
* Ausgabe-Steuerung
* Ausgabe-Bitmuster in BL
* benutzt AL, BL, CX und DX
*****
:FF2E B90800 MOV CX,0008 ;Zählschleife
:FF31 8B1EBAFF MOV DX,[FFBA] ;Adr. Druckerport
:FF35 B030 MOV AL,30 ;Ruhepegel am Port
:FF37 D0D3 RCL BL,1 ;teste Ausgabe-Muster
:FF39 7302 JNB FF3D ;DATA-OUT low
:FF3B 0C04 OR AL,04 ;DATA-OUT high
:FF3D EE OUT DX,AL ;Ausgabe an Port
:FF3E C008 OR AL,08 ;CLOCK setzen
:FF40 EE OUT DX,AL ;Ausgabe an Port
:FF41 E2F2 LOOP FF35 ;Schleifenende
:FF43 B039 MOV AL,39 ;LOAD-OUT high
:FF45 EE OUT DX,AL ;Ausgabe an Port
:FF46 FB STI ;Interrupt freigeben
:FF47 80F01 CMP BH,01 ;teste Einsprungart
:FF48 F4F4 JZ FF4F
:FF4C CA0200 RETF 0002 ;Rücksprung von Mn(Betr.)
:FF4F CB RETF ;Rücksprung von INIT
*****
* Routine zur Interface-Steuerung
* Eingabe-Steuerung
* Einsprung mit der USRO-Funktion
*****
:FF50 FA CLI ;Interrupt sperren
:FF51 8A07 MOV AL,[BX] ;Argument von USRR
:FF53 8B1EBAFF MOV DX,[FFBA] ;Adr. Druckerport
:FF57 3CA0 CMP AL,A0 ;Analog-Eingabe?
:FF59 743C JZ FF97 ;weiter zu FF97
:FF5B 3C90 CMP AL,90 ;Analog-Eingabe?
:FF5D 7438 JZ FF97 ;weiter zu FF97
:FF5F A2B8FF MOV [FFBF],AL ;Eingabe-Maske speichern
*****
* Interface-Ansteuerung
* benutzt AL, CX und DX
*****
:FF62 B90800 MOV CX,0008 ;Zählschleife
:FF65 B032 MOV AL,32 ;LOAD-IN high
:FF67 EE OUT DX,AL ;Ausgabe an Port
:FF68 0C08 OR AL,08 ;CLOCK setzen
:FF6A EE OUT DX,AL ;Ausgabe an Port
:FF6B D0D4 RCL AH,1 ;LSBit freisachen
:FF6D 8B1EBCFF MOV DX,[FFBC] ;Adr. Busy-Leitung
:FF71 EC IN AL,DX ;DATA-IN einlesen
:FF72 2480 AND AL,80 ;Maskiere DATA-IN
:FF74 3C80 CMP AL,80 ;teste DATA-IN
:FF76 7503 JNZ FF7B ;DATA-IN high (0)
:FF78 80CC01 OR AH,01 ;DATA-IN low (1)
:FF7B F8 CLC
:FF7C 8B1EBAFF MOV DX,[FFBA] ;Adr. Druckerport
:FF80 B030 MOV AL,30 ;Ruhepegel am Port
:FF82 EE OUT DX,AL ;ausgeben
:FF83 B038 MOV AL,38 ;CLOCK setzen
:FF85 EE OUT DX,AL ;Ausgabe an Port
:FF86 E2E3 LOOP FF8B ;Schleifenende
:FF88 A0B8FF MOV AL,[FFBF] ;hole Bitmaske
:FF8B 20E0 AND AL,AH ;Bit herausmaskieren
:FF8D 3C00 CMP AL,00 ;teste Bit
:FF8F 7402 JZ FF93 ;Rückgabe 0
:FF91 B001 MOV AL,01 ;Rückgabe 1
:FF93 8B01 MOV [BX],AL ;in FAC ablegen
:FF95 FB STI ;Interrupt freigeben
:FF96 CB RETF ;Rücksprung von USRO
*****
:FF97 B9FF03 MOV CX,03FF ;Zählschleife
:FF9A EE OUT DX,AL ;Monoflop triggern
:FF9B B038 MOV AL,38 ;CLOCK setzen
:FF9D EE OUT DX,AL ;Ausgabe an Port
:FF9E 8B1EBCFF MOV DX,[FFBC] ;Adr. Busy-Leitung
:FFA2 EC IN AL,DX ;COUNT-IN einlesen
:FFA3 D0D0 RCL AL,1 ;COUNT-IN testen
:FFA5 7303 JNB FFAB ;COUNT-IN low
:FFA7 E2F9 LOOP FFAB ;weiter abfragen
:FFA9 B8FF03 MOV AX,03FF ;Analogwert ausrechnen
:FFAC 29C8 SUB AX,CX ;als 1023-CX
:FFAE 90 NOP ;Platz für SAR AX,1
:FFAF 90 NOP
:FFB0 90 NOP ;Platz für SAR AX,1
:FFB1 90 NOP
:FFB2 8807 MOV [BX],AL ;in FAC ablegen
:FFB4 886701 MOV [BX+1],AH
:FFB7 FB STI
:FFB8 CB RETF ;Interrupt freigeben
;Rücksprung von USRO
*****
:FFBA Adresse für Datenausgabe
:FFBC Adresse für Busy-Leitung
:FFBE Speicher für Ausgabevariable
:FFBF Speicher für Bitmaske

```

## The fischertechnik computing Interface

Dear friend of fischertechnik:

fischertechnik computing has been developed to allow computer owners to control technical models with a computer, thereby extending the computer's capabilities. fischertechnik computing includes the fischertechnik computing Kit, the fischertechnik computing construction sets, the fischertechnik computing interfaces and software. Computer owners can now simulate technical functions and processes, solve problems and have a lot of fun controlling models with the computer.

What is needed for controlling the models? First, the fischertechnik model which will perform the processes. Second, a personal computer such as the one you own to provide for control and coordination. Third, an interface which will link the two. The item in your hands is the fischertechnik computing interface. Control signals from the computer, e.g. "Turn

on motor!" are converted by the interface into currents strong enough to make a motor move. This is called "output". The point of view is from the computer outward.

The opposite direction is also possible and occurs in practice. The models are equipped with pushbutton switches, potentiometers, etc., in order to inform the computer about what is happening on the model. Here again, the interface provides assistance by modifying the signals in such a way that they represent an input which is meaningful to the computer.

The fischertechnik interface provides the following functions:

- With it you can control fischertechnik motors, lamps, electromagnets, etc.
- With it you can interrogate eight pushbuttons or switches.
- Additionally, two inputs provide values from variable signal transmitters such as potentiometers.

What would be the use of all the electrical connections between the computer and the fischertechnik model provided by the interface without a means of activating them? We are referring to software, of course. The software is located on a diskette. The diskette contains a program which expands the language of your computer in such a way that it can control the interface. This program will be the core of your own programs. The diskette contains a lot more, though: in order for you to study and learn the use of these new resources, sample programs for all fischertechnik computing models are also included on it.

As you can see, a lot of challenging problems are waiting to be solved. I hope you will enjoy solving them! Yours sincerely,



## Connecting the Interface

The fischertechnik interface for the IBM Personal Computer can be connected to the IBM PC, the IBM PPC, the IBM XT and the IBM AT computers. These computers have also created a standard, the standard of IBM compatibility. A number of other computers are therefore available which resemble the IBM computers in many ways. Compatible computers may not and must not be one hundred percent identical with the IBM models. This is why we cannot say for sure which compatibles will allow fischertechnik computing to operate without modification. You will have to try it out for yourself if you are the owner of an IBM compatible computer; perhaps your dealer will give you the opportunity to do so.

Due to the great variety of models and designations, from now on we will refer to the IBM or compatible personal computers using the short term "PC".

The fischertechnik computing interface is hooked up to the parallel printer interface of the PC. The interface can be connected to all the compatibles since there are no differences as far as the parallel port is concerned.

Another feature which these computers have in common is the fact that they all understand the same operating system, MS-DOS or PC-DOS, starting with version 2.0. The fischertechnik interface is normally operated with diskette BASIC which is usually loaded by typing in BASIC. If the programs use screen graphics, a graphics board and the expanded version of BASIC, BASICA, or GWBASIC, are required. Let's get back to the hardware. The fischertechnik interface is connected to the PC's printer port, as we have already mentioned above. Use the following procedure:

- Make sure the computer is turned off.
- Turn the computer around so that you are facing the ports on the rear panel. Locate the parallel printer port. This port is a jack connector which will accept 25 pins. You cannot mistake it for the

other interfaces, for example, the serial interface, because they have pins, not jacks.

- You may be electrostatically charged to several thousand volts by frictional electricity without noticing it and without it doing you any harm. However, these voltages are harmful to the circuits contained in the interface and the computer. It is therefore necessary to discharge any electrostatic charges by touching a grounded object or the metal housing of the computer.
- Place the interface in front of you. A flat ribbon cable is attached to the interface. At the end of the cable is a 20-pole connector which fits into the adapter. The adapter carries a second connector which fits to the printer port of your PC. The connector is plugged into the parallel printer port; plugging it in the wrong way is impossible due to the trapezoidal shape of the connector.
- Connect the interface to the power supply unit. The interface expects an unfiltered direct current between 6 and 10 volts. If a stabilized direct voltage source is used, the power supply must be adjusted to approximately 8 volts (loadable to approximately 2 amps). Using the fischertechnik computing power supply unit will guarantee the correct voltage. Plug the output cable of the power supply unit (red flat plug) into the jack marked with a "+" on the interface, and the output cable with the green flat plug into the jack marked with a "-". Two pairs of jacks are provided; it does not matter which pair you use.
- For models with up to two motors, the fischertechnik power supply units mot 4 may also be used, i.e. especially for the models of the fischertechnik computing kit. If you intend to use your own constructions with a greater number of motors or the training robot or Plotter/Scanner kits, you will need a more powerful power source. In this case, you can either switch to the more powerful fischertechnik computing power supply unit or

plug a second fischertechnik power supply unit mot 4 into the empty jacks.

- When using the mot 4 power supply units, you use the two-wire cable and flat plugs supplied with the kits to make the power cord.
- Connect the fischertechnik computing model to the interface. Use the twenty-wire ribbon cable supplied with the kits and the fischertechnik computing kit. This cable can also be ordered separately from the fischertechnik Service Set.
- The turn-on sequence of the interface and the computer does not matter. If you are not using the interface while working with other programs, it is not necessary to unplug the interface. Just leave it turned off.
- Side effect caused by the interface: as long as the interface is plugged in, you cannot use a printer which is to be connected to the same port. However, the operating system of the PC is designed to handle up to four printer ports, so you can use another port if necessary.

Prior to working with the models you should also always discharge possible electrostatic charges by touching a grounded metal object, e.g. the metal housing of the PC.

Enough about hardware – in the next section we'll deal with the fischertechnik computing software.



## fischertechnik computing Software

Anyone who has toyed with the idea of controlling some sort of device or model with a computer may know from experience or may have heard from friends that this is not the easiest of tasks. That it requires a thorough knowledge of the computer, the microprocessor, the input/output devices and machine language.

These difficulties, which were considerable, were a good reason for many not to experiment with this fascinating field of computing – up to now, that is. fischertechnik computing provides the proper interface for your computer, and with it, the software you need to make it all work smoothly and easily.

