sinclair

ZX Spectrum+

Gebruikshandleiding

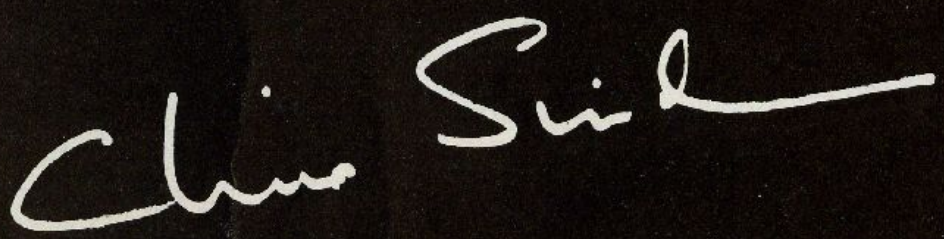
ZX Spectrum+ Gebruikshandleiding

SPECTRUM SOFTWARE

De gehele Spectrum computer software range (inclusief alle bestaande titels) is geheel compatibel met je nieuwe ZX Spectrum +

INTRODUCTIE BIJ DE ZX SPECTRUM+

Sinclair Research heeft lange tijd de vooruitgang in de microchip-technologie, die computers voor iedereen beschikbaar heeft gemaakt, geleid. Volgend op de komst van 's werelds eerste goedkope microcomputer, de ZX80, hebben we in zijn opvolgers (de ZX81, ZX Spectrum en QL computers) steeds grotere computing-capaciteit gecombineerd met steeds betere kwaliteit. Gemak in het gebruik is ook ons wachtwoord geweest, zowel wat betreft ontwerp als ook de manier waarop ze werken. De ZX Spectrum+ brengt Sinclair Research weer een stap verder op deze weg. Hier is een machine met al de beste eigenschappen van de Spectrum in een betere versie, hetgeen deze krachtigste en meest populaire van alle microcomputers nog gemakkelijker in het gebruik maakt. Het is onze hoop, dat je je voordeel doet met de vele mogelijkheden waarin je nieuwe computer voorziet.



INHOUD

Geschreven door Neil Ardley
Uitgegeven door Dorling Kindersley Ltd
in samenwerking met
Sinclair Research Ltd

HOE GEBRUIKEN WE DIT BOEK?

Deze handleiding voor je ZX Spectrum + bevat vier met kleur aangegeven hoofdstukken. Om een bepaald hoofdstuk op te slaan hoef je het boek dus alleen bij de juiste kleur te openen.

1 GA AAN DE GANG

Hoe sluit je je ZX Spectrum + aan?. ■ Afstemmen van je TV.. ■ Het opzetten van de storingzoeker.. Wat kan je ZX Spectrum + doen?. ■ Hoe gebruik je kant en klare software?. ■ Hoe laad je een programma?. ■ Storingszoeker voor Software-lading

2 BEGIN MET PROGRAMMEREN

Het toetsenbord-het besturingspaneel van je computer.. ■ Hoe gebruik je de toetsen?. ■ De televisie-rekenmachine.. ■ Hoe gebruik je kleur?. ■ Eenvoudige doe-het-zelf grafieken.. ■ Het beeldschermkladblok.. ■ Ontwerp je eigen patronen en plaatjes.. ■ Hoe maak je je eigen computer tekens?. ■ Animatie.. ■ Hoe maak je muziek en geluidseffecten?. Hoe bewaar je je eigen programma's?. ■ Storingszoeker voor het bewaren van Software

3 LEER JE ZX SPECTRUM + KENNEN

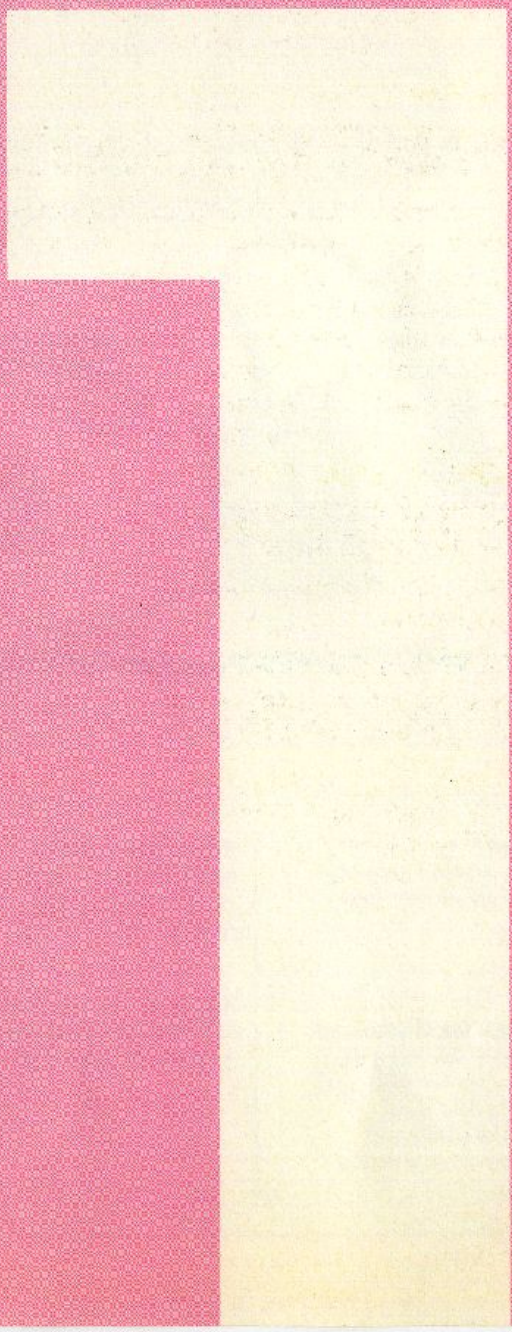
Hoe ziet het er van binnen uit?. ■ Hoe werkt je ZX Spectrum + ■ Hoe sluit je de randapparatuur aan?. ■ ZX Spectrum + geheugen kaart

4 LEER SINCLAIR BASIC KENNEN

Programmeurs' naslaghandleiding tot de Sinclair BASIC keywords.. ■ Spectrum beeldscherm rapporten.. ■ Voorbij BASIC ■ Computer-jargon – wat het betekent

GA AAN DE GANG

Dit hoofdstuk laat zien hoe je erachter komt, wat je Spectrum + kan doen. Je vindt hierin hoe je de computer aan moet sluiten, zodat hij altijd klaar voor actie is. Daarna heb je de keuze. Je kunt verscheidene programma's intoetsen, die laten zien wat de Spectrum kan doen (kleur-grafieken, geluidseffecten), of je kunt leren hoe je kant en klare software gebruikt, zoals computerspelletjes. Maar wat je ook kiest, je zult je gauw amuseren met je nieuwe computer.



HOE SLUIT JE DE ZX SPECTRUM + AAN

Bekijk eerst de controle- lijst hieronder om er zeker van te zijn, dat je alles hebt. Volg daarna de instructies voor de aansluiting op de blad- zijde hiernaast.

Sluit alles stevig aan. Als je per ongeluk de stroom- voorziening verbreekt of afsluit, gaan je programma's en al je imformatie en resultaten verloren.

Wanneer je klaar bent met de computer, zet dan de stroom uit bij het stopcontact aan de muur (als daar een schakelaar zit), en trek de stekker uit het stopcontact

Controlelijst: Heb je alles?

Bij het uitpakken vind je:

1 Je ZX Spectrum +

2 De ZX Stroom voorziening- Dit produceert de 9 Volt DC voorziening, die de Spectrum nodig heeft.

3 Het antennesnoer- dit verbindt je Spectrum met een televisietoestel.

4 Het cassettesnoer dit verbindt je Spoectrum met een cassetterecorder

5 Een garantiekaart, die je moet invullen en terugsturen.

6 Cassette behorende bij de gebruikershandleiding.

7 Deze handleiding.

Zelf heb je nodig:

1 Een televisietoestel.

2 Een cassetterecorder

3 Een stekker

Vragen en antwoorden.

Moet ik een kleuren TV hebben?
Nee. Op een zwart-wit toestel echter kun je de kleuren, die de Spectrum produceert niet zien.

Kan ik elke TV gebruiken?
De Spectrum geeft beeld met elk televisietoestel. Zoniet, dan kan het zijn, dat de televisie en de computer verschillende beeldsystemen hebben. Dit kan wanneer de televisie erg oud is, of wanneer TV en Spectrum in 2 verschillende landen gekocht zijn. Vraag in geval van twijfel advies aan je televisiehandelaar.

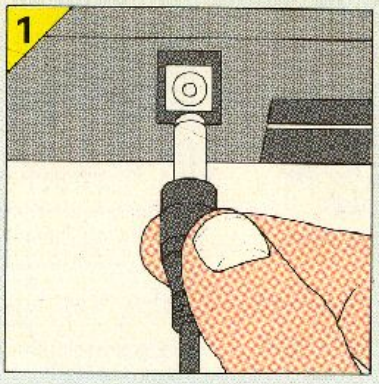
Kan ik een monitor in plaats van een televisietoestel gebruiken?
Ja. Je handelaar kan mischien een monitor leveren met een beter beeld voor de Spectrum.

Kan ik de ZX 16K RAM gebruiken?
Nee. Dit RAM-pakket kan alleen gebruikt worden met de Sinclair ZX 81 computer.

Aansluiting van je ZX Spectrum +

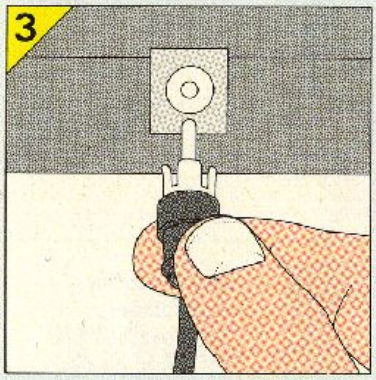
Maak eerst de stekker aan de blootgelegde draadeinden van het snoer. Zet dan een 3A zekering in de stekker. De Spectrum vereist geen geaarde connectie, zelfs niet bij gebruik van een stekker met 3 poten.

Volg vervolgens de illustraties voor de verbinding van de Spectrum aan de stroomvoorziening en aan het televisietoestel. Sla, wanneer het systeem eenmaal aangesloten is, de bladzijde om voor de afstemming.

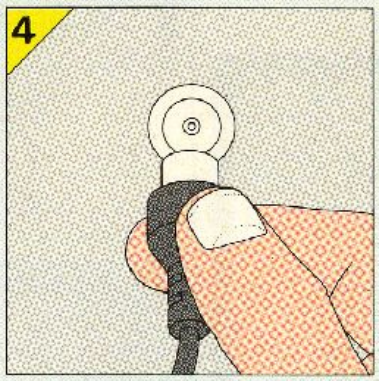


Zet de kleine stekker aan het stroomvoorzienings- snoor in het contact gemerkt 9V DC aan de Spectrum.

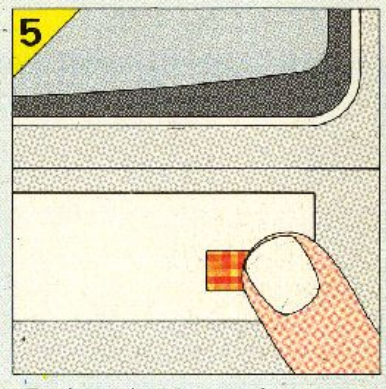
Druk de hoofdstekker in een stopcontact. Je Spectrum heeft geen aan-en-uit schakelaar.



Breng het antennesnoer in het contact gemerkt TV aan de Spectrum. Slechts één van de stekkers aan het antennesnoer past in dit contact.



Maak de antennekabel van je televisietoestel los. Zet de andere stekker aan het antennesnoer van de Spectrum in het antennecontact aan het toestel.

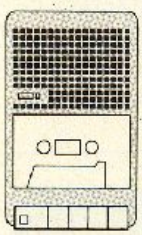


Zet het televisietoestel aan en zet het geluid zo zacht mogelijk. Nu ben je klaar om het toestel af te stemmen voor ontvangst van het Spectrumsignaal.

Contacten en connectors van de Spectrum.

Stroomcontact
De 9 Volt DC Voorziening, die geproduceerd wordt door de ZX Stroom voorziening transformator, wordt d.m.v. dit contact aangesloten.

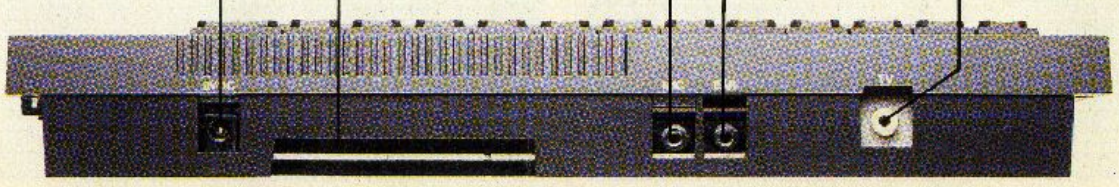
Randconnector
Een grote verscheidenheid van hardware, inclusief Microdrives, printers en modems kunnen hieraan aangesloten worden.



Ear-contact
De koptelefoon van de cassetterecorder wordt hieraan verbonden.

TV-contact
Het antennesnoer van een televisietoestel wordt hieraan aangesloten.

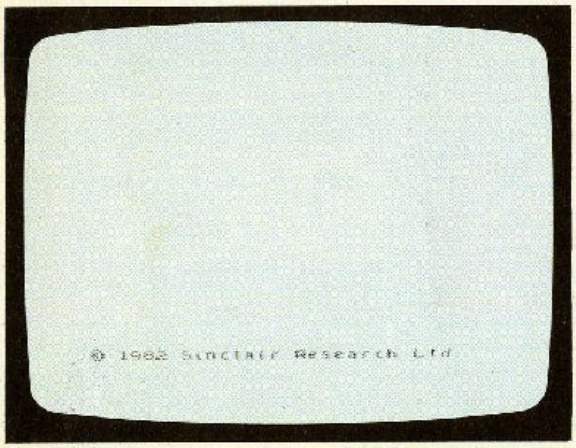
MIC-contact
De microfoon van een cassetterecorder wordt hieraan aangesloten



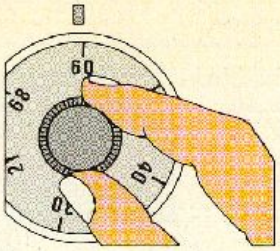
AFSTEMMEN VAN TV

Je Spectrum geeft een kleurentelevisie video signaal op de frequentie van kanaal 36 (UHF). Je televisie moet dus op dit kanaal afgestemd worden om het computerbeeld te kunnen ontvangen.

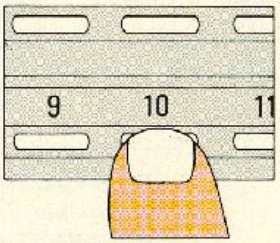
Draai, na aansluiting van de Spectrum aan het TV-toestel, aan de afstemknop totdat je de Sinclair copyright boodschap te zien krijgt.
(zie eerste scherm hieronder)

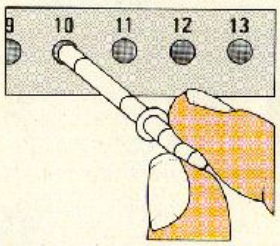


Afstemsystemen

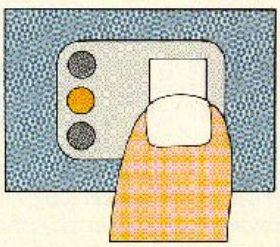


Variabele afstemming
Bij variabele afstemming kun je elk kanaal krijgen. Blijf aan de knop draaien totdat je de copyright boodschap krijgt





Drukknop-afstemming
Kies een drukknop en stel bij tot je de copyright boodschap krijgt. Gebruik indien mogelijk een en knop, die over is. Op die manier hoef je niet iedere keer opnieuw je televisietoestel af te stemmen.

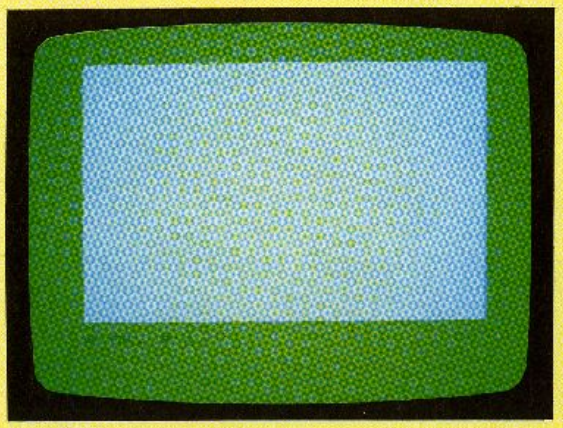


Elektronische afstemming
Deze methode is hetzelfde als de drukknopafstemming, behalve dat het toestel zichzelf op het vereiste kanaal afstemt.

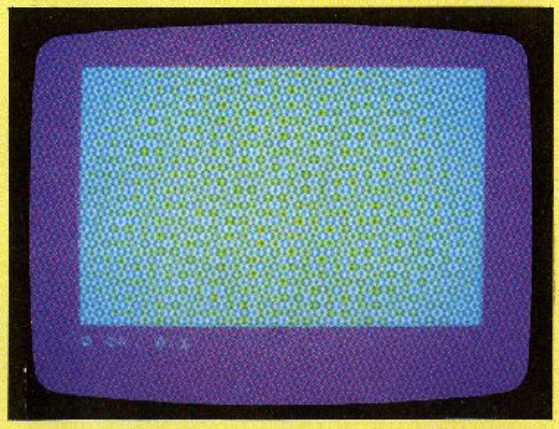
Hoe test je de kleuren van de Spectrum.

Druk eenvoudigweg de B toets in en dan een nummer van 1 tot 6. De copyright boodschap verdwijnt; eerst verschijnt het woord BORDER en dan het nummer. Druk nu de ENTER toets in. Het "border" gebied op het scherm krijgt nu de kleur aangegeven op de nummer-toets. De schermen hieronder laten zien wat er gebeurt als je BORDER 4 en ENTER intoetst, en vervolgens BORDER 3 en ENTER. Bij intoetsen van BORDER 7 wordt de border wit.

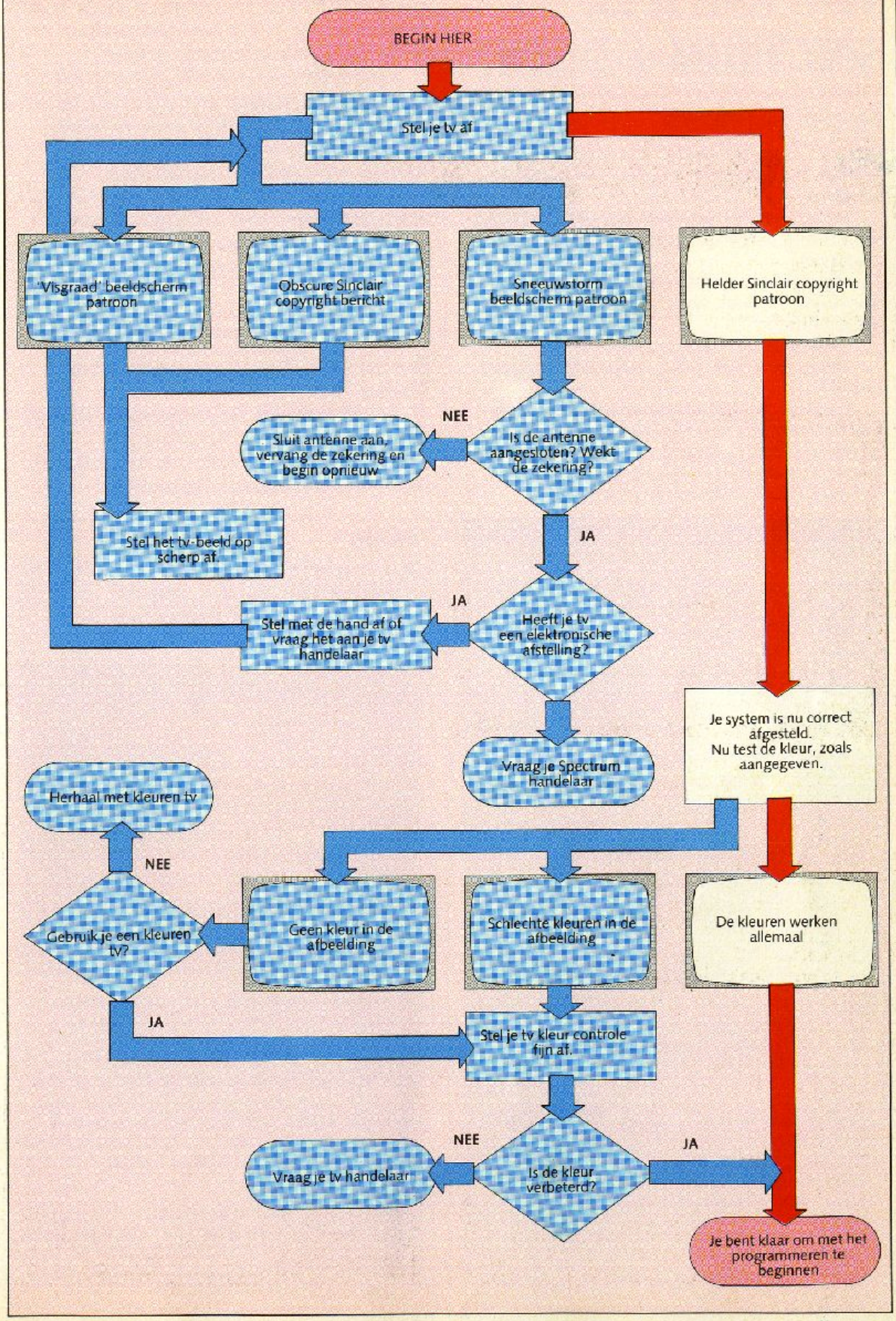
BORDER 4



BORDER 3



Schema van de probleem-oplosser



WAT KAN JE ZX SPECTRUM + DOEN?

Begin met experimenteren

Probeer, nu je Spectrum is aangesloten en je televisie is afgestemd, een paar toetsen in te drukken.

Je zult woorden en letters op het scherm zien verschijnen, en misschien ook een paar nummers.

Het is echter onwaarschijnlijk dat de Spectrum iets doet, tenzij je weet hoe je moet programmeren.

