Page 1 -- Front cover

Interface for Acorn Model B Computer

Page 2 -- Contents

Page 3

FISCHERTECHNIK COMPUTING INTERFACE

[Translator's note: A translation of Artur Fischer's letter has already been done, and I feel that it would be better not to "improve" it]

Page 4

CONNECTION OF THE INTERFACE

The Fischertechnik Acorn computer interface (Fischertechnik Part No. 30564) is for the British Broadcasting Corporation microcomputer system, made by Acorn Limited, which we will call the 'BBC computer'. The computer was developed for a television series in Great Britain and is becoming steadily more popular in Germany as well. Usually, it is found as the model B version. There is also a Model A version, which is a stripped down version with less memory and less external connections. Normally, the model A has to be expanded in order to be compatible with Fischertechnik Computing. There is another model, the "model 64" which includes a second 6502 microprocessor and 64k of RAM. It goes without saying that this high-performance computer is fully compatible with Fischertechnik Computing.

The Fischertechnik Computing interface is connected to the User Port. This is done as follows:

* Make sure that the computer is switched off.

* Turn the computer upside down to reveal the external connections.

* The User Port is in the middle of the row of connectors and is clearly labelled "User Port".

* In order to prevent a build up of static electricity in your body, which might harm the interface, touch the outer sleeving of the Video Out socket on the rear of the computer briefly with a finger.

* Now prepare the interface. The 20 way ribbon cable which is fixed to the interface must be connected to the User Port.

* Now the socket on the cable has to be brought into contact with the connector on the computer so that the bump on the socket fits into the cutout on the connector shell. Check carefully that all 20 pins fit into the socket and then press the socket into the connector shell. The ejector levers will click shut when the connection is properly made.

* Now you can turn the computer the right way up again and lead the cable neatly under the bottom of the computer and out at the back.

* Now connect the Fischertechnik Computing interface to a suitable power supply. The interface requires a Direct Current voltage ranging between 6 and 10 Volts. Thus, you have to connect one of the sockets marked "+" on the interface to the positive "+" socket of the power supply unit and similarly with the "-" line. It does not matter which of the pairs of sockets on the interface you use. The power connection sockets are duplicated because not more than two motors should be driven simultaneously using the Fischertechnik power supply. With larger models, therefore, it is necessary to use two mains units. In that case, the mains units are coupled, and the available current is doubled.

Older versions of the Fischertechnik power supply have an A.C. outlet. Do not use it on the Fischertechnik Computing interface! Use the controllable front D.C. outlet which should be turned up to maximum. On all Fischertechnik power supply apparatuses the "-" socket is found where the rotary control knob points. One thing at least need not worry you: if you mix up the power connections, neither your interface nor your computer will be damaged. The interface is protected against a reversed power supply.

* Connect your Fischertechnik Computing model to the interface. For this purpose, the assembly sets and the Fischertechnik Computing assembly kits include a twenty way ribbon cable. This cable may also be available as a separate part in a Fischertechnik Service Set.

* IMPORTANT NOTE! The connectors for the model and the User Port are the same type of plug. Never, never, connect the model directly to the computer! The models must always be driven via the interface!

* It does not matter whether you switch on the interface first or the computer. When you are not using the interface and are running other programs, you need not disconnect the interface cable. When not in use, the interface simply stays switched off.

* Side effects of the interface: as long as the interface is plugged in, you cannot of course use any other devices which also use the User Port.

Before operating a model you should, forthe sake of safety, always discharge any possible electrostatic charge in your body by touching an earthed metallic object, for example a radiator, or as already described above, the outer sleeving of the Video Out socket on the back of the computer.

We have now finished with the hardware. The next section will deal with the Fischertechnik Computing software.

Page 5 -- View of underside of computer

Page 6

Fischertechnik Computing software

Anyone who has tried to control machines or models using a computer will have found out or will have been told by his friends that things are not quite so easy. You would have to have a detailed knowledge of the computer, the microprocessor and the interface, as well as being familiar with machine code. Up until now, this was true, and many people were prevented from tackling this interesting area of computing. From now on, this no longer need be the case. The package containing the interface includes this documentation and software on a floppy disk. One of the programs is particularly important, the so-called basic driver routine.

By now you should have connected the interface to your computer as described in the previous section. If not, do it now. Switch the computer on and insert the Fischertechnik Computing floppy disk into the drive unit.

Type the following line on the keyboard:

CHAIN "FISCHER"

After a brief pause a message will appear on the screen, followed by the catalogue of the floppy disk.