In order to avoid losing the software through accidental deletion or damage to the diskette, you should make a copy of the fischertechnik computing diskette. For this purpose you use the program DISKCOPY contained on the MS-DOS system diskette. Proceed exactly according to the instructions of the MS-DOS manual and the program prompts. When you have made the copy, store the fischertechnik computing diskette in a safe place inaccessible to the natural enemies of diskettes such as sand, heat, cats or magnetic fields. Work with the copy and only use the original fischertechnik computing diskette to make another copy if necessary. Additional copies are especially useful if you want to change the programs while retaining the contents of the original diskette. You are restricted to making copies for your own use.

Some fischertechnik programs will write data to a diskette. While you are at it, you should format some additional diskettes.

If you have not already connected the interface to your computer as described in the previous section, you should do so now. Turn the computer on. Start BASIC by typing

### BASIC

When BASIC has been loaded, insert the fischer-

technik computing diskette drive A:. Now select the appropriate sub-directory using the command:

### CHDIR"A:ENGLISH"

Load the program called FISCHER using the following command:

### RUN"A:FISCHER"

After a few seconds, a message will appear on the screen asking you whether you want to adapt the software to your PC. If you are using the diskette for the first time and your computer is not an IBM PC, answer the question by typing

Y

for "yes". The program will then ask you for the processing speed of your computer and the port number of the printer card used. Finally, the program INTERFAC.COM is configured for the working conditions of your PC. It is not necessary to go through this configuration routine every time you use the program. Since the configuration is written to diskette, the appropriate software will be available whenever the system is subsequently started. If you should use your fischertechnik computing diskette on another computer, however, you may have to run the configuration routine again.

Finally, the program will display the directory of the diskette.

Next load the driver program DRIVER.BAS into the computer. Type

### RUN"A:DRIVER"

After having loaded the program the diskette drive will spin again briefly, since the driver program is loading the machine language program INTERFAC.COM which contains the above-mentioned detail knowledge about the input/output devices of the computer. Subsequently you will be asked for the screen color setting. At this point you can optimize the screen color settings for your monitor using the

function keys. Then the PC will go back to the ready prompt.

No apparent changes have taken place. However, your PC now has a number of new commands available which were not previously a part of BASIC in this form. These commands are tailored exactly to your PC and the fischertechnik computing interface. They will use the CALL command and the USR function. The driver program itself is written in BASIC and generates the parameters of the machine language program calls in the variable storage memory of the PC.

The only commands you need to be able to use are the following ones:

The motor output M1 is controlled with the following commands:

**CALL M1(MCW)**  
**(clockwise rotation)**

**CALL M1(MCCW)**  
**(counterclockwise rotation)**

**CALL M1(MOFF)**  
**(turn off)**

The command parameters designate the motor, motor 1 in this case, and the mode of operation. The same command is used with the parameters M2, M3 and M4 to control the other three outputs. The 10 inputs of the interface are read with the aid of the USR function. The function

### USR(E1)

is 1 if +5 volts apply at input E1 of the interface. Otherwise USR(E1) will output 0. Similarly, USR(E2)...USR(E8) will display the status of the other digital inputs.

The analog inputs EX and EY are each connected to +5V via a potentiometer (4.7 k $\Omega$ ). The functions

**USR(EX)**  
**USR(EY)**

will then produce a value within the range between

20 and 230 according to the position of the potentiometer. The actual limits may be slightly different. They depend on the execution speed of your PC's microprocessor. If you find an extremely different range, make sure to run through the program FISCHER and to configure the file INTERFAC.COM. Analog inputs are mostly used for measuring positions. If, for example, a robot arm is driven by a motor, and potentiometer EX is adjusted synchronously with the motion of the arm, then the program, by repeatedly calling the function

### **USR(EX)**

can trace the movement of the robot exactly. If no potentiometer is connected to inputs EX or EY, there will be an overflow of the internal evaluation counter. The value of the function is then either 255, 511 or 1023.

The last of the new commands is

### **CALL INIT**

This command is used to switch the interface to a well-defined initial state. The command can also be used to turn off all motor channels simultaneously.

Enough introductory remarks, though. Connect a fischertechnik motor to M1 using the twenty-wire, color-coded ribbon cable. Use the yellow and the orange cables in the upper half of the ribbon cable. Type:

### **CALL M1(MCW)**

The motor will briefly start up and then stop again. Congratulations! You've just learned how to control the most complicated fischertechnik models with your PC.

However, you would probably like to find out why the motor has stopped. Didn't you turn it on? Isn't there a separate command for turning it off, as described above? Well – the motor was stopped, but the value

stored in the interface says that it is still running. The interface itself has "gone to sleep". This always happens when another command is not received within half a second and is programmed this way for safety reasons. Let's say you are testing a new program. The probability that there is a mistake hidden away somewhere in the program borders on certainty. The program run is interrupted with a message such as

### **Syntax error in...**

The motor, which was turned on a short time ago, however, continues running and is about to demolish that beautiful model you worked so hard to build. You would have to run to the power supply unit and turn the power off.

How good it is to know, then, that the motor will stop by itself. Even if you have interrupted program execution by pressing CTRL-BREAK. And if the program is resumed, the first command "wakes up" the interface, and none of the motors has been forgotten. The procedure can go on as if nothing had ever happened.

There is a reason why the interface is turned off with a delay. Due to calculations, there will always be delays between input and output commands to the interface. These delays must be bridged. You can tell by looking at the light emitting diode (LED) on the interface whether the interface is being activated by input or output commands. The LED not only indicates voltage but also shows if the interface is operative.

Next we will take a brief look at the input commands. Connect a pushbutton switch between E1 (the brown wire on the lower edge) and +5V (the red wire in the center of the ribbon cable).

Type in the following command:

### **PRINT USR(E1)**

Depending on whether or not the pushbutton switch between E1 and +5V was depressed when the

return key on the computer was pressed, the screen displays a 1 or a 0.

Since the motor is still connected to the output from before, it will start up once more. Input commands will also reactivate the outputs of the interface! Connect a 4.7 kΩ potentiometer between EX and +5V. Adjust the slider to a medium position and type in

### **PRINT USR(EX)**

The number which is displayed on the screen must be in the range between 0 and 255.

You will regularly need the driver program which you were using just now. Every program written to control models via the fischertechnik computing interface begins with this routine which installs the new commands. The driver program is already contained in all the sample programs on the diskette. All programs that you write yourself must start with the driver program and continue on subsequent line numbers.

In order to be able to observe the potentiometer more easily, we will now write the first fischertechnik computing program in this manner. The driver program is loaded into the computer and occupies the line numbers from 1 to 500. Type in:

### **510 PRINT USR(EX)**

**520 GOTO 510**

**RUN**

There may be a brief delay while the INTERFAC.COM file is being loaded. The question concerning the color screen settings will be displayed. And then it will begin: all at once the screen will be filled with numbers which will continuously scroll up. If you then take the potentiometer in your hand and move the slider, you will be able to observe the change in the numbers. Turn the potentiometer from one stop to the other. The numbers displayed should be between 0 and 255. The display value, however, will not quite reach 0 or 255; a range of between 20 and 230

should be displayed in practice. Press CTRL-BREAK to interrupt the program.

For those who are interested in more detailed information about these processes and who do not merely wish to use the programs included on the diskette, we will provide some detailed information here. The function of the driver program is to write a short machine language program to a memory range of the PC. This program is contained on the diskette as the INTERFAC.COM file. The machine language program occupies the memory range from FF00 to FFFF (addresses given in hexadecimal notation). The memory range available to BASIC is thus shortened by 256 Bytes, which, however, does not normally cause a noticeable limitation.

Besides the machine language program, the standardized parameters M1, M2, M3, M4, E1, E2, E3, E4, E5, E6, E7, E8, EX, EY, MCW, MCCW and MOFF are set. Also, the variables beginning with the letters E or M are defined as integer variables.

The following limitations therefore apply when you write your own BASIC programs:

Just as reserved BASIC key words such as PRINT or STOP cannot be used as variables, the use of the above-mentioned parameters as variables is prohibited. However, the limitation does not apply to variables of another type (strings, real), so that

#### MOFF\$, M1!, E1! . . .

may be used.

The USR function 0 is used by the driver program. You can use the other USR functions in your programs as desired.

The above-mentioned memory range from FF00 to FFFF cannot be used for other purposes.

Those interested in further details, e.g. in the electronic processes in the interface or adaptations to other programming languages, should refer to the description of the interface and the source code.

The last function of the base program is to turn on the interface and turn off all connected loads. This is caused by issuing the following command:

#### 500 CALL INIT

This switches the interface to the ready state for a subsequent user program. All sample programs on the diskette are built according to this pattern. We suggest you study them for hints and ideas to use in designing your own programs.

