HANDBOOK

PROFI
COMPUTING
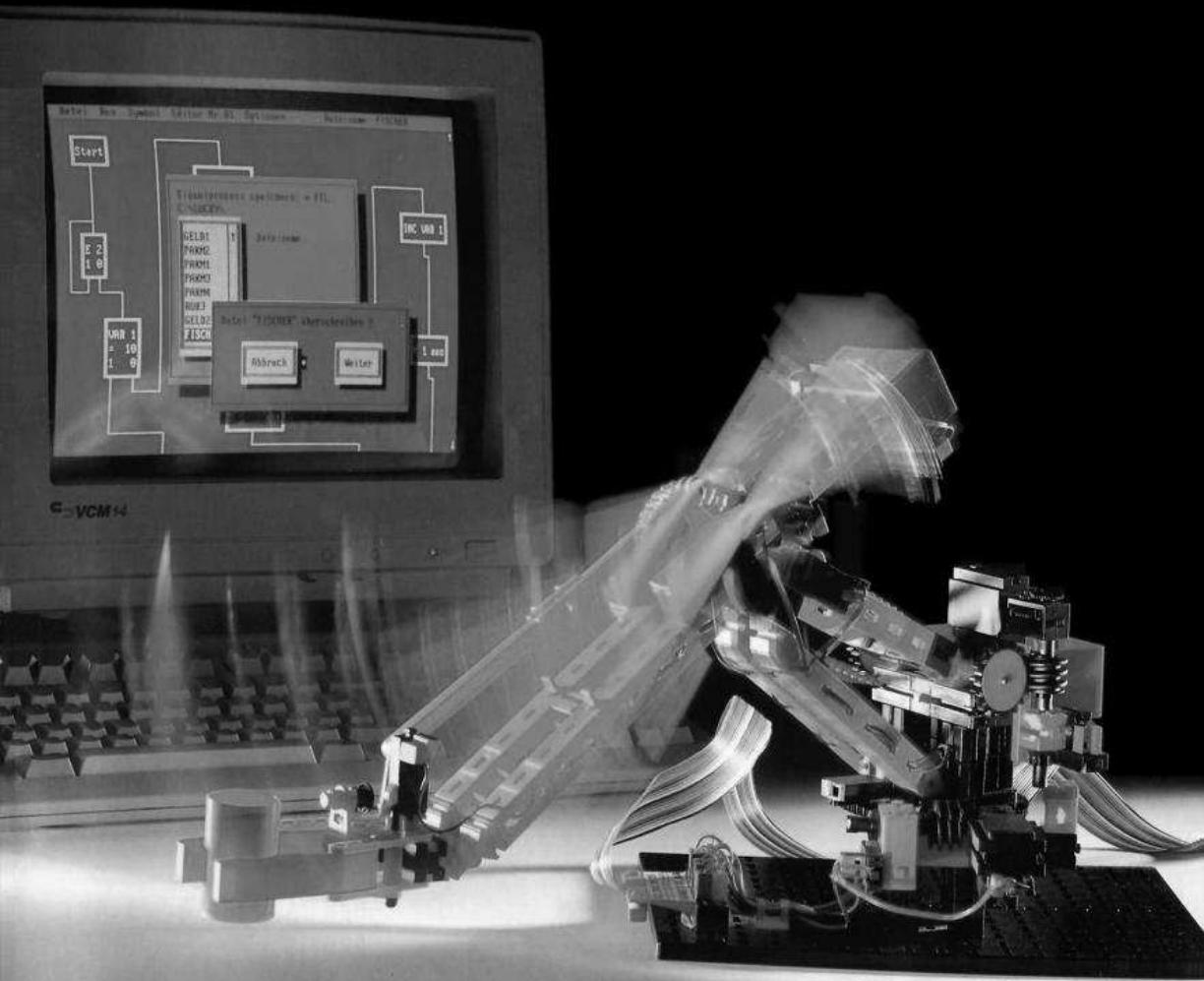
fischertechnik®

PROFI COMPUTING

English version

Contents

All references to diagrams not contained in this experiment manual refer to the German version of the manual.

All software instructions in this experiment manual refer to Lucky Logic for MS DOS.

There is a detailed description of Lucky Logic for Windows in the "Lucky Logic Windows Software Manual".

Lucky Logic Windows contains example programs for all fischertechnik models in the PROFI COMPUTING kit.

Profi Computing

Dear fischertechnik enthusiast,

When the microprocessor, the "heart" of every computer, was first developed, nobody even dreamed about the sort of programs which are so commonplace today - programs such as those for word processing, bookkeeping, graphics, computer games, programming languages etc.; the idea was simply to create a freely programmable circuit for controlling devices and machines.

You've probably seen and heard the reports in the newspapers and on the radio, and the programmes on television about computers, robots and the automation of entire factories. At home too, you will find small computers all over the place for controlling various functions - in the lift, in the video recorder or CD player, in the washing machine, in the microwave, in the computer printer and in the car.

Control systems were already in existence, of course, before the microcomputer was even invented. These early control systems were "hardwired" however, and had to be rewired carefully every time a change was made to the operating cycle of the machine - an enormously expensive procedure in terms of both time and money. The microprocessor made such changes simple - all you have to do is alter the control programs, and there is no need for any soldering or conversion work. The road to computer-controlled machines was opened up. Small computers are used very often, for example, to control lathes or drilling and milling machines. They are programmed either directly at a keyboard or by transferring control data which has been generated at another computer. These "numerically controlled machine tools" progressed into robots with even more capabilities. It is nowadays possible, for example, to build different models and versions of the same car at one and the same production line - with either two or four doors, or with a sunroof or a turbo engine; the computer controls the manufacturing process for all models precisely.

The heavy work is performed by robots. The word "robot" isn't very old. It was invented by the Czech author, Karel Capek, around 70 years ago: he wrote about a scientist who invented artificial human beings - puppets which were seemingly capable of moving around by themselves. He called these machine-people "robots", which comes from the Czech word "robota", meaning something like "slave work". The same idea has been used over and over again by other authors in hundreds of science fiction novels and films.

Real robots are completely different from the inventions of our imagination. They aren't awesome, super-intelligent beings made of metal, but machines which are controlled by computers and which perform highly specific jobs, but do not have any intelligence of their own. However, unlike "normal" machines, robots are extremely flexible and can be used to perform all sorts of different tasks. The same robot can paint a car one week and weld together body parts the next, simply by changing the computer program (the paint sprayer must of course be replaced by a welding machine). Robots are generally used to work in situations which would be too difficult, too dangerous or too tiring for human beings. They are also employed in factories because they can work more rationally - they don't go home in the evenings or take holidays and they need neither sleep nor food. And in nuclear power stations and on distant planets they do work which no human would be able to manage at all. The intelligence and creativity of this combination of a computer and a robot however comes entirely from the program - and the program is written by man.

Controlling machines and robots is also the subject of the fischertechnik Profi Computing kit. The various models take us on an excursion into industrial control by computer. Motors are turned into the "arms" and "hands" of the computer, while photo-cells and switches act as its sense organs - the computer and the model thus work hand in hand. This experiment kit demonstrates the possibilities of computer control systems in miniature. And the instruction manual and the special fischertechnik control software ensure that you soon become familiar with programming. fischertechnik Profi Computing can of course also be combined with any other fischertechnik components (e.g. Profi Sensoric), and you can give your

imagination free rein. When you have got properly used to the models and the software, you can attempt to put some of your own ideas

Before you start...

... there are a few important things you should do before you start experimenting with your kit, as well as some tips and instructions we would like to give you.

What else you need

Since the models are going to be controlled by computer, there are a few things you will need in addition to the construction kit:

Obviously a computer (IBM PC XT/AT or compatible, e.g. Atari-ST, Amiga 500/2000, 3000), with mouse,

An interface (=an electronic circuit which links the computer up to the fischertechnik components),

The fischertechnik LUCKY LOGIC control software (which acts as the "fuel" for the whole thing).

The software and the interface are not included in your kit. When you purchase the software, you will find a detailed operating manual enclosed. It also contains a few exercises which introduce you to controlling the fischertechnik interface. A lot of the information in the manual is repeated again here, except that in this manual the descriptions are much shorter and refer only to specific models.

Assembling plugs

All the cables are fitted with plugs. Fig. 1 (see "Bild 1" of the German manual) shows you how to assemble the plugs.

Preparing cables

Cables of different lengths are used to connect motors, lamps, sensors, the power supply and

Into practice or to combine two different models together.

the interface. You must cut up the 2-core cable enclosed with the kit to the following lengths:

> 4 sections, each 20 cm long
> 3 sections, each 25 cm long
> 3 sections, each 30 cm long
> 3 sections, each 35 cm long
> 3 sections, each 40 cm long
> 2 sections, each 45 cm long

The remainder of the cable (approx. 90 cm) is used as the power supply cable between the power supply unit and the interface.

Stripping cables and fitting plugs

Cut open the ends of the 2-core cables to a length of about 3cm (Fig. 2). To strip the cables, score carefully around the plastic insulation with a knife, about 4mm from the end. Take care not to damage the fine copper cores of the cable. Then pull the plastic insulation away from the copper conductor. Bend back the copper wires and then screw on the plug. Undo the screw in the plug to do so, and insert the end of the cable. Then tighten the screw gently, taking care not to squeeze the insulation too hard (Fig. 3).

The polarity of the components is important in some circuits, in other words it sometimes makes a difference which way round you connect a component. The kit contains red plugs and green plugs, so that you don't get them mixed up later on when you connect them up; the colors have the following meanings:

Red plug = (+)
Green plug = (–)

Plugs with the same color must thus be connected to both the ends of each cable core-- red plugs to red cores and green plugs to red/green cores (Fig. 4).

Longer cable sections than the precut lengths are required for some models. The combinations you need are shown in the wiring diagrams of the models. In order to connect two cable sections together, you must replace the plugs at one end of the cable with plug sockets. You must fit a plug into the socket before you attempt to secure the cable in it, to stop the end of the cable from being pulled too far inside (Fig. 5).

