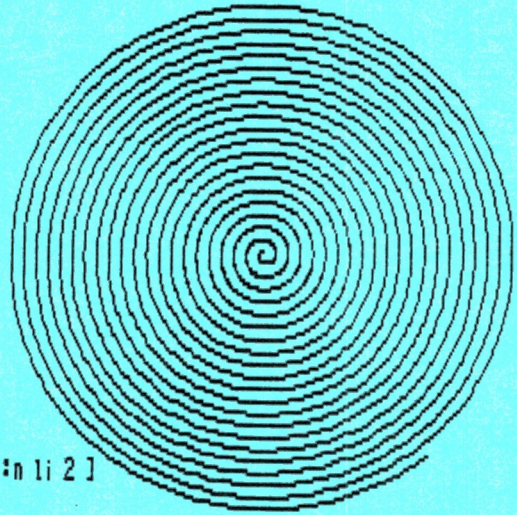
Der NDR-Klein-Computer

GOSI Einführung



```
lerne spiro :n
wh 45 [ schr16tel :n li 2 ]
spiro :n+1
ende

Ok gelernt, Platz zum lernen: 2438 Bytes und fuer Namen: 662 Bytes.
spiro 1
```

Franzis' SoftwareService

Vorwort

GOSI ist eine leicht zu erlernende graphische Sprache, die ähnlich wie LOGO Sprachelemente für einen Schreibzeiger besitzt, den man mit Befehlen bewegen kann.
GOSI besitzt Befehlselemente, die über die Befehle z.B. von BASIC weit hinausgehen, so ist es möglich der Sprache neue Wörter beizubringen und sie von da an zu verwenden. Dadurch wird die Bedienung sehr erleichtert und Programme werden anschaulich und lesbar.
In GOSI ist nicht nur eine Einsteigersprache, die leicht zu erlernen ist, sondern man kann mit ihr auch Steuerungsaufgaben verwirklichen. So kann man sich z.B. Wörter für Ventilan, Ventilaus etc. selbst definieren und dann damit arbeiten. Die dazu nötigen Grund-Befehle, die man braucht um mit der Aussenwelt zu arbeiten, sind in GOSI natürlich vorhanden.
Ferner gibt es Befehle um einen Drucker anzusteuern und man kann seine Programme und Daten auf einem Kassettenrekorder speichern und laden.

München, den 28.3.1984

Rolf-Dieter Klein

Inbetriebnahme von GOSI

GOSI wird in zwei EPROMs geliefert, das sind zwei Bausteine in denen die Sprache programmiert ist. Diese beiden Bausteine werden auf die SBC-II-Karte gesteckt, und zwar das EPROM mit der Beschriftung 0 in den Sockel 0 und das EPROM mit der Beschriftung 1 in den Sockel 1. Das Grundprogramm wird nicht zum Betrieb von GOSI benötigt.
In den Sockeln 2 und 3 verbleiben die RAM-Bausteine, die GOSI zum arbeiten benötigt.

Arbeitsspeicher:

GOSI 0	0 bis FFF 0–4095
GOSI 1	1000 bis 1FFF 4097–8191
	32768 – 34384 (848)
Arbeitsspeicher GOSI	8000 bis ca. 8350
Programmspeicher	8350 bis 8FFF 34384 – 36863 (3247)

Einführung in die Programmierung mit GOSI

Nach dem Einschalten der Versorgungsspannung meldet sich GOSI wie im Bild 1 gezeigt.
Dabei flimmert ein helles Feld links darunter. Dieses Feld nennen die Fachleute CURSOR, oder Blinker.
Der CURSOR sagt einem wo man gerade schreiben kann. Wenn man nun eine Taste drückt, zum Beispiel A, so erscheint an dieser Stelle ein A und der CURSOR rückt um eins nach rechts.
Dem Computer muß man stets sagen, wann er den eingegebenen Text verarbeiten soll. Dazu dient die Taste mit der Aufschrift CR. Bei manchen Tastaturen steht auch RETURN oder einfach ein gewinkelter Pfeil darauf. Die Abkürzung CR steht für Carriage Return zu deutsch Wagenrücklauf. Bei eine Schreibmaschine gibt es diese Taste auch. Wenn man sie drückt, so wird das eingespannte Papier vorgeschoben und man kann in der nächsten Zeile zu schreiben beginnen.
Wir geben einmal CR an. Ist in der Zeile mit dem Cursor irgend ein Zeichen gewesen, z.B. das A, so gibt der Computer eine Fehlermeldung aus. Bild 2 zeigt das Beispiel.
?A bedeutet, daß der Computer nicht verstanden hat was er mit dem A tun soll. "Fehler." bedeutet, das wir einen Fehler gemacht haben. Der Computer ist so freundlich uns das mitzuteilen.

```
GOSI - Graphisch orientierte Sprache I
(C) Muenchen 1984 Rolf-Dieter Klein Vers 1.1
"Ein Hauch von LOGO ..."
```

```
GOSI - Graphisch orientierte Sprache I
(C) Muenchen 1984 Rolf-Dieter Klein Vers 1.1
"Ein Hauch von LOGO ..."
A

?A
Fehler.
```