```
10 REM Interface driver routine for
20 REM IBM Personal Computer
30 REM Copyright (C) fischerwerke 1986
40 REM Version 2.1
50 CLEAR ,&HFF00
60 DEF SEG
70 DEF USR=&HFF50
80 BLOAD"A:INTERFAC.COM",&HFF00
90 DEFINT E,M
100 LET S=7 : H=0 : R=0
110 SCREEN 0,1 : KEY OFF : WIDTH 40
120 COLOR S,H,R : CLS
130 LOCATE 2,1,0
140 PRINT" fischertechnik"
150 PRINT" computing"
160 PRINT
170 PRINT" Adjust screen colours ? (y/n)"
180 Y$=INKEY$
190 IF Y$="n" OR Y$="N" THEN GOTO 420
200 IF Y$("<"y" AND Y$("<"Y" THEN GOTO 180
210 KEY 1,"r" : KEY 3,"h" : KEY 5,"s" : KEY 7,"p"
220 COLOR S,H,R
230 CLS
240 LOCATE 2,1
250 PRINT" fischertechnik"
260 PRINT" computing"
270 PRINT
280 PRINT" F1 - Border"
290 PRINT" F3 - Paper"
300 PRINT" F5 - Pen"
310 PRINT" F7 - Start of program"
320 F$=INKEY$
330 IF F$="" THEN GOTO 320
340 IF F$="p" THEN GOTO 400
350 IF F$="r" THEN LET R=R+1 AND 15
360 IF F$="h" THEN LET H=H+1 AND 7
370 IF F$="s" THEN LET S=S+1 AND 15
380 GOTO 220
390 REM
400 F$=INKEY$:KEY 1,"LIST ":KEY 3,"LOAD"+CHR$(34)
410 KEY 5,"CONT"+CHR$(13):KEY 7,"TRON"+CHR$(13)
420 INIT=&HFF00:M1=INIT+6:M2=M1+4:M3=M2+4:M4=M3+4
430 E1=1:E2=2:E3=4:E4=8:E5=16
440 E6=32:E7=64:E8=128
450 EX=180:EY=144
460 MCW=85:MCCW=170:MOFF=255
500 CALL INIT
```

## The Diagnostic Program

If you have constructed a fischertechnik computing model, you may find that things do not work as you had expected. No wonder considering the great number of lines running between the model and the interface. If only one pushbutton switch were incorrectly wired, this could produce the most astounding results. Matters become twice as complicated when you write your own programs. Where do you start looking for errors? Is it a hardware problem? Is the software to blame?

In order to help you test the hardware clearly and conveniently, we have developed the diagnostic program. It is included on the fischertechnik computing diskette as the DIAGNOST.BAS file. Always load this program for testing a model. It allows you to monitor all the inputs and to verify whether their behavior corresponds to your expectations.

Select a control output using the numeric keys. It is displayed inversely on the screen. This selected output can then be turned on (clockwise or counterclockwise rotation) using the function keys. Besides verifying whether the motor is operational, you can also check if it runs in the desired direction of rotation. If this is not the case, the two motor connections must be exchanged.

Hit the space bar to turn off the selected motor. Hitting any other key will stop **all** motors. Use the key X to exit from the program.