Maar maak je geen zorgen.

Er kan niets fout gaan, welke toetsen je ook indrukt.

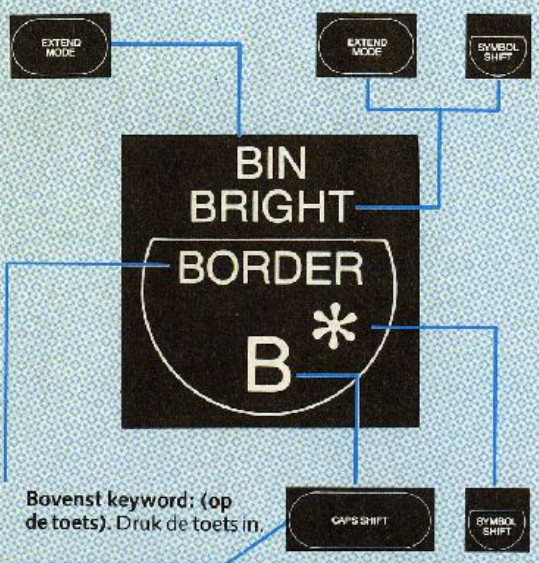
Druk nu de RESET knop aan de linkerkant van de computer in, en je bent klaar om te beginnen. Op de volgende vier bladzijden vind je enige voorbeelden voor op het TV-scherm, van wat je met je Spectrum kunt doen.

Hoe toets je in?

Voordat je een woord, letter of nummer intoetst, moet je eerst zijn positie op de toets bepalen. Gebruik dezelfde volgorde van selectortoetsen als hier getoond wordt.

Bovenste keyword. Druk EXTEND MODE in, vervolgens de toets.

Onderste keyword of teken. Druk EXTEND MODE in, hou dan SYMBOL SHIFT ingedrukt en druk op de toets.



Bovenst keyword: (op de toets). Druk de toets in.

Letter of nummer. (op de toets). Druk de toets in en houd hem ingedrukt. Gebruik CAPS SHIFT voor hoofdletters

Onderste keyword of teken. (op de toets). Hou SYMBOL SHIFT ingedrukt, en druk de toets in

zie voor verdere details over de toetsen blz. 20-21

Programmeer vervolgens je spectrum

De Spectrum kan veel doen, maar je moet het een aantal instructies geven, die tezamen een programma vormen.

Hier is een aantal korte programma's, die de Spectrum laten werken en die kleur, geluid en grafieken laten zien. Je hoeft alleen de programma-instructies in te toetsen, precies zoals ze hier staan. De beelden laten zien, wat je verwacht, maar als je de kolom "Hoe verander je een programma" hiernaast leest, kun je zelf met de programma's experimenteren.

Hoe toets je een programma in en RUN je het?

Elke set instructies, staat in een lijst die "listing" heet.

Je zult zien, dat de programma-listings verscheidene secties bevatten, elk beginnend met een nummer 1Ø,2Ø, enz. Elke sectie heet een programmalijn.

NAMEN

```
10 BORDER 1: INK RND*7
20 PAPER RND*7
30 PRINT " ZX Spectrum +";
40 GO TO 10
```

De naam ZX Spectrum+ verschijnt in vele kleuren over het gehele scherm. Vervolgens, stopt de computer en de boodschap **"Scroll?"** verschijnt onder op het scherm. Om het beeld naar boven te laten schuiven (scroll) kun je elke toets gebruiken, behalve N, SPACE, BREAK en STOP. Als je het schuiven stopzet, en BREAK indrukt, daarna R (RUN) en vervolgens ENTER, verschijnen de namen in een ander patroon en andere kleuren.

Probeer dit

Verander in lijn 3Ø "ZX Spectrum+" in je eigen naam, tussen aanhalingstekens ("), bij voorbeeld

```
3Ø PRINT "JOHN    ";
```

Vergeet niet de punt-komma (;)

In elke programmalijn zie je hele woorden of afkortingen die twee of meer letters bevatten, zoals PRINT, LET, RND, PI, PAPER EN GOTO. Dit zijn keywords en je kunt ze niet woord voor woord in te toetsen. Zoek in plaats daarvan de toets met het betreffende keyword (bv. PRINT is op de P toets) en volg vervolgens de instructies in de kolom "Hoe toets je in?."

Als je een lijn intoetst verschijnt die onder p het scherm. Druk aan het einde van de programmalijn de ENTER toets in.

Nu verschijnt de lijn boven aan op het scherm.

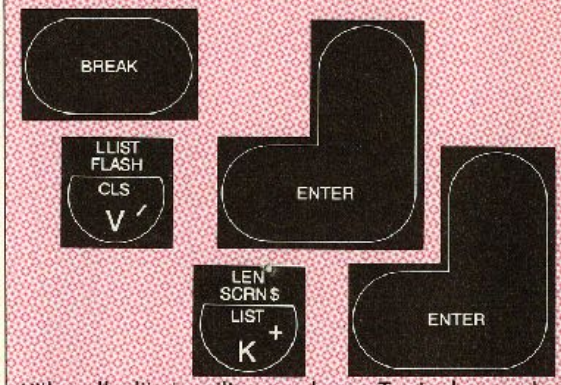
Toets elke volgende lijn op dezelfde manier in.

Als je bij ongeluk een verkeerde toets indrukt, kijk dan op de volgende bladzijde naar de kolom "Hoe corrigeer je fouten?."

Druk, als alle lijnen zijn ingetoetst R in. Het keyword RUN verschijnt. Druk nu ENTER in, en je Spectrum gaat door het programma.

Hoe je verander je een programma?

Wacht tot het programma eindigt of beëindig het door BREAK in te drukken. Druk dan V (CLS) in, daarna ENTER, vervolgens K (LIST) en dan weer ENTER. De programma listing (lijst van Lijnen) verschijnt op het scherm.

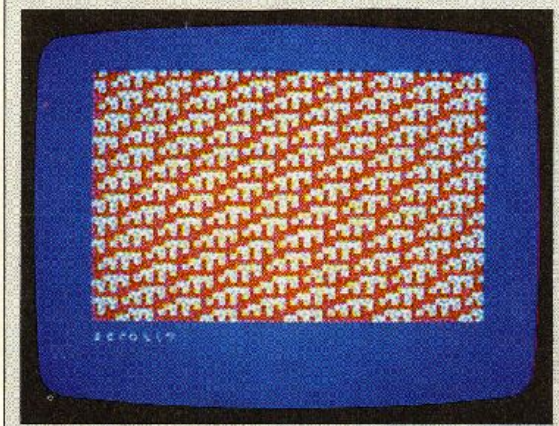


Kijk welke lijn je wilt veranderen. Toets daarna de hele nieuwe lijn in, met het lijnnummer en druk ENTER in. De nieuwe lijn verschijnt in de lijst.

Druk R (RUN) en ENTER in en het nieuwe programma begint.

PATRONEN

```
10 LET a$=""
20 FOR x=1 TO 7
30 LET a$=a$+CHR$ (RND*14+129)
40 NEXT x
50 INK RND*7
60 BORDER RND*7
70 PRINT a$;
80 GO TO 70
```



Wanneer je het programma laat werken, vormt zich een geometrisch gekleurd patroon op het scherm. Als het scherm vol is stopt de display met de scroll?-boodschap.

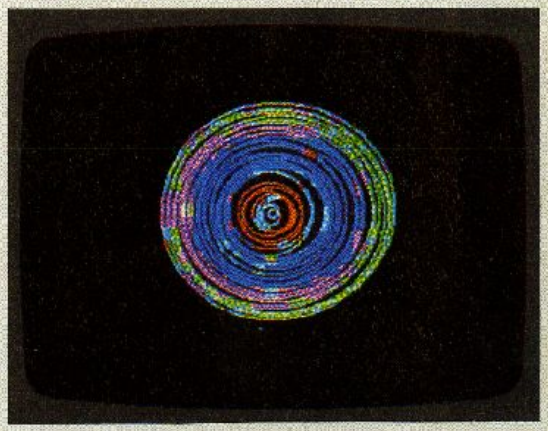
Om meer van dit patroon te zien kun je elke toets (behalve N, SPACE, BREAK en STOP) indrukken en het beeld schuift naar boven. Druk, om een nieuw patroon in een andere kleur-combinatie te krijgen, N in, wanneer de scroll? boodschap verschijnt. Druk daarna BREAK in, gevolgd door R (RUN) en ENTER.

Probeer dit

Verander in lijn 2Ø de 7 in een ander nummer, zodat een ander soort patroon onstaat. Probeer bv. 8

FLITSENDE CIRKELS

```
 10 BORDER 0: PAPER 0: CLS
 20 CIRCLE INK RND*6; FLASH RND
;120+RND*8,80+RND*8,RND*80
 30 BEEP 0.1,RND*60
 40 IF RND>.9 THEN GO TO 60
 50 GO TO 20
 60 FOR y=-2 TO 4
 70 FOR x=0 TO 6
 80 BORDER x
 90 BEEP .05,x*y
100 NEXT x
110 NEXT y
120 RUN
```



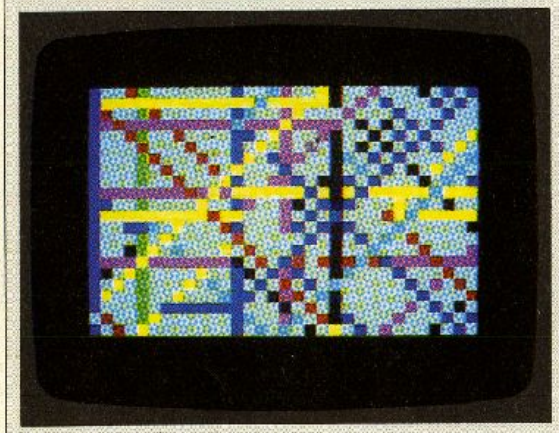
Een aantal bijna concentrische flitsende cirkels in verschillende kleuren vormen zich op het scherm. Dan flitst plotseling de border. De computer maakt een trillend geluid, en vervolgens ontstaat er een nieuwe set van cirkels.

Probeer dit

Druk, voordat je het programma list, PAPER 7 in. Druk dan ENTER in. Toets vervolgens (met gebruik van de K toets) lijn 2Ø weer in, met weglating van de keywords FLASH RND; de cirkels flitsen nu niet meer.

WAANZIN MOZAÏEK

```
  5 BORDER 0: CLS
 10 LET h=16: LET v=11
 20 LET x=INT (RND*3-1): LET y=
INT (RND*3-1)
 30 INK RND*7
 40 FOR z=1 TO 20
 50 PRINT AT v,h;CHR$ 143
 60 LET h=h+x
 70 LET v=v+y
 80 IF h<0 THEN LET h=31
 90 IF h>31 THEN LET h=0
100 IF v<0 THEN LET v=21
110 IF v>21 THEN LET v=0
120 NEXT z
130 GO TO 20
```



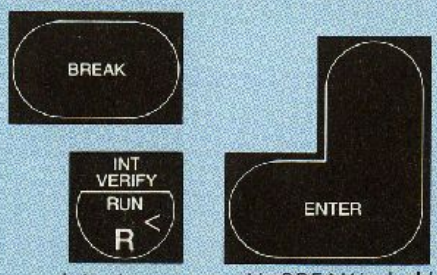
Een gekleurd vierkant vliegt over het hele scherm en bouwt zo een gekleurd patroon op. Telkens als je het programma opnieuw start ontstaat er een ander patroon.

Probeer dit
Verander in lijn 5Ø 143 in 42 en je ziet sterren! Probeer andere nummers van 33 tot 142 uit. Kijk op de tekenskaart op blz. 51 om te controleren wat er gebeurt.

Hoe start je een programma opnieuw?

Sommige programma's zoals bv. STARS AND STRIPES eindigen en produceren het rapport Ø OK en het laatste lijnnummer in het programma. Dit betekent, dat het hele programma is geëindigd. Om opnieuw te beginnen moet je R (RUN) en ENTER indrukken.

Andere programma's blijven doorlopen, zoals WAANZIN MOZAÏEK, of beginnen automatisch opnieuw, zoals OCHTEND GLOREN. Om deze



programma's te stoppen moet je BREAK indrukken. Houd deze toets ingedrukt tot het programma stopt en het BREAK-rapport verschijnt. Om opnieuw te starten hoef je alleen R (RUN) en ENTER in te drukken.

Hoe corrigeer je fouten?

Als je een verkeerde toets indrukt of als je de shift of EXTEND MODE-toetsen niet goed indrukt is er geen reden tot zorg. Druk de DELETE-toets in en het laatste keyword, teken, letter of nummer verdwijnt Houd DELETE ingedrukt om meer uit te wissen.

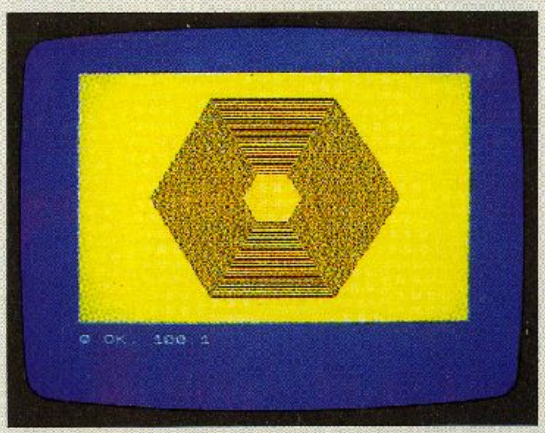
Als je een fout hebt gemaakt ná het indrukken van ENTER kan er een flitsend vraagteken in de lijn verschijnen. Op die plaats zit de fout. Druk DELETE in om de lijn tot aan het vraagteken uit te wissen Toets de lijn daarna correct in en druk op ENTER.

Als het je lukt een incorrecte lijn in te toetsen kan het programma stoppen. Het produceert een rapport met het nummer van de foute lijn onder op het scherm.

Toets de hele lijn correct in, druk dan ENTER in, en vervolgens R (RUN) en weer ENTER. Nu moet het programma werken.

POLYHEDRA

```
  5 BORDER 1: PAPER 6: CLS
 10 INPUT n
 20 FOR r=20 TO 80 STEP 2
 25 LET x=128: LET y=87
 30 LET h1=x-r: LET v1=y
 35 PLOT h1,v1
 40 FOR a=Ø TO 361 STEP 360/n
 50 LET h2=x-r*COS (a*PI/180)
 60 LET v2=y+r*SIN (a*PI/180)
 70 DRAW h2-h1,v2-v1
 80 LET h1=h2: LET v1=v2
 85 BEEP 0.02,r-20
 90 NEXT a
100 NEXT r
```

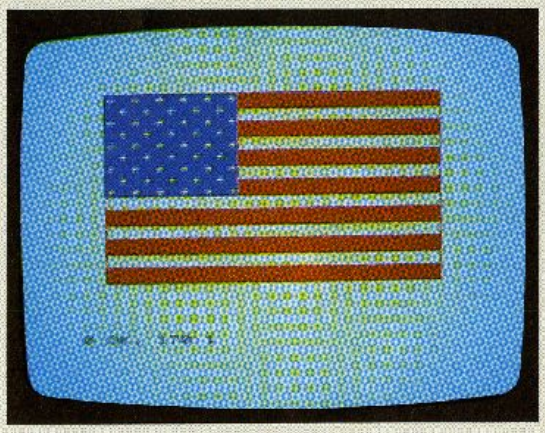


Eerst zie je een leeg scherm. Toets 6 in, daarna ENTER. Een zeskantige vorm ontstaat. Start het programma opnieuw als het geëindigd is en toets een ander nummer in, zodat je een vorm met een verschillend aantal kanten krijgt.

Probeer dit
Verander in lijn 2Ø het nummer 2 in een ander nummer. Het patroon wordt sneller opgebouwd naarmate het nummer groter is en de polyhedra's (veelzijdige figuren) worden wijder.

STARS AND STRIPES

```
 10 INK 2
 20 PAPER 7
 30 CLS
 40 FOR x=28 TO 148 STEP 20
 50 FOR z=0 TO 11
 60 PLOT 16,z+x: DRAW 216,0
 70 NEXT z
 80 NEXT x
 90 PLOT 16,28: DRAW 0,131
100 PLOT 232,28: DRAW 0,131
110 PAPER 1
120 INK 7
130 FOR x=2 TO 8 STEP 2
140 PRINT AT x,2;"* * * * * *"
150 PRINT AT x+1,2;" * * * * *
"
160 NEXT x
170 PRINT AT x,2;"* * * * * *"
```



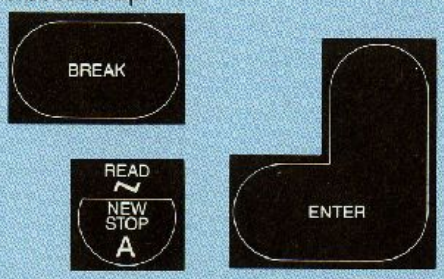
De vlag van de Verenigde Staten verschijnt op het scherm.

Probeer dit
Verander de kleurnummers van de vlag. De kleur van de strepen is in lijn 1Ø, de sterren in lijn 12Ø en de achtergrond van de sterren in lijn 11Ø.

Hoe begin je met een nieuw programma

Wanneer je met een programma klaar bent en je wilt een geheel nieuw programma intoetsen, wacht dan tot het oude programma geëindigd is of stop het door BREAK in te toetsen.

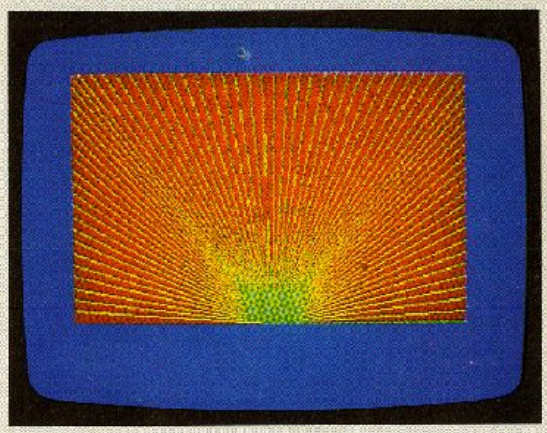
Je hebt dan de keuze tussen twee manieren om het oude programma van het computergeheugen af te wissen. Een manier is om twee toetsen in te drukken nl. A (NEW) en daarna ENTER. Het scherm wordt even zwart en dan verschijnt de copyright-boodschap.



Een gemakkelijker manier is om de reset toets in te drukken. Dit heeft hetzelfde effect als wanneer je de stekker eruit trekt en er weer induwt.

OCHTEND GLOREN

```
 10 BORDER RND*6
 20 INK RND*7
 30 PAPER RND*6
 40 CLS
 50 LET z=RND*10+2
 60 FOR x=0 TO 174 STEP z
 70 PLOT 128,0
 80 DRAW -128,x
 90 BEEP .01,x/3
100 NEXT x
110 FOR x=-127 TO 127 STEP z
120 PLOT 128,0
130 DRAW x,175
140 BEEP .01,60
150 NEXT x
160 FOR x=174 TO 0 STEP -z
170 PLOT 128,0
180 DRAW 127,x
190 BEEP .01,x/3
200 NEXT x
210 PAUSE 200
220 GO TO 10
```



Een plaatje als het "ochtend gloren" bouwt zich elke paar seconden in verschillende kleuren op. Wacht, als het scherm blanco wordt. Er komt gauw een nieuwe zonsopgang.

Probeer dit
Verander in lijn 21Ø het nummer 2ØØ in een ander nummer, zodat de tijd, dat elke zonsopgang op het scherm blijft, verandert. 2ØØ is gelijk aan 4 seconden.

Wat verder?

Je hebt nu een keuze. Als je een of meer van deze programma's wilt behouden, zodat je ze later weer kunt gebruiken, kun je ze op cassettetapes opnemen.

Kijk op bladzijde 38 in "Hoe bewaar je je eigen programma?" hoe je dit moet doen.

Als je door wilt gaan met experimenten met je Spectrum kun je en Hoofdstuk 2 BEGIN MET PROGRAMMEREN meer vinden. Tot zover heb je alleen de programma's uitgeprobeerd zonderdat je precies begrijpt **hoe** ze werken. In Hoofdstuk 2 vind je meer details over het programmeren van de Spectrum. Als je een paar software tapes wilt uitproberen, sla dan de bladzijde om en kijk in "Hoe gebruik je kant en klare software?"

HOE GEBRUIK JE KANT EN KLARE SOFTWARE

Bij het intoetsen van een programma in de Spectrum produceer je een aantal elektronisch gecodeerde signalen. De codes gaan naar het geheugen van de Spectrum, waar ze opgeslagen worden, zodat de computer ze kan gebruiken als het programma afgedraaid wordt. De codes blijven in het geheugen totdat je ze ofwel verwijdert (bv. door NEW in te toetsen of op de reset-knop te drukken) ofwel de Spectrum uitzet.

Het is echter niet altijd nodig een programma in te toetsen als je je Spectrum wilt gebruiken. In plaats daarvan kun je kant en klare software kopen, dat programma's bevat, die direct en automatisch in de computer gevoerd kunnen worden. Het gebruik van kant en klare software bespaart je niet alleen de moeite van het steeds maar weer te moeten intoetsen van programma's, maar ook stelt het je in staat om een voorraad van programma's op te bouwen, die klaar voor gebruik zijn. Het zou je dagen of weken duren om ze zelf te schrijven. Softwarefabrikanten produceren allerlei soorten programma's, die geschreven zijn door de beste programmeurs. Een groot aantal is beschikbaar voor de Spectrum. Kijk in de Sinclair Spectrum Software Catalogus om een idee te krijgen van de soort programma's, die je zou kunnen gebruiken.

Hoe worden programma's in de Spectrum geladen

De codesignalen op en een software-tape bestaan uit hoge en lage pieptonen, die opgenomen zijn met een snelheid van ong. 1500 per seconde. Wanneer je een software-tape op je cassetterecorder terugspoelt, produceert de recorder een serie pieptonen, die tezamen het programma vormen. Je hoeft alleen maar de cassetterecorder met je Spectrum te verbinden en de codes gaan direct in het geheugen van de Spectrum. Dit heet het laden van een programma. Op deze twee bladzijden vind je hoe je de cassetterecorder aansluit. Hoe je het gebruikt kun je zien op bladzijden 14-15.