Paper-wire strip and cable clips

The paper-wire strip can be used to combine several different cables to form a so-called "cable harness" (Fig. 6). You should use it whenever you need to tidy up your wiring, or if there are any cables hanging down which might interfere with the model, such as the long cables of the plotter.

You can use the cable clips to secure one or more cables to a building block or to the experiment baseplate (to stop them from getting stuck in a gear wheel, for example) (Fig. 7).

Power supply unit (230V~ / 8V =)

Power (230V~ / 8V=) is supplied to the models by the fischertechnik power supply unit (Art. No. 30180). This has two pairs of jacks (in other words four altogether) marked (+) and (–). The two (+) jacks and the two (–) jacks are connected together inside the power supply; it therefore does not matter which of each pair you use. It is however important to connect the power supply unit correctly, since the model will not work if it is connected the wrong way round, and you may even damage components (Fig. 8).

Important:
The power supply unit must be checked at regular intervals for any signs of potential danger (such as damage to the cable, plug, casing or any other parts). If any such damage is found, the unit must not be used again until it has been properly repaired!

Interface

The interface links the fischertechnik models to the computer since the computer ports cannot provide enough power on their own to supply a motor, for example. In addition, the interface protects the ports of the computer if you make a mistake when wiring up a model. It also incorporates an automatic timer, which cuts off the power to the motors if the computer stops sending control commands. You will often wish to make changes when testing the control software. The timer stops the model automatically, so that you don't have to interrupt the power supply to the interface every time you want to modify the program. The model carries on again as normal as soon as the computer sends more commands. There is one thing you should always remember though:

IMPORTANT:
The computer and the interface should always be switched off before the interface is connected or disconnected.

The interface also has two ports, EX and EY, for measuring resistances, to which you can connect the photo-sensor or a potentiometer (a variable resistor), for example.
How and where to make the interface connections is described in the interface manual, specifically for your computer.

Control software

Special control software - LUCKY LOGIC - is available for controlling the models; it is extremely simple to use. You can position the drawings of the switches, lamps and motors individually on the screen and draw connectors with the mouse. When you run the control program, the model is controlled by the computer via the interface. A detailed description of the control commands is enclosed with the software disk. The fischertechnik control software can also be used to test the interface and the connecting cable. To do so, you must connect

motors and lamps to the outputs (M1 to M4) and switches to the inputs (E1 to E8). After connecting interface plus power supply to the computer select "Interface diagnosis" in the software menu. The positions of the various switches are displayed on the screen permanently. You can test a motor by "clicking" it with the mouse. Select the direction of rotation with the two mouse buttons (right = clockwise, left = counterclockwise). If you call up this diagnostic function before operating a model with the control program, you will able to see immediately whether or not the motors are connected the right way round.

Power supply

The model and the interface are supplied with power by the fischertechnik power supply (230V~ / 8V=) unit mentioned earlier. The connecting cable has one red plug and one green plug. The interface has two pairs of jacks marked (+) and (–). The two (+) jacks and the two (–) jacks are connected in parallel inside the interface. It therefore does not matter which of each pair you use. It is however important to connect the interface to the power supply correctly, since it will not work if it is connected the wrong way round:

> Red   plug connected to a (+) jack
> Green plug connected to a (–) jack

Connecting the computer to the interface

An adapter is provided for connecting the interface to the printer port of the computer. It takes the form of a small printed circuit board with two plug-in connectors. One of these connectors matches the computer, while the other consists of two rows of 10 pins each in a small casing. The plug on the end of the gray connecting cable must be connected here. On top of the plug is a small, central lug, which fits exactly into the recess in the plug-in connector on the printed circuit board. This is designed to prevent the plug from being inserted the wrong way round.

Connecting the interface to the model

The interface must now be connected to the model. The construction kit includes a 20-pin, multicolored ribbon cable. At one end of this cable is a 20-pin plug, which must be connected to the interface. This plug likewise has a lug which fits into the recess in the casing of the interface, so that once again you cannot insert it the wrong way round. The cable colors are printed on the casing as well, just to make sure. First, however, you must prepare the other end of the cable for connecting the models. The procedure is very similar to that described earlier for assembling the plugs.
The construction kit includes a 28-pin socket connector, to which you can connect the fischertechnik plugs.

Start by assembling the socket connector with all 28 jacks as shown in Fig. 9. Fig. 9 also shows how to use a plug as an assembly tool. Don't forget to stick the polarization strip onto the top of the socket connector before you screw on the cable.
Screw the 20 cores of the ribbon cable and the comb to the socket connector. Fig. 10 shows you exactly which cores must be connected to which jacks:

1. Separate the end of the cable into individual cores to a length of about 3 to 4 cm.

2. Strip the cores carefully to a length of approximately 4 mm.

3. Bend the bare cable ends back over the insulation.

4. Insert the cores in the appropriate jacks in the socket connector and tighten the screw gently.

5. Secure a metal comb (contact bridge) to one end, to connect several jacks together.

Comb/contact bridge

Brown 1	Brown 1
Red 1	Red 1
Orange 1	Orange 1
Yellow 1	Yellow 1
Green 1	Blue 1
Blue 1	Green 1
Violet 1	Violet 1
Gray 1	Gray 1
White 1	White 1
Black 1	Black 1
Brown 2	Brown 2
Red 2	Orange 2
Orange 2	Green 2
Yellow 2	Violet 2
Green 2	White 2

Red 2
Yellow 2
Blue 2
Gray 2
Black 2
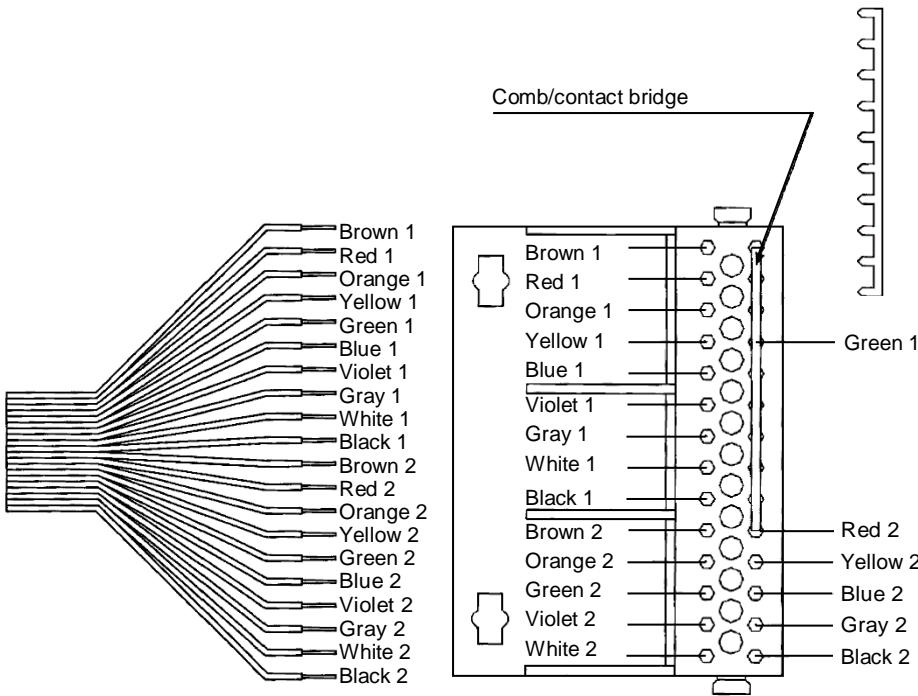
You should take your time about connecting the many cable cores, and be sure to connect them exactly right. (Fitting the cores in a rush and then having to search out your mistakes afterwards is likely to take a lot longer.)

When you have finished making all the connections, you should carry out a visual inspection before continuing. Check that all the cables are connected exactly as they should be (even experienced professionals can get distracted sometimes and make a mistake as a result). You could save yourself a lot of problems later on and avoid possibly damaging the interface. Troubleshooting is much more difficult once the model is connected. Don't connect the cable to the interface until you are sure that the wiring really is correct. Then fit the strain relief for the cable to the socket connector as shown in Fig. 11.

The basics of electrical engineering

Current flowing in a circuit

Not all substances conduct electricity equally well. A current flows especially well in metals, for example in the thin copper wires used in fischertechnik kits. Brass, iron, lead, tin and fischertechnik metal axles are also good conductors. If two conductors come into contact with one another, the current can also flow across the point of contact (we benefit from this characteristic, for example, when we connect plugs and sockets). Other substances are either poor conductors of electricity or do not conduct it at all. This is the reason for using plastic to protect the copper cores of the cables against accidental contact, plastic being a so-called "insulator," i.e. a definite non-conductor. Air, glass, dry wood and the majority of non-metal substances are also insulators.

A power source, e.g. a battery or a separate power supply unit, is needed to operate electrical loads (lamps, electromagnets, motors). Imagine the power source to be like a water pump, which pushes the electricity through the cables and the loads. Like the pump in an aquarium, it needs a closed circuit for the current to be able to flow. If the circuit is interrupted at any point, the electricity is no longer able to flow. In the same way as the pump can generate a particular water pressure, depending on its capacity, different power sources can supply different voltages (measured in volts). The power supply unit, for example, supplies 8 volts, while a normal power outlet supplies 220 volts, or in special cases 380V. Above a particular level (around 50-60 V), the voltage starts to become dangerous as far as people are concerned. The components in the fischertechnik kits operate with a safe voltage of 6-9V; you must never use them with a higher voltage, or you may damage them beyond repair. Now you can see the point of the power supply unit: it turns the 220V which comes out of the power outlet into the safe voltage you need to operate your fischertechnik models.

Every load requires a particular amount of electricity - once again like the water flowing through a pipe. And just as a water tap puts up a resistance against the water flow, the load acts as a resistance against the electric current. The lower the resistance of the load, the higher will be the current and vice versa. The intensity of the current is specified in electrical engineering in "amperes" (abbreviated to "A"), e.g. if you look on the circuit-breaker in your house, you will normally find either "10A" or "16A" indicated.

