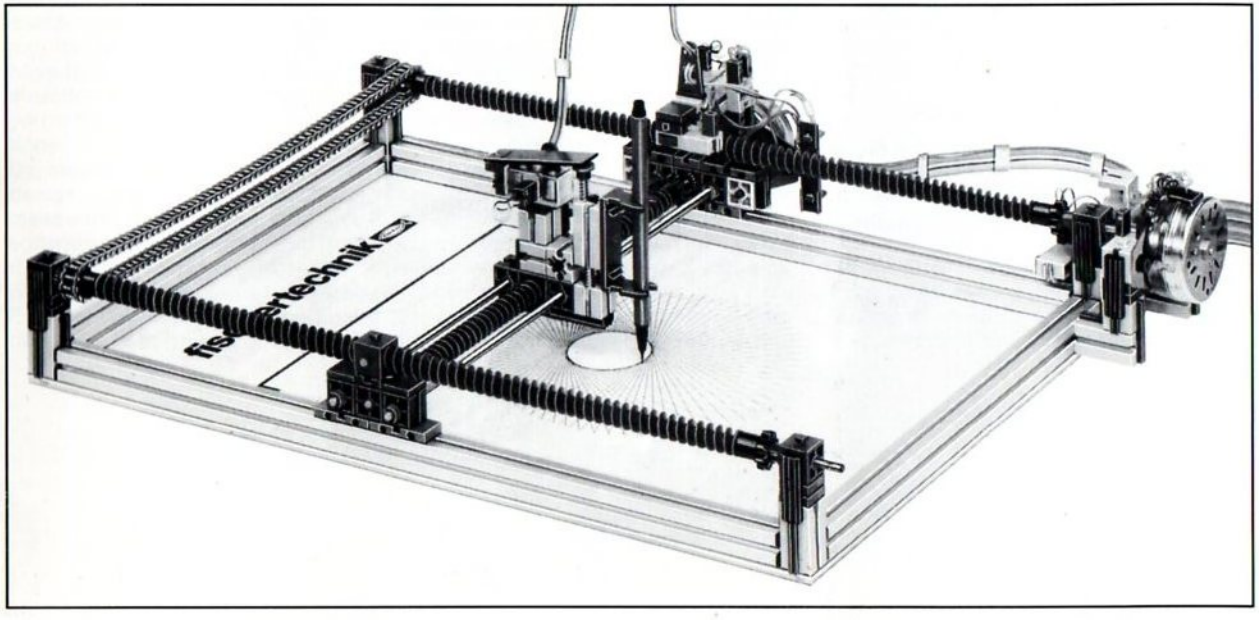
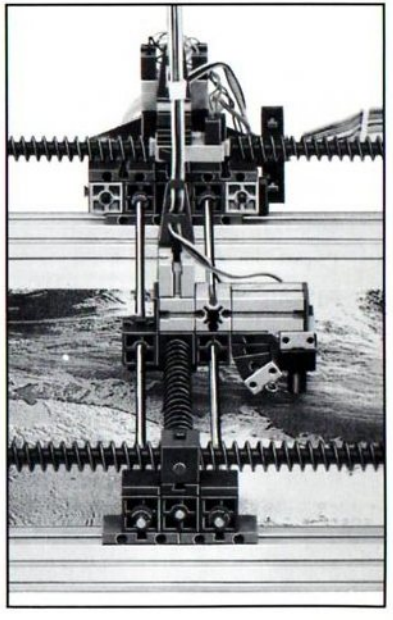
fischertechnik®
COMPUTING

Instructions Plotter/Scanner





Index

Introduction	3
Control of stepping motors	4
Interface and software	5
Set up of the plotter	6
Control of the plotter	6
Plotter software	9
HOME	
LINE	
RLINE	
MOVE	
RMOVE	
SET ORIGIN	
ORIGIN	
CHAR	
RCHAR	
SCALE	
Plottertools	12
AXIS	
CIRCLE	
BOX	
Use of the plotter routines	12
User programs	16
FUNCTION	
PARAM.F	
D3	
Scanner operation	17
Digital image recording	18
Image evaluation	18
Pattern recognition	19
CNC-drilling machine	19
Printout of the programs	19

fischertechnik computing Plotter/Scanner

Dear friend of fischertechnik,

there is hardly any other technical instrument which is as versatile in application as a computer. One of the most fascinating fields of computer technology, however, is the control of technical models. Buying the fischertechnik computing kit PLOTTER/SCANNER you not only acquired a model but also a fully operative peripheral unit for your computer. This is possible on account of a skilful utilization of precise components but also on account of special parts like e.g. the two stepping motors. And yet you do not have to be limited due to the advantages of a construction kit: during assembly and upon application you will see what way and manner the device functions.

By means of the plotter you can prepare drafts, designs, mathematical functions, diagrams and measuring data. This is simplified by a graduated software concept leading you from the first steps of the stepping motor via elementary subroutines of a plotter to user programs of versatile applicability. Even an entire library of characters for lettering of drawings is at your disposition. The software has been explained; at each step you may extend or convert the software according to your own ideas.

Apart from purely technical applications, the aesthetics of computer graphics must not be neglected. Fantastic graphs can be drawn by means of the plotter, and on top of it all the plotter can be converted to a scanner by a few manipulations only. Whereas the plotter is an output device, i.e. data existing in the computer is committed to paper "externally", the scanner is an input device. In this not commonly known configuration, the writing mechanism of the plotter is substituted by a recording head by means of which the gray scales of the original are read. Hence the flow of information goes from the original to the computer.

The software of the scanner is future-oriented. The methods of digital recording of images and their evaluation are discussed. The question is within which shortest period of time a computer program might recognize again an object shown only once. Thus, question of robotics are refered to. You can think yourselves into the recording head of the scanner regarding it as a robot exploring its environment. An advantage which cannot be contradicted: with a few strokes you can have a sketch of the environment of the robot on paper which has been inserted beneath the recording head.

There are still a lot of other applications which may be realized by means of the plotter/scanner. What about a CNC drilling machine? Or a X-Y-coordinate table? You may tackle these projects, too, because whatever may still be missing for the realization of your individual application, it may easily be supplemented out of the large program of fischertechnik. Of course, all parts are matching and can be arbitrarily combined.

I am convinced that the fischertechnik computing plotter/scanner will motivate you to perform quite a series of your own further experiments and will increase your knowledge and experience in this field considerably.

Yours faithfully,

Control of the stepping motors

One of the most important components of the plotter/scanner are the two stepping motors for driving the two axles. Stepping motors differ principally from the D.C. motors with which perhaps you might be more familiar. D.C. motors are e.g. the three fischertechnik motors, the 6-V-motor, the mini-motor and the S-type motor. These D.C. motors – as defined by the name – only need direct voltage for operation. Such direct voltage may be supplied by batteries or a power pack. In most cases, there will be still a switch between the power pack and the motor. In that way the supply voltage can be switched off and on by the switch. In addition, for some types of switches it is possible to choose the polarity of the voltage for controlling the sense of rotation of the motor. In our case, the switch is formed by the fischertechnik computing interface as we intend to control the models by means of a computer.

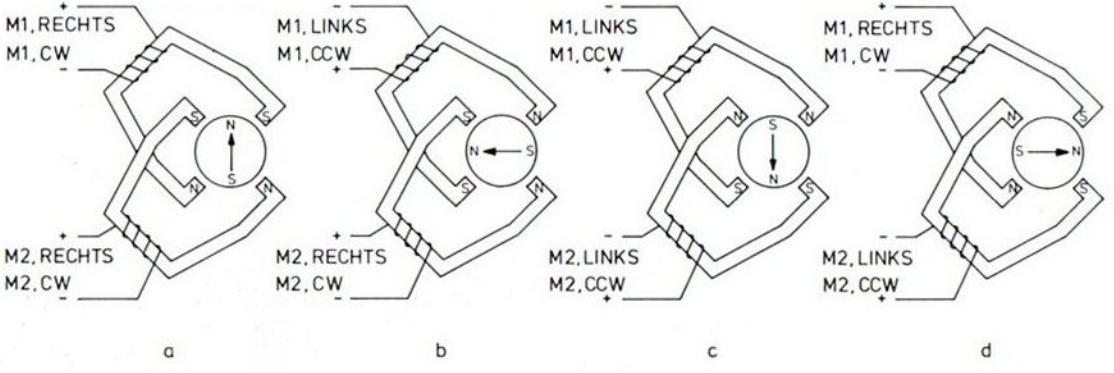
Compared to such operation of D.C. motors, there are some differences when using stepping motors. The first difference you will observe, compared to D.C. motors, is that the stepping motor is provided with four instead of two connections. The stepping motor is equipped with two magnet systems which are controlled independently from each other. Fig. 1 shows a schematic graph of a stepping motor. The two magnet systems are staggered by 90 degree. A permanent magnet has been arranged between the magnetic poles which is connected to the motor shaft. In order to simplifiy matters, the latter has been shown in the figure as a compass needle. This compass needle will now take a position according to the magnetic field of the two coils. Take one stepping motor out of the kit and fix it in the small supporting bracket. In this way and manner you can observe the motor without problems. Now hook up the stepping motor to the interface. For this purpose you connect the red and the black strand of the stepping motor to M1. Hence, the output M1 supplies the magnetic system 1 of the stepping motor. The green and the grey strand of the stepping motor are connected to M2, so that the magnetic system 2 is supplied via M2. The colour-coded flat strip cable serves for connection to the interface. One end of the cable has been equipped with a 20-pole plug which is plugged into the interface. With the cable leaving the interface to the right side, the lowest core of the cable will be brown and the topmost black. All colours, however, occur twice, in the following denoted e.g. as red 1 and red 2. From below upwards numbering is effected in the order: brown 1, red 1, ... black 1, brown 2, ... black 2. You will find M1 on the cables orange 2-yellow 2, M2 on green 2-blue 2. Provide these four cores – as described later in details – with four fischertechnik plugs. Here again the cable connections according to colours:

interface orange 2 – stepping motor red
interface yellow 2 – stepping motor black
interface green 2 – stepping motor green
interface blue 2 – stepping motor grey

Important: Interface and plotter must be supplied from a power unit of sufficient capacity. We recommend the fischertechnik computing power unit, which can provide up to 2 amps of current at the correct voltage. Optionally, the supply may also be effected by means of two power units, supplying 1 amp each.

Now load the diagnostic routine from the fischertechnik floppy disk or cassette. In case you should not yet be acquainted with this program: it serves for controlling directly by the computer keyboard all outputs of the interface. Moreover, all inputs are monitored and displayed on the screen. The latter is not yet required for the time being. Select output M1 by actuation of key "1". It is indicated in inverse character font. Subsequently press the key as shown in the menue so that M1 is switched in "clockwise" pole direction. Now you select output M2 by actuation of key "2". By pressing the appropriate key, the same pole direction is achieved as for M1.

Fig. 1



Contrary to D.C. motors, the stepping motor does not start running when applying the operating voltage. Perhaps, you will only note a little jerk of the stepping motor but now the stepping motor keeps the motor shaft in its position by applying an astonishing force. In order to understand this behaviour we again should have a look at the model of the stepping motor of fig. 1a. By the currents of M1 and M2 a magnetic field is generated in the stepping motor in which the two north-poles are at the bottom side and the two south-poles at the upper side. Hereby, the motor shaft with the compass needle adjusts in a way that the north-pole points between the two south-poles pulling with united strength. Correspondingly, the south-pole of the compass needle is attracted by the two north-poles on the left. Just to remember physics at school: different magnetic poles attract each other, equal magnetic poles are repelling each other.
In order to move the motor we alter the pattern of the magnetic fields generated by M1 and M2. Switch the outputs M1 and M2 in the following order:

M1 counter-clockwise
M2 counter-clockwise
M1 clockwise
M2 clockwise

In doing this observe the shaft of the stepping motor. At each change of pole direction it will continue to turn a little bit. Fig. 1b through 1d show the interrelation between the pole directions and the positionings of the shaft. It should be said on this occasion that the stepping motor does not perform any quarter turns as may perhaps be assumed on account of the drawing. In reality, the magnetic systems of the stepping motor are provided with a higher pole number so that the stepping motor performs steps of 7.5 degree. Thus, after four steps, the initial position is reached again from the electrical point of view, whereas a full rotation is achieved after 48 steps only.

Since it is laborious to steadily change the pole direction of the two motor outputs by hand, it is simpler to have the same performed by computer program. The STEP program later in this manual controls the stepping motor. You may choose whether the next step of the motor should be performed upon actuation of a key or after a predetermined period of time.
The program also shows the great advantage of stepping motors. By detailed recording of the individually generated steps the position of the stepping motor respectively of the driven mechanisms can be recognized at any time. Consequently, no additional position indicators are required, as in case of D.C. motors. It is exactly this characteristic which has made stepping motors so popular in computer-controlled appliances of all kinds. Perhaps, your computer installation is already equipped with the one or the other stepping motor. The drive of read/write-heads of floppy disk drives, of the matrix printing head and the paper feed of the printer are effected by stepping motors. Add to this the write head control of the fischertechnik computing plotter.

Interface and software

We should start with a brief remark regarding the documentation of the programs for fischertechnik computing. In the instruction manual the programs have been printed in the notation of Commodore 64. With the appropriate interface for your computer, a floppy disk or cassette is delivered which also includes the programs. The BASIC notations of the various computers differ slightly. If you should not own a Commodore 64 but another type of computer, the program on the floppy disk or cassette will not exactly coincide with the program printed here. It has already been adapted to the respective type of computer. Those points in which deviations will occur at any rate, are marked by an asterisk at the beginning of the respective line in the print of the program. If you now want to compare the printed program with the program read in you must pay special attention when an asterisk appears. The instructions for the interface will give you further advice and assistance for adapting the programs.
The instruction manual of the interface also includes an explanation as to how the signals of the interface are processed resp. generated by BASIC. It should be mentioned that the control of an output is effected by calling in a machine language program. The calling parameter is composed by the number of the output M1, M2, M3 or M4 and the operational mode CW (clockwise), CCW (counter-clockwise) or OFF. Some examples are:

```
* SYS M1,CW
* SYS M3,CCW
* SYS M4,OFF
```

First of all, however, the command

```
* SYS INIT
```

has to be given bringing the interface into an initial state. Simultaneously, all motors are switched off so that this command may serve for a simultaneous switch-off of all motors, too. The inputs of the interface are operated by the USR-function. By the parameters E1 through E8, the eight inputs are interrogated to which the microswitches are connected. Other ON-OFF-signals may be put in here, too.
The functions USR(EX) and USR(EY), however, serve for the input of gradually variable electrical values. Input EY will be used later for interrogation of the photoresistor in the recording head.
It is also important to know that the interface incorporates a monitoring circuit for data transfer. If within half a second no new command – either output or input command – should be given, all motors will be

switched off by this circuit. When stopping the computer program you thus need not switch off the power supply of the motors. When the transfer of data is initiated again, the interface will operate all motors again as before.
The machine language program effecting the data transfer between computer and interface must also be stored in the computer. The so-called driver routine serves for this purpose which is also included in the floppy disk resp. cassette. As well, it is a part of any other fischertechnik computing program occupying the line numbers 1 through 500. The driver routine is documented in the instructions for your interface. In the program lists of this instruction, this part, however, will not appear as it will differ according to the type of computer concerned. The machine language program must fit in all details to the structure of the hard and software of the computer.
If you do not use a fischertechnik computing interface but another interface circuit, the information given hitherto will not apply in every detail. Anyhow, the ideas outlined above can be realized with any other appropriate hardware.

Set up of the plotter

Refer to the main manual to see how to assemble the complete plotter. Take to pieces the test assembly for the motor and set up the plotter. Having completed the mechanical assembly of the plotter, check the two driving shafts again for easy-running. Subsequently, the model is connected at the interface. In this respect, refer to the scheme for the flat ribbon cable on the following pages, fig. 2. Bear in mind that the cut-off sections of the main cable are still needed for further connections. Now the cable ends are carefully installed over a length of approx. 3 to 5 mm without damaging the fine cores of the strand. The cores are then twisted. Loosen the small screw of the fischertechnik plug and insert the end of the cable into the sleeve. Now the screw is tightened again but not too much in order to avoid that the cable is squeezed off (fig. 3). You may perform the continuity check of the connection cables by means of the set-up of fig. 4, while the main cable should be tested using the diagnostic routine. Connect the lamp successively to all outputs M1 to M4 and switch on the output. The lamp should be lit. The digital inputs E1 through E8, on the other hand, are checked by connecting a switch between +5V and the respective input line. In a similar way and manner you proceed with the two analog inputs EX and EY with the exception that in this case you make use of the photoresistor. Direct the latter against a light source and the program will indicate low numerical values. Cover it with your hand and the numerical values will increase.
For wiring please strictly adhere to the scheme and the photo showing the ready wired model. If the cable cores are interchanged by mistake, the software will surely not operate correctly and in case of false lengths of the cables the free motion of the plotter might be restricted.
The acrylic glass plate serving as paper table has been marked with a positioning guide for a correct centering of the drawing paper. A paper of size DIN-A-4 (approx. letter size) on that plate will be covered over its whole area by the plotter. The paper may easily be held in place by some pieces of adhesive tape. A good advice to conclude with: as the base plate of the plotter is transparent, you may place the plotter also onto a daylight projector and plot live onto transparent foil by means of fibre pens.

Control of the plotter

In the preceding section we have explained how to control the stepping motor. The plotter is equipped with two stepping motors moving independently the x- and y-axis. It would be only natural to adapt the switching sequence, valid for one motor, to two motors. But this is easier said than done. With its two coil systems each motor occupies two outputs of the interface. Hence the drive of the plotter already fully utilizes the interface. And now the electromagnet of the writing pen has still to be added...
We can cope with this problem by trick: the first coil system of the two stepping motors each is connected at output M1. The second coil system of the x-motor is connected at output M2 and the second coil system of the y-motor at output M3, leaving output M4 free for connection of the writing pen. The disadvantage we have to accept for this free output: the stepping motors can no longer be controlled independently from each other. If e.g. a vertical line is to be drawn, i.e. only the y-motor is to be driven, it will be subjected to the following switching sequence:

	M1	M3	
1	CW	CW	
2	CCW	CW	
3	CCW	CCW	
4	CW	CCW	
5	CW	CW	etc.

At the same time the output M2 for the x-motor remains unaltered at pole direction clockwise. As, however, M1 is also connected at the x-motor, the x-motor is subjected to the following switching sequence during y-movement:

	M1	M2	
1	CW	CW	
2	CCW	CW	
3	CCW	CW	
4	CW	CW	
5	CW	CW	etc.