```

500 CALL INIT
510 REM
520 REM fischertechnik computing
530 REM
540 REM Diagnostic Program
550 REM
560 REM Copyright (C) fischerwerke 1986
570 REM
580 REM Function
590 REM Using this program you may control
600 REM all functions of a model.
610 REM All inputs are displayed on the screen.
620 REM All outputs are controlled via
630 REM keyboard commands.
1000 REM
1010 REM *****
1020 REM *** Initialize control ***
1030 REM *****
1040 REM
1050 DIM STA(4),STA$(2): REM status of the motors
1060 DIM M(4): REM addresses for motor 1 thru 4
1070 DEFINT Z
1080 GOSUB 4000: REM set up of screen
1090 REM switch all motors off
1100 FOR I=1 TO 4
1110 LET STA(I)=MOFF
1120 NEXT I
1130 REM control codes
1140 LET SP$=CHR$(32)
1150 KEY 1,CHR$(224): LET F1$=CHR$(224)
1160 KEY 3,CHR$(225): LET F3$=CHR$(225)
1170 KEY 5,CHR$(226): LET F5$=CHR$(226)
1180 KEY 7,CHR$(227): LET F7$=CHR$(227)
1190 REM text for status display
1200 STA$(0)=" cw"
1210 STA$(1)="ccw"
1220 STA$(2)="off"
1230 LET M(1)=M1:M(2)=M2:M(3)=M3:M(4)=M4
1240 LET MOT=1: REM start with motor 1
2000 REM
2010 REM *****
2020 REM *** display of input ***
2030 REM *****
2040 REM
2050 GOSUB 5050: REM erase motor status display
2060 LOCATE 20,2
2070 PRINT USR(E1);
2080 PRINT TAB(5);USR(E2);
2090 PRINT TAB(8);USR(E3);
2100 PRINT TAB(11);USR(E4);
2110 PRINT TAB(14);USR(E5);
2120 PRINT TAB(17);USR(E6);
2130 PRINT TAB(20);USR(E7);
2140 PRINT TAB(23);USR(E8);
2150 PRINT TAB(26);USING"####";USR(EX);
2160 PRINT TAB(31);USING"####";USR(EY)
3000 REM

```

```

3010 REM *****
3020 REM *** control of output ***
3030 REM *****
3040 REM
3050 K$=INKEY$
3060 LET F=0: REM flag for valid command
3070 M=VAL(K$): REM numeric key
3080 IF M>=1 AND M<=4 THEN LET MOT=M:LET F=1
      GOSUB 5050
3090 LET Z=STA(MOT)
3100 IF K$=SP$ THEN LET Z=MOFF:STA(MOT)=MOFF
      LET F=1
3110 IF K$=F1$ THEN LET Z=MCCW:LET F=1
3120 IF K$=F3$ THEN LET Z=MCCW:STA(MOT)=MCCW
      LET F=1
3130 IF K$=F5$ THEN LET Z=MCW:STA(MOT)=MCW
      LET F=1
3140 IF K$=F7$ THEN LET Z=MCW:LET F=1
3150 IF K$<"x" THEN GOTO 3240
3160 CALL INIT
3170 WIDTH 80
3180 COLOR 7,0
3190 KEY 1,"LIST"
3200 KEY 3,"LOAD"+CHR$(34)
3210 KEY 5,"CONT"+CHR$(13)
3220 KEY 7,"TRON"+CHR$(13)
3230 KEY ON : END
3240 IF K$="" OR F=1 THEN GOTO 3330
3250 REM all other keys -> emergency shut down
3260 CALL INIT
3270 FOR I=1 TO 4
3280 LET STA(I)=MOFF
3290 NEXT I
3300 LET Z=MOFF
3310 GOSUB 5050
3320 REM output to selected motor
3330 M=M(MOT)
3340 CALL M(Z)
3350 LOCATE 22,(5*MOT-2):COLOR H,S:PRINT"M";MOT
3360 S$=STA$(INT(Z/100))
3370 LOCATE 24,(5*MOT-2):PRINT S$;:COLOR S,H
3380 GOTO 2060
4000 REM
4010 REM *****
4020 REM *** set up of screen ***
4030 REM *****
4040 REM
4050 CLS
4060 PRINT
4070 PRINT" fischertechnik"
4080 PRINT" computing"
4090 PRINT
4100 COLOR H,S
4110 PRINT"
4120 PRINT" Diagnostic Program
4130 PRINT"
4140 COLOR S,H

```