If one of your circuits doesn't work for some reason, you should start by checking all the contact points, e.g. for loose plugs or loose cable connections in the plugs. Just by pulling a cable too hard too often, you can cause its copper cores to break or tear. The consequence is a "loose contact" - a little mischief-maker which only gives you trouble some of the time, but is all the harder to locate as a result.

Electrical engineering "all mapped out"

In the same way as you need a map to find your way around in a strange town, you will not get far in electrical engineering without another kind of map - the "circuit diagram". And just as a city map is not an aerial photograph, including such things as houses and trees (which would only make it confusing), a circuit diagram does not show realistic drawings of cables, lamps, switches and motors. Instead, each component is represented by a simple symbol, and the connecting cables between the components are shown as lines. If there is an electrical connection between two cables, a large dot will mark the connection point. If two lines intersect without a point, there is no electrical connection. However, there is no need for us to simplify our circuit diagrams quite as much as an engineer would. When you study a fischertechnik circuit, you will still be able to see at a glance which component is which.

Electrical components

1. Switches

The switches have three connections (terminals): the middle terminal (1) is movable and is connected to the bottom terminal (2) when it is not actuated; the latter is therefore known as a normally closed contact or break contact. If the switch is pressed, the contact is reversed and a connection is formed between the middle terminal (1) and the top terminal (3), the normally open contact or make contact. If the switch is actuated by means of a pulse wheel, it closes five times per revolution (Fig. 12).

A switch is thus always connected to an E-port of the interface.
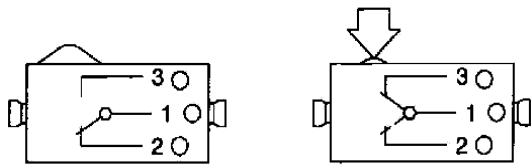
Switch in normal
(off) position

Switch in working
(on) position
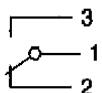
Contact between
1 and 2
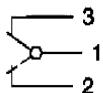
Contact between
1 and 3

Fig.12
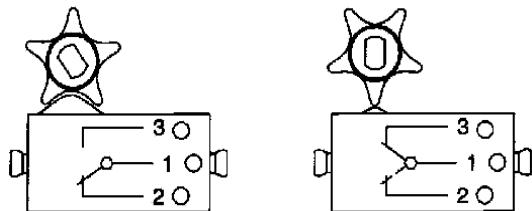
Switch with pulse wheel

2. Lamps

The light holder comprises a plastic base with two little tubes, between which you can fit the lamp (Fig. 13).

Fig 13

Ball lamp

Lens bulb   Photo-transistor

You can thus connect the power supply either to one end or to both ends (Fig. 14). If you connect both the power supply plugs to the

same tube however, you will cause a short-circuit! Since the tubes are like socket connectors, you can save on cables if you connect two lamps together as shown at the bottom of Fig. 14.

Make sure you use the right type of lamp for each model. Never fit a colored light cap over a lens bulb, for example, since the heat which is generated can cause the cap to become deformed after only a short time (burning glass effect). The lens bulb is used mainly in conjunction with the light sensor for building light barriers.

Never remove a lamp from the light holder by pulling the glass tube, but instead always lever it out with a screwdriver.

A lamp is connected either directly to the power supply or to an M-port of the interface.

3. Motor

Electric DC motor which can be connected in three different ways (Fig. 15). You can also combine the different connection options (e.g. by inserting the plug from the front on the right and from above on the left). The two plugs must however NEVER be connected to the same side of the motor, or you will cause a short-circuit.

There is an additional rule to remember about the motors. With a lamp, it is irrelevant which way round you connect it; the lamp lights up whenever the circuit is closed. With a motor, on the other hand, the direction in which it rotates depends on the polarity of the terminals, in other words where (+) and (–) are connected.

You can perform a simple test with the power supply unit and a motor. Connect the two together. The motor rotates in one direction. If

13

you then swap the motor connections around, the motor will turn in the opposite direction. If a motor in one of your models turns the wrong way, all you need to do is to reverse the terminals.

A motor is always connected to an M-port of the interface.

4. Light sensor (photo-transistor)

The photo-transistor conducts electricity better the more light it has.

> The photo-transistor is the only component which must always be connected a particular way round.

It is enclosed in the kit fully assembled. The (+) end is marked in red. When you connect it to the 28-pin socket take great care to insert the (+) end of the photo-transistor in one of the jacks (Fig. 16) in the row connected together by the long metal comb. NEVER connect it to the M-ports or directly to the power supply unit! Maximum voltage 12 V. The light sensor must

always be connected to an E-input of the interface. It is illuminated by a lamp. If the beam of light between the lamp and the photo-transistor is interrupted, the effect on the computer is exactly the same as if a switch with a normally closed contact is opened - in other words like an interrupt. The arrangement comprising a photo-transistor and a lamp which is shown at the bottom of Fig. 16 is referred to by the professional as a "light barrier".

> Rule of thumb:
> If light hits the photo-transistor the input of the interface is switched to "1". If the beam of light is interrupted, the input changes to "0".

> All the electrical components must always be operated using a fischer-technik power supply (power supply unit or batteries), since they may otherwise be destroyed if the operating voltage is too high.

red mark
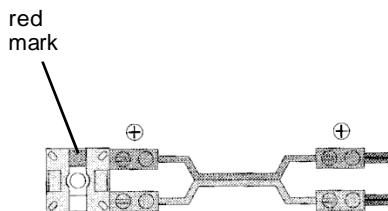
Photo-transistor connection

⊕ Connection

Light barrier connection
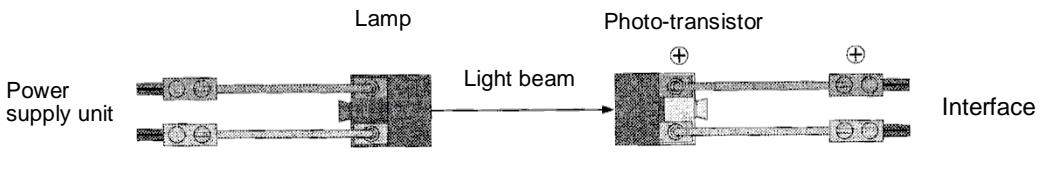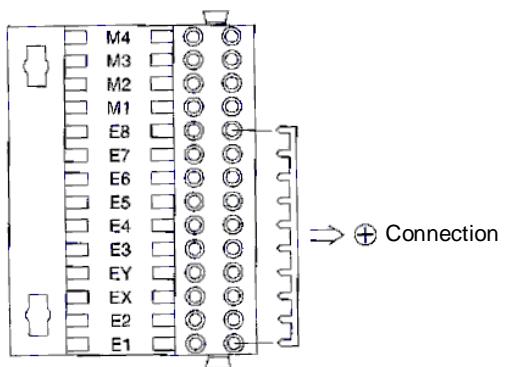
Fig.16

Assembling the mechanical components

Two mechanical components, the spring hinged block and the hinged block, must first be assembled before you can use them. We then want to give a few tips about assembling the components you will need for the various models.

1. Spring hinged block

The spring hinged block consists of five different parts. The assembly sequence is shown in Fig. 17. When fitting the spring, make sure that the bent ends are properly inserted in the small holes in the two halves of the hinged block. The additional building block with the link is merely used as an assembly tool. Finally, screw together with two coins - not too tight however, as the hinge must still be able to move freely.

2. Hinged block

Assemble this hinged block simply by fitting the two parts together (Fig. 18).

3. Flat hub

Screw the hub chuck and the hub nut to the gear wheel. Then fit onto the axle and tighten the hub nut firmly (Fig. 19).

4. Locking worm

Screw the chuck nut a few turns onto the worm, then fit onto the axle and tighten the chuck nut firmly. The locking worm has a few additional sections on some models (Fig. 20).

5. Chain and crawler track link

Assemble the chain by fitting the individual links together until it is long enough (the instructions for the models always tell you how long the chain should be). You can build a crawler (or a conveyor) by adding the crawler track links. These are held in place by the square lugs on the side (Fig. 21).

6. Robot gripper

To stop small or smooth parts from falling out of the gripper, the two self-adhesive disks must be stuck onto its surfaces (Fig. 22).

7. Angular blocks

The construction kit contains four different angular blocks, which are not always easy to distinguish in the assembly drawings for the models. Fig. 23 clearly shows the differences in their angles. The size of the angle is printed on the end of each building block.

8. Reducing gear 65:1 and wheel axle

Without the gear, the models would run much too fast and the motor wouldn't be able to produce enough power. Fig. 24 shows how you can combine the wheel axle with the other latching parts.

9. Try out the mechanics manually

It is often a good idea to try out the models first manually after assembling them, to make sure that the mechanical components are working correctly (e.g. by opening and closing the robot's gripper). To do so, simply pull the gear wheel and the wheel axle about 4mm out of the gear (Fig. 25).

Building the models

The models themselves must be constructed step by step, as shown in the assembly diagrams. Each time you proceed to the next step, the new components are illustrated in color. If they must be built onto a part which is already assembled, this part is shown in white. Before you begin a new step, sort out all the components you need and then build them onto your model. The next step never starts until you have used up all the parts. Make sure that you fit the building blocks the right way round, so that you don't get stuck later on.

It's also very important to tighten all the axle screw connections (e.g. hubs) properly.

Electrical components are used as well in all models, namely switches, motors and light sensors (photo-transistors). The connection to the computer interface is always formed with the 28-pin socket connector. The assembly manual tells you how the components are connected to the interface and how long the cables should be.

When you control one of your models by hand, you can see what the model is doing and react accordingly; the computer on the other hand is "blind". Since it - or, to be more precise, the control program - requires information on the model's behavior (e.g. on the position of a sliding bar), switches or photo-sensors are used in many of the models. The "feedback" they provide enables the program to determine the state of the model and respond as necessary.