Kabelkonfektionierung · Ribbon cable configuration · Schéma de câblage

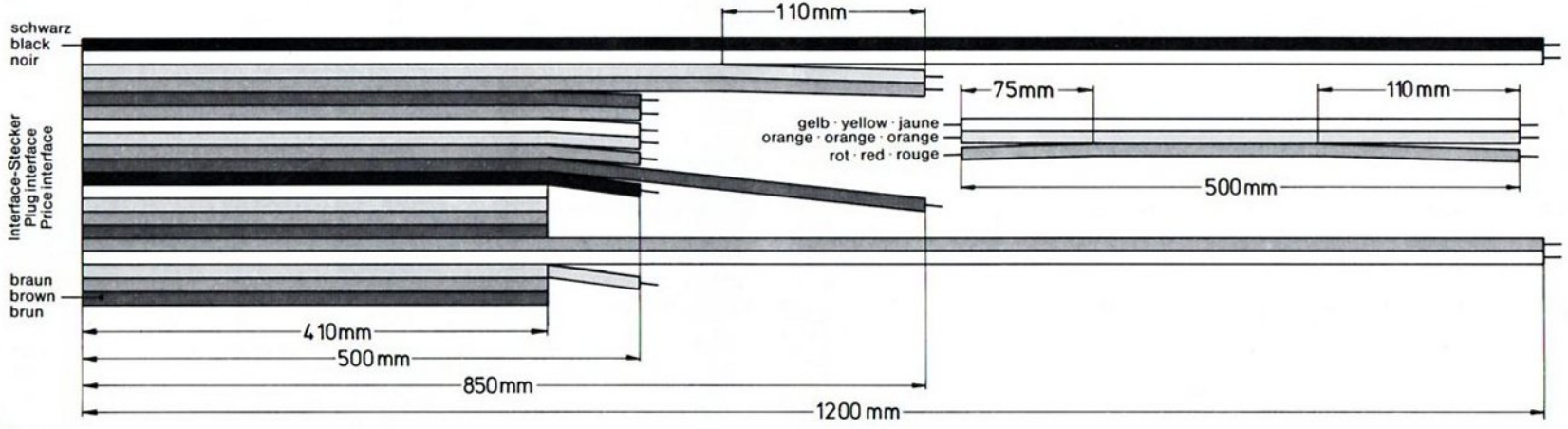


Fig. 2

Steckermontage
Plug installation
Assemblage de cables

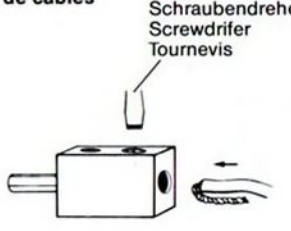
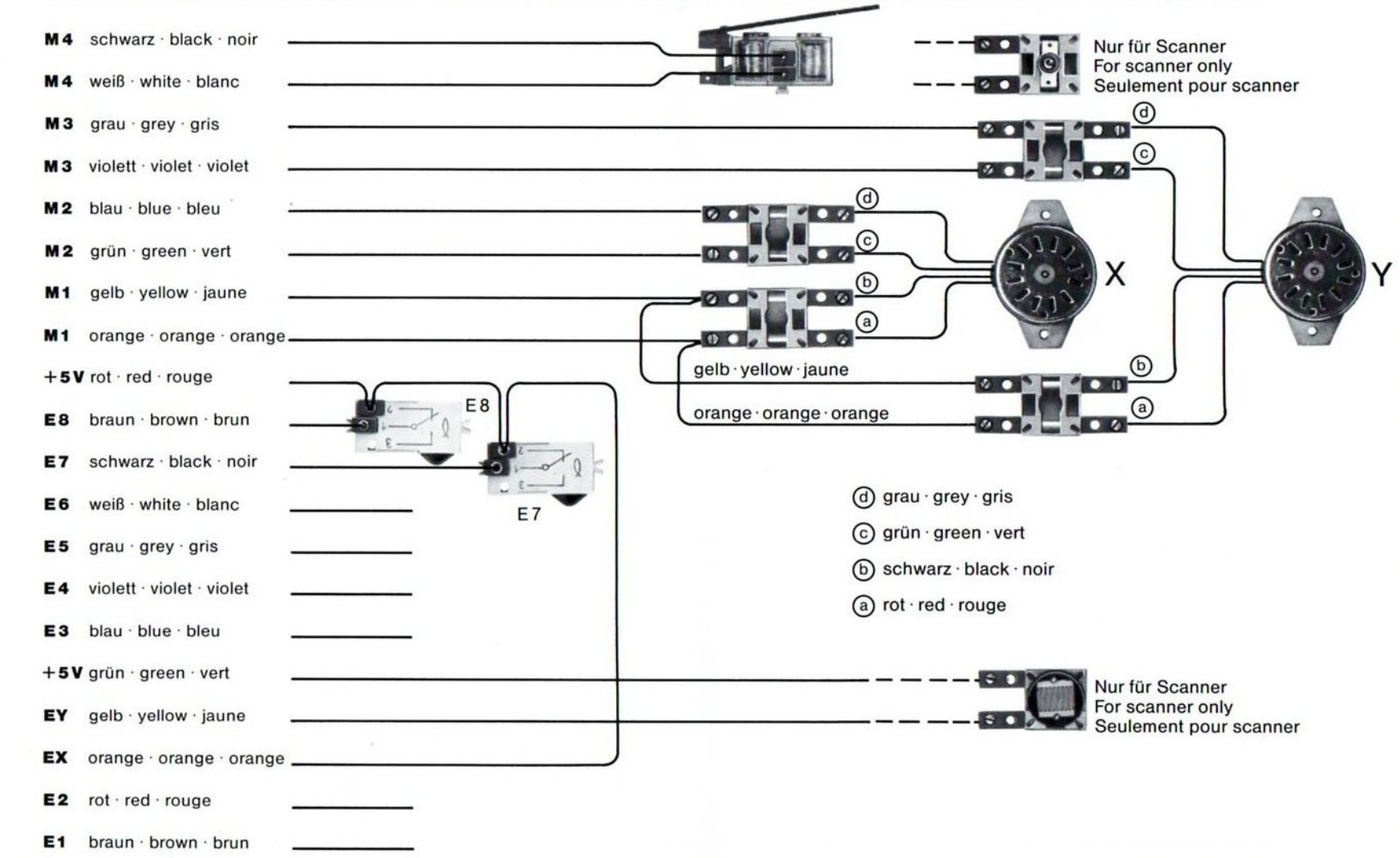


Fig. 3

Durchgangsprüfung
Continuity tester
Contrôle du passage

Fig. 4

Verdrahtungsplan Plotter/Scanner · Circuit layout Plotter/Scanner · Plan de câblage de la table trâcante/du scanner



When passing from 1 to 2, the x-motor will perform a step. Idem, it will perform a step when passing from 3 to 4, but in the opposed direction. You may convince yourself of this fact by looking at fig. 1a through 1d. On the whole, the "uninterested" motor will perform one step only (by moving to and fro). Hereby, however, the accuracy of the drawing is not impaired as one step alone does not yet lead to a significant advance but is compensated for by the required allowance of the plotter. We only should take care in programming that we do not accumulate those steps. Therefore the programs are written in a way that they call in only multiples of four consecutive steps which we will call plotter steps. What is the size of one plotter step or rather which resolution may be achieved with the plotter? Twelve plotter steps effect one revolution of the plotter spindle and hence a displacement of 1.5 mm * π. Consequently one plotter step is equivalent to a distance of 0.3927 mm or approx. 0.015".
On such basis, the subroutines have been written controlling the motors according to the direction of the axes. This leads to a total of four cases:

positive x-direction (to the right)
subroutine from line 50.000 onwards
negative x-direction (to the left)
subroutine from line 51.000 onwards
positive y-direction (upwards)
subroutine from line 52.000 onwards
negative y-direction (downwards)
subroutine from line 53.000 onwards

Apart from the movements in axial directions, the plotter may also draw diagonals. In such case, the x- and y-motor will operate simultaneously. Again, we can differ between four cases, depending on the direction of movement of the two motors:

positive x- and positive y-direction
(to the right and upwards)
subroutine from line 54.000 onwards
negative x- and positive y-direction
(to the left and upwards)
subroutine from line 56.000 onwards
negative x- and negative y-direction
(to the left and downwards)
subroutine from line 57.000 onwards
positive x- and negative y-direction
(to the right and downwards)
subroutine from line 55.000 onwards

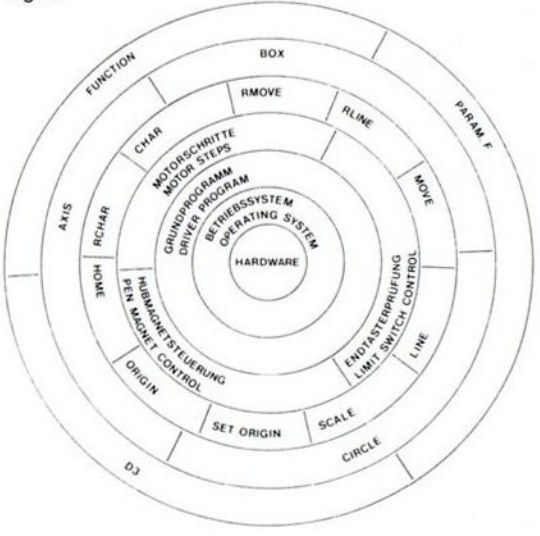
Any and all more complicated movements can be derived from these eight basic movements. It exists e.g. a subroutine for drawing straight lines between any optional points on the drawing paper. In such case, it has to be determined first of all which of the motors has to travel the longer distance. This motor operates permanently. The motor with the shorter distance is controlled to effect either no or a diagonal step as per requirements. The steps are distributed so that the drawn line is never too much away from the ideal line of connection (cf. fig. 5). For this purpose, the gradient of the distance is computed. At each step of the motor permanently operating the value of the gradient is summed up. As soon as the value 1 is exceeded the counter is reset and a diagonal impulse is generated.

Fig. 5

Plotter software

In order not to lose track of all plotter subroutines, we shall classify them according to a scheme. For this purpose we avail ourselves of the shell structure frequently used in the development of software in which one subroutine shell is placed around the next one like the skins of an onion, as shown in fig. 6 below. The internal core is formed by the hardware, i.e. the computer with screen and floppy disk or cassette station, the interface and the plotter. This core is surrounded by shell No. 1 being the operating system of your computer, i.e. that program enabling your computer to understand BASIC. Shell No. 2 holds the driver routine mentioned already above with generation of all the interface commands. The subsequent shell No. 3 includes the control of the stepping motors and of the writing pen magnet as

Fig. 6



well as the interrogation of the limit keys. The tasks of shell No. 4 are even more complex. It includes the above mentioned program for drawing a straight line; moreover, it includes quite a series of further useful subroutines which we shall discuss below. Now we come to shell No. 5. These subroutines include the so-called plotter tools. In software engineering such programs are defined as tools which are performing comprehensive and frequently repeated jobs. In this case, it concerns the drawing of coordinate axes, rectangulars and circles. Also this item will be discussed later on in this text.
The ultimate shell represents the user program i.e. you are writing. To assist you in the beginning, we have provided you with three examples of user programs. The essence of this shell structure is that subroutine calls should always be effected from the outside to the inside only. On basis of this knowledge, you can specifically remove those shells you do not need when developing own programs and can supplement other subroutines.
Back to shell No. 4: As routines of shell No. 4 the following subroutines are defined in program PLOT:

HOME

Parameter: none
Subroutine entry line: 40000

Function: Moves the plotter into the physical origin of coordinates, in which case the writing pen is lifted. The two stepping motors are actuated until the two limit keys react. Hereafter, the plotter will run back to the drawing area until the limit keys are no longer actuated. Add to this two extra steps so that now a safe distance from stop is ensured. Finally a series of flags are set to zero and, amongst others, zero of the coordinate system is stipulated at its present position.
This command is typically the first command before all other plotter commands. It is recommended also as last command as by this command the writing pen is lifted, the plotting head moved to the outer edge and the paper released.

LINE

Parameter: X and Y
Subroutine entry line: 44000
Function: The program draws a line from the present position of the plotter to the point identified by X and Y. The writing pen is lowered for drawing.

RLINE

Parameter: X and Y
Subroutine entry line: 46000
Function: The program draws a line from the present position of the plotter to the point displaced by X and Y. The difference, as compared to LINE, is that not the actual target is indicated but only the distance to reach the latter.

MOVE

Parameter: X and Y
Subroutine entry line: 45000
Function: MOVE moves the writing pen towards the target (X,Y) without drawing, i.e. the writing pen is lifted. Otherwise, MOVE corresponds to the subroutine LINE.

RMOVE

Parameter: X and Y
Subroutine entry line: 47000
Function: As MOVE corresponds to LINE, RMOVE corresponds to RLINE. I.e. no drawing is effected and the indication of coordinates is relative by indication of the value of the distance.

SET ORIGIN

Parameter: X and Y
Subroutine entry line: 42000
Function: From now on, the zero point of the coordinate system is shifted to point (X,Y). The origin of the coordinates is indicated referring to the hitherto valid coordinate system. Before the first call of subroutine SET ORIGIN, the coordinate system coincides with the mechanical system of the plotter, hence it is situated in the left bottom corner near the limit keys.

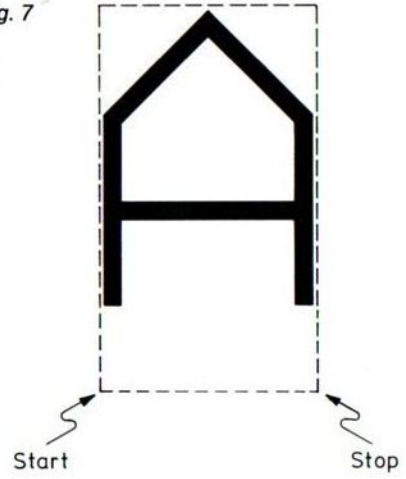
ORIGIN

Parameter: none
Subroutine entry line: 41000
Function: Raises the writing pen and moves the plotter to the origin of the coordinate system. The latter may not necessarily coincide with the left bottom corner of the plotter but may also have been shifted to another position by means of the above described command SET ORIGIN.

CHAR

Parameter: X, Y, A$, R and G
Subroutine entry line: 48000
Function: By this subroutine, the letters which have been filed in the variable A$, are drawn at the position indicated by X and Y. More precisely: You have to imagine the letter surrounded by a frame (fig. 7). The subroutine assumes that the pen is situated at the left side of the letter frame as indicated by the position start. When the letter has been finished, the writing pen is situated at the right side and lifted off. The available letters of the plotter are represented in fig. 8. Parameter R defines the direction of writing of the plotter. When lettering graphs/diagrams the requirement may occur to produce vertically arranged texts or texts written upside down. Fig. 9 shows the letters on basis of various values for R. By G the size of the character is determined. For G, any optional integer value can be chosen, as long as the drawing area is not exceeded. Fig. 9 also shows the effect of parameter G.

Fig. 7



RCHAR

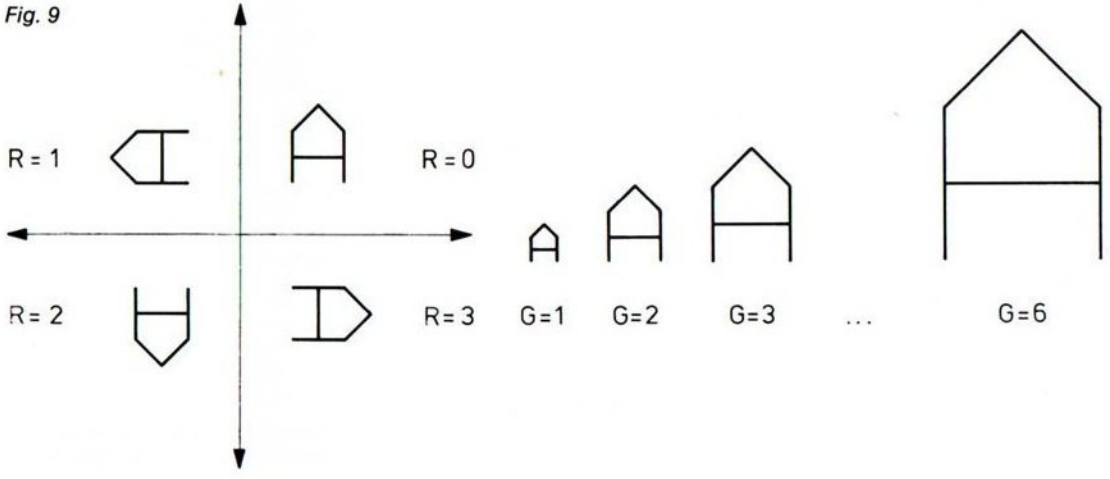
Parameter: X, Y, A$, R and G
Subroutine entry line: 49000
Function: This subroutine performs the same functions as CHAR. As already in case of the above mentioned subroutines, the positioning is differing in that only the distance to the present position in X and Y is transmitted. Especially in case of continuous texts RCHAR is of interest. As after drawing of the first letter the plotting pen is located at the right side of the letter frame, the program RCHAR called with X=0 and Y=0 may add directly the next letter.

SCALE

Parameter: X1, X2, Y1, Y2
Subroutine entry line: 43000
Function: You may not always want to compute for all commands the number of the steps of the stepping motors as coordinate. By means of the subroutine SCALE you may provide the left border of the paper with value X1, the right border with value X2. Correspondingly you apply Y1 for the bottom and Y2 for the top side. All subsequent information of coordinates refer to this new coordinate system.

Fig. 8

Fig. 9



Plotter tools

When loading the program PLOT a further three subroutines are loaded apart from the subroutines discussed above.

AXIS

Parameter: XA, YA, XE, YE, X1, Y1, X2, Y2, SK, X$ and Y$
Subroutine entry line: 22000
Function: The program AXIS serves for drawing the system of the two coordinates. This subroutine will be of particular use if you want to draw functions, measured values and results or distributions. It includes quite a series of parameters by means of which you can create the system of the two coordinates. The positioning of the coordinate system is defined by the four variables XA, XE, YA, YE. XA defines the beginning of the x-axis, XE the end of same. Correspondingly YA and YE define the beginning and the end of the y-axis. XA, XE and YA, YE are indicated in plotter steps referring to the left bottom corner of the plotter. Similar to the suroutine SCALE, however, the subroutine AXIS applies to these plotter steps the actual values of your coordinate system. X1 represents the value at XA, X2 at XE, Y1 at YA and Y2 at YE. In this way and manner, coordinate axes of any optional dimension whatsoever may be drawn with any optional ranges of values. If the parameter SK should differ from 0, both axes are lettered. The subroutine will try to design the coordinate system on basis of values of experience. If you should prefer a deviating design, you may intervene on your own by refering to the documentation of the programs. Additionally, a short legend may be indicated for each axis which is filed under the character string variables X$ and Y$. Size and direction of lettering are defined by subroutine AXIS but may, of course, be altered as per requirements.

CIRCLE

Parameter: X, Y, RX, RY, WA and WE
Subroutine entry line: 21.000
Function: The second tool is a subroutine for drawing circles. However, not only circles may be drawn but – as a generalization of the circle – also ellipses. This is controlled by the indication of separate radii in x- and y-direction, RX and RY. Moreover, not the whole circle or the whole ellipse has to be drawn. By WA and WE the angle at the beginning and at the end of the circular resp. elliptical arc is indicated. The counting of the angle is effected in degrees in counter clockwise direction, beginning at the positive x-axis.

BOX

Parameter: XA, XE, YA, YE and S
Subroutine entry line: 20.000
Function: By this subroutine, a rectangle is drawn with XA, YA drawing the left bottom corner point, XE, YE the right top corner point. If S differs from zero, the rectangular area is hatched diagonally. The absolute value of S (i.e. omitting the prefix) indicates the spreading of the hatching. The sign controls whether the hatching is to be performed from bottom left to top right (in case of positive value) resp. from top left to bottom right (in case of negative value).

Use of the plotter routines

When you have loaded the program PLOT, all above mentioned subroutines are at your disposition. However, after starting the program PLOT it does nothing else than positioning the plotter in its home position and loading the character set. Nothing more? Indeed, your part is now to come. It is your task to prescribe what should be plotted. For this purpose you may write a program in the range of line number 1000 to 19000.

Have a close look to this range by giving the command:

```
LIST 1000–19000
```

You should, depending on the computer system, read something like:

```
1000 REM
1010 REM ****************************
1020 REM START OF THE USER PROGRAM
1030 REM ****************************
19000 END
```

Let us get acquainted with the most simple commands, LINE and MOVE, in first instance. LINE plots a line from the actual position of the pen to the position given by X and Y. Insert the following lines in program PLOT:

```
1060 LET X=200 : Y=150
1070 GOSUB 44000 :REM LINE
```

When starting the program you will observe a straight line reaching from the starting position (X=0, Y=0) to the position (X=200, Y=150). The drawing may be continued from that position. Add e.g.

```
1080 LET X=150 : Y=50
1090 GOSUB 44000 :REM LINE
```

In general you will not be able to plot a drawing at one stroke. For movements with the pen lifted you will use the subroutine MOVE. When adding:

```
1040 LET X=50 : Y=200
1050 GOSUB 45000 : REM MOVE
```

the path will no longer start in the left lower corner of the paper.
What is the permissible range of X and Y? Just test it yourself! Delete line number 1040 through 1090 (or just over-write them) and type in the following tiny program

```
1040 FOR T=10 TO 700 STEP 10
1050 LET X=T : Y=T
1060 GOSUB 44000 :REM LINE
1070 PRINT "LINE TO X=";XJ;" Y=";YJ
1080 PRINT " XOUT=";XOUT;
     "YOUT=";YOUT
1090 NEXT T
1100 LET X=400 : Y=200
1110 GOSUB 44000 :REM LINE
```

Observe the plotter as well as the printout on the screen. The plotter will in first instance follow a line inclined by 45 degrees since X and Y are incremented by same steps. When reaching the value 500 the pen is located at the upper edge of the paper. From now on the plotter will proceed to the right. The pen is lifted although the command LINE is given. When superceeding the value 680, the plotter will even come to a complete stop. So, exceeding the plot area ($0<X<680$, $0<Y<500$) will not lead to any damage or misalignement of the plotter. The plotter tries to follow the prescribed movement by following the border of the paper. You may observe from the screen printout, that the values of the internal coordinates (XJ and YJ for X- and Y-direction, respectively) continue to be defined. Once the target position falls within the plot area, the plotter activates the pen again. You also might have observed the relation between the flags XOUT and YOUT and the out-of-range situation. By testing those flags after a movement of the plotter you might easily interrogate if the plotter has left its admissible range.

In the following interesting example we will make use of the flags as well as introduce relative movements (RLINE and RMOVE). Delete lines 1040 through 1110 or just load once again the program PLOT.

In an extremely simplified simulation experiment we will plot the path of a molecule in gas. The molecule collides with the molecules surrounding it and therefore obtains from time to time a new momentum and direction. The new direction and range is subject to statistical distribution. In total we observe a zig-zag path (Brownian movement). Starting point of the path is the middle of the paper. Once the path leaves the paper, the program stops.

```
1040 REM
1050 REM SIMULATION OF MOLECULAR
     MOVEMENT
1060 LET X=340 : Y=250 : REM MIDDLE OF
     THE PAPER
1070 GOSUB 45000 :REM MOVE
1080 LET SC=0 :REM STEP COUNTER
1090 LET A=100 :REM MAXIMUM RANGE
1100 LET SC=SC+1 :REM COUNT STEPS
1110 LET X=A*(RND(1)-0.5)
1120 LET Y=A*(RND(1)-0.5)
1130 GOSUB 46000 :REM RLINE
1140 PRINT "POSITION ";SC" AT X,Y: ";XJ,YJ
1150 IF XOUT THEN GOTO 1180
1160 IF YOUT THEN GOTO 1180
1170 GOTO 1100 :REM NEXT STEP
1180 PRINT "PATH STOPPED"
1220 GOSUB 40000 :REM HOME
```

The programming example demonstrates, how the subroutine RLINE appends path elements starting from the position reached so far. If you require the absolute position, however, you still may obtain it by means of the variables XJ and YJ (see line 1140).

Do you prefer lettering of your drawing? Just add the following lines to your program:

```
1190 LET X=0 : Y=0 : R=0 : G=2
1200 LET A$="MOLECULAR PATH"
1210 GOSUB 48000 :REM CHAR
```

When finishing the plot, the text "MOLECULAR PATH" will be written to the lower left corner of the paper. If you prefer a lettering using upper and lower case characters, you may write line 1200 as well

```
1200 LET A$="Molecular path"
```

This holds for computers which strictly use the ASCII standard character set. As some computers deviate, we give the ASCII standard in the table below. By the way, if you want to look up the definition of a specific character in the range of DATA lines, you just add 62000 to the ASCII code number to obtain the line number of the DATA statement.

