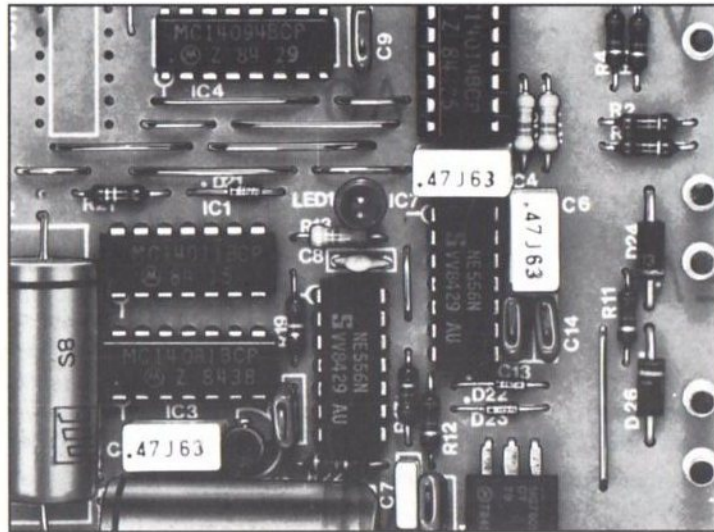
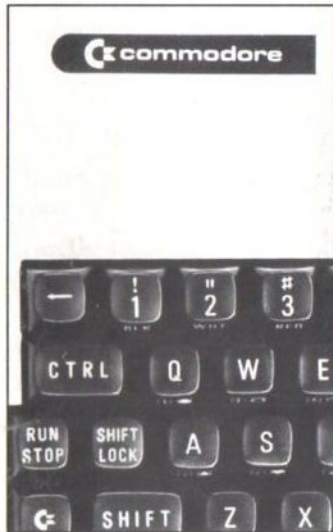


# fischertechnik<sup>®</sup>

## COMPUTING

Interface für Commodore Computer



## Inhalt

|   |    |
|---|----|
| Anmerkung   | 4  |
| Einführung  | 5  |
| Anschluß des Interface  | 6  |
| fischertechnik COMPUTING Software                               | 8  |
| Diagnoseprogramm  | 13 |
| Anschluß von zwei Interface                                     | 15 |
| Benutzung von fischertechnik Elektromechanik und Elektronik     | 16 |
| Funktionsweise des Interface und des Interface-Treiberprogramms | 17 |
| Checkliste  | 22 |
| Technische Daten  | 23 |
| Verdrahtungsplan der Interface Ein- und Ausgänge                | 44 |

Diese Anleitung beschreibt die Benutzung des fischertechnik COMPUTING Interface am Commodore C64, C64-II, SX64, C128 und C128-D.

Die Benutzung des fischertechnik COMPUTING Interfaces an den Computern der Commodore PC-Serie wird in der Anleitung "Interface für IBM Personal Computer" beschrieben.

Die Benutzung des fischertechnik COMPUTING Interfaces an den Computern der Commodore Amiga-Serie wird in der Anleitung "Interface für Commodore Amiga Computer" beschrieben.

2. erweiterte Auflage

© 1988 fischerwerke Artur Fischer GmbH & Co. KG

## Table of Contents

|  |    |
|--|----|
| Important Notice   | 24 |
| Introduction   | 25 |
| Connecting the Interface                                       | 26 |
| fischertechnik COMPUTING Software                              | 28 |
| The Diagnostic Program   | 33 |
| Control of two Interfaces                                      | 35 |
| Using the fischertechnik Electromechanics and Electronics Kits | 36 |
| Operation of the Interface and the Interface Driver Program    | 37 |
| Check List   | 42 |
| Technical Data   | 43 |
| Circuit Layout of the Interface Inputs and Outputs             | 44 |

This manual describes the operation of the fischertechnik COMPUTING Interface using the Commodore computers C64, C64-II, SX64, C128, and C128-D.

Using the fischertechnik COMPUTING interface with computers of Commodore's PC line is described in the manual "Interface for IBM Personal Computer".

Using the fischertechnik COMPUTING interface with computers of Commodore's Amiga line is described in the manual "Interface for Commodore Amiga computers".

2<sup>nd</sup> extended edition

## Bitte kontrollieren Sie die Vollständigkeit Ihres fischertechnik COMPUTING Systems

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

Für den... benötigen Sie neben dem Bausatz noch...

- fischertechnik COMPUTING Baukasten**
- das fischertechnik Interface
  - den Adapter für den Userport (Gutschein)
  - die Diskette COMPUTING (Gutschein)
  - das fischertechnik Netzgerät COMPUTING

- fischertechnik COMPUTING Trainingsroboter**
- das fischertechnik Interface
  - den Adapter für den Userport (Gutschein)
  - die Diskette Trainingsroboter (Gutschein)
  - das fischertechnik Netzgerät COMPUTING

- fischertechnik COMPUTING Plotter/Scanner**
- das fischertechnik Interface
  - den Adapter für den Userport (Gutschein)
  - die Diskette Plotter/Scanner (Gutschein)
  - das fischertechnik Netzgerät COMPUTING

- fischertechnik COMPUTING EXPERIMENTAL**
- den Adapter für den Userport (Gutschein)
  - die Diskette EXPERIMENTAL (Gutschein)

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

## Please check the completeness of your fischertechnik COMPUTING system.

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

Using the... you will require additionally...

- fischertechnik COMPUTING robotics kit**
- the fischertechnik interface
  - the adapter for the user port (coupon)
  - the diskette COMPUTING (coupon)
  - an adequate power supply (see text)

- fischertechnik COMPUTING Training Robot**
- the fischertechnik interface
  - the adapter for the user port (coupon)
  - the diskette Training Robot (coupon)
  - an adequate power supply (see text)

- fischertechnik COMPUTING Plotter/Scanner**
- the fischertechnik interface
  - the adapter for the user port (coupon)
  - the diskette Plotter/Scanner (coupon)
  - an adequate power supply (see text)

- fischertechnik COMPUTING EXPERIMENTAL**
- the adapter for the user port (coupon)
  - the diskette EXPERIMENTAL (coupon)

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

## Anmerkung

Es wurden alle erdenklichen Maßnahmen getroffen, um die Richtigkeit dieser Produkt-Dokumentation zu gewährleisten. Da jedoch die fischerwerke Artur Fischer GmbH & Co. KG ständig an der Verbesserung ihrer Produkte arbeiten, können wir keine Garantie für die Vollständigkeit und Richtigkeit dieser Dokumentation seit ihrem Erscheinen übernehmen.

Amiga, Commodore 64, Commodore 128, Commodore VC20 und CBM sind Warenzeichen der Commodore Electronics Limited.

Apple ist ein eingetragenes Warenzeichen der Apple Computer Inc.

Atari ist ein eingetragenes Warenzeichen der Atari Corporation.

Centronics ist ein eingetragenes Warenzeichen der Data Computer Corporation.

CPC 464, CPC 664 und CPC 6128 sind Warenzeichen von Amstrad Consumer Electronics plc.

CP/M ist ein Warenzeichen von Digital Research Inc.

IBM und IBM PC sind eingetragene Warenzeichen der International Business Machines Corporation.

### **Bescheinigung des Herstellers/Importeurs**

Hiermit wird bescheinigt, daß das

#### **fischertechnik COMPUTING Interface**

(Gerät, Typ, Bezeichnung)

in Übereinstimmung mit den Bestimmungen der AmtsblVfg 1046/1984

funkentstört ist.

Der Deutschen Bundespost wurde das Inverkehrbringen dieses Gerätes angezeigt und die Berechtigung zur Überprüfung der Serie auf Einhaltung der Bestimmungen eingeräumt.

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

## Das 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 die fischertechnik computing Bausätze (Baukasten COMPUTING, Trainingsroboter, Plotter/Scanner, EXPERIMENTAL) 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 computer-gesteuerten Modellen zu haben.

Was braucht man zum Steuern der Modelle? Zunächst einmal das fischertechnik Modell zur Ausführung der Abläufe. Dann einen Commodore Computer, 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!", werden 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.
- Darüberhinaus 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. Versuchen Sie es einmal, Sie werden viel Spaß haben.

Ihre  
Artur und Klaus Fischer

## Anschluß des Interface

Das fischertechnik COMPUTING Interface Art. Nr. 30 566 paßt, neben einer Reihe anderer Computer, an die meisten Commodore Computer der 8-Bit-Technologie. Bedingung ist, daß die Computer mit einer Benutzerschnittstelle ausgestattet sind. Dies trifft insbesondere für die Computer C64, C64-II, SX64, C128 und C128-D zu. Auch die hier nicht genannten VC20, PET und CBM Computer sowie Plus-4 können mit dem Interface verbunden werden, jedoch besteht keine Unterstützung dieses Anschlusses durch fischertechnik COMPUTING Software. Prüfen Sie bei diesen Computern, ob für die von Ihnen gewünschten fischertechnik COMPUTING Baukasten evt. noch ältere fischertechnik COMPUTING Software zur Verfügung steht oder ob Sie anhand der im folgenden beschriebenen Dokumentation selbst in der Lage sind, die erforderliche Software zu erstellen.

Auf keinen Fall benutzt werden können die Computer 116 und C16, denn sie besitzen keine Benutzerschnittstelle.

Wenn Sie einen Commodore PC oder einen Commodore Amiga besitzen, eignet sich die hier beschriebene Anschlußmethode, Software und Vorgehensweise nicht. Sie benötigen dann die Adaption und die

Anleitung für IBM-PC bzw. Commodore Amiga.

Der Anschluß des fischertechnik COMPUTING Interface erfolgt an die Benutzer E/A Schnittstelle (Userport) des Computers. 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 die Benutzer E/A Schnittstelle. Es ist die ganz rechte Leiste, siehe Abbildung.
- 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, z.B. eines Wasserhahns.
- Legen Sie sich nun das Interface zu recht.
- Mit dieser Anleitung haben Sie auch den Steckadapter passend zu Ihrem Computer geschickt bekommen.
- Der Steckadapter wird zwischen Interface und Computer eingefügt. Auf der

einen Seite trägt er einen Stecker, der genau zu Ihrem Interfacekabel paßt, am anderen Ende einen Stecker, der zu der Benutzer E/A Schnittstelle paßt. Das Flachbandkabel des Interfaces wird in die zwanzigpolige Steckerleiste des Steckadapters eingesteckt. Eine Aussparung am Gehäuse und eine Nase am Stecker gewährleisten, daß beides richtig herum zusammengesteckt wird.

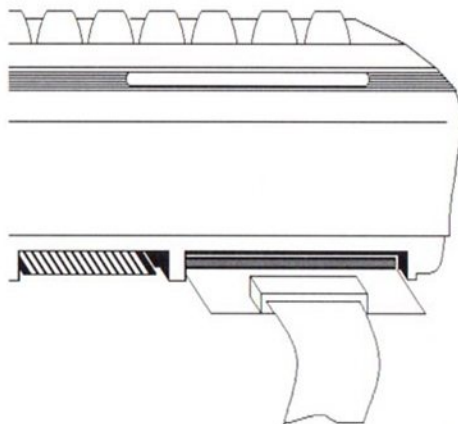
- Der andere Stecker des Adapters wird auf den Benutzer E/A Anschluß aufgesteckt. Der Stecker ist gegen Verpolung geschützt, da die Trennrippe in dem Stecker zu dem Schlitz in der Verbindungsleiste passen muß.
- Schließen Sie das Interface an das Netzgerät an. Das Interface erwartet ungesieberte Gleichspannung zwischen 6 und 10 Volt. Beim Einsatz von stabiler Gleichspannung sind ca. 8 Volt (belastbar bis ca. 2 Ampere) einzustellen. Um ganz sicher zu gehen, sollten Sie das fischertechnik Netzgerät COMPUTING verwenden. Stecken Sie den Kabelausgang des Netzgeräts mit dem roten Flachstecker in die mit + gekennzeichnete Buchse des Interface und den Kabelausgang mit grünem Flach-

- stecker 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 kann auch das 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 Netzgerät COMPUTING 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 zweiadrigte Leitung und Flachstecker zur Herstellung des Anschlußkabels.
  - Verbinden Sie das fischertechnik COMPUTING Modell mit dem Interface. Hierzu dient das den fischertechnik COMPUTING Bausätzen beigefügte zwanzigadrigte Flachbandkabel. Dieses Kabel ist auch als Einzelteil 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 keine Geräte benutzen, die auch an diesen Anschluß eingesteckt werden. Hierzu gehört das V24/RS232-Modul und das Telefonmodem.