```

4150 LOCATE 9,2
4160 PRINT"select motor: keys 1-4"
4170 PRINT" selected motor counter-cw (pulse): F1"
4180 PRINT" selected motor counter-cw (perm.): F3"
4190 PRINT" selected motor clockwise (perm.): F5"
4200 PRINT" selected motor clockwise (pulse): F7"
4210 PRINT" selected motor off: space bar"
4220 PRINT" all motors off : any other key"
4230 PRINT" end of program : x"
4240 PRINT
4250 PRINT" E1 E2 E3 E4 E5 E6 E7 E8 EX EY"
4260 PRINT
4270 RETURN
5000 REM
5010 REM *****
5020 REM *** erase display of motors ***
5030 REM *****
5040 REM
5050 LOCATE 22,3
5060 PRINT"M 1 M 2 M 3 M 4 "
5070 PRINT
5080 FOR I=1 TO 4
5090 S$=STA$(INT(STA(I)/100))
5100 PRINT" ";S$;
5110 NEXT
5120 RETURN

```

## Microsoft BASIC

The fischertechnik computing programs included on the supplied diskette are written in Microsoft BASIC. With a few exceptions containing commands for screen graphics, only the simplest version of BASIC is required. The programs documented in the fischertechnik computing programming instructions, however, have been written using the BASIC language of another computer. In general, we tried to avoid using computer-specific features in these programs, but this was not entirely possible, so you will have to search the programs printed in the instructions for modifications. The most important lines with changes have been marked with an asterisk in front of the line number. We will briefly summarize the differences here:

fischertechnik  
computing  
Programming Instructions

Microsoft  
BASIC

**SYS M1, CW**  
**SYS INIT**  
**PRINT CHR\$(147)**  
**GET A\$**

**CALL M1(MCW)**  
**CALL INIT**  
**CLS**  
**A\$=INKEY\$**

Cursor control is simplified in Microsoft BASIC, since the LOCATE command is available. Inverse text lines cannot be created using the PRINT command, but require the use of the COLOR command in each case. On the other hand, some PRINT commands are simplified by the use of the PRINT command with the format specification (PRINT USING). The convenient availability of a variety of commands in Microsoft BASIC may cause some conflicts when naming variables. In these cases, the variable names given in the programming instructions must be modified.

However, all necessary changes have been made in the programs on the diskette. The diskette programs may also deviate from the printed documentation in other respects where this was advantageous.

## Check List

Should you ever have a problem with the fischertechnik computing interface, and it does not work as you expect it to, then check the following points with the diagnostic program:

The diagnostic program displays 1 for E1 through E8, although no model is connected. – The interface has not been connected to the computer or to the power supply unit.

One of the inputs E1 through E8 shows the reverse effect from what you would expect when the pushbutton switch is actuated. – Opening and closing functions of the pushbutton switch have been reversed.

One of the inputs E1 through E8 always shows 0, although it is connected and has been actuated. – Check for a broken or improperly connected wire.

One of the inputs E1 through E8 always shows 1, even though no model is connected. – The input gate IC 4014 has probably been damaged by overvoltage (electrostatic discharge).

A motor output does not work. – Check for a broken or improperly connected wire.

A motor output only works in one direction of rotation. – The power stage of the motor output is defective.

The motor rotates slowly or works intermittently. – Either the power supply unit is overloaded by too many motors (use two mot 4 power supply units or the more powerful computing power supply unit) or the power supply unit has not been turned up all the way when using the adjustable output.

In case of defect, send the unit to your dealer or the company representing and importing fischertechnik.

## Technical Data

fischertechnik computing interface IBM Personal Computer for IBM Personal Computer and compatible computers, Part No. 30567.

Four outputs for connection of motors, lamps, electromagnets, etc. (M1 to M4). Polarity of output (direction of rotation) controllable.

Load capacity: 1 A continuous, 1.5 A peak.

Eight inputs for digital signals (E1 through E8). Due to internal wiring, the connection of electromagnetic devices (pushbutton switches, switches, relays) in positive logic and the connection of TTL outputs is also possible. Built-in overvoltage protection.

Two inputs for analog signals (EX and EY). Transmitters with resistances between 0 and 5 k $\Omega$ , e.g. potentiometers, photoresistors, etc., can be connected.

Circuitry for monitoring the stream of data. If no data signals are received from the IBM Personal Computer, the interface will disable all outputs after 0.5 seconds. The signals are stored in memory.

Circuitry for monitoring the software. Grave errors will trigger the guard circuit, in this case without delay.

The guard circuitry will also react to an insufficient supply of power, be it through overload or insufficient voltage from the power supply unit.

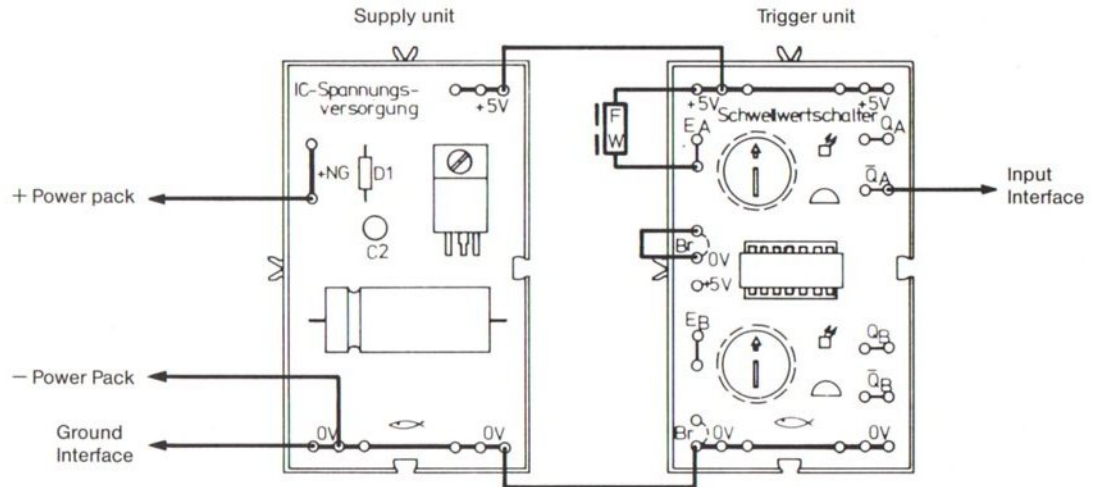
Software with interface control commands and sample programs for the fischertechnik computing models is supplied with the kit.

## Using the fischertechnik Electromechanics and Electronics Kits

The fischertechnik computing interface is compatible with the components and electronic devices of the above-mentioned kits. Other types of pushbutton switches and switches may be used in place of the mini pushbuttons used with the fischertechnik computing models, e.g. the large pushbutton or the pole changing switch, but also the reed contact or the switching contact of a relay. Be careful when using pushbuttons and switches which you have constructed yourself from articulated elements and springs – they could cause contact bounce. In such cases, we recommend having the program check the input several times and consider the result valid only if the same value appears twice in a row.

The analog inputs of the interface can be connected to sensors with an output resistance between 0 and 5 k $\Omega$ . The potentiometers from the fischertechnik computing kit may be used, of course. Other devices, such as the photoresistor, may be used as well.

The motor outputs of the interface feature high load capacity. Not only the mini motors, but also the S-motor and the N-motor can be controlled via the interface, whereby one lamp may be connected in parallel as a function indicator. Besides motors, the electromagnet and the relay RBII are also suitable. Signals from the electronic modules with integrated circuits of the TTL family (e.g. trigger unit) may also be fed to the interface inputs. First, however, the common point of reference must be established by connecting the ground circuit of the electronic module to the ground jack of the interface. Figure 2 shows the design of a light barrier. The trigger unit is used to adjust the reaction threshold of the light barrier.



## Operation of the Interface and the Interface Driver Program

If you use the fischertechnik computing software or write programs yourself according to the notes in the previous sections, most likely you will not need the information that follows. If, however, you intend to write the programs in a language other than BASIC, would like to speed them up through complex procedures in machine language, wish to extend the functions of the interface or simply want to glimpse behind the scenes, then the following information will most certainly be helpful. In this case, however, you should have a basic knowledge of machine language and digital electronics, since this is about the "bits and pieces".