At this point, we will spare a few thoughts for those people who do not own a floppy disk drive unit or Disk Filing System. We are sorry to have to ask them to take their Fischertechnik Computing floppy disk along to someone who has a BBC computer with a disk drive. Using that, you can load all the programs individually from the disk and then record them onto a blank cassette. When the programs are safely on tape you will be ready to start, as the Fischertechnik Computing programs do not actually require the disk operating system. If the worst comes to the worst, you will have to copy from the program listings found in this and other Fischertechnik Computing documentation.

You may now load the computer with the basic routine DRIVER.

LOAD "DRIVER"

After typing the command "RUN", the usual prompt symbol will return after a short pause during which two numbers are printed on the screen. Externally, nothing seems to have changed, and yet your BBC computer now possesses some new commands which were not part of BASIC before. These commands are specifically for the BBC computer and the Fischertechnik Computing interface. The basic driver routine embodies the above mentioned detailed knowledge of the computer modules in machine code.

Instead of knowing about the computer in detail, you will only have have to remember the following BASIC statements:

The motor output M1 is affected by:

```
CALL M1, GO%      CALL M1, HALT%
CALL M1, CW%      CALL M1, CCW%
```

The command parameters define the motor and the operating mode. GO% means "Switch on", HALT% means "Switch off". CW% (ClockWise) effects a rotation of the motor in a clockwise direction, CCW% (CounterClockWise) in an anti-clockwise direction.

The same commands, but using M2, M3 or M4, control the remaining three outputs. It is also worth remembering that GO% always starts a motor in a clockwise direction.

The ten digital inputs are polled with BASIC's USR function. This function, however, returns additional information which is not required and which has to be truncated using a mask bit. Because of this, it is better to use the user-defined function FNUSR:

```
FNUSR (E1)
```

returns 1 if the input E1 on the interface is connected to +5V. Otherwise, FNUSR (E1) returns the number 0. Similarly, FNUSR (E2) ... FNUSR (E8) return the status of the other digital inputs.

The analogue inputs EX and EY are connected to +5V via potentiometers (4.7 kOhm) each. The functions

```
FNUSR (EX)
FNUSR (EY)
```

will then return a value ranging between 0 and 255, depending on the positions of the potentiometers.

If, for instance, the arm of a robot is driven by a motor and the potentiometer EX is moved by the motion of the arm, a program can precisely follow the motion of the robot by repeatedly calling the function

```
FNUSR (EX)
```

However, you should never use these commands if there is no potentiometer actually connected to the respective line! Otherwise the computer would wait forever for a signal and you could only release the computer from this state by pressing the emergency key BREAK. Your program would be erased too! So remember, to prevent losing your program, it should always be saved onto disk or cassette before entering the "RUN" command.

The last of the new commands is:

CALL INIT

This is used to put the interface into a well defined initial state. It may also be used if all motor channels are to be switched off at once.

Page 7

Now we can get to work. Connect a Fischertechnik motor to the M1 terminals on the interface, using the 20 way colour-coded ribbon cable provided. M1 uses the yellow and orange cores in the upper half of the ribbon cable. Now type in:

CALL M1, GO%

The motor will start briefly and then stop again. Savour the moment, because you have controlled some quite complicated Fischertechnik equipment for the first time using your BBC computer.

There remains the question why the motor stopped again. We switched it on, and have not yet given the separate switching off command described above, but the motor stopped anyway. However, the interface still "knows" that it ought to be running, but has decided to "go to sleep". This will always happen if within half a second no new command is given. This is done for safety reasons. Imagine yourself to be testing a new program: it is almost certain that somewhere in the program there would be an error, in which case the computer would stop with an unwelcome message like:

MISTAKE AT LINE ...

If this happened, the motor which had just been switched on would not stop and would begin to demolish your carefully built model. You would have to rush to the power supply to switch it off as quickly as possible.

[listing of DRIVER here]

Page 8 (left)

It is comforting to know that the motor will always stop automatically. This will also happen if you stop the program by pressing the ESCAPE key.

If the program is continued (for example with a GOTO statement) the first command encountered will "wake up" the interface and will not have forgotten any of the motors. The program will continue as if nothing had happened.

This feature, that the interface switches itself off, has been chosen deliberately. There will always be some sort of time lag between input/output statements, and half a second of running should not cause any major disasters. A glance at the LED on the interface will show if input/output is taking place. The LED thus shows whether the interface is in use and acts as a check that the correct supply voltage is being used.

Now let's have a brief look at the input commands. Connect a switch between E1 (the brown cable core at the bottom) and +5V (the red cable core in the middle of the ribbon cable).