Wir starten am Besten noch mal von vorne und drücken die RESET-Taste an der SBC-II-Karte oder schalten die Spannung mal kurz aus.
Nun wollen wir mal einen richtigen Befehl eingeben.
Dazu tippen wir VORWAERTS 100 ein.
Wichtig ist das wir das Wort mit AE schreiben und nicht das Ä verwenden, denn sonst versteht der Computer uns nicht.
Noch tut sich gar nichts. Na klar, denn wir müssen noch die Taste CR drücken. Nun malt der Computer eine senkrechte Linie. Tut er das nicht, so haben wir einen Eingabefehler gemacht. Bild 3 zeigt das Bild, wie es sein soll.
Wichtig ist, das man auch den Freiraum zwischen VORWAERTS und der Zahl 100 eingibt. Dies geschieht mit der langen Taste auf der Tastatur, die meißt keine Beschriftung trägt. Der Rechner versteht die Befehle nur, wenn sie eindeutig voneinander getrennt sind.

```
GOSI - Graphisch orientierte Sprache I
(C) Muenchen 1984 Rolf-Dieter Klein Vers 1.1
"Ein Hauch von LOGO ..."
vorwaerts 100
```

Nun einmal ein paar weitere Befehle. Wir tippen ein RECHTS 45 und die Taste CR nicht vergessen, der Schreibzeiger zeigt nun 45 Grad nach Rechts. Im Bild sieht man ihn nicht, da er nur auf dem Bildschirm eingeblendet wird. Dann geben wir den Befehl VORWAERTS 50 und die Taste CR anschliessend und es ergibt sich ein Bild wie in Bild 4.

```
GOSI - Graphisch orientierte Sprache I
(C) Muenchen 1984 Rolf-Dieter Klein Vers 1.1
"Ein Hauch von LOGO ..."
vorwaerts 100
rechts 45
vorwaerts 50
```

Nun kann man durch Kombination der Befehle VORWAERTS und RECHTS Figuren zeichnen.
Wir wollen aber einmal den Bildschirm löschen, das heißt alles was auf dem Bildschirm sichbar war ausradieren. Dazu gibt es auch einen Befehl. Wir tippen BILD ein und drücken anschließend die Taste CR. Nun wird alles gelöscht und in der Bildmitte erscheint der Zeiger für die Zeichnungen, der manchmal auch Schildkröte oder Igel genannt wird. Der Cursor ist ebenfalls sichtbar.
Nun wollen wir mal die Ziffer 1 auf den Bildschirm malen. Dazu noch folgender Hinweis. Befehle wie VORWAERTS, RECHTS kann man abkürzen. Anstelle VORWAERTS auszuschreiben kann man auch VW und anstelle von RECHTS RE schreiben.
Entsprechend zu VORWAERTS gibt es auch RUECKWAERTS oder abgekürzt RW und LINKS oder abgekürzt LI.
Im übrigen kann man auch mehr als nur einen Befehle auf eine Zeile schreiben, die Befehle müssen nur durch mindestens einen Freiraum getrennt sein.

Wir tippen also mal ein:
RE 90 VW 20 RW 10 LI 90 VW 100 LI 135 VW 20
und anschließend die Taste CR.
Obrigends ist es egalt ob man Groß- oder Kleinbuchstaben beim tippen der Befehle verwendet, der Computer versteht beide Schreibweisen. Großbuchstaben erhält man wenn man zuerst die Taste SHIFT und dann den dazugehörigen Buchstaben eingibt.
Bild 5 zeigt das Ergebnis unserer Arbeit.

```
re 90 vw 20 rw 10 li 90 vw 100 li 135 vw 20
```

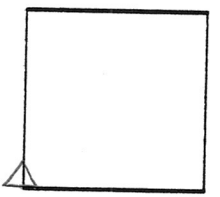
Wir geben wieder den Befehl BILD ein (und CR nicht vergessen) um einen neuen, gelöschten Bildschirm zu erhalten.
Nun kann man schon eine ganze Menge mit den bisherigen Befehlen anfangen, jedoch fehlt eine Kleinigkeit. Will man zum Beispiel eine unterbrochene Linie zeichnen, so geht das bisher noch nicht. Dazu gibt es ein paar neue Befehle:
STIFTHOCH bewirkt, das nachfolgende Befehle wie VORWAERTS oder RUECKWAERTS keine Spur mehr hinterlassen. STIFTAB hebt diesen Befehl wieder auf und danach kann wieder gezeichnet werden. Beide Befehle kann man auch abkürzen, SH für STIFTHOCH und SA für STIFTAB.
Bild 6 zeigt ein Beispiel.

```
vw 50 stifthoch vw 50 stiftab vw 50
```

Will man eine Reihe von Befehlen mehrfach wiederholen, so gibt es bei Computern eine raffinierte Vorrichtung dazu, die Wiederholschleife.
Beispiel: Wir wollen ein Quadrat zeichnen. Dazu kann man folgende Befehle geben.
VW 100 RE 90 VW 100 RE 90 VW 100 RE 90 VW 100 RE 90
Dann die Taste CR eingeben und die Leertaste (für den Freiraum) nicht vergessen.
Ein Rechteck erscheint auf dem Bildschirm.
Nun aber die Wiederholschleife. Der Befehle lautet WIEDERHOLE oder abgekürzt WH.