The fischertechnik interface handles a number of tasks which we would like to discuss with the aid of the block diagram. On the left side you see the signals from and to the computer. Note how little they have in common with outputs M1 to M4 and inputs E1 through E8 and EX and EY. The reason for this is that the number of data lines available at the computer port is significantly lower than the number of lines required on the model side of the interface. This limited number of data lines must therefore be employed in such a way as to control all signals on the model side. The concept employed is that of multiple use of the data lines with the aid of shift registers. In this way, for example, only three data lines are required for controlling the output. A parallel connection scheme would have required eight data lines. Let's take a closer look at the output at connections M1 to M4. The data lines required are designated DATA OUT, CLOCK and LOAD OUT. If there is an output, the data for all four motors are transmitted in each case, i. e. a whole byte (a byte because each of the four motors requires two bits for controlling the direction of rotation). The motor outputs to which the signal does not apply are thus once again supplied with the current state which is buffered in the computer as an output word. For output, the bits of the output word are sequen-

tially (with the most significant bit first) fed to the DATA OUT line. When the signal at the CLOCK output goes from low to high, the bit is transferred to a shift register. Then the next bit at DATA OUT follows, and is likewise transferred to the shift register with the next CLOCK pulse. The previous bit has been shifted one position to the right in the shift register in order to make room for the subsequent bit. After a total of eight such data transfers, the whole output word has been transferred to the shift register. The bit first transferred has been shifted all the way to the right in the course of the data transfer. Thus far, the activity in the shift register has not had any effect on its outputs. The output amplifiers are not controlled directly by the shift register, but rather via an in-line storage register which is integrated in the shift register module. Only when the LOAD OUT output goes from low to high are the data transferred to the storage register. The timing of the signals is shown in the pulse diagram.

Whether the data are fed to the power amplifiers, however, depends on the enabling control of the memory module. The enabling circuit is controlled by a monostable multivibrator. This circuit generates an enabling signal with a duration of half a second if there is a pulse on the CLOCK line. We may assume that the power amplifiers receive a signal first since the data were just transferred with the aid of the CLOCK line. If no more data are transmitted within the next half second, however, the monostable multivibrator will flip back to the stable state and the enabling signal is removed. The monostable multivibrator, by the way, can be retriggered, i. e. the time of half a second is always calculated from the time of the last CLOCK pulse.

The monostable multivibrator also has an enabling input. The output to the amplifiers can be immediately inhibited via this input. On the fischertechnik interface this occurs when an invalid data pattern, which would command the connected motor to

simultaneously turn clockwise and counterclockwise, applies at the output of the storage register. We will proceed with the transfer of the digital signals to inputs E1 through E8. Basically the input is a reversal of the output process described above. The output signal LOAD IN causes the transfer of the data applying at the inputs to the input shift register. This always involves all eight inputs, even though only one of them is to be interrogated. When applying to the shift register, each pulse of the CLOCK line will cause the transfer of one bit on the input line DATA IN, the bit from E8 first and the one from E1 last. By testing this line, the computer can "collect" the bits and reassemble them into a data word. The desired bit is subsequently filtered out and transferred to the BASIC program.

Since the same CLOCK line is used for data transmission as for output, the digital input will also activate the monostable multivibrator, which controls the enabling signal for the output data. Malfunctioning of the output shift register caused by the multiple function of the CLOCK line is not to be expected since the current output data are not contained in the output shift register, but in the storage register. The former is controlled by the CLOCK pulses, unlike the latter, which only reacts to the LOAD OUT signal.

That leaves the analog inputs EX and EY. The potentiometers or other variable resistors are used as the timing element in two additional monostable multivibrator circuits. A low resistance value is converted to a short pulse, a high resistance value to a pulse with a long duration. The pulse itself is triggered by the starting signals TRIGGER-X and TRIGGER-Y (with negative logic), respectively, and then appears on the COUNT IN line. A machine language program determines the pulse duration by means of the number of loops which can be executed during the duration of the pulse. This number is fed back to the BASIC program which calls this function. You can see that there is no direct relationship between the



analog value and the angle position of the resistance of the potentiometer. The clock rate of the processor, however, is involved. The microprocessor of the AT, of course, can go through a greater number of loops during the duration of the pulse than the microprocessor of the XT. For this reason, the count is adapted to the required value range by shifting it to the right. There is a linear relationship between the number determined in the end and the resistance. If required, this value must be converted into angular degrees or resistance values by means of calibration.

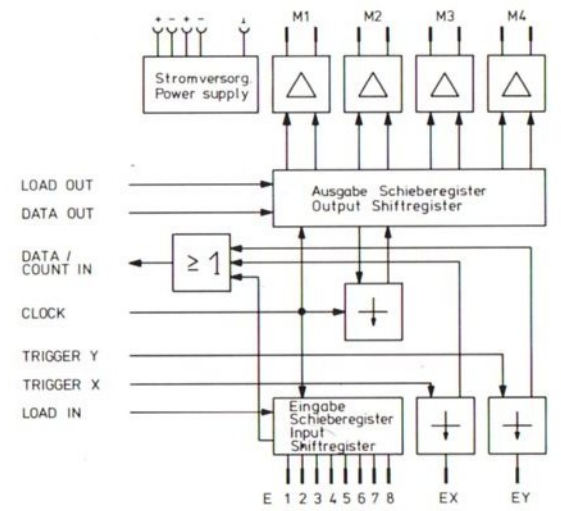
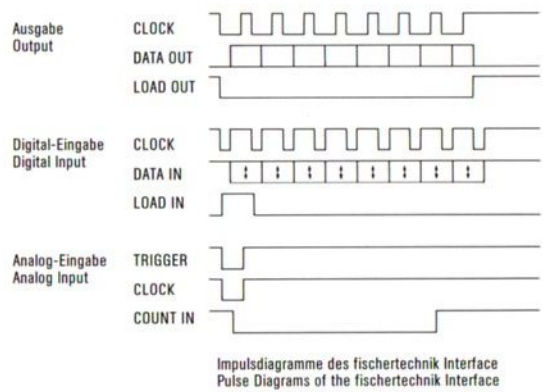
At this point, we will briefly review the connection between the interface and the PC. As you know from the section about connecting the interface, the computer's parallel printer port is used for this purpose. Out of the eight data lines of this interface, the lower six are used for the output signals discussed above. (Also refer to the following table.)

The input lines pose a slight problem in that the parallel printer interface has only one input line available applying to all PC's. This is the active signal from the printer, BUSY. There is no conflict, however, if all input lines are combined using an OR circuit. Since the machine language program "knows" which input function it has requested, it has the capability of interpreting the signals on this one input line correctly.

### Connecting the fischertechnik Interface to the Parallel Printer Port

Interface Signal	Printer Signal	Pin Number
LOAD OUT	Data bit 1	2
LOAD IN	Data bit 2	3
DATA OUT	Data bit 3	4
CLOCK	Data bit 4	5
TRIGGER-X	Data bit 5	6
TRIGGER-Y	Data bit 6	7
DATA/COUNT IN	Busy	11

The following machine language program controls the processes described above. For the parameter transfer to the machine language program and back refer to the description of the CALL command or the USR function, respectively, in the BASIC handbook of your PC. The following source code assumes a printer address of 03BC (hexadecimal) which is stored in FFFA and the subsequent memory addresses. The FISCHER program modifies this address if required due to the configuration of your computer. The NOP (no operation) commands at the end of the routine for the analog input serve as place holders. When the fischertechnik interface is used on PC's with faster microprocessors, the required number of right shift commands is inserted here by the FISCHER program.