Unfortunately, just the popular Commodore computers, frequently used in the documentation of the fischertechnik programs, uses a character set strongly deviating from ASCII. As long as you operate the computer in the capital letter/graphics mode, the capital letters, numbers and punctuation marks coincide with the plotter's character set. If you switch to the lower case/upper case mode by pressing the SHIFT- and Commodore keys at the same time, all upper case letters have been converted to lower case letters on the screen. This magic, however, applies not for the character set of the plotter. Nevertheless you may adopt the Commodore character set for your plotter. You just have to reorganize the sequence of DATA lines in the definition of the character set. Lines 62064 through 62095 go to lines 62192 through 62223. The DATA lines being previously there are just overwritten. The lines 62096 through 62127 move to the range 62064 through 62095. Finally you may fill up the range 62096 through 62127 with undefined characters (ten times the zero). However leaving the characters unchanged would do as well. Finally you should con-

ASCII	DEZ	HEX	ASCII	DEZ	HEX	ASCII	DEZ	HEX	ASCII	DEZ	HEX
NUL	00	00	SP	32	20	•②	64	40	•⑦	96	60
SOH	01	01	!	33	21	A	65	41	a	97	61
STX	02	02	"	34	22	B	66	42	b	98	62
ETX	03	03	•⓪	35	23	C	67	43	c	99	63
EOT	04	04	•①	36	24	D	68	44	d	100	64
ENQ	05	05	%	37	25	E	69	45	e	101	65
ACK	06	06	&	38	26	F	70	46	f	102	66
BEL	07	07	'	39	27	G	71	47	g	103	67
BS	08	08	(	40	28	H	72	48	h	104	68
HT	09	09	)	41	29	I	73	49	i	105	69
LF	10	0A	*	42	2A	J	74	4A	j	106	6A
VT	11	0B	+	43	2B	K	75	4B	k	107	6B
FF	12	0C	,	44	2C	L	76	4C	l	108	6C
CR	13	0D	-	45	2D	M	77	4D	m	109	6D
SO	14	0E	.	46	2E	N	78	4E	n	110	6E
SI	15	0F	/	47	2F	O	79	4F	o	111	6F
DLE	16	10	0	48	30	P	80	50	p	112	70
DC1	17	11	1	49	31	Q	81	51	q	113	71
DC2	18	12	2	50	32	R	82	52	r	114	72
DC3	19	13	3	51	33	S	83	53	s	115	73
DC4	20	14	4	52	34	T	84	54	t	116	74
NAK	21	15	5	53	35	U	85	55	u	117	75
SYN	22	16	6	54	36	V	86	56	v	118	76
ETB	23	17	7	55	37	W	87	57	w	119	77
CAN	24	18	8	56	38	X	88	58	x	120	78
EM	25	19	9	57	39	Y	89	59	y	121	79
SUB	26	1A	:	58	3A	Z	90	5A	z	122	7A
ESC	27	1B	;	59	3B	•③	91	5B	•⑧	123	7B
FS	28	1C	<	60	3C	•④	92	5C	•⑨	124	7C
GS	29	1D	=	61	3D	•⑤	93	5D	•⑩	125	7D
RS	30	1E	>	62	3E	•⑥	94	5E	•⑪	126	7E
US	31	1F	?	63	3F	—	95	5F	DEL	127	7F

•	⓪	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪
Hexadezimal	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
English (USA)	#	$	@	[	\	]	^	`	{	|	}	~
Deutsch	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß

vince yourself that you still have the same total number of DATA lines.
The whole process should be done using a programming toolkit, that allows renumbering including block transfer. It may be done using the fischertechnik character generator as well. When using a C64, you call the latter by

GOTO 63000

Here are some more tricks concerning the use of the character set. In some cases not all characters are accessible from the keyboard. You may e.g. meet difficulties to generate the vertical bar (code number 124). In that case you have to merge this symbol into the string using the CHR$-function:

LET A$="coordinates (X"+CHR$(124)+"Y)"

To print out numbers you may make use of the STR$-function. Let's stick to our example and modify line 1200 once more:

```
1200 LET A$="Molecular Path in"
     +STR$(SC)+" Steps"
```

You may create special effects by superimposing text strings. Either you chose the appropriate text coordinates for the strings or you chose the ASCII control code BACKSPACE (code number 8) which moves the plotter pen back by one character frame. Using the string

A$="0"+CHR$(8)+"/"

generates a zero with a slash. This is mentioned just to demonstrate the method. If you really prefer the zero with slash, you should modify the character set permanently. The control code 8 is, by the way, the only ASCII control code evaluated by the plotter.

Once again a hint for users of Commodore Computers. If you prefer to use the CURSOR LEFT symbol instead of BACKSPACE you may modify line number 48070 in the program PLOT:

```
48070 IF MID$(A$, I0,1)=CHR$(157)
      THEN XNEW%=-5 : YNEW%=3 :
      PEN=0 : GOTO 48220
```

The next example is devoted to the subroutines SET ORIGIN, ORIGIN and SCALE. Those subroutines serve your ease of programming. In most cases the range of the values to be plotted will not coincide with the number of steps to control the motors. Let's assume you want to plot a temperature curve. The span of the temperature will range from –20 to 40 degree Celsius and will be plotted along the y-axis. The x-axis will arranged to contain a time scale from 0 to 60 minutes. The type of the task is very typical and it may easily occur that you sit down every time, looking up formulas and running the risc of introducing bugs in the program. There is, however, no need to do so, since you just give this task to subroutine SCALE. The example below demonstrates the use of subroutine SCALE. It does not really plot a temperature curve but just a frame around the area to contain the curve. Of course you may improve and extend the program later on. We will also present suggestions to simplify the program in the ongoing text.

```
1040 REM
1050 REM FRAME
1060 LET X1=0 : X2=60 :REM TIME SPAN
1070 LET Y1=-20 : Y2=40 :REM
     TEMPERATURE SPAN
1080 GOSUB 43000 :REM SCALE
1090 GOSUB 1170 : REM PLOT FRAME
1100 GOSUB 41000 : REM ORIGIN
1140 END
1150 REM
1160 PLOT FRAME
1170 LET X=0 : Y=-20
1180 GOSUB 45000 : REM MOVE
1190 LET X=0 : Y=40
1200 GOSUB 44000 : REM LINE
1210 LET X=60 : Y=40
1220 GOSUB 44000 : REM LINE
1230 LET X=60 : Y=-20
1240 GOSUB 44000 : REM LINE
1250 LET X=0 : Y=-20
1260 GOSUB 44000 : REM LINE
1270 RETURN
```

In line 1100 the plotter moved to its origin, which means X=0, Y=0. Note that in the case chosen this is not the lower left corner of the paper. The origin of the system of coordinates may be displaced to any position. The subroutine SET ORIGIN serves for this task. Since the conversion factors are not affected by displacing the origin, the lower and upper boundaries are displaced by the same measure. We will study the effect of the subroutine SET ORIGIN by adding three lines:

```
1110 LET X=30 : Y=10
1120 GOSUB 42000 : REM SET ORIGIN
1130 GOSUB 1170 : REM PLOT FRAME
```

Run the program once more and compare with the previous result. The first frame is plotted as before. After the pen is moved to the origin, the latter is displaced. When plotting the frame once more you will recognize the displacement. In the example chosen, the displacement leads to an out-of-range situation at the upper and right border of the frame, which you will recognize from the lifting of the pen. You may imagine the effect of the subroutine SET ORIGIN like moving a window across the plotter area.

As mentioned above we can simplify the program. Instead of the subroutine plotting the frame, you may use the plotter tool BOX as well. Replace lines 1170 through 1270 by the following:

```
1170 LET XA=0 : ZA=-20 : XE=60 :
     YE=40 : S=0
1180 GOSUB 20000 :REM BOX
1190 RETURN
```

Once you have become acquainted with the control parameters of SCALE and BOX, you will find it easy to use the plotter tool AXIS. Let's stick to the example already chosen, the temperature curve with a temperature span from –20 to 40 degree Celsius and the time interval of 60 minutes. The following program lines define the system of coordinates for this case.

```
1040 REM
1050 REM SYSTEM OF COORDINATES FOR
     TEMPERATUR CURVE
1060 LET XA=30 : YA = 0 : XE=680 :
     YE=500
1070 REM USES ALMOST THE WHOLE
     PAPER SIZE
1080 LET X1=0 : Y1=-20 : X2=60 : Y2=40
1090 REM DEFINES THE RANGE OF
     VALUES
1100 SK=1 :REM LETTERING OF THE AXES
1110 X$="time (min)" :REM TEXT X-AXIS
1120 Y$="temperature ("+CHR$(127)+"C)"
1130 GOSUB 22000 : REM AXIS
```

In line 1120 the text of the lettering has been supplemented by a special character. The symbol denotes the degree character, the superscript zero. Since this character is not yet defined in the program PLOT, it has to be generated by the character generator program. A corresponding run yielded the DATA line replacing the previous one:

```
62127 DATA 1657,5869,8998,9786,6600,0,0,0,0,0
```

The subroutine AXIS defines the scale factors and displacements, very similar to SCALE, depending on the given range of plotter steps and the corresponding spans of the values. You may convince yourself by plotting a diagonal line, the end-points of which are given in the scaled system.

```
1140 LET X=0 : Y=-20
1150 GOSUB 45000 : REM MOVE
1160 LET X=60 : Y=40
1170 GOSUB 44000 : REM LINE
```

The subroutine AXIS also simplifies the task to use just a part of the paper format. If you would like to restrict the plot to the left half of the paper you just have to modify line 1060 (and 1070):

```
1060 LET XA=30 : YA=0 : XE=340 : YE=500
1070 REM USES HALF OF THE PAPER
```

Now some hints for the use of the subroutine CIRCLE. As described above the subroutine may not only be used to plot circles or ellipses, but also arcs of circles and ellipses. The arc is defined by its starting and ending angle. These angular parameters WA and WE have to be given in degrees. A lot of computers, however, do not use degrees as angular measure but only radians. In that case you may use one of the following conversions:

```
WDEGREE=WRAD*45/ATN(1)
WDEGREE=WRAD*180/PI :REM IF PI
IS DEFINED
```

Alternatively you may modify the subroutine CIRCLE by eliminating the factor ATN (1)/45 in lines 21040, 21050, 21090, 21100, 21130 and 21140, e.g.:

```
21040 LET X=XA+RX*COS(WA)
```

Very similar to the examples above you may write more complex programs. Three such programs are on the floppy disk resp. cassette. They are meant to familiarize you with the use of the plotter and the plotter subroutines. Hence, these programs may also serve as examples for your own program developments. Two of the programs serve for the recording of functions, the third one shows the projection of a three-dimensional object onto the paper.

Program FUNCTION

By means of the program FUNCTION you may depict on the plotter any optional function of type Y=F(X). The only thing you have to do in this case is to define the function. For this purpose, line 1000 has to be modified accordingly. In our example, the line reads as follows:

```
1000 DEF FNY(X)=SIN(X)*EXP(-0.02*X)
```

Hereby, the plot of a damped sinusoidal oscillation is produced as shown in fig. 10. For technical reasons, the name of the function must be input again as character string for the lettering of the plot:

```
1010 F$="SIN(X)*EXP(-0.02*X)"
```

Besides, the value range of the independent variable X has to be indicated in line 1020. Our example reads:

```
1020 XA=0 : XE=90
```

You may alter all three lines according to your requirements. The rest will be done by the program for you. It will compute minimum and maximum of the function so that the paper is optimally utilized. It creates the coordinate system and then draws the function.
Depending on the actual computer the running time of the program may range between 20 and 80 minutes. Therefore, in case of repeated runs, we recommend to compile the program after the definition of the function, to make use of the full velocity of the plotter.

Program PARAM.F

Many charts cannot be written as a function Y=F(X). Remember e.g. the circle (for which we have a special subroutine, too). In the above notation, its functional equation would read:

```
1000 DEF FNY(X)=SQR(R*R-X*X)
```

Hereby, however, only the upper semicircle would be drawn. Moreover, error messages would occur if the drawing area should be chosen larger than the radius R. It is better to describe such functions as twodimensional functions of a common parameter T. The program PARAM.F has been designed for this purpose. Analogously, the function is defined again with the command DEF:

```
1000 DEF FNX(T)=COS(T)
1010 DEF FNY(T)=SIN(T)
1020 F$="CIRCLE"
1030 TA=0:TE=6.2832:TS=0.1
1040 AX=1:REM DRAW COORDINATE
     SYSTEM
1050 XA=1.1:XE=1.1:YA=1.1:YE=1.1
```

The above example shows how to draw the circle without difficulties by means of the program PARAM.F. Lines 1000 to 1030 are to be used like the corresponding lines of the program FUNCTION. Lines 1040 and 1050 should still be explained. By the value of the variable AX it is determined whether a coordinate system is to be drawn. In the above case, it is drawn; if AX = 0, it would be suppressed. The program PARAM.F does not perform a scaling of the range of values in order to avoid undesireable distortions of the graphs. Therefore, line 1050 indicates the limit values in both directions of axis. If the functional value should exceed these limits, the

writing pen will move along the border. Let's arouse your enthusiasm for the 'beauty' of some of the following functions.

Lissajous'figure

```
1000 DEF FNX(T)=SIN(F1*T+P1)
1010 DEF FNY(T)=SIN(F2*T)
1020 LET TA=0:TE=8*ATN(1):TS=0.02
1030 LET XA=-1.3:XE=1.3:YA=-1.3:YE=1.3
1040 LET AX=0:REM NO COORDINATE
     SYSTEM
1050 LET F1=2:F2=3:P1=ATN(1)
```

Lissajous' figure 2
idem, however,

```
1050 LET F1=5:F2=7:P1=2*ATN(1)
```

Program D3

Some remarks to the use of the program D3 are in order. Starting the program you will recognize that it plots a three-dimensional object. This is, however, not the only object to be plotted. The program has been constructed in a way to read the information about the object from a couple of DATA lines. By modifying the DATA lines you may plot any object which is made up of straight edges. The angle of observation is later on defined, when running the program, which means nothing else that you may turn and tilt the object as you like.

How to convert the construction of the object in an array of DATA lines? You start with a three-dimensional system of coordinates, in which you place the object. Take e.g. x for the width, y for the depth and z for the height of the object. The range of values you use, is arbitrary, since the program D3 scales the drawing to make optimal use of the paper size. Write down the corners of the object in triples of the values, x, y, and z. Bring those corners in a sequence so that you may plot later on the object uninterruptedly if possible. A corner of the object will appear several times in the list if there are more than two edges merging into it.
Once you have arranged the corners in the most time effective sequence, you have to supplement each line in the list by a 1 (one) or 0 (zero) at fourth position. When encountering a 1, the plotter will connect the previous corner with the actual one. In case of a 0 the actual position will be reached with the pen lifted.
The list of numbers generated in the described manner is typed into the program as DATA lines. The list is closed by the special DATA-line with the constants 0,0,0,–1. The DATA-lines are supposed to be located subsequent to the character set of the program.

Character generator

The characters used in the subroutines CHAR resp. RCHAR are coded in a larger block of DATA-lines. In order to enable you to modify the characters or to define new characters (ASCII-codes No. 128 through 255 are not yet occupied!) a character generator program is part of the software. Depending on the type of computer concerned it is either integrated in the program PLOT or separately loadable. The structure of the program strongly depends on the graphics system of the computer so that a universally applicable description cannot be given. The operation of the program is, therefore, self explanatory.

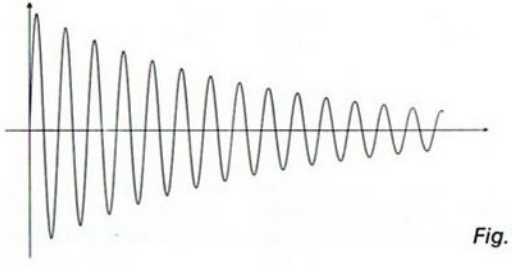


Fig. 10

Scanner operation

The modular concept of fischertechnik allows a simple conversion of the plotter to obtain a scanner. If in case of the plotter the output of the data available in the computer is realized on paper, the direction of the data in case of the scanner is reversed. The original inserted in the scanner is scanned by a recording head. Amongst others, this head includes a lighting facility in order to ensure a uniform illumination of the original beneath the read head.
Furthermore, it is equipped with a photoresistor. A photoresistor will alter its resistance according to the intensity of its illumination. The brighter it is illuminated, the lower the resistance value. The photoresistor is screened in a way that it is not directly radiated by the lighting. Only the light reflected by the original is registered by the photoresistor. The quantity of light, however, depends on the gray shading of the original. Thus it is possible to 'read' the original.

The resistance value of the photoresistor is determined by means of the initially described analog input EY. For initial testing you may load the diagnostic routine from the floppy disk or cassette, already described above. Now start the program. Depending on the type of interface concerned you most probably will note an overflow of EY at 255 or a number higher than 255. Now the time has come to switch on the light! Hence, select M4 (by actuation of the key 4) and switch on the output as indicated by the programs menu. Now the lamp is illuminated and immediately EY will indicate lower values. Insert a piece of white paper beneath the read head and take a note of value EY. Now colour a larger area by application of a black fibre pen or use a piece of black cardboard. You should take down this value, too. Both values will define the dark-bright-range. The gray shades should be between the two extreme values. You may perform a test to this effect. Unfortunately, we cannot give any exact details regarding the extent of values because it depends on many

factors: the respective type of interface concerned, the scatter of the lamp and photoresistor, the brightness of the paper and the environmental light. At any rate, the value read-in should vary between 0 and 255.
For the next experiment you will draw some black lines of different widths onto white paper. Pass these lines or strokes transversally through beneath the recording head. You will observe that only in case of lines wider than 5 mm you will approximate the previously registered black level with the head positioned exactly above recording the line. The finer the line, the more the value read-in will approximate the white level. In case of very fine lines, the value of EY will not alter at all when passing the line underneath the read head. In any case, however, it should be possible to safely recognize lines of a width of 2 mm.

Digital image recording

Perhaps, you will have already envisioned systems in which images can be recorded by means of a video camera, transferred to the computer and evaluated there. Well, such an installation will have its price. If you spend sufficient time you may achieve the same result by means of the scanner.The program SCANNER serves for scanning an original picture or image. It is scanned line by line. You may select the format of the image. Now the image is recorded. During the period the scanner is operating, the lighting conditions of the environment should not change. The best would be to darken the room a bit. When the data of the image are available in the main storage of the computer you may experiment with them to your heart's content. For repeated evaluation, the data first of all are filed on floppy disk or cassette.

Image evaluation

For the following evaluation of images we need a graphic system for our home resp. personal computer. New types of models are already equipped accordingly. For previous types additional equipment will normally be available on basis of which screen graphics can be used. But there are also some computers of latest techniques which incorporate the necessary hardware for screen graphics but do not support it by the respective software. In this case you may aquire an extended software and install it before loading the following programs. For the Commodore 64 Computer we have used the BASIC-extension SIMONS BASIC which may be obtained from Commodore, in the following program documentation. If you should want to adapt the programs to other computers or other software systems you should pay special attention to all program lines marked by an asterisk. As we are very sparing of graphic commands it will not entail significant difficulties to find out equivalent commands and to modify the programs.

Let us now tackle our first image evaluation by means of the program B&W. We intend to perform a gray shade separation. That means that all elements of the data field are represented by a bright screen point on the condition that their value does not exceed a predetermined limit value. If the value is greater than or equal to the limit, it is represented as a dark screen point. You may input the limit value by the computer keyboard. Perform a test with various values. Please also determine the limits of the value range. They have been obtained if the whole image is either white or dark.
Also the image resolution may be altered subsequently; it naturally can only be coarser. The program D.PIC is a further development on basis of the previous one. You are asked for the limit value and the resolution. Digitize a portrait for instance. Have the portrait displayed on the screen with the suitable threshold. Now reduce bit by bit the resolution and observe when the person on the screen can no longer be recognized. Digitizing photos you will obtain in this way and manner fascinating alienations of images which may also be used as an artistic means of design and creation. For most of the computers, so-called hardcopy programs have already been published in computer journals. These are programs which will copy the momentaneous screen content distortionless on the printer. You may utilize such programs for recording your creations on paper. But not only art, also engineering and medicine benefit from the digital processing of images. Remember e.g. X-ray pictures. If not bones but organs inside the body are concerned, the conventional X-ray photography suffers from a deficient contrast. This deficiency is coped with by a digital processing of pictures. By means of the computer not only one grey shade can be stipulated as limit beween black and white but it is possible to determine several limit values. Consequently, the range of grey shades is separated in intervals. To each interval, a screen colour is now assigned optionally. What previously have been undistinguishable grey tones, now turn to be clearly distinguishable colours. Now it is possible to recognize details in the X-ray photography which previously would barely have been found out. On basis of the program COLOR and a very unclear photography you may convince yourselves of the efficiency of this method. By the way, this method and process is adopted not only in the field of medicine but also for the evaluation of aerial photographs. Also photographs taken by means of other sensors like e.g. infrared and thermal photographs, ultrasonic and radiometric records or computer tomographies are evaluated in this way and manner. These very useful techniques, too, again allow interesting creations in the field of the computer graphics.