Software vragen en antwoorden

Wat betekent software?
Software is de algemene naam voor programma's, die in de computer ingevoerd worden. Hardware is de term voor de eigenlijke machines, d.w.z. de computers zelf en eventuele randapparatuur.

Waarom wordt software geproduceerd op tapes?
Cassettetapes zijn gemakkelijk in het gebruik en vereisen geen speciale apparatuur. Een goedkope cassetterecorder is alles wat je nodig hebt om deze soort software te laden.

Hoe klinken op tape opgenomen programma's?
Draai er één af op je cassetterecorder zonder die op de Spectrum aan te sluiten. Je zult dan een hoog gillend geluid horen. Dit wordt veroorzaakt door codesignalen, die naar de luidspreker van de cassetterecorder gaan i.p.v. naar de computer. De signalen worden met zo'n hoge snelheid van de cassette naar de Spectrum gezonden, dat het onmogelijk is om de individuele geluiden te onderscheiden.

Zijn er andere soorten software?
Ja. Je kunt programma's krijgen in ROM-pakketten (cartridges). De cartridges worden ingebracht in een interface, die in de achterkant van je Spectrum past. Een programma op een ROM-cartridge laadt onmiddellijk, zonder wachttijd.

Software is ook beschikbaar op Microdrive-cartridges. Deze bevatten programma's, die in magnetische vorm opgenomen zijn, zoals bij cassette-tapes. Een cartridge kan verscheidene programma's bevatten en elk programma kan binnen enkele seconden geladen worden. Microdrive cartridges worden gebruikt met een Microdrive-unit. (zie blz. 46).

Wat is de beste cassetterecorder?
Je kunt een goedkope draagbare cassetterecorder gebruiken, bij voorkeur aangesloten op het lichtnet, en niet op batterijen. De recorder moet een volumeregelaar hebben, maar een toonregelaar is niet noodzakelijk. Er bestaan ook speciale computer-cassetterecorders. Deze zijn ontworpen om programma's op te slaan en te laden op een meer betrouwbare manier dan gewone recorders.

Moet er speciale zorg besteed worden aan programma's op tape?
Zoals elke vorm van magnetische opslag, kunnen programma's op cassette aangetast worden door magnetische velden. Berg je cassette dus niet op vlakbij iets, dat een sterke elektrische stroom gebruikt. Hou ze ook zoveel mogelijk stofvrij.

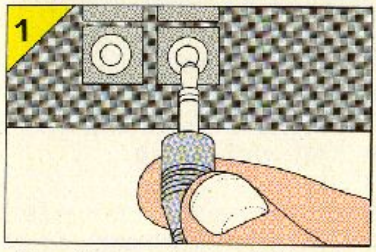
Werkt elke soort software?
Nee. Alleen software, gemaakt voor de ZX Spectrum of ZX Spectrum+ laadt.

Hoe sluit je je cassetterecorder aan?

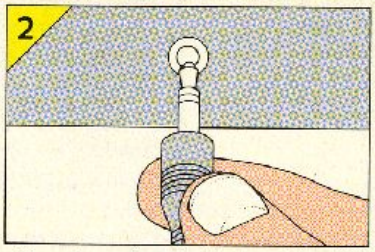
Het bij de Spectrum geleverde cassettesnoer is om aan te sluiten op je cassetterecorder. Dit is het snoer met aan elk uiteinde 2 kleine stekkers. Zet de cassetterecorder naast je Spectrum en sluit het snoer aan, zoals afgebeeld is.

De cassetterecorder en de Spectrum mogen aan of uit staan, terwijl je aansluit. Het is echter verstandig om een eventuele cassette uit de recorder te nemen vóór het aan of uit zetten. Dit beveiligt de opgeslagen programma's.

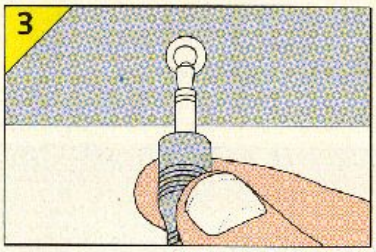
De juiste aansluiting



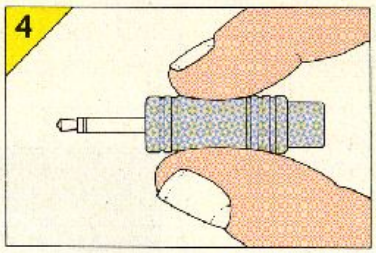
Duw één van de vier stekkers in het EAR-contact aan de achterkant van de Spectrum.



Duw de andere stekker met dezelfde kleur in het EAR-contact aan de cassetterecorder (als dit aanwezig is).



Als de cassetterecorder geen EAR-contact heeft, verbind de stekker dan aan een koptelefoon-punt, als dat er is. Probeer het anders aan te sluiten op een contact van een aparte luidspreker.



Als de stekker aan het cassettesnoer niet past in het contact van de cassetterecorder heb je een adaptor of een speciaal snoer met de juiste stekkers nodig. Het Spectrum EAR-contact heeft een 3,5 mm "jack' stekker nodig en een input-signaal van ong. 1 Volt.

Software tips

■ De Spectrum's cassettesnoer heeft gekleurde stekkers met codes, zodat de stekker van de computer en de cassette niet verwisseld worden. Als je een cassette recorder met de Spectrum gebruikt, probeer hetzelfde systeem te gebruiken. Ein kleur voor de EAR-contacten en de andere voor de MIC contacten.

■ Sommige cassette recorders kunnen interfering krijgen van andere electrische apparaten. Dit kan distorties leiden tussen de computer en cassette recorder, als gevolg dat de recorder niet goed op zal nemen. Als je cassette soms niet werkt, probeer het ergens anders te gebruiken zodat het niet naast de tv of computer staat.

EAR- en MIC-contacten
Bij het laden van programma's mag zowel het EAR-contact als het MIC-contact aangesloten zijn, zoals hier afgebeeld. Maar bij het bewaren van programma's moet het EAR-snoer eruit gehaald worden.

HOE LAAD JE EEN PROGRAMMA

Nu je een cassetterecorder op je Spectrum hebt aangesloten, ben je klaar om een programma te laden en af te draaien. Je kunt kant en klare tapes gebruiken of je eigen tapes met je eigen programma's. In beide gevallen is de procedure precies hetzelfde.

Zet de cassetterecorder aan. Zorg dat de Spectrum stroom krijgt, zet dan de cassette in de recorder. Als er al een programma in de computer zit, wacht dan eerst tot dit geëindigd is, of stop het door op BREAK te drukken. Daarna kun je NEW intoetsen of op de RESET-knop drukken om het programma uit het Spectrum-geheugen te verwijderen, maar dit is niet noodzakelijk, aangezien bij het laden het geheugen eerst wordt schoongewist.

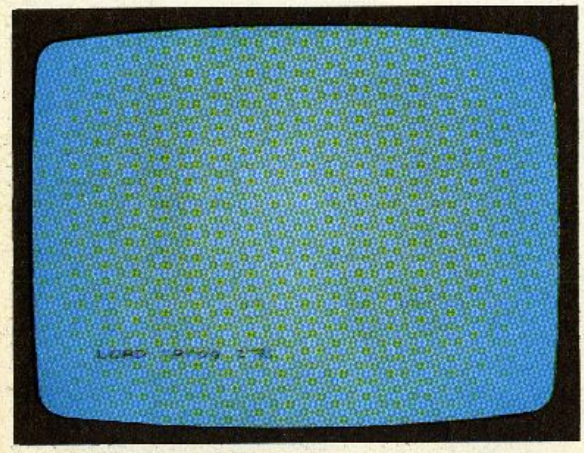
Volg nu de genummerde instructies. Als er iets verkeerd gaat, kijk dan op bladzijde 16 in "Software lading storingszoeker"

1 Zet de cassette in de recorder en spoel het terug naar het begin.

2 Zet de volume- en toonregelaars van de cassetterecorder op de vereiste niveau's. Probeer het volume uit op twee-derde maximum en zet de toonregelaar (als die er is) op maximum hoogte.

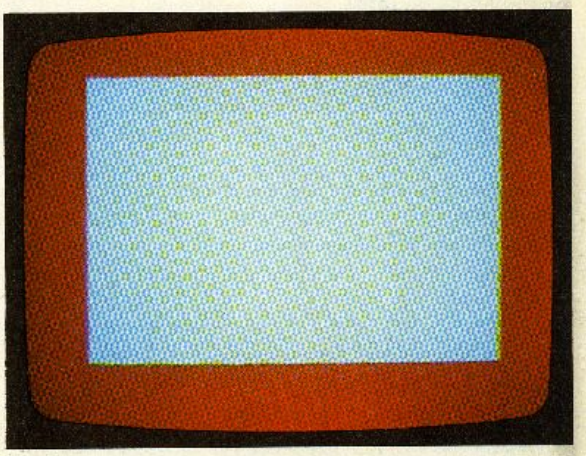
3 Druk J in en LOAD verschijnt op het scherm. Toets daarna de programmanaam in tussen aanhalingstekens, bv.:

LOAD "prog 1"



4 Druk op ENTER. Het scherm wordt blank.

5 Start de tape. De border van het scherm moet nu rood of blauw worden, of afwisselend rood en blauw. Dit geeft aan, dat de Spectrum naar een programma zoekt.

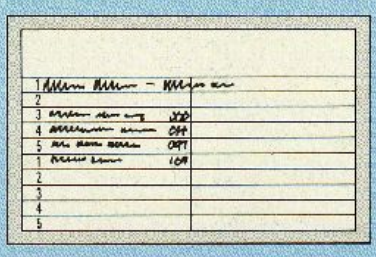


6 Na een paar seconden beginnen rode en blauwe strepen zich over de border op en neer te bewegen. Dit betekent, dat de Spectrum begonnen is een signaal te ontvangen.

Software-lading tips

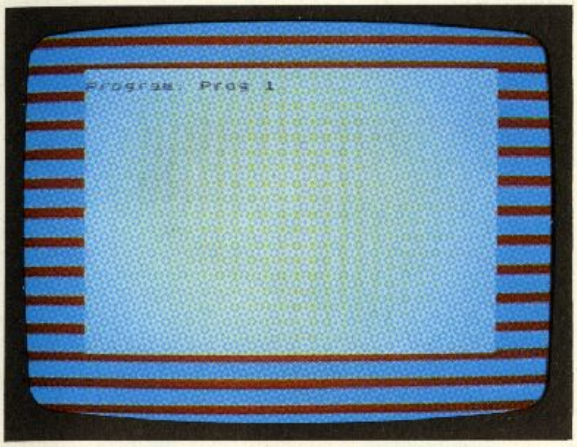
Hier zijn enkele tips, die je helpen tijd te besparen bij het laden.

1 Label alle tapes duidelijk, zodat je het programma gemakkelijk kunt vinden. Als een tape meer dan één programma bevat, schrijf dan de namen van de programma's in volgorde op het label. Onthoud, dat de programmanaam precies zo gespeld wordt als de computer nodig heeft.



2 Als je cassetterecorder een teller heeft kun je die gebruiken om snel een programma te vinden op een tape met meer dan één programma per kant. Zet de teller bij het begin van de tape op nul. Toets daarna LOAD in, gevolgd door een programmanaam (tussen aanhalingstekens), die niet op de tape staat. Draai de tape af en de Spectrum noemt elk programma zonder het te laden. Schrijf de tellernummers op het label naast de programmanamen. Later kun je dan snel en gemakkelijk het programma vinden, dat je zoekt.

7 Het woord Program: gevolgd door de programmanaam, of Bytes: gevolgd door een naam of letter, verschijnt op het scherm. Dit geeft aan, dat de computer het programma met succes heeft opgespoord.



8 De rode en blauwe strepen verschijnen weer, terwijl de computer wacht op het laden van het programma.

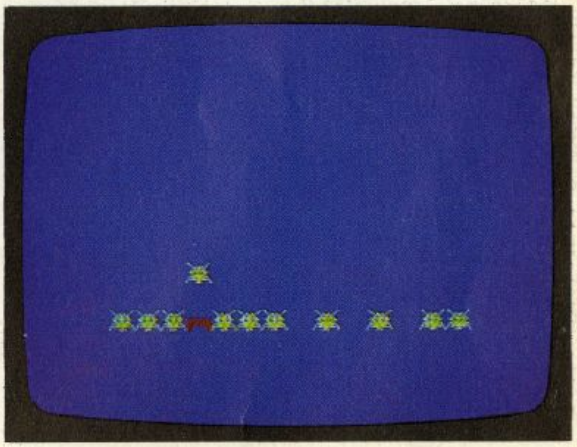
9 Een patroon van gele en blauwe strepen verschijnt in de border. Dit geeft aan, dat de Spectrum het programma laadt. Laden kan enkele minuten duren, als het programma erg lang is.

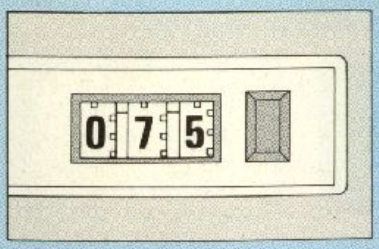
10 Bewerkingen 7, 8 en 9 kunnen zich één of meerdere keren herhalen, als het programma in secties verdeeld is.

11 Het programma kan automatisch beginnen, wanneer het geladen is. Denk eraan de tape stil te zetten.

12 Als het programma niet automatisch begint, wordt het scherm blank en het rapport Ø OK, Ø:1 verschijnt. Stop de tape.

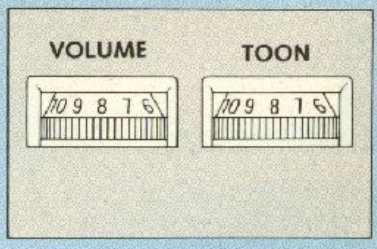
13 Druk op R(RUN) en ENTER. Het programma begint nu.





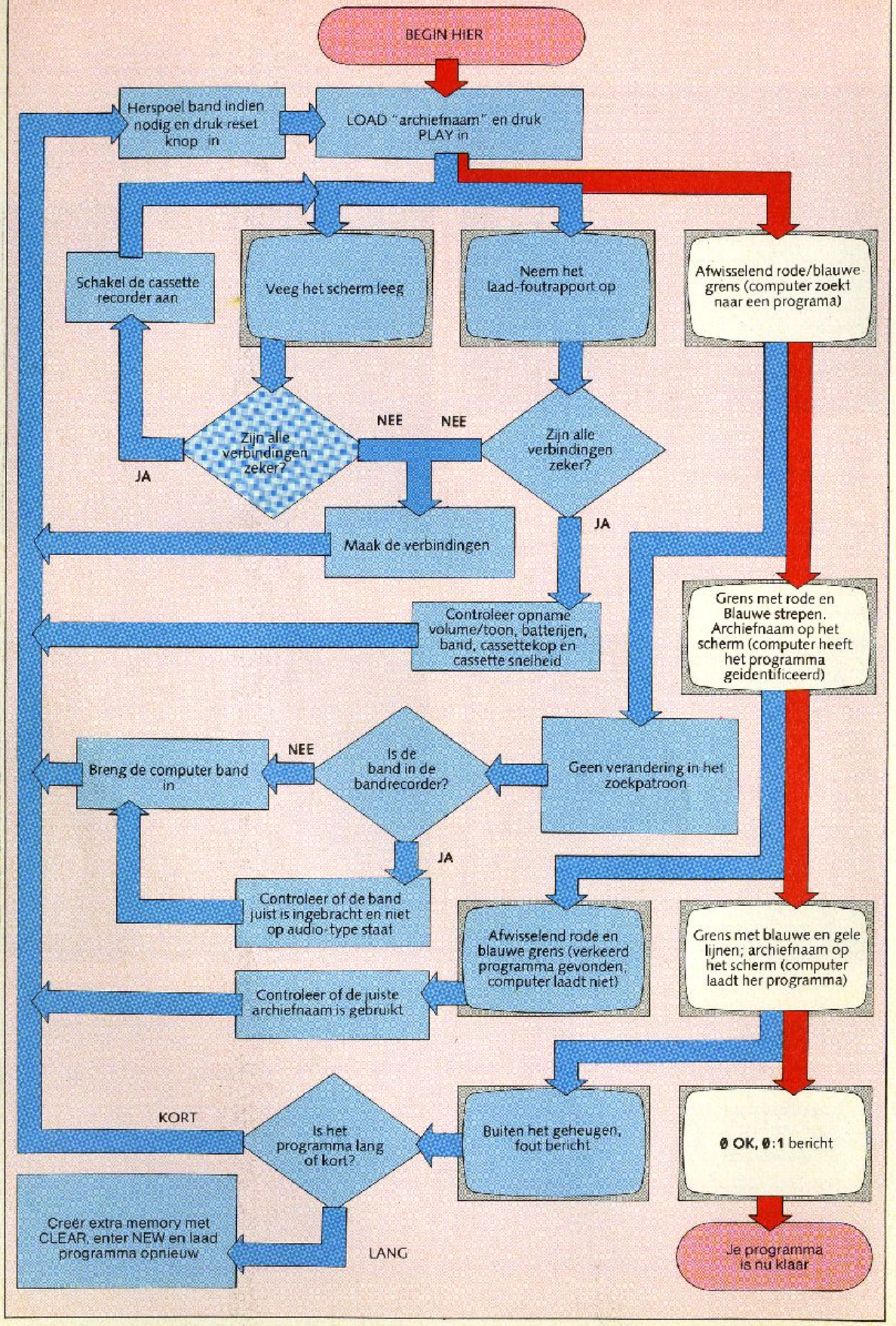
3 Als de tape bij het juiste programma is of als je de programmanaam niet weet, toets dan in: LOAD" ", in plaats van LOAD gevolgd door de naam tussen aanhalingstekens. Er mag geen spatie tussen de twee aanhalingstekens zijn. Je Spectrum laadt vervolgens het eerste programma, dat het vindt. Als de programmanaam die verschijnt niet degene is, die je zoekt druk dan op BREAK, draai de tape door en probeer het weer.

4 Denk aan de niveau's van volume- en toonregelaars.



Regel deze niveau's op de cassetterecorder vóór het laden.

Software laad probleem-oplosser



BEGIN MET PROGRAMMEREN

Dit hoofdstuk is een inleiding tot het schrijven van programma's voor de ZX Spectrum+. Het laat zien hoe je het toetsenbord gebruikt. Daarna zie je hoe je de Spectrum aan het werk zet.
De korte programma's in dit hoofdstuk concentreren zich op de speciale eigenschappen van de Spectrum.
Wanneer je begint met programma's schrijven, kun je dus alles wat je computer kan doen, gebruiken.

TOETSENBORD EN BESTURINGSPANEEL VAN JE COMPUTER

DE ZX Spectrum+ heeft zijn eigen taal, nl. de computertaal, die bekend staat onder de naam BASIC. Om de computer je instructies te laten gehoorzamen, moet je hem programmeren door ertegen te "praten" in BASIC. Dit doe je met gebruik van het toetsenbord. Verder kun je ook d.m.v. het toetsenbord de computer besturen, terwijl het je programma's afdraait.

De dialect versie, die de Spectrum verstaat, is een eenvoudige maar krachtige vorm van deze taal.

Bovendien heeft de Spectrum een geweldige eigenschap, die het programmeren veel gemakkelijker maakt.

Dit is nl., dat je met één toets een heel keyword kan intoetsen.

Toetsen en keywords

Keywords zijn speciale woorden in BASIC, die de computer opdragen iets te doen- bv. woorden als PRINT en INPUT. Bij de meeste

GRAPH
Deze toets wordt gebruikt om de vormen of grafiektekens op toetsen 1 tot 8 te selecteren.

Als je deze toets indrukt en dan een nummertoets met of zonder gebruik van de CAPS SHIFT-TOETS, verschijnt er een grafiekteken op het scherm. Druk altijd GRAPH weer in, als je de computer weer normaal wilt laten werken.

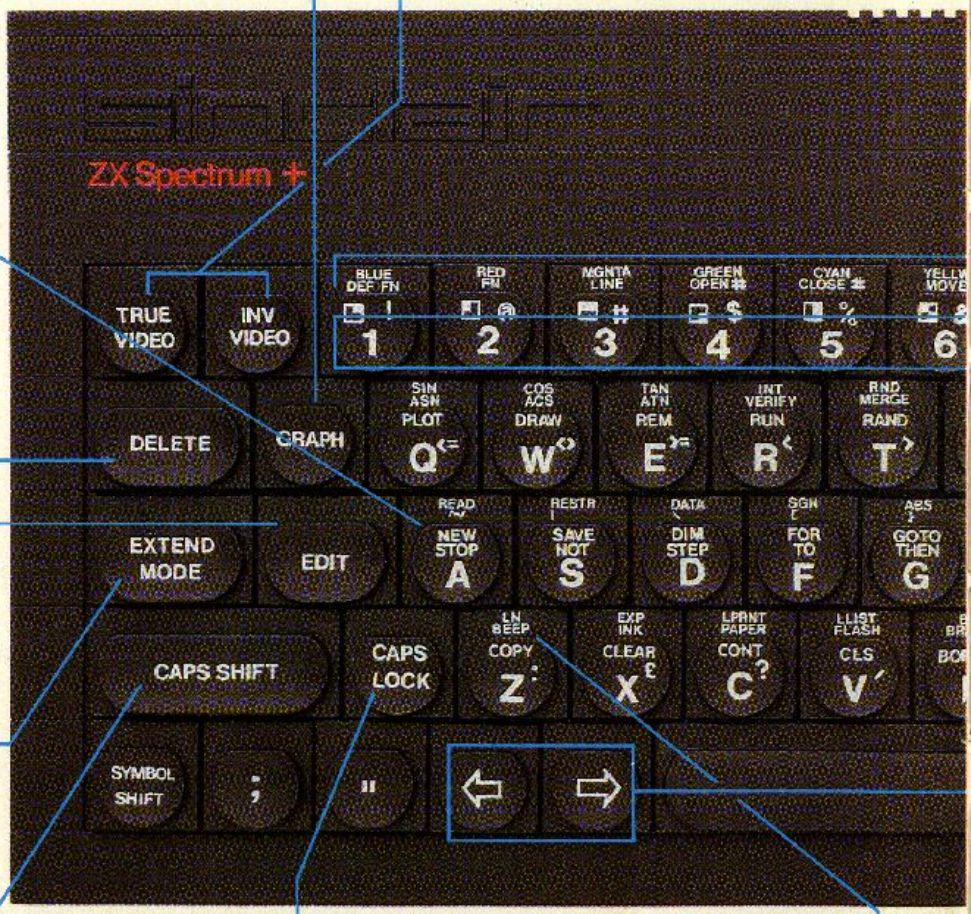
TRUE VIDEO en INV VIDEO
Deze toetsen voegen regelcodes in in de programmalijnen om normale of omgekeerde kleuren te produceren.

