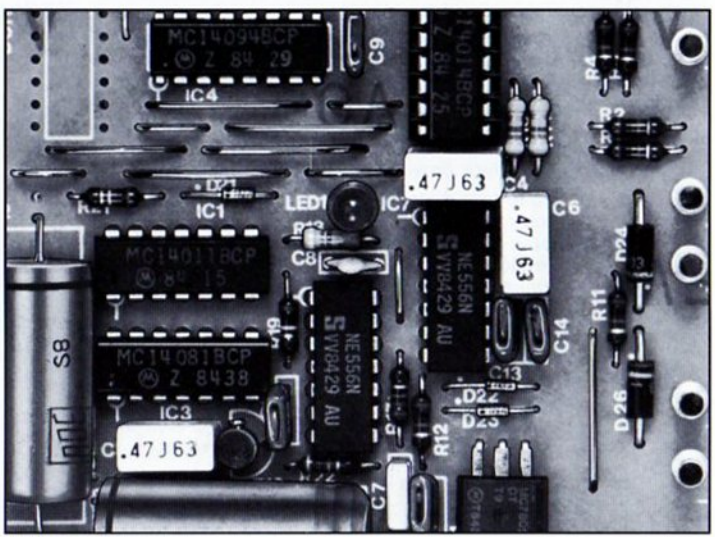
fischertechnik® COMPUTING

Interface Apple II Computer



Contents

The fischertechnik Computing Interface

Dear Friend of fischertechnik:

The fischertechnik computer interface has been developed to let personal computer owners explore the fascinating world of computer control and robotics.

Using the appropriate interface and software combination, along with the fischertechnik Robotic Computing Kit, computer owners can build unique working models that will learn various tasks and perform them under computer control. It's computers, it's robotics, it's building projects, it's fun *and* educational! In addition to family use, the projects are ideal for schools and for industrial prototypes and simulations.

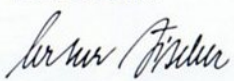
The hardware unit in this package is the interface that allows the computer and the model to talk to each other. It provides the input/output pathways that are needed to carry information *to* the computer and control signals *from* the computer to the motors and other operating parts of the models. The interface provides the following functions:

— Up to four motors, lamps, and magnets can be controlled.
— Input from as many as eight on-off switches can be monitored.
— Two additional inputs are available for sensing the values produced by variable resistance controls (potentiometers).

Hardware alone is, of course, only part of the solution. Also included with your interface kit is software on a floppy disk that interprets the signals received and sends the signals required to control the actions of the models. So that you can study and learn from these programs, written copies of all the software are provided in the programming/construction manual—study them, learn from them, and then (after making a safely stored archive copy) revise and expand them to develop your own new applications.

fischertechnik computing...it's a whole new world of experience with your computer and the fischertechnik system. Enjoy....

Yours sincerely,



Artur Fischer

Installing the Interface

The fischertechnik Apple II interface can be used with the Apple II, II+, and IIe computers, plus some Apple compatibles. It will not, however, work properly with the new Apple IIc, since this computer uses a different input/output design.

The fischertechnik interface is connected to the computer using the control-unit I/O port on the computer circuit board. You will have to open your computer's case to install the interface. Proceed as follows:

— MAKE SURE THE COMPUTER IS SWITCHED OFF AND DISCONNECTED FROM THE POWER SUPPLY.

— Open the cover of the computer by pressing inward on the upper section of the case at the rear until it releases from its catch and moves upward.

— Again, be sure the computer is switched off! The IIe is equipped with an internal power control lamp—this must NOT be illuminated.

— Now look at the right rear side. To the right of the line of long sockets, near the video and tape recorder sockets, you will find an unused 16-pin IC socket.

— Discharge any possible static electricity by touching the casing of the power supply.

— Now prepare your interface by removing it from its packing and freeing the connecting cable from its foam rubber protection. At the end of this cable, you will find a 16-pin plug. Its pins are quite delicate, so be careful not to bend them. If any are bent, straighten them *very carefully* with small pliers.

— On tne Apple IIe, lead the interface cable through a hole on the rear of the computer. With older models, thread the cable through one of the slots in the case.

— Now place the cable plug into the socket you identified earlier, with the cable pointing *to the right* toward the side of the case. Check carefully to be sure *all* 16 pins are fitted properly into the socket and then smoothly and evenly press the plug all the way in.