Noch einmal zur Erinnerung:

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

Doch nun genug von der Hardware, im nächsten Abschnitt wollen wir uns die fischertechnik COMPUTING Software vornehmen.



## fischertechnik COMPUTING Software

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 Maschinsprache 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 jeweiligen fischertechnik COMPUTING Baukastens sind die passenden Steuerprogramme enthalten. Füllen Sie den Gutschein aus und schicken Sie ihn an die fischerwerke - die zu dem Baukasten passende Diskette kommt postwendend. Die vorliegende Anleitung sagt Ihnen zudem, wie Sie solche Steuerprogramme selbst schreiben.

Um sicher zu sein, daß Sie nicht bei einem versehentlichen Löschen oder einer Beschädigung der Diskette ohne Software dastehen, sollten Sie sich von der fischertechnik COMPUTING Diskette eine Kopie anfertigen. Benutzen Sie hierfür das Programm COPY/ALL, das mit der Systemdis-

kette des Diskettenlaufwerks mitgeliefert wird, oder ein anderes Kopierprogramm. Verfahren Sie exakt nach den Angaben des Programms. Wenn Sie die Kopie vorliegen haben, verstauen Sie die fischertechnik COMPUTING Original-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 wohlgerneht 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 anschließend den Computer ein.

Wenn Sie einen C128 oder C128-D besitzen, so schalten Sie den Computer jetzt in den C64-Modus. Geben Sie folgendes

Kommando ein:

**GO64**

Beantworten Sie die Frage "ARE YOU SURE?" mit

**Y**

Danach erscheint das Einschaltbild des C64 und Sie können alles folgend Beschriebene und die gesamte fischertechnik COMPUTING Software des C64 unverändert benutzen.

Nach dem Systemstart legen Sie die fischertechnik-Diskette in das Diskettenlaufwerk ein. Rufen Sie das Programm FISCHER mit folgendem Kommando auf:

**LOAD"0:\*",8**

Das Programm FISCHER wird geladen. Geben Sie nun das Kommando für den Programmstart ein:

**RUN**

Nach dem Start des Programms FISCHER erhalten Sie Auskunft über Hard- und Software Ihres fischertechnik COMPUTING Baukastens. Das Programm FISCHER nennt Ihnen ggf. auch Ergänzungen zu der Anleitung und neueste Informationen zu der fischertechnik COMPUTING Software und den Modellen.

Laden Sie nun das Grundprogramm GRUNDPR. Da dieses Programm in speziellen Versionen für die verschiedenen



Baukästen vorliegt, müssen Sie einen der beiden nachfolgend beschriebenen Befehle benutzen:

**LOAD"GRUNDPR.BAS",8**

oder

**LOAD"GRUNDPR.64",8**

oder

**LOAD"ROBOTSYS.64",8**

Schauen Sie ggf. in dem Inhaltsverzeichnis der Diskette nach, welches Grundprogramm in Ihrem Fall zutrifft:

**LOAD"\$",8**

**LIST**

Das Grundprogramm richtet alles ein, was Sie zum Datenaustausch mit dem Interface benötigen. Das Grundprogramm ist auch Bestandteil aller folgenden Beispielprogramme.

Das Grundprogramm enthält die eingangs erwähnten Detailkenntnisse über die Ein- und Ausgabebausteine des Computers. Durch seine Benutzung vereinfacht sich die Steuerung des Interfaces ganz wesentlich. Wir wollen an dieser Stelle anmerken, daß die Grundprogramme der verschiedenen Disketten durchaus unterschiedlich sind. Je nach Thema des Baukastens enthalten sie spezielle Kommandos: Roboterbefehle für den Trainingsroboter, Meßwerterfassungsbefehle für EXPERIMENTAL etc. Sie

dürfen somit die Grundprogramme der Disketten nicht austauschen.

Was wir im folgenden beschreiben, ist ein Grundgerüst an Befehlen, wie es in jedem Grundprogramm vorkommt. Starten Sie hierzu das Grundprogramm:

**RUN**

Nach Durchlauf durch das Programm wird sich der Computer wieder mit seinem Bereitzeichen melden.

Äußerlich scheint sich nichts geändert zu haben. Dennoch besitzt Ihr Commodore nun einige neue Befehle, die vorher in dieser Form nicht im BASIC enthalten waren. Diese Befehle sind genau auf Ihren Commodore Computer und das fischertechnik COMPUTING Interface abgestimmt. Die Befehle benutzen das SYS-Kommando und die USR-Funktion für den Aufruf von Maschinenprogrammen. Das Grundprogramm selbst ist in BASIC geschrieben. Es legt in dem Speicher des Computers ein Maschinenprogramm (Interface Treiber) ab, das es aus DATA-Zeilen oder von Diskette einliest. Außerdem erzeugt es im Variablenspeicher des Computers die Parameter jener Maschinenprogrammaufrufe (M1, M2, M3, M4, EX, EY, E1 ... RECHTS, LINKS, AUS).

Sie brauchen daher nur noch die folgend

beschriebenen BASIC-Befehle zu beherrschen:

Der Motorausgang M1 wird gesteuert mit:

**SYS M1,EIN**

**SYS M1,AUS**

**SYS M1,RECHTS**

**SYS 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 +5V verbunden ist. Sonst zeigt USR(E1) den Wert 0. Entsprechend erhält man mit USR(E2) ... USR(E8) die Zustände der übrigen Digitaleingänge.

Die Analogeingänge EX und EY werden über je einen Sensor, z.B. ein Potentiometer (4.7 kΩ), mit +5V verbunden. Die Funktionen

**USR(EX)**

**USR(EY)**

führen dann einen Wert aus dem Bereich

15 bis 200, je nach Stellung der Potentiometer. Obige Grenzwerte können auch leicht anders ausfallen, je nach Streuung der Bauelemente.

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

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

aufruft, die Bewegung des Roboters genau verfolgen.

Wenn kein Sensor an den Eingang EX bzw. EY angeschlossen ist, wirkt sich aus, daß in der Interfaceschaltung ein 100 k $\Omega$  Widerstand parallel zu dem Eingang geschaltet ist. Der Funktionswert beträgt dann zwischen 4000 und 5000. Wir empfehlen, die Analogeingänge nie unbeschaltet zu lassen, wenn sie unbenutzt sind. Sie werden dann am besten direkt mit der Referenzspannung +5V verbunden; dies entspricht einem Widerstand von 0  $\Omega$ .

Der letzte der neuen Befehle ist

#### **SYS INIT**

Dieser wird benutzt, um das Interface in einen wohldefinierten Anfangszustand zu versetzen. Er kann auch benutzt werden,

wenn alle Motorkanäle mit einem Male abgeschaltet werden sollen.

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:

#### **SYS M1,EIN**

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

Doch zunächst interessiert uns auch die Frage, wieso der Motor wieder stehenblieb. Hatten wir ihn nicht richtig eingeschaltet? Gibt es zum Ausschalten nicht, wie oben beschrieben, einen eigenen Befehl? Nun, der Motor ist zwar stehen geblieben, 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 stehen bleiben wird. Auch dann, wenn Sie mit dem Tastendruck STOP den Programmlauf unterbrechen. Und wenn es dann wieder weitergeht (z.B. mit dem Kommando CONT), so wird mit dem ersten Befehl das Interface wieder "aufgeweckt" und hat keinen der Motoren vergessen. Der Ablauf kann fortgesetzt werden, als wäre 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 allein der Spannungsanzeige sondern auch der

Betriebsanzeige der Datenübertragung. 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 vorher 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 k $\Omega$  wie es z.B. im fischertechnik COMPUTING Baukasten enthalten ist, 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 15 und ca. 200 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 oder einem ähnlichen 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 anschließend 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 Zeilennummern bis 500. Geben Sie daher ein:

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

Es dauert einen ggf. kurzen Moment, bis das Maschinenprogramm geladen ist. 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 15 und 200 liegen. Zum Benden des Pro-

gramms müssen Sie die Stop-Taste 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, ein Maschinenprogramm im Speicher des Computers abzulegen. Das Maschinenprogramm liegt entweder als direkt ladbares Programm auf der fischertechnik-Diskette vor oder es werden die DATA-Zeilen im Grundprogramm gelesen, anhand einer Prüfsumme kontrolliert und im Speicher abgelegt. Neben dem Maschinenprogramm selbst werden noch die genormten Parameter und Einsprungadressen INIT, M1, M2, M3, M4, E1, E2, E3, E4, E5, E6, E7, E8, EX, EY, EIN, RECHTS, LINKS und AUS gesetzt.

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. Dabei beachten Sie bitte, daß Commodore BASIC Variablen

nur anhand der ersten beiden Zeichen im Namen unterscheidet. Namen wie

**EXTRA, M114, EINMAL, LIFT, REIN ...** sind deshalb ebenfalls verboten. Die Einschränkung betrifft jedoch nicht Variablen anderer Typs (Strings, Real), so daß

**EINGABE\$, M1%, AUTO\$ ...** durchaus vorkommen dürfen.

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 Motorausgänge. Dies wird durch den Befehl

**500 SYS 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.

Nachstehend drucken wir das Grundprogramm GRUNDPR.64 ab, mit dem die grundlegenden Experimente durchgeführt werden können.

```
1 PRINT CHR$(147)
5 PRINT"GRUNDPROGRAMM WIRD GELADEN"
10 REM INTERFACE PROGRAMM FUER COMMODORE 64/128
20 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
30 REM AUFRUF DES PROGRAMMS MIT
40 REM SYS M1,EIN SYS M1,AUS
50 REM SYS M1,RECHTS SYS M1,LINKS
60 REM USR(E1) USR(EX) USR(EY)
70 REM M1 BIS M4 SIND MOTORANSTEUERUNGEN
80 REM E1 BIS E8 SIND DIGITALEINGAENGE
90 REM EX UND EY SIND ANALOGEINGAENGE
100 DATA 52736,169,0,240,38,169,3,208,10,53573
110 DATA 169,12,208,6,169,48,208,2,54395
120 DATA 169,192,120,133,255,32,253,174,55723
130 DATA 165,254,5,255,133,254,32,158,56979
140 DATA 183,138,37,255,133,255,165,254,58399
150 DATA 69,255,133,254,168,169,63,141,59651
160 DATA 3,221,162,8,169,48,6,254,60522
170 DATA 144,2,9,4,141,1,221,9,61053
180 DATA 8,141,1,221,202,208,237,169,62240
190 DATA 57,141,1,221,132,254,88,96,63230
200 DATA 120,32,161,183,224,162,240,51,64403
210 DATA 224,146,240,47,134,255,169,50,65668
220 DATA 141,1,221,9,8,141,1,221,66411
230 DATA 162,8,10,44,1,221,16,2,66875
240 DATA 9,1,160,48,140,1,221,160,67615
250 DATA 56,140,1,221,202,208,235,37,68715
```

```
260 DATA 255,168,240,2,160,1,32,162,69735
270 DATA 179,88,96,169,255,141,4,221,70888
280 DATA 141,5,221,169,185,141,14,221,71985
290 DATA 142,1,221,162,58,142,1,221,72933
300 DATA 173,4,221,162,3,202,208,253,74159
310 DATA 56,237,4,221,208,242,162,56,75345
320 DATA 142,1,221,56,169,255,237,4,76430
330 DATA 221,168,169,255,237,5,221,32,77738
340 DATA 145,179,76,137,206,0,0,0,78481
350 DATA 1,2,4,8,16,32,64,128,78736
360 DATA 162,146,255,170,85,85,80,206,79925
370 READ INIT : M1=INIT
380 FOR M3=0 TO 24 : FOR M2=0 TO 7
390 READ M4 : POKE INIT+M3*8+M2,M4
400 M1-M1+M4 : NEXT
410 READ M4 : IF M1<>M4 THEN PRINT"DATAFEHLER IN ZEILE";M3*10+100:END
420 NEXT
430 READ E1,E2,E3,E4,E5,E6,E7,E8
440 M1=M1+E1+E2+E3+E4+E5+E6+E7+E8
450 READ M4 : IF M1<>M4 THEN PRINT"DATAFEHLER IN ZEILE 350" : END
460 READ EX,EY,AUS,LINKS,RECHTS,EIN,M2,M3
470 M1=M1+EX+EY+AUS+LINKS+RECHTS+EIN+M2+M3
480 READ M4 : IF M1<>M4 THEN PRINT"DATAFEHLER IN ZEILE 360" : END
485 POKE785,M2 : POKE786,M3
490 M1=INIT+4 : M2=M1+4 : M3=M2+4 : M4=M3+4
500 SYS INIT
```