NEW
Deze toets veegt het BASIC geheugengebied van de computer schoon, zodat elk programma uitgewist wordt.

DELETE
Deze toets wordt gebruikt, als je een verkeerde toets indrukt en je wilt een keyword, letter, nummer of teken verwijderen-zie blz. 10.

EDIT
Deze toets wordt gebruikt om een lijn in een programma te veranderen, zonder de lijn helemaal opnieuw te schrijven-zie blz. 21

EXTEND MODE
Deze toets kiest het bovenste keyword boven de toets (bij elke toets). Gevolgd door SYMBOL SHIFT en een toets, kiest het teken of keyword direct boven de toets zie blz. 20-21



CAPS SHIFT
Druk deze toets met een lettertoets in om een hoofdletter te verkrijgen.

Als je een aantal hoofdletters wilt, kun je CAPS LOCK gebruiken.

CAPS LOCK
Gebruik deze toets, als je de hele tijd hoofdletters wilt krijgen. Druk de toets opnieuw in om weer kleine letters te krijgen.

BEEP
Deze toets produceert het keyword, dat de geluids-synthesiser regelt.

computers moet je elke letter van een keyword intoetsen, zoals bij een schrijfmachine. Je moet dus ook elk woord absoluut correct spellen. Bij de Spectrum echter druk je eenvoudigweg één toets in om het hele keyword te krijgen.

Sinclair BASIC heeft meer dan 80 keywords, die toegankelijk zijn d.m.v. 36 toetsen (26 lettertoetsen en 10 nummertoetsen).

Veel toetsen produceren niet één, maar meerdere keywords, die de computer herkent. De meeste toetsen geven keywords zowel als een letter, nummer, teken of zelfs een vorm (grafiekteken), die allemaal in een programma gebruikt kunnen worden.

Het kiezen van keywords en tekens

Op het Spectrum-toetsenbord zijn twee toetsen, die je veel zult gebruiken. Dit zijn EXTEND MODE en SYMBOL SHIFT. Met deze toetsen kun je kiezen welke keywords en tekens op de andere toetsen je wilt laten verschijnen op het scherm. Nu je vertrouwd bent geraakt met het toetsenbord, zullen de volgende twee bladzijden je precies laten zien hoe je een teken op het toetsenbord selecteert. Als je dit eenmaal weet, kun je beginnen met je eigen programma's te schrijven.

KLEUR-DISPLAY TOETSEN
Deze 6 toetsen produceren keywords, die de manier waarop de spectrum kleur op het scherm afbeeldt, regelen.

NUMMER TOETSEN
Deze toetsen produceren nummers, maar kunnen ook regelcodes in programma's invoeren voor de afgebeelde kleuren zie blz. 33. De keywords direct boven de toetsen 4 tot Ø, behalve toets 8, worden alleen met ZX Microdrives gebruikt.

BREAK
Deze toets stopt een programma. Het wist een programma niet uit het computergeheugen.

ENTER
Druk ENTER in om een programmalijn in het geheugen van de Spectrum in te toetsen. ENTER wordt ook vaak gebruikt om informatie in de computer te voeren tijdens een programma.

SYMBOL SHIFT
Houdt deze toets ingedrukt en druk op een letter of nummer toets om een lager gelegen keyword of teken op de toets te selecteren.
Bij gebruik van deze toets na EXTEND MODE wordt het symbool of keyword direct boven de toets geselecteerd.
Zie blz. 20-21

SPATIE-BALK
Deze produceert een spatie, net zoals een spatie-balk op een schrijfmachine

CURSOR-TOETSEN
Deze toetsen doen de cursor bewegen in de zelfde richting als de pijlen. Deze toetsen worden vaak door programma's gebruikt om de beweging van vormen op het scherm te regelen.
Ze worden ook gebruikt bij het bewerken van programma's.

HOE GEBRUIK JE DE TOETSEN

Je kunt zoveel als zes verschillende keywords, letters, nummers of tekens krijgen van de meeste toetsen op je ZX Spectrum+. Het is echter niet ingewikkeld om een teken of keyword te selecteren, als je eenmaal vertrouwd geraakt bent met één van de speciale eigenschappen van de Spectrum Bij het indrukken van een toets hangt het resultaat, dat op het scherm verschijnt, af van de modus, waarin de computer op dat moment staat. De verschillende modi geven verschillende typen informatie, zoals keywords, letters of grafische tekens. Het voordeel hiervan is, dat bij het gebruik van het toetsenbord, de Spectrum je in feite helpt bij het kiezen van de modi zodat je de instructies en informatie in de juiste volgorde intoetst.

Keywords modus

Zet de Spectrum aan of druk de reset-knop in, zodat de copyright-boodschap verschijnt. Druk nu ENTER in. Een flitsende K verschijnt nu in de linker benedenhoek van het scherm. Het flitsende vierkantje wordt cursor genoemd.

Het geeft aan waar iets op het scherm gaat verschijnen. De K geeft aan, dat de computer zich in de keyword-modus bevindt Bij het indrukken van elke toets verschijnt nu het bovenste keyword op het verhoogde deel van de toets.

Bij het indrukken van Q bijvoorbeeld verschijnt het keyword **PLOT**. Druk op de DELETE-toets om het keyword te verwijderen en probeer andere toetsen uit. Nummer-toetsen geven nummers, maar zodra je op een letter-toets drukt, verschijnt het bovenste keyword op het verhoogde deel van de toets.

Gebruik DELETE weer, zodat de K cursor verschijnt. Druk nu op SYMBOL SHIFT, houdt de toets ingedrukt en druk op een lettertoets. Deze keer verschijnt het keyword of teken, dat net boven de letter op het verhoogde deel van de toets staat. Bij een nummer-toets verschijnt het teken aan de rechterkant op het verhoogde deel van de toets.

Hoe kies je een keyword, symbool of teken

Hier kun je zien, hoe je elk keyword, symbool of teken op zowel een letter als een nummertoets selecteert. Kijk bij het selecteren van een toetsfunctie eerst waar het zich bevindt op de toets. Besluit vervolgens, d.m.v. de twee voorbeeld-toetsen, welke andere toetsen je eventueel moet veranderen in de juiste modus.

Letter Key

Keyword (K) modus

Toets alleen in — **BORDER**

SYMBOL SHIFT en toets in- *

Extended (E) modus

EXTEND MODE toets dan alleen- **BIN**

EXTEND MODE dan SYMBOL SHIFT en toets in- **BRIGHT**

Letter (L) modus

Toets alleen in — b

CAPS SHIFT en toets in — B

SYMBOL SHIFT en toets in — *

Hoofdletter (C) modus

CAPS LOCK Toets dan alleen in — B

CAPS LOCK dan SYMBOL SHIFT en toets in- *

Grafiek (G) modus

GRAPH (User-defined Graphics) dan alleen toetsen A tot U

Number key

Keyword (K) modus

toets dan alleen in- 3

SYMBOL SHIFT en toets in #

Extended (E) modus

EXTEND MODE toets dan alleen- magenta (kleurregelings-code)

EXTEND MODE dan SYMBOL SHIFT en toets in- LINE

Letter (L) modus

Toets alleen in — 3

SYMBOL SHIFT en toets in- #

Hoofdletter (C) modus

CAPS LOCK Toets alleen in — 3

CAPS LOCK dan SYMBOL SHIFT en toets in #

Grafiek (G) modus

GRAPH Toets dan alleen in ▀

GRAPH dan CAPS SHIFT en toets in ▄

Letter en hoofdletter modi

Na een keyword of teken in keyword modus geproduceerd te hebben, verandert de computer de cursor automatisch naar L. Het is nu in de letter modus.

Bij het indrukken van elke lettertoets verschijnt nu een kleine letter. Als je op een nummer drukt, verschijnt er een nummer. Als je een hoofdletter wilt, druk dan CAPS SHIFT in en dan een lettertoets.

Als je allemaal hoofdletters wilt, moet je eerst CAPS LOCK indrukken. De cursor verandert naar C. Je Spectrum is nu in de hoofdletter modus. Elke keer, als je een lettertoets indrukt, krijg je een hoofdletter. Om terug te kerem naar de letter modus (L) moet je CAPS LOCK opnieuw indrukken.

Extended modus

De volgende modus wordt "extended modus" genoemd. Je krijgt deze door de EXTEND MODE-toets in te drukken. De cursor verandert nu naar E. Als je nu op een lettertoets drukt verschijnt het bovenste keyword boven de toets. Druk bv. B in, en **BIN** verschijnt. Om het onderste keyword of teken boven de toets te krijgen moet je eerst de SYMBOL SHIFT indrukken en naar beneden houden en vervolgens een lettertoets indrukken. Met toets B krijg je nu bijvoorbeeld **BRIGHT**. Na indrukken van een toets (of modus. EXTEND MODE) keert de computer automatisch terug naar letter of hoofdletter modus.

Grafiek modus

De vijfde modus wordt grafiek modus genoemd en wordt verkregen door de GRAPH-toets in te drukken. De cursor verandert naar G. Als je toetsen 1 tot 8 indrukt zie je, dat de grafiektekens, die op deze toetsen staan, verschijnen.

Druk nu CAPS SHIFT in en dan elk nummer van 1 tot 8. De grafiektekens verschijnen opnieuw, maar deze keer zijn zwart en wit omgekeerd. Om de grafiek modus te verlaten moet je altijd opnieuw GRAPH indrukken.

Editing met de Spectrum

Wanneer je de Spectrum commando's geeft of je schrijft een programma, zul je soms fouten in de commando's of in de programmalijnen willen corrigeren. Je kunt dit gemakkelijk doen d.m.v. editing.

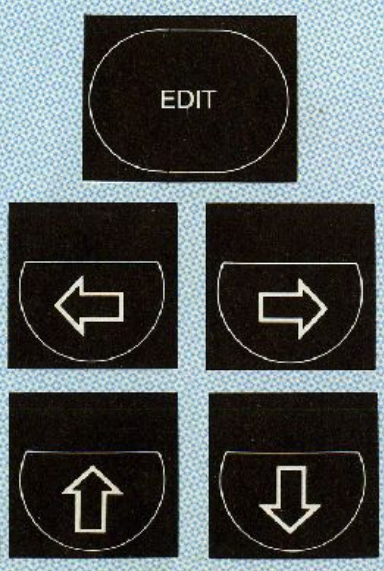
Hoe corrigeer je een fout?
Als je probeert een lijn of commando in te toetsen in verkeerd BASIC, laat Spectrum een flitsend "?" zien naast de fout. Beweeg de cursor m.b.v. de linker of rechter cursorregeltoets naar de fout. Wis de fout uit door DELETE in te drukken, of voeg toe wat vereist is. Druk vervolgens ENTER in.

Stel bij voorbeeld, dat je de computer 7 met 8 wilt laten vermenigvuldigen en je vergeet SYMBOL SHIFT in te drukken om het * teken te krijgen. In feite toets je nu in

PRINT 7b8

De Spectrum kan dit commando niet gehoorzamen, dus bij het intoetsen van ENTER verschijnt er een flitsend vraagteken bij de b, waar de fout zit.

Je hoeft nu alleen de cursor net rechts van de fout te brengen en DELETE in te drukken. De b wordt verwijderd; druk vervolgens SYMBOL SHIFT en B in om * te krijgen en toets dan ENTER in, zodat de computer het juiste commando gehoorzaamt; je hoeft niet eerst de cursor terug te brengen naar het einde van de lijn.



Hoe verander je een programmalijn?
Wanneer je een programma schrijft bouw je een aantal genummerde lijnen op, die tezamen een "listing" worden genoemd. Als je na het schrijven van het programma de lijst wilt laten verschijnen, druk dan K (LIST) en ENTER in. Nu moet je een > teken bij één van de programmalijnen zien Als dit niet het geval is, gebruik dan de op of neer cursortoets. Als je nu op EDIT drukt wordt de betreffende lijn onder op het scherm gedupliceerd en kan m.b.v. de cursor-toetsen veranderd worden, zoals hiervoor beschreven is. Door dan ENTER in te drukken, wordt de nieuwe lijn in het programma geplaatst. Als je een andere lijn wilt veranderen, breng dan het > teken met de cursor-toets naar de betreffende lijn en toets vervolgens EDIT in Als dit te lang duurt, toets dan LIST in gevolgd door het lijnnummer, daarna EDIT. Om een hele lijn uit het programma te verwijderen hoef je slechts het lijnnummer en ENTER in te toetsen. Als je een programma afdraait met een fout erin, zul je een "error" - rapport zien.

Dit wordt uitgelegd op blz. 74

DE TELEVISIE REKEN-MACHINE

De ZX spectrum+ kan uiterst snel en met grote nauwgezetheid berekeningen maken. Het heeft alleen de getallen nodig om mee te werken en tekens zoals + en − om het te instrueren wat te doen met die getallen.

Toets eerst deze instructie in. Het + teken (±) vind je op de K-toets:

PRINT 6 + 2

Dit is een commando. Wanneer je ENTER intoets verdwijnt het commando en het antwoord, het getal 8, verschijnt op het scherm.

Je Spectrum gebruikt vijf tekens die bekend staan als "arithmetic operators" voor calculaties. Onderaan deze bladzijde in het schema kun je zien wat elk van deze operators doet. Je kunt ze allemaal op dezelfde manier met PRINT gebruiken. De Spectrum kan ook de calculaties tegelijkertijd met de resultaten afbeelden.

Toets dit commando in:

PRINT "6+2=";6+2

De computer beeldt af:

6+2=8

Wat er gebeurt is, dat PRINT *alles* tussen aanhalingstekens (") op het scherm afbeeldt, dus 6+2= verschijnt. De punt-comma geeft de Spectrum opdracht om het resultaat direct na het = teken af te drukken

De Spectrum Calculatie

De volgende "arithmetic operators" worden door de Spectrum gebruikt om wiskundige bewerkingen uit te voeren Merk op, dat de computer de tekens × en ÷ niet gebruikt

Symbool	Toets	Functie	Example
+	K	Optellen	8+2=1Ø
−	J	Aftrekken	8−2= 6
*	B	Vermenigvuldigen	8*2=16
/	V	Delen	8/2= 4
↑	H	Machtsverheffen	8↑2=64

JE EERSTE PROGRAMMA

Wanneer een commando eenmaal uitgevoerd is, vergeet de Spectrum het. Als je wilt, dat de computer de berekening herhaalt, kun je het schrijven als een programma. Toets de volgende instructie in en druk ENTER in:

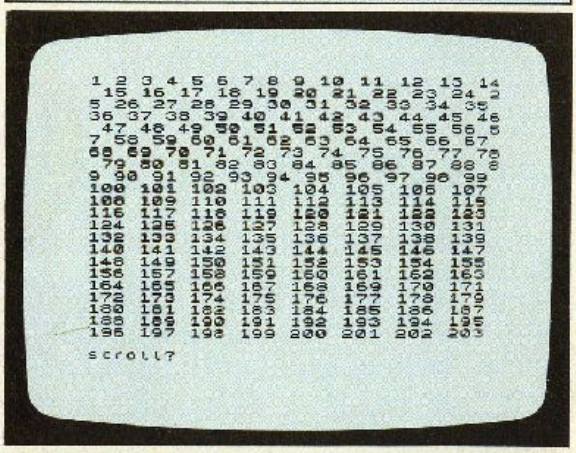
1Ø print 6+2

Deze keer wordt het niet direct gehoorzaamd. I.p.v. beeldt de computer de instructie af op het scherm.

Toets R (RUN) en ENTER in. Het resultaat (8) verschijnt nu. De hele instructie is nu een computer-programma. Als je een nummer aan het begin zet, plaatst de Spectrum de instructie in zijn geheugen, maar voert hem niet uit zonder opdracht. Wanneer je een programma afdraait d.m.v. het intoetsen van R (RUN) en ENTER, wordt de instructie uitgevoerd. Het wordt nu een "statement" genoemd i.p.v. een commando, en vormt nu een genummerde lijn. Programma "statements" worden uitgevoerd in volgorde van de nummers, en deze bestaan meestal uit tientallen. Toets het volgende programma in. Denk eraan na iedere lijn ENTER in te toetsen. Toets, als je klaar bent, R (RUN) en ENTER in. Wanneer je het programma afgedraaid hebt moet je hetvolgende zien:

NUMBER CHART

```
10 LET n=1
20 PRINT n;" ";
30 LET n=n+1
40 GO TO 20
```



Alle nummers van 1 tot 2Ø3 worden afgebeeld. Druk nu op welke toets dan ook, behalve N, Spatie-balk, STOP of BREAK. Een hele nieuwe serie nummers verschijnt.

Dit programma gebruikt een variabele. In dit geval wordt de variabele n genoemd. Elke letter of woord kan gebruikt worden- n is hier

eenvoudigweg een afkorting voor nummer. Een variabele heeft een waarde, die verandert gedurende het afdraaien van het programma. In lijn 1Ø wordt het keyword LET gebruikt om de waarde op 1 te stellen. Lijn 2Ø beeldt de waarde af, gevolgd door een spatie. Dan, in lijn 3Ø wordt LET weer gebruikt; deze keer om de waarde met 1 te vermeerderen, zodat n nu een waarde van 2 heeft. Lijn 4Ø gebruikt het keyword GOTO om het programma terug te sturen naar lijn 2Ø, die nu 2 afbeeldt. Dit wordt steeds herhaald, totdat de getallen het scherm opgevuld hebben.

Hoe laat je een programma om een nummer vragen?

Stop het programma d.m.v. de BREAK-toets. Toets nu een nieuwe lijn in.

1Ø INPUT n

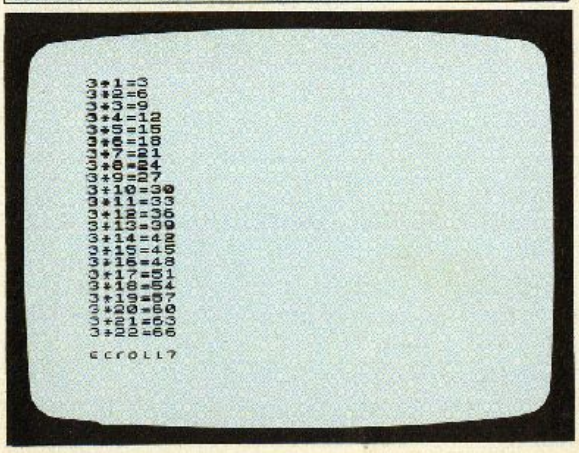
Deze lijn komt in de plaats van de oude lijn 1Ø in het programma. Bij het afdraaien van het programma wacht de computer nu, tot je een nummer hebt ingetoetst. Toets een nummer in en druk ENTER in. De nummers beginnen nu bij het getal, dat je hebt ingetoetst. Dit gebeurt omdat INPUT n de waarde van n gelijk maakt aan het ingetoetste getal.

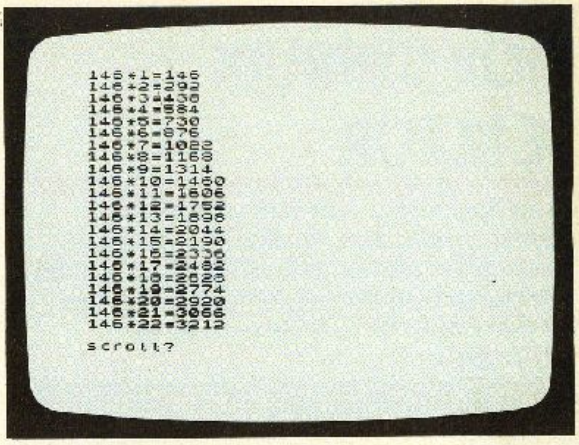
Programmering van een vermenigvuldigingstafel.

Druk op de RESET-knop om het oude programma te verwijderen en toets het nieuwe in. Dit programma laat de Spectrum vermenigvuldigen. Toets een willekeurig getal in en de tafel van dat getal verschijnt. Als je nu op een willekeurige toets drukt, behalve N, BREAK of de spatiebalk, gaat de tafel door. Druk BREAK in en draai het programma weer af om zo een nieuwe tafel te krijgen. Hier is het programma en wat je moet zien, als je 3 en vervolgens 146 intoetst.

VERMENIGVULDIGINGSTAFEL

```
10 LET x=1
20 INPUT n
30 PRINT n;"*";x;"=";n*x
40 LET x=x+1
50 GO TO 30
```





Waarom je haakjes moet gebruiken

Soms moet je in een berekening haakjes gebruiken. Toets de volgende twee commando's in en vergelijk de resultaten:

PRINT 6+2/4
PRINT (6+2)/4

De eerste geeft 6.5 als resultaat, de tweede geeft 2. De reden voor deze verschillende resultaten ligt in het feit, dat de computer een ingebouwd systeem van prioriteiten heeft, die het gebruikt in berekeningen.

Het voert eerst ↑ uit, dan * of /, en tenslotte + of −, maar het voert berekeningen tussen haakjes altijd eerst uit. Dus in het eerste commando hierboven deelt de computer eerst 2 door 4 en telt dan het resultaat (Ø.5) op bij 6.

In het tweede commando telt de computer 6 en 2 bij elkaar op en deelt dan door 4.

De Spectrum en leestekens

De Spectrum gebruikt een verscheidenheid van leestekens. Ze zijn zeer belangrijk, omdat ze ook gebruikt worden als instructies, die invloed hebben op de manier waarop de computer een programmalijn begrijpt, of waarop het een display produceert.

- **Punt-komma.** Bij gebruik met PRINT geeft het de computer opdracht aan weerskanten ervan iets af te beelden op het scherm.
- **Dubbele punt.** Geeft het einde van een "statement" in een programmalijn aan en het begin van de volgende.
- **Aanhalingstekens.** Alles hiertussen wordt behandeld als text, en niet als nummers of variabelen. Aanhalingstekens beginnen en eindigen een string.
- **Komma.** Bij gebruik met PRINT geeft het de computer opdracht om iets ofwel in het midden van de lijn of aan het begin van de volgende lijn te zetten. Gebruik de komma niet om duizenden, of miljoenen aan te geven.
- **Punt.** Is ofwel een decimaalpunt of een punt aan het einde van een zin.