A few tips regarding the software

The models can be controlled by computer in two different ways, namely with the fischertechnik LUCKY LOGIC control software or directly with a programming language such as Pascal or Basic. Both alternatives are described in this manual. Programming with LUCKY LOGIC is explained in detail in the software manual, and so the tips you will find here are aimed more at those of you who want to write your own programs.

Only the Turbo-Pascal programs for IBM are reproduced in this manual. The programs for

the Atari and the Amiga have been written in C and have exactly the same structure as the Pascal programs. The Turbo-Pascal driver (FISCHER.TPU file) has been compiled using Version 6.0; if your version of Turbo-Pascal is an earlier one (e.g. 5.0 or 5.5), all you need to do is compile the source program FISCHER.PAS again.

The very first time you start up the program, it requires two pieces of information about your computer:

1. The speed at which it works.

2. The printer port (= printer interface, e.g. LPT1/LPT2).

These two items of information are saved in the FISCHER.DAT file and can be retrieved by the programs over and over again. The same file is also used by LUCKY LOGIC, by the way. If you connect your interface to a different printer port or if you change your computer, simply delete the FISCHER.DAT file. The software then requests the new values.

For Program see disk/ Annex

When querying the switches, it is often not enough simply to ask whether or not a switch is pressed - the program must wait until the switch returns to its normal position again (e.g. if the switch is actuated by a pulse wheel). You must wait, in other words, until the switch is pressed:

REPEAT UNTIL Taster(1);
              [Example: switch at E1]

The program must then wait until the pulse wheel releases the switch again:

REPEAT UNTIL NOT Taster(1):

Don't be afraid to experiment with your own programs - as soon as the program stops, the interface stops the motors as well. You can of

course also incorporate an "EMERGENCY STOP" function in the program by inserting the following line at the appropriate points (especially in loops):

```
If KEYPRESSED then BEGIN
Motoren_aus; Halt; END;
```

As soon as you press a key on the computer, the model will come to a standstill and the program is terminated.

> The important thing is for the interface to be connected to the computer - otherwise the program may "get stuck" in a loop, since there is no feedback from the interface.

The LUCKY LOGIC software package (Art. No. 30521) includes the following components:

IBM - LUCKY LOGIC program and sample files (*.FTL and *.FTG)

- Turbo-Pascal drivers, sample pro-grams (*.PAS), exe-cutable Pascal programs (*.EXE)

- Gw-Basic drivers and sample program

Amiga - LUCKY LOGIC program and sample files (DFO: and DFO:\LL)

- C drivers, sample programs (DFO:\ models\ source\-.C), executable C programs (DFO:\models)

Atari - LUCKY LOGIC program and sample files (A**:** and A:\LUCKYDAT*.FTG *.FTL)

- Turbo-C drivers, sample programs (A:\program*.C), executable C programs (A:\program*.TOS)

Instructions for Atari and Amiga users

Amiga

The fischer disk is already configured for the Amiga. Inset the disk in the drive and open it by clicking the mouse twice. In addition to the LUCKY LOGIC program (see LUCKY LOGIC instruction manual), you will also find two drivers (subdirectories). The executable programs are stored in the "MODELS" driver and the C source programs in the "SOURCE" driver. To start the programs for the models, just open the "MODELS" driver and click the program you want.

Atari

The disk for the Atari is also pre-configured. The LUCKY LOGIC program (LUCKYLOG.PRG) is contained in the "root" directory (main directory) on the disk. The "README" file contains additional information on the program. You can either read this file on the screen or print it out. The LUCKY LOGIC control programs (*.FTU*.FTG) are contained in the "LUCKYDAT" directory. The "PROGRAM" subdirectory contains both the executable programs (*.TOS) and the C source programs (*.C/Turbo C). Once again, you can start your models simply by clicking the TOS programs. That's all the preparation you need. You are now ready to read through this instruction manual at your leisure, and to stop to browse every now and then - and of course to choose your first model.

Reflex tester

This model will allow you to test your own reflexes and those of your friends. When one of the two lamps lights up, you must press the switch in front of it as fast as possible (red or yellow). The time between the lamp lighting up and you pressing the switch is measured.
Ball lamps must be used in the lightholders.

The various steps are shown on pages 30 of the German manual.

Wiring

The two lamps must be connected to outputs M1 and M2, and the switches to inputs E1 and E2. Both switches use the normally open contact. The switch connected to E1 belongs to the lamp at M1 and the switch connected to E2 to the lamp at M2. It is important to use the right cable lengths.

Now start the LUCKY LOGIC program and test the wiring by selecting OPTIONS INTERFACE DIAGNOSIS in the menu. When you press the switches, the input indications shown below should change accordingly. Then switch the M-outputs (lamps) by clicking them with the mouse as described in the software manual. If everything is working properly, you can load and start the software.

Test table:

E1	Switch for lamp 1
E2	Switch for lamp 2
M1	Lamp 1
M2	Lamp 2

Software

The software for this model has been written in Turbo-Pascal. The Pascal program is contained in the REAKTION.PAS file. Start the REAKTION program. The computer waits for a random length of time (between 0 and 10 seconds) before switching on one of the two lamps. You must now press the switch which belongs to this lamp as fast as possible. The start and stop times are displayed on the screen.

For Program see disk/ Annex

Tune memory game

This model tests your powers of memory. The computer plays a tune consisting of four different notes. There is a separate switch for each note (black, blue, yellow and red) with a lamp in the center. When the computer plays the tune, the lamps light up in the switches as appropriate. Then it's your turn. You must now play back the tune by pressing the switches in the same order as they were actuated originally by the computer.
Ball lamps must be used in the light holders.

The various steps are shown on pages 36 and 37 of the German manual.

Wiring

(See page 39 of the German manual)

The lamps must be connected to outputs M1 to M4, and the switches to inputs E1 to E4. All the switches use the normally open contact. The switch connected to E1 belongs to the lamp at M1 and the switch connected to E2 to the lamp at M2, etc. It is important to use the right cable lengths.

Now start the LUCKY LOGIC program and test the wiring by selecting OPTIONS INTERFACE DIAGNOSIS in the menu. When you press the switches, the input indications shown in the test table should change accordingly. Then switch the M-outputs (lamps) by clicking them with the mouse as described in the software manual. If everything is working properly, you can load and start the software.

Test table:

E1	Switch for lamp 1
E2	Switch for lamp 2
E3	Switch for lamp 3
E4	Switch for lamp 4
M1	Lamp 1
M2	Lamp 2
M3	Lamp 3
M4	Lamp 4

Software

The software for this model has been written in Turbo-Pascal. The Pascal program is contained in the MELODIE.PAS file. Start the MELODIE program. First of all, a random tune composed using four different notes is saved. The tune has a total of 30 notes. The program then plays one note, and the lamp contained in the switch to which this note belongs lights up. You must then press the switch to indicate that you have made a mental note of it. The program then plays two notes in the same way and asks you to play them back; it continues with 3, 4 and 5 notes, etc. This game is quite easy when you only have to remember a few notes, but we bet hardly any of your friends will be able to manage all 30.

For Program see disk/ Annex

Turtle

The turtle is a small vehicle with two driven front wheels, which can travel along routes specified by the program. Each of the drive wheels has a separate motor, in other words it can turn forwards or backwards independently of the other. This trick means that the turtle can not only travel forwards and backwards - it is also capable of going round any bends and even of turning on the spot, depending on how the motors are controlled.

The table below shows all the different possible turtle movements better than any description:

Turtle movement	Motor M1 CW	Motor M1 CW
Standstill	Off	Off
Forward	Forward	Forward
Backwards	Backwards	Backwards
Forward RH curve	Off	Forward
Forward LH curve	Forward	Off
Backward RH curve	Backwards	Off
Backward LH curve	Off	Backwards
CW rotation	Backwards	Forward
CCW rotation	Forward	Backwards

Switches operated by means of pulse wheels are attached to the two motors, to allow the computer to establish how far the turtle has already travelled. The switches close five times per revolution of the drive shaft. The switch connected to E1 belongs to the motor at M1, while the switch connected to E2 belongs to the motor at M2.

A pickup fork is mounted at the front of the turtle; it can be moved up and down (like on a fork-lift truck) via motor M3. It enables the turtle to pick up a pallet and put it down again elsewhere. Once again, switches are necessary to tell the computer program when the fork has

reached its upper or lower limit position. They are set correctly if you install them exactly as shown in the diagrams. The switch connected to E3 defines the lower limit position, while that connected to E4 defines the upper limit position. The turtle travels best on a smooth, flat surface. Carpets are not suitable. The connecting cable to the interface should hang down from above, to prevent the turtle from getting caught up in it. The various steps are shown on pages 45 - 47 of the German manual.

Wiring

The switches and motors must be wired up as shown in the diagram (see page 48 of the German manual). The switch connected to E1 belongs to the motor at M1, while the switch connected to E2 belongs to the motor at M2. M3 moves the fork. The switch connected to E3 defines the upper limit position, while that connected to E4 defines the lower limit position. It is important to use the right cable lengths. Now start the LUCKY LOGIC program and test the wiring by selecting OPTIONS INTERFACE DIAGNOSIS in the menu. When you press the switches, the input indications shown in the test table should change accordingly. Then switch the M-outputs (motors) by clicking them with the mouse as described in the software manual. If everything is working properly, you can load and start the software.

Test table:

E1	Step counter for right-hand drive motor; signal only changes if motor running
E2	Step counter for left-hand drive motor; signal only changes if motor running
E3	Fork upper limit position
E4	Fork lower limit position
M1	Right-hand drive motor
M2	Left-hand drive motor
M3	Drive motor for fork

Software

The turtle program has been written with LUCKY LOGIC. It consists of four files, namely TURTLE.FTG, TURTLE1.FTL, TURTLE2.FTL and TURTLE3.FTL. Load the combined TURTLE process and start the program by selecting RUN COMBINED PROCESS in the menu.

For Program see disk/ Annex

The turtle moves straight ahead a short distance, then picks up the "pallet", moves on a bit further and puts the pallet back down again. When you program your own routes, you must remember three things:

Processes 1 and 2 for the two drive motors must be coordinated with one another, since their direction of rotation determines the path travelled by the turtle (bends, etc.).