— Reinstall the lid of the computer.

— Next, prepare the fischertechnik power supply by attaching a red plug from the computing kit to the wire marked plus (+) on the power supply case, and a green plug to the wire marked minus (−). Use the small screwdriver from the kit to attach the plug, being careful not to overtighten the screw which will force the wire out of the plug. Then attach the power supply to the interface, using the red plug for the plus terminal on the interface and green for the minus. You can use either pair of sockets on the interface.

— Models are connected to the interface using the 20-wire multicolored flat cable supplied with the model kit.

— You may turn on either the computer or the interface first...it makes no difference. If you should wish to use the computer without the interface, simply unplug the power supply from the wall. There is no need to disconnect the interface from the computer.

— Please note the following:
As long as the interface is plugged in, you obviously cannot use any other accessory which would be plugged into the same socket.

Apple IIe owners must observe an additional precaution. Some of the signal lines used by the interface are also connected to the 9-pin socket on the rear of the computer, and to the Apple keys at the front of the keyboard. DO NOT use the Apple keys or plug anything into the 9-pin socket if the interface is switched on. Also, do not be surprised if your joysticks do not operate as usual while the interface is plugged into the computer.

Also be sure NOT to use the joystick firing button with the fischertechnik interface switched on as this could damage the interface.

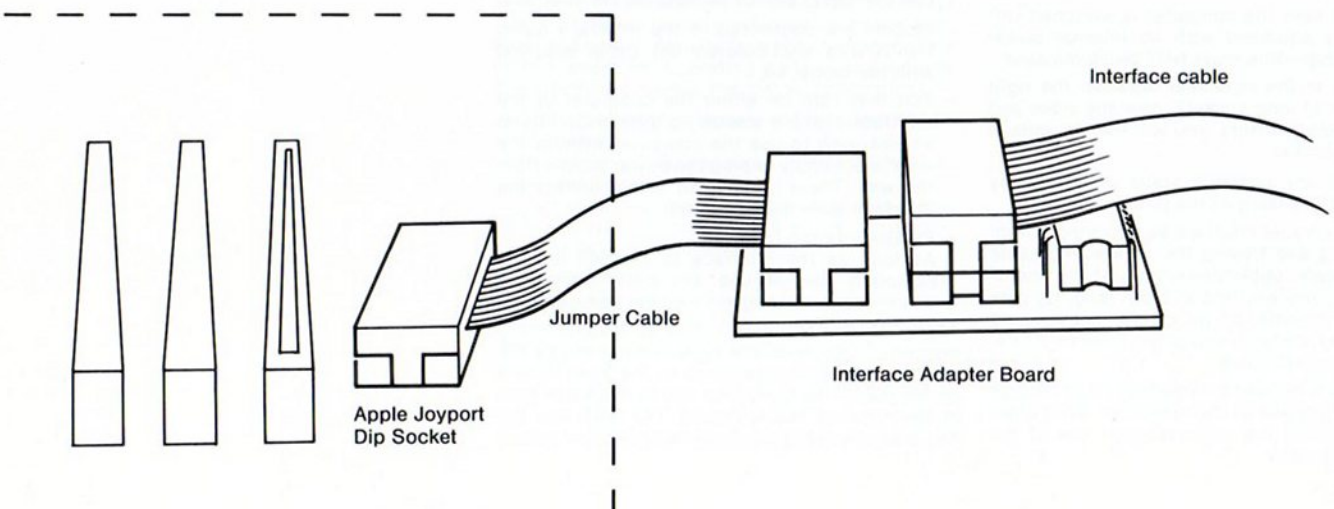
When working with the models, you should discharge any buildup of static electricity by touching a grounded object, like the case of the Apple power supply (at the rear, near the circuit breaker).

Enough of hardware...in the next section, we'll discuss fischertechnik computing software.

Instructions for connecting the Interface adapter between the fischertechnik Apple interface and a Apple IIe computer.

Connect the ribbon cable coming from the interface to the dip socket on the adapter board as shown below. The cable should lay to the right of the adapter board. The ribbon cable coming from the adapter board is then plugged into the Apple IIe joyport dip socket. Peel the protective tape off of the back of the adapter board and stick it to the back of the Apple IIe case and route the interface cable through one of the slots in the back of the computer. The fischertechnik interface will now work correctly with either type of Apple IIe computer.