HOE GEBRUIK JE KLEUR

Je ZX Spectrum+ kan acht verschillende kleuren produceren, en elke kleur heeft een kleur-codenummer. Je kunt elke kleur op drie verschillende manieren gebruiken – als een border-kleur, een inkt-kleur en als een papier-kleur.

ZX Spectrum+ kleurcodes

Dit schema laat de kleuren en codes zien, die door de Spectrum gebruikt worden. Je hoeft deze codes niet te onthouden; de nummer-toetsen zijn ook gemerkt met de kleurnamen.

Nummer	Kleur
Ø	Zwart
1	Blauw
2	Rood
3	Magenta (Paars)
4	Groen
5	Cyan (Blauw-Groen)
6	Geel
7	Wit

De feiterlijke kleuren die je op je televisietoestel krijgt hangen af van het toestel, de kleurafstelling en contrast – en helderheidsregelaars. Denk eraan, dat je een kleurentoestel nodig hebt.

Drie manieren van kleurgebruik

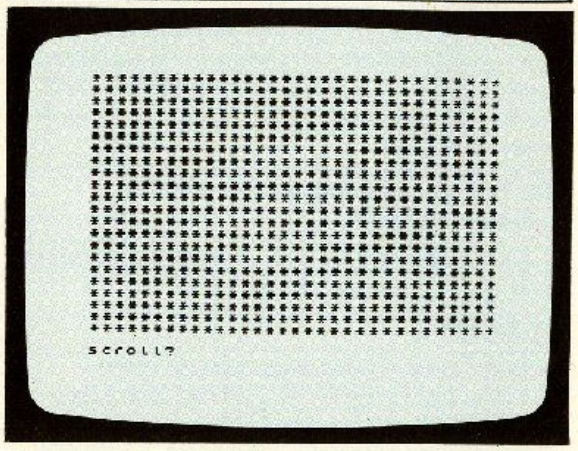
Je kunt kleur op drie verschillende manieren regelen. De border-kleur is de kleur van de border rond het centrale display-gebied. De inkt-kleur is de kleur waarin tekens (letters, nummers, tekens en grafische vormen) en punten en lijnen afgebeeld worden De papier-kleur is de kleur van de achtergrond, ofwel het gehele displaygebied, ofwel een vierkant rond elk teken.

Wanneer je de Spectrum aanzet, gebruikt het de van te voren afgestelde kleuren. De inkt-kleur is zwart, de border- en papierkleuren zijn wit. Je kunt deze kleuren onmiddellijk veranderen door via het toetsenbord directe commando's in te toetsen. Je hebt dit al zien werken op bladzijden 6 en 7.

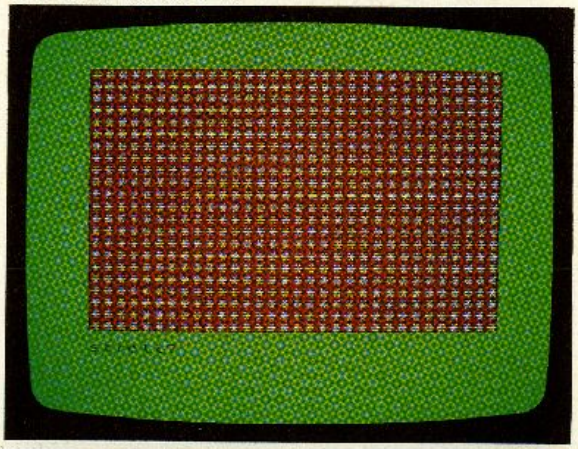
druk nu op de RESET-knop, toets dit eenvoudige programma in en draai het af.

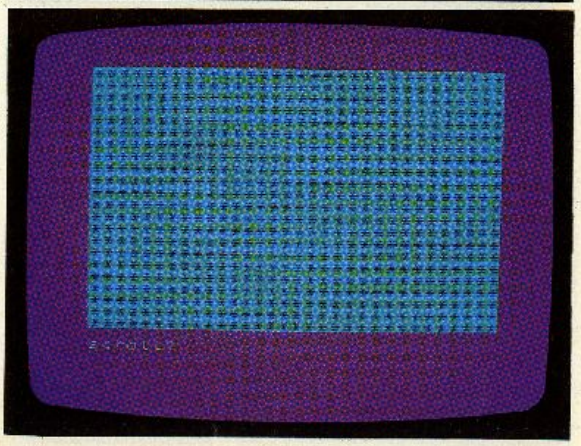
COLOUR TESTER

```
10 PRINT "*";
20 GO TO 10
```



Een patroon van sterren ontstaat in zwart en wit. Druk nu BREAK in en enkele kleurcommando's in. Toets BORDER, INK en PAPER en, elk gevolgd door een getal van Ø tot 7. Toets dan ENTER in en draai het programma opnieuw af. Hier zijn twee displays ; de eerste met BORDER 4, PAPER 2 en INK 7, en de tweede met BORDER 3, PAPER 5 en INK 1.





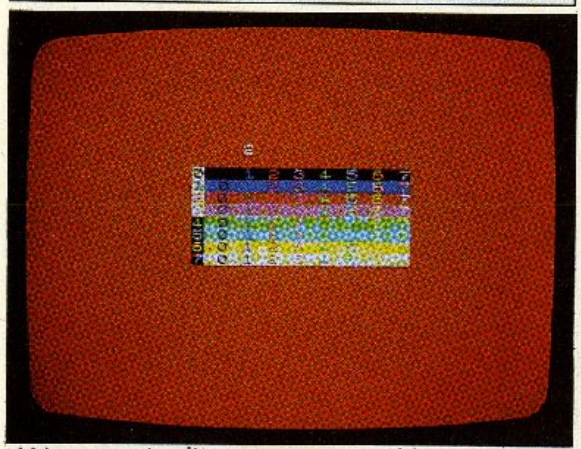
Hoe schrijf je programma's met kleuren?

Je kunt BORDER-, PAPER- en INK-keywords in een programma gebruiken om text, tafels, patronen en plaatjes in allerlei kleuren te laten verschijnen. Het gebruik van BORDER in een programmalijn doet de kleur van de border veranderen zodra de Spectrum bij deze lijn aankomt. INK alléén in een lijn verandert de inkt-kleur van alle tekens of lijnen die erna verschijnen. Alleen PAPER in een lijn verandert de papier-kleur, maar alleen rond de tekens (inclusief punten en lijnen). Als je een bepaalde kleur voor de gehele achtergrond van het display-gebied wilt, moet je PAPER laten volgen door CLS.

Je kunt INK en PAPER ook gebruiken na PRINT. In dit geval hebben alleen die tekens, die afgebeeld worden door PRINT- deze INK- en PAPER- kleuren. Het volgende programma laat alle border-, inkt- en papierkleuren zien. Ook demonstreert het hoe je INK en PAPER gebruikt na PRINT.

KLEUR COMBINATIES

```
  10 FOR b=0 TO 7
  20 BORDER b: PAPER b: CLS
  30 PRINT AT 6,12; INK 9;b
  40 FOR p=0 TO 7
  50 PRINT AT p+8,8; INK p; PAPE
R 9;p;
  60 BEEP 0.5,b*p-20+p
  70 FOR i=0 TO 7
  80 PRINT INK i; PAPER p;" ";i;
  90 BEEP 0.01,i*5
 100 NEXT i
 110 NEXT p
 120 NEXT b
```



Wanneer je dit programma afdraait, zul je alle combinaties van border, papier en inkt zien. Het programma heeft drie variabelen; b voor het bordernummer, i voor het inktnummer en p voor het papiernummer. BEEP produceert het geluid, en de lijnen, die beginnen met FOR en NEXT geven het begin en einde van een programmalus aan, die alle kleurnummers van Ø tot 7 in volgorde verandert. Merk op, dat INK en PAPER allebei een waarde van 9 kunnen hebben.

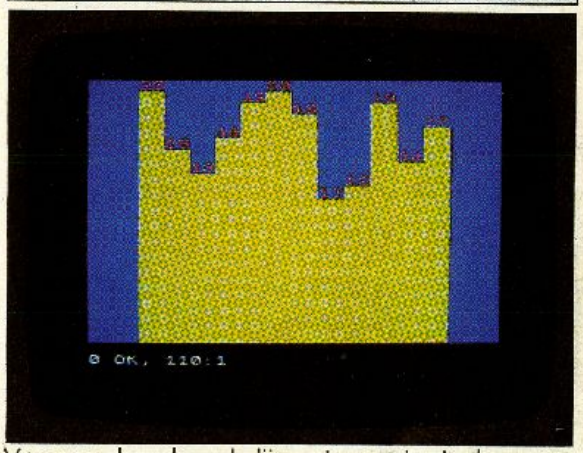
Dit maakt de inkt of papierkleur ofwel zwart of wit, zodat het contrasteert met de achtergrond of een teken.

Gekleurde balken programmeren

Het volgende programma gebruikt de Spectrum-kleuren om een balk-grafiek te produceren. Het laat twaalf over-dag-temperaturen zien als gele kolommen met nummers. Toets in lijn 6Ø twee spaties in tussen de aanhalingstekens.

BALK GRAFIEK

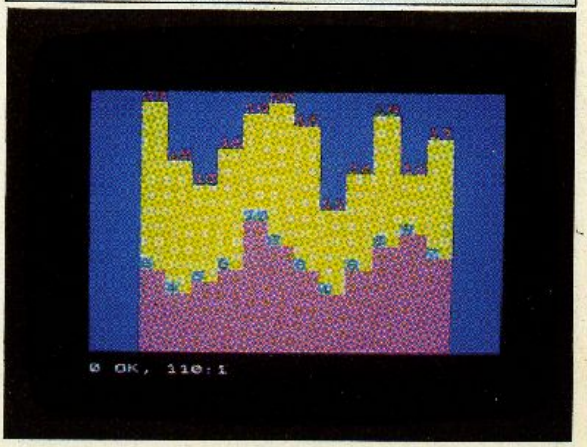
```
  10 BORDER 0: PAPER 1: CLS
  20 LET c=4
  30 FOR x=1 TO 12
  40 READ t
  50 FOR l=21 TO 21-t STEP -1
  60 PRINT PAPER 6;AT l,c;"  "
  70 NEXT l
  80 PRINT INK 2;AT 20-t,c;t
  90 LET c=c+2
 100 NEXT x
 110 DATA 20,15,13,16,19,20,16,1
1,12,19,14,17
```



Voeg nu de volgende lijnen toe en toets de nieuwe lijn 11Ø in zoals afgebeeld. De grafiek verschijnt nu in twee kleuren. Sla bladzijde 33 op om uit te vinden over READ en DATA.

DUBBELE BALK GRAFIEK

```
  85 READ t
  86 FOR l=21 TO 21-t STEP -1
  87 PRINT PAPER 3;AT l,c;"  "
  88 NEXT l
  89 PRINT INK 1; PAPER 5;AT 20-
t,c;t
 110 DATA 20,6,15,4,13,5,16,6,19
,10,20,8,18,6,11,4,14,6,19,8,14,
9,17,7
```



DOE-HET ZELF GRAFIEKEN

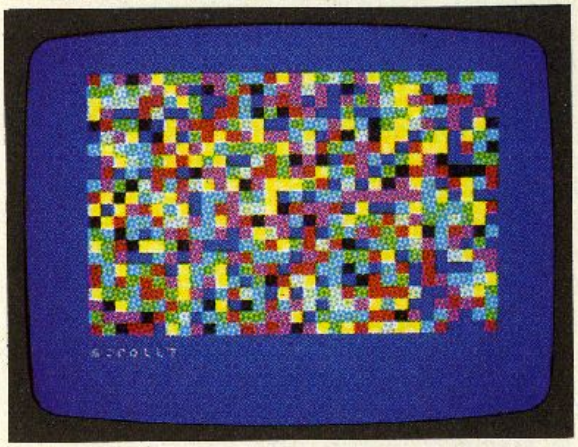
Je ZX Spectrum+ kan grafieken met lage resolutie en hoge resolutie geven. Beide soorten kunnen tegelijkertijd op het scherm verschijnen. Grafieken met lage resolutie bestaan op het scherm uit blokken van kleur. Op deze twee bladzijden zul je zien hoe je deze blokken via het toetsenbord produceert en hoe je ze op het scherm plaatst.

Het lage-resolutie-scherm

Op het lage-resolutie-scherm zijn er 32 horizontale posities voor tekens, en 22 vertikale. Elke scherm-positie heeft twee nummers om het te identificeren. Eerst komt het lijnnummer, dat de vertikale positie op het scherm aangeeft. De bovenste lijn is Ø en de onderste 21. Vervolgens komt het kolomnummer, dat de horizontale positie aangeeft. De meest linkse kolom is Ø, de meest rechtse 31. (op blz. 80 kun je de lay-out van het lage-resolutie-raster zien). Het volgende programma vult deze tekenposities in met kleur. Het keyword RND (op de R-toets) kiest een willekeurige inkt-kleur.

VIERKANTEN

```
10 BORDER 1: INK RND*7
20 PRINT "■";
30 GO TO 10
```



Hier verschijnen vierkanten over het gehele scherm. Om een teken in een bepaalde positie te laten verschijnen, moet je het keyword AT, samen met PRINT gebruiken. AT wordt geplaatst na PRINT en gevolgd door het lijnnummer, een komma, een kolomnummer en een punt-komma. Het volgende commando bijvoorbeeld laat een

PRINT AT 11,16;"*"

sterretje zien in lijn 11, kolom 16. Dit is in het midden van het scherm.

Hoe teken je regenboog-patronen

Een goede manier om gekleurde patronen te krijgen is om FOR..NEXT lussen in je grafiek-programma's te gebruiken. FOR..NEXT lussen zijn delen van een programma, die zichzelf een aantal keren herhalen.

Hoe selecteer je grafiek-tekens?

Je ZX Spectrum+ heeft een aantal grafiek-tekens op het toetsenbord, die het programmeren van grafieken met lage resolutie gemakkelijk maken. Je kunt ze zien op toetsen 1 tot 8. Om de grafiek-tekens op het scherm te krijgen moet je GRAPH intoetsen en vervolgens de toetsen 1 tot 8, met een spatie tussen elk van deze.

De grafiek-tekens verschijnen onder op het scherm. Het witte gedeelte van elk teken op de toets is de inktkleur, het zwarte gedeelte de papierkleur. Druk nu de toetsen opnieuw in, terwijl je tegelijkertijd CAPS SHIFT ingedrukt houdt. Deze keer verschijnen de tekens met de inkt – en papierkleuren omgedraaid. Dit is precies, hoe je grafiek-tekens in programma lijnen zet.

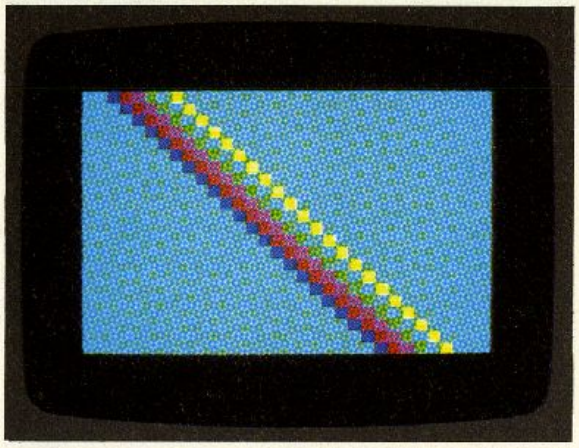
GRAPH
Deze toets wordt gebruikt om de Spectrum in de grafiek-modus te zetten.

Toets 8
Deze toets wordt vaak samen met GRAPH en CAPS SHIFT gebruikt om een heel gekleurd vierkant te produceren.

In de lijn, die begint met een lus kun je de computer opdragen hoeveel keer je de lus uitgevoerd wilt hebben. Terwijl het dit doet kan het gebruikt worden om bijvoorbeeld tekens op het scherm te plaatsen. Je bent niet beperkt tot één lus per programma. Je kunt een lus binnen een andere programmeren, vaak met zeer bruikbare resultaten. Het volgende programma laat zien hoe twee FOR..NEXT lussen (de ene "genesteld" in de andere) gebruikt kunnen worden om de kleuren en posities, geproduceerd door INK en AT, te veranderen. Aan het einde van deze bladzijde kun je zien hoe je deze lussen programmeert.

REGENBOOG

```
 5 BORDER 0: PAPER 5: CLS
10 LET x=1
20 FOR l=0 TO 21
30 FOR c=1 TO 6
40 PRINT INK c;AT l,c+x;"■"
50 NEXT c
60 LET x=x+1
70 NEXT l
```

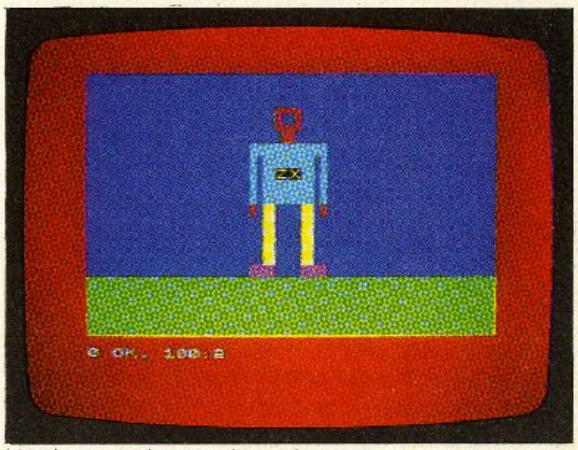


Het programmeren van plaatjes

In grafieken met lage resolutie kun je plaatjes "schilderen" door de posities en kleuren van de grafiek-tekens uit te werken. Je kunt je eigen plaatje ontwerpen door het lage_resolutie_raster te gebruiken, op blz. 80. Toets vervolgens, door de grafiek-tekens te selecteren zoals op de blz. hiernaast is aangegeven, de programmalijnen één voor één in. Het volgende programma laat zien wat je kunt produceren. Alle vormen hierin vind je op de nummertoetsen. Je kunt wachten met het afdraaien van het programma, totdat je alle lijnen hebt ingetoetst, maar als je het na elke ingetoetste lijn afdraait, kun je zien hoe de verschillende delen van de robot opgebouwd worden. (Onthoud, dat als je de verkeerde grafiektekens intoetst, je ze precies zo corrigeert als bij nummers of letters).

ZX ROBOT

```
  5 BORDER 2: PAPER 1: CLS
 10 PRINT INK 2;AT 3,15;"■■"
 15 PRINT INK 2;AT 4,15;"■■"
 20 PRINT INK 2;AT 5,15;"■■"
 30 PRINT INK 5;AT 6,13;"■■■■■"
"
 40 FOR l=7 TO 10: PRINT INK 5;
AT l,13;"■■■■■■■": NEXT l
 45 PRINT INK 6; PAPER 0;AT 8,1
5;"ZX"
 50 PRINT INK 2;AT 11,13;"■■■
■"
 60 FOR l=11 TO 15: PRINT INK 6
;AT l,14;"■ ■": NEXT l
 70 PRINT INK 3;AT 16,13;"■■■";T
AB 17;"■■■"
 80 FOR l=17 TO 21: FOR c=0 TO
31
 90 PRINT INK 4;AT l,c;"■"
100 NEXT c: NEXT l
```



Het keyword TAB, dat in lijn 7Ø verschijnt na PRINT, wordt gebruikt om een teken te plaatsen in de lijn waar de computer op dat moment mee aan het werk is. TAB wordt gevolgd door een getal van Ø tot 31, om de kolom-positie te bepalen.

HOE GEBRUIK JE FOR..NEXT LUSSEN

Een FOR..NEXT lus begint altijd met een lijn, die de keywords FOR en TO bevat, samen met een variabele met zijn begin – en eindwaarde, bijvoorbeeld;

3Ø FOR C=1 TO 6

De variabele hier is C. De lus, die hiermee begint bevat dan lijn(en), die de computer een bewerking laten herhalen. Deze lijn(en) kunnen ook zelf de variabele C gebruiken. FOR..NEXT lussen eindigen altijd met het keyword NEXT en de variabele, jibvoorbeeld:

5Ø NEXT C

Wanneer het programma wordt afgedraaid, wordt de hele lus van FOR tot NEXT een bepaald aantal keren herhaald. De variabele begint met de eerste waarde voor TO wordt elke keer vermeerderd met 1 totdat het de limiet na TO bereikt. In dit geval herhaalt de lus zich 6 keer, met C beginnend bij 1, vervolgens vermeerderd tot 2, 3, 4, 5 en tenslotte 6. In het eerste programma op bladzijde 25 worden drie lussen in een "nest" gebruikt. Dit betekent, dat bij elke cyclus van de "buitenste" lus, de "middelste" lus door al zijn cyclussen gaat. De "binnenste lus" gaat het vaakst door zijn cyclussen, namelijk elke keer, dat de "middelste" lus door een cyclus gaat. Je kunt een nest zo diep maken als je wilt, maar denk eraan, dat "genestelde" lussen elkaar nooit mogen overlappen.

HET SCHERM-SCHETSBOEK

Grafische afbeeldingen op de ZX Spectrum+ zijn niet beperkt tot patronen en plaatjes met lage resolutie. Je kunt je Spectrum+ gebruiken om gedetailleerde beelden met scherpe omtrekken, en rechte of gebogen lijnen en hoeken te creëren.

Grafische afbeeldingen met hoge resolutie bestaan uit vele stippen, die naast elkaar geplaatst worden om een lijn te vormen of een vorm met kleur in te vullen. Elke stip is 1/64 deel van de grootte van de vierkanten, die je bij lage-resolutie-grafieken gebruikt. als je het volgende commando

PLOT 128,87

intoest, zie je zo'n stip in het centrum van het beeldscherm. De stippen, die bij hoge-resolutie-grafieken gebruikt worden, worden "pixels" genoemd. Dit is een afkorting voor beeldcellen. Net zoals bij lage-resolutie-tekens, heeft elke pixel ook twee nummers ndig om zijn positie te bepalen.

Het hoge-resolutie-raster.