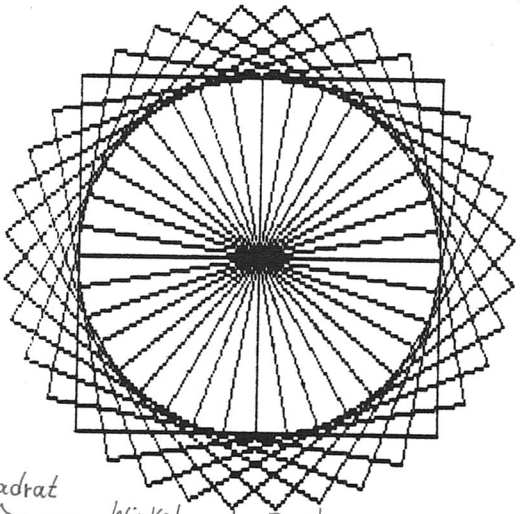
WH 4 Ä VW 100 RE 90 Ü

Anstelle der eckigen Klammern ist hier ein Ä und Ü gedruckt, genauso wie man es auf einer deutschen Tastatur eingeben muß. Auf dem Bildschirmausdruck in Bild 7 sieht man die eckigen Klammern.
Eckige Klammer auf ist die SHIFT-Taste und die Taste ä.
Eckige Klammer zu ist die SHIFT-Taste und die Taste ü.
Die Anzahl der Wiederholungen ist hier mit 4 angegeben, da die Schleife vier Mal durchlaufen werden soll, also alles was in den eckigen Klammern steht wird viermal wiederholt.



wh 4 [vw 100 re 90]

Die Wiederholschleife kann man auch ineinander schachteln. Bild 8 zeigt ein Beispiel.
Die Befehle VW 100 RE 90 werden viermal wiederholt, man erhält also ein Quadrat. Und das Quadrat wird 36 mal ausgegeben und mit dem Befehl RE 10 jeweils und 10 Grad gedreht. 36 mal 10 Grad gibt aber 360 Grad weshalb sich eine geschlossene Figur ergibt. Man kann hier einmal selbst mit unterschiedlichen Zahlen experimentieren und wird die unterschiedlichsten Figuren erhalten.
Hier gleich ein Hinweis. Auf der Tastatur befinden sich vier Pfeiltasten. Damit kann man den Cursor an eine beliebige Stelle im Bildschirmfenster hinpositionieren. Pfeil nach oben bewegt den Cursor nach oben, Pfeil nach unten, entsprechend nach unten. Nun kann man zur alten Zeile zurückfahren und CR eingeben, die Zeile mit den Befehlen wird erneut ausgeführt. Man kann auch Veränderungen an der Zeile vornehmen, bevor man CR drückt. Mit der Taste DEL lassen sich Zeichen löschen, mit CTRL-V (Die Tasten CONTROL und V gleichzeitig drücken) wird ein Zeichen eingefügt und mit CTRL-G (CONTROL und G gleichzeitig) wird ein Zeichen rechts neben dem Cursor gelöscht. Damit lassen sich auch einfach Eingabefehler korrigieren und man muß nicht die ganze Zeile neu tippen.



Anzahl d. Widerholungen
Quadrat
Seite
Winkel der Drehung

```
wh 36 [ wh 4 [ vw 100 re 90 ] re 10 ]
```

Nun kommt was ganz raffiniertes. Wir wollen dem Computer neue Wörter beibringen. Dazu gibt es den Befehl LERNE.
Wir haben schon öfters das Quadrat gezeichnet. Schön wäre es wenn wir dem Computer das Wort QUADRAT beibringen könnten.
Dazu tippen wir folgendes ein

LERNE QUADRAT

und dann natürlich CR nicht vergessen. Nichts passiert und das ist ok so.
Dann geben wir ein