NOTE: Disconnect the adapter board from the joyport dip socket when not in use.



fischertechnik Computing Software

Anyone who has thought of controlling appliances or models with a computer will know that this is not the easiest of tasks. You must be familiar with the input/output structure of the computer, with the requirements for a suitable interface, and with programming in order to make it all work. Until now, that is. fischertechnik computing provides the proper interface for your computer, and with it, the software (programs) you need to make it all work smoothly and easily.

If you have not as yet connected the interface to your computer, we suggest that you do so now. Then place the fischertechnik computing disk in the disk drive and turn on the computer. After a few seconds, a message will appear on the screen, along with the directory of the disk.

Now load the basic program DRIVER.AII from the disk. When the program has loaded, type RUN. After a few moments, the computer will show READY. Although nothing seems to have happened, your Apple now contains some new commands which were not there before. These commands have been designed to mate the fischertechnik computing interface to the computer so that the operation will be smooth and simple for you. Instead of detailed programming knowledge, the few BASIC commands we'll describe here will give you complete control over your fischertechnik computing models.

The motor output M1 is controlled as follows:

CALL M1, CW
(Clockwise)
CALL M1, CCW
(Counterclockwise)
CALL M1, OFF

The same commands offer similar control over outputs M2, M3, M4.

The status of the ten input lines is checked by means of the BASIC USR function. The value of USR(E1) = 1 if the line carries +5 volts; otherwise, the function returns a value of 0. In the same way, the status of the other digital inputs can be checked by the functions USR(E2)...USR(E8).

The analog inputs EX and EY are connected to +5 volts through one potentiometer (4.7K ohms) each. The functions USR(EX) and USR(EY) will have a range of values from 0 to 255, depending on the position of the potentiometer. Using this information, the motion of a robot arm or other turning device can be measured if it is connected in such a way as to turn the potentiometer, and if the program repeatedly calls the function to measure the value.

The last of the new commands is **CALL INIT.** This is used to bring the interface to an initial state, ready to begin a new cycle of commands. It may also be used for switching off all motor channels at once.

Let's try a brief practical experiment at this point. Connect a fischertechnik motor to output circuit M1 using the flat 20-wire cable from the kit. See page 8 of the Programming/Kit-Building Instructions book for instructions on preparing the cable for use. Use the orange and yellow wires from the part of the cable farthest from the three longer wires. Now type in the command CALL M1, CW and press RETURN.

The motor should start and run for a short time, then stop. Congratulations! You've just learned how to control even the most complicated fischertechnik models with your computer.

At this point, you might be wondering why the motor stopped. You told it to run, but you didn't give it a command to stop. Isn't a separate command required for stopping? The answer is not always. In fact, the interface still stores the command that tells the motor to run. But the interface itself has also decided to halt the action. This will always occur if no new command is received within a half-second of a previous command, for reasons of safety.

Imagine for a moment that you are testing a new program. If you are like most of us, it is almost certain that there will be some error, and the computer will stop, perhaps displaying a message such as

SYNTAX ERROR IN LINE....

Without the built-in protection, if you had given the command to start the motor, it would continue to run, possibly demolishing that beautiful model you worked so hard to build. It's nice to know that the system will help prevent this type of disaster. This protection will also come in handy if you should interrupt program operation with a Control C.

If you resume operation by using the CONT command, the first command sent to the interface by the program will "wake up" the interface, and the program will continue to flow as if nothing had happened.

Whether or not the interface is activated by input or output commands can be checked by a glance at the LED on the interface. It not only indicates voltage but also shows if the interface is operative.

Now let's try an input command. Connect a switch between wire E1 (the long brown wire at the edge of the cable) and +5 volts (the red wire at the middle of the cable). Now enter the following test:

PRINT USR(E1)

Depending on whether the switch between E1 and +5V is closed or open when you press the

RETURN key of the computer, a 1 or a 0 will appear on the screen. If the motor is still connected to the cable, it will move again, as input commands also reactivate the interface.

Now connect a potentiometer between EX and +5V. The cable diagram on the back page will help you identify the proper wire to use. Turn the slider of the potentiometer to a centered position, and then type:

PRINT USR(EX)

The number which appears on your screen should be between 0 and 255.

If you have entered the BASIC routine by hand from the printed listing, you should now save it to tape or disk, as your equipment permits. You will need to load and run this program first before entering the programs that control the individual fischertechnik computing models, since this routine installs the command set you will need.