## Das Diagnoseprogramm

Wenn Sie ein fischertechnik COMPUTING Modell aufgebaut haben, werden Sie vielleicht die Erfahrung machen, das nicht alles so läuft, wie Sie sich das vorgestellt haben. Wen wundert das bei dieser großen Zahl von Leitungen, die zwischen Modell und Interface hin und her laufen. 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 den fischertechnik COMPUTING Disketten als DIAGNOSE.64 oder 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.

Die Ausgänge des Interface können Sie ebenfalls steuern. Beachten Sie dazu die Bildschirmanzeige des Programms. Wählen Sie einen Ausgang des Interface mit den Zahlentasten aus. Der angewählte Ausgang wird auf dem Bildschirm invers angezeigt. Den Ausgang können Sie im

Programm DIAGNOSE.64 nun einschalten (Rechts- und Linkslauf) und ausschalten. Dafür werden die Buchstabentasten R, L und A benutzt. Die andere Varianten des Diagnoseprogramms, DIAGNOSE.BAS, verwendet die Funktionstasten: in diesem Fall können Sie zwischen fünf Betriebsarten wählen. Mit F1 wird der angewählte Motor eingeschaltet, jedoch nur solange, wie Sie die Funktionstaste drücken. Gleiches gilt für F7, jedoch wird die andere Drehrichtung angewählt. Die dazwischenliegenden Funktionstasten F3 und F5 schalten den Motor dauerhaft ein. Er muß dann durch Betätigung der Leertaste ausdrücklich wieder abgeschaltet werden. Die erstgenannten Funktionstasten werden Sie benutzen, wenn Sie einen Roboterarm fein positionieren wollen. Die zuletztgenannten Tasten dienen dazu, einen Ausgang auch dann eingeschaltet zu lassen, wenn man zu einem anderen Motor wechselt. Dies ist der Fall, wenn z.B. ein Elektromagnet als Greifer eines Roboters ein Eisenteil festhalten soll, selbst wenn der Roboter zwischen durch geschwenkt wird.

Mit dem Diagnoseprogramm 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.

Mit Tastendruck auf den Buchstaben C können Sie alle Motoren abschalten und mit X das Programm verlassen.

## Diagnoseprogramm für C64

```
500 SYS INIT
510 REM
520 REM FISCHERTECHNIK COMPUTING
530 REM
540 REM"DIAGNOSEPROGRAMM
550 REM"DATEI: DIAGNOSE.BAS
560 REM"COPYRIGHT (C) FISCHERWERKE 1986
570 REM
580 REM
590 REM"FUNKTION:
600 REM
610 REM"MIT DEM PROGRAMM WERDEN SAEMTLICHE
620 REM"FUNKTIONEN DER MODELLE UEBERPRUEFT.
630 REM"ALLE EINGAENGE WERDEN ANGEZEIGT.
640 REM"ALLE AUSGAENGE WERDEN UEBER DIE
650 REM"COMPUTERTASTATUR ANGESTEUERT.
660 LET VTS="(HOME){24*DOWN}"
1000 REM
1010 REM *****
1020 REM"*** STEUERUNG VORBEREITEN ***
1030 REM *****
1040 REM
1050 DIM STA(4),STAS(2):REM"STATUS DER 4 MOTOREN
1060 DIM M(4):REM"ADRESSEN FUER MOTOR 1-4
1070 GOSUB 4050 :REM"BILDAUFBAU
1080 REM"ALLE MOTOREN AUS
1090 FOR I=1 TO 4
1100 LET STA(I)=AUS
1110 NEXT I
1120 REM"STEUERZEICHEN
1130 LET SP$(CHR$(32) :REM"LEERTASTE
1140 LET F1$(CHR$(133) :REM"F1
1150 LET F3$(CHR$(134) :REM"F3
1160 LET F5$(CHR$(135) :REM"F5
1170 LET F7$(CHR$(136) :REM"F7
1180 REM"TEXTE FUER STATUSANZEIGE
1190 STAS(0)="RE."
1200 STAS(1)="LI."
1210 STAS(2)="AUS"
1220 LET M(1)=M1:M(2)=M2:M(3)=M3:M(4)=M4
1230 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 LOESCHEN
2060 PRINT LEFT$(VTS,20);
2070 PRINT TAB(2);USR(E1);
2080 PRINT TAB(5);USR(E2);
2090 PRINT TAB(8);USR(E3);
2100 PRINT TAB(11);USR(E4);
2110 PRINT TAB(14);USR(E5);
2120 PRINT TAB(17);USR(E6);
2130 PRINT TAB(20);USR(E7);
2140 PRINT TAB(23);USR(E8);
2150 IF X THEN PRINT TAB(28);RIGHT$( " "+STR$(USR(EX)),3);
2160 IF Y THEN PRINT TAB(34);RIGHT$( " "+STR$(USR(EY)),3)
3000 REM
3010 REM *****
3020 REM"*** STEUERUNG DER AUSGABE ***
3030 REM *****
3040 REM
3050 GET K$
3060 LET F=0 :REM"FLAGGE FUER GUELTIGES KOMMANDO
3070 LET 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$="X"THEN SYS INIT:PRINT CHR$(147):END
3160 IF K$="" OR F=1 THEN GOTO 3260
3180 REM"ALLE ANDEREN TASTEN -> NOT-AUS
3190 SYS INIT
3200 FOR I=1 TO 4
3210 LET STA(I)=AUS
3220 NEXT I
3230 LET ZU = AUS
3240 GOSUB 5050
3250 REM"AUSGABE AN GEWAELHTEN MOTOR
3260 SYS M(MOT),ZU
3270 PRINT LEFT$(VTS,22);
3280 PRINT TAB(5*MOT-3);"{RVS-ON}M";MOT;"{4*LEFT}";
3290 PRINT"(2*DOWN)";
3300 LET SS=STAS(INT(ZU/100))
3310 PRINT SS;
3320 PRINT "{RVS-OFF}";
3330 GOTO 2060
4000 REM
4010 REM *****
4020 REM"*** BILDAUFBAU ***
```

```

4030 REM *****
4040 REM
4050 PRINT CHR$(147)
4060 PRINT" FISCHERTECHNIK"
4070 PRINT" COMPUTING"
4080 PRINT
4090 PRINT"[RVS-ON] |";
4100 PRINT"[RVS-ON] |DIAGNOSEPROGRAMM |";
4110 PRINT"[RVS-ON] |";
4130 PRINT LEFT$(VT$,9);
4140 PRINT" ANALOGEINGABE EX BENUTZT (J/N)";
4150 GET X$
4160 IF X$<>"J" AND X$<>"N" THEN GOTO 4150
4170 PRINT X$
4180 X=X$="J"
4190 PRINT" ANALOGEINGABE EY BENUTZT (J/N)";
4200 GET Y
4210 IF Y$<>"J" AND Y$<>"N" THEN GOTO 4200
4220 PRINT Y$
4230 Y=Y$="J"
4240 PRINT LEFT$(VT$,9);
4250 PRINT" MOTORANWAHL : TASTEN 1-4 "
4260 PRINT" GEWAELHLTER MOTOR LINKS (PULS) : F1 "
4270 PRINT" GEWAELHLTER MOTOR LINKS (DAUER) : F3"
4280 PRINT" GEWAELHLTER MOTOR RECHTS (DAUER) : F5"
4290 PRINT" GEWAELHLTER MOTOR RECHTS (PULS) : F7
4300 PRINT" GEWAELHLTER MOTOR AUS : LEERTASTE"
4310 PRINT" ALLE MOTOREN AUS : BELIEBIGE TASTE"
4320 PRINT" PROGRAMMENDE : X
4330 PRINT
4340 PRINT" E1 E2 E3 E4 E5 E6 E7 E8 EX EY"
4350 PRINT
4360 RETURN
5000 REM
5010 REM *****
5020 REM"*** MOTORANZEIGE LOESCHEN ***
5030 REM *****
5040 REM
5050 PRINT LEFT$(VT$,22);
5060 PRINT " M 1 M 2 M 3 M 4"
5070 PRINT
5080 FOR I=1 TO 4
5090 S$=STA$(INT(STA(I)/100))
5100 PRINT " ";S$;
5110 NEXT
5120 RETURN

```

## Anschluß von zwei Interface

Wenn für größere Projekte die Zahl der Interface-Eingänge oder Ausgänge nicht ausreicht, so können zwei Interface an einen C64/128-Computer angeschlossen werden. Dabei wird ein Interface an den Computer angeschlossen. Das zweite Interface wird an eine Steckverbindung des ersten Interface angeschlossen.

Bei den fischertechnik COMPUTING Interface wird als Steckverbinder ein 20-poliger Anschluß verwendet (zwei Reihen à 10 Stifte). An diesen Anschluß kann ein weiteres fischertechnik COMPUTING Interface, Artikel 30 566, direkt (ohne Steckadapter) angeschlossen werden. Ältere fischertechnik-Interface verwenden als Steckverbindung einen 16-poligen Sockel. In diesem Fall muß das zweite Interface durch die Service-Abteilung fischertechnik der fischerwerke angeschlossen werden.

Die Kombination zweier Interface erlaubt die Steuerung von bis zu acht Motoren und die Abfrage von 16 Digitaleingängen. Die Analogeingänge des zweiten Interface können nicht genutzt werden; nur die Eingänge des ersten Interface sind verfügbar.

Zur Steuerung beider Interface ist ein spezielles Treiberprogramm notwendig. Dieses ist per Gutschein bei den fischerwerken abrufbar.