Pattern recognition

By scanning a picture line by line, the picture is collected in total. Sometimes, however, the complete information of a picture is not required. Or a complete information would even be more of a hindrance because then the desired information would be obtained only after sampling a large quantity of data. In such case, we speak of pattern recognition. Pattern recognition is required e.g. in robotics. Let us assume that a robot should pick up a part which is transported on a conveyor belt, and install that part in an appliance. Usually the part will arrive on the conveyor belt in any optional position whatsoever. Perhaps, it may even occur that wrong parts are transported on that conveyor belt. The robot must not use such parts but must remove them from the belt. In order to solve this problem, an image processing unit is installed in the robot control system.

The same is tried now with the scanner. We assume that it would be our task to recognize a filled black circle on a white background. By means of a black fibre pen you may draw the circle on a sheet of white paper. On account of the difference in brightness, the circle can be detected without difficulties by means of the photoresistor. Now we can utilize our above described scanner program and scan the whole area. In such case, however, we would be confronted with the above mentioned disadvantages of a considerably long scanning period as well as the extraction of the data.
Let us accelerate the procedure. First of all, the scanner has to search for the circle. Subsequently, the scanner moves along the black-white-border of the figure. In doing this, it will oscillate to and fro between the areas and proceed in a zig-zag mode. Proceeding consequently in this way and manner it must arrive again at the initial position, whereafter the trajectory traveled is evaluated. By approximation the center of the circle is defined as center of gravity of the path traveled by the scanner. This entails the advantage that it is easily programmable and that it leads to results which are sufficiently exact for our purposes. The circle radius results as average value of all distances from the center to the various points on the path traveled. Hence, our example includes all data required for pattern recognition.
In a teach-in phase we can offer to the program a reference circle. If the latter has been recorded by the program PATTERN it will be able upon actuation of a key to study further figures. If center and radius coincide it will be a circle which is completely identical to the reference circle. If on the other hand the radius coincides with the reference circle but the location of the center does not, it will be a circle which is only similar to the reference circle. And exactly this describes the situation for the robot where the parts may arrive in any optional position whatsoever on the conveyor belt. The program will then compute the displacement of the center as referred to the reference circle. If, however, neither the center nor the radius should coincide with the original, it will be a differently large circle. This may occur e.g. if the video system of the robot observes the circle from different distances. In such case, the program PATTERN will also compute the relative distance as referred to that of the reference circle.
Perform experiments with your program and extend it. Computer vision is one of the most interesting fields of research of robotics and we would not exclude that perhaps you will find a more skilful algorithm.

CNC drilling machine

Like any other fischertechnik construction kit, the plotter/scanner may be combined with other fischertechnik components and expanded to more applications. If you still own one or two mini-motors, e.g. out of the fischertechnik service set, you may install a drill stand instead of the recording or write head. The lifting motor of the drill stand is connected at output M4 of the interface. Hereby, the drill is lifted and lowered. The drilling machine itself is represented by a further mini-motor. It is connected directly to the power unit and runs permanently. Of course, you cannot drill in reality by means of this motor, also the suspension would not be stiff enough – but for training the plotter/scanner of this design will be excellent. Also the operating software will be quickly available, as it needs insignificant alterations only as compared to the plotter software. The control of the lifting magnet is omitted; it is substituted by a piece of program which moves the drill downwards, remains in standby position and moves the drill upwards again.

Printout of the programs

On the following pages the BASIC-programs for the plotter/scanner are printed. The programs have been listed in the notation of Commodore 64 and partly need the BASIC-extension program SIMONS BASIC. The programs may also be loaded from the fischertechnik floppy disk training robot/plotter/scanner. The same also applies to the corresponding programs for other types of computers.
In case the floppy disk resp. cassette is not contained in the kit already address your request for the floppy disk resp. cassette to your authorized dealer or company representing fischertechnik in your country. You should indicate details of the type of your computer and disk drive.

Prog. STEP

```
* 500 SYS INIT
  510 REM
  520 REM FISCHERTECHNIK COMPUTING
  530 REM
  540 REM PROGRAM STEP.64
  550 REM
  560 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1985
  570 REM
  580 REM CONNECTION TO THE INTERFACE
  590 REM
  600 REM COIL 1 - M1
  610 REM COIL 2 - M2
  620 REM
  630 REM FUNCTION:
  640 REM CONTROL OF THE STEPPER MOTOR
  650 REM IN EITHER SINGLE STEP OR FREE
  660 REM RUN MODE.
  670 REM
* 1000 PRINT CHR$(147)
  1010 PRINT"FISCHERTECHNIK"
  1020 PRINT"COMPUTING"
  1030 PRINT
  1040 PRINT"STEPPER CONTROL"
  1050 PRINT:PRINT
  1060 PRINT"CONTROL IN SINGLE STEP"
  1070 INPUT"OR IN FREE RUN (S/R)";A$
  1080 IF A$="S" THEN GOTO 1110
  1090 IF A$<>"R" THEN GOTO 1060
  1100 INPUT"DELAY TIME";D
  1110 REM BEGIN OF PHASE CYCLES
  1120 REM
  1130 REM PHASE 1
* 1140 SYS M1,CW
* 1150 SYS M2,CW
  1160 GOSUB 2000
  1170 REM
  1180 REM PHASE 2
* 1190 SYS M1,CCW
* 1200 SYS M2,CW
  1210 GOSUB 2000
  1220 REM
  1230 REM PHASE 3
* 1240 SYS M1,CCW
* 1250 SYS M2,CCW
  1260 GOSUB 2000
  1270 REM
  1280 REM PHASE 4
* 1290 SYS M1,CW
* 1300 SYS M2,CCW
  1310 GOSUB 2000
  1320 REM
  1330 REM END OF CYCLE
  1340 GOTO 1140
  2000 REM DELAY ROUTINE
  2010 IF A$="S" THEN GOTO 2070
  2020 FOR Z=0 TO D
  2030 REM DELAY LOOP
  2040 NEXT Z
  2050 RETURN
  2060 REM SINGLE STEP
  2070 INPUT"HIT >RETURN< TO CONTINUE";B$
  2080 RETURN
```