Now let's write a small program to observe more about the action of the potentiometer. Since the starting routine is already in the computer from lines 1 through 500, we will start with:

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

It will take a moment as the main routine runs, and then the screen will fill with numbers. Now take the potentiometer and carefully turn the center wheel to vary the position of the slider. Turn from one end to the other, and observe the changing numbers. They should range from 0 to 255, although it is possible that the value of 255 will not be reached. To end this test program, press CTRL-C.

For those of you who want a deeper understanding of how this all works, read on. The purpose of the first BASIC program is to write a short machine language program into an unused area of the computer's memory by using the DATA lines of the program. While this is being written to memory, it is simultaneously being verified with a checksum to discover if any error may have occurred in the transfer of the numerical values. The machine language program occupies the address range $300-3A7.

In addition, the standardized parameters INIT M1, M2, M3, M4, CW, CCW, OFF, E1, E2, E3, E4, E5, E6, E7, E8, EX, and EY are set, and the entry point of the USR function is set.

For BASIC programs which you may write yourself to control your models, the following restrictions must be observed:

You cannot use the reserved BASIC key words like PRINT or STOP as variables, nor can you use the parameters listed above. Also, remember that Applesoft can distinguish only the first two letters of variable names, so variable names such as EXTRA or M114 are also unusable since they start with reserved characters that are the same as the command parameters.

This restriction applies only to variables of the same type, so string or integer variables such as:

M1% or **EXTRA$**

may be used.

Obviously, the USR function cannot be used for other purposes in your programs.

The memory area $300-3A7, used by the machine language program, cannot be used for other purposes.

The last line of the startup routine activates the interface and clears it of all commands. This is caused by the command:

500 CALL INIT

This means that control of an active interface is passed to a user program at this point. All sample programs furnished on the disk are numbered above 500 so that they may be loaded and run after the starting routine. Any programs which you write should follow this format. We suggest you study the programs we furnish for hints and ideas to use in designing your own programs.

Interface Program

```
1  HOME
5  PRINT "LOADING DRIVER"
10  REM INTERFACE PROGRAM FOR APPLE II
20  REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
30  REM COMMANDS FOR THE DRIVER :
40  REM CALL M1,CW          CALL M1,OFF
50  REM CALL M1,CW          CALL M1,CCW
60  REM USR(E1)...USR(E8) , USR(EX) , USR(EY)
70  REM M1    M4 ARE MOTOR SELECT COMMANDS
80  REM E1    E8 ARE DIGITAL INPUTS
90  REM EX AND EY ARE ANALOGUE INPUTS
100  DATA 768,169,0,240,48,169,3,208,10,1615
110  DATA 169,12,208,6,169,48,208,2,2437
120  DATA 169,192,120,141,167,3,169,44,3442
130  DATA 32,192,222,173,166,3,13,167,4410
150  DATA 45,167,3,141,167,3,173,166,6236
160  DATA 3,77,167,3,141,166,3,168,6964
170  DATA 162,8,44,94,192,44,92,192,7792
180  DATA 14,166,3,144,3,44,93,192,8451
190  DATA 44,95,192,202,208,236,44,89,9561
200  DATA 192,44,88,192,140,166,3,88,10474
210  DATA 96,120,32,12,225,166,161,224,11510
220  DATA 6,240,51,224,7,240,47,142,12467
230  DATA 167,3,44,94,192,44,91,192,13294
240  DATA 44,95,192,44,90,192,162,8,14121
250  DATA 10,44,97,192,16,2,9,1,14492
260  DATA 44,94,192,44,95,192,202,208,15563
270  DATA 239,45,167,3,168,240,2,160,16587
280  DATA 1,32,1,227,80,96,44,94,17170
290  DATA 192,44,95,192,138,41,1,170,18043
300  DATA 32,30,251,76,145,3,0,0,18580
310  DATA 1,2,4,8,16,32,64,128,18835
320  DATA 7,6,255,170,85,85,76,89,3,19611
330  READ INIT: LET M1 = INIT
340  FOR M3 = 0 TO 20: FOR M2 = 0 TO 7
350  READ M4: POKE INIT + M3 * 8 + M2,M4
360  LET M1 = M1 + M4: NEXT
370  READ M4: IF M1 <  > M4 THEN  PRINT "DATA ERROR
     IN LINE ";100 + 10 * M3: END
380  NEXT
390  READ E1,E2,E3,E4,E5,E6,E7,E8
400  LET M1 = M1 + E1 + E2 + E3 + E4 + E5 + E6 +
     E7 + E8
410  READ M4: IF M1 <  > M4 THEN  PRINT "DATA ERROR
     IN LINE 360": END
420  READ EX,EY,OFF,CCW,CW,CW,M2,M3,M4
430  POKE 10,M2: POKE 11,M3: POKE 12,M4
440  LET M1 = M1 + EX + EY + OFF + CCW + CW + CW +
     PEEK (10) +  PEEK (11) +  PEEK (12)
450  READ M4: IF M1 <  > M4 THEN  PRINT "DATA ERROR
     IN LINE 320": END
460  LET M1 = INIT + 4:M2 = M1 + 4:M3 = M2 + 4:M4
     = M3 + 4
500  CALL INIT
```