Het hoge-resolutie-raster bevat 256 pixels horizontaal en 176 vertikaal. Anders dan bij de lage resolutie-afbeeldingen echter, geeft het eerste nummer de horizontale positie aan. Deze positienummers lopen van Ø aan de linkerkant van het scherm tot 255 aan de rechterkant. Het tweede nummer is het vertikale positienummer, maar de nummers lopen van Ø onderaan het scherm tot 175 bovenaan. Positie Ø,Ø is de linker beneden-hoek en niet zoals bij lage resolutie de linkerbovenhoek. Op blz. 80 vind je een schema voor het hoge-resolutie-raster.

Het maken van grafieken en tekeningen

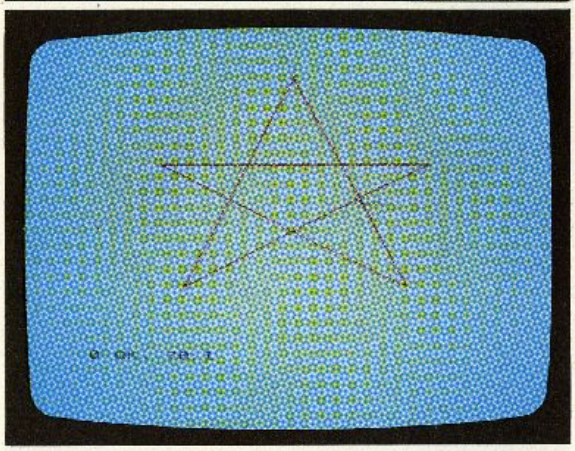
Je hebt slechts drie keywords nodig voor het maken van hoge-resolutie-grafieken: PLOT, DRAW, en CIRCLE. PLOT wordt gevolgd door het horizontale en vertikale positie- nummer, gescheiden door een komma, en het plaatst de pixel in deze positie. DRAW wordt ook gevolgd door twee nummers, maar dit zijn geen positie-nummers. In plaats daarvan geven ze de afstand tussen twee pixelposities aan, horizontaal en vertikaal. DRAW trekt dan een lijn tussen die twee posities.

De eerste positie is Ø,Ø als PLOT of DRAW nog niet in het programma gebruikt zijn. Als ze wel gebruikt zijn, is deze positie de laatste PLOT-positie of de laatste positie bereikt door DRAW, afhankelijk van welke het meest recent is.

De DRAW-statement trekt vervolgens een lijn naar de nieuwe positie. Probeer het volgende programma.

STER

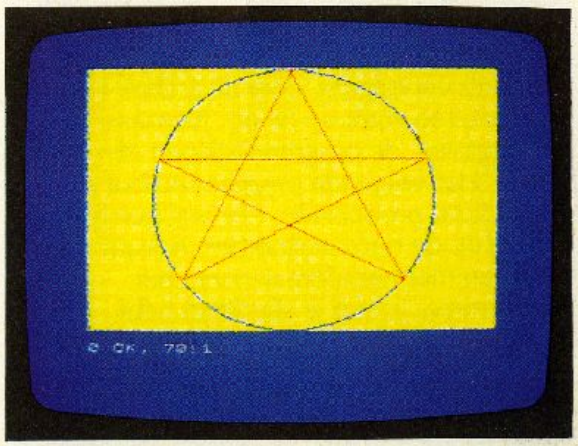
```
10 INK 2
20 PLOT 128,174
30 DRAW 70,-140
40 DRAW -152,80
50 DRAW 164,0
60 DRAW -150,-80
70 DRAW 70,140
```



PLOT beweegt de start- positie naar de bovenkant van het scherm. De vijf DRAW-statements trekken vervolgens de vijf rode lijnen. Voeg nu de volgende lijnen toe aan het programma:

4 BORDER 1: PAPER 6: INK 1:CLS
5 CIRCLE 128,87,87

Draai het programma opnieuw af. De rode ster verschijnt nu in een cirkel op gekleurd papier.



CIRCLE heeft drie waarden nodig. De eerste twee geven de positie van het middelpunt van de cirkel aan. Het derde nummer is de straal. Je kunt nog een derde waarde aan de DRAW-statements toevoegen.

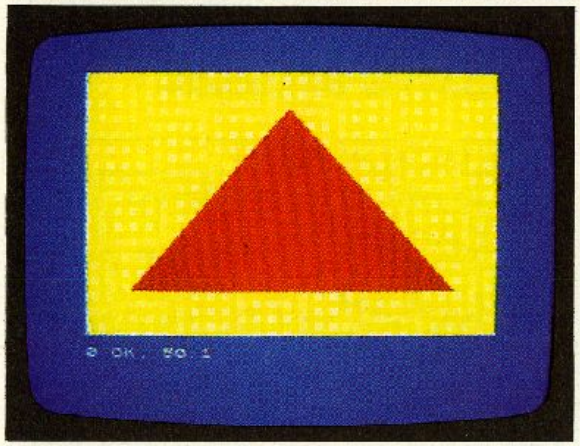
Probeer waarden tussen 2 en –2 en kijk wat er gebeurt.

Hoe vul je vormen op.

Je kunt gemakkelijk opgevulde vormen in hoge resolutie produceren door vele lijnen dichtbij elkaar te trekken. Dit kan gedaan worden met een FOR..NEXT-lus, die de DRAW-posities verandert zodat ze elke keer met 1 vermeerderd worden.

OPGEVUIDE DRIEHOEK

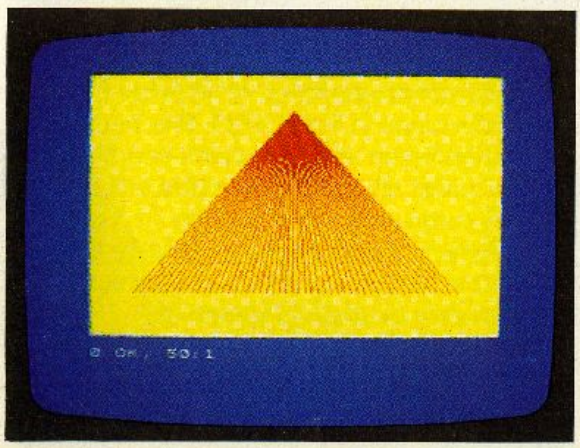
```
  10 BORDER 1: PAPER 6: INK 2: C
LS
  20 FOR x=-100 TO 100
  30 PLOT 128,150
  40 DRAW x,-120
  50 NEXT x
```



Je krijgt een interessant effect als je lijnen een klein stukje van elkaar af trekt. Dit kun je doen door het keyword STEP en een nummer aan de FOR-statement toe te voegen. Toets een andere lijn 20 in en draai het programma opnieuw af.

2Ø FOR X+1ØØ TO 1ØØ STEP 4

Deze keer verschijnt het waaiervormige figuur, dat hieronder is afgebeeld. De reden is:STEP vermeerdert X steeds met sprongen van 4 i.p.v. met 1.

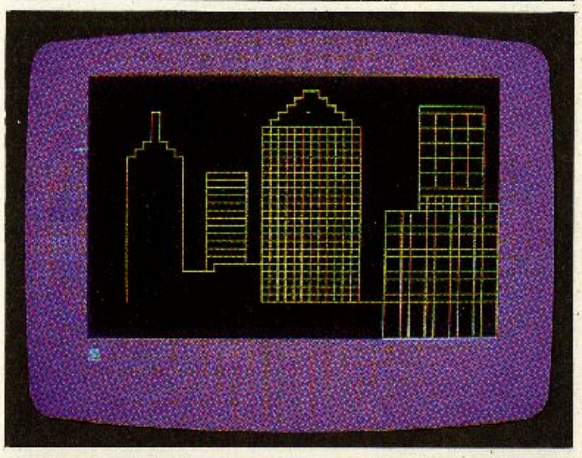


Je scherm-schetsboek

Als je een plaatje of patroon wilt maken hoef je niet iedere keer een programma te schrijven. In plaats daarvan kun je een programma gebruiken, waarmee je een plaatje direct op het scherm opbouwt Het begint met het keyword INPUT, dat vraagt om een inkt-nummer. Daarna (weer m.g.v. INPUT; deze keer met een $-teken om een string te labelen) trekt de computer korte lijnen, iedere keer als je één van vier bepaalde toetsen indrukt u,d,l of r, gevolgd door ENTER. Het gebruikt IF en THEN om beslissingen te nemen.

SCHETSBOEK

```
  10 INPUT "INK ";i
  20 BORDER 3: PAPER Ø: INK i: C
LS
  30 PLOT 25,25
  40 LET x=5
  50 INPUT k$
  60 IF k$="u" THEN DRAW Ø,x
  70 IF k$="d" THEN DRAW Ø,-x
  80 IF k$="r" THEN DRAW x,Ø
  90 IF k$="l" THEN DRAW -x,Ø
 100 GO TO 50
```



Het nemen van beslissingen met IF en THEN

De lijnen 6Ø en 9Ø in het SKETCHPAD programma bevatten IF en THEN-statements. Deze laten je Spectrum+ beslissingen nemen. In dit geval controleert de computer of de ingetoetste letter een u, d, l or r is. Als (IF) één van deze letters is ingetoetst, dan (THEN) wordt de computer opgedragen om een lijn naar boven, naar beneden, naar links of naar rechts te trekken. Het trekt geen lijn als een hoofdletter is ingetoetst.

IF wordt altijd gevolgd door iets dat de Spectrum+ controleert, om te zien of het waar is, of het gebeurt (hier om te zien of bepaalde toetsen zijn ingedrukt). Als het waar is, of gebeurt wordt de actie na THEN uitgevoerd. Alles wat na THEN komt, is afhankelijk van de beslissing. Als het niet waar is of niet gebeurt, gaat het programma naar de volgende lijn.

11Ø IF b=5 THEN PRINT "*" : GOTO 2ØØ

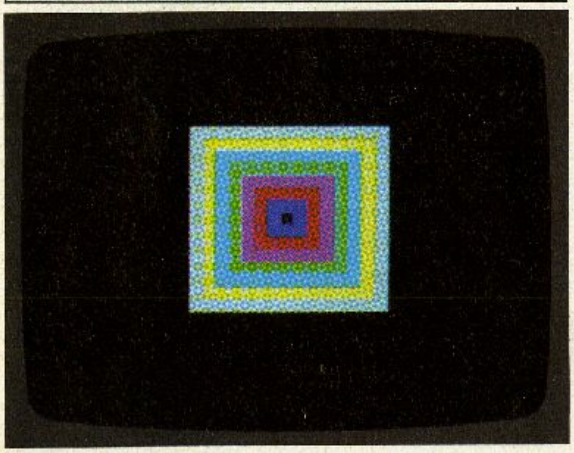
de computer gaat maar tot lijn 2ØØ als b 5 is.

HET ONTWERPEN VAN PATRONEN EN PLAATJES

Je kunt met je ZX Spectrum+ allerlei patronen en plaatjes produceren, met gebruik van lage-resolutie-grafieken, hoge-resolutie-grafieken of beide. De beste manier is om eerst je ontwerp uit te tekenen op een copie van de rasters op blz. 80. Voor het tekenen van patronen en plaatjes kun je vaak FOR..NEXT lussen gebruiken, die een deel van een programma een bepaald aantal keren herhalen. Elke keer kunnen de posities en kleuren van de tekens of lijnen veranderen, gewoonlijk op een regelmatige manier. Hier is een programma, dat deze techniek gebruikt.

VIERKANTEN

```
10 BORDER 0: PAPER 0: CLS
20 FOR x=7 TO .0 STEP -1
30 INK x
40 FOR l=11-x TO 11+x
50 FOR c=16-x TO 16+x
60 PRINT AT l,c;"■"
70 NEXT c
80 NEXT l
90 NEXT x
```



Dit programma bevat drie FOR..NEXT lussen. De x-lus verandert de kleur en ook de grootte van de grote vierkanten, die geproduceerd worden, terwijl de l-lus en de c-lus de lijn en kolompositie van de kleine vierkantjes verandert.

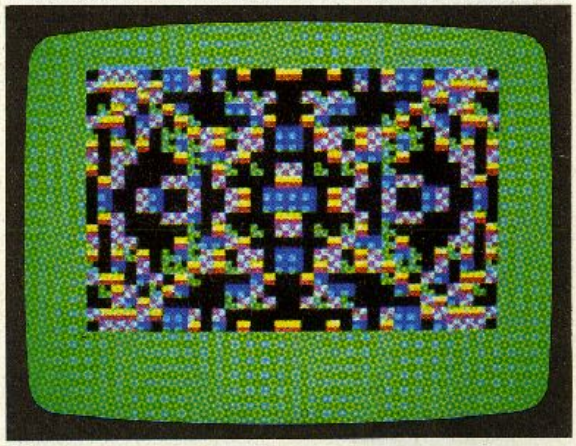
Willekeurige effecten en subroutines

Het gebruik van lussen hoeft niet steeds, als het programma afgedraaid wordt, identieke patronen te geven. Door het keyword RND (afkorting voor RaNDom+willekeurig) in lussen te gebruiken kun je kleuren, posities en andere display-eigenschappen iedere keer anders maken.

Kijk naar het mozaiek programma op blz. 10. Het werkt, omdat de inktkleur RND*7 is, wat betekent elk getal met een decimaal-punt van Ø tot 7. INK rondt dit af naar het dichtstbijzijnde hele getal. Dus, elke keer, dat een vierkant getoond wordt is de kleur een willekeurige kleur van INK Ø tot 7. Het volgende programma tekent symmetrische patronen op het scherm. Het gebruikt RND om de tekens en hun posities te veranderen. De variabelen i en p geven de inkt- en papierkleuren en a geeft aan hoeveel patronen er getekend worden (in dit geval vier). De variabele n geeft het aantal tekens in elk patroon, terwijl x een willekeurig nummer is van 129 tot 142. De statement GOSUB 1ØØØ in lijn 5Ø stuurt de computer naar een subroutine.

SYMMETRICAL PATTERNS

```
  10 BORDER 4: PAPER 0: CLS
  20 LET i=4: LET p=0
  30 FOR a=1 TO 4
  40 LET x=RND*13+129
  50 FOR n=1 TO 40: GO SUB 1000:
NEXT n
  60 LET i=i+1: LET p=p+1
  70 PAUSE 100
  80 NEXT a
  90 STOP
1000 LET l=INT (RND*11)
1010 LET c=INT (RND*16)
1020 INK i: PAPER p
1030 PRINT AT l,c;CHR$ x
1040 PRINT AT l,31-c;CHR$ x
1050 PRINT AT 21-l,c;CHR$ x
1060 PRINT AT 21-l,31-c;CHR$ x
1070 BEEP 0.01,l*c/3
1080 RETURN
```



Een subroutine is een groep van lijnen, die werkt als een programma. In dit programma is de subroutine in lijn 1ØØØ. Hte beeldt een grafiekteken af in vier delen van het scherm zodat elk zich op dezelfde afstand van het centrum bevindt (positie 11,16). Deze afstand wordt gegeven door de lijnen 1ØØØ en 1Ø1Ø; 1 geeft de afstand tussen de lijnen en c de afstand tussen de kolommen. INT rondt het willekeurige nummer af tot een heel nummer, zodat 1 elk nummer van Ø tot 1Ø kan zijn en c elk nummer van Ø tot 15.

De lijnen 1Ø3Ø tot 1Ø6Ø beelden vervolgens het grafiekteken af, waarvan de code x is (zie het grafiektekenschema op blz. 51) BEEP maakt een geluid, waarvan de hoogte gerelateerd is aan de positie. RETURN in lijn 1Ø8Ø stuurt het programma terug naar de volgende statement na GOSUB in lijn 5Ø. Lijn 6Ø verandert de inkt- en papierkleuren. PAUSE 1ØØ in lijn 7Ø vertraagt het programma voor 2 seconden voordat het terug "lust" naar het begin. STOP in lijn 9Ø is nodig om te vermijden, dat het programma na de vierde lus in de subroutine loopt.

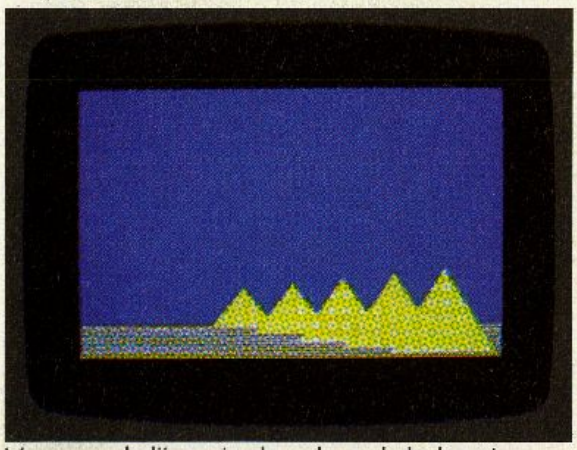
Je kunt dit programma veranderen door 4 in lijn 3Ø en 4Ø in lijn 5Ø te veranderen in andere nummers. Als je het bereik van x in lijn 4Ø wijder maakt krijg je andere tekens op het scherm.

Het gebruik van FOR..NEXT-lussen in grafieken.

FOR..NEXT-lussen kunnen zeer effectief gebruikt worden in hoge-resolutie-grafieken, om plaatjes te creëren. Toets het volgende programma in en draai het af. Met gebruik van slechts PLOT en DRAW trekken de twee FOR..NEXT-lussen eerst lijnen op de grond en vervolgens vijf opgevulde driehoeken of pyramides.

PYRAMIDEN

```
 10 BORDER 0: PAPER 1: INK 6
 20 CLS
 30 FOR y=0 TO 20 STEP 2
 40 PLOT 0,y
 50 DRAW 255,0
 60 NEXT y
 70 FOR n=100 TO 220 STEP 30
 80 FOR x=-10-n/10 TO 10+n/10
 90 PLOT n,35+n/10
100 DRAW x,-n/4
110 NEXT x: NEXT n
```



Voeg nu de lijnen in de volgende kolom toe. Wanneer je het opnieuw afdraait, zie je een laserstraal, die voortdurend opschiet in de nachtelijke hemel, daarbij uitbarstingen van sterren creërend. Het wordt getekend uit de hoek van het scherm naar positie x,y, waarbij de variabelen x en y willekeurige nummers zijn.

```
120 LET x=RND*255
130 LET y=RND*104+71
140 LET l=INT ((175-y)/8)
150 LET c=INT (x/8)
160 PLOT 0,0: DRAW OVER 1;x,y
170 BEEP 0.01,x/4
180 PLOT 0,0: DRAW OVER 1;x,y
190 PRINT AT l,c;"*"
200 GO TO 120
```

Merk het gebruik van OVER 1 op in lijn 16Ø en 18Ø, die identiek zijn. OVER 1 laat de eerste lijn de laserstraal tekenen terwijl de tweede lijn die weer verwijdert zonder de rest van het plaatje te veranderen. Bewaar dit programma omdat je het later weer nodig hebt.

FLASH, BRIGHT en INVERSE

Deze drie keywords kunnen de kleuren van de Spectrum+ pas echt aan het werk zetten. Elk keyword wordt gevolgd door ofwel Ø of 1 en je kunt ze in PRINT-statements zetten mits je na Ø of 1 een punt-komma zet. FLASH 1 laat de teken-posities tussen de inkt- en papierkleuren flitsen terwijl BRIGHT 1 de kleuren helderder maakt. INVERSE 1 verandert de inktkleur in de papierkleur en vice versa. Het gebruik van Ø na deze keywords herstelt de display. Probeer de veranderingen op deze twee bladzijden uit om te zien hoe de keywords werken. Verander in het VIERKANTEN- programma het vierkant in lijn 60 in een ster en voeg toe:

15 INVERSE 1

De sterren verschijnen nu in het zwart (de papierkleur) tegen gekleurde banden (de veranderende inktkleuren). Toets INVERSE Ø in voordat je doorgaat.

Verander in het SYMMETRISCHE PATRONEN programma de volgende lijnen om te zien hoe BRIGHT en FLASH werken.

15 BRIGHT 1
16 FLASH 1

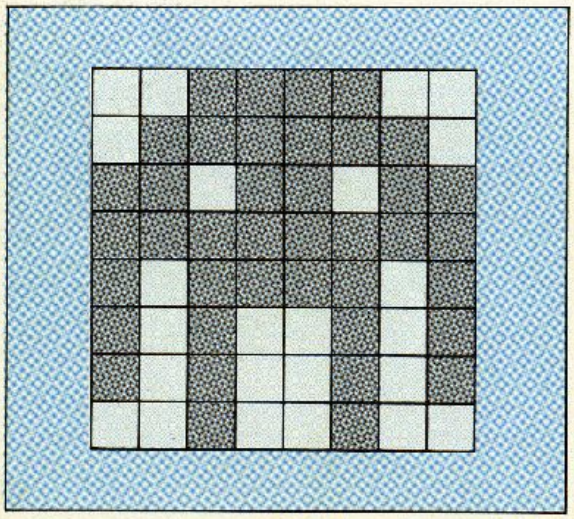
Merk op hoe FLASH er voor zorgt, dat het patroon heen en weer lijkt te bewegen. Toets FLASH Ø:CLS in om het flitsen te stoppen.

HOE MAAK JE COMPUTER-TEKENS?

Je ZX Spectrum+ is niet beperkt tot alleen de grafiektekens, die je via het toetsenbord intoetst. In een speciaal deel van het geheugen kan het andere tekens opslaan, die je zelf ontwerpt. Deze heten "user-defined graphics"-tekens, en elk programma kan een maximum van 21 van deze tekens bevatten. Elk teken bestaat uit 64 kleine stippen of pixels van inkt-kleur. Deze zijn opgezet in acht rijen van acht pixels elk. Elk teken bezet een teken-positie op het lage-resolutie-raster.

Het ontwerpen van een grafiek-teken

Teken eerst een 8 x 8-raster zoals hieronder afgebeeld. Vul dan enkele vierkanten op om het teken te maken. Deze vierkanten vertegenwoordigen de inkt-gekleurde pixels. Teken dan in of beschouw elk vol vierkant als 1 en elk leeg vierkant als Ø. Hier is een ontwerp voor een spin.