Prog. PLOT

```
* 500 SYS INIT
  510 REM
  520 REM FISCHERTECHNIK COMPUTING
  530 REM
  540 REM PLOT.64
  550 REM
  560 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 1985
  570 REM
  580 REM PLOTTER TOOLS:
  590 REM
  600 REM 20000 BOX
  605 REM 21000 CIRCLE
  610 REM 22000 AXIS
  630 REM
  640 REM PLOTTER COMMANDS:
  650 REM
  660 REM 40000 HOME
  665 REM 41000 ORIGIN
  670 REM 42000 SET ORIGIN
  675 REM 43000 SCALE
  680 REM 44000 LINE
  685 REM 45000 MOVE
  690 REM 46000 RLINE
  695 REM 47000 RMOVE
  700 REM 48000 CHAR
  705 REM 49000 RCHAR
  710 REM
  720 REM PLOTTER MOVEMENTS:
  730 REM
  740 REM 50000 +X-STEP
  750 REM 51000 -X-STEP
  760 REM 52000 +Y-STEP
  770 REM 53000 -Y-STEP
  780 REM 54000 +X/+Y-DIAGONAL
  790 REM 55000 +X/-Y-DIAGONAL
  800 REM 56000 -X/+Y-DIAGONAL
  810 REM 57000 -X/-Y-DIAGONAL
  820 REM 58000 PEN DOWN
  830 REM 59000 PEN UP
  840 REM 60000 LIMIT SWITCH ?
  850 REM 61000 OUT OF RANGE ?
  860 REM
* 870 PRINT CHR$(147)
  880 PRINT"PLOTTER-INITIALISATION"
  890 GOSUB 40000 :REM HOME
  900 REM
* 910 PRINT CHR$(147)
  920 PRINT"READING CHARACTER SET"
  930 DIM Z%(255,9)
  940 FOR I=0 TO 255
  950 FOR J=0 TO 9
  960 READ Z%(I,J)
  970 NEXT J,I
* 980 PRINT CHR$(147)
  1000 REM
  1010 REM ************************************
  1020 REM ***    START OF USERPROGRAM     ****
  1030 REM ************************************
  19000 END
  19800 REM
  19810 REM IMPORTANT GLOBAL VARIABLES:
  19820 REM
  19830 REM XM,YM: WIDTH,HEIGHT OF PLOTTER AREA
  19835 REM IN PLOTTER STEPS
  19840 REM XJ,YJ: ABSOLUTE COORDINATES
  19845 REM OF THE ACTUAL POINT
  19850 REM        (0,0) = LEFT BOTTOM CORNER
  19855 REM SCALE=1 PLOTTER STEP
  19860 REM XH,YH: VIRTUAL (LOGICAL) ZERO
  19865 REM IN ABSOLUTE COORDINATES
  19870 REM XS,YS: SCALE, NUMBER OF
  19875 REM PLOTTER STEPS PER LOGIC UNIT
  19880 REM XO,YO: (=XOUT,YOUT) FLAGS;
  19890 REM        BECOME 1, WENN XJ,YJ ARE
  19895 REM OUT OF THE RANGE 0...XM,0...YM
  19900 REM
  19910 REM ********************************
  19920 REM *** P L O T T E R T O O L S ***
  19930 REM ********************************
  19940 REM
  19950 REM VARIABLES USED IN THIS ROUTINE:
  19960 REM
  19970 REM G,R,RX,RY,S,SK,W,WA,WE,WS,A$,X$,Y$
  19980 REM X,X0,X1,X2,X3,XA,XE,XH,XS,Y
  19985 REM Y0,Y1,Y2,Y3,YA,YD,YE,YH,YS
  19990 REM
  20000 REM *** BOX ***
  20010 REM
  20020 IF XA > XE THEN X = XE:XE = XA:XA = X
  20030 IF YA > YE THEN Y = YE:YE = YA:YA = Y
  20040 LET X=XA:LET Y=YA:GOSUB 45000
  20050 LET X=XA:LET Y=YE:GOSUB 44000
  20060 LET X=XE:LET Y=YE:GOSUB 44000
  20070 LET X=XE:LET Y=YA:GOSUB 44000
  20080 LET X=XA:LET Y=YA:GOSUB 44000
  20090 IF S=0 THEN RETURN
  20100 REM HATCH
  20110 FOR XD=0 TO (XE-XA)+(YE-YA) STEP ABS(S)
```

```
20120 IF XD((YE-YA) THEN LET X=XA:Y=YE-XD:GOTO 201
      40
20130 LET X=XA+XD+YA-YE:Y=YA
20140 IF S < 0 THEN LET Y=YE+YA-Y
20150 GOSUB 45000
20160 IF XD)(XE-XA) THEN LET X=XE:Y=YE+XE-XA-XD:GO
      TO 20180
20170 LET X=XA+XD:Y=YE
20180 IF S < 0 THEN LET Y=YE+YA-Y
20190 GOSUB 44000
20200 NEXT XD
20210 RETURN
20220 REM
21000 REM *** CIRCLE ***
21010 REM
21020 LET RX=ABS(RX):LET RY=ABS(RY)
21030 LET XA=X:LET YA=Y:REM SAVE CENTER COORDINATE
      S
21040 LET X=XA+RX*COS(ATN(1)*WA/45)
21050 LET Y=YA+RY*SIN(ATN(1)*WA/45)
21060 GOSUB 45000
21065 REM ANGLE STEPS
21070 LET WS=SGN(WE-WA)*800/(RX*XS+RY*YS+40)
21080 FOR W=WA TO WE STEP WS
21090 LET X=XA+RX*COS(ATN(1)*W/45)
21100 LET Y=YA+RY*SIN(ATN(1)*W/45)
21110 GOSUB 44000
21120 NEXT W
21130 LET X=XA+RX*COS(ATN(1)*WE/45)
21140 LET Y=YA+RY*SIN(ATN(1)*WE/45)
21150 GOSUB 44000:REM CLOSE CIRCLE SECTOR
21160 RETURN
21170 REM
22000 REM *** AXIS ***
22010 REM
22020 REM XA,XE,YA,YE SPECIFY POSITION OF THE AXES
22025 REM USING ACTUAL SCALE FACTOR
22030 REM X1,X2,Y1,Y2 ARE THE COORDINATES
22040 REM OF THE END POINTS OF THE AXES
22050 REM
22060 REM SCALING THE PLOTTER AREA
22065 REM CORRESPONDING TO THE AXES
22070 REM
22080 REM X0,X3,Y0,Y3: NEW COORDINATES
22085 REM AT THE BORDERS OF THE PLOTTER AREA
22090 REM
22100 LET X0=X1-(X2-X1)*(XA+XH/XS)/(XE-XA)
22110 LET X3=X2+(X2-X1)*((XM-XH)/XS-XE)/(XE-XA)
22120 LET Y0=Y1-(Y2-Y1)*(YA+YH/YS)/(YE-YA)
22130 LET Y3=Y2+(Y2-Y1)*((YM-YH)/YS-YE)/(YE-YA)
22140 LET XS=XM/(X3-X0)
22150 LET YS=YM/(Y3-Y0)
22160 LET XH=-X0*XS
22170 LET YH=-Y0*YS
22180 REM DRAW AXES
22190 IF Y1 > 0 OR Y2 < 0 THEN 22430
22200 REM X-AXIS
22210 LET X=X1:LET Y=0:GOSUB 45000
22220 LET X=X2:LET Y=0:GOSUB 44000
22230 LET X=-10/XS:LET Y=5/YS:GOSUB 46000
22240 LET X=10/XS:LET Y=-5/YS:GOSUB 47000
22250 LET X=-10/XS:LET Y=-5/YS:GOSUB 46000
22260 REM SCALING AND LETTERING
22270 IF SK=0 THEN GOTO 22430
22275 REM STEP WIDTH FOR SCALING
22280 LET XD=10↑INT(LOG(X2-X1)/LOG(10)-.3)
* 22290 IF (LEN(STR$(XD))+1)*10 > XD*XS THEN XD=XD*5
      :GOTO 22310
* 22300 IF (LEN(STR$(XD))+1)*40 < XD*XS THEN XD=XD/2
22310 LET X0=0
22320 IF X0+1.2*XD < X2 THEN X0=X0+XD:GOTO 22320
22325 REM NO LETTERING AT ZERO
22330 IF ABS(X0) < XD/2 THEN GOTO 22390
22340 LET X=X0:LET Y=3/YS:GOSUB 45000
22350 LET X=X0:LET Y=-3/YS:GOSUB 44000
* 22360 LET A$=STR$(X0):LET R=0:LET G=2
* 22365 IF X0>0 THEN LET A$="+"+RIGHT$(A$,LEN(A$)-1)
22370 LET X=X0-5*LEN(A$)/XS:LET Y=-18/YS
22380 GOSUB 48000
22390 IF X0-1.2*XD > X1 THEN LET X0=X0-XD:GOTO 223
      30
22400 LET A$=X$:LET R=0:LET G=2
22410 LET X=X2-10*LEN(A$)/XS:LET Y=12/YS
22420 GOSUB 48000
22430 IF X1 > 0 OR X2 < 0 THEN RETURN
22440 REM Y-AXIS
22450 LET X=0:LET Y=Y1:GOSUB 45000
22460 LET X=0:LET Y=Y2:GOSUB 44000
22470 LET X=-5/XS:LET Y=-10/YS:GOSUB 46000
22480 LET X=5/XS:LET Y=10/YS:GOSUB 47000
22490 LET X=5/XS:LET Y=-10/YS:GOSUB 46000
22500 REM SCALING AND LETTERING
22510 IF SK=0 THEN RETURN
22515 REM STEP WIDTH FOR SCALING
22520 LET YD=10↑INT(LOG(Y2-Y1)/LOG(10)-.3)
22530 IF YD*YS < 20 THEN YD=YD*100:GOTO 22550
22540 IF YD*YS > 60 THEN YD=YD/2
22550 LET Y0=0
22560 IF Y0+1.2*YD < Y2 THEN LET Y0=Y0+YD:GOTO 225
      60
22565 REM NO LETTERING AT ZERO
22570 IF ABS(Y0) < YD/2 THEN 22630
22580 LET Y=Y0:LET X=-3/XS:GOSUB 45000
22590 LET Y=Y0:LET X=3/XS:GOSUB 44000
* 22600 LET A$=STR$(Y0):LET R=0:LET G=2
* 22605 IF Y0>0 THEN LET A$="+"+RIGHT$(A$,LEN(A$)-1)
22610 LET X=8/XS:LET Y=Y0-6/YS
22620 GOSUB 48000
22630 IF Y0-1.2*YD > Y1 THEN Y0=Y0-YD:GOTO 22570
22640 LET A$=Y$:LET R=1:LET G=2
22650 LET X=-12/XS:LET Y=Y2-10*LEN(A$)/YS
22660 GOSUB 48000
22670 RETURN
22680 REM
39910 REM ****************************************
      **
39920 REM **     S I M P L E   C O M M A N D S
      **
39930 REM ****************************************
      **
39940 REM
39950 REM VARIABLES USED IN THIS ROUTINE:
39960 REM
39970 REM A$,E7,E8,I0,J0,J1,D,DX,DY,NX,NY,PE
39975 REM R,XA%,YA%,XN%,YN%,Z%,Z%(,)
39980 REM X,Y,X1,Y1,X2,Y2,XH,YH,XJ,YJ
39985 REM XM,YM,X0,Y0,XS,YS,XY
39990 REM
40000 REM *** HOME ***
40010 REM
40020 GOSUB 59000:REM PEN UP
40030 LET XJ=0:LET YJ=0
40040 IF USR(E7)=1 AND USR(E8)=1 THEN GOSUB 57060:
      GOTO 40040
* 40050 IF USR(E7)=1 THEN GOSUB 51050:GOTO 40050
* 40060 IF USR(E7)=0 THEN GOSUB 50050:GOTO 40060
* 40070 IF USR(E8)=1 THEN GOSUB 53050:GOTO 40070
* 40080 IF USR(E8)=0 THEN GOSUB 52050:GOTO 40080
40090 GOSUB 54060:GOSUB 54060
40100 LET XM=680:YM=500
40110 LET XH=0:YH=0:XS=1:YS=1
40120 LET XOUT=0:YOUT=0
40130 RETURN
40140 REM
41000 REM *** ORIGIN ***
41020 GOSUB 59000:REM PEN UP
41030 LET X=0:LET Y=0
41040 GOSUB 45000
41050 RETURN
41060 REM
42000 REM *** SET ORIGIN ***
42020 LET XH=XH+X*XS
42030 LET YH=YH+Y*YS
42040 RETURN
42050 REM
43000 REM *** SCALE ***
43010 REM
43020 IF NOT (X1 < X2) OR NOT (Y1<Y2) THEN RETURN
43030 LET XS=XM/(X2-X1)
43040 LET YS=YM/(Y2-Y1)
43050 LET XH=-X1*XS
43060 LET YH=-Y1*YS
43070 RETURN
43080 REM
44000 REM *** LINE ***
44010 REM
44020 GOSUB 58000:REM PEN DOWN
44030 LET X=INT(XH+XS*X-XJ+.5)
44040 LET Y=INT(YH+YS*Y-YJ+.5)
44050 REM X,Y CONTAIN NOW THE RELATIVE DISTANCE
44055 REM IN PLOTTER STEPS
44057 REM NUMBER OF PLOTTER STEPS TO THE TARGET PO
      INT
```

```
 44060 LET DX=ABS(X):LET DY=ABS(Y)
 44070 LET NX=0:LET NY=0:REM STEP COUNTER
 44080 LET D=0:REM DISTANCE TO THE OPTIMAL LINE
 44090 IF NX=DX AND NY=DY THEN RETURN
 44100 IF ABS(D+DY) < ABS(D+DY-DX) THEN 44170
 44110 IF ABS(D-DX) < ABS(D+DY-DX) THEN 44220
 44120 REM DIAGONAL STEP
*44130 ON 1-2*(X>0)-(Y>0) GOSUB 57000,56000,55000,5
       4000
 44140 LET NX=NX+1:LET NY=NY+1
 44150 LET D=D+DY-DX
 44160 GOTO 44090
 44170 REM X-STEP
*44180 ON -(X>0)+1 GOSUB51000,50000
 44190 LET NX=NX+1
 44200 LET D=D+DY
 44210 GOTO 44090
 44220 REM Y-STEP
*44230 ON -(Y>0)+1 GOSUB 53000,52000
 44240 LET NY=NY+1
 44250 LET D=D-DX
 44260 GOTO 44090
 44270 REM
 45000 REM *** MOVE ***
 45010 REM
 45020 GOSUB 59000:REM PEN UP
 45030 GOTO 44030
 45040 REM
 46000 REM *** RLINE ***
 46010 REM
 46020 GOSUB 58000:REM PEN DOWN
 46030 LET X=INT(XS*X+.5)
 46040 LET Y=INT(YS*Y+.5)
 46050 GOTO 44050
 46060 REM
 47000 REM *** RMOVE ***
 47010 REM
 47020 GOSUB 59000:REM PEN UP
 47030 GOTO 46030
 47040 REM
 48000 REM *** CHAR ***
 48010 REM
 48020 GOSUB 45000:REM MOVE TO X,Y
 48030 IF A$="" THEN RETURN
 48040 FOR I0=1 TO LEN(A$)
 48050 LET J0=0
 48060 LET XPREV%=0:LET YPREV%=3:REM STARTING POINT
 48070 IF MID$(A$,I0,1)=CHR$(8) THEN XN%=-5:YN%=3
       :PEN=0:GOTO48220
 48080 LET Z%=Z%(ASC(MID$(A$,I0,1)),J0):LET J1=0
 48090 IF Z%=0 THEN 48200
 48100 LET XN%=INT(Z%/1000):LET YN%=INT(Z%/100)-1
       0*XN%
 48110 LET Z%=100*(Z%-1000*XN%-100*YN%)
 48120 LET PEN=0
 48130 IF XN% > 4 THEN LET XN%=XN%-5:LET PEN=-1
 48140 GOSUB 48250
 48150 LET XPREV%=XN%:LET YPREV%=YN%
 48160 IF Z% <> 0 THEN LET J1=1:GOTO 48100
 48170 IF J0=9 THEN GOTO 48200
 48180 IF J1=1 THEN LET J0=J0+1:GOTO 48080
 48190 IF Z%(ASC(MID$(A$,I0,1)),J0+1) <> 0 THEN LET
       J1=1:GOTO 48100
 48200 REM LETTER IS DRAWN, GOTO RIGHT LOWER CORNER
 48210 LET XN%=5:LET YN%=3:LET PEN=0
 48220 GOSUB 48250
 48230 NEXT I0
 48240 RETURN
 48250 REM DRAW,STRETCH AND TURN VECTOR
 48260 LET X=(XN%-XPREV%)*G
 48270 LET Y=(YN%-YPREV%)*G
 48280 IF R=1 THEN XY=X:X=-Y:Y=XY
 48290 IF R=2 THEN X=-X:Y=-Y
 48300 IF R=3 THEN XY=X:X=Y:Y=-XY
 48310 LET X=X/XS:LET Y=Y/YS
* 48320 IF PEN THEN GOSUB 46000:REM RLINE
* 48330 IF NOT PEN THEN GOSUB 47000:REM RMOVE
 48340 RETURN
 48350 REM
 49000 REM *** RCHAR ***
 49010 REM
 49020 GOSUB 47000:REM RMOVE X,Y
 49030 GOTO 48040
 49040 REM
 49920 REM ****************************************
 49930 REM E L E M E N T A R Y  M O V E M E N T S
 49940 REM ****************************************
 49950 REM
 49960 REM VARIABLES USED IN THIS ROUTINE
 49970 REM
 49980 REM M1...M4,E7,E8,RE,LI,EI,AU,IN,IT,MG$
 49985 REM XJ,YJ,XM,YM,X0,Y0
 49990 REM
 50000 REM *** +X-MOVEMENT ***
 50010 REM
 50020 LET XJ=XJ+1
 50030 GOSUB 61000: REM *** OUT? ***
 50040 IF XOUT THEN RETURN
* 50050 SYS M1,CC:SYS M2,CW:SYS M3,CW
 50060 GOSUB 60000:REM *** ET ??? ***
* 50070 SYS M1,CC:SYS M2,CC:SYS M3,CW
 50080 GOSUB 60000:REM *** ET ??? ***
* 50090 SYS M1,CW:SYS M2,CC:SYS M3,CW
 50100 GOSUB 60000:REM *** ET ??? ***
* 50110 SYS M1,CW:SYS M2,CW:SYS M3,CW
 50120 GOSUB 60000:REM *** ET ??? ***
 50130 RETURN
 50140 REM
 51000 REM *** -X-MOVEMENT ***
 51010 REM
 51020 LET XJ=XJ-1
 51030 GOSUB 61000: REM *** OUT? ***
 51040 IF XOUT THEN RETURN
* 51050 SYS M1,CW:SYS M2,CC:SYS M3,CW
 51060 GOSUB 60000:REM *** ET ??? ***
* 51070 SYS M1,CC:SYS M2,CC:SYS M3,CW
 51080 GOSUB 60000:REM *** ET ??? ***
* 51090 SYS M1,CC:SYS M2,CW:SYS M3,CW
 51100 GOSUB 60000:REM *** ET ??? ***
* 51110 SYS M1,CW:SYS M2,CW:SYS M3,CW
 51120 GOSUB 60000:REM *** ET ??? ***
 51130 RETURN
 51140 REM
 52000 REM *** +Y-MOVEMENT ***
 52010 REM
 52020 LET YJ=YJ+1
 52030 GOSUB 61000: REM *** OUT? ***
 52040 IF YOUT THEN RETURN
* 52050 SYS M1,CC:SYS M2,CW:SYS M3,CW
 52060 GOSUB 60000:REM *** ET ??? ***
* 52070 SYS M1,CC:SYS M2,CW:SYS M3,CC
 52080 GOSUB 60000:REM *** ET ??? ***
* 52090 SYS M1,CW:SYS M2,CW:SYS M3,CC
 52100 GOSUB 60000:REM *** ET ??? ***
* 52110 SYS M1,CW:SYS M2,CW:SYS M3,CW
 52120 GOSUB 60000:REM *** ET ??? ***
 52130 RETURN
 52140 REM
 53000 REM *** -Y-MOVEMENT ***
 53010 REM
 53020 LET YJ=YJ-1
 53030 GOSUB 61000: REM *** OUT? ***
 53040 IF YOUT THEN RETURN
* 53050 SYS M1,CW:SYS M2,CW:SYS M3,CC
 53060 GOSUB 60000:REM *** ET ??? ***
* 53070 SYS M1,CC:SYS M2,CW:SYS M3,CC
 53080 GOSUB 60000:REM *** ET ??? ***
* 53090 SYS M1,CC:SYS M2,CW:SYS M3,CW
 53100 GOSUB 60000:REM *** ET ??? ***
* 53110 SYS M1,CW:SYS M2,CW:SYS M3,CW
 53120 GOSUB 60000:REM *** ET ??? ***
 53130 RETURN
 53140 REM
 54000 REM *** +X/+Y-DIAGONAL ***
 54010 REM
 54020 LET XJ=XJ+1:LET YJ=YJ+1
 54030 GOSUB 61000: REM *** OUT? ***
 54040 IF XOUT THEN 52040
 54050 IF YOUT THEN 50040
* 54060 SYS M1,CC:SYS M2,CW:SYS M3,CW
 54070 GOSUB 60000:REM *** ET ??? ***
* 54080 SYS M1,CC:SYS M2,CC:SYS M3,CC
 54090 GOSUB 60000:REM *** ET ??? ***
* 54100 SYS M1,CW:SYS M2,CC:SYS M3,CC
 54110 GOSUB 60000:REM *** ET ??? ***
* 54120 SYS M1,CW:SYS M2,CW:SYS M3,CW
 54130 GOSUB 60000:REM *** ET ??? ***
 54140 RETURN
 54150 REM
 55000 REM *** +X/-Y-DIAGONAL ***
 55010 REM
 55020 LET XJ=XJ+1:LET YJ=YJ-1
 55030 GOSUB 61000: REM *** OUT? ***
 55040 IF XOUT THEN 53040
 55050 IF YOUT THEN 50040
* 55060 SYS M1,CW:SYS M2,CW:SYS M3,CC
 55070 GOSUB 60000:REM *** ET ??? ***
```

```
* 55080 SYS M1,CC:SYS M2,CW:SYS M3,CC
  55090 GOSUB 60000:REM *** ET ??? ***
* 55100 SYS M1,CC:SYS M2,CC:SYS M3,CW
  55110 GOSUB 60000:REM *** ET ??? ***
* 55120 SYS M1,CW:SYS M2,CC:SYS M3,CW
  55130 GOSUB 60000:REM *** ET ??? ***
* 55140 SYS M1,CW:SYS M2,CW:SYS M3,CW
  55150 GOSUB 60000:REM *** ET ??? ***
  55160 RETURN
  55170 REM
  56000 REM *** -X/+Y-DIAGONAL ***
  56010 REM
  56020 LET XJ=XJ-1:LET YJ=YJ+1
  56030 GOSUB 61000: REM *** OUT? ***
  56040 IF XOUT THEN GOTO 52040
  56050 IF YOUT THEN GOTO 51040
* 56060 SYS M1,CW:SYS M2,CC:SYS M3,CW
  56070 GOSUB 60000:REM *** ET ??? ***
* 56080 SYS M1,CC:SYS M2,CC:SYS M3,CW
  56090 GOSUB 60000:REM *** ET ??? ***
* 56100 SYS M1,CC:SYS M2,CW:SYS M3,CC
  56110 GOSUB 60000:REM *** ET ??? ***
* 56120 SYS M1,CW:SYS M2,CW:SYS M3,CC
  56130 GOSUB 60000:REM *** ET ??? ***
* 56140 SYS M1,CW:SYS M2,CW:SYS M3,CW
  56150 GOSUB 60000:REM *** ET ??? ***
  56160 RETURN
  56170 REM
  57000 REM *** -X/-Y-DIAGONAL ***
  57010 REM
  57020 LET XJ=XJ-1:LET YJ=YJ-1
  57030 GOSUB 61000: REM *** OUT? ***
  57040 IF XOUT THEN GOTO 53040
  57050 IF YOUT THEN GOTO 51040
* 57060 SYS M1,CW:SYS M2,CC:SYS M3,CC
  57070 GOSUB 60000:REM *** ET ??? ***
* 57080 SYS M1,CC:SYS M2,CC:SYS M3,CC
  57090 GOSUB 60000:REM *** ET ??? ***
* 57100 SYS M1,CC:SYS M2,CW:SYS M3,CW
  57110 GOSUB 60000:REM *** ET ??? ***
* 57120 SYS M1,CW:SYS M2,CW:SYS M3,CW
  57130 GOSUB 60000:REM *** ET ??? ***
  57140 RETURN
  57150 REM
  58000 REM *** MG-ON ***
  58010 REM
  58020 LET MG$="ON"
  58030 IF (XOUT) OR YOUT THEN RETURN
* 58040 SYS INIT
  58050 FOR Z=1 TO 400:NEXT
* 58060 SYS M4,CW
* 58070 FOR Z=1 TO 100:SYS M4,CW:NEXT
  58080 RETURN
  58090 REM
  59000 REM *** MG-OFF ***
  59010 REM
  59020 LET MG$="OUT"
* 59030 SYS M4,CCW
* 59040 SYS M4,OFF
  59050 FOR Z=1 TO 100:NEXT
  59060 RETURN
  59070 REM
  60000 REM *** LIMIT SWITCH CLOSED ***
  60010 REM
* 60020 IF (USR(E7)=0 AND XJ <> 0) OR (USR(E8)=0 AND
      YJ <> 0) THEN GOTO 60040
  60030 RETURN
* 60040 PRINT CHR$(147);CHR$(18)"WARNING! PLOTTER IS
    N'T ADJUSTED";CHR$(147)
  60050 STOP
  60060 REM
  61000 REM *** OUT OF RANGE ? ***
  61010 REM
  61020 IF (XOUT) OR YOUT THEN GOTO 61090
  61030 REM LAST POSITION WAS INSIDE
  61040 LET XOUT=XJ < 0 OR XJ > XM
  61050 LET YOUT=YJ < 0 OR YJ > YM
  61060 IF (XOUT) OR YOUT THEN GOSUB 59030
  61080 RETURN
  61090 REM LAST POSITION WAS OUTSIDE
  61100 LET XOUT=XJ < 0 OR XJ > XM
  61110 LET YOUT=YJ < 0 OR YJ > YM
  61120 IF (XOUT) OR YOUT THEN RETURN
  61130 IF MG$="CW"THEN GOSUB 58000
  61150 RETURN
  61900 REM ** CHARACTER SET ***
  62000 DATA0,0,0,0,0,0,0,0,0,0
  62001 DATA0,0,0,0,0,0,0,0,0,0
  62002 DATA0,0,0,0,0,0,0,0,0,0
  62003 DATA0,0,0,0,0,0,0,0,0,0
  62004 DATA0,0,0,0,0,0,0,0,0,0
  62005 DATA0,0,0,0,0,0,0,0,0,0
  62006 DATA0,0,0,0,0,0,0,0,0,0
  62007 DATA0,0,0,0,0,0,0,0,0,0
  62008 DATA0,0,0,0,0,0,0,0,0,0
  62009 DATA0,0,0,0,0,0,0,0,0,0
  62010 DATA0,0,0,0,0,0,0,0,0,0
  62011 DATA0,0,0,0,0,0,0,0,0,0
  62012 DATA0,0,0,0,0,0,0,0,0,0
  62013 DATA0,0,0,0,0,0,0,0,0,0
  62014 DATA0,0,0,0,0,0,0,0,0,0
  62015 DATA0,0,0,0,0,0,0,0,0,0
  62016 DATA0,0,0,0,0,0,0,0,0,0
  62017 DATA0,0,0,0,0,0,0,0,0,0
  62018 DATA0,0,0,0,0,0,0,0,0,0
  62019 DATA0,0,0,0,0,0,0,0,0,0
  62020 DATA0,0,0,0,0,0,0,0,0,0
  62021 DATA0,0,0,0,0,0,0,0,0,0
  62022 DATA0,0,0,0,0,0,0,0,0,0
  62023 DATA0,0,0,0,0,0,0,0,0,0
  62024 DATA0,0,0,0,0,0,0,0,0,0
  62025 DATA0,0,0,0,0,0,0,0,0,0
  62026 DATA0,0,0,0,0,0,0,0,0,0
  62027 DATA0,0,0,0,0,0,0,0,0,0
  62028 DATA0,0,0,0,0,0,0,0,0,0
  62029 DATA0,0,0,0,0,0,0,0,0,0
  62030 DATA0,0,0,0,0,0,0,0,0,0
  62031 DATA0,0,0,0,0,0,0,0,0,0
  62032 DATA0,0,0,0,0,0,0,0,0,0
  62033 DATA2975,2473,0,0,0,0,0,0,0,0
  62034 DATA3987,1967,0,0,0,0,0,0,0,0
  62035 DATA1369,3983,4757,595,0,0,0,0,0,0
  62036 DATA2379,4788,6857,6686,9584,6455,0,0,0,0
  62037 DATA9908,5767,6858,4594,8485,9500,0,0,0,0
  62038 DATA4573,6354,5587,8879,6958,5793,0,0,0,0
  62039 DATA2989,8877,0,0,0,0,0,0,0,0
  62040 DATA3977,7583,0,0,0,0,0,0,0,0
  62041 DATA1977,7563,0,0,0,0,0,0,0,0
  62042 DATA696,3864,1884,0,0,0,0,0,0,0
  62043 DATA696,2874,0,0,0,0,0,0,0,0
  62044 DATA2484,8372,0,0,0,0,0,0,0,0
  62045 DATA696,0,0,0,0,0,0,0,0,0
  62046 DATA2374,0,0,0,0,0,0,0,0,0
  62047 DATA9900,0,0,0,0,0,0,0,0,0
  62048 DATA557,6989,9795,8363,5500,0,0,0,0,0
  62049 DATA3363,2379,6800,0,0,0,0,0,0,0
  62050 DATA869,8998,9754,5393,0,0,0,0,0,0
  62051 DATA869,8998,9786,7636,9594,8363,5400,0,0,0
  62052 DATA4555,7939,8300,0,0,0,0,0,0,0
  62053 DATA463,8394,9687,6756,5999,0,0,0,0,0
  62054 DATA566,8695,9483,6354,5869,8998,0,0,0,0
  62055 DATA9959,0,0,0,0,0,0,0,0,0
  62056 DATA1657,5869,8998,9786,6655,5463,8394,9586,
      0,0
  62057 DATA463,8394,9889,6958,5766,8697,0,0,0,0
  62058 DATA2374,2677,0,0,0,0,0,0,0,0
  62059 DATA2675,2473,6200,0,0,0,0,0,0,0
  62060 DATA4956,9300,0,0,0,0,0,0,0,0
  62061 DATA595,4757,0,0,0,0,0,0,0,0
  62062 DATA996,5300,0,0,0,0,0,0,0,0
  62063 DATA869,8998,9775,2473,0,0,0,0,0,0
  62064 DATA463,8394,8565,5667,8796,8517,5869,8998,0
      ,0
  62065 DATA5679,9693,4555,0,0,0,0,0,0,0
  62066 DATA8394,9586,6636,9798,8959,1963,0,0,0,0
  62067 DATA4889,6958,5463,8394,0,0,0,0,0,0
  62068 DATA8394,9889,5919,6300,0,0,0,0,0,0
  62069 DATA5999,3656,393,0,0,0,0,0,0,0
  62070 DATA5999,3656,0,0,0,0,0,0,0,0
  62071 DATA4889,6958,5463,8394,9676,0,0,0,0,0
  62072 DATA5906,9649,9300,0,0,0,0,0,0,0
  62073 DATA1383,2379,1989,0,0,0,0,0,0,0
  62074 DATA463,7384,8929,9900,0,0,0,0,0,0
  62075 DATA5906,9906,9300,0,0,0,0,0,0,0
  62076 DATA953,9300,0,0,0,0,0,0,0,0
  62077 DATA5976,9993,0,0,0,0,0,0,0,0
  62078 DATA5993,9900,0,0,0,0,0,0,0,0
  62079 DATA458,6989,9894,8363,5400,0,0,0,0,0
  62080 DATA1369,989,9897,8666,0,0,0,0,0,0
  62081 DATA458,6989,9894,8363,5425,9300,0,0,0,0
  62082 DATA1369,989,9897,8666,2693,0,0,0,0,0
  62083 DATA463,8394,9586,6657,5869,8998,0,0,0,0
  62084 DATA2379,999,0,0,0,0,0,0,0,0
  62085 DATA954,6383,9499,0,0,0,0,0,0,0
  62086 DATA973,9900,0,0,0,0,0,0,0,0
  62087 DATA956,6375,8396,9900,0,0,0,0,0,0
  62088 DATA993,4953,0,0,0,0,0,0,0,0
```

```
62089 DATA976,7326,9900,0,0,0,0,0,0,0
62090 DATA999,5393,0,0,0,0,0,0,0,0
62091 DATA5679,9693,4555,958,4998,0,0,0,0,0
62092 DATA457,6888,9794,8363,5409,5849,9800,0,0,0
62093 DATA954,6383,9499,1968,3988,0,0,0,0,0
62094 DATA779,9700,0,0,0,0,0,0,0,0
62095 DATA90,0,0,0,0,0,0,0,0,0
62096 DATA3979,7887,0,0,0,0,0,0,0,0
62097 DATA667,8796,9345,6554,6383,9400,0,0,0,0
62098 DATA953,667,8796,9483,6354,0,0,0,0,0
62099 DATA4687,6756,5463,8394,0,0,0,0,0,0
62100 DATA4993,4687,6756,5463,8394,0,0,0,0,0
62101 DATA585,9687,6756,5463,8394,0,0,0,0,0
62102 DATA2378,8999,3666,0,0,0,0,0,0,0
62103 DATA261,8192,9687,6756,5463,8394,0,0,0,0
62104 DATA5906,6787,9693,0,0,0,0,0,0,0
62105 DATA3363,2377,6728,7900,0,0,0,0,0,0
62106 DATA161,7277,6728,7900,0,0,0,0,0,0
62107 DATA1963,1597,1593,0,0,0,0,0,0,0
62108 DATA303,3363,2379,6900,0,0,0,0,0,0
62109 DATA5787,9693,2377,0,0,0,0,0,0,0
62110 DATA753,667,8796,9300,0,0,0,0,0,0
62111 DATA456,6787,9694,8363,5400,0,0,0,0,0
62112 DATA157,667,8796,9483,6354,0,0,0,0,0
62113 DATA4197,4687,6756,5463,8394,0,0,0,0,0
62114 DATA5706,6787,9600,0,0,0,0,0,0,0
62115 DATA463,8394,8565,5667,8796,0,0,0,0,0
62116 DATA1686,2973,8300,0,0,0,0,0,0,0
62117 DATA754,6383,9447,9300,0,0,0,0,0,0
62118 DATA773,9700,0,0,0,0,0,0,0,0
62119 DATA755,6375,8395,9700,0,0,0,0,0,0
62120 DATA793,397,0,0,0,0,0,0,0,0
62121 DATA773,4761,0,0,0,0,0,0,0,0
62122 DATA797,5393,0,0,0,0,0,0,0,0
62123 DATA667,8796,9345,6554,6383,9419,6839,8800,0
      ,0
62124 DATA456,6787,9694,8363,5439,8819,6800,0,0,0
62125 DATA3988,1968,754,6383,9447,9300,0,0,0,0
62126 DATA5869,8998,9786,7636,9594,8373,6400,0,0,0
62127 DATA0,0,0,0,0,0,0,0,0,0
62128 DATA0,0,0,0,0,0,0,0,0,0
62129 DATA0,0,0,0,0,0,0,0,0,0
62130 DATA0,0,0,0,0,0,0,0,0,0
62131 DATA0,0,0,0,0,0,0,0,0,0
62132 DATA0,0,0,0,0,0,0,0,0,0
62133 DATA0,0,0,0,0,0,0,0,0,0
62134 DATA0,0,0,0,0,0,0,0,0,0
62135 DATA0,0,0,0,0,0,0,0,0,0
62136 DATA0,0,0,0,0,0,0,0,0,0
62137 DATA0,0,0,0,0,0,0,0,0,0
62138 DATA0,0,0,0,0,0,0,0,0,0
62139 DATA0,0,0,0,0,0,0,0,0,0
62140 DATA0,0,0,0,0,0,0,0,0,0
62141 DATA0,0,0,0,0,0,0,0,0,0
62142 DATA0,0,0,0,0,0,0,0,0,0
62143 DATA0,0,0,0,0,0,0,0,0,0
62144 DATA0,0,0,0,0,0,0,0,0,0
62145 DATA0,0,0,0,0,0,0,0,0,0
62146 DATA0,0,0,0,0,0,0,0,0,0
62147 DATA0,0,0,0,0,0,0,0,0,0
62148 DATA0,0,0,0,0,0,0,0,0,0
62149 DATA0,0,0,0,0,0,0,0,0,0
62150 DATA0,0,0,0,0,0,0,0,0,0
62151 DATA0,0,0,0,0,0,0,0,0,0
62152 DATA0,0,0,0,0,0,0,0,0,0
62153 DATA0,0,0,0,0,0,0,0,0,0
62154 DATA0,0,0,0,0,0,0,0,0,0
62155 DATA0,0,0,0,0,0,0,0,0,0
62156 DATA0,0,0,0,0,0,0,0,0,0
62157 DATA0,0,0,0,0,0,0,0,0,0
62158 DATA0,0,0,0,0,0,0,0,0,0
62159 DATA0,0,0,0,0,0,0,0,0,0
62160 DATA0,0,0,0,0,0,0,0,0,0
62161 DATA0,0,0,0,0,0,0,0,0,0
62162 DATA0,0,0,0,0,0,0,0,0,0
62163 DATA0,0,0,0,0,0,0,0,0,0
62164 DATA0,0,0,0,0,0,0,0,0,0
62165 DATA0,0,0,0,0,0,0,0,0,0
62166 DATA0,0,0,0,0,0,0,0,0,0
62167 DATA0,0,0,0,0,0,0,0,0,0
62168 DATA0,0,0,0,0,0,0,0,0,0
62169 DATA0,0,0,0,0,0,0,0,0,0
62170 DATA0,0,0,0,0,0,0,0,0,0
62171 DATA0,0,0,0,0,0,0,0,0,0
62172 DATA0,0,0,0,0,0,0,0,0,0
62173 DATA0,0,0,0,0,0,0,0,0,0
62174 DATA0,0,0,0,0,0,0,0,0,0
62175 DATA0,0,0,0,0,0,0,0,0,0
62176 DATA0,0,0,0,0,0,0,0,0,0
62177 DATA0,0,0,0,0,0,0,0,0,0
62178 DATA0,0,0,0,0,0,0,0,0,0
62179 DATA0,0,0,0,0,0,0,0,0,0
62180 DATA0,0,0,0,0,0,0,0,0,0
62181 DATA0,0,0,0,0,0,0,0,0,0
62182 DATA0,0,0,0,0,0,0,0,0,0
62183 DATA0,0,0,0,0,0,0,0,0,0
62184 DATA0,0,0,0,0,0,0,0,0,0
62185 DATA0,0,0,0,0,0,0,0,0,0
62186 DATA0,0,0,0,0,0,0,0,0,0
62187 DATA0,0,0,0,0,0,0,0,0,0
62188 DATA0,0,0,0,0,0,0,0,0,0
62189 DATA0,0,0,0,0,0,0,0,0,0
62190 DATA0,0,0,0,0,0,0,0,0,0
62191 DATA0,0,0,0,0,0,0,0,0,0
62192 DATA0,0,0,0,0,0,0,0,0,0
62193 DATA0,0,0,0,0,0,0,0,0,0
62194 DATA0,0,0,0,0,0,0,0,0,0
62195 DATA0,0,0,0,0,0,0,0,0,0
62196 DATA0,0,0,0,0,0,0,0,0,0
62197 DATA0,0,0,0,0,0,0,0,0,0
62198 DATA0,0,0,0,0,0,0,0,0,0
62199 DATA0,0,0,0,0,0,0,0,0,0
62200 DATA0,0,0,0,0,0,0,0,0,0
62201 DATA0,0,0,0,0,0,0,0,0,0
62202 DATA0,0,0,0,0,0,0,0,0,0
62203 DATA0,0,0,0,0,0,0,0,0,0
62204 DATA0,0,0,0,0,0,0,0,0,0
62205 DATA0,0,0,0,0,0,0,0,0,0
62206 DATA0,0,0,0,0,0,0,0,0,0
62207 DATA0,0,0,0,0,0,0,0,0,0
62208 DATA0,0,0,0,0,0,0,0,0,0
62209 DATA0,0,0,0,0,0,0,0,0,0
62210 DATA0,0,0,0,0,0,0,0,0,0
62211 DATA0,0,0,0,0,0,0,0,0,0
62212 DATA0,0,0,0,0,0,0,0,0,0
62213 DATA0,0,0,0,0,0,0,0,0,0
62214 DATA0,0,0,0,0,0,0,0,0,0
62215 DATA0,0,0,0,0,0,0,0,0,0
62216 DATA0,0,0,0,0,0,0,0,0,0
62217 DATA0,0,0,0,0,0,0,0,0,0
62218 DATA0,0,0,0,0,0,0,0,0,0
62219 DATA0,0,0,0,0,0,0,0,0,0
62220 DATA0,0,0,0,0,0,0,0,0,0
62221 DATA0,0,0,0,0,0,0,0,0,0
62222 DATA0,0,0,0,0,0,0,0,0,0
62223 DATA0,0,0,0,0,0,0,0,0,0
62224 DATA0,0,0,0,0,0,0,0,0,0
62225 DATA0,0,0,0,0,0,0,0,0,0
62226 DATA0,0,0,0,0,0,0,0,0,0
62227 DATA0,0,0,0,0,0,0,0,0,0
62228 DATA0,0,0,0,0,0,0,0,0,0
62229 DATA0,0,0,0,0,0,0,0,0,0
62230 DATA0,0,0,0,0,0,0,0,0,0
62231 DATA0,0,0,0,0,0,0,0,0,0
62232 DATA0,0,0,0,0,0,0,0,0,0
62233 DATA0,0,0,0,0,0,0,0,0,0
62234 DATA0,0,0,0,0,0,0,0,0,0
62235 DATA0,0,0,0,0,0,0,0,0,0
62236 DATA0,0,0,0,0,0,0,0,0,0
62237 DATA0,0,0,0,0,0,0,0,0,0
62238 DATA0,0,0,0,0,0,0,0,0,0
62239 DATA0,0,0,0,0,0,0,0,0,0
62240 DATA0,0,0,0,0,0,0,0,0,0
62241 DATA0,0,0,0,0,0,0,0,0,0
62242 DATA0,0,0,0,0,0,0,0,0,0
62243 DATA0,0,0,0,0,0,0,0,0,0
62244 DATA0,0,0,0,0,0,0,0,0,0
62245 DATA0,0,0,0,0,0,0,0,0,0
62246 DATA0,0,0,0,0,0,0,0,0,0
62247 DATA0,0,0,0,0,0,0,0,0,0
62248 DATA0,0,0,0,0,0,0,0,0,0
62249 DATA0,0,0,0,0,0,0,0,0,0
62250 DATA0,0,0,0,0,0,0,0,0,0
62251 DATA0,0,0,0,0,0,0,0,0,0
62252 DATA0,0,0,0,0,0,0,0,0,0
62253 DATA0,0,0,0,0,0,0,0,0,0
62254 DATA0,0,0,0,0,0,0,0,0,0
62255 DATA0,0,0,0,0,0,0,0,0,0
63000 DEF FN X(Z)=135+10*Z
63010 DEF FN Y(Z)=150-10*Z
63015 REM USE THIS PROGRAM WITH GOTO 63000
* 63020 PRINT CHR$(147)
63030 PRINT" CHARACTER DESIGN PROGRAMM "
63032 PRINT
* 63035 PRINT"PROGRAM NEEDS 'SIMONS BASIC'"
63040 PRINT:PRINT" FUNCTION DESCRIPTION"
63050 PRINT:PRINT" CRSR UP      : UP
```

```
63060 PRINT:PRINT" CRSR DOWN   : DOWN
63070 PRINT:PRINT" CRSR LEFT   : LEFT
63080 PRINT:PRINT" CRSR RIGHT  : RIGHT
63090 PRINT:PRINT" L   :LINE FROM LAST POINT"
63100 PRINT:PRINT" M   :MOVE FROM LAST POINT"
63110 PRINT:PRINT" C   :CLEAR SCREEN"
63120 PRINT:PRINT" CR  :ACCEPT CHARACTER"
63130 PRINT:PRINT" X   :EXIT ":PRINT
63140 DIM S%(19)
63150 INPUT" ASCII CODE OF CHARACTER (0-255)";ZN
63160 IF ZN<0 OR ZN>255 THEN GOTO 63150
63170 FOR I=0 TO 19:LET S%(I)=0:NEXT:LET I%=0
* 63180 HIRES1,0
63190 FOR Y=0 TO 9
63200 FOR X=0 TO 4
* 63210 PLOT FN X(X),FN Y(Y),1
63220 NEXT
63230 NEXT
63240 LET X=0:LET Y=3
63250 LET XO=X:LET YO=Y
* 63260 CIRCLE FN X(X),FN Y(Y),2,2,1
63270 GET A$:IF A$="" THEN 63270
* 63280 IF A$=CHR$(145) THEN GOSUB 63380
* 63290 IF A$=CHR$(17) THEN GOSUB 63430
63300 IF A$=CHR$(29) THEN GOSUB 63480
* 63310 IF A$=CHR$(157) THEN GOSUB 63530
63320 IF A$="C" THEN GOTO 63170
63330 IF A$="X" THEN STOP
63340 IF A$=CHR$(13) THEN GOTO 63670
63350 IF A$="M" THEN GOSUB 63580
63360 IF A$="L" THEN GOSUB 63580
63370 GOTO 63270
63380 IF Y=9 THEN RETURN
* 63390 CIRCLE FN X(X),FN Y(Y),2,2,0
63400 LET Y=Y+1
* 63410 CIRCLE FN X(X),FN Y(Y),2,2,1
63420 RETURN
63430 IF Y=0 THEN RETURN
* 63440 CIRCLE FN X(X),FN Y(Y),2,2,0
63450 LET Y=Y-1
* 63460 CIRCLE FN X(X),FN Y(Y),2,2,1
63470 RETURN
63480 IF X=4 THEN RETURN
* 63490 CIRCLE FN X(X),FN Y(Y),2,2,0
63500 LET X=X+1
* 63510 CIRCLE FN X(X),FN Y(Y),2,2,1
63520 RETURN
63530 IF X=0 THEN RETURN
* 63540 CIRCLE FN X(X),FN Y(Y),2,2,0
63550 LET X=X-1
63560 CIRCLE FN X(X),FN Y(Y),2,2,1
63570 RETURN
63580 IF I%=19 THEN GOTO 63670
63590 IF A$="L" THEN S%(I%)=50
63600 LET S%(I%)=S%(I%)+10*X+Y
63610 LET I%=I%+1
* 63620 CIRCLE FN X(X),FN Y(Y),2,2,0
* 63630 IF A$="L" THEN LINE FN X(XO),FN Y(YO),FN X(
      X),FN Y(Y),1
63640 LET XO=X:LET YO=Y
* 63650 CIRCLE FN X(X),FN Y(Y),2,2,1
63660 RETURN
63670 FOR I=0 TO 18 STEP 2
63680 IF S%(I)=0 THEN D$(I/2)="0":NEXT:GOTO 63770
63690 LET A$=STR$(S%(I))
63700 LET B$=STR$(S%(I+1))
63710 IF LEN(A$)=1 THEN A$="0"+A$
63720 IF LEN(B$)=1 THEN B$="0"+B$
63730 D$(I/2)=A$+B$
63740 LET A$="":LET B$=""
63750 LET D$(I/2)=MID$(D$(I/2),2,2)+MID$(D$(I/2),5
      ,2)
63760 NEXT
63770 LET A$=STR$(62000+ZN)+"DATA"+D$(0)+","+D$(1)
      +","+D$(2)+","+D$(3)+","+D$(4)
63780 A$=A$+","+D$(5)+","+D$(6)+","+D$(7)+","+D$(8
      )+","+D$(9)
63790 PRINT"INSERT THE LINE BELOW IN YOUR"
63800 PRINT"PROGRAM BY MOVING THE CURSOR"
63810 PRINT"ON IT AND PRESS >RETURN<."
63820 PRINT A$
```