WH 4 [VW 100 RE 900

auch hier passiert nach CR-Eingabe nichts weiter.
Dann geben wir den Befehl

ENDE

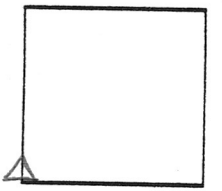
ein (auch mit CR). Nun antwortet der Computer, wie in Bild 9 sichtbar. Er sagt es sind noch 2524 Bytes zum lernen frei und 679 Bytes für Namen. Diese Angabe kann unterschiedlich ausfallen, sie gibt aber immer an wieviel Platz noch im Speicher des Computers bleibt.
Nun um das Quadrat jetzt auf den Bildschirm zu bringen tippen wir QUADRAT ein (CR nicht vergessen) und es sieht dann aus wie in Bild 10.

```
lerne quadrat
wh 4 [ vw 100 re 90 ]
ende
```

2525 680

```
Ok gelernt, Platz zum lernen: 2524 Bytes und fuer Namen: 679 Bytes.
```

[9]



```
lerne quadrat
wh 4 [ vw 100 re 90 ]
ende

Ok gelernt, Platz zum lernen: 2524 Bytes und fuer Namen: 679 Bytes.
quadrat
```

[10]

Wir wollen einmal ein kleines Experiment machen. Dazu wird das folgende Programm eingetippt:

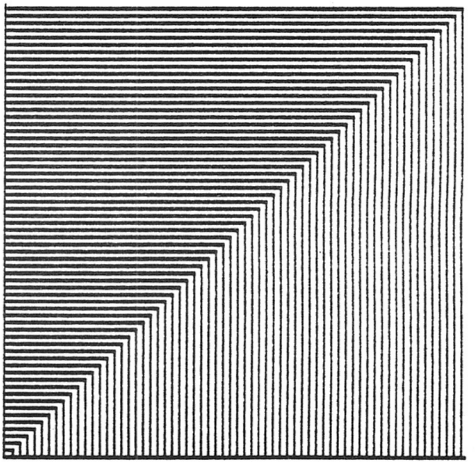
```
LERNE Q1 :N
QUADRAT :N
Q1 :N+4
ENDE
```

und jedesmal CR nicht vergessen. Das QUADRAT muß vorher schon eingegeben worden sein.
Nun geben wir folgenden Befehl:

```
Q1 1
```

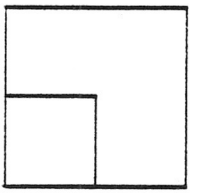
Es werden sehr schnell viele immer größer werdende Quadrate auf dem Bildschirm erscheinen.
Zum Anhalten können wir die Tasten CONTROL und S drücken.
Will man weiter machen, so drückt man CONTROL und Q.
Will man das Ganze beenden, so drückt man zuerst CONTROL und S und dann CONTROL und C. Es erscheint dann die Meldung "Ende." auf dem Bildschirm.

Diese Art der Programmierung nennt man REKURSION, denn in der Definition von Q1 wurde Q1 wieder selbst verwendet.
Das Programm hört nie auf zu laufen, da wir nicht programmiert haben, wie lange es laufen soll, wie das geht erfahren wir später.



```
lerne q1 :n
quadrat :n
q1 :n+4
ende
q1 1

Ende.
```

Nun ist es langweilig für jede Quadrat-Größe ein neues Wort zu lehren. Das geht in GOSI einfacher. Wir tippen ein:
LERNE QUADRAT :LAENGE
und die Taste CR. :LAENGE ist ein sogenannter Parameter. Später können wir eine Zahl dafür eingeben.
Mit
WH 4 [VW :LAENGE RE 90]
und
ENDE
wird der Rest eingegeben.
Hier wurde anstelle VW 100 der Ausdruck VW :LAENGE verwendet. :LAENGE ist ein Platzhalter (Parameter) der erst später durch eine Zahl ersetzt wird.
Wir können nun QUADRAT 100 oder QUADRAT 50 etc. eingeben und jedesmal wird ein Quadrat der gewünschten Größe auf dem Bildschirm erscheinen.
Bild 11 zeigt das Beispiel.
Obrigends spielt der gewählte Name keine Rolle, er darf beliebig lang sein, aber nur Buchstaben und Zahlen enthalten, keine Sonderzeichen wie Komma etc.



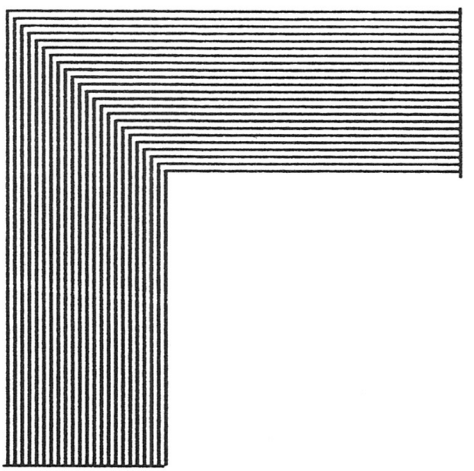
```
lerne quadrat :laenge
wh 4 [ vw :laenge re 90 ]
ende

Ok gelernt, Platz zum lernen: 2512 Bytes und fuer Namen: 679 Bytes.
quadrat 100 quadrat 50
```

Will man Teile eines Bildes löschen, so geht das mit dem Befehl STIFT 1 alle Befehle wie VW, RW wirkend dann löschend. Mit STIFT 0 kommt man wieder in den Normalzustand zurück.