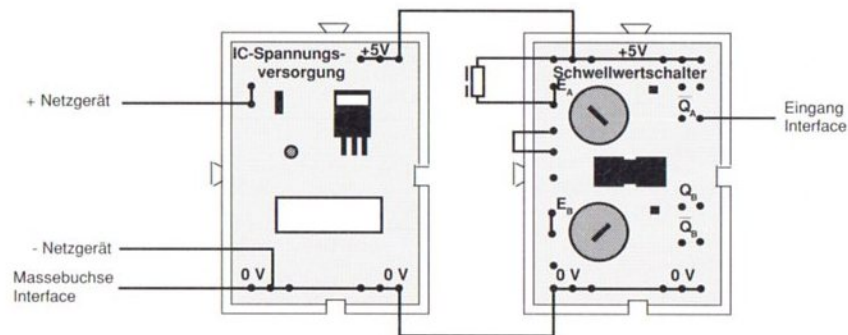
## Benutzung von fischertechnik Elektromechanik und Elektronik

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 Polwendschalter aus dem fischertechnik **Elektromechanik** Baukasten, aber auch den Reedkontakt oder den Schaltkontakt eines Relais (aus der **ec** Serie). 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 aus dem fischertechnik **Elektronik** Baukasten, verwendet werden. Die Motorausgänge des Interface sind kräftig

belastbar. Nicht nur die mini-Motoren, auch der S-Motor und der 6V-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 aus dem fi-

schertechnik **Elektronik** Baukasten) können ebenfalls in die Eingänge des Interface eingespeist werden. Als gemeinsamer Bezugspunkt ist jedoch zuvor die Masseschleife 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 auf der nächsten Seite 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 Kommando 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 durch geschoben 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 DATA-OUT erfolgt die Übernahme in das Speicherregister. Die zeitliche Abfolge der Signale können Sie dem Impulsiagramm entnehmen.

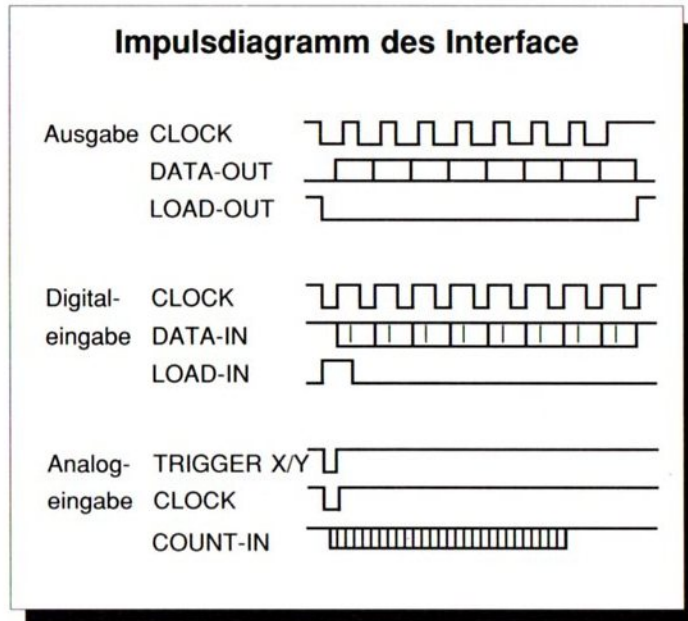
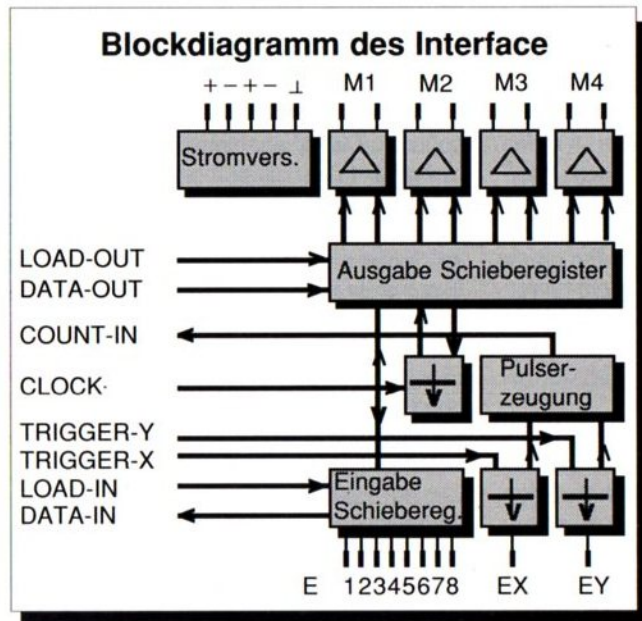
Ob die Daten allerdings auch die Leistungsverstärker durchsteuern, hängt wiederum von der Freigabesteuerung des Speicherbausteins ab. Die Freigabesteuerung erfolgt durch ein Monoflop. Diese Schaltung erzeugt ein Freigabesignal von einer halben Sekunde Dauer, wenn ein Impuls auf

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 weiterer Datentransfer erfolgen, so kippt das Monoflop wieder in seinen stabili-

len 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.

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 Eingabeschieberegister 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 "auf-sammeln" 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.

Blieben zum Schluß 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 das Startsignal TRIGGER-X bzw. TRIGGER-Y (mit negativer Logik) ausgelöst. Der Impuls wirkt auf den Toreingang eines Oszillators im Interface. Während der Impulsdauer erscheint ein Impulszug des Oszillators auf der Leitung COUNT-IN, die mit dem Zähleringang des Ein-/Ausgabebausteins verbunden ist. Der Zählerstand nach Ablauf des Impulses ist wiederum ein Maß für den Widerstandswert am Eingang. Die nicht vorhersagbaren Wartezyklen des Mikroprozessors beim C64/128 sind der Grund für diese aufwendige Signalaufbereitung. Sie sehen, daß der Analogwert weder etwas mit der Winkelstellung noch dem Widerstandswert der Potentiometer zu tun hat. Jedoch besteht zwischen der letztlich ermittelten Zahl und dem Widerstandswert 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 Computer verbunden wird. Es wird, wie Sie ja schon von dem Kapitel über den Anschluß des Interface wissen, der Userport verwendet. Von dessen acht Datenleitungen werden

die untersten sechs für die oben besprochenen Ausgabesignale verwendet, siehe auch die nachstehende Tabelle.

Die übrigen Datenleitungen sind auf Eingabe geschaltet. Die Leitung DATA-IN wird mit PB7 verbunden. Die Leitung COUNT-IN wird dem Eingang CNT2 zugeführt.

Wie in der Beschreibung der Software erwähnt, gibt es unterschiedliche Treiberprogramme mit verschiedenen Spezialbefehlen. Das nachstehend aufgeführte Maschinenprogramm führt die zuvor beschriebenen Abläufe zur Interfacesteuerung aus, wie sie z.B. im fischertechnik COMPUTING Baukasten verwendet werden.

#### Anschluß des fischertechnik Interface

| Interface-Signal | Userport-Signal | Stift    |
|------------------|-----------------|----------|
| <b>LOAD-OUT</b>  | <b>PB0</b>      | <b>C</b> |
| <b>LOAD-IN</b>   | <b>PB1</b>      | <b>D</b> |
| <b>DATA-OUT</b>  | <b>PB2</b>      | <b>E</b> |
| <b>CLOCK</b>     | <b>PB3</b>      | <b>J</b> |
| <b>TRIGGER-X</b> | <b>PB4</b>      | <b>H</b> |
| <b>TRIGGER-Y</b> | <b>PB5</b>      | <b>J</b> |
| <b>DATA-IN</b>   | <b>PB7</b>      | <b>L</b> |
| <b>COUNT-IN</b>  | <b>CNT2</b>     | <b>6</b> |

## Interface Treiber für C64

```

; C64 Interface Treiber
; Version 7C
; File C64IF7.C
;
; (C) Artur Fischer Forschung, 1984
; Steuerung des fischertechnik Interface
; mit Hilfe der C64 SYS- und USR-Kommandos.
;
; Initialisierung:
; SYS INIT
; Initialisiert das Interface und
; schaltet alle Motoren aus.
; Ausgabesteuerung:
; SYS <motornummer>,<betrieb>
; <motornummer> ist M1, M2, M3 oder M4.
; <betrieb> ist RECHTS, LINKS, EIN oder AUS.
; EIN entspricht RECHTS.
; Eingabesteuerung:
; Digitaleingabe-Kommandos:
; USR(<digitaleingang>)
; <digitaleingang> ist E1, E2, E3, E4,
; E5, E6, E7 oder E8.
; Analogeingabe-Kommandos:
; USR(<analogeingang>)
; <analogeingang> ist EX oder EY.
;
; Anschluß des Interface an den Userport:
; Interface Userport Leitung
; OV GND A
; LOAD-OUT PB0 C
; LOAD-IN PB1 D
; DATA-OUT PB2 E
; CLOCK PB3 F
; TRIGGER-X PB4 H
; TRIGGER-Y PB5 J
; DATA-IN PB7 L
; COUNT-IN CNT2 6
;*****
; .OS ;Erzeuge Object-Code
; .BA $CF00 ;Programm im Freispeicher
;*****
; CBM-BASIC Routinen
;*****
YFAC .DE $B3A2 ;Wandle (Y) in Floating
AYFAC .DE $B391 ;Wandle (A/Y) in Floating
CKCOM .DE $AEFD ;Prüfe auf Komma
FACINT .DE $B7A1 ;Wandle Floating in Integer

GETBYTE .DE $B79E ;Hole Byte-Ausdruck
;*****
; Ein- und Ausgaberegister
;*****
UP .DE $DD01 ;Userport Datenregister
DRR .DE $DD03 ;Userport Datenrichtung
TIL .DE $DD04 ;Timer Low Register
TIH .DE $DD05 ;Timer High Register
TIC .DE $DD0E ;Timer Control Register
;*****
; Variablen in der Zero-Page
;*****
AVAR .DE $FE ;Ausgabe-Variablen
MASK .DE $FF ;Masken-Variablen
;*****
; Routinen zur Ausgabesteuerung
; Einsprung mit SYS-Kommando
;*****
CF00 78 INIT SEI ;Interrupt sperren
CF01 A9 00 LDA #$00 ;alle Motoren aus
CF03 F0 26 BEQ STVAR ;BRANCH ALWAYS
CF05 A9 03 M1 LDA #*00000011 ;Motor 1
CF07 D0 0A BNE BOUT ;BRANCH ALWAYS
CF09 A9 0C M2 LDA #*00001100 ;Motor 2
CF0B D0 06 BNE BOUT ;BRANCH ALWAYS
CF0D A9 30 M3 LDA #*00110000 ;Motor 3
CF0F D0 02 BNE BOUT ;BRANCH ALWAYS
CF11 A9 C0 M4 LDA #*11000000 ;Motor 4
;*****
CF13 78 BOUT SEI ;Interrupt sperren
CF14 85 FF STA MASK ;Rette Bit-Maske
CF16 20 FD AE JSR CKCOM ;Prüfe auf Komma
CF19 A5 FE LDA AVAR ;Hole vorige Ausgabe
CF1B 05 FF ORA MASK ;Setze beide Bits
CF1D 85 FE STA AVAR ;Zwischenspeichern
CF1F 20 9E B7 JSR GETBYTE ;Hole 2. Argument
CF22 8A TXA
CF23 25 FF AND MASK ;Maskiere Motor
CF25 85 FF STA MASK ;Zwischenspeichern
CF27 A5 FE LDA AVAR ;Hole Ausgabe-Variablen
CF29 45 FF EOR MASK ;Setze Bit (10, 01, 00)
CF2B 85 FE STVAR STA AVAR ;Speichern zur Ausgabe
CF2D A8 TAY ;Kopie in Y-Reg.
;*****
; Routine zur Interface-Steuerung
; Ausgabesteuerung
; Ausgabe-Bitmuster in AVAR