Now type:

```
PRINT FNUSR (E1)
```

Depending on whether the switch between E1 and +5V was being pushed when you pressed RETURN, a 1 or a 0 will appear on the screen. If the motor that you tested before is still connected to the output it will now move again briefly. So you can see that input commands will reactivate the outputs of the interface as well as output commands.

Now connect a potentiometer (value 4.7 kOhm) between EX and +5V. Turn the knob about half way round and type:

```
PRINT FNUSR (EX).
```


The resulting number which is printed on the screen should be in the range 0 to 255.

If your computer does not have a disk system and if for any reason you have not yet obtained a tape copy, you will have had to type in the whole of the basic driver routine manually. You should save it now onto either tape or floppy disk. This routine will always be needed to run models using the Fischertechnik Computing interface, as each program has to start with the routine in order for the new commands to be installed.

So as to make it easier to check the potentiometer, we shall now write our first Fischertechnik Computing program. The basic driver routine should already be in the computer occupying line numbers 1 to 500. If not, LOAD it. Now type:

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

There will be a short pause while the basic driver routine is assembled and ... here it comes! Suddenly the screen is filled with numbers which scroll upwards and off the top of the screen. If you now hold the potentiometer and turn the knob you will see that the numbers change. Rotate the potentiometer from one limit to the other. The resulting numbers should be in the range 0 to 255, although these figures themselves might not quite be reached.

To end the program you have to press ESCAPE. For those who want to learn more about the individual functional sequences and who do not only want to use the programs supplied on the floppy disk, we offer here some more detailed information. The basic driver routine loads a short piece of machine code into the storage area reserved by the variable M%. The machine code is written in BASIC's built-in Assembly Language, the source code for which can be examined. Because the code is reassembled at each run, this provides an automatic check on any syntax errors which might have crept in during copying.

Apart from the machine code itself, the basic driver routine also defines the standardized parameters INIT, M1, M2, M3, M4, CW%, CCW%, GO%, HALT%, E1, E2, E3, E4, E5, E6, E7, E8, EX and EY. The bit mask for the USR function is set and the function FNUSR is defined. If you are going to write your own BASIC programs, you should take note of the following restrictions: You should not use the standard parameters mentioned above or BASIC reserved words (like PRINT or STOP) as variable names. If you do use the standard parameters for your own variables, the interpreter will not notice the error, as they are not BASIC reserved words.

* You should not attempt to define a further function using the name FNUSR.

* The above mentioned storage area (M%) cannot be used for other purposes.

The last function of the basic driver routine consists in switching on the interface and switching off all the outputs. This is done with the command:

```
500 CALL INIT
```

This means that the interface will be initialised and transferred in an operative state to the user's program which should be numbered to follow line 500. All sample programs on the floppy disk are designed according to the same principle. You should study them thoroughly for hints and ideas.

The Test Program

When you have built a Fischertechnik Computing model, it is possible that it does not work exactly as you expected. Small wonder when you consider the large number of wires leading between the model and the interface! Even if only one switch is wrongly wired, strange and perplexing results can be experienced. If you have written the controlling program yourself, the problem is twice as hard: where should you start in your quest for the fault, in the hardware or the software?

The test program was developed so that you can test your hardware easily and confidently. It is called DIAGNOS on the Fischertechnik Computing floppy disk. Always load and use this program to test your models. With it you can see the state of all the inputs at once, and check whether they are acting as you expected or not.

Using the number keys 1 to 4 on the keyboard, choose one of the outputs. The chosen output is shown on the screen. You can now switch the chosen output on (clockwise or anticlockwise) or off. In this way, you can not only check that the motor runs, but also that it is running in the right direction. If it runs the wrong way, simply swap the leads to the motor in question.

The "C" key switches all the motors off, and "X" ends the program.

[listing of DIAGNOS lines 500 onwards in here]

SPECIAL FEATURES OF ACORN BASIC

The Fischertechnik Computing programs on the floppy disk supplied are written in BBC BASIC. The program listings included in the Fischertechnik Computing programming instructions, however, are written in BASIC on another computer. We have tried to keep the number of computer-specific features and characteristics as small as possible. You should check the printed programs for possible changes in any case. The most important lines are identified by an asterisk before the line number. A list of the main differences follows:

Fischertechnik Computing Instructions for Programming	BBC computer BASIC
SYSM.....	CALL M.....
SYS INIT	CALL INIT
USR(E..)	FNUSR (E..)
PRINT CHR$(149)	CLS or MODE 6

Add to this that on account of the undemanding nature of BASIC, conflicts may arise if variables are named badly. All these alterations, however, have already been carried out in the programs on the floppy disk. Moreover, the floppy disk programs may also differ from the printed documentation in other details where small improvements have been made.