Bild 13 zeigt ein Beispiel, das durch CTRL-S (CONTROL und Taste S) angehalten und durch CTRL-C abgebrochen wurde.

Der Befehl MITTE setzt den Schreibzeiger wieder in die Bildmitte zurück.

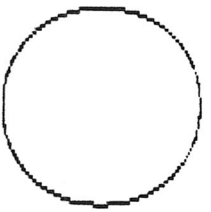
Das Bild flimmert normalerweise immer etwas, da der eigentliche Bildteil und der Schreibzeiger quasi gleichzeitig dargestellt werden, indem sehr schnell zwei unterschiedliche Bildseiten abwechselnd angezeigt werden. Dies kann man durch Eingabe des Befehls
VI
unterbinden. Mit ZI kann man den Vorgang wieder rückgängig machen. VI entstammt der LOGO-Sprache un bedeutet Verstecke Igel.



```
sh mitte sa stift 1 ql 1

Ende.
```

Wir wollen einmal einen Kreis auf dem Bildschirm darstellen. Da wir nur einzelne Punkte darstellen können müssen wir den Kreis annähern, einen mathematisch exacten Kreis können wird nicht bilden.
Wir schreiten einen Schritt vorwärts und drehen dann um ein Grad. Das Ganze wird 360 mal wiederholt. Auf dem Bildschirm erscheint eine Kreisform, Bild 14 zeigt das Ergebnis.

Will man kleinere Kreise darstellen, so kann man schneller drehen, z.B. um 2 Grad und dafür nur 180 mal die Schleife durchlaufen lassen. Damit man aber alle Stufungen erreichen kann gibt es noch einen weiteren Befehl:
SCHR16TEL
damit kann man um 1/16 Bildpunkt schreiten und damit auch sehr fein positionieren.
Beispiel:
WH 360 [SCHR16TEL 2 RE 10] zeichnet einen sehr kleinen Kreis. Bild 15 zeigt das Beispiel.

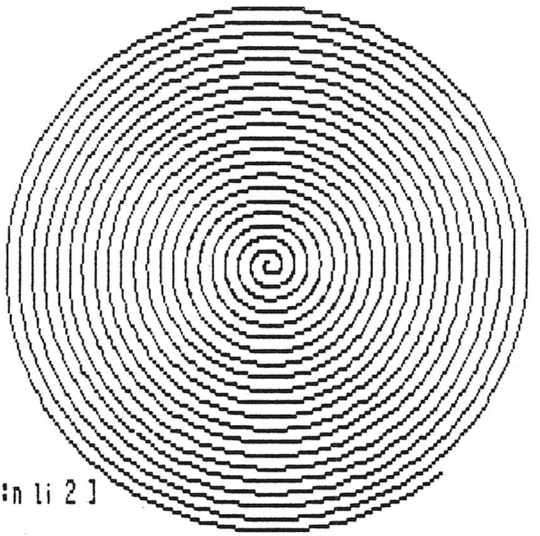


```
wh 360 [ vw 1 re 1 ]
```

```
wh 360 [ schr16tel 2 re 1 ]
```

Nun wäre es einmal interessant eine Spirale zu zeichnen, dazu zeigt Bild 16 das Beispiel.

Ähnlich interessante Ergebnisse liefert das Programm aus Bild 17, Bild 18 und Bild 19 zeigen Varianten davon. Hier wurden nun mehr als ein Parameter verwendet. Die Anzahl ist nicht begrenzt, nur durch die Anzahl der Zeichen pro Zeile.



```
lerne spiro :n
wh 45 [ schri6tel :n li 2 ]
spiro :n+1
ende

Ok gelernt, Platz zum lernen: 2438 Bytes und fuer Namen: 662 Bytes.
spiro 1
```

```
lerne vspirale :laenge :winkel
vw :laenge re :winkel
vspirale (:laenge+3) :winkel
ende

Ok gelernt, Platz zum lernen: 2356 Bytes und fuer Namen: 633 Bytes.
vspirale 5 144
```