```

```

; Benutzt A- und X-Register
;*****
CF2E A9 3F LDA #32F ;Setze Datenrichtung
CF30 8D 03 DD STA DRR
CF33 A2 08 LDX #08 ;Zähler für 8 Bit
CF35 A9 30 LOOP LDA #30 ;Ruhe-signal am Userport
CF37 06 FE ASL AVAR ;Teste Ausgabebit
CF39 90 02 BCC DOUT ;DATA-OUT low
CF3B 09 04 ORA #04 ;DATA-OUT high
CF3D 8D 01 DD DOUT STA UP ;Ausgabe an Userport
CF40 09 08 ORA #08 ;setze CLOCK
CF42 8D 01 DD STA UP ;Ausgabe an Userport
CF45 CA DEX ;Schleifen-zähler
CF46 D0 ED BNE LOOP ;Schleifen-ende LOOP
CF48 A9 39 LDA #39 ;Setze LOAD-OUT
CF4A 8D 01 DD STA UP ;Ausgabe an Userport
CF4D 84 FE STY AVAR ;Restauriere AVAR
CF4F 58 CLI ;Interrupt freigeben
CF50 60 RTS ;Rücksprung in BASIC
;*****
; Routine zur Interface-Steuerung
; Eingabesteuerung
; Einsprung mit der USR-Funktion
; Einsprungadresse muß unter der Adresse
; $0311 und $0312 gespeichert sein.
;*****
CF51 78 BINP SEI ;Interrupt sperren
CF52 20 A1 B7 JSR FACINT ;Argument in Int. wandeln
CF55 E0 A2 CPX #A2 ;Analogeingabe?
CF57 F0 33 BEQ POTS ;weiter zu POTS
CF59 E0 92 CPX #92 ;Analogeingabe?
CF5B F0 2F BEQ POTS ;weiter zu POTS
CF5D 86 FF STX MASK ;Rette Eingabemaske
;*****
; Interface-Steuerung
; zur Digitaleingabe
; Benutzt A-, X- und Y-Register
;*****
CF5F A9 32 SHIN LDA #32 ;Setze LOAD-IN
CF61 8D 01 DD STA UP ;Ausgabe an Userport
CF64 09 08 ORA #08 ;Setze CLOCK
CF66 8D 01 DD STA UP ;Ausgabe an Userport
CF69 A2 08 LDX #08 ;Zähler für 8 Bit
CF6B 0A LOOP2 ASL A ;A-Reg. links schieben
CF6C 2C 01 DD BIT UP ;Teste DATA-IN
CF6F 10 02 BPL MAKEIN ;DATA-IN ist low
CF71 09 01 ORA #01 ;DATA-IN ist high

CF73 A0 30 MAKEIN LDY #30 ;Lösche CLOCK
CF75 8C 01 DD STY UP ;Ausgabe an Userport
CF78 A0 38 LDY #38 ;Setze CLOCK
CF7A 8C 01 DD STY UP ;Ausgabe an Userport
CF7D CA DEX ;Schleifen-zähler
CF7E D0 EB BNE LOOP2 ;Schleifen-ende LOOP2
CF80 25 FF AND MASK ;Maskiere Bit
CF82 A8 TAY ;Speichere in Y-Register
CF83 F0 02 BEQ CVAR ;Rückgabe 0
CF85 A0 01 LDY #01 ;Rückgabe 1
CF87 20 A2 B3 CVAR JSR YFAC ;Wandle in Floating
CF8A 58 END CLI ;Interrupt freigeben
CF8B 60 RTS ;Rücksprung in BASIC
;*****
; Analog-Eingabe
; Wenn das Argument der USR-Funktion
; entweder $92 (Eingang EY) oder $A2
; (Eingang EX) ist, wird dieses Unter-
; programm angesprochen.
; Benutzt A-, X- und Y-Register.
;*****
CF8C A9 FF POTS LDA #FFF ;Setze Zählerreg. $FFFF
CF8E 8D 04 DD STA TIL
CF91 8D 05 DD STA T1H
CF94 A9 B9 LDA #B9 ;Setze Timer-Control-Reg.
CF96 8D 0E DD STA TIC
CF99 8E 01 DD STX UP ;Triggere Monoflop
CF9C A2 3A LDX #3A ;Lösche Trigger-Signal
CF9E 8E 01 DD STX UP ;Ausgabe an Userport
CFA1 AD 04 DD TST LDA T1L ;Teste Timer-Low-Register
CFA4 A2 03 LDX #03 ;Verzögerungsschleife
CFA6 CA DELAY DEX
CFA7 D0 FD BNE DELAY
CFA9 38 SEC ;Subtr. Timer-Low-Reg.
CFAA ED 04 DD SBC T1L ;vom vorigen Wert.
CFAD D0 F2 BNE TST ;Kommen noch Pulse?
CFAF A2 38 LDX #38 ;Setze CLOCK, LÖ. LOAD-IN
CFB1 8E 01 DD STX UP ;Ausgabe an Userport
CFB4 38 SEC ;Ermittle Zählerstand
CFB5 A9 FF LDA #FFF
CFB7 ED 04 DD SBC T1L ;Low Byte
CFBA A8 TAY ;in Y-Register
CFBB A9 FF LDA #FFF
CFBD ED 05 DD SBC T1H ;High Byte
CFC0 20 91 B3 JSR AYFAC ;Wandle A/Y in Floating
CFC3 4C 8A CF JMP END ;Rücksprung in BASIC
.EN

```

## 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 Drehrichtung. - Leistungsstufe des Motors defekt.

- Motor läuft sehr langsam oder setzt aus. - Entweder Netzgerät durch zuviele Motoren überlastet (zwei Netzgeräte mot4 oder das stärkere COMPUTING Netzgerät verwenden) oder Netzgerät bei Verwendung des regelbaren Ausgangs nicht voll aufgedreht.
- Die Abfrage eines Analogkanals benötigt eine spürbare Zeit. - Es wurde ein unbeschalteter Analogeingang abgefragt.

Bei Defekten schicken Sie das Gerät bitte an folgende Adresse ein:

**fischerwerkeArtur Fischer GmbH&Co.KG**  
**Abteilung Service**  
**D-7244 Tumlingen / Waldachtal**

## Technische Daten

fischertechnik COMPUTING Interface zum Anschluß an die Benutzerschnittstelle (Userport) der Commodore Computer C64, C64-II, SX64, C128 und C128-D, Art.Nr. 30566. Mit anderen Adaptern und Softwarepaketen kann das Interface auch an Computern mit Parallel-Druckerschnittstellen eingesetzt werden.

### Ausgabe:

4 Ausgänge zum Anschluß von Motoren, Lampen, Elektromagneten.. (M1 bis M4). Polarität des Ausgangs steuerbar. Belastbarkeit: 1A Dauerstrom, 1,5A Spitzstrom.

Schutzmaßnahmen: Bei einem Kurzschluß am Ausgang schaltet eine Kurzschlußsicherung den Ausgangsverstärker ab. Das Einschalten erfolgt automatisch beim Entfernen des Kurzschlusses. Bei Überlastung des Ausgangs steigt die Temperatur im Ausgangsverstärker. Bei Überschreitung der zulässigen Temperatur spricht eine Temperatursicherung im Ausgangsverstärker an. Das Wiedereinschalten erfolgt nach Abkühlung des Ausgangsverstärkers.

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

### Eingabe:

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.

2 Eingänge für analoge Signale (EX und EY).

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

### Stromversorgung:

Betriebsspannung: 6 bis 10 V ungesieberte Gleichspannung oder 8 V gesieberte Gleichspannung.

Die Verwendung des fischertechnik Netzgeräts COMPUTING wird empfohlen.

### Anschlüsse:

Interface-Anschlußkabel.

Netzgerät-Anschluß (doppelt), 2 Buchsen, 2,5 mm  $\varnothing$ . Massebuchse, 2,5 mm  $\varnothing$ . Modellanschluß, 20-polig nach DIN 41651. Anschluß für weiteres Interface

### Abmessungen:

150 x 90 x 30 mm,  
1200 mm Anschlußkabel,  
transparenter Deckel.

## Important Notice

All measures and efforts have been taken to guarantee the correctness of this product documentation. However, since fischerwerke Artur Fischer GmbH & Co. KG constantly work on the improvement of their products, we cannot guarantee the completeness and correctness of this documentation from the time of printing onwards.

Amiga, Commodore 64, Commodore 128, Commodore VIC20 und CBM are trademarks of Commodore Electronics Limited.

Apple is a registered trademark of the Apple Computer Inc.

Atari is a registered trademark of the Atari Corporation.

Centronics is a registered trademark of the Data Computer Corporation.

CPC 464, CPC 664 und CPC 6128 are trademarks of the Amstrad Consumer Electronics plc.

CP/M is a trademark of the Digital Research Inc.

IBM and IBM PC are registered trademarks of the International Business Machines Corporation.

This equipment generates and uses radio frequency energy. If it is not properly installed and used in strict accordance with the manufacturer's instructions, this equipment may interfere with radio and television reception. This device has been tested and found to comply with the limits for a class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference in a residential installation. If you suspect interference, you can test this equipment by turning it off and on. If you determine that there is interference with radio or television reception, try one or more of the following measures to correct it:

- reorient the receiving antenna
- move the interface and the computer away from the receiver
- change the relative positions of the equipment and the receiver
- plug the power supply of the interface (as well as the computer equipment) in a different outlet so that the equipment and the receiver are on different branch circuits.

If necessary, consult your fischertechnik dealer or an experienced radio/television technician for additional suggestions. You may also wish to consult the following booklet, which was prepared by the Federal Communications Commission:

"How to Identify and Resolve Radio-TV Interference Problems"

This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402, Stock No. 004-000-00345-4.



## 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 construction sets (fischertechnik COMPUTING kit, Training Robot, Plotter/Scanner, EXPERIMENTAL), 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 Commodore 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 four fischertechnik motors, lamps, electromagnets etc.
- With it you can interrogate eight push-buttons 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. Just try it out - you will have lots of fun!

Sincerely yours  
Artur and Klaus Fischer

## Connecting the Interface

The fischertechnik COMPUTING interface part number 30 566 can be connected to a whole range of computers, among of which are also the Commodore's 8 bit computers, i.e. C64, C64-II, SX64, C128, and C128-D. Even the old VIC20, PET and CBM computers as well as the Plus-4 belong to this class of computers; they are not supported by the fischertechnik COMPUTING software, however. You may consider to write the software for those computers along the guide-lines given in this manual.

You cannot use a Commodore 116 or C16 computer, since those computers are not equipped with an userport.

If you own a computer of Commodore's PC- or Amiga-series the connection, the software and the procedure described in this manual do not apply. You will require the adapter, the software and the manual for IBM-PC or Commodore Amiga, respectively.

The fischertechnik COMPUTING interface is connected to the userport of the Commodore computer. 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 userport. It is the printed

circuit board connector at the right (see illustration on the next page).

- 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, i.e. a water tap.
- Place the interface in front of you.
- Together with this instruction manual you got the adapter connector fitting to the userport.
- The adapter is inserted between the interface and the computer. The adapter carries two connectors. One plug fits to the connector of the interface cable. The other plug fits to the userport. Insert the cable of the interface into the adapter connector. A reversed insertion is prohibited by the shape of the connector.
- Turn the adapter so that the connectors are on the top side. Now plug the adapter onto the userport of your computer. Also in this case plugging in the wrong way is impossible. So if it does

not seem to fit, please check once more. Do not force it on.

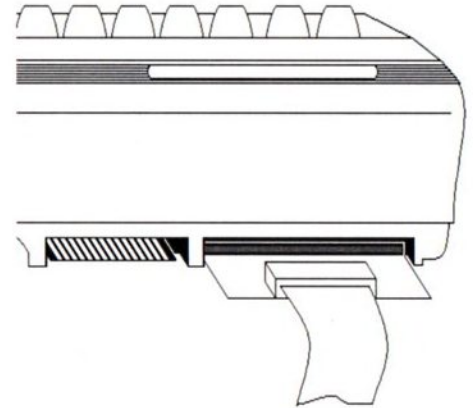
- Connect the interface to the power supply unit. The interface expects an unfiltered direct current with a voltage between 6 and 10 volts. If a stabilized direct voltage source is used the power supply must be adjusted to approximately 8 volts (loadable to approximately 2 amps). Using the fischertechnik COMPUTING power supply unit will guarantee the correct voltage. Plug the output cable of the power supply unit with the 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 unit mot 4 may also be used, i.e. especially for the models of the fischertechnik COMPUTING robotics 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 mot4 into the empty jacks.