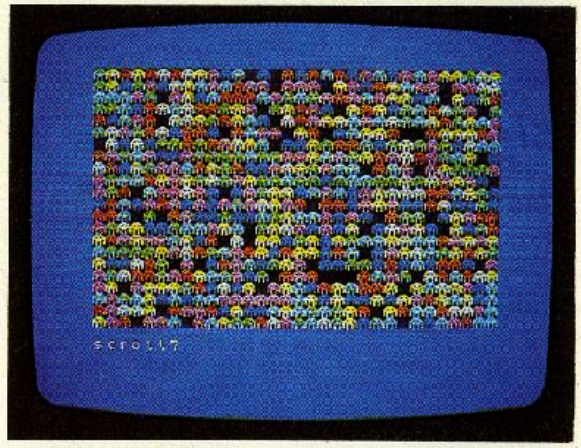
Elk van de user-defined graphics-tekens wordt geïdentificeerd door een letter van a tot u (of A tot U- dat maakt geen verschil). Om het teken te programmeren toets je acht POKE USR statements in, elk eindigend met BIN, gevolgd door een tweetallig nummer bestaande uit de acht Ø-en en 1-en in elke rij van het raster. Laten we het spin-teken s noemen.

```
10 POKE USR "s",BIN 00111100
11 POKE USR "s"+1,BIN 01111110
12 POKE USR "s"+2,BIN 11011011
13 POKE USR "s"+3,BIN 11111111
14 POKE USR "s"+4,BIN 10111101
15 POKE USR "s"+5,BIN 10100101
16 POKE USR "s"+6,BIN 10100101
17 POKE USR "s"+7,BIN 00100100
```

Draai nu dit programma af en druk dan GRAPH en S in. In plaats van S verschijnt de spin! Voeg de volgende lijnen toe, met behoud van de spin in lijn 3Ø en draai het programma af. Spinnen verschijnen nu over het hele scherm.

SPINNEN

```
20 BORDER 1: PAPER 0: CLS
30 PRINT INK RND*7;"s";
40 GO TO 30
```



Onthoud bij het ontwerpen van je eigen tekens, dat je ze niet op het scherm kunt zien, vóórdat je een programma afgedraaid hebt, dat ze bepaalt. Tot dan toe verschijnen ze alleen in listings als letters.

Hoe "meng" je kleuren met gespikkelde vierkanten.?

Je kunt op je Spectrum + heel gemakkelijk gemengde kleuren simuleren. Om dit te doen moet je een teken maken, dat 50 procent inkt-kleur en 50 procent papier-kleur geeft bij gebruik. Je hoeft alleen maar twee pixel-lijnen te bepalen en dan de

```
10 FOR x=0 TO 6 STEP 2
20 POKE USR "a"+x,BIN 10101010
30 POKE USR "a"+x+1,BIN 01010101
40 NEXT x
```

computer te instrueren om die afwisselend in een teken te gebruiken.
Als je het programma afdraaid moet je een gestippeld vierkant zien. Als je dezelfde techniek gebruikt in een programma dat kleur-keywords gebruikt, zal het gestippelde vierkant een kleur produceren die een mengsel is van de papier en inkt kleur in het programma.

Het vereenvoudigen van grafieken met READ en DATA

Er is een gemakkelijker manier om je grafiektekens te maken, en dat is decimale getallen te gebruiken met READ en DATA. Verander eerst de acht tweetallige nummers in decimale getallen. Doe dit door PRINT BIN in te toetsen, gevolgd door een getal, bijvoorbeeld:

PRINT BIN ØØ1111ØØ

De Spectrum+ beeldt 6Ø af, de decimale equivalent voor ØØ1111ØØ. Bij de spin zijn de acht decimale getallen 6Ø, 126, 219, 255, 189, 165, 165 en 36.

Nu kun je READ en DATA gebruiken. Deze twee keywords voorzien je van een gemakkelijke manier om vele waarden (zoals nummers) in te voeren in de variabelen in een programma. READ wordt gevolgd door een variabele: één of meer letters als je te doen hebt met nummers, of een enkele letter gevolgd door $ als je te doen hebt met strings.

Wanneer je Spectrum+ READ tegenkomt, zoekt het naar de eerste DATA-statement in het programma. Deze statement bevat een lijst van waarden gescheiden door komma's. De computer neemt de eerste waarde van de lijst en deze wordt aan de variabele na READ gegeven. De volgende keer dat de computer READ tegenkomt wordt de tweede waarde aan de variabele gegeven enz., precies op volgorde.

Hier is een nieuw programma, dat de spin produceert.

```
 10 FOR x=0 TO 7
 20 READ y
 30 POKE USR "s"+x,y
 40 NEXT x
 50 DATA 60,126,219,255,189,165
,165,36
```

Het programma slaat feitelijk elke willekeurige acht decimale nummers op in het geheugen, om een teken te maken. Verander s in lijn 3Ø in de letter, die je wilt en toets na DATA in lijn 5Ø, de acht nummers in, elk gescheiden door een komma.

Druk op GRAPH en de letter om het teken te verkrijgen, nadat je het programma hebt afgedraaid.

Het tekenen van een schaakbord

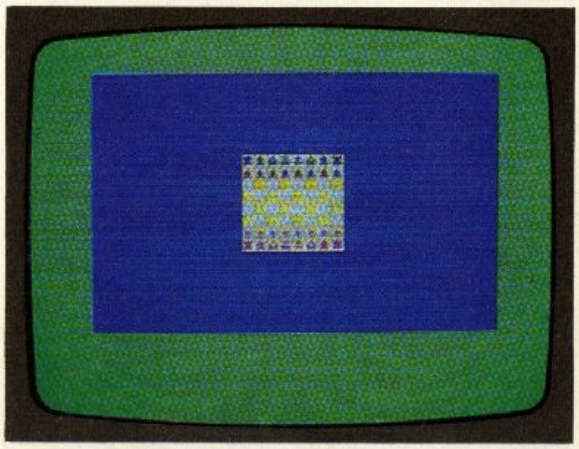
Hier is een programma, dat een schaakbord op het scherm afbeeldt en dan de stukken uitlegt, klaar voor een spelletje. Je kunt kleuren laten verschijnen in een listing m.g.v. kleurregelingscodes – zie het schema hieronder.

SCHAAKBORD

```
 10 FOR x=1 TO 8
 20 READ a$
 30 GO SUB 500
 40 NEXT x
 50 BORDER 4: PAPER 1: CLS
 60 FOR l=7 TO 14 STEP 2: FOR c
=12 TO 19 STEP 2
 70 PRINT AT l,c;" ";AT l+1,c;
"  "
 80 NEXT c: NEXT l
 90 PRINT AT 7,12;"RNBQKBNR"
100 PRINT AT 8,12;"PPPPPPPP"
110 PRINT AT 13,12;"pppppppp"
120 PRINT AT 14,12;"rnbqkbnr"
130 GO TO 130
500 FOR n=0 TO 7
510 READ y
520 POKE USR a$+n,y
530 NEXT n
540 RETURN
```

De stukken worden bepaald door een subroutine, die begint in lijn 5ØØ.

```
 550 DATA "p",0,0,16,56,56,16,12
4,0
 560 DATA "r",0,84,124,56,56,124
,124,0
 570 DATA "n",0,16,56,120,24,56,
124,0
 580 DATA "b",0,16,40,68,108,56,
124,0
 590 DATA "k",0,16,56,16,56,68,5
6,0
 600 DATA "q",0,84,40,16,108,124
,124,0
```



Het gebruik van kleurregelingscodes

In plaats van het gebruik van keywords zoals INK en PAPER, kun je regelingscodes zetten in PRINT-statements na de eerste aanhalingstekens. De tekens veranderen dan van kleur in de listing zelf en verschijnen ook in deze kleuren op het scherm.

Druk, om de codes te krijgen eerst EXTEND MODE in, en dan een nummer-toets met of zonder CAPS SHIFT. Dit schema laat zien hoe je elk van deze kleuren kunt selecteren.

DRUKKEN ⇨	1	2	3	4	5	6	7	8	9	0
EXTEND MODE + CAPS SHIFT	BLUE INK	RED INK	MGNTA INK	GREEN INK	CYAN INK	YELLW INK	WHITE INK	FLASH OFF	FLASH ON	BLACK INK
EXTEND MODE	BLUE PAPER	RED PAPER	MGNTA PAPER	GREEN PAPER	CYAN PAPER	YELLW PAPER	WHITE PAPER	BRIGHT OFF	BRIGHT ON	BLACK PAPER

ANIMATIE

Computer graphics zien er het best uit als ze over het scherm bewegen. Het produceren van animatie op je Spectrum is niet moeilijk. Alles wat je hoeft te doen, is steeds de positie van een teken of lijn te veranderen. De beste manier om dit te doen is door één of meer FOR..NEXT-lussen te gebruiken.

Vertikale en horizontale beweging

Toets dit programma in en draai het af. Als je niet op de reset-knop hebt gedrukt of de Spectrum hebt uitgezet sinds het produceren van het spinteken op blz. 32 hoef je de lijnen 10 tot 50 niet in te toetsen. Het grafiek-teken is nog in het geheugen onder "s".

VALLENDE SPIN

```
  5 BORDER 3: PAPER 5: CLS
 10 FOR x=0 TO 7
 20 READ y
 30 POKE USR "s"+x,y
 40 NEXT x
 50 DATA 60,126,219,255,189,165
,165,36
 60 FOR x=0 TO 7
 70 READ y
 80 POKE USR "t"+x,y
 90 NEXT x
100 DATA 16,16,16,16,16,16,16,1
6
110 FOR l=0 TO 20
120 PRINT AT l,3; INK 0;"|"
130 PRINT AT l+1,3; INK 2;"■"
140 NEXT l
```

Elke keer, dat je het programma afdraait, valt de spin aan zijn draad op het scherm naar beneden.

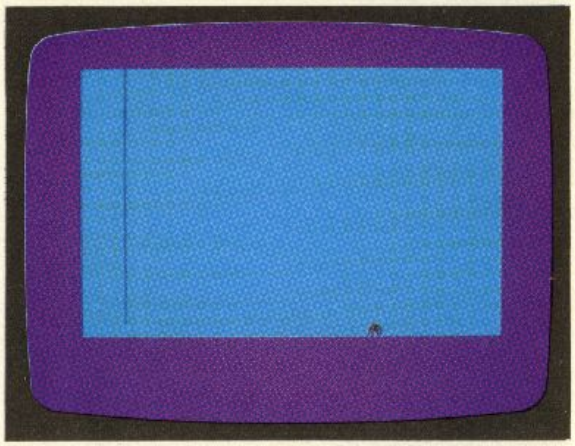
De lijnen 60 tot 100 produceren een ander grafisch teken ("t") voor de draad. De animatie heeft plaats in lijnen 110 tot 140, die een FOR.NEXT-lus vormen waarin 1 (het lijnnummer) varieert van 0 tot 20. Elke keer, dat de lus zich herhaalt, wordt een bepaalde lengte van draad in een positie afgebeeld en de spin in de volgende positie eronder. De volgende keer wordt de spin vervangen door nog een stuk draad en de spin verschijnt er weer onder. Op deze manier valt de spin snel langs zijn draad naar beneden, totdat hij lijn 21 bereikt, de onderkant van het beeldscherm. Om de actie te vertragen, kun je de volgende lijn toevoegen.

135 PAUSE 10

Dit laat de computer een-vijfde seconde wachten elke keer vóór de spin in de volgende positie afgebeeld wordt. Voeg nu deze extra lijnen aan het programma toe en draai het opnieuw af. Je zult de animatie zien, maar in een andere richting.

RENNENDE SPIN

```
150 FOR c=3 TO 30
160 PRINT AT 21,c;" "
170 PRINT AT 21,c+1; INK 2;"■"
180 NEXT c
```



Nu snelt de spin naar één kant, zodra hij de onderkant bereikt. De extra lijnen vormen een andere FOR..NEXT-lus, die de kolompositie c regelt. Merk op, dat eerst een spatie wordt afgebeeld en dan de spin in de volgende kolompositie. Dit laat de spin verwijnen uit een positie en verschijnen in de volgende, bewegend naar rechts. Het is altijd beter een teken uit te wissen, voordat je het print in de volgende positie. Dit helpt om flikkeren in bewegende beelden te vermijden of te verminderen.

Doel-oefeningen

In veel computer spelletjes, doet de actie zich voor wanneer twee bewegende vormen botsen of als een object getroffen wordt door een straal.

Het waarnemen van botsingen is niet moeilijk. Als twee tekens zijn afgebeeld op positie 1,c (voor lijn en kolom) en positie v,h (voor vertikaal en horizontaal), dan moeten ze als 1=v en c=h, in dezelfde positie zijn. Je kunt dit als een statement schrijven, bijvoorbeeld:

160 IF 1=v AND c=h THEN PRINT "CRASH!"

Een andere manier om botsingen te vinden is om kleur te gebruiken. Verwijder het spin-programma door NEW in te toetsen. Toets dan het hele PYRAMIDS-programma op blz. 31 in of laad het programma als je het bewaard hebt op tape. Je kunt het nu combineren met je spin (nog in het geheugen tenzij je reset

hebt ingedrukt of de computer uitgezet hebt) om een nieuw programma te produceren. Voeg eerst de volgende lijnen toe, die een explosie-graphic creëren, genoemd "e".

```
    5 FOR x=0 TO 7
    6 READ y
    7 POKE USR "e"+x,y
    8 NEXT x
    9 DATA 145,82,44,121,158,52,7
4,137
```

Wis nu lijn 19Ø uit en voeg de volgende lijnen toe.

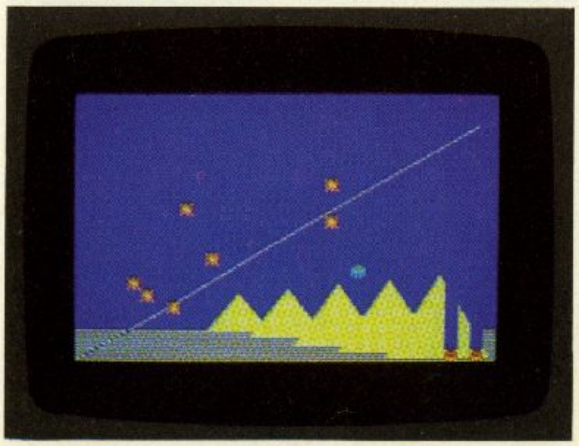
SPINNEN EN PYRAMIDEN

```
  114 LET h=RND*31
  115 FOR v=0 TO 20
  117 PRINT AT v,h;" ";AT v+1,h;
INK 4;"S"
  200 NEXT v
  205 PRINT AT 21,h; FLASH 1; INK
 2; PAPER 6;"S"
  210 GO TO 114
```

De uitbarstingen van sterren verschijnen niet meer. In plaats daarvan vallen spinnen door de lucht en vreten de pyramides en de grond weg. Wat je gedaan hebt is het toevoegen van een FOR..NEXT-lus, waarin v en h de positie van de spin geven. De variabele h is willekeurig, zodat de spinnen op verschillende plaatsen over het scherm verschijnen. Voeg nu de volgende lijnen toe.

ONTPLOFFENDE SPINNEN

```
  190 IF ATTR (v+1,h)=14 THEN GO
TO 500
  500 PRINT AT v+1,h; FLASH 1; PA
PER 2;"E"
  510 PAUSE 100
  520 GO TO 114
```



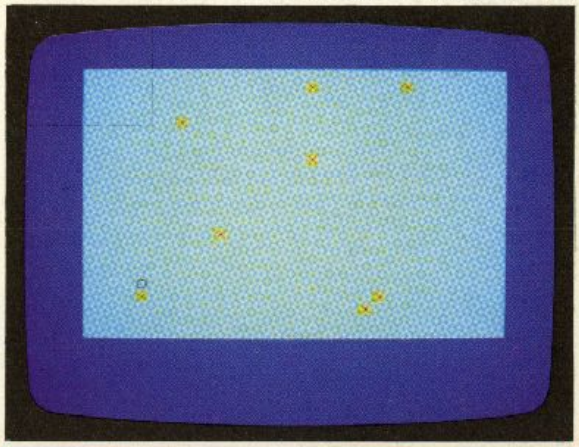
Wanneer de laserstralen de spinnen treffen, worden ze even geel. Wanneer de lijn, geproduceerd door DRAW, een spin-teken-positie binnenkomt verandert de inkt-kleur in dezefde kleur als de lijn, die geel is. In lijn 19Ø, neemt ATTR waar, of de spin geel wordt en stuurt dan de computer naar de explosie-subroutine in lijn 5ØØ.

Een bal stuiten

Vele grafische programma's hebben vormen, die van de kanten van het scherm afstuiten. Dit programma laat zien hoe dat gedaan wordt. De variabelen v en h werken op de dezelfde manier als in het EXPLODING SPIDERS-programma, maar +1 of −1 wordt toegevoegd aan v of h om de bal naar beneden, boven links of rechts te laten gaan. SCREENS controleert of er een X op positie v, h is.

BOUNCING BALL

```
   10 BORDER 1
   20 FOR z=1 TO 10
   30 LET h=INT (RND*25): LET v=I
NT (RND*21)
   40 PRINT INK 2; PAPER 6; FLASH
 1;AT v,h;"X"
   50 NEXT z
   60 LET x=1: LET y=1
   70 PRINT AT v,h;" "
   80 LET v=v+y: LET h=h+x
   90 IF h=0 OR h=31 THEN LET x=-
x: BEEP .2,24
  100 IF v=0 OR v=21 THEN LET y=-
y: BEEP .2,12
  110 IF SCREEN$ (v,h)="X" THEN P
RINT INK 1; PAPER 5;AT v,h;"!*!*
!": STOP
  120 PRINT AT v,h;"O"
  130 PAUSE 2
  140 GO TO 70
```



Het gebruik van attributen

Het keyword ATTR neemt de "attributen" in een bepaalde positie op het scherm waar. De attributen zijn de inkt- en papierkleuren en of een positie flitsend of helder is. In het EXPODING SPIDERS-programma verzekert ATTR, dat de spin wordt vernietigd als hij geel wordt. Dit is dan zijn inkt-kleur (nummer 6). De papierkleur is blauw (nummer 1). Dit betekent, dat zijn attributen totaal 14 zijn (6 voor gele inkt plus 8 voor blauw papier).
In de Programmeur's Naslag-Handleiding kun je zien hoe je aan deze nummers komt.

HOE MAAK JE MUZIEK EN GELUIDSEFFECTEN?

De ZX Spectrum+ bezit een geluids-synthesizer, die met een grote verscheidenheid van muziek en geluidseffecten leven in je programma's brengt. Het is gemakkelijk te gebruiken, zelfs als je weinig of geen kennis van muziek hebt. De synthesizer produceert een geluids-signaal, dat naar de interne luidspreker van de spectrum+ gaat.

Het programmeren van geluid.

Voor de produktie van geluid met je Spectrum+ gebruik je maar één keyword-BEEP. Het wordt gevolgd door twee nummers of variabelen, die nummers vertegenwoordigen. Het eerste vertelt de computer hoe lang (in seconden) het geluid moet duren, het tweede vertelt het hoe hoog of laag het in toon moet zijn. De toonhoogte wordt gemeten in halve tonen. De toonwaarden zijn Ø voor de middelste C, 1 voor C# –1 voor B# (Cb) enz. Draai het volgende programma af.

```
10 FOR p=69 TO -60 STEP -1
20 BEEP .2,p
30 PRINT AT 0,0;"  ";AT 0,0;p
40 NEXT p
```

De Spectrum+ loopt door zijn hele bereik van noten van de hoogste toon. (69) tot de laagste (–6Ø). Je zult zien, dat de hoogste noten bijna onhoorbaar zijn en de laagste bijna als een klik kinken. Het schema onderaan deze bladzijde laat de toon waarden zien van een aantal noten, zodat je daarmee een blad muziek in een programma kunt veranderen.

Geluidseffecten

Je kunt allerlei soorten geluids-effecten uit de Spectrum+ krijgen, gewoonlijk door BEEP in een lus te zetten, die de toonwaarde snel verandert. Probeer de volgende programma's. Merk op, dat de lengte-waarden zeer kort zijn, zo kort als een honderdste van een seconde. Druk BREAK in om de geluste programma's te eindigen.

BUBBLING

```
10 LET p=INT (RND*40)-30
20 BEEP 0.05,p: BEEP 0.05,p+7:
BEEP 0.05,p+4
30 GO TO 10
```

Dit programma speelt een groep van drie noten steeds op willekeurige toonhoogten. Het toonhoogte-bereik is wijd, maar je kunt de waarden in lijn 1Ø veranderen om het bereik te wijzigen.

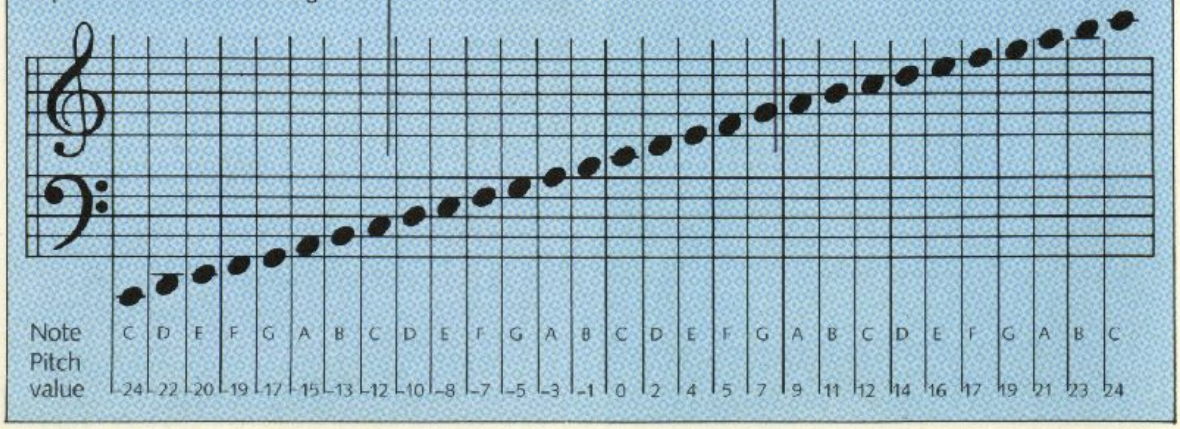
MACHINE

```
10 FOR x=12 TO 36
20 BEEP .01,x
30 BEEP .01,24-x
40 NEXT x
50 GO TO 10
```

Dit programma produceert twee geluiden, de ene gaat om hoog in toon terwijl de andere naar beneden gaat. Dit gebeurt, omdat de twee BEEP statements twee noten steeds slechts een honderste seconde op verschillende toonhoogten laten klinken.