Bild 17

Bild 18

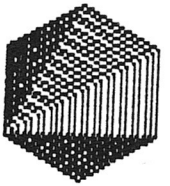
Bild 19


```
vspirale 5 120

Ende.
```

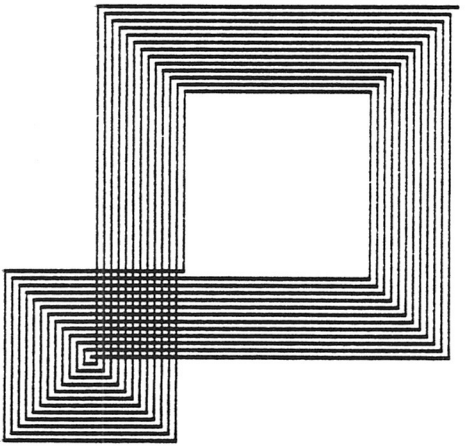
Bild 19

Nun aber zum Versprochenen Abbruchkriterium. Bisher mußten wir alle Programme mit CTRL-S CTRL-C abbrechen, das soll nun anders werden.
Der Befehl WENN erlaubt es Bedingungen abzufragen und mit dem Befehl RUECKKEHR oder abgekürzt RK kann man einen automatischen Abbruch erzwingen.
Bild 20 zeigt ein Beispiel.
Mit den Operationen, wie) ((=)= = oder () kann man Bedingungen angeben, wie in der Mathematik gebräuchlich.

Bild 21 zeigt ein weiteres Beispiel. Es dürfen beliebig viele solcher Bedingungen in einem Programm vorkommen.
Wird eine zweite Klammer bei WENN angeben, so ist das der sogenannte SONST-Teil, der nur dann ausgeführt wird, wenn die Bedingung nicht erfüllt ist.

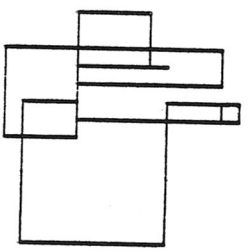


```
lerne mitabbruch :n
wenn :n>50 [ rueckkehr ]
wh 6 [ vw :n re 60 ]
mitabbruch :n+2
ende

mitabbruch 1
```



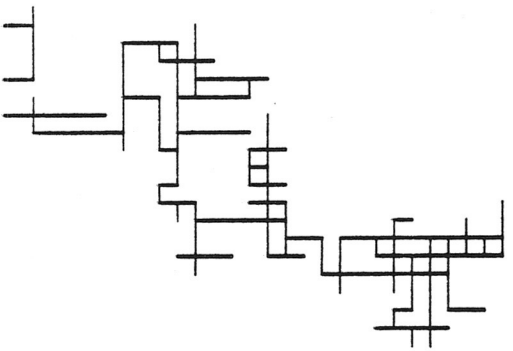
```
lerne spez :n
wenn :n>200 [ rueckkehr ]
wenn :n<100 [vw :n] [rw :n]
re 90
spez :n+2
ende
spez 1
```

Mit WENN lassen sich ganz interessante Dinge programmieren. Bild 22 zeigt ein weiteres Beispiel. :TASTE ist ein fest eingebauter Befehl. :TASTE wartet auf die Eingabe einer Taste von der Tastatur. Man erhält dann einen Wert der dem ASCII-Code (siehe Buch Mikrocomputer selbstgebaut und programmiert) entspricht.
Mit SETZE "name wert kann man einem Namen, vor dem ein Anführungszeichen stehen muß einen Wert zuordnen.
SETZE "A 1 belegt A mit dem Wert 1
:A liefert dann den Wert für weitere Operationen. Den Doppelpunkt darf man nicht vergessen.
Das Programm aus Bild 22 wartet auf die Eingabe von Tasten. Mit SHIFT V (Die Tasten SHIFT und V gleichzeitig drücken) bewegt man den Schreibzeiger um 10 Schritte vorwärts. Mit SHIFT R um 10 Schritte rückwärts und mit SHIFT L und SHIFT E kann man ihn drehen. Damit haben wir eine Art Mini-Sprache realisiert, mit der man auch Figuren zeichnen kann.