- When using the mot4 power supply unit, 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 fischertechnik COMPUTING kits. This cable can also be ordered as a spare part.

- 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 effects caused by the interface: as long as the interface is plugged in, you cannot use any device which should be plugged into the userport. This includes the RS232C module and the telephone modem.

Once more for your remembrance: Prior to working with the models you should also always discharge possible electrostatic charges by touching a grounded metal object, i.e. a water tap. 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 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, however. And with the fischertechnik COMPUTING kits comes the software you need to make it all work smoothly and easily. Just tick the appropriate box on the coupon and mail it to the closest fischertechnik representative as given in the address list. On return of post there will be mailed the disk fitting to your construction kit and containing all the control programs. Additionally this manual details you how to write such control programs yourself.

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 COPY/ALL contained on the system diskette of the

disk drive. Proceed exactly according to the instructions of 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. The procedure is detailed in the disk-drive manual.

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. If you use a C128 or C128-D computer you should flip it to C64 mode. Type the following command:

**GO64**

Now appears the question "ARE YOU SURE?". Answer it with

**Y**

Shortly after that appears the screen display of the C64 and you may use the whole range of fischertechnik COMPUTING software without modification.

After the system has started insert the fischertechnik COMPUTING diskette in the disk drive. Load the program FISCHER by typing in:

**LOAD"0:\*",8**

The disk drive will spin and the program is loaded into the computer. Now give the command for program start:

**RUN**

The program will display some messages which give you a description of the hardware and software of your fischertechnik COMPUTING construction kit. The program may also display some additional information and last minute news not contained in the instruction manual, so watch out.

Next load the interface driver program DRIVER into the computer. As the interface driver is different for each of the fischertechnik COMPUTING kits select one of the following commands:

**LOAD"DRIVER.BAS",8**

or

**LOAD"DRIVER.64",8**

or

**LOAD"ROBOTSYS.64",8**

If you are not sure, which program is contained on the diskette you should call in the disk directory:

```
LOAD"$",8  
LIST
```

The interface driver program installs everything you need for data exchange with the interface. The driver program is also contained in all other sample programs on the disk.

The interface driver program contains the above-mentioned detail knowledge about the input/output devices of the computer. By the use of the interface driver program the control of the interface is simplified to a great extent. We would like to note that the interface driver programs on the different fischertechnik COMPUTING diskettes differ from each other. According to the topic of the construction kit it contains specialized commands: robotic commands for the Training Robot, or data taking commands for the EXPERIMENTAL kit etc. Therefore you must not exchange the driver programs.

What is described in this manual is a nucleus of commands as is available in every interface driver program. Start the driver program:

```
RUN
```

After completion of the program the compu-

ter will display its ready prompt again.

There is no apparent change on your computer. However your Commodore computer 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 Commodore computer and the fischertechnik COMPUTING interface. They will use the SYS command and the USR function. The driver program itself is written in BASIC and deposits a machine language program in the computers memory by either loading it from disk or by reading a couple of DATA lines. It also generates the parameters of the machine language program calls as BASIC variables (M1, M2, M3, M4, EX, EY, E1 ... CW, CCW, OFF). The only BASIC commands you need to be able to use are the following ones:

The motor output M1 is controlled with the following commands:

```
SYS M1,CW  
(clockwise rotation)  
SYS M1,CCW  
(counter-clockwise rotation)  
SYS M1,OFF  
(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 sensor, e.g. a potentiometer (4.7 kΩ). The functions

```
USR(EX)
```

```
USR(EY)
```

will then produce a value within the range between 15 and 200 according to the position of the potentiometer. The actual limits may be slightly different.

Analog inputs are mostly used for measuring positions. If, for example, a robot arm like the one in the fischertechnik COMPUTING robotics kit 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 sensor is connected to inputs EX or EY, the effect of 100 k $\Omega$  resistors contained in the interface circuit will be perceptible. They are parallel to the external sensor and avoid "hanging" of the interface. The analog value will range between 4000 and 5000 in that case. We recommend, however, never to leave the analog inputs without connection. In case you don't use them, just make a short between the analog input and the reference voltage, +5 V; this corresponds to a resistance of 0  $\Omega$ . 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,CW**

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 computer. At this point you might be wondering why

the motor 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 the STOP key. And if the program is resumed by the command CONT the first interface 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 keyboard 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 k $\Omega$  potentiometer like the one in the fischertechnik COMPUTING robotics kit 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 15 and 200.

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 or a similar 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 lines till 500. You therefore should type in:

**510 PRINT IN (EX)**

**520 GOTO 510**

Then give the command

**RUN**

There may be a brief delay while the machine language program is loaded. 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 approximately be between 15 and 200. Activate the stop key 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 install the machine language program which provides the essential code for data transfer to the interface. The machine language program is either loaded from disk or read in from DATA lines, controlled by calculating a check sum and deposited in the computer's memory. Besides the program itself, the standardized parameters INIT, M1, M2, M3, M4, E1, E2, E3, E4, E5, E6, E7, E8, EX, EY, CW, CCW, and OFF are set.

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. Also keep in mind that Commodore Basic differentiates between variables using only

the leading two characters of the variable's name. Therefore variable names like:

**EXTRA, M114, OFTEN, CCD, EYE** are unuseable. However, the limitation does not apply to variables of another type (strings, real), so that

**OFF\$, M1!, E1!...** may be used.

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 operation and the source code.

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

**500 SYS 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.

On the next page the listing of a simple interface driver program is given, which may serve for the experiments described. Remember that your fischertechnik COMPUTING kit may use a different driver programs supplying additional commands.

## Interface Driver Program for C64

```
1 PRINT CHR$(147)
5 PRINT"LOADING DRIVER"
10 REM INTERFACE PROGRAM FOR COMMODORE 64/128
20 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
30 REM COMMANDS OF THE DRIVER
40 REM SYS M1,OFF
50 REM SYS M1,CW      SYS M1,CCW
60 REM USR(E1)      USR(EX)      USR(EY)
70 REM M1 - M4 ARE MOTOR CONTROL COMMANDS
80 REM E1 - E8 ARE DIGITAL INPUTS
90 REM EX AND EY ARE ANALOG INPUTS
100 DATA 52736,169,0,240,38,169,3,208,10,53573
110 DATA 169,12,208,6,169,48,208,2,54395
120 DATA 169,192,120,133,255,32,253,174,55723
130 DATA 165,254,5,255,133,254,32,158,56979
140 DATA 183,138,37,255,133,255,165,254,58399
150 DATA 69,255,133,254,168,169,63,141,59651
160 DATA 3,221,162,8,169,48,6,254,60522
170 DATA 144,2,9,4,141,1,221,9,61053
180 DATA 8,141,1,221,202,208,237,169,62240
190 DATA 57,141,1,221,132,254,88,96,63230
200 DATA 120,32,161,183,224,162,240,51,64403
210 DATA 224,146,240,47,134,255,169,50,65668
220 DATA 141,1,221,9,8,141,1,221,66411
230 DATA 162,8,10,44,1,221,16,2,66875
240 DATA 9,1,160,48,140,1,221,160,67615
250 DATA 56,140,1,221,202,208,235,37,68715

260 DATA 255,168,240,2,160,1,32,162,69735
270 DATA 179,88,96,169,255,141,4,221,70888
280 DATA 141,5,221,169,185,141,14,221,71985
290 DATA 142,1,221,162,58,142,1,221,72933
300 DATA 173,4,221,162,3,202,208,253,74159
310 DATA 56,237,4,221,208,242,162,56,75345
320 DATA 142,1,221,56,169,255,237,4,76430
330 DATA 221,168,169,255,237,5,221,32,77738
340 DATA 145,179,76,137,206,0,0,0,78481
350 DATA 1,2,4,8,16,32,64,128,78736
360 DATA 162,146,255,170,85,80,206,79840
370 READ INIT : M1=INIT
380 FOR M3=0 TO 24 : FOR M2=0 TO 7
390 READ M4 : POKE INIT+M3*8+M2,M4
400 M1=M1+M4 : NEXT
410 READ M4 : IF M1<>M4 THEN PRINT"DATA ERROR IN LINE":M3*10+100:END
420 NEXT
430 READ E1,E2,E3,E4,E5,E6,E7,E8
440 M1=M1+E1+E2+E3+E4+E5+E6+E7+E8
450 READ M4 : IF M1<>M4 THEN PRINT"DATA ERROR IN LINE 350" : END
460 READ EX,EY,OFF,CCW,CW,M2,M3
470 M1=M1+EX+EY+OFF+CCW+CW+M2+M3
480 READ M4 : IF M1<>M4 THEN PRINT"DATA ERROR IN LINE 360" : END
485 POKE785,M2 : POKE786,M3
490 M1=INIT+4 : M2=M1+4 : M3=M2+4 : M4=M3+4
500 SYS 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 every fischertechnik diskette and labelled as the DIAGNOST.64 or 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.

You may control the outputs of the interface as well. Proceed as indicated by the program's screen display. Select the output channel using the numeric keys. The selected channel will be indicated in reverse characters. Using the DIAGNOST.64 file you may now control the output to operate

a motor in clockwise or counter-clockwise sense of rotation or you may switch off the motor by hitting the keys Q, P, and O, respectively.

The DIAGNOST.BAS version of the diagnostic program uses the function keys for motor control, in which case you may select from five different modes of operation. Pressing F1 will operate the motor in clockwise sense of rotation. Operation will last only as long as you are pressing the key. The same holds for F7; the other sense of rotation is selected, however. The keys F3 and F5 switch on the motor permanently. It has to stopped explicitly by pressing the space bar. Normally you will prefer the first pair of function keys for fine positioning of a robotic arm. The latter function keys serve for keeping a device in operation even when selecting other output channels. To give an example: you will use F3 or F5 if you want to activate an electromagnet acting as a robot's gripper and keep it activated even during movements of the arm.

Please note that the diagnostic program will not only serve for testing if a device is operational at all. It will as well tell you if the sense of rotation is the correct one. If you

want to change the sense of rotation it is easiest to reverse the two motor wire connections.

Pressing on the key "C" will switch off all motors (emergency stop) and pressing the key "X" will quit the program.