Process 3 for the pickup fork is controlled by means of VAR 3. The fork must be "ready" before the turtle starts moving again. The computer establishes this by querying E3 or E4 (depending whether the fork is supposed to be up or down).

If the runtimes of processes 1 and 2 are different (e.g. when the turtle travels round a bend), the longer of the two processes should always be the one which issues the order to the fork. This is clearly demonstrated in the example.

Safe with combination lock

You will probably have already seen doors that can't be opened with a key, but where you must enter a number at a keyboard instead. This model of a safe with a combination lock works in a similar way. The lid opens automatically if you enter the right code by means of the four switches connected to E1 to E4. You can choose the code yourself and enter it in the program (the code used in the example is 1-3-2).

The various steps are shown on pages 54 - 55 of the German manual.

You can try out the mechanical parts and set the limit switch correctly by pulling the gear wheel and the wheel axle roughly 4mm out of the reducing gear. If you turn the gear wheel counterclockwise, the lid will open, while if you turn it clockwise, the lid will close.
Two limit switches are used to tell the computer when the lid has reached its limit position (either fully open or closed tight). The limit switch for when the lid is closed (connected to E6) is mounted inside the safe. The limit switch for when the lid is open (connected to E5) is

mounted on the outside on the hinge. You must adjust the position of this switch so that it is closed when the lid is open.

Wiring

(See page 56 of the German manual)

The wiring is not very complicated. The four code switches must be connected to inputs E1 to E4, the outside limit switch (lid open) to E5 and the inside limit switch (lid closed) to E6. The motor is connected to M1. It is important to use the right cable lengths and to check that the limit switches are working properly. Now start the LUCKY LOGIC program and test the wiring by selecting OPTIONS INTERFACE DIAGNOSIS in the menu. When you press the switches, the input indications shown below should change accordingly.
Then switch the M-output (motor) by clicking it with the mouse as described in the software manual. If everything is working properly, you can load and start the software.

Test table:

E1	Code switch 1
E2	Code switch 2
E3	Code switch 3
E4	Code switch 4
E5	Limit switch for "lid open"
E6	Limit switch for "lid closed"
M1	Drive motor for lid

Software

The combination lock has been programmed with LUCKY LOGIC. Since only one motor must be driven, there is only one "single process" in the SAFE.FTL file. Load the single process and start the program by selecting RUN SINGLE PROCESS in the menu.

Only three switches are used in the programming example. In order to open the safe, the switches must be switched on and then off again, one after the other, in the following order: E1 - E2 - E3. If you want, you can program switch E4 as well by opening up the connector to the "M1CCW" "control statement" and inserting switch E4 there.

For Program see disk/ Annex

Money-changing machine

A lot of ticket machines and vending machines give change if you put too much money in (because you haven't got the correct amount on you). Some machines can even give you change for banknotes. You can build a money-changing machine like this as a fischertechnik model.

The amount which is dispensed from three tubes filled with coins is controlled by the program. The coins must not be thinner than the coin bar (so don't use very small ones). When you insert the tubes, take care not to push them in too far, or you may cause the coin bar to jam and your money-changing machine won't work properly.

If the slide to which the coin bar mechanism is secured does not move freely on the two metal axles, a small drop of household oil or sewing-machine oil on the guide axles will probably help. Make sure that the chuck nut on the locking worm and the collet chuck are properly tightened, or the axles will simply spin round. If you want to try out the mechanical parts by moving the slide by hand, pull the gear wheel and the wheel axle roughly 4mm out of the gear. You can then turn the axle and the worm freely.

All the switches send information to the computer in this model, since it must determine not only the position of the coin bar, but also the path travelled by the slide. The switch at E4 is actuated by a pulse wheel - five times per revolution of the axle and worm.

The various steps are shown on pages 61 - 63 of the German manual.

Wiring

(See page 65 of the German manual)

The motor connected to M1 controls the coin bar, while that connected to M2 controls the horizontal movement of the slide. The switches at E1 and E2 tell the computer when the two limit positions of the coin bar have been reached (E1 = coin bar right back, E2 = coin bar right forward). You may have to adjust the positions of these switches a little after the trial run. The switch at E3 reports the right-hand limit position of the slide, while that at E4 counts the revolutions of the shaft and the worm. It is important to use the right cable lengths; you must make a cable out of two sections (20cm and 30cm) for the connection to the switch at E3. To do so, replace the plugs at one end of one of the two cables with plug sockets. How to do this is described at the beginning of the manual.

Now start the LUCKY LOGIC program and test the wiring by selecting OPTIONS INTERFACE DIAGNOSIS in the menu. When you press the switches, the input indications shown in the test table should change accordingly. Then switch the M-outputs (motors) by clicking them with the mouse as described in the software manual. Be careful with motor M1: you must stop the motor in good time, so that the coin bar does not move too far forward or back. If everything is working properly, you can load and start the software.

Test table:

E1	Coin bar backward
E2	Coin bar forward
E3	Right-hand limit position of slide
E4	Pulse wheel of slide drive
M1	Drive motor for coin bar
M2	Drive motor for slide

Software

The software for the money-changing machine has been written with LUCKY LOGIC. It consists of three files, namely MONEY.FTG, MONEY1.FTL and MONEY2.FTL. Load the combined MONEY process and start the program by selecting RUN COMBINED PROCESS in the menu.

Process 1 controls the coin bar. The coin bar moves right back first of all, and then right forward, and a coin is dispensed. If you want to dispense several coins, you must issue the same order to the coin bar the appropriate number of times (only one coin is output at a time in this example).

The second process positions the slide in front of the desired coin tube. Only the first two tubes are used in the example: you can program the path to the third tube easily yourself by deleting the end symbol and inserting the control instructions for the third tube instead. The new control section must begin again with the VAR2=0 symbol and end with the order to the coin bar.

For Program see disk/ Annex

Coded card reader with automatic insertion

Although all coded cards, such as cheque cards, have the same dimensions, they function in different ways. The simplest types of coded card are "transilluminated" with infrared light; a non-transparent foil with holes in certain places is integrated in the card. The light penetrates the card through these holes and sets off a switching action.

The card code is determined by the positions and number of the holes. This type of card is used, for example, for company ID cards, which establish the number of hours worked at the same time.

Our coded card reader uses visible light instead of infrared light. The advantage of this is that you can make the coded cards yourself. The coded card is the same size as a cheque card (Fig. 26) and is inserted automatically. The code comprising two rows of holes is read and interpreted by the computer. The model uses two light barriers, one for each row of holes, to read the code. How to make the coded card is described on page 23.

If you hold the card with the arrow pointing away from you, the row of holes on the left is responsible for the "clock" and the row of holes on the right for the data. The row of holes for the "clock" is necessary to inform the coded card reader when it can scan a hole in the right-hand row (whether or not there is actually a hole there depends on how you have coded the card). All 5 holes in the left-hand row are in other words punched in every card. The actual code is contained in the right-hand row (there is either a hole or no hole, depending on the program).

The switch for the slot of the card reader (in which the card is inserted) must be set so that it is actuated when a card is introduced. The motor then swallows the card automatically and then ejects it again afterwards.

Ball lamps are used for the light barriers, since there is very little space between the light sensor and the lamp. If you were to use lens bulbs instead, the cardboard might not be able to shut out all the light and the code would not be recognized correctly as a result.

The various steps are shown on pages 71 - 73 of the German manual.

The coded card

Cut the coded card for the reader out of medium-strength cardboard. The cardboard must be strong enough to be able to actuate the switch. Fig. 26 shows the card dimensions and the possible positions of the code holes. The corners of the card should be slightly rounded, to prevent it from getting caught on the building blocks. Which holes you punch out for the code is a secret only you must know. Of course, you must match the program to whichever hole code you choose.

There is a simple way of making the holes: first cut out the coded card and mark the holes on it with a black felt-tip pen. Then take an ordinary hole punch and empty all the "confetti" into the bin. Next, remove the cover from the underside of the hole punch (if it won't come off or if the bottom isn't see-through, this trick unfortunately won't work). Then hold the hole punch upside down (with the underside facing upwards). You should now be able to see the round punch mechanism through the hole in the metal. If you insert the card in the punch, you will be able to see through to the black hole markings as well and you can then punch the holes.

Wiring

When you wire up this model, you must make sure that there are no cables in the way of the card slot. The best method is to lay all the cables - except the power supply cable to the motor- on the side where the switch is. Connect the motor to M1, and the lamps to M2 and M3. The switch must be connected to E1, and the two light sensors to E2 and E3. Make sure that the photo-transistors are connected the right way round! It is important to use the right cable lengths.

Now start the LUCKY LOGIC program and test the wiring by selecting OPTIONS INTERFACE DIAGNOSIS in the menu. When you press the switches, the input indications shown in the test table should change accordingly. Then switch the M-outputs (motors or lamps) by clicking them with the mouse as described in the software manual. If everything is working properly, you can load and start the software.

Test table:

E1	Switch at card slot
E2	Light sensor for 1st row of holes (on right)
E3	Light sensor for 2nd row of holes (on left)
M1	Drive motor for inserting card
M2	Lamp 1 (ball lamp)
M3	Lamp 2 (ball lamp)

Software

The software for the coded card reader has been written with LUCKY LOGIC. It consists of four files, namely CODE.FTG, CODE1.FTL, CODE2.FTL and CODE3.FTL. Load the combined CODE process and start the program by selecting RUN COMBINED PROCESS in the menu.

Only the fifth hole is interpreted in the programming example (so as not to make things too complicated).

If all the holes are to be tested, the right-hand row (E2) must be checked whenever there is a hole in the left-hand row (E3). If the program recognizes the code correctly, the reader returns the card. If the code on the card is not the same as the code in the control program, the reader keeps the card. In this case, you can only get it back again by means of the diagnosis option in LUCKY LOGIC.