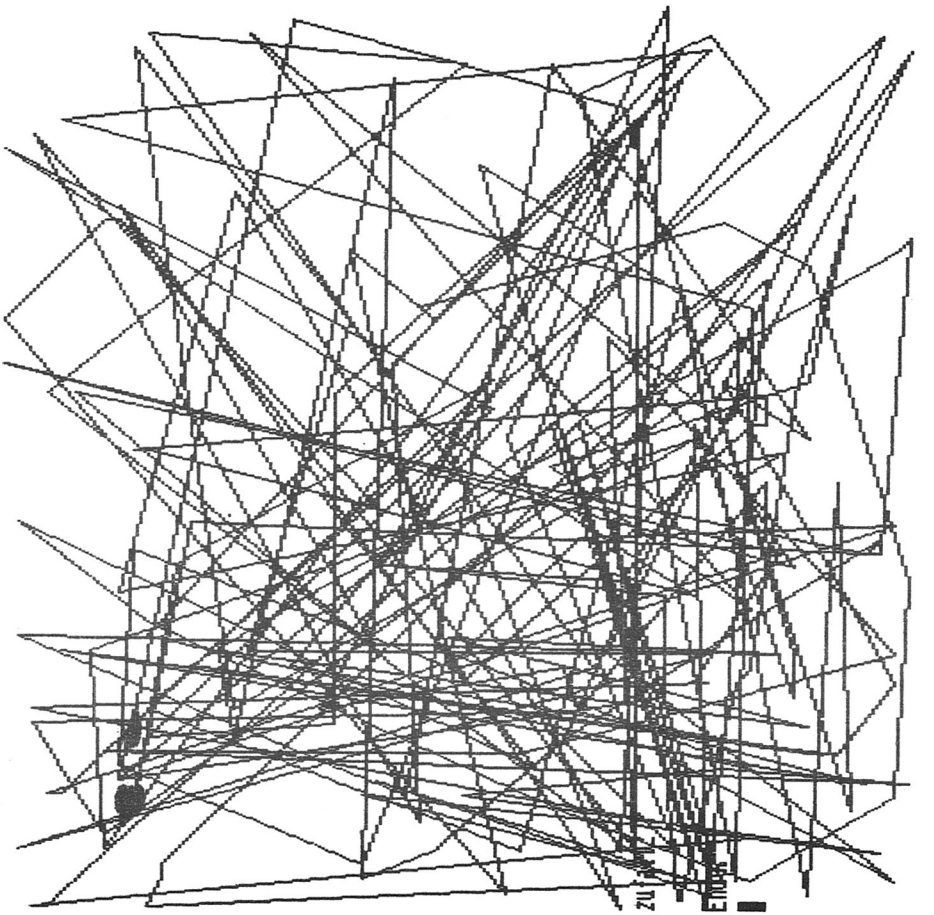
```
manu
setze "a :taste
wenn :a="V [vw 10] wenn :a="R [rw 10]
wenn :a="L [li 90] wenn :a="E [re 90]
manu
ende
manu
```

Nun kommt noch was tolles. Der Zufall. Es gibt in GOSI eine Variable :ZZ, die immer unterschiedliche Werte liefert. Dazu wird noch eine Zahl hinten angestellt, z.B. :ZZ 10 liefert eine Zufallszahl zwischen 0 und 9.
Bild 23 zeigt ein Programm damit und das graphische Ergebnis, das aber bei jedem anders ausfallen muß.
Bild 24 zeigt ein anderes Programm und Bild 25 ,das Ergebnis. Der Befehl AUFXY n1 n2 positioniert den Schreibzeiger direkt auf die absoluten Koordinaten, n1 steht für die x-Koordinate und n2 für die y-Koordinate. Der Wertebereich liegt zwischen 0 und 511.
Bild 25 zeigt ein weiteres Zufalls-Programm und Bild 26 das Bild dazu.



```
lerne zufall
setze "a :zz 3
wenn :a=0 [vw 10] wenn :a=1 [rw 10]
wenn :a=2 [re 90] wenn :a=3 [li 90]
zufall
ende
zufall
```

```
lerne zufall2
aufxy (:zz 64)*8 (:zz 64)*8
zufall2
ende

Ok gelernt, Platz zum lernen: 2404 Bytes und fuer Namen: 666 Bytes.
```



Nun ein anderes wichtiges Kapitel. Das Darstellen von Funktionen. Bild 27 zeigt das Programm und Bild 28 das Ergebnis. Mit dem Befehl LINIE n1 n2 n3 n4 lassen sich Linie von x1=n1 y1=n2 nach x2=n3 y2=n4 zeichnen. Dabei gilt für x der Wertebereich 0 bis 511 und für y der Wertebereich 0 bis 255. (Achtung dies ist anders wie bei AUFXY etc.).

Das Programm läßt sich leicht auf andere Funktionen übertragen, jedoch muß man aufpassen, da GOSI ja keine Kommastellen besitzt und nur mit ganzen Zahlen rechnen kann.

```
lerne zufall3
sh aufxy (:zz 64)*8 (:zz 64)*8 sa
setze "a :zz 30 wh 180 [schr16tel :a+1 li 2]
zufall3
ende

Ok gelernt, Platz zum lernen: 2304 Bytes und fuer Namen: 656 Bytes.
```

```
lerne fun :t
setze "erg ((:sin :t)/8)*((:cos :t*17)/8)/8
linie :t 128 :t :erg+128
fun :t+1
ende

Ok gelernt, Platz zum lernen: 2214 Bytes und fuer Namen: 650 Bytes.
```