Prog. FUNCTION

```
1000 DEF FNY(X)=SIN(X)*EXP(-X/10)
1010 LET F$="SIN(X)*EXP(-X/10)"
1020 LET XA=0: LET XE=62.8
1030 REM
1100 REM *** PROGRAM "FUNCTION" ***
1110 PRINT:PRINT"       F I S C H E R T E C H N I
     K "
1111 PRINT:PRINT"            C O M P U T I N G"
1112 PRINT
1113 PRINT" ONE-DIMENSIOAL FUNCTION PLOT"
1115 PRINT CHR$(29);
1117 PRINT F$
1120 REM CALCULATE MIN AND MAX OF THE FUNCTION
1130 LET MI=FNY(XA)
1140 LET MA=FNY(XA)
1150 FOR X=XA TO XE STEP (XE-XA)/200
1160 IF FNY(X)>MA THEN LET MA=FNY(X)
1170 IF FNY(X)<MI THEN LET MI=FNY(X)
1180 NEXT X
1190 LET X1=XA-(XE-XA)/20
1200 LET X2=XE+(XE-XA)/20
1210 LET Y1=MI-(MA-MI)/20
1220 LET Y2=MA+(MA-MI)/20
1230 LET XA=10: LET XE=XM-10
1240 LET YA=10: LET YE=YM-10
1250 LET X$="X"
1260 LET Y$=F$
1270 LET SK=1
1280 GOSUB 22000: REM DRAW AXES
1290 LET X=X1+(X2-X1)/22
1300 LET Y=FNY(X)
1310 GOSUB 45000: REM GO TO START
1320 FOR X0=X1+(X2-X1)/22 TO X2-(X2-X1)/22 STEP (X
     2-X1)/220
1330 LET X=X0: LET Y=FNY(X)
1340 GOSUB 44000
1350 NEXT X0
1360 GOSUB 40000
1370 END
```

Prog. PARAM.F

```
1000 DEF FNX(T)=COS(T)
1010 DEF FNY(T)=SIN(T)
1020 LET F$="CIRCLE"
1030 LET TA=0: LET TE=6.28: LET TS=0.1
1040 REM LOOP FROM 0 TO 2*π STEP 0.1
1050 LET AX=1: LET SK=1
1060 REM DRAW AXIS AND SCALE THEM
1070 LET XA=-1.1: LET XE=1.1: LET YA=-1.1: LET YE=
     1.1
1080 REM
1100 REM *** PROGRAM "PARAM.F" ***
1110 PRINT"       F I S C H E R T E C H N I K"
1111 PRINT:PRINT"            C O M P U T I N G"
1113 PRINT:PRINT" TWO - DIMENSIOAL FUNCTION PLOT "
1114 PRINT CHR$(29);
1115 PRINT F$
1120 REM SAME SCALE FOR X AND Y
1130 REM BOX XA,YA,XE,YE IN THE MIDDLE OF THE PAPE
     R
1140 REM
1150 IF (XE-XA)*YM>(YE-YA)*XM THEN GOTO 1190
```

```
1160 LET YS=YM/(YE-YA): LET YH=-YA*YS
1170 LET XS=YS: LET XH=(XM-(XE+XA)*XS)/2
1180 GOTO 1210
1190 LET XS=XM/(XE-XA): LET XH=-XA*XS
1200 LET YS=XS: LET YH=(YM-(YE+YA)*YS)/2
1210 IF AX=0 THEN GOTO 1250
1220 LET X1=XA: LET X2=XE: LET Y1=YA: LET Y2=YE
1230 LET X$=F$: LET Y$=" "
1240 GOSUB 22000
1250 LET X=FNX(TA)
1260 LET Y=FNY(TA)
1270 GOSUB 45000: REM GOTO START
1280 FOR T=TA TO TE STEP TS
1290 LET X=FNX(T)
1300 LET Y=FNY(T)
1310 GOSUB 44000
1320 NEXT T
1330 LET X=FNX(TE)
1340 LET Y=FNY(TE)
1350 GOSUB 44000
1360 GOSUB 40000
1370 END
```

Prog. D 3

```
1000 REM *** PROGRAM 'D3' ***
1010 PRINT"      F I S C H E R T E C H N I K"
1011 PRINT:PRINT"              C O M P U T I N G"
1013 PRINT:PRINT" PLOT OF A THREE - DIMENSIONAL OB
     JECT"
* 1014 PRINT CHR$(29);
1015 PRINT F$
1020 DIM X(100,3)
1030 REM ************************************
1040 LET I=0
1050 READ X(I,1),X(I,2),X(I,3),X(I,0)
1060 IF X(I,0)=-1 THEN GOTO 1100
1070 LET I=I+1
1080 GOTO 1050
* 1100 PRINT CHR$(147)
1110 PRINT"PROJECTION ANGLE IN DEGREES:"
1120 INPUT"THETA:";TH: LET TH=ATN(1)*TH/45
1130 INPUT"PHI  :";PH: LET PH=ATN(1)*PH/45
1140 LET N1=SIN(TH)*COS(PH)
1150 LET N2=SIN(TH)*SIN(PH)
1160 LET N3=COS(TH)
1170 LET N0=SQR(N1*N1+N2*N2)
1180 LET I=0: REM DETERMINE THE SIZE OF THE PICTUR
     E
1190 IF X(I,0)=-1 THEN GOTO 1280
1200 GOSUB 1430
1210 IF I=0 THEN LET XA=X:XE=X:YA=Y:YE=Y:GOTO 1260
1220 IF X<XA THEN LET XA=X
1230 IF X>XE THEN LET XE=X
1240 IF Y<YA THEN LET YA=Y
1250 IF Y>YE THEN LET YE=Y
1260 LET I=I+1
1270 GOTO 1190
1280 REM SAME SCALE ON BOTH AXES
1285 REM PICTURE IN THE MIDDLE OF THE PAPER
1290 IF(XE-XA)*YM)(YE-YA)*XM THEN GOTO 1330
1300 LET YS=YM/(YE-YA): LET YH=-YA*YS
1310 LET XS=YS: LET XH=(XM-(XE+XA)*XS)/2
1320 GOTO 1350
1330 LET XS=XM/(XE-XA): LET XH=-XA*XS
1340 LET YS=XS: LET YH=(YM-(YE+YA)*YS)/2
1350 LET I=0
1360 IF X(I,0)=-1 THEN GOTO 1410
1370 GOSUB 1430
1380 ON X(I,0)+1 GOSUB 45000,44000: REM MOVE OR LI
     NE
1390 LET I=I+1
1400 GOTO 1360
1410 GOSUB 40000
1420 END
1430 REM ORTHOGONAL PROJECTION ON THE PLANE (N1,N2
     ,N3)
1440 IF ABS(N0)(0.01 THEN GOTO 1480
1450 LET X=(X(I,1)*N2-X(I,2)*N1)/N0
1460 LET Y=(X(I,3)-(X(I,1)*N1+X(I,2)*N2+X(I,3)*N3)
     *N3)/N0
1470 GOTO 1510
1480 REM PROJECTION PARALLEL TO THE Z-AXIS
1490 LET X=X(I,1)
1500 LET Y=X(I,2)
1510 RETURN
62300 REM
62301 REM EDGES OF THE 3-DIMENSIONAL OBJECT
62302 REM
62303 REM THE FIRST THREE NUMBERS ARE THE X-,Y-
62304 REM AND Z-COORDINATES OF THE OBJECT EDGES.
62305 REM THE FOURTH NUMBER IS  1 WHEN THE EDGE IS
       LINKED BY A LINE WITH
62306 REM THE PREVIOUS EDGE, OTHERWISE THE NUMBER
      IS 0.
62307 REM FOR THE LAST ENTRY IN THE  TABLE THIS VA
     LUE IS -1
62308 REM
62309 REM
62310 DATA 2,2,0,0
62311 DATA 10,2,0,1
62312 DATA 10,10,0,1
62313 DATA 2,10,0,1
62314 DATA 2,2,0,1
62315 DATA 2,2,3,1
62316 DATA 10,2,3,1
62317 DATA 10,2,0,1
62318 DATA 10,2,3,0
62319 DATA 10,10,3,1
62320 DATA 10,10,0,1
62321 DATA 10,10,3,0
62322 DATA 2,10,3,1
62323 DATA 2,10,0,1
62324 DATA 2,10,3,0
62325 DATA 2,2,3,1
62326 DATA 3,3,3,0
62327 DATA 4,3,3,1
62328 DATA 4,4,3,1
62329 DATA 3,4,3,1
62330 DATA 3,3,3,1
62331 DATA 3,3,9,1
62332 DATA 4,3,9,1
62333 DATA 4,3,3,1
62334 DATA 4,3,9,0
62335 DATA 4,4,9,1
62336 DATA 4,4,3,1
62337 DATA 4,4,9,0
62338 DATA 3,4,9,1
62339 DATA 3,4,3,1
62340 DATA 3,4,9,0
62341 DATA 3,3,9,1
62342 DATA 8,3,3,0
62343 DATA 9,3,3,1
62344 DATA 9,4,3,1
62345 DATA 8,4,3,1
62346 DATA 8,3,3,1
62347 DATA 8,3,9,1
62348 DATA 9,3,9,1
62349 DATA 9,3,3,1
62350 DATA 9,3,9,0
62351 DATA 9,4,9,1
62352 DATA 9,4,3,1
62353 DATA 9,4,9,0
62354 DATA 8,4,9,1
62355 DATA 8,4,3,1
62356 DATA 8,4,9,0
62357 DATA 8,3,9,1
62358 DATA 8,8,3,0
62359 DATA 9,8,3,1
62360 DATA 9,9,3,1
62361 DATA 8,9,3,1
62362 DATA 8,8,3,1
62363 DATA 8,8,9,1
62364 DATA 9,8,9,1
62365 DATA 9,8,3,1
62366 DATA 9,8,9,0
62367 DATA 9,9,9,1
62368 DATA 9,9,3,1
62369 DATA 9,9,9,0
62370 DATA 8,9,9,1
62371 DATA 8,9,3,1
62372 DATA 8,9,9,0
```

```
62373 DATA 8,8,9,1
62374 DATA 3,8,3,0
62375 DATA 4,8,3,1
62376 DATA 4,9,3,1
62377 DATA 3,9,3,1
62378 DATA 3,8,3,1
62379 DATA 3,8,9,1
62380 DATA 4,8,9,1
62381 DATA 4,8,3,1
62382 DATA 4,8,9,0
62383 DATA 4,9,9,1
62384 DATA 4,9,3,1
62385 DATA 4,9,9,0
62386 DATA 3,9,9,1
62387 DATA 3,9,3,1
62388 DATA 3,9,9,0
62389 DATA 3,8,9,1
62390 DATA 0,0,9,0
62391 DATA 12,0,9,1
62392 DATA 12,12,9,1
62393 DATA 0,12,9,1
62394 DATA 0,0,9,1
62395 DATA 0,0,11,1
62396 DATA 12,0,11,1
62397 DATA 12,0,9,1
62398 DATA 12,0,11,0
62399 DATA 12,12,11,1
62400 DATA 12,12,9,1
62401 DATA 12,12,11,0
62402 DATA 0,12,11,1
62403 DATA 0,12,9,1
62404 DATA 0,12,11,0
62405 DATA 0,0,11,1
62406 DATA 0,0,0,-1
```