For Program see disk/ Annex

CD player

The CD player can read data which you enter on a paper disk in a predefined pattern, in the form of light and dark fields. The paper disk has three different tracks. On the next two pages are diagrams of what the disk must look like. Don't cut up this manual - instead either photocopy the pages or transfer the pattern to a sheet of white paper with compasses and a ruler. Fill in the data fields carefully using a black felt-tip pen (caution - don't draw over the edges!). Then cut out the two holes for the rotary axle (in the center, 4mm diameter) and the wheel axle (on the side, 17 mm).

Make sure that the hole for the rotary axle is exactly central, or the player will not be able to read your disks properly.

To insert the data disk, raise the top of the read head and fit the disk onto the axle with the Z30 gear wheel. The wheel axle secures the disk in one of the holes of the gear wheel. This axle also acts as an index cam for the reference switch.

Fig. 27 shows a coded disk for you to photocopy (or trace) and fill in yourself, while Fig. 28 contains a finished coded disk for note sequence 0 to 15 (it can be copied too).

When you make the disk, there are a few things you should remember:

- The Z30 gear wheel underneath the data disk must be level with the bottom of the read head, or the data disk will not rotate correctly.

- The reference switch at E1, which is actuated by the index cam (wheel axle) of the data disk, must be displaced if necessary, so that it is operated accurately (wheel axle between segments 13 and 14).

- The degree of transparency of the data disk decides whether the reader must be fitted with a ball lamp or with a lens bulb. If you use normal photocopy paper, a ball lamp is sufficient. If you use paper which is less translucent (e.g. recycled paper), you will have to use a lens bulb instead and fit a no. 5 building block underneath the lamp holder.

- The slide with the read head must be able to move freely. You can check this by pulling out the gear wheel and the wheel axle roughly 4 mm. Then turn the worm by hand. Please lay the cables so that they are not touching any moving parts of the model and so that they cannot get stuck between the worm and the gear wheels.

The various steps are shown on pages 82 - 85 of the German manual.

Wiring

The CD player works with two motors and a lamp. The motor connected to M1 turns the data disk, while that connected to M2 moves the read head back and forth. The lamp for the reader is connected to the interface output. The CD player has a whole series of switches for providing feedback information. The switch connected to E1 is actuated by the index cam of the data disk. The switch at E2 is turned on and off at regular intervals by the pulse wheel of the data disk drive. These two switches tell the computer exactly which field is currently being read on a particular track of the data disk. Field 1 is detected by the switch at E1. The program then counts on one each time the switch at E2 is actuated. The switch at E3 is operated by the pulse wheel at the motor of the reader; it enables the computer to select a track on the data disk. The switch at E4 determines the limit position of the read head (fully extended). When the program is started, the read head is therefore first moved to its limit position. Finally, the light sensor for reading the data is connected to E5. Make sure that the photo-transistor is connected the right way round! It is important to use the right cable lengths.

Now start the LUCKY LOGIC program and test the wiring by selecting OPTIONS INTERFACE DIAGNOSIS in the menu. When you press the switches, the input indications shown in the test table should change accordingly. Then switch the M-outputs (motors or lamps) by clicking

them with the mouse as described in the software manual. If everything is working properly, you can load and start the software.

Test table:

E1	Wheel axle of data disk (field 0)
E2	Pulse wheel at drive of data disk
E3	Pulse wheel at drive of read head
E4	Limit position of read head (fully extended)
E5	Light sensor (read data)
M1	Drive of data disk
M2	Drive of read head
M3	Reader lamp

Software

The software for this model has been written in Turbo-Pascal. This Pascal program is contained in the READCD.PAS file. Start the READCD program.

The program reads the bits on the data disk individually; their meanings are as follows: white field = 0, black field = 1. The disk has room for 3 tracks with 25 fields each, in other words a total of 75 bits. Since it is only possible to represent two different notes with each bit however, four consecutive bits are always used to code a note - just as shown in Fig. 29. 16 different notes can be represented in this way. The codes are carried over from one track to the next (the last bit of track one and the first 3 bits of track 2 make one note, as do the last two bits of track 2 and the first two bits of track 3). When the program is started, all three tracks are read one after the other, and then reproduced by the computer as a sequence of notes. The coded disk shown in Fig. 28 represents all 16 notes. You can store your own tunes by filling in the fields on the blank coded disk (copy!) with a black felt-tip pen - or, if you wish, you can use the disk for something else in a program you have written yourself.

For Program see disk / Annex

How the CD works

You are bound to be familiar with the shiny music CDs, on which music has been recorded in a totally different way to conventional records. The groove on an ordinary record has tiny "peaks" and "valleys", which are scanned by the stylus and converted into electrical current fluctuations. These current fluctuations are boosted electrically, and then made audible again in a loudspeaker by the diaphragm inside the speaker moving back and forth synchronously with them. The music on a CD (compact disk) is stored completely differently. When the music is recorded, the notes which are picked up by the microphone are stored as numeric (digital) values. In fact, to be more precise, it is only the volume which is stored as a digital value - it is stored several thousands of times per second however. The columns of numbers which are obtained in this manner reproduce the music extremely precisely. These numbers are then stored on the CD as an almost infinite series of microscopically small light and dark points. A normal lamp is no longer any good for differentiating between these points, because the beam of light would be much too wide. This is why a CD player uses a laser beam instead.

Laser is a special method of creating a beam of light with extraordinary properties. The light has only one color, whereas ordinary white light, on the other hand, is made up of a number of different colors. Normal light is propagated (spreads out) in all directions; laser beams on the other hand are projected more or less parallel to one another. In contrast with normal light, the light waves of a laser beam all oscillate synchronously. The light is extremely bright and intensive as a result.

A CD player projects a very fine laser beam onto the disk, so that the information contained in each point can be scanned individually. The reproduction process is the reverse of the recording process.

CDs are also used in computer engineering - when it comes down to it, it is irrelevant whether you store music or any other form of data. A single CD has room for more than 600 million letters, corresponding to roughly 350,000 typewritten pages, which would make a tower about 17 m high if piled on top of one another. It is however also possible to store images and tones on a computer CD, which is known by the way as a "CD ROM".

Sorting machine

Sorting and separating are two very important operations in the automatic manufacturing process for goods or products of all types. ("separating" means passing on individual parts, belonging to a large quantity which is ready for further processing, one at a time.) The sorting machine enables different parts to be recognized and kept separate from one another.

The various steps are shown on pages 95 - 98 of the German manual.

This model sorts out fischertechnik building blocks of different lengths. They are placed onto the chute with the peg or pin facing upwards. They are initially held in position by the sliding bar. As soon as you start up the unit, it separates the building blocks and passes them on one at a time. The first building block then drops onto the conveyor. At the end of the conveyor there are two light barriers made up of two light sensors and a lens bulb.

If a long building block is conveyed past the light barriers, they will both be interrupted temporarily. The building block will be recognized as a long part, and the belt separator which is arranged perpendicular to the conveyor takes it to the tray for long building blocks. If a short building block is conveyed past the light barriers, only one of them will be interrupted at a time and the belt separator will move in the opposite direction to that described above. The program can measure the distance travelled by the belt separator with the aid of a pulse wheel, and stop it when the building block has dropped off the belt.

Setting the switches for the sliding bar

Position the finished model in front of you with the feeding chute on the left and the belt separator on the right. There are two switches on the sliding bar, which provide feedback

information on the bar's position. To set the switches, pull the reducing gear away from the motor about 2mm, so that you can move the sliding bar by hand. Then position a building block upright in the feeding chute (with the peg facing upwards) in front of the bar. Push the sliding bar forward as far as it will go and adjust the position of the right-hand switch so that it just about opens at the edge of the gear. Now pull back the bar so that the next building block can slide along freely. Adjust the position of the left-hand switch so that it just about opens with the sliding bar in this position. Finally, move the reducing gear back to its original position.

Wiring

(See page 99 of the German manual)

The motor connected to M1 drives the belt separator, while that connected to M2 drives the conveyor. The motor at M3 is responsible for the movement of the sliding bar, and the lamp is connected to output M4. The switches at E1 and E2 report the limit positions of the sliding bar (E1 = left-hand switch, E2 = right-hand switch). The first photo-transistor (as seen in the direction of the conveyor belt) is connected to E3 and the second photo-transistor to E4. The switch which is actuated by the pulse wheel of the belt separator is connected to E5. Make sure that the photo-transistors are connected the right way round! It is important to use the right cable lengths.

Now start the LUCKY LOGIC program and test the wiring by selecting OPTIONS INTERFACE DIAGNOSIS in the menu. When you press the switches, the input indications shown in the test table (page 27) should change accordingly. Then switch the M-outputs (motors or lamps) by clicking them with the mouse as described in the software manual. If everything is working properly, you can load and start the software.

Test table:

E1	Sliding bar limit position (chute clear)
E2	Sliding bar limit position (in chute)
E3	Light sensor (front)
E4	Light sensor (back)
E5	Pulse wheel for belt separator
M1	Belt separator drive
M2	Conveyor drive
M3	Sliding bar drive
M4	Lens bulb

Software

The software for the sorting machine has been written with LUCKY LOGIC. It comprises five files, namely SORTING.FTG, SORTING1.FTL, SORTING2.FTL, SORTING3.FTL and SORTING4.FTL. Load the combined SORTING process and start the program by selecting RUN COMBINED PROCESS in the menu. Processes nos. 2 and 4 merely switch on the conveyor belt and the lamp permanently; only the other two processes are therefore important as regards full operation of the machine. Process 1 controls the belt separator. As long as neither of the two light barriers is interrupted (i.e. the photo-transistor is not lit), the belt separator does not move (E3 and E4 set to 1). If one of the light barriers is interrupted by a long building block, the motor of the belt separator (M1) starts rotating clockwise. If there is a low building block on the belt separator, on the other hand, the motor M1 rotates in the opposite direction. The second light barrier ensures that the belt separator only moves if there is a building block on the conveyor. The belt stops again after a fixed time (20 pulses of the pulse wheel of the belt separator).