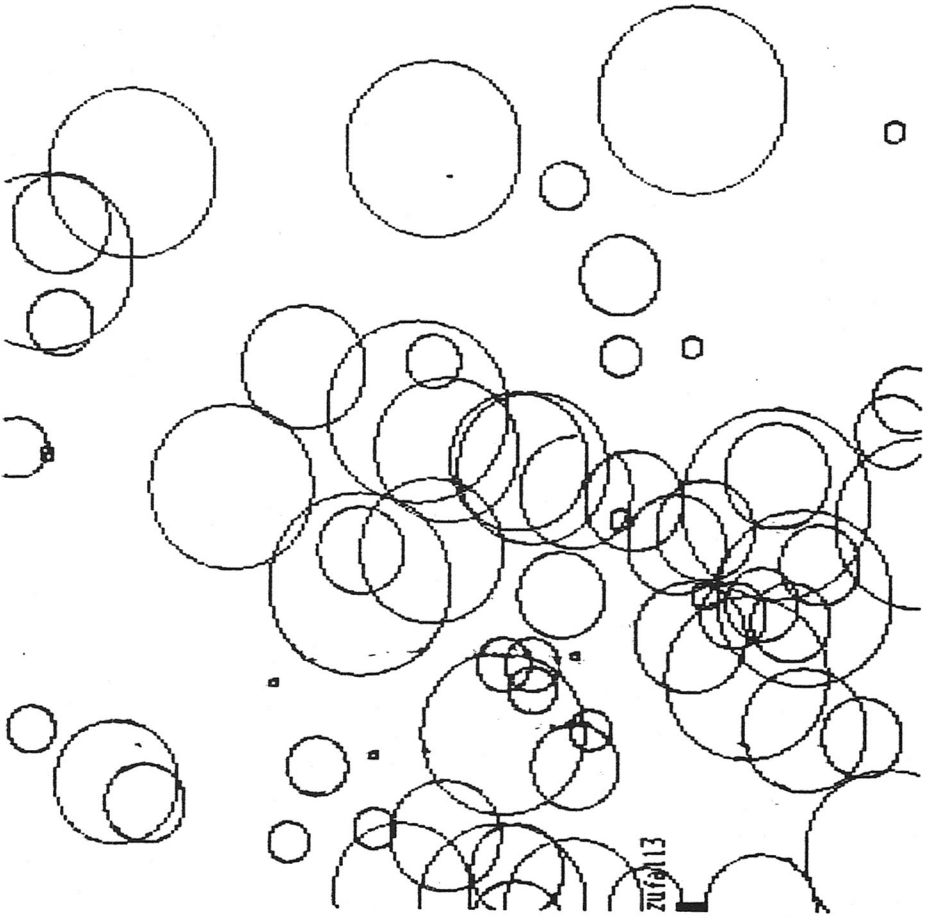


Bild 26

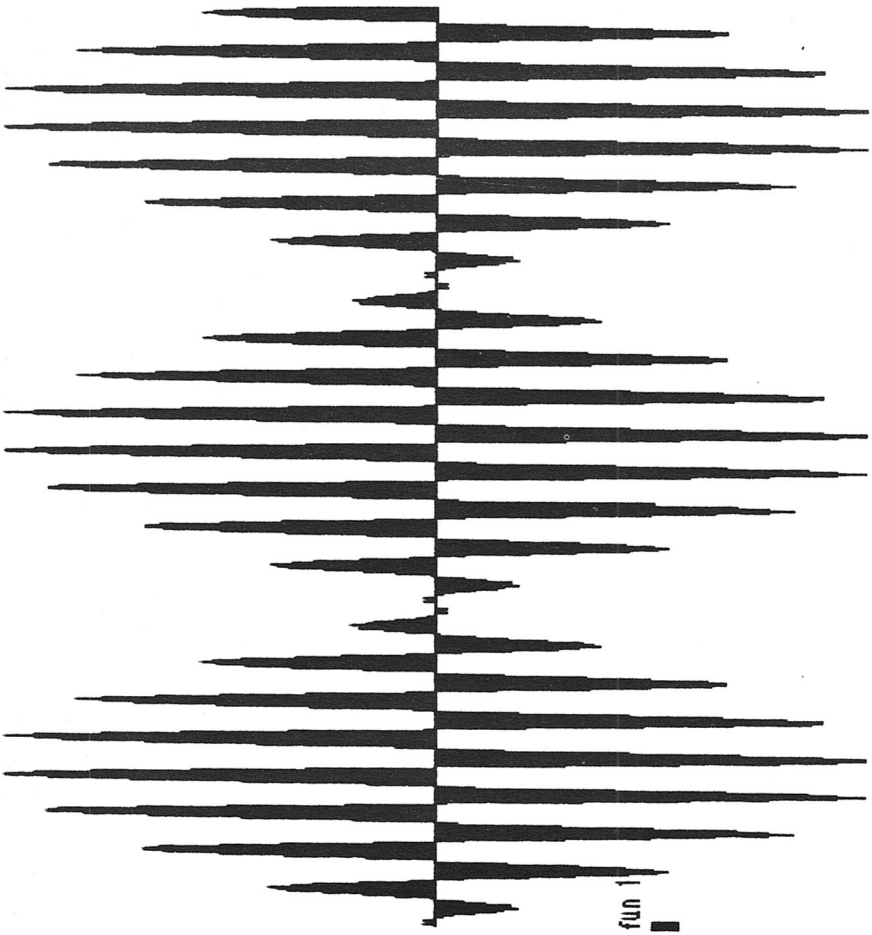


Bild 20