Prog. SCANNER

```
* 500 SYS INIT
  510 REM
  520 REM PLOTTER COMMANDS:
  530 REM 40000 HOME
  540 REM PLOTTER MOVEMENTS:
  550 REM 50000 +X-SCHRITT
  560 REM 51000 -X-SCHRITT
  570 REM 52000 +Y-SCHRITT
  580 REM 53000 -Y-SCHRITT
  590 REM 60000 LIMIT SWITCH?
  600 REM
* 610 PRINT CHR$(147)
  620 PRINT"FISCHERTECHNIK"
  630 PRINT"COMPUTING"
  640 PRINT
  650 PRINT"SCANNER-INITIALISATION"
  660 GOSUB40000 :REM HOME
  670 FOR Z=1 TO 4
  680 GOSUB 50000 : GOSUB 52000
  690 NEXT Z
 1000 REM FISCHERTECHNIK COMPUTING
 1010 REM
 1020 REM SCANNER
 1030 REM
 1040 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 198
      5
 1050 REM
 1060 REM FUNCTION:
 1070 REM PROGRAM RECORDS GREY TONES WITH A LDR
 1080 REM AND STORES THEM ON DISK.
 1090 REM DATA OF THE PICTURE
 1100 DIM G%(159,99)
* 1110 LET E=8 :REM FLOPPY DISK
* 1120 REM FOR CASSETTE: LET E=1 AND DELETE LINES 14
      70,
 1130 REM 1480, 1590 AND 1610-1670
 1140 LET GH=0:GL=255 :REM MIN AND MAX OF BRIGHTNES
      S
 1150 PRINT:PRINT"PUT THE PICTURE IN THE SCANNER"
 1160 PRINT"WITH THE LEFT BOTTOM CORNER UNDER THE
       HEAD OF THE SCANNER."
 1170 PRINT:PRINT"POSITION THE LDR VIA THE CURSOR K
      EYS"
 1180 PRINT"IN THE RIGHT TOP CORNER.
 1185 PRINT"WHEN READY PRESS >RETURN<"
 1190 GET S$
* 1200 IF S$=CHR$(29) AND XJ<640 THEN GOSUB 50000
* 1210 IF S$=CHR$(157) THEN GOSUB 51000
* 1220 IF S$=CHR$(17) THEN GOSUB 53000
* 1230 IF S$=CHR$(145) AND YJ<400 THEN GOSUB 52000
* 1240 IF S$=CHR$(13) THEN GOTO 1260
 1250 GOTO 1190
 1260 LET XE=INT(XJ/4)-1 :REM BORDER OF THE PICTURE
 1270 LET YE=INT(YJ/4)-1
 1280 PRINT:PRINT"CAUTION!  RECORDING"
* 1285 G%=USR(EY): REM DON'T USE THE FIRST VALUE
 1290 REM SCAN LOOP
 1300 FOR Y=0 TO YE
 1310 REM FORWARD
 1320 IF Y=INT(Y/2)*2 THEN GOSUB 2000
 1330 REM BACKWARD
 1340 IF Y>INT(Y/2)*2 THEN GOSUB 3000
 1350 REM GO TO NEXT LINE
 1360 FOR Z=1 TO 4
 1370 GOSUB 53000
 1380 NEXT Z
 1390 NEXT Y
* 1400 SYS INIT :REM SWITCH EVERYTHING OFF
 1410 REM SAVE
 1420 PRINT:PRINT"RECORDING FINISHED"
 1430 INPUT"FILENAME";F$
 1440 IF F$="" THEN END
* 1450 OPEN 15,8,15
* 1460 OPEN 1,E,2,F$+",W"
* 1470 INPUT#15,FF,FB$
* 1480 IF FF>0 THEN GOTO 1620
* 1490 PRINT#1,XE
* 1500 PRINT#1,YE
* 1510 PRINT#1,GH
* 1520 PRINT#1,GL
 1530 FOR Y=0 TO YE
 1540 FOR X=0 TO XE
* 1550 PRINT#1,G%(X,Y)
 1560 NEXT X
 1570 NEXT Y
* 1580 CLOSE 1
* 1590 CLOSE 15
 1600 END
 1610 REM ERROR MESSAGE
* 1620 CLOSE 1
* 1630 CLOSE 15
* 1640 PRINT"DISK ERROR: ";FB$
 1650 INPUT"NEW ATTEMPT (Y/N)";S$
 1660 IF S$="Y" THEN GOTO 1430
 1670 END
 2000 REM FORWARD
 2010 FOR X=XE TO 0 STEP -1
 2020 GOSUB 4000:REM MEASURE GREY TONE
 2030 FOR Z=1 TO 4
 2040 GOSUB 51000 :REM -X-STEP
 2050 NEXT Z
 2060 NEXT X
 2070 RETURN
 3000 REM BACKWARD
 3010 FOR X=0 TO XE
 3020 FOR Z=1 TO 4
 3030 GOSUB 50000 :REM +X-STEP
 3040 NEXT Z
 3050 GOSUB 4000:REM MEASURE GREY TONE
 3060 NEXT X
 3070 RETURN
 4000 REM MEASURE GREY TONE
* 4010 SYS INIT :REM MOTORS OFF
 4020 FOR Z=0 TO 20
* 4030 SYS M4,CW  :REM LAMP ON
 4040 REM WAIT FOR FULL BRIGHTNESS
 4050 NEXT Z
* 4060 G%=USR(EY)
* 4070 SYS M4,OFF :REM LAMP OFF
* 4080 IF G%=0 THEN LET G%=255 :REM ONLY C64,VC20,AC
      ORN
 4090 IF G%>255 THEN LET G%=255
 4100 IF G%>GH THEN LET GH=G%
 4110 IF G%<GL THEN LET GL=G%
 4120 G%(X,Y)=G%
```

```
 4130 RETURN
 40000 REM *** HOME ***
 40010 REM
 40030 LET XJ=0:LET YJ=0
* 40050 IF USR(E7)=1 THEN GOSUB 51050:GOTO 40050
* 40060 IF USR(E7)=0 THEN GOSUB 50050:GOTO 40060
* 40070 IF USR(E8)=1 THEN GOSUB 53050:GOTO 40070
* 40080 IF USR(E8)=0 THEN GOSUB 52050:GOTO 40080
 40100 LET XM=680:YM=500
 40110 LET XH=0:YH=0:XS=1:YS=1
 40120 LET XOUT=0:YOUT=0
 40130 RETURN
 40140 REM
 50000 REM *** +X-MOVEMENT ***
 50010 REM
 50020 LET XJ=XJ+1
 50040 IF XOUT THEN RETURN
* 50050 SYS M1,CC:SYS M2,CW:SYS M3,CW
 50060 GOSUB 60000:REM *** ET ??? ***
* 50070 SYS M1,CC:SYS M2,CC:SYS M3,CW
 50080 GOSUB 60000:REM *** ET ??? ***
* 50090 SYS M1,CW:SYS M2,CC:SYS M3,CW
 50100 GOSUB 60000:REM *** ET ??? ***
* 50110 SYS M1,CW:SYS M2,CW:SYS M3,CW
 50120 GOSUB 60000:REM *** ET ??? ***
 50130 RETURN
 50140 REM
 51000 REM *** -X-MOVEMENT ***
 51010 REM
 51020 LET XJ=XJ-1
 51040 IF XOUT THEN RETURN
* 51050 SYS M1,CW:SYS M2,CC:SYS M3,CW
 51060 GOSUB 60000:REM *** ET ??? ***
* 51070 SYS M1,CC:SYS M2,CC:SYS M3,CW
 51080 GOSUB 60000:REM *** ET ??? ***
* 51090 SYS M1,CC:SYS M2,CW:SYS M3,CW
 51100 GOSUB 60000:REM *** ET ??? ***
* 51110 SYS M1,CW:SYS M2,CW:SYS M3,CW
 51120 GOSUB 60000:REM *** ET ??? ***
 51130 RETURN
 51140 REM
 52000 REM *** +Y-MOVEMENT ***
 52010 REM
 52020 LET YJ=YJ+1
 52040 IF YOUT THEN RETURN
* 52050 SYS M1,CC:SYS M2,CW:SYS M3,CW
 52060 GOSUB 60000:REM *** ET ??? ***
* 52070 SYS M1,CC:SYS M2,CW:SYS M3,CC
 52080 GOSUB 60000:REM *** ET ??? ***
* 52090 SYS M1,CW:SYS M2,CW:SYS M3,CC
 52100 GOSUB 60000:REM *** ET ??? ***
* 52110 SYS M1,CW:SYS M2,CW:SYS M3,CW
 52120 GOSUB 60000:REM *** ET ??? ***
 52130 RETURN
 52140 REM
 53000 REM *** -Y-MOVEMENT ***
 53010 REM
 53020 LET YJ=YJ-1
 53040 IF YOUT THEN RETURN
* 53050 SYS M1,CW:SYS M2,CW:SYS M3,CC
 53060 GOSUB 60000:REM *** ET ??? ***
* 53070 SYS M1,CC:SYS M2,CW:SYS M3,CC
 53080 GOSUB 60000:REM *** ET ??? ***
* 53090 SYS M1,CC:SYS M2,CW:SYS M3,CW
 53100 GOSUB 60000:REM *** ET ??? ***
* 53110 SYS M1,CW:SYS M2,CW:SYS M3,CW
 53120 GOSUB 60000:REM *** ET ??? ***
 53130 RETURN
 53140 REM
 60000 REM *** LIMIT SWITCH CLOSED ***
 60010 REM
* 60020 IF (USR(E7)=0 AND XJ<>0) OR (USR(E8)=0 AND Y
       J<>0) THEN GOTO 60040
 60030 RETURN
* 60040 PRINT CHR$(147);CHR$(18)"CAUTION! SCANNER IS
       N'T ADJUSTED";CHR$(146)
 60050 STOP
```

Prog. B & W

```
 1000 REM FISCHERTECHNIK COMPUTING
 1010 REM
 1020 REM PICTURE EVALUATION
 1030 REM PROGRAM B&W
 1040 REM
* 1050 REM THIS PROGRAM NEEDS
* 1060 REM >SIMONS BASIC< !
 1070 REM
 1080 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 198
      5
 1090 REM
 1100 REM FUNCTION DESCRIPTION:
 1110 REM THE PROGRAM SEPARATES THE GREY TONES
 1120 REM WHICH WERE RECORDED BY THE PROGRAM 'SCANN
      ER'.
 1130 REM YOU CAN DEFINE THE THRESHOLD BETWEEN
 1140 REM BLACK AND WHITE.
 1150 REM THE DATA IS READ FROM DISK.
* 1160 LET E=8 :REM FLOPPY DISK
* 1170 REM FOR CASSETTE: LET E=1 AND DELETE THE LINE
       NUMBERS 1230,
* 1180 REM 1270, 1290, 1300 AND 1630-1670
* 1190 PRINT CHR$(147)
 1200 PRINT"FISCHERTECHNIK"
 1210 PRINT"COMPUTING"
* 1215 PRINT:PRINT"PROGRAM NEEDS 'SIMONS BASIC'"
 1220 PRINT
* 1230 DIR"$":*=S" :REM DIRECTORY
 1240 PRINT:PRINT
 1250 INPUT "FILENAME ";F$
 1260 IF F$="" THEN END
* 1270 OPEN 15,8,15
* 1280 OPEN 1,E,2,F$+",R"
* 1290 INPUT#15,FF,FB$
* 1300 IF FF>0 THEN GOTO 1640
* 1310 INPUT#1,XE :REM BORDER OF THE PICTURE
* 1320 INPUT#1,YE
* 1330 INPUT#1,GH :REM MIN AND MAX OF BRIGHTNESS
* 1340 INPUT#1,GL
 1350 REM ASK FOR THRESHOLD
 1360 PRINT:PRINT"THRESHOLD BETWEEN ";GL;" AND ";GH
 1370 INPUT GW
 1380 IF GW<GL OR GW>GH THEN GOTO 1360
 1390 PRINT:PRINT"WHEN THE PICTURE IS DRAWN YOU CAN
 1400 PRINT"LEAVE THE PROGRAM HITTING ANY KEY !"
 1410 FOR Z=0 TO 1000
 1420 REM DELAY LOOP
 1430 NEXT Z
 1440 REM SWITCH ON HIRES GRAPHIC
* 1450 HIRES 0,1
 1460 FOR Y=0 TO YE
 1470 FOR X=0 TO XE
 1480 LET X1=2*X
 1490 LET X2=X1+1
 1500 LET Y1=199-2*YE+2*Y-2
 1510 LET Y2=Y1+1
* 1520 INPUT#1,G%
 1530 REM DRAW BLOCK
* 1540 IF G%>=GW THEN BLOCK X1,Y1,X2,Y2,1
 1550 NEXT X
 1560 NEXT Y
* 1570 CLOSE 1
 1580 REM SCAN KEYBOARD FOR END OF PROGRAM
* 1590 GET S$
 1600 IF S$="" THEN GOTO 1590
* 1610 PRINT CHR$(147)
 1620 END
 1630 REM ERROR MESSAGE
* 1640 PRINT"DISK ERROR: "FB$
* 1650 CLOSE 1
* 1660 CLOSE 15
 1670 END
```

Prog. D.PIC

```
1000 REM FISCHERTECHNIK COMPUTING
1010 REM
1020 REM PICTURE EVALUATION
1030 REM PROGRAM D.PIC
1040 REM
* 1050 REM THIS PROGRAM NEEDS
* 1060 REM >SIMONS BASIC< !
1070 REM
1080 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 198
     5
1090 REM
1100 REM FUNCTION
1110 REM THE PROGRAM SEPARATES THE GREY TONES
1120 REM WHICH WERE RECORDED BY THE PROGRAM 'SCAN
     NER'.
1130 REM YOU CAN DEFINE THE THRESHOLD BETWEEN
1140 REM BLACK AND WHITE AND THE GRAPHIC RESOLUTIO
     N.
1150 REM THE SIZE OF THE PICTURE CAN BE ADJUSTED,
     SO
1160 REM THAT IT FITS ON THE WHOLE SCREEN.
1170 REM THE DATA IS READ FROM DISK.
1180 REM DATA OF THE PICTURE
1190 DIM W%(159,3)
* 1200 LET E=8 :REM FLOPPY DISK
* 1210 REM FOR CASSETTE: LET E=1 AND DELETE THE LINE
      NUMBERS 1270,
* 1220 REM 1310, 1330, 1340, 1830 UND 1890-1930 .
* 1230 PRINT CHR$(147)
1240 PRINT"FISCHERTECHNIK"
1250 PRINT"COMPUTING"
1260 PRINT
* 1265 PRINT"PROGRAM NEEDS 'SIMONS BASIC'"
1267 PRINT
* 1270 DIR"$:*=S" :REM DIRECTORY
1280 PRINT:PRINT
1290 INPUT "FILENAME";F$
1300 IF F$="" THEN END
* 1310 OPEN 15,8,15
* 1320 OPEN 1,E,2,F$+",R"
* 1330 INPUT#15,FF,FB$
* 1340 IF FF>0 THEN GOTO 1900
* 1350 INPUT#1,XE :REM BORDER OF THE PICTURE
* 1360 INPUT#1,YE
* 1370 INPUT#1,GH :REM MIN AND MAX OF BRIGHTNESS
* 1380 INPUT#1,GL
1390 REM ASK FOR THRESHOLD
1400 PRINT:PRINT"THRESHOLD BETWEEN ";GL;" AND ";GH
1410 INPUT GW
1420 IF GW<GL OR GW>GH THEN GOTO 1400
1430 PRINT:INPUT"RESOLUTION X-AXIS (40/80/160)";AX
1440 IF AX<>40 AND AX<>80 AND AX<>160 THEN GOTO 14
     30
1450 PRINT:INPUT"RESOLUTION Y-AXIS (25/50/100)";AY
1460 IF AY<>25 AND AY<>50 AND AY<>100 THEN GOTO 14
     50
1470 LET AX=160/AX:AY=100/AY
1480 PRINT:INPUT"OPTIMIZE PICTURE (Y/N)";OPT$
1490 PRINT:PRINT"WHEN THE PICTURE IS DRAWN YOU CAN
     "
1500 PRINT"LEAVE THE PROGRAM HITTING ANY KEY!"
1510 FOR Z=0 TO 1000
1520 REM DELAY LOOP
1530 NEXT Z
1540 IF OPT$="N" THEN LET R=2:GOTO 1600
1550 LET RX=INT(160/XE)
1560 LET RY=INT(100/YE)
1570 IF RX>RY THEN LET R=2*RY:GOTO 1600
1580 LET R=2*RX
1590 REM SWITCH ON HIRES GRAPHIC
* 1600 HIRES 0,1
1610 FOR Y=0 TO YE STEP AY
1620 FOR Z=0 TO AY-1
1630 FOR X=0 TO XE
* 1640 INPUT#1,W%(X,Z)
1650 NEXT X
1660 NEXT Z
1670 FOR X=0 TO XE STEP AX
1680 LET G%=0
1690 FOR Z=0 TO AY-1
1700 FOR U=0 TO AX-1
1710 G%=G%+W%(X+U,Z)
1720 NEXT U
1730 NEXT Z
1740 LET X1=R*X
1750 LET X2=X1+R*AX-1
1760 LET Y1=199-R*YE+R*Y-R*AY
1770 LET Y2=Y1+R*AY-1
1780 REM DRAW BLOCK
* 1790 IF G%/(AX*AY))=GW THEN BLOCK X1,Y1,X2,Y2,1
1800 NEXT X
1810 NEXT Y
* 1820 CLOSE 1
* 1830 CLOSE 15
1840 REM SCAN KEYBOARD FOR END OF PROGRAM
* 1850 GET S$
1860 IF S$="" THEN GOTO 1850
* 1870 PRINT CHR$(147)
1880 END
1890 REM ERROR MESSAGE
* 1900 PRINT"DISK ERROR: "FB$
* 1910 CLOSE 1
* 1920 CLOSE 15
1930 END
```