The Diagnostic Program

Once you complete a fischertechnik computing model, you may sometimes find that things do not operate exactly as you expected. Which is not so surprising, considering the large number of wires running between model and interface. If only one switch is not where you expect it to be, complications could arise. And the situation gets even harder to figure out when you've written the programs yourself. Where do you look? Is it a hardware problem? Is the software to blame?

In order to help you test the hardware clearly and easily, the diagnostic program has been developed. It is on the fischertechnik computing floppy disk as DIAGNOSTIC.AII. This program should always be loaded for testing a new model and/or new program. With DIAGNOSTIC, you can test all inputs and determine if their action is what you expect it to be.

Control outputs are selected with the numerical keys. They are displayed inversely on the screen, so you can switch on (right-hand or left-hand rotation) and off the selected output. In this way, you can determine not only if a motor starts at all, but whether it turns in the desired direction. If not, you simply have to reverse the two motor wire connections.

C switches off all motors, and X allows you to exit the program.

```
500  CALL INIT
600  REM
610  REM FISCHERTECHNIK COMPUTING
620  REM
630  REM DIAGNOSTIC PROGRAM
640  REM
650  REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1984
660  REM
800  REM FUNCTION DESCRIPTION
810  REM  THIS PROGRAM CHECKS ALL FUNCTIONS OF THE MODELS.
820  REM ALL INPUTS DISPLAYED
830  REM ALL OUTPUTS ARE CONTROLLED VIA THE KEYBOARD
900  HOME
910  PRINT "FISCHERTECHNIK"
920  PRINT "COMPUTING"
930  PRINT
940  PRINT "DIAGNOSTIC PROGRAM"
950  PRINT
1000  DIM STA(4),STA$(4): REM MOTOR STATUS
1010  FOR I = 1 TO 4
1020  LET STA(I) = OFF:STA$(I) = "OFF"
1030  NEXT I
1050  DIM M(4): REM 6502-PROG. ADRESSES FOR MOTORS 1-4
1060  LET M(1) = M1:M(2) = M2:M(3) = M3:M(4) = M4
1070  LET M = 1: REM ACT. MOTOR
1110  PRINT "C O M M A N D S"
1120  PRINT
1130  PRINT " C : ALL MOTORS OFF"
1140  PRINT " A : ACT. MOTOR OFF"
1150  PRINT "  Q : ACT. MOTOR CCW"
1160  PRINT " P : ACT. MOTOR  CW"
1170  PRINT " X : EXIT"
1180  PRINT "1-4: SELECT MOTOR NUMBER"
1190  PRINT : PRINT
1200  PRINT " E1 E2 E3 E4 E5 E6 E7 E8   EX  EY"
1210  PRINT
1230  LET CLR$ =     " +  CHR$ (8) +  CHR$ (8) +  CHR$ (8)
1240  VTAB 17
1250  HTAB 2: PRINT CLR$; USR (E1);
1260  HTAB 5: PRINT CLR$; USR (E2);
1270  HTAB 8: PRINT CLR$; USR (E3);
1280  HTAB 11: PRINT CLR$; USR (E4);
1290  HTAB 14: PRINT CLR$; USR (E5);
1300  HTAB 17: PRINT CLR$; USR (E6);
1310  HTAB 20: PRINT CLR$; USR (E7);
1320  HTAB 23: PRINT CLR$; USR (E8);
1330  HTAB 28: PRINT CLR$; USR (EX);
1340  HTAB 32: PRINT CLR$; USR (EY);
1400  LET K =  PEEK ( - 16384) - 128
1410  IF K < 0 THEN 2000: REM NO KEY PRESSED
1420  POKE  - 16368,0
1500  IF K =  ASC ("C") THEN  FOR I = 1 TO 4: LET STA(I) =
      OFF:STA$(I) = "OFF": NEXT
1510  IF K =  ASC ("O") THEN  LET STA(M) = OFF:STA$(M)   "OFF"
1520  IF K =  ASC ("Q") THEN  LET STA(M) = CCW:STA$(M) = "CCW"
1530  IF K =  ASC ("P") THEN  LET STA(M) = CW:STA$(M) = "CW "
1540  IF K =  ASC ("X") THEN  VTAB 23: END
1550  IF K > 48 AND K < 53 THEN  LET M = K   48
2000  FOR I = 1 TO 4
2005  CALL M(I),STA(I)
2010  IF I = M THEN  INVERSE
2020  VTAB 20: HTAB 5 * I - 3
2030  PRINT "M";I; CHR$ (8); CHR$ (8);
2040  VTAB 22
2050  PRINT STA$(I);
2060  NORMAL
2070  NEXT I
2200  GOTO 1240
```