Process 3 controls the sliding bar for separating the building blocks. The bar first moves right back (a block moves over in front of it) and then right forward (the block drops onto the conveyor). The sliding bar does not move again until the building block has passed the light barrier-this is important in order to ensure that a big enough distance is left between each building block.

For Program see disk / Annex

Sorting machine

(see page 98 of the German manual)

You can use the sorting machine to sort 7.5 and 15 building blocks. You will have to adjust the two photo-transistors E3 and E4 to match the height of the building blocks.

Parcel-flipping machine

Parcels must be flipped over until the address is facing upwards, so that when they are distributed to the destination Post Office, no time is wasted. The parcel-flipping machine turns a parcel, which has been fed to it via a chute, until the black side is on top. (This corresponds to the side with the address label in a real parcel-flipping machine.) By the way, letters are also flipped automatically in a distributing Post Office until the stamp is in the top right-hand corner, so that they can be stamped automatically (the stamps are printed in a color which lights up when it is irradiated with ultraviolet light).

The various steps are shown on pages 105 - 108 of the German manual.

The parcel is pushed sideways along the chute towards the belt. The switch on the chute is actuated and the unit is switched on automatically. The conveyor transports the parcel underneath the reflected light barrier and then comes to a standstill (this is easy to see from the second, horizontal light barrier). The program can tell whether or not the black side is already facing upwards from the different reflections of the yellow and black surfaces.
If the reflected light barrier detects a yellow surface, the parcel must be turned. The conveyor belt starts up again and conveys the parcel on to the flipping station. The parcel is raised by means of the vertical chain and the crawler track links secured to it, and flipped over by the moving conveyor belt. To make sure that the parcel is only flipped over once, there is a switch at the rear of the turning station which is actuated by the crawler track links (when this happens, the computer switches off the vertical chain again). The parcel is then conveyed back to the light barrier. The distance travelled by the conveyor belt is reported to the computer by means of a switch with a pulse wheel. The entire procedure is then repeated until the black side is on top.
When the reflected light barrier recognizes the black surface, the parcel is sitting correctly. The conveyor then returns from the light barrier, and the parcel is conveyed by the sliding bar

from the conveyor belt to the delivery chute. The unit is then ready to accept the next parcel. The two light barriers should be fitted with lens bulbs if possible. Make sure to tighten the chuck nut of the locking worm and the hub nuts of the gear wheels properly!

Setting the switches for the sliding bar

Position the finished model in front of you with the 28-pin socket connector facing towards you. To set the switches, pull the mini-motor rack away from the motor about 2mm so that you can move the sliding bar by hand. Now move the bar forward past the conveyor to the delivery chute. Adjust the position of the left-hand switch, so that it is just about actuated at the edge of the reducing gear (hold the sliding bar firmly with your other hand). Then pull back the bar until the conveyor is completely clear and adjust the position of the red mounting plate so that the right-hand switch just about closes (hold the sliding bar with your other hand again). Finally, return the reducing gear to its original position.

Wiring

(See page 110 of the German manual)

This time the interface is used to almost its full capacity, which means you must be particularly careful when wiring up the model. The motor connected to M1 moves the sliding bar. The motors at M2 and M3 drive the turning station and the conveyor. The lens bulbs of the two light barriers are connected to M4. The switches at E1 (eject) and E2 (belt clear) report the position of the sliding bar. The switch at E3 is actuated by the pulse wheel of the conveyor belt drive, the switch on the chute is connected to E4 and the switch at the flipping station to E5. The two photo-transistors are connected to E6 (horizontal light barrier) and E7 (reflected light barrier). It is important to use the right cable lengths.
Now start the LUCKY LOGIC program and test the wiring by selecting OPTIONS INTERFACE DIAGNOSIS in the menu. When you press the switches, the input indications shown in the test

table should change accordingly. Then switch the M-outputs (motors or lamps) by clicking them with the mouse as described in the software manual. If everything is working properly, you can load and start the software.

Test table:

E1	Sliding bar limit position (eject parcel)
E2	Sliding bar limit position (conveyor clear)
E3	Pulse wheel on conveyor drive
E4	Trip contact for chute
E5	Switch at flipping station (vertical chain)
E6	Horizontal photo-transistor on conveyor
E7	Photo-transistor for reflected light barrier
M1	Sliding bar drive
M2	Flipping station drive
M3	Conveyor drive
M4	Lens bulbs of the two light barriers

Software

The software for the parcel-flipping machine has been written with LUCKY LOGIC. It consists of five files, namely PARCEL.FTG, PARCEL1.FTL, PARCEL2.FTL, PARCEL3.FTL and PARCEL4.FTL. Load the combined PARCEL process and start the program by selecting RUN COMBINED PROCESS in the menu.

The first process controls the sliding bar for ejecting the parcel. The bar is initially moved to its home position (right back). The process then waits until an order is issued by process 3 via VAR 1 to eject the parcel (the sliding bar moves forward and then back again).

Process no. 2 controls the flip belt. Initially the flip belt carries on running until the switch at E5 is closed by a crawler track block. The belt then stops and waits for an order from process no. 3 (VAR 2). The flip belt then runs again until the next time the switch at E5 is actuated, in order to flip the parcel over.

The third process controls the conveyor, and issues orders to process no. 1 (eject) and process no. 2 (turn). When a parcel is loaded (this is reported by the switch at E4), it is conveyed until it is underneath the light barrier. The barrier at E6 determines when the parcel has been conveyed far enough. The brightness of the surface is then measured by the light barrier at E7. If the top side is black, the parcel is conveyed to the sliding bar (5 pulses of the pulse wheel at the conveyor belt) and the bar order is issued to process 1 (VAR 1).

If the surface is yellow, the parcel is conveyed to the flipping station and the order to turn it over is issued to process no. 2 via VAR 2; the parcel is then conveyed back underneath the light barrier and the surface brightness is measured again - this procedure is repeated until the parcel is ejected.

For Program see disk/ Annex

Parcel-flipping machine   (See page 108 of the German manual)

Use the three white 30x45 mounting plates and the black 30x45 plate for the "parcel" instead of the yellow 30x45 plates.

White

White

Black

White
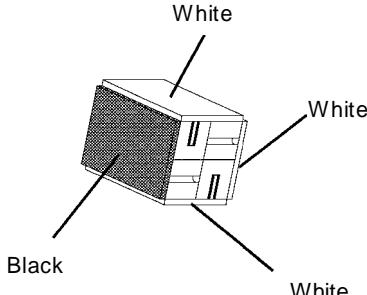
Curve tracer

The curve tracer (or recorder) can be used in two different ways. If the paper is moved forward a little at a time at regular intervals (e.g. once every second), the time history of the recorded data can be represented. If, for example, you move the pen a little to the right in the program every time the paper is fed one step until it arrives at the right-hand margin, and then move it a little to the left with every step until it arrives at the left-hand margin, the tracer will plot a wavy line.

The engineer calls this type of tracer an "Xt recorder ("t" is the technical abbreviation for time and "X" represents the horizontal movement of the pen).

If the paper is moved back and forth by the program, the pen can draw over the entire sheet of paper. This is what the engineer calls an "X Y recorder" ("X" and "Y" are the two directions in which the paper and the recorder move). If only the pen moves, or only the paper, then horizontal or vertical lines are plotted.

The chart paper should be 14cm wide, while the length is unimportant. To insert the paper, feed it in from the recorder side until it is taken up by the rollers. You can control the paper feed motor with the diagnosis function of LUCKY LOGIC in order to set the chart to its home position.

Set the pen so that when it is raised, its tip is roughly 3 to 4mm above the paper. This is also the setting in which the switch for the pen position is actuated. If you stop drawing for some time (or if you aren't using the recorder any more), you should put the cap back on the pen again to prevent it from drying out.

The recorder will move more easily if you apply a drop of household oil to the guide rod underneath the drive worm. Make sure that the hub nuts of the gear and driving wheels, and the chuck nuts of the worms, are properly tightened.

The various steps are shown on pages 117 - 121 of the German manual.

Wiring

(See page 123 of the German manual)

The cables should be laid so that they cannot interfere with operation of the model. Bunch the long cables for driving the pen together in the middle with a paper-wire strip. The motor connected to M1 moves the recorder, the motor at M2 raises and lowers the pen and that at M3 moves the paper. The switch at E1 is actuated by the pulse wheel on the recorder drive. The switch at E2 is actuated when the pen is raised. The switch at E3 is actuated by the pulse wheel on the chart drive, while that at E4 reports the limit position of the recorder. You must make a cable out of two sections (25cm and 30cm) for the connection to the switch at E4. To do so, replace the plugs at one end of one of the two cables with plug sockets. It is important to use the right cable lengths.

Connect a sensor to the EX input (e.g. a potentiometer or a photo-transistor). The change in the electrical resistance over the course of time is then plotted by the curve tracer.

Now start the LUCKY LOGIC program and test the wiring by selecting OPTIONS INTERFACE DIAGNOSIS in the menu. When you press the switches, the input indications shown in the test table should change accordingly. Then switch the M-outputs (motors or lamps) by clicking them with the mouse as described in the software manual. If everything is working properly, you can load and start the software.

Test table:

E1	Pulse wheel for recorder drive
E2	Pen position (1 = up, 0 = down)
E3	Pulse wheel for paper feed
E4	Recorder position
EX	Sensor input (e.g. potentiometer)
M1	Recorder drive (cw/ccw)
M2	Raise/lower pen
M3	Paper feed drive

Test table:

| E1	Pulse wheel for recorder drive
E2	Pen position (1 = up, 0 = down)
E3	Pulse wheel for paper feed
E4	Recorder position
EX	Sensor input (e.g. potentiometer)
M1	Recorder drive (cw/ccw)
M2	Raise/lower pen
M3	Paper feed drive

Software

The software for this model has been written in Turbo-Pascal. The Pascal program is contained in the KENNLINI.PAS file. Now start the KENNLINI program.