```

*****
* IBM-PC Interface Driver Routines
*
* Version 1.2
* File INTERFAC.COM
*
* Copyright (C) fischerwerke,1986
* Control of the fischertechnik Interface
* output and input using the CALL-command
* and the USRO-function.
*****
* This program requires a Centronics parallel
* printer interface as I/O-port.
*
* The machine language program is located
* above the BASIC Ramtop and will not be
* destroyed by simple commands like
* NEW and LOAD.
* The program requires 256 bytes of RAM taken
* from the BASIC memory.
*
* Output control:
* CALL motornumber(operation)
* Motornumbers are M1, M2, M3, and M4.
* Operation is mcw (motor clockwise), mcow
* (motor counter-clockwise) and moff (motor off).
*
* CALL INIT
* Initializes the interface and switches off all
* motors.
*
* input control:
*
* Digital input commands:
* USR(digital_input_line)
* Digital_input_lines are E1, E2, E3, E4, E5,
* E6, E7, and E8.
*
* Analog input commands:
* USR(analog_input_line)
* Analog_input_lines are EX and EY.
*
* Connection of the interface to the printer port:
* interface printer-port pin-number
* 0V GND 17-25
* LOAD-OUT Data 1 2
* LOAD-IN Data 2 3
* DATA-OUT Data 3 4
* CLOCK Data 4 5
* TRIGGER-X Data 5 6
* TRIGGER-Y Data 6 7
* DATA/COUNT-IN -Busy 11
*****
* Routine for output control
* entry via CALL command
*****

```

```

:FF00 B80001 MOV BX,0100 :INIT
:FF03 FA CLI :disable interrupt
:FF04 EB24 JMP FF2A
:FF08 B003 MOV AL,03 :M1
:FF08 EBOA JMP FF14
:FF0A B00C MOV AL,0C :M2
:FF0C E006 JMP FF14
:FF0E B030 MOV AL,30 :M3
:FF10 E020 JMP FF14
:FF12 B0C0 MOV AL,C0 :M4
:FF14 FA CLI :disable interrupt
:FF15 BALEBEFF MOV BL,[FFBE] :get previous value
:FF19 08C3 OR BL,AL
:FF1B 55 PUSH BP
:FF1C 89E5 MOV BP,SP
:FF1E 8B7606 MOV SI,[BP+06] :get operation code

```

```

:FF21 8A24 MOV AH,[SI]
:FF23 5D POP BP
:FF24 B700 MOV BH,00
:FF26 20C4 AND AH,AL
:FF28 30E3 XOR BL,AH
:FF2A 881EBEFF MOV [FFBE],BL
*****
* Routine for interface control
* output control
* output bit pattern in BL
* uses AL, BL, CX, and DX
*****
:FF2E B90800 MOV CX,0008 :loop counter
:FF31 8B16BAFF MOV DX,[FFBA] :addr. printer port
:FF35 B030 RCL BL,1 :static bit pattern
:FF37 D0D3 RCL AL,1 :test output word
:FF39 7302 JNB FFFD :DATA-OUT low
:FF3B 0C04 OR AL,04 :DATA-OUT high
:FF3D EE OUT DX,AL :output to port
:FF3E 0C08 OR AL,08 :set CLOCK
:FF40 EE OUT DX,AL :output to port
:FF41 E2F2 LOOP FFF5 :end of loop
:FF43 B039 MOV AL,39 :LOAD-OUT high
:FF45 EE OUT DX,AL :output to port
:FF46 FB STI :enable interrupt
:FF47 80FF01 CMP BH,01 :test where you came from
:FF4A 7403 JZ FFAF
:FF4C CA0200 RETF 0002 :return from Mn(op)
:FF4F CB RETF :return from INIT

```