## Diagnostic Program for C64

```
500 SYS INIT
510 REM
520 REM FISCHERTECHNIK COMPUTING
530 REM
540 REM"DIAGNOSTIC PROGRAM
550 REM"FILE: DIAGNOST.BAS
560 REM"COPYRIGHT (C) FISCHERWERKE 1986
570 REM
580 REM"FUNCTION
590 REM
600 REM"USING THIS PROGRAM YOU MAY CONTROL
610 REM"ALL FUNCTIONS OF A MODEL.
620 REM"ALL INPUTS ARE DISPLAYED ON THE SCREEN.
630 REM"ALL OUTPUTS ARE CONTROLLED VIA
640 REM"KEYBOARD COMMANDS.
650 LET VT$="{HOME}{24*DOWN}"
1000 REM
1010 REM *****
1020 REM"*** INITIALIZE CONTROL ***
1030 REM *****
1040 REM
1050 DIM STA(4),STA$(2):REM"STATUS OF THE MOTORS
1060 DIM M(4):REM"ADDRESSES FOR MOTOR 1 THRU 4
1070 GOSUB 4050 :REM"SCREEN DISPLAY
1080 REM"SWITCH ALL MOTORS OFF
1090 FOR I=1 TO 4
1100 LET STA(I)=OFF
1110 NEXT I
1120 REM"CONTROL CODES
1130 LET SP$=CHR$(32) :REM"SPACE
1140 LET F1$=CHR$(133) :REM"F1
1150 LET F3$=CHR$(134) :REM"F3
1160 LET F5$=CHR$(135) :REM"F5
1170 LET F7$=CHR$(136) :REM"F7
1180 REM"TEXT FOR STATUS DISPLAY
1190 STA$(0)="CW "
1200 STA$(1)="CCW"
1210 STA$(2)="OFF"
1220 LET M(1)=M1:M(2)=M2:M(3)=M3:M(4)=M4
1230 LET MOT=1:REM"START WITH MOTOR 1
2000 REM
2010 REM *****
2020 REM"*** DISPLAY OF INPUT ***
2030 REM *****
2040 REM
2050 GOSUB 5050 :REM"ERASE MOTOR STATUS DISPLAY
2060 PRINT LEFT$(VT$,20);
2070 PRINT TAB(2);USR(E1);
2080 PRINT TAB(5);USR(E2);
2090 PRINT TAB(8);USR(E3);
2100 PRINT TAB(11);USR(E4);
2110 PRINT TAB(14);USR(E5);
2120 PRINT TAB(17);USR(E6);
2130 PRINT TAB(20);USR(E7);
2140 PRINT TAB(23);USR(E8);
2150 IF X THEN PRINT TAB(28);RIGHT$( " "+STR$(USR(EX)),3);
2160 IF Y THEN PRINT TAB(34);RIGHT$( " "+STR$(USR(EY)),3)
3000 REM
3010 REM *****
3020 REM"*** CONTROL OF OUTPUT ***
3030 REM *****
3040 REM
3050 GET K$
3060 LET F=0 :REM"FLAG FOR VALID COMMAND
3070 LET M=VAL(K$) :REM"NUMERIC KEY
3080 IF M>=1 AND M<=4 THEN LET MOT=M : LET F=1 : GOSUB 5050
3090 LET ZU=STA(MOT)
3100 IF K$=SP$ THEN LET ZU=OFF:STA(MOT)=OFF:LET F=1
3110 IF K$=F1$ THEN LET ZU=CCW :LET F=1
3120 IF K$=F3$ THEN LET ZU=CCW:STA(MOT)=CCW:LET F=1
3130 IF K$=F5$ THEN LET ZU=CW:STA(MOT)=CW:LET F=1
3140 IF K$=F7$ THEN LET ZU=CW:LET F=1
3150 IF K$="X" THEN SYS INIT:PRINT CHR$(147):END
3160 IF K$="" OR F=1 THEN GOTO 3260
3180 REM"ALL OTHER KEYS -> EMERGENCY SHUT DOWN
3190 SYS INIT
3200 FOR I=1 TO 4
3210 LET STA(I)=OFF
3220 NEXT I
3230 LET ZU = OFF
3240 GOSUB 5050
3250 REM"OUTPUT TO SELECTED MOTOR
3260 SYS M(MOT),ZU
3270 PRINT LEFT$(VT$,22);
3280 PRINT TAB(5*MOT-3);"[RVS-ON]M";MOT;"[4*LEFT]";
3290 PRINT"[DOWN]{DOWN}";
3300 LET S$=STA$(INT(ZU/100))
3310 PRINT S$;
3320 PRINT "(RVS-OFF)";
3330 GOTO 2060
4000 REM
4010 REM *****
4020 REM"*** SCREEN DISPLAY ***
4030 REM *****
```

```

4040 REM
4050 PRINT CHR$(147)
4060 PRINT" FISCHERTECHNIK"
4070 PRINT" COMPUTING"
4080 PRINT
4090 PRINT"(RVS-ON) |";
4100 PRINT"(RVS-ON)|DIAGNOSTIC PROGRAM |";
4110 PRINT"(RVS-ON) |";
4130 PRINT LEFT$(VT$,9);
4140 PRINT" DO YOU USE POTENTIOMETER EX (Y/N)";
4150 GET X$
4160 IF X$<>"Y" AND X$<>"N" THEN GOTO 4150
4170 PRINT X$
4180 X=X$="Y"
4190 PRINT" DO YOU USE POTENTIOMETER EY (Y/N)";
4200 GET Y$
4210 IF Y$<>"Y" AND Y$<>"N" THEN GOTO 4200
4220 PRINT Y$
4230 Y=Y$="Y"
4240 PRINT LEFT$(VT$,9);
4250 PRINT" SELECT MOTOR: KEYS 1-4 "
4260 PRINT" SELECT MOTOR COUNTER-CW (PULSE): F1 "
4270 PRINT" SELECT MOTOR COUNTER-CW (PERM.): F3"
4280 PRINT" SELECT MOTOR CLOCKWISE (PERM.) : F5"
4290 PRINT" SELECT MOTOR CLOCKWISE (PULSE) : F7
4300 PRINT" SELECT MOTOR OFF : SPACE BAR"
4310 PRINT" ALL MOTORS OFF : ANY OTHER KEY"
4320 PRINT" END OF PROGRAM : X
4330 PRINT
4340 PRINT" E1 E2 E3 E4 E5 E6 E7 E8 EX EY"
4350 PRINT
4360 RETURN
5000 REM
5010 REM *****
5020 REM"*** ERASE DISPLAY OF MOTORS ***
5030 REM *****
5040 REM
5050 PRINT LEFT$(VT$,22);
5060 PRINT " M 1 M 2 M 3 M 4"
5070 PRINT
5080 FOR I=1 TO 4
5090 S$=STAS(INT(STA(I)/100))
5100 PRINT " ";S$;
5110 NEXT
5120 RETURN

```

## Control of two Interfaces

If in the course of great projects the number of interface inputs or outputs becomes insufficient you may connect two interfaces to one single C64-computer. The procedure is as follows: the first interface is connected to the computer as usual. The second interface is connected to a pin connector in the first interface.

The connector of the fischertechnik COMPUTING interface is a 20 pole connector consisting of two rows of ten pins each. You may plug to this connector another fischer-technik COMPUTING interface (order number 30 566) directly without using any adapter. Earlier versions of the interface use a 16 pin IC socket as connector. Those interfaces may be used as well, however you should consult the fischertechnik representative for installing a second interface.

The combination of two interface allows the control of as much as eight motors und 16 digital inputs. The analog channels of the second interface may not be used, however; there remain the two analog channels of the first interface.

The control of both interfaces requires a special driver program. Ask your fischer-technik representative for more information.

## Using the fischertechnik Electromechanics and Electronic Kits

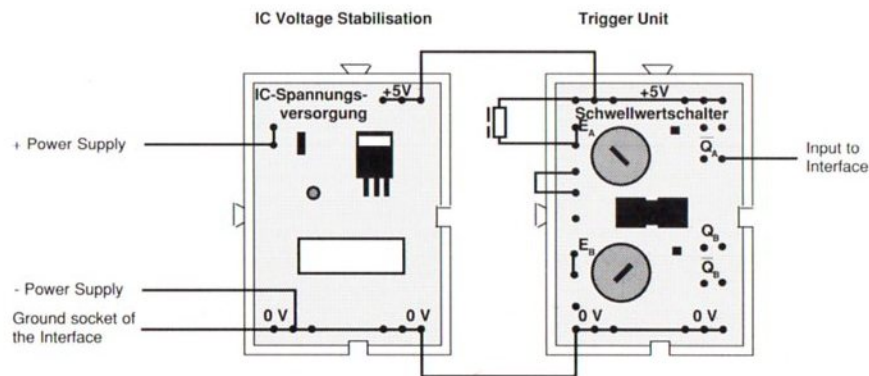
The fischertechnik computing interface is compatible with the components and electronic devices of the above-mentioned kits. Other types of pushbutton switches and switches may be used in place of the mini pushbuttons used with the fischertechnik computing models, e.g. the large pushbutton or the reversing switch contained in the fischertechnik **Electromechanics** kit, but also the reed contact or the switching contact of a relay (e.g. from the **ec** kits). 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 from the fischertechnik **Electronics** kit, may be used as well.

The motor outputs of the interface feature high load capacity. Not only the mini mo-

tors, but also the S-motor and the 6V-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 from the fischertechnik

**Electronics** kit) may also be fed to the interface inputs. 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. The figure 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 (see next page). 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 sequentially (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 storage register which is integrated in the shift register chip. 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.

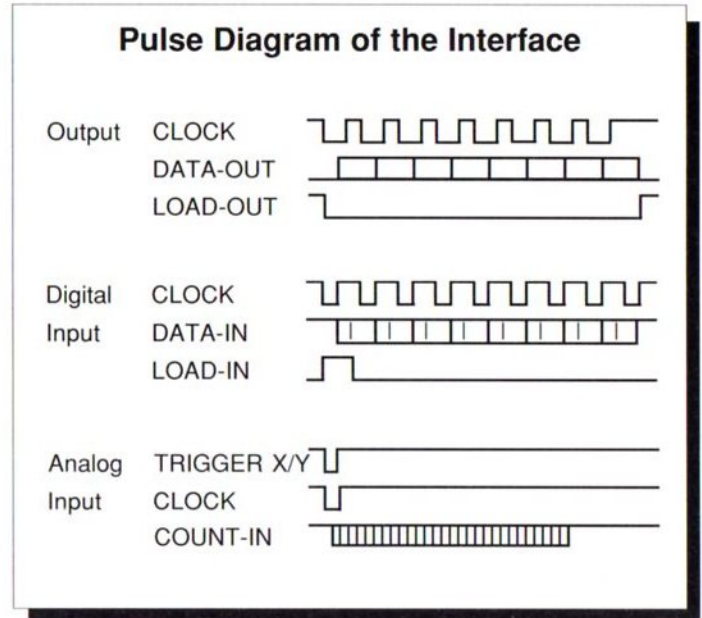
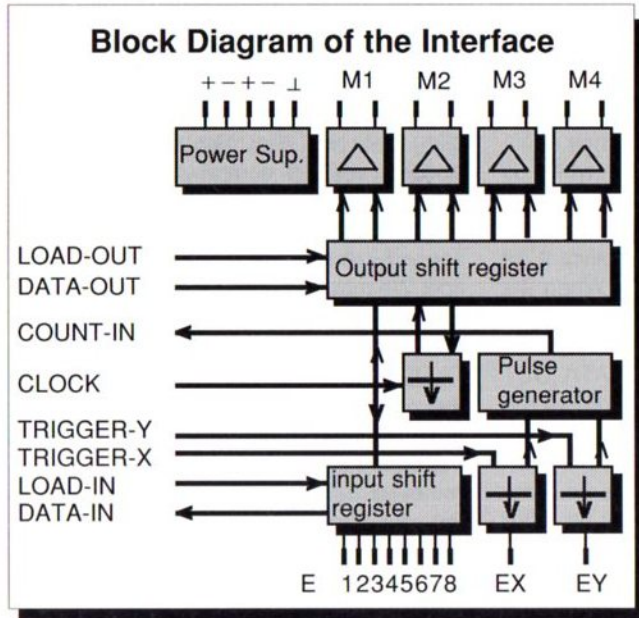
Whether the data are fed to the power amplifiers, however, depends on the enabling control of the memory module. The enabling circuit is controlled by a monostable multivibrator. This circuit generates an enabling signal with a duration of half a second if there is a pulse on the CLOCK line. We may assume 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.

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. The pulse operates the gate of an oscillator on the interface board. While the pulse is active, a pulse train is generated by the oscillator and fed via the line COUNT-IN to the counter input of the userport. The contents of the counting register after termination of the pulse train is a measure of the pulse duration as well. The reason for this effort are the unpredictable wait states of the processor of the C64 and C128.