Applesoft BASIC

The fischertechnik computing programs on the disk we supply are written in Applesoft BASIC. The program listings in the booklet of fischertechnik computing programming instructions, however, have been written in a more general form of BASIC. We have tried to reduce the number of machine-specific instructions, but it was impossible to eliminate them all. You will, therefore, have to examine the printed programs and modify them where required for use with the Apple II. These are the main differences in commands:

fischertechnik computing programs	Applesoft BASIC
SYS M....	**CALL M....**
SYS INIT	**CALL INIT**
PRINT CHR$ (147)	**HOME**

In addition, because of the graphic commands which are part of Applesoft, some conflict may arise with naming the variable. For example, the commands GR and RED are included in the variable names GREEN and RED, which will be found in the Traffic Light program. In such cases, modify the programs to change the variable names: GN for GREEN and RD for RED.

These changes *have already been made* for you in the programs supplied on the disk; the notes above apply only to programs you enter yourself from the printed listings. The disk programs may also vary from the printed listings in other details where the change offers some advantages. You can, of course, print listings for yourself of the disk programs we have supplied.

Troubleshooting Your System

If at any time the fischertechnik computing interface should not operate as you expect, we suggest you check the following items by using the DIAGNOSTIC program:

- The interface shows a 1 value at E1 through E8 even though no model has been connected—indicates that the interface is not connected to the computer or to the power supply.
- One of the inputs E1 through E8 shows the reverse effect from what you would expect when opening or closing a switch—check the operation of the switch to be sure it really is opening or closing when you expect it to.
- One of the inputs E1 through E8 always shows the value 0 even though it is connected and activated—check for a broken or improperly connected wire.
- One of the inputs E1 through E8 always shows a value of 1 although no model has been connected—probably the input gate IC 4014 has been damaged by excess voltage or static electricity.
- One motor output does not work—check the cabling.
- One motor output works in one direction only—the motor is defective.
- Motor operates very slowly or fails—be sure that you are using the fischertechnik power supply and that it is properly connected. Do not add electrical accessories to the set in addition to the four motors and four lamps.
- Value of a potentiometer is always 255—check the cable. If using variable resistors other than the ones that come with your kit, the resistance value must range between 0 and 5K (5,000) ohms.

Technical Data

fischertechnik computing interface for Apple II, Part No. 30569.

Four outputs, M1 through M4, for the control of motors, lamps, solenoids.

Polarity of output (direction) controllable.

Load capacity: 1.5 amps peak, 1 amp continuous.

Eight inputs, E1 through E8, for digital signals. Internal circuitry permits the connection of keys, switches, and relays in positive logic as well as Transistor Transistor Logic (TTL) outputs. Protection against overvoltage included.

Two inputs, EX and EY, for analog signals. Devices with resistance values ranging between 0 and 5,000 (5K) ohms may be used, including potentiometers, phototransistors, etc.

Safety circuitry—if no data signals are received from the computer within 0.5 second of the last command, all outputs are switched to the inactive state (off). The last active command remains stored. The guard circuit will also operate, without delay, in the case of serious syntax errors in the programming. It will also operate if low voltage is detected, either from overload or by defect of the power supply.