The pen can move right across the paper from one end to the other in 360 steps. When the program has read the EX input, it converts the measured value to ensure that the limits of the paper are not exceeded (i.e. so that you do not plot beyond the margins). The recorder represents the electrical resistance of a sensor at the EX input. You can try this out, for example, using one of the photo-transistors in the construction kit (make sure you connect it the right way round!) by slowly darkening it (or alternatively you could set up the recorder in a dimly lit room).

The chart paper is fed on one step every second, so that the time history of the input value at EX is plotted. You can also work with different times, e.g. one minute instead of one second, by altering the program.

For Program see disk/ Annex

Curve tracer   (See page 120/121 of the German manual)

Step 4

E4
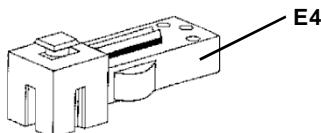
Step 5
Fit only one switch (E1) here.

Plotter

A plotter is a drawing machine which can be controlled by a computer. The pen of the plotter can be moved in two directions, which means that you can produce any drawings with the aid of the program. The drawing paper is approximately 15 x 21cm in size (DIN AS format, like a school exercise book). Stick the sheet of paper to the experiment base plate with adhesive tape. If the paper is very thin, you can place a sheet of cardboard underneath it or simply lay several sheets of paper on top of one another.

Set the pen so that when it is raised, its tip is roughly 3 to 4mm above the paper. This is also the setting in which the switch for the pen position is actuated. If you stop drawing for some time (or if you aren't using the recorder any more), you should put the cap back on the pen again to prevent it from drying out.

The plotter works in a similar way to the X t recorder. The pen is moved in the lengthways direction (along the "X-axis") by moving the slide along the two worm drives, which are linked together by a chain drive. The position of the slide is fed back by a switch with a pulse wheel. A line in the lengthways direction can be measured in this way in sections of about 0.5 mm. The pen is moved on the slide in the crossways direction (along the "Y-axis"). The length of the line in this direction is determined by means of reference pulses from a light barrier above the chain. Whenever the light beam is interrupted by a link of the chain, the light barrier supplies a "0"; otherwise it supplies a "1". One pulse corresponds to a line section approximately 4.5mm long. You must be very careful when you assemble the slide, to make sure that the pen holder can move easily and precisely along the series of building blocks with the rack. It is advisable to apply a drop of household oil to the guide axle.

When you switch the plotter on, the pen will be positioned somewhere above the drawing surface. Reference switches are provided for both the directions of movement, so that the program has a fixed starting position for counting the steps. The program retracts the pen in the lengthways and crossways directions until both the reference switches are closed. This position is then the home position for your drawing.

Make sure that all the hub nuts of the gear wheels and the chuck nuts of the worms are properly tightened.

The various steps are shown on pages 129 - 132 of the German manual.

Wiring

(See page 134 of the German manual)

The cables should be laid carefully and secured together with a paper-wire strip in several places, so that they cannot interfere with the movement of the slide.

The motor connected to M1 moves the slide along the worms in the X-direction, while the motor at M2 moves the recorder in the Y-direction. The motor at M3 raises and lowers the pen and the lamp for the light barrier is connected to M4. The switch at E1 is actuated by the pulse wheel for the X-axis. The switches at E2 (X-direction) and E3 (Y-direction) report the limit positions of the plotter. The switch at E4 is actuated when the pen is raised and the photo-transistor is connected to E5.

You must make a cable out of two sections (25cm and 35cm) for the connections to the switch at E4 and to the photo-transistor at E5. To do so, replace the plugs at one end of one cable with plug sockets. It is important to use the right cable lengths.

Now start the LUCKY LOGIC program and test the wiring by selecting OPTIONS INTERFACE DIAGNOSIS in the menu. When you press the switches, the input indications shown in the test table should change accordingly. Then switch the M-outputs (motors or lamps) by clicking them with the mouse as described in the software manual. If everything is working properly, you can load and start the software.

Test table:

| E1	Switch for pulse wheel (X-axis)
E2	Recorder limit position (Y-axis)
E3	Slide limit position (X-axis)
E4	Pen position (1= up, 0 = down)
E5	Photo-transistor
M1	Worm drive (X-axis)
M2	Rack drive (Y-axis)
M3	Raise /lower pen
M4	Lens bulb

Software

The software for this model has been written in Turbo-Pascal. The Pascal program is contained in the PLOTTER.PAS file. Now start the PLOTTER program.

The plotter software contains the most important procedures for controlling the model: initialize plotter, raise/lower pen and go to any position on the paper. The example produces a very simple drawing. All you need to do to make your own drawings is to alter the instructions in the main program. The comments in the listing tell you exactly how the plotter works.

For Program see disk/ Annex

Plotter   (See page 132 of the German manual)

Fit two switches (E1 and E2) in the step shown on page 132.

E2
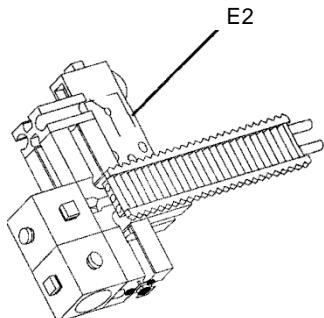
Robot

Robots are used above all in factories, where they often have to work together with other automatic machines. The majority of them are bolted to the floor and perform all their work at this fixed location. We mentioned in the introduction that robots don't look anything like artificial human beings. Their design depends instead on the tasks they must perform: a robot which wires electronic circuits is small and slenderly built - the robots that fit car bodies onto the chassis on the other hand are monsters that weigh a ton.

The fischertechnik robot moves in two directions - it can either turn about itself or raise and lower its "arm". At the end of its arm is a gripper for picking up different objects. The construction kit contains a yellow tubular sleeve with a base, which the gripper finds easy to handle. You can stick the enclosed adhesive disks onto the surface of the gripper, to enable it to grip small parts with a smooth surface more easily. The space in which the gripper works is circular, with a radius of about 30cm. As with the previous models, switches for the limit positions and switches with pulse wheels are used to tell the computer the position of the turntable and the arm. There is one robot position in which all the limit-position switches (= reference switches) are actuated. The robot must be set to this position when the program is started. The computer can then count the pulses which are supplied to it by the pulse switches at the three drive motors.

When you assemble the model, it is very important to tighten the hubs of the gear wheels properly. The pickup arm which is driven via the chain is very heavy in particular. The drive motor for the gripper is mounted at the rear of the pickup arm to counterbalance its weight as far as possible; the gripper is operated via a shaft with a universal joint.

The various steps are shown on pages 141 - 149 of the German manual.

Wiring

(See page 151 of the German manual)

All the cables - with the exception of the supply cables to the turntable motor and to the pulse switch - are laid through the hole in the center of the turntable, so as not to interfere with the movement of the robot.

The motor connected to M1 turns the robot, while the motor at M2 raises and lowers its arm and the motor at M3 opens and closes the gripper. The switch for the limit position of the gripper is connected to E1, and the pulse switch for the turntable drive to E2. The switch at E3 is closed when the arm is set to its limit position (raised). The pulse switch at E4 is actuated by the gripper drive and that at E5 by the arm drive. Finally, the switch for the limit position of the turntable is connected to E6. You must make cables out of two sections (30/35 cm for E1 and 25/40 cm for E5) for the connections to the switches at E1 and E4. To do so, replace the plugs at one end of one of the two cables with plug sockets. It is important to use the right cable lengths.

Now start the LUCKY LOGIC program and test the wiring by selecting OPTIONS INTERFACE DIAGNOSIS in the menu. When you press the switches, the input indications shown in the test table should change accordingly. Then switch the M-outputs (motors or lamps) by clicking them with the mouse as described in the software manual. If everything is working properly, you can load and start the software.

Test table:

E1	Gripper limit position (fully open)
E2	Pulse switch for turntable drive
E3	Arm limit position (fully raised)
E4	Pulse switch for gripper drive
E5	Pulse switch for arm drive
E6	Turntable limit position
M1	Turntable drive
M2	Raise/lower arm
M3	Open/close gripper

Software

The software for the robot has been written with LUCKY LOGIC. It consists of four files, namely ROBOT.FTG, ROBOT1.FTL, ROBOT2.FTL and ROBOT3.FTL. Load the combined ROBOT process and start the program by selecting RUN COMBINED PROCESS in the menu.

The example shows a very simple action of the robot's arm. The robot picks up the tubular sleeve in one place and puts it down again somewhere else. The important thing with this model is the interaction of the three processes. First of all the robot's arm is moved to its home position - all the limit-position switches are closed. Then, starting at this position, the target positions of the various components of the robot (turntable, arm, gripper) can be specified by counting the number of pulses supplied by each pulse wheel.

Process no. 1 controls the turntable. After the first revolution, the process waits until the gripper has picked up the tubular sleeve and then continues to the second position. This time it waits until the sleeve has been put down. The rotary angle can be defined by means of the number of pulses supplied by the pulse wheel (switch at E2).

The principle of the second process (raising and lowering the arm) is similar. Before the arm can be lowered, the process must wait until the turntable (process no. 1) has reached the desired position. A "wait loop" of this type is required for every turntable position.

The third process controls opening and closing the gripper. How many steps of the pulse wheel are necessary to pick up an object securely depends on its diameter. In the case of the tubular sleeve 30 steps are needed. You will have to find out how many you need for other objects by trial and error.

If you want to program your own sequences of movements, the best way to do so is to start by creating and testing every single process. Leave room in the circuit at the points at which the program will have to wait for a ready message from the other two processes, so that you can incorporate the queries later on. If the single processes work correctly, you can insert the queries they issue to one another and then test the combined process. It is not always easy to coordinate the three different movements - but with a little patience and practice we are sure you will manage.

For Program see disk / Annex

Power supply (230V~ / 8V=)

**We recommend the fischertechnik power supply, (230V~ / 8V=)
Art. no. 30180.**

Caution

Be careful not to hurt yourself on components with sharp or pointed edges!

If you have technical problems, don't hesitate to get in touch with us at:

fischertechnik Service Department
D - 72178 Waldachtal
Phone 07443/12-4369
Fax 07443/12-4591
Email: fischertechnik-service@fischerwerke.de