You can see that the analog value not directly gives the angle of the potentiometer slider or the value of the resistance. Nevertheless 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 Commodore computer. As you know from the section about connecting the interface, the computer's userport is employed for this purpose. Out of the eight data lines of this port, the lower six are used for the output signals discussed above (Also refer to the following table.).

The remaining port bits are used for input. DATA-IN is connected to PB7. The counting input is on CNT2.

As detailed in the chapter on the fischertechnik COMPUTING software there exist different interface driver programs containing specialized commands. The machine language program listed next is the one used in the fischertechnik COMPUTING robotic kit and provides the most frequently used commands.

## Connecting the fischertechnik Interface to the Userport

| Interface Signal | Userport Signal | Pin      |
|------------------|-----------------|----------|
| <b>LOAD OUT</b>  | <b>PB0</b>      | <b>C</b> |
| <b>LOAD IN</b>   | <b>PB1</b>      | <b>D</b> |
| <b>DATA OUT</b>  | <b>PB2</b>      | <b>E</b> |
| <b>CLOCK</b>     | <b>PB3</b>      | <b>F</b> |
| <b>TRIGGER-X</b> | <b>PB4</b>      | <b>H</b> |
| <b>TRIGGER-Y</b> | <b>PB5</b>      | <b>J</b> |
| <b>DATA IN</b>   | <b>PB7</b>      | <b>L</b> |
| <b>COUNT-IN</b>  | <b>CNT2</b>     | <b>6</b> |

```

; C64 Interface Driver Program
;
; Version 7C
; File C64IF7.C
;
; (C) Artur Fischer Forschung, 1984
; Control of the fischertechnik Interface
; using the C64 SYS- and USR-commands
; for output and input, respectively.
;
; Initializing:
; SYS INIT
; Initializes the interface and
; shuts down all motor outputs.
; Output control:
; SYS <motornumber>,<mode>
; <motornumber> is M1, M2, M3 or M4.
; <mode> ist CW, CCW, or OFF.
; Input control:
; Digital input commands:
; USR(<digital_input>)
; <digital_input> is E1, E2, E3, E4,
; E5, E6, E7, or E8.
; Analog input commands:
; USR(<analog_input>)
; <analog_input> is EX or EY.
;
; Connection of the interface to the userport:
; Interface Userport Pin
; OV GND A
; LOAD-OUT PB0 C
; LOAD-IN PB1 D
; DATA-OUT PB2 E
; CLOCK PB3 F
; TRIGGER-X PB4 H
; TRIGGER-Y PB5 J
; DATA-IN PB7 L
; COUNT-IN CNT2 6
;*****
.OS ;Generate object code
.BA $CF00 ;Program in free RAM area
;*****
; CBM-BASIC Routines
;*****
YFAC .DE $B3A2 ;Convert (Y) to floating
AYFAC .DE $B391 ;Convert (A/Y) to floating
CKCOM .DE $AEFD ;Check for comma

FACINT .DE $B7A1 ;Convert floating to integer
GETBYTE .DE $B79E ;Get byte expression
;*****
; Input and output registers
;*****
UP .DE $DD01 ;Userport data register
DRR .DE $DD03 ;Userport data direction
TIL .DE $DD04 ;Timer low register
TIH .DE $DD05 ;Timer high register
TIC .DE $DD0E ;Timer control register
;*****
; Variables in the zero page
;*****
AVAR .DE $FE ;output variable
MASK .DE $FF ;mask variable
;*****
; Routines for output control
; Call in using the SYS command
;*****
CF00 78 INIT SEI ;Disable interrupt
CF01 A9 00 LDA #$00 ;All motors off
CF03 F0 26 BEQ STVAR ;BRANCH ALWAYS
CF05 A9 03 M1 LDA #$00000011 ;Motor 1
CF07 D0 0A BNE BOUT ;BRANCH ALWAYS
CF09 A9 0C M2 LDA #$00001100 ;Motor 2
CF0B D0 06 BNE BOUT ;BRANCH ALWAYS
CF0D A9 30 M3 LDA #$00110000 ;Motor 3
CF0F D0 02 BNE BOUT ;BRANCH ALWAYS
CF11 A9 C0 M4 LDA #$11000000 ;Motor 4
;*****
CF13 78 BOUT SEI ;Disable interrupt
CF14 85 FF STA MASK ;Save bit mask
CF16 20 FD AE JSR CKCOM ;Check for comma
CF19 A5 FE LDA AVAR ;Get previous output
CF1B 05 FF ORA MASK ;Set both bits
CF1D 85 FE STA AVAR ;Intermediate storage
CF1F 20 9E B7 JSR GETBYTE ;Get 2nd argument
CF22 8A TXA
CF23 25 FF AND MASK ;Mask motor
CF25 85 FF STA MASK ;Intermediate storage
CF27 A5 FE LDA AVAR ;Get output variable
CF29 45 FF EOR MASK ;Set bit (10, 01, 00)
CF2B 85 FE STVAR STA AVAR ;Save for output
CF2D A8 TAY ;Copy in Y-reg.
;*****
; Routine for interface control
; Output control

```



```

; Output bit pattern in AVAR
; Uses A- and X-registers
;*****
CF2E A9 3F LDA #S3F ;Set data direction
CF30 8D 03 DD STA DRR
CF33 A2 08 LDX #S08 ;Loop for 8 bits
CF35 A9 30 LOOP LDA #S30 ;static pattern userport
CF37 06 FE ASL AVAR ;Test output bit
CF39 90 02 BCC DOUT ;DATA-OUT low
CF3B 09 04 ORA #S04 ;DATA-OUT high
CF3D 8D 01 DD DOUT STA UP ;Output to userport
CF40 09 08 ORA #S08 ;set CLOCK
CF42 8D 01 DD STA UP ;Output to userport
CF45 CA DEX ;Loop counter
CF46 D0 ED BNE LOOP ;end of LOOP
CF48 A9 39 LDA #S39 ;Set LOAD-OUT
CF4A 8D 01 DD STA UP ;Output to userport
CF4D 84 FE STY AVAR ;Restore AVAR
CF4F 58 CLI ;Enable interrupt
CF50 60 RTS ;Return to BASIC
;*****
; Routine for interface control
; Input control
; Call in using the USR-function
; Starting address must be stored in
; $0311 and $0312.
;*****
CF51 78 BINP SEI ;Disable interrupt
CF52 20 A1 B7 JSR FACINT ;Convert arg. to integer
CF55 E0 A2 CPX #SA2 ;Analog input?
CF57 F0 33 BEQ POTS ;Go to POTS
CF59 E0 92 CPX #S92 ;Analog input?
CF5B F0 2F BEQ POTS ;Go to POTS
CF5D 86 FF STX MASK ;Save input mask
;*****
; Interface Control
; for digital input
; Uses A-, X- and Y-registers
;*****
CF5F A9 32 SHIN LDA #S32 ;Set LOAD-IN
CF61 8D 01 DD STA UP ;Output to userport
CF64 09 08 ORA #S08 ;Set CLOCK
CF66 8D 01 DD STA UP ;Output to userport
CF69 A2 08 LDX #S08 ;Loop counter for 8 bit
CF6B 0A LOOP2 ASL A ;Shift left A-reg.
CF6C 2C 01 DD BIT UP ;Test DATA-IN
CF6F 10 02 BPL MAKEIN ;DATA-IN is low
CF71 09 01 ORA #S01 ;DATA-IN is high
CF73 A0 30 MAKEIN LDY #S30 ;Reset CLOCK
CF75 8C 01 DD STY UP ;Output to userport
CF78 A0 38 LDY #S38 ;Set CLOCK
CF7A 8C 01 DD STY UP ;Output to userport
CF7D CA DEX ;Decrement loop counter
CF7E D0 EB BNE LOOP2 ;End of LOOP2
CF80 25 FF AND MASK ;Mask bit
CF82 A8 TAY ;Save in Y-register
CF83 F0 02 BEQ CVAR ;Return 0
CF85 A0 01 LDY #S01 ;Return 1
CF87 20 A2 B3 CVAR JSR YFAC ;Convert to floating
CF8A 58 END CLI ;Enable interrupt
CF8B 60 RTS ;Return to BASIC
;*****
; Analog input
; If the argument is either $92 or $A2
; an analog input is invoked. Control
; jumps here.
; Uses A-, X- and Y-registers.
;*****
CF8C A9 FF POTS LDA #SFF ;Set Count register to $FFFF
CF8E 8D 04 DD STA TIL
CF91 8D 05 DD STA TIH
CF94 A9 B9 LDA #SB9 ;Set timer control reg.
CF96 8D 0E DD STA TIC
CF99 8E 01 DD STX UP ;Trigger One-shot
CF9C A2 3A LDX #S3A ;Reset trigger
CF9E 8E 01 DD STX UP ;Output to userport
CFA1 AD 04 DD TST LDA TIL ;Test timer low register
CFA4 A2 03 LDX #S03 ;Delay loop
CFA6 CA DELAY DEX
CFA7 D0 FD BNE DELAY
CFA9 38 SEC ;Subtract timer low reg.
CFAA ED 04 DD SBC TIL ;from previous value.
CFAD D0 F2 BNE TST ;Pulses still arriving?
CFAF A2 38 LDX #S38 ;Set CLOCK, reset LOAD-IN
CFB1 8E 01 DD STX UP ;Output to userport
CFB4 38 SEC ;Calculate contents
CFB5 A9 FF LDA #SFF
CFB7 ED 04 DD SRC TIL ;Low byte
CFBA A8 TAY ;in Y register
CFBB A9 FF LDA #SFF
CFBD ED 05 DD SBC TIH ;High byte
CFC0 20 91 B3 JSR AYFAC ;Convert A/Y to floating
CFC3 4C 8A CF JMP END ;Return to BASIC
.EN

```

## 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 mot4 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.
- The interrogation of analog input channel requires a perceptible amount of time. – An open analog channel (without any sensor connected to it) has been called in.

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

## Technical Data

fischertechnik COMPUTING interface for connection to the userport of the Commodore computers C64, C64-II, SX64, C128, and C128-D, Part No. 30566.

The interface may be used also on a wide range of other computers by connecting it with other adapters to parallel printerports and using different software packages.

### Output

Four outputs for connection of motors, lamps, electromagnets, etc. (M1 to M4).

Polarity of output (direction of rotation) controllable.

Load capacity: 1 Amp continuous, 1.5 Amp peak current.

Protection: On a short circuit of the output, an electronic fuse will shut down the output amplifier. The output amplifier will be powered up again on removal of the short circuit. On overloading the output the temperature in the output amplifier will rise. When exceeding the permissible temperature an electronic fuse will shut down the output amplifier. The output amplifier will be powered again, when the temperature has decayed.

Circuitry for monitoring the stream of data:

If no data signals are received from the Computer, the interface will disable all outputs after 0,5 seconds. The signals are stored in memory.

### Input:

Eight inputs for digital signals (E1 through E8). Due to internal wiring, the connection of electromagnetic devices (pushbutton switches, relay contacts) in positive logic as well as the connection of TTL outputs is possible.

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

### Power supply:

Operating voltage: 6 to 10 V unfiltered dc

voltage or 8 V stabilized dc voltage.

The use of the fischertechnik COMPUTING power supply is recommended.

### Connectors:

Interface cable fitting to userport adapter connector.

Power supply (twice), two jacks, 2.5 mm (approx. 1/10") diameter.

Ground socket, 2.5 mm (approx. 1/10") diameter.

Modell connector, 20 poles according IEC. Connector for additional interface.

### Dimensions:

150 x 90 x 30 mm (approx. 6 inches x 3.5 inches x 1.2 inches),

1200 mm (approx. 47 inches) cable length, transparent lid.

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