With the interface unit, you will receive a floppy disk containing software with interface control commands and sample programs for the operation of the fischertechnik computing models.

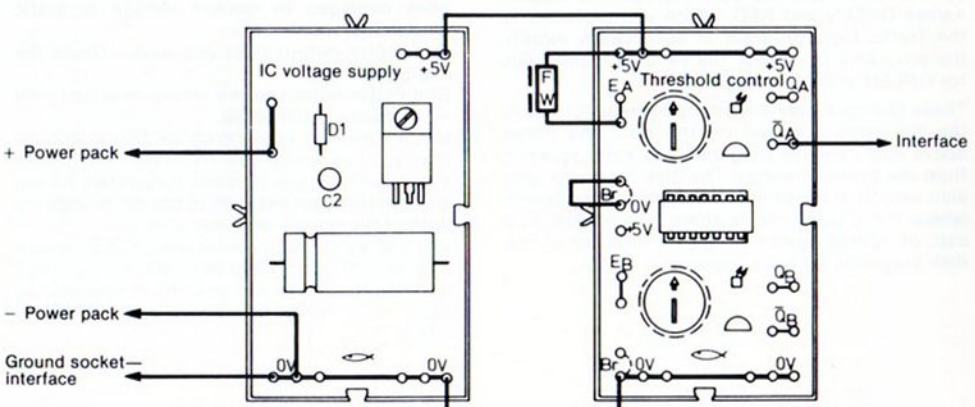
Other Applications

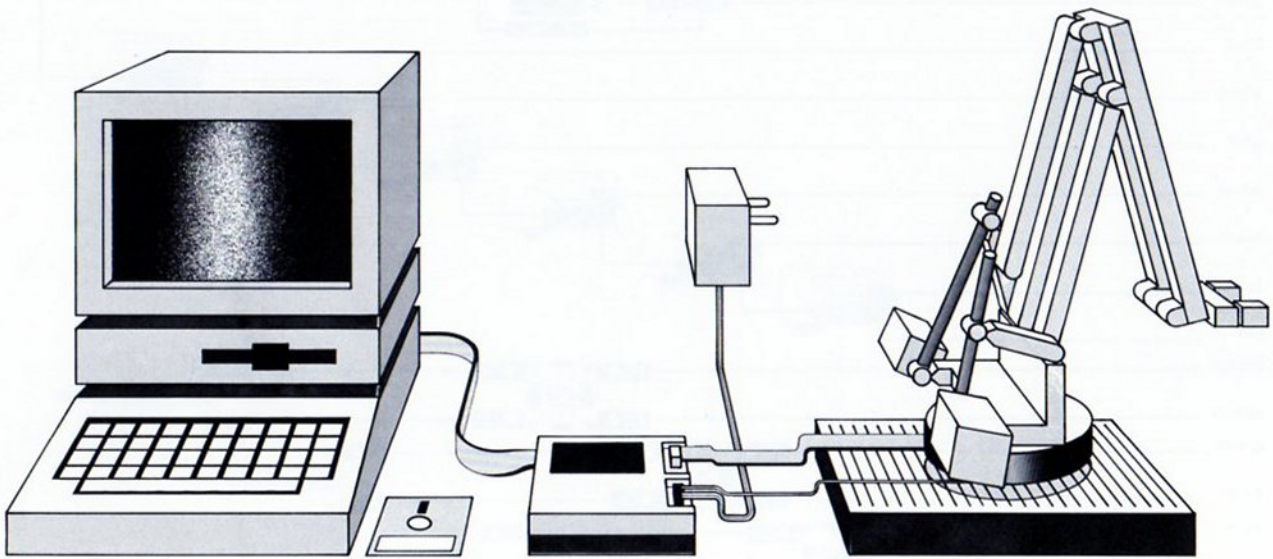
The fischertechnik computing interface is, of course, compatible with the components and electronic modules of the fischertechnik computing kits. In addition to the mini-switches supplied, you may also connect switches of other types and designs; that is, larger switches, toggle switches, or relay-type switches may be used. Be careful when using switches in which contact bounce may occur. 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 may be wired to any sensor featuring as output a resistance between 0 and 5K ohms. In addition to the potentiometers supplied with the kit, this would include similar devices of other designs, or more sophisticated sensors such as photoresistors (phototransistors).