```

*****
* Routine for interface control
* input control
* entry via USRO function
*****

```

```

:FF50 FA CLI :disable interrupt
:FF51 8A07 MOV AL,[BX] :get argument of USR
:FF53 8B16BAFF MOV DX,[FFBA] :addr. printer port
:FF57 3CA0 CMP AL,A0 :analog input?
:FF59 743C JZ FFF7
:FF5B 3C90 CMP AL,90 :analog input?
:FF5D 7438 JZ FFF7 :go to FFF7
:FF5F A2B8FF MOV [FFBF],AL :save input bit mask
*****
* interface control
* uses AL, CX, and DX
*****
:FF62 B90800 MOV CX,0008 :loop counter
:FF65 B032 MOV AL,32 :LOAD-IN high
:FF67 EE OUT DX,AL :output to port
:FF68 0C08 OR AL,08 :set CLOCK
:FF6A EE OUT DX,AL :output to port
:FF6B D0D4 RCL AH,1 :clear LSBit
:FF6D 8B16BCFF MOV DX,[FFBC] :addr. busy-line
:FF6F EC IN AL,DX :read DATA-IN
:FF72 2480 AND AL,80 :mask DATA-IN
:FF74 3C80 CMP AL,80 :test DATA-IN
:FF76 7503 JNZ FFFB :DATA-IN high (0)
:FF78 90CC01 OR AH,01 :DATA-IN low (1)
:FF7B F6 CLC
:FF7C F6 CLC
:FF7E 8B16BAFF MOV DX,[FFBA] :addr. printer port
:FF80 B030 MOV AL,30 :static bit pattern
:FF82 EE OUT DX,AL :output to port
:FF83 B038 MOV AL,38 :set CLOCK
:FF85 EE OUT DX,AL :output to port
:FF86 E2E3 LOOP FFB8 :end of loop
:FF88 A0B8FF MOV AL,[FFBF] :get bit mask
:FF8B 20E0 AND AL,AH :mask bit
:FF8D 3C00 CMP AL,00 :test bit
:FF8F 7402 JZ FFF9
:FF91 B001 MOV AL,01 :return 1
:FF93 8807 MOV [BX],AL :deposit in FAC
:FF95 FB STI :enable interrupt
:FF96 CB RETF :return from USRO

```

```

*****
* analog input
* The argument of the USRO function either being
* 90 (input EY) or A0 (input EX) will lead the
* program flow to this sub-routine.
* Uses AX, CX, and DX.
*****
:FF97 B9FF03 MOV CX,03FF :loop counter
:FF9A EE OUT DX,AL :trigger one-shot
:FF9B B038 MOV AL,38 :set CLOCK
:FF9D EE OUT DX,AL :output to port
:FF9E 8B16BCFF MOV DX,[FFBC] :addr. busy line
:FFA2 EC IN AL,DX :read COUNT-IN
:FFA3 D0D0 RCL AL,1 :test COUNT-IN
:FFA5 7303 JNB FFA9 :COUNT-IN low
:FFA7 E2F9 LOOP FFA2 :continue testing
:FFA9 B8FF03 MOV AX,03FF :eval. analog value
:FFAC 29C8 SUB AX,CX :as 1023-CX
:FFAE 90 NOP :space for SAR AX,1
:FFAF 90 NOP
:FFB0 90 NOP :space for SAR AX,1
:FFB1 90 NOP
:FFB2 8807 MOV [BX],AL :deposit in FAC
:FFB4 867010 MOV [BX+1],AH
:FFB7 FB STI :enable interrupt
:FFB8 CB RETF :return from USRO

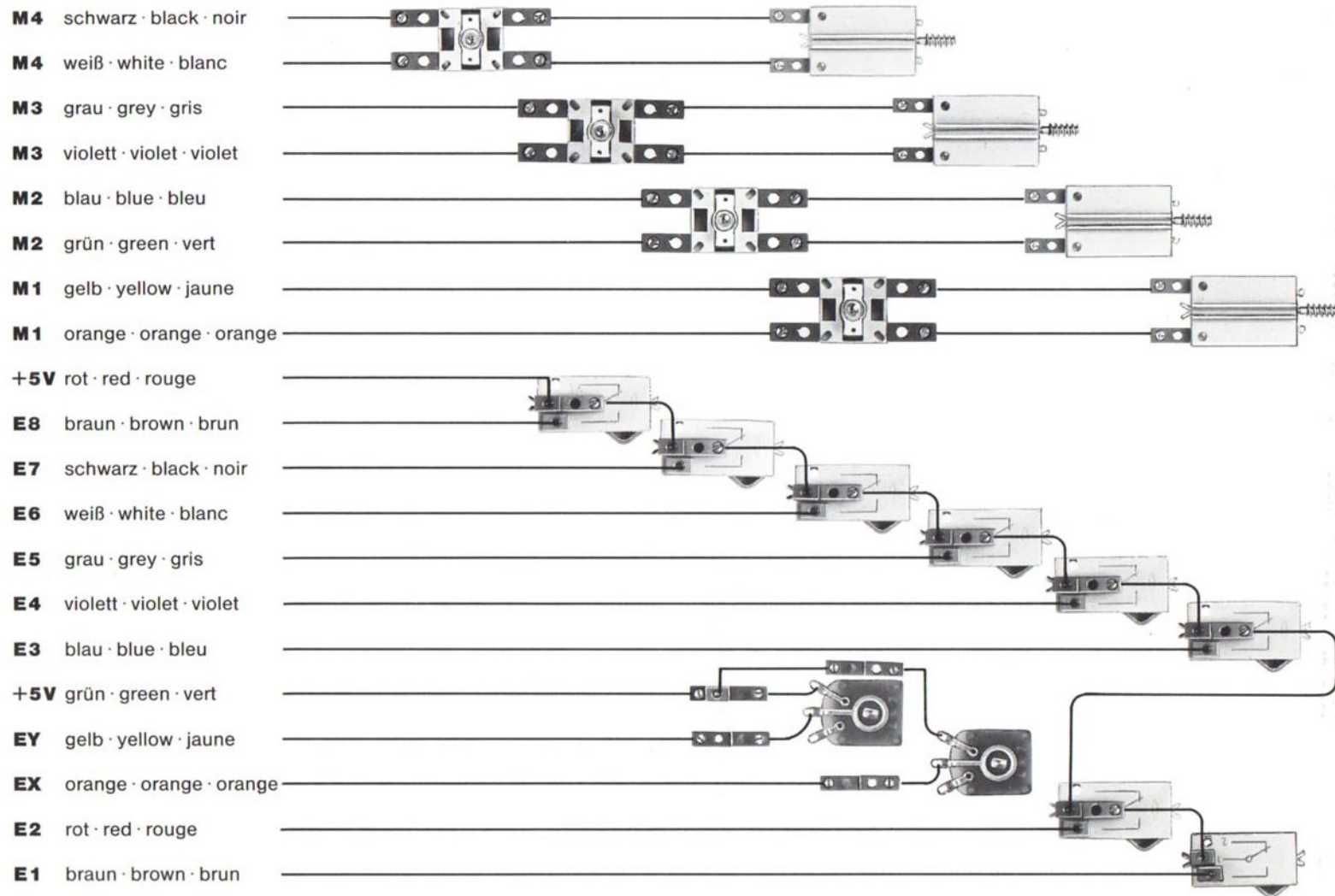
```

```

:FFBA address of printer port
:FFBC address of busy line
:FFBE memory location of output variable
:FFBF memory location of bit mask

```

## Verdrahtungsplan der Interface Ein- und Ausgänge · Circuit layout of the Interface Inputs and Outputs



**fischertechnik computing System**