CHECK LIST

If your Fischertechnik Computing interface does not work at all, or does not work as expected, please check the following list using the test program DIAGNOS before assuming that your interface is faulty:

If all inputs E1 to E8 show a 1 although nothing is connected to them, then the interface is either not connected to the computer, or to a suitable power supply.

If one of the inputs E1 to E8 always shows the opposite result to the one expected, then the switch is wired incorrectly. It is wired "normally open" when it should be "normally closed", or vice versa.

If one of the inputs E1 to E8 always shows a 0, even though a switch is connected and pressed, check the wiring carefully.

If one of the inputs E1 to E8 always shows a 1, even when no model is attached, then possibly the 4014 chip on the interface is damaged by an overload (electrostatic charge).

If one motor does not run, check the wiring carefully.

If one motor will only run in one direction but not the other, then the motor is probably defective.

If the motors run very slowly or keep stopping, then either the power supply is overloaded by trying to run too many motors at once (in which case use an additional Fischertechnik power supply), or the variable output of the power supply is not turned up high enough.

If other devices with a higher resistance are used instead of the potentiometers supplied, it is possible that the value is higher than shown by the FNUSR function. This is not a fault. The analogue routine will merely take a noticeable time to return a result.

If the computer "hangs", and the ESCAPE key has no effect, check the wiring to the potentiometers. The software is waiting for an analogue value, and nothing is connected to the channel in question.

If you find that your interface is indeed faulty, please return it to your Fischertechnik service centre.

TECHNICAL DATA

Fischertechnik Computing Interface Acorn BBC, Fischertechnik Part number 30564.

4 outputs for use with motors, lamps, electromagnets... (M1 to M4)

Outputs are direction switchable.

Current requirements: 1 amp in normal use, 1.5 amps peak load

8 inputs for digital signals (E1 to E8)

Internal switching allows both the positive logic connection of electromagnetic devices such as push-buttons, switches and relays, and also the connection of TTL level outputs. Protection against overload is incorporated.

2 analogue inputs (EX and EY)

Variable resistances may be connected to the analogue channels. Resistances can vary between 0 and 5 kOhm, for example potentiometers, light dependent resistors...

"Watchdog" switching off of the data stream. When no data is received from the BBC computer for more than 0.5 seconds, all outputs are switched off. The nominal output is stored until a new input is received.

Automatic protection against software errors. In the event of a "fatal" syntax error, the "watchdog" switching comes into effect. This happens if the power supply to the interface is too high or too low as well.

Software allowing the interface to be controlled, as well as demonstration programs, is included in the interface package. The basic driver routine checks for syntax errors, specifically for the number and type of parameter.

USE OF FISCHERTECHNIK ELECTROMECHANICS AND ELECTRONICS

The Fischertechnik Computing interface is compatible with the individual parts and electronic modules of the kits named above. As well as the mini-switches provided with the Computing Kit, you may also use other types of buttons or switches. For example, you may use the large switch or the reversing switch from these kits, or even the reed switch or the switched output from a relay. Care should be taken if home made switches are used, made of hinge blocks and springs, as problems may be experienced with "switch bounce". In such cases we would recommend that the state of the switch should be checked several times in succession from the software, and the result assumed to be valid only when the same result is obtained twice in succession.

The analogue inputs may be connected to any sensor which provides a variable resistance between 0 and 5 kOhm as an output. The potentiometers found in the Fischertechnik Computing Kit are naturally suitable, but you may also use other Fischertechnik parts such as the light dependent resistor.

The motor outputs on the interface will take a heavy load. Instead of the mini-motors provided, you may also use the Fischertechnik S-motor or the big N-motor with a signal lamp connected in parallel too. As well as motors, Fischertechnik electromagnets and RBII relays may be used.

Those outputs from the electronics modules which are compatible with the TTL family of integrated circuits (for example the variable-level switch) can be fed into the inputs on the interface. The 0V rail on the electronics modules should always be connected to the common ground socket on the interface to provide a common reference value. The illustration shows how to construct a light sensitive switch. The variable-level switch serves to adjust the sensitivity of the device.

Translation of Illustration Captions:
"IC-Spannungsversorgung" = "Power Supply For Integrated Circuits"
"Schwellwertschalter" = "Variable-level Switch"
"+ Netzgeraet" = "+5V Power Supply"
"- Netzgeraet" = "0V Power Supply"
"Massebuchse Interface" = "Common Ground on the Interface"
"Eingang Interface" = "Input to the Interface"

WIRING DIAGRAM FOR INTERFACE INPUTS AND OUTPUTS.

Art.-Nr. 32 421 · 12.85 ZE