Prog. COLOR

```
1000 REM FISCHERTECHNIK COMPUTING
1010 REM
1020 REM PICTURE EVALUATION
1030 REM PROGRAM COLOUR
1040 REM
* 1050 REM THIS PROGRAM NEEDS
* 1060 REM >SIMONS BASIC< !
1070 REM
1080 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 198
     5
1090 REM
1100 REM FUNCTION
1110 REM THE PROGRAM SEPARATES THE GREY TONES
1120 REM WHICH WERE RECORDED BY THE PROGRAM 'SCAN
     NER'.
1130 REM YOU CAN DEFINE THE THRESHOLD BETWEEN
1140 REM DIFFERENT COLOURS AND THE GRAPHIC RESOLU
     TION.
1150 REM THE SIZE OF THE PICTURE CAN BE ADJUSTED,
     SO
1160 REM THAT IT FITS ON THE WHOLE SCREEN.
1170 REM THE GREY TONES ARE DISPLAYED AS COLOURS
     .
1180 REM THE DATA IS READ FROM DISK
1190 REM DATA OF THE PICTURE
1200 DIM W%(159,3)
* 1210 LET E=8 :REM FLOPPY DISK
* 1220 REM FOR CASSETTE: LET E=1 AND DELETE LINE NUM
     BER 1280,
* 1230 REM 1320, 1340, 1350, 2070 UND 2130-2170
* 1240 PRINT CHR$(147)
1250 PRINT"FISCHERTECHNIK"
1260 PRINT"COMPUTING"
* 1265 PRINT:PRINT"PROGRAM NEEDS 'SIMONS BASIC'"
1270 PRINT
* 1280 DIR"$:*=S" :REM DIRECTORY
1290 PRINT:PRINT
1300 INPUT "FILENAME";F$
1310 IF F$="" THEN END
* 1320 OPEN 15,8,15
* 1330 OPEN 1,E,2,F$+",R"
* 1340 INPUT#15,FF,FB$
* 1350 IF FF>0 THEN GOTO 2140
* 1360 INPUT#1,XE :REM BORDER OF THE PICTURE
* 1370 INPUT#1,YE
* 1380 INPUT#1,GH :REM MIN AND MAX OF BRIGHTNESS
* 1390 INPUT#1,GL
1400 REM ASK FOR THRESHOLD
* 1410 PRINT CHR$(147):INPUT"HOW MANY COLOURS (2-4)"
     ;FA
1420 IF FA<2 OR FA>4 THEN GOTO 1410
1430 PRINT:PRINT"THRESHOLD BETWEEN ";GL;" AND ";GH
1440 REM DISPLAY ALL AVAILABLE COLOURS AT THE BO
     TTOM OF THE SCREEN
```

```
1450 FOR F=0 TO 15
1460 LET C1=160:C2=176+F
* 1470 IF F>9 THEN LET C1=177:C2=166+F
* 1480 POKE 1984+2*F,C1
* 1490 POKE 1985+2*F,C2
* 1500 POKE 56256+2*F,F
* 1510 POKE 56257+2*F,F
1520 NEXT F
1530 INPUT"BACKGROUND COLOUR";FR(0)
1540 FOR F=1 TO FA-1
1550 INPUT "THRESHOLD";GW(F)
1560 IF GW(F)<GL OR GW(F)>GH THEN GOTO 1550
1570 INPUT"CORRESPONDING COLOUR";FR(F)
1580 NEXT F
1590 FOR F=0 TO 31
* 1600 POKE 1984+F,32
1610 NEXT F
1620 PRINT:INPUT"RESOLUTION X-AXIS (40/80/160)";AX
1630 IF AX<>40 AND AX<>80 AND AX<>160 THEN GOTO 16
     20
1640 PRINT:INPUT"RESOLUTION Y-AXIS (25/50/100)";AY
1650 IF AY<>25 AND AY<>50 AND AY<>100 THEN GOTO 16
     40
1660 LET AX=160/AX:AY=100/AY
1670 PRINT:INPUT"OPTIMIZE PICTURE (Y/N)";OPT$
1680 PRINT:PRINT"WHEN THE PICTURE IS DRAWN YOU CAN
     "
1690 PRINT"LEAVE THE PROGRAM HITTING ANY KEY!"
1700 FOR Z=0 TO 1000
1710 REM DELAY LOOP
1720 NEXT Z
1730 IF OPT$="N" THEN LET R=2:GOTO 1790
1740 LET RX=INT(160/XE)
1750 LET RY=INT(100/YE)
1760 IF RX>RY THEN LET R=2*RY:GOTO 1790
1770 LET R=2*RX
1780 REM SWITCH ON HIRES GRAPHIC AND DEFINE BACKGR
     OUND COLOURS
* 1790 POKE 53281,FR(0)
* 1800 POKE 53280,FR(0)
* 1810 HIRES 0,FR(0)
1820 MULTI FR(1),FR(2),FR(3)
1830 FOR Y=0 TO YE STEP AY
1840 FOR Z=0 TO AY-1
1850 FOR X=0 TO XE
* 1860 INPUT#1,WX%(X,Z)
1870 NEXT X
1880 NEXT Z
1890 FOR X=0 TO XE STEP AX
1900 LET G%=0
1910 FOR Z=0 TO AY-1
1920 FOR U=0 TO AX-1
1930 G%=G%+WX%(X+U,Z)
1940 NEXT U
1950 NEXT Z
1960 LET X1=R*X/2
1970 LET X2=X1+R*AX/2-1
1980 LET Y1=199-R*YE+R*Y-R*AY
1990 LET Y2=Y1+R*AY-1
2000 REM DRAW BLOCK
2010 FOR F=1 TO FA-1
* 2020 IF G%/(AX*AY)>=GW(F) THEN BLOCK X1,Y1,X2,Y2,
     F
2030 NEXT F
2040 NEXT X
2050 NEXT Y
* 2060 CLOSE 1
* 2070 CLOSE 15
2080 REM SCAN KEYBOARD FOR END OF PROGRAM
* 2090 GET S$
2100 IF S$="" THEN GOTO 2090
* 2110 PRINT CHR$(147)
2120 END
2130 REM ERROR MESSAGE
* 2140 PRINT"DISK ERROR: "FB$
* 2150 CLOSE 1
* 2160 CLOSE 15
2170 END
```

Prog. PATTERN

```
* 500 SYS INIT
510 REM PLOTTER COMMANDS:
520 REM 40000 HOME
530 REM PLOTTER MOVEMENTS:
540 REM 50000 +X-STEP
550 REM 51000 -X-STEP
560 REM 52000 +Y-STEP
570 REM 53000 -Y-STEP
580 REM 54000 +X/+Y-STEP
590 REM 55000 +X/-Y-STEP
600 REM 56000 -X/+Y-STEP
610 REM 57000 -X/-Y-STEP
620 REM 60000 LIMIT SWITCH ?
630 REM 61000 OUT OF RANGE ?
640 PRINT CHR$(147)+CHR$(5)
650 POKE 53280,0: REM WHITE LETTERS ON
660 POKE 53281,0: REM BLACK BACKGROUND
670 PRINT"FISCHERTECHNIK"
680 PRINT"COMPUTING"
690 PRINT
700 PRINT">>> SCANNER INITIALISATION"
710 GOSUB40000 :REM HOME
720 REM
1000 REM FISCHERTECHNIK COMPUTING
1010 REM
1020 REM PROGRAM PATTERN
1030 REM
1040 REM COPYRIGHT (C) ARTUR FISCHER FORSCHUNG 198
     5
1050 REM
1060 REM FUNCTION DESCRIPTION:
1070 REM PATTERN RECOGNITION
1080 REM IN THE FIRST RUN THE SCANNER SCANS
1090 REM A CIRCLE (BLACK ON WHITE ).
1100 REM IN FURTHER RUNS THE PROGRAM CAN RECOGNIZE
1110 REM IDENTICAL, SHIFTED AND BIGGER OR SMALLER
     CIRCLES.
1120 REM
1130 DIM X%(1000),Y%(1000)
1140 REM MEASURE REFERENCE CIRCLE
1150 PRINT"PLEASE PUT REFERENCE CIRCLE IN
      THE SCANNER"
1160 LET RF=1 :REM REFERENCE CIRCLE FLAG
1170 PRINT">RETURN<, WHEN YOU ARE READY."
* 1180 GET A$
1190 IF A$<>CHR$(13) THEN GOTO 1180
1200 GOSUB 2000: REM PATTERN REGOGNITION
1210 IF RF=0 THEN GOTO 1290
1220 LET XR=XC
1230 LET YR=YC
1240 LET RR=RM
1250 LET RF=0
1260 GOTO 1510
1270 REM ANALYSIS DATA
1280 REM POSITION OF THE CIRCLE
1290 LET L=1: REM FLAG-IDENTICAL CENTRE
1300 IF XC>XR+2 THEN LET L=0
1310 IF XC<XR-2 THEN LET L=0
1320 IF YC>YR+2 THEN LET L=0
1330 IF YC<YR-2 THEN LET L=0
1340 LET R=1: REM FLAG-IDENTICAL RADIUS
1350 IF RM<RR-1 THEN LET R=0
1355 IF RM>RR+1 THEN LET R=0
1360 PRINT">>> ANALYSIS OF THE DATA :"
1370 IF R=0 OR L=0 THEN GOTO 1400
1380 PRINT"   IDENTICAL CIRCLE !"
1390 GOTO 1510
1400 IF L=0 THEN GOTO 1440
1410 PRINT"   SAME POSITION, SIZE IN THE"
1420 PRINT"   PROPORTION  ";RM/RR
1430 GOTO 1510
1440 IF R=0 THEN GOTO 1480
1450 PRINT"   SAME CIRCLE, BUT SHIFTED AS FOLLOWS:
     "
1460 PRINT"   X=";XC-XR;"  Y=";YC-YR
1470 GOTO 1510
1480 PRINT"   CIRCLE SHIFTED AND DIFFERENT SIZE"
1490 PRINT"   SIZE PROPORTION    :";RM/RR
1500 PRINT"   X=";XC-XR;"  Y=";YC-YR
1510 GOSUB 40000
```

```
1520 PRINT "READY FOR THE NEXT FIGUR.  >RETURN<"
* 1530 GET A$
1540 IF A$<>CHR$(13) THEN GOTO 1530
* 1550 PRINT CHR$(147);"PLEASE PUT CIRCLE IN SCANNER
     !"
1560 GOTO 1170
1900 REM SUBROUTINE PATTERN RECOGITION
1910 REM THE SCANNER FOLLOWS THE BORDERLINE OF THE
      FIGURE
1920 REM AND CALCULATES THE CENTRE OF THE CIRCLE
1930 REM (APPROXIMATE CENTRE OF GRAVITY OF THE PAT
     H)
1940 REM THE RADIUS IS CALCULATED FROM THE DISTANC
     ES
1950 REM TO THE CENTRE.
2000 PRINT">> SEARCH LINE BY LINE <<"
2010 LET GL=255:GH=0: REM MIN AND MAX OF BRIGHTNES
     S
2020 LET GE=10: REM THRESHOLD FOR BLACK
2030 GOSUB 4000: REM FORWARD
2040 IF GF THEN GOTO 2110
2050 GOSUB 6000: REM NEXT LINE
2060 GOSUB 5000: REM BACKWARD
2070 IF GF THEN GOTO 2110
2080 GOSUB 6000: REM NEXT LINE
2090 GOTO 2030
2100 REM
2110 PRINT">> FIGUR IS FOUND!":PRINT
2120 LET Z=-1: REM COUNTER FOR POINTS OF THE PATH
2130 GOSUB 9000: REM STEPPERALGORITHM
2140 LET Z=Z+1: REM INCREMENT COUNTER
2150 LET X%(Z)=XJ/4: REM SAVE PATH
2160 LET Y%(Z)=YJ/4
2170 REM CHECK IF END OF PATH
2180 IF Z<20 THEN GOTO 2130
2190 IF X%(Z)>X%(0)+2 THEN GOTO 2130
2200 IF X%(Z)<X%(0)-2 THEN GOTO 2130
2210 IF Y%(Z)>Y%(0)+2 THEN GOTO 2130
2220 IF Y%(Z)<Y%(0)-2 THEN GOTO 2130
2230 PRINT:PRINT:PRINT">> FIGUR IS SCANNED!"
2240 PRINT">> EVALUATION ACTIV..."
2250 LET ZMAX=Z
2260 IF ZMAX<10 THEN GOTO 3000: REM ERROR MESSAGE
2270 REM CALCULATE CENTRE OF CIRCLE
2280 PRINT">> CALCULATE CENTRE OF THE CIRCLE <<"
2290 LET XC=0:YC=0
2300 FOR Z=0 TO ZMAX
2310 LET XC=XC+X%(Z)
2320 LET YC=YC+Y%(Z)
2330 NEXT Z
2340 LET XC=XC/(ZMAX+1): REM AVERAGE
2350 LET YC=YC/(ZMAX+1): REM AVERAGE
2360 REM CALCULATE RADIUS OF THE CIRCLE
2370 PRINT">> CALCULATE RADIUS OF THE CIRCLE <<"
2380 LET RM=0
2390 FOR Z=0 TO ZMAX
2400 LET RM=RM+SQR((XC-X%(Z))↑2+(YC-Y%(Z))↑2)
2410 NEXT Z
2420 LET RM=RM/(ZMAX+1): REM AVERAGE
2430 REM FINAL RESULTS
2440 PRINT">> FINAL RESULTS:"
2450 PRINT"   CENTRE (X,Y): ";
2460 PRINT XC,YC
2470 PRINT"   RADIUS R: ";RM
2480 RETURN
3000 REM ERROR MESSAGE
3010 PRINT"NOT ENOUGH POINTS OF THE PATH! "
3020 END
4000 REM FORWARD
4010 LET S=2: REM DIRECTION EAST
4020 FOR X=0 TO XM/4-1
4030 FOR Z=1 TO 4
4040 GOSUB 50000: REM +X
4050 NEXT Z
4060 GOSUB 7000: REM MEASURE GREY VALUE
4070 IF G<GL THEN GL=G: REM ADJUST MIN
4080 LET GF=(G)GL+GE): REM THRESHOLD
4090 IF GF THEN RETURN
4100 NEXT X
4110 RETURN
4120 REM
5000 REM BACKWARD
5010 LET S=6: REM DIRECTION WEST
5020 FOR X=XM/4-1 TO 0 STEP -1
5030 FOR Z=1 TO 4
5040 GOSUB 51000: REM -X
5050 NEXT Z
5060 GOSUB 7000: REM MEASURE GREY TONE
5070 IF G<GL THEN GL=G: REM ADJUST MIN
5080 LET GF=(G)GL+GE): REM THRESHOLDD
5090 IF GF THEN RETURN
5100 NEXT X
5110 RETURN
5120 REM
6000 REM ONE LINE UP
6010 FOR Z=1 TO 4
6020 GOSUB 52000: REM +Y
6030 NEXT Z
6040 RETURN
6050 REM
7000 REM MEASURE GREY TONE
* 7010 SYS INIT: REM ALL MOTORS OFF
7020 FOR T=0 TO 20
* 7030 SYS M4,CW: REM LAMPE ON, WAIT
7040 NEXT T
* 7050 G=USR(EY)
* 7060 SYS M4,OFF
7070 RETURN
7080 REM
8000 REM SUBROUTINE TO FOLLOW THE BORDERLINE
8010 REM
8020 REM THIS SUBROUTINE DRIVES THE SCANNER
8030 REM ALONG THE BORDERLINE BY FOLLOWING ALGORIT
     HM
8040 REM
8050 REM WHITE POINT -> TURN 45 DEGREES WEST, MOVE
      ONE STEP FORWARD
8080 REM BLACK POINT -> TURN 45 DEGREES EAST, MOVE
      ONE STEP TO THE RIGHT
8090 REM
8120 REM SOURCE:
8130 REM JOHN BILLINGSLEY
8140 REM ROBOTICS AND SENSORS ON THE COMMODORE COM
     PUTER.
8150 REM SUNSHINE BOOKS LONDON
8160 REM
9000 REM ALGORITHM - FOLLOW BORDERLINE
9010 GOSUB 7000: REM MEASURE GREY VALUE
9020 IF G<GL THEN LET GL=G: REM ADJUST MIN
9030 IF G>GH THEN LET GH=G: REM ADJUST MAX
9040 GS=(GL+GH)*0.5: REM TRESHOLD = AVERAGE
9050 IF G>GS THEN LET R=R+1:S=R+2:L=0:H=H+1
9060 IF G<=GS THEN LET R=R-1:S=R:H=0:L=L+1
9070 REM DETERMINE IF SCANNER STILL PROCEEDS.
9080 IF L>15 THEN GOTO 9220
9090 IF H>15 THEN GOTO 9220
9100 REM DIRECTIONS IN THE RANGE 0 - 7
9110 IF S>7 THEN LET S=S-8
9120 IF S<0 THEN LET S=S+8
9130 IF R>7 THEN LET R=R-8
9140 IF R<0 THEN LET R=R+8
9150 IF G>GS THEN PRINT" ";
* 9160 IF G<=GS THEN PRINTCHR$(18);" ";CHR$(146);
9170 REM 4 STEPS IN DIRECTION S
9180 FOR T=1 TO 4
9190 ON S+1 GOSUB 52000,54000,50000,55000,53000,57
     000,51000,56000
9200 NEXT T
9210 RETURN
9220 REM LOST BORDERLINE
9230 PRINT"SCANNER LOST BORDERLINE !"
9240 END
40000 REM *** HOME ***
40010 REM
40030 LET XJ=0:LET YJ=0
* 40050 IF USR(E7)=1 THEN GOSUB 51050:GOTO 40050
* 40060 IF USR(E7)=0 THEN GOSUB 50050:GOTO 40060
* 40070 IF USR(E8)=1 THEN GOSUB 53050:GOTO 40070
* 40080 IF USR(E8)=0 THEN GOSUB 52050:GOTO 40080
40100 LET XM=680:YM=500
40110 LET XH=0:YH=0:XS=1:YS=1
40120 LET XOUT=0:YOUT=0
40130 RETURN
40140 REM
50000 REM *** +X-MOVEMENT ***
50010 REM
50020 LET XJ=XJ+1
* 50050 SYS M1,CC:SYS M2,CW:SYS M3,CW
50060 GOSUB 60000:REM *** ET ??? ***
* 50070 SYS M1,CC:SYS M2,CC:SYS M3,CW
50080 GOSUB 60000:REM *** ET ??? ***
* 50090 SYS M1,CW:SYS M2,CC:SYS M3,CW
50100 GOSUB 60000:REM *** ET ??? ***
* 50110 SYS M1,CW:SYS M2,CW:SYS M3,CW
50120 GOSUB 60000:REM *** ET ??? ***
```

```
 50130 RETURN
 50140 REM
 51000 REM *** -X-MOVEMENT ***
 51010 REM
 51020 LET XJ=XJ-1
* 51050 SYS M1,CW:SYS M2,CC:SYS M3,CW
 51060 GOSUB 60000:REM *** ET ??? ***
* 51070 SYS M1,CC:SYS M2,CC:SYS M3,CW
 51080 GOSUB 60000:REM *** ET ??? ***
* 51090 SYS M1,CC:SYS M2,CW:SYS M3,CW
 51100 GOSUB 60000:REM *** ET ??? ***
* 51110 SYS M1,CW:SYS M2,CW:SYS M3,CW
 51120 GOSUB 60000:REM *** ET ??? ***
 51130 RETURN
 51140 REM
 52000 REM *** +Y-MOVEMENT ***
 52010 REM
 52020 LET YJ=YJ+1
* 52050 SYS M1,CC:SYS M2,CW:SYS M3,CW
 52060 GOSUB 60000:REM *** ET ??? ***
* 52070 SYS M1,CC:SYS M2,CW:SYS M3,CC
 52080 GOSUB 60000:REM *** ET ??? ***
* 52090 SYS M1,CW:SYS M2,CW:SYS M3,CC
 52100 GOSUB 60000:REM *** ET ??? ***
* 52110 SYS M1,CW:SYS M2,CW:SYS M3,CW
 52120 GOSUB 60000:REM *** ET ??? ***
 52130 RETURN
 52140 REM
 53000 REM *** -Y-MOVEMENT ***
 53010 REM
 53020 LET YJ=YJ-1
* 53050 SYS M1,CW:SYS M2,CW:SYS M3,CC
 53060 GOSUB 60000:REM *** ET ??? ***
* 53070 SYS M1,CC:SYS M2,CW:SYS M3,CC
 53080 GOSUB 60000:REM *** ET ??? ***
* 53090 SYS M1,CC:SYS M2,CW:SYS M3,CW
 53100 GOSUB 60000:REM *** ET ??? ***
* 53110 SYS M1,CW:SYS M2,CW:SYS M3,CW
 53120 GOSUB 60000:REM *** ET ??? ***
 53130 RETURN
 53140 REM
 54000 REM *** +X/+Y-DIAGONAL ***
 54010 XJ=XJ+1:YJ=YJ+1
* 54050 SYSM1,CC:SYSM2,CW:SYSM3,CW
 54060 GOSUB60000:REM *** ET ??? ***
* 54070 SYSM1,CC:SYSM2,CC:SYSM3,CC
 54080 GOSUB60000:REM *** ET ??? ***
* 54090 SYSM1,CW:SYSM2,CC:SYSM3,CC
 54100 GOSUB60000:REM *** ET ??? ***
* 54110 SYSM1,CW:SYSM2,CW:SYSM3,CW
 54120 RETURN
 55000 REM *** +X/-Y-DIAGONAL ***
 55010 XJ=XJ+1:YJ=YJ-1
* 55050 SYSM1,CW:SYSM2,CW:SYSM3,CC
 55060 GOSUB60000:REM *** ET ??? ***
* 55070 SYSM1,CC:SYSM2,CW:SYSM3,CC
 55080 GOSUB60000:REM *** ET ??? ***
* 55090 SYSM1,CC:SYSM2,CC:SYSM3,CW
 55100 GOSUB60000:REM *** ET ??? ***
* 55110 SYSM1,CW:SYSM2,CC:SYSM3,CW
 55120 GOSUB60000:REM *** ET ??? ***
* 55130 SYSM1,CW:SYSM2,CW:SYSM3,CW
 55140 RETURN
 56000 REM *** -X/+Y-DIAGONAL ***
 56010 XJ=XJ-1:YJ=YJ+1
* 56050 SYSM1,CW:SYSM2,CC:SYSM3,CW
 56060 GOSUB60000:REM *** ET ??? ***
* 56070 SYSM1,CC:SYSM2,CC:SYSM3,CW
 56080 GOSUB60000:REM *** ET ??? ***
* 56090 SYSM1,CC:SYSM2,CW:SYSM3,CC
 56100 GOSUB60000:REM *** ET ??? ***
* 56110 SYSM1,CW:SYSM2,CW:SYSM3,CC
 56120 GOSUB60000:REM *** ET ??? ***
* 56130 SYSM1,CW:SYSM2,CW:SYSM3,CW
 56140 RETURN
 57000 REM *** -X/-Y-DIAGONAL ***
 57010 XJ=XJ-1:YJ=YJ-1
* 57050 SYSM1,CW:SYSM2,CC:SYSM3,CC
 57060 GOSUB60000:REM *** ET ??? ***
* 57070 SYSM1,CC:SYSM2,CC:SYSM3,CC
 57080 GOSUB60000:REM *** ET ??? ***
* 57090 SYSM1,CC:SYSM2,CW:SYSM3,CW
 57100 GOSUB60000:REM *** ET ??? ***
* 57110 SYSM1,CW:SYSM2,CW:SYSM3,CW
 57120 RETURN
 60000 REM *** FINAL SWITCH CLOSED ***
* 60010 IF(USR(E7)=0ANDXJ<>0)OR(USR(E8)=0ANDYJ<>0)TH
      ENGOTO60030
 60020 RETURN
* 60030 PRINT CHR$(147);CHR$(18)"WARNING PLOTTER ISN
       'T ADJUSTED";CHR$(146)
 60040 STOP
```


39462 11.86 ZE