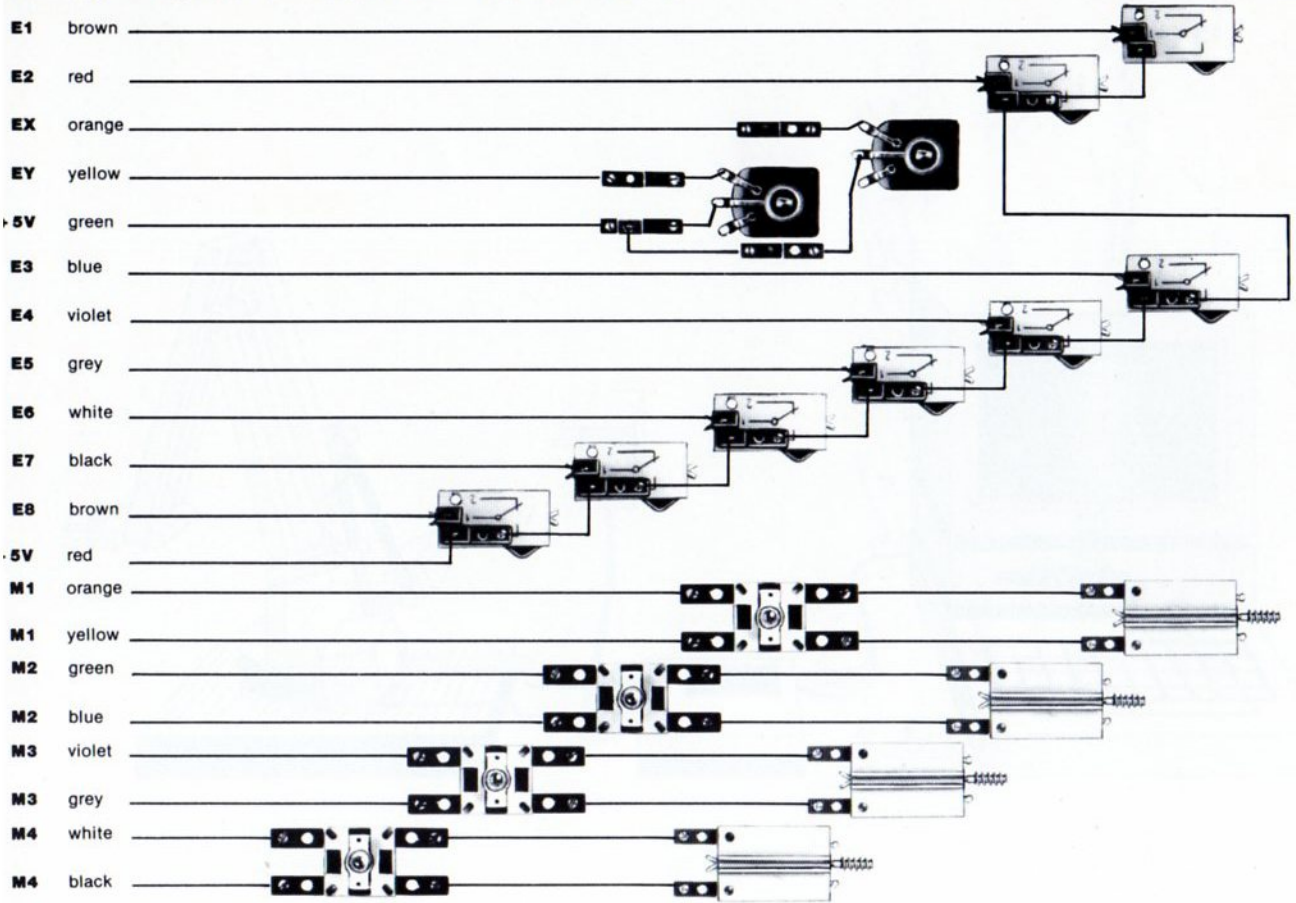
The motor outputs of the interface feature high load capacity. In addition to the fischertechnik mini-motors, the S-motor and N-motor can be used, and a small lamp for indication of function may be wired in parallel with the motor.

The signals of electronic modules with TTL logic outputs may also be connected to the interface. The ground circuit of the module must be connected to the ground of the interface in order to establish a reference point for the circuit. The illustration shows the design of a light-sensitive switch. The variable resistor sets the operating threshold of the photoresistor.





Wiring Diagram for Interface Inputs/Outputs



Litho in U.S.A. 9/89 Art. No. 39491