Toonwaarden voor het maken van muziek

Hier zijn de Spectrum's toonwaarden van laag tot de top van bas en de drievoudige notenbalken. Voeg 1 extra tot de toonhoogte voor een hoge toon; haal er 1 af voor een lage toon.



LIFTOFF

```
10 FOR P=1 TO 48 STEP 0.2
20 BEEP .01,P: BEEP .01,P-6
30 NEXT P
```

Dit programma lijkt op het MACHINE-programma, maar nu gaan de noten allebei naar boven, met een hoogteverschil van zes halve tonen. Bovendien veranderen de toonwaarden elke keer met Ø,2 – een vijfde van een halve toon. Dit laat het geluid langzaam in toon omhoog gaan. Probeer andere kleine toonveranderingen door de STEP-waarde te veranderen.

KEYBOARD CONVERTER

```
10 LET P=CODE INKEY$
15 IF P=0 THEN GO TO 10
20 BEEP .04,(P-30)/2
30 GO TO 10
```

Dit programma wacht tot je een willekeurige toets indrukt. Wanneer je dit doet, geeft elke toets een ander geluid. Merk op, dat als je CAPS SHIFT ingedrukt houdt terwijl je op een andere toets drukt, het geluid lager wordt. Dit programma werkt, omdat CODE INKEY$ iedere keer als je een toets indrukt een andere waarde aan p geeft. De tweede lijn zorgt ervoor dat de computer geen geluid maakt als er geen toets is ingedrukt. Je kunt de waarden die CODE geeft zien in de teken tabel op blz. 51.

Geluid en beeld

De gehuidseffecten, die je Spectrum kan produceren gaan het beste samen met actie op het scherm. Ga, om te demonstreren hoe je geluid aan programma's kan toevoegen, terug naar het SCUTTLING SPIDER-programma op blz. 34.

Onthoud, dat je een PAUSE-statement invoegt in lijn 135 om de actie te vertragen. In plaats hiervan, kun je een pauze programmeren, die geluid produceert. Verander lijn 135 in:

```
135 GOSUB 5ØØ
```

Draai het programma af.

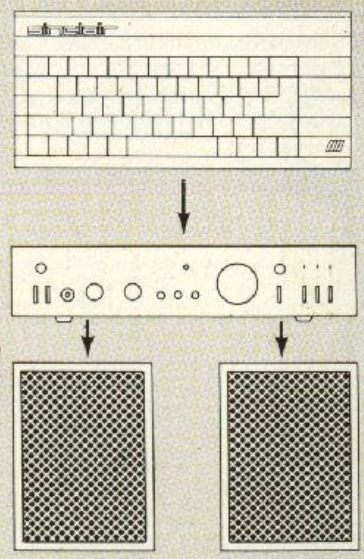
```
200 STOP
500 FOR p=40-l TO 38-l STEP -1
510 BEEP Ø.Ø2,p
520 NEXT p
530 RETURN
```

De spin maakt nu een vibrerend geluid terwijl hij valt. De subroutine speelt drie noten zeer snel. De noten worden lager naarmate de spin lager op het scherm wordt afgebeeld. Probeer nog meer noten toe te voegen door lijn 5ØØ te veranderen. Je kunt de noten ook versnellen of vertragen door Ø.Ø2 in lijn 51Ø te veranderen.

Hoe versterkje je je Spectrum+

Je kunt het geluid van je Spectrum+ harder maken door het EAR-contact of MIC-contact te verbinden met een versterker, luidspreker of kop telefoon.

De eenvoudigste manier is om het Spectrum-cassettesnoer te gebruiken. Verbind hiermee het EAR- of MIC-contact met het MIC-contact van een cassetterecorder. Neem de cassette eruit, indien nodig, zet de cassetterecorder aan en druk op PLAY, REWIND (REVERSE) of FAST FORWARD (CUE). Stel het volume van de cassette-recorder bij. Je moet nu het computer-geluid uit de luidspreker van de cassetterecorder horen. Je kunt ook koptelefoons gebruiken. Bovendien kun je je Spectrum aansluiten op een hi-fi-installatie als je een echt vol geluid wilt. Je hebt een speciaal snoer nodig met een 3.5mm jack-plug, die in de Spectrum past en een stekker voor het contact van de hi-fi-versterker. De Spectrum produceert een lijnsignaal, dat lijkt op dat van cassette- decks en recorders, dus het REPLAY of LINE IN- contact van de versterker moet werken.



HOE BEWAAR JE JE EIGEN PROGRAMMA'S?

Je zult al gauw je eigen programma's willen bewaren. Om dit te doen moet je een cassetterecorder aansluiten op je Spectrum en "save" het programma dat in de computer zit. Telkens, wanneer je het programma nodig hebt, laad je het van het cassettebandje terug in de computer met gebruik van de laad-procedure beschreven op blz. 14-15. Op deze twee bladzijden kun je zien hoe je een programma bewaart ("save") en of je het correct gedaan hebt.

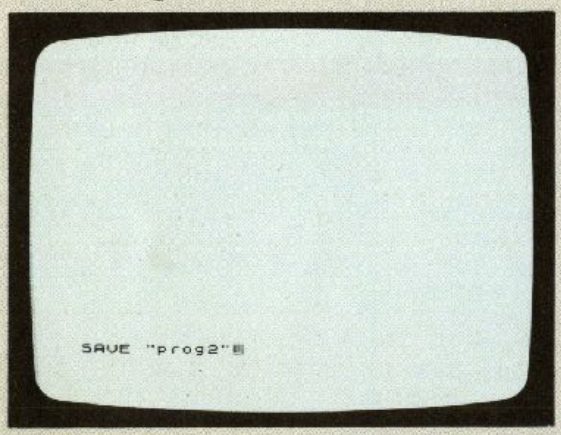
Het opnemen van je eigen programma's

1 Sluit eerst je Spectrum aan op een geschikte cassetterecorder, m.g.v. het cassettesnoer, zoals beschreven is op blz. 14. Zorg ervoor, dat alleen het Spectrum MIC-contact is aangesloten op de cassetterecorder.

2 Als de cassetterecorder een record-niveau of een volume-regelaar heeft, zet die dan op tweederde maximum. Het geeft niet, als je het vergeet, aangezien het recording-niveau automatisch afgesteld wordt.

3 Toets SAVE in, gevolgd door de naam van het programma tussen aanhalingstekens, bijvoorbeeld:

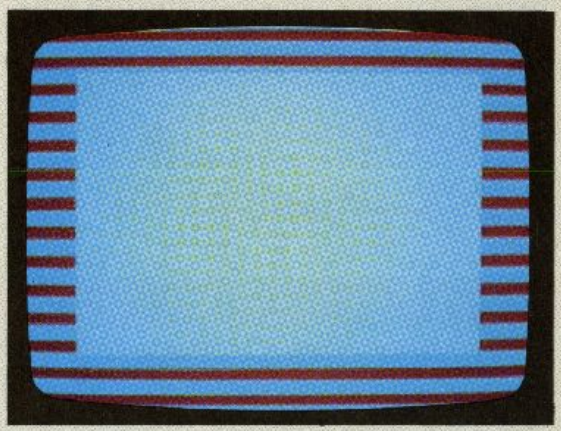
SAVE "prog2"



Elke willekeurige combinatie van max. tien letters en nummers kan gebruikt worden. Druk nu ENTER in. De SAVE-lijn verdwijnt en je zult zien, dat de cassetterecorder de instructie van de Spectrum opvolgt.

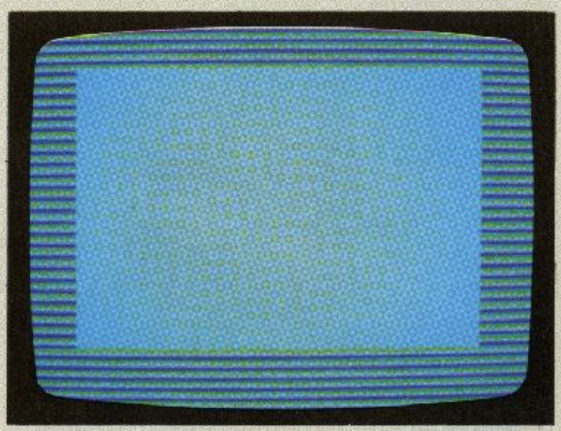
4 Zet de cassetterecorder op RECORD en PLAY tegelijkertijd. Druk dan op een willekeurige toets op de Spectrum.

5 Wacht nu terwijl de Spectrum het programma bewaart. Eerst moet je blauwe en vervolgens rode strepen zich langzaam over het scherm zien bewegen.



Dan krijg je een korte verschijning van blauwe en gele strepen. Dit gebeurt terwijl de Spectrum de naam van het programma naar de tape stuurt.

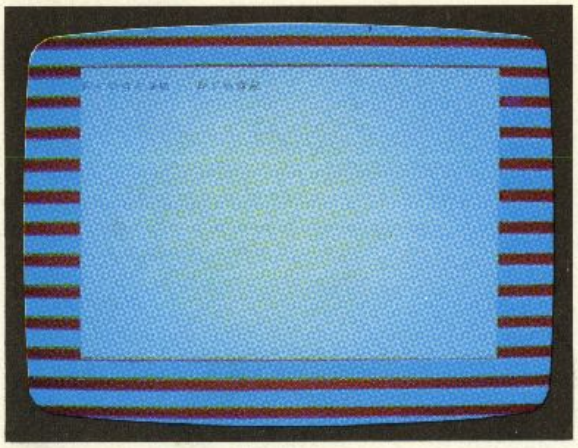
6 Vervolgens is er even niets, en dan weer blauwe en rode strepen. Dit wordt opnieuw gevolgd door blauwe en gele strepen terwijl de Spectrum nu het programma naar de tape zendt. Een lang programma kan enkele seconden duren.



7 Wanneer het programma naar de tape is gezonden verschijnt het rapport Ø OK,Ø:1 op het scherm. Stop de tape. Het programma is nu bewaard. Als je wilt, kun je het nu controleren.

Controleer het programma

Hoewel de computer het programma naar de tape heeft gezonden, kun je er niet zeker van zijn, dat het programma met succes op het bandje is opgenomen. Gelukkig kan de Spectrum dit voor je controleren. Deze procedure wordt verificatie genoemd. Spoel eerst het bandje terug naar het begin van het programma; sluit dan het Spectrum EAR-contact aan op het EAR-contact van de cassetterecorder. (Je kunt de MIC-contacten aangesloten laten). Toets nu VERIFY in, gevolgd door de programma naam tussen aanhalingstekens. Toets dan ENTER in de tape. Dezelfde volgorde van blauwe en rode, en blauwe en gele strepen verschijnt. De programma naam verschijnt en blijft op het scherm totdat de verificatie voltooid is.



Wanneer de tweede blauwe en gele sectie eindigt moet het rapport

ØOK,Ø:1

verschijnen. Dit betekent, dat je Spectrum het programma op tape heeft vergeleken met het programma in het geheugen, en ontdekt heeft, dat ze precies hetzelfde zijn.

Je kunt er nu zeker van zijn, dat je programma veilig op het bandje staat.

Tips voor het bewaren van software

1. Schrijf de naam van een programma op het cassette-label of op een kaart. Gebruik dezelfde grote of kleine letters als op het scherm. Als de cassetterecorder een teller heeft, gebruik die dan om de locatie van een programma te bepalen en schrijf het nummer bij de naam.
2. Plaats voordat je een programma bewaart, de programma naam in het programma m.g.v. een REM-statement, bijvoorbeeld:

```
5 REM SPIN program Version 3
```

De computer negeert bij het afdraaien alle REM-statements. Je kunt REM gebruiken om opmerkingen enz. in het programma te zetten.

Als je dit rapport niet krijgt, is er iets verkeerd gegaan. controleer eerst de Software-lading storings- zoeker op blz. 16, aangezien het kan zijn, dat het programma wel veilig op tape staat, maar dat het niet voor verificatie teruglaad in de computer. Als hier iets fout is, corrigeer dan de fout, spoel de tape terug en controleer het programma opnieuw. Als de computer het programma nog niet verifieert, consulteer dan de Software-saving storingszoeker op de volgende bladzijde, druk niet op NEW of RESET en zet de computer niet uit, aangezien je dan het programma verliest zonder dat je een betrouwbare kopie op tape hebt.

Automatische programma-start

Je kunt SAVE laten volgen door de programmanaam en dan LINE 1, bijvoorbeeld:

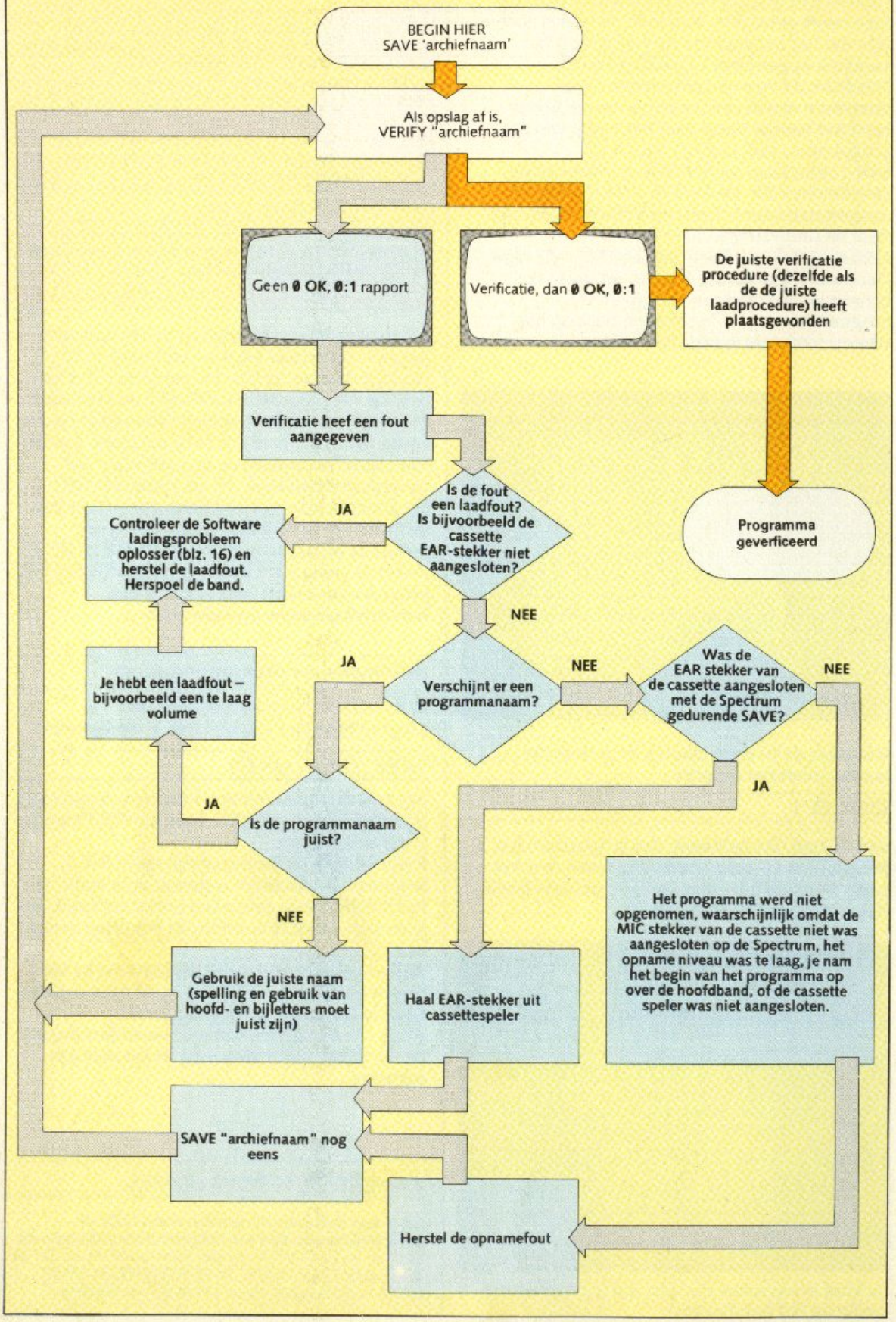
SAVE "SPIN" LINE 1

het bewaren gaat net zoals tevoren, maar voeg bij het verifieren *niét* LINE 1 toe ná VERIFY en de programmanaam.
Programma's die bewaard zijn met LINE 1, beginnen automatisch wanneer je ze laadt. Je hoeft RUN niet te gebruiken (maar onthoud, dat je de tape moet stopzetten als het programma begint)
Wat er gebeurt, is dat het programma bij lijn 1 begint en als er geen lijn 1 is, gaat de computer naar de eerste lijn in het programma.
Als je 1 in een ander nummer verandert begint de computer automatisch bij de lijn met dat nummer.

Saving CODE, SCREEN$ en DATA

SAVE kan ook gebruikt worden met CODE of SCREEN$ om een deel van het Spectrum-geheugen op te slaan en met DATA om een array op te slaan. Zie de delen over SAVE CODE, SAVE SCREEN$ en SAVE DATA in de Programmeur's Naslag-Handleiding.

Software opslag probleem-oplosser



LEER JE ZX SPECTRUM+ KENNEN

Dit hoofdstuk neemt je mee naar binnen in de ZX Spectrum+. Het legt uit, hoe de verschillende componenten onder het toetsen bord werken en hoe ze verbonden zijn om de computer te laten functioneren.. Het laat ook zien hoe je "peripherals" kunt gebruiken. Dit zijn toegevoegde apparaten, die van je Spectrum een volledig computer-systeem maken. Tenslotte vind je hier meer over de technische kant van je computer – inclusief de manier, waarop het geheugen is georganizeerd, samen met de Spectrum's technische specificatie.

HOE ZIET HET ER VAN BINNEN UIT

Lees verder om dat uit te vinden, maar probeer niet je ZX Spectrum+ open te maken om te zien hoe het werkt. Je verspeelt je garantie als je dat doet en je kan ernstige schade aanbrengen. Binnen in de kast zitten twee lintconnectors, die het toetsenbord met de rest van de componenten verbinden. Deze zijn allemaal gemonteerd op een circuit-bord met gedrukte bedrading. Het bord heeft standaard elektrische componenten, zoals weerstanden en condensators, maar de meest opvallende delen zijn de zwarte rechthoekige microchips, die ofwel enkelvoudig zijn opgesteld ofwel in blokken.

Binnen in de chip

Het functionerende deel van een microchip is feitelijk veel kleiner, dan het plastic omhulsel, waar de chip in zit. Het omhulsel is in de eerste plaats ontworpen om alle connecties, die de chip nodig heeft, te steunen, zodat het aangesloten kan worden op de contacten op het circuitbord. De chip zelf is een dun silicium-plaatje, dat vele duizenden elektrische verbindingen bevat. Elk verbindingspunt werkt als een schakeling, die de signalen kan stoppen, doorgeven of opslaan. Hoewel dit een eenvoudige procedure is, zijn er zoveel verbindingspunten, die allemaal samenwerken, dat ze signalen kunnen produceren, die informatie opslaan of verwerken.

Hoe zijn chips verbonden?

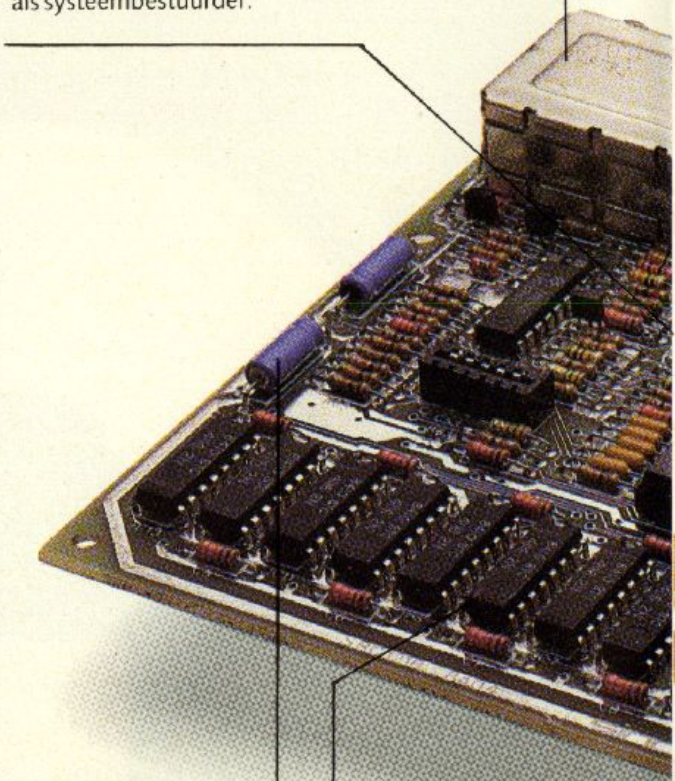
In het kort is de Spectrum dus een elektrisch circuit van enorme complexiteit. Codesignalen, bestaande uit stroomstootjes, flitsen voortdurend langs de paden binnen en tussen de chips en componenten, om zo de computer te laten werken. Hoe wordt alles zo geregeld, dat het juiste signaal op het juiste moment in de juiste plaats aankomt? Binnen in een van de chips is de computer-klok. Het tikt d.m.v. stroomtootjes: 3,5 mil joen per seconde. Deze stootjes bewegen zich regelmatig door de circuits om de codesignalen te produceren, die de werking van elk onderdeel besturen.

Binnen in je Spectrum. In deze afbeelding van het Spectrum circuitbord zijn de twee lintconnectors verwijderd.

Wanneer de Spectrum in gebruik is brengt het indrukken van een toets twee draden onder het toetsenbord met elkaar in contact. Dit stuurt een codesignaal naar de CPU.

Uncommitted Logic Array (ULA) Deze chip brengt de display van de informatie, die in RAM gehouden wordt voort. Het werkt ook als systeembestuurder.

TV Output. Dit produceert het signaal, dat naar het televisietoetsel gaat.



PAL (Phase Alternation Line) encoder. Dit verandert signalen, die geproduceerd worden door de computercircuits, in een kleurentelevisie-signaal.

Random Access Memory (RAM). Deze chips bevatten het programma, dat in de computer gevoerd wordt en elke speciale informatie, die het programma nodig heeft, zoals waarden van variabelen. De inhoud van de 48K van RAM kan gewijzigd worden via het toetsenbord, of kan helemaal uitgewist worden door op reset te drukken of de computer uit te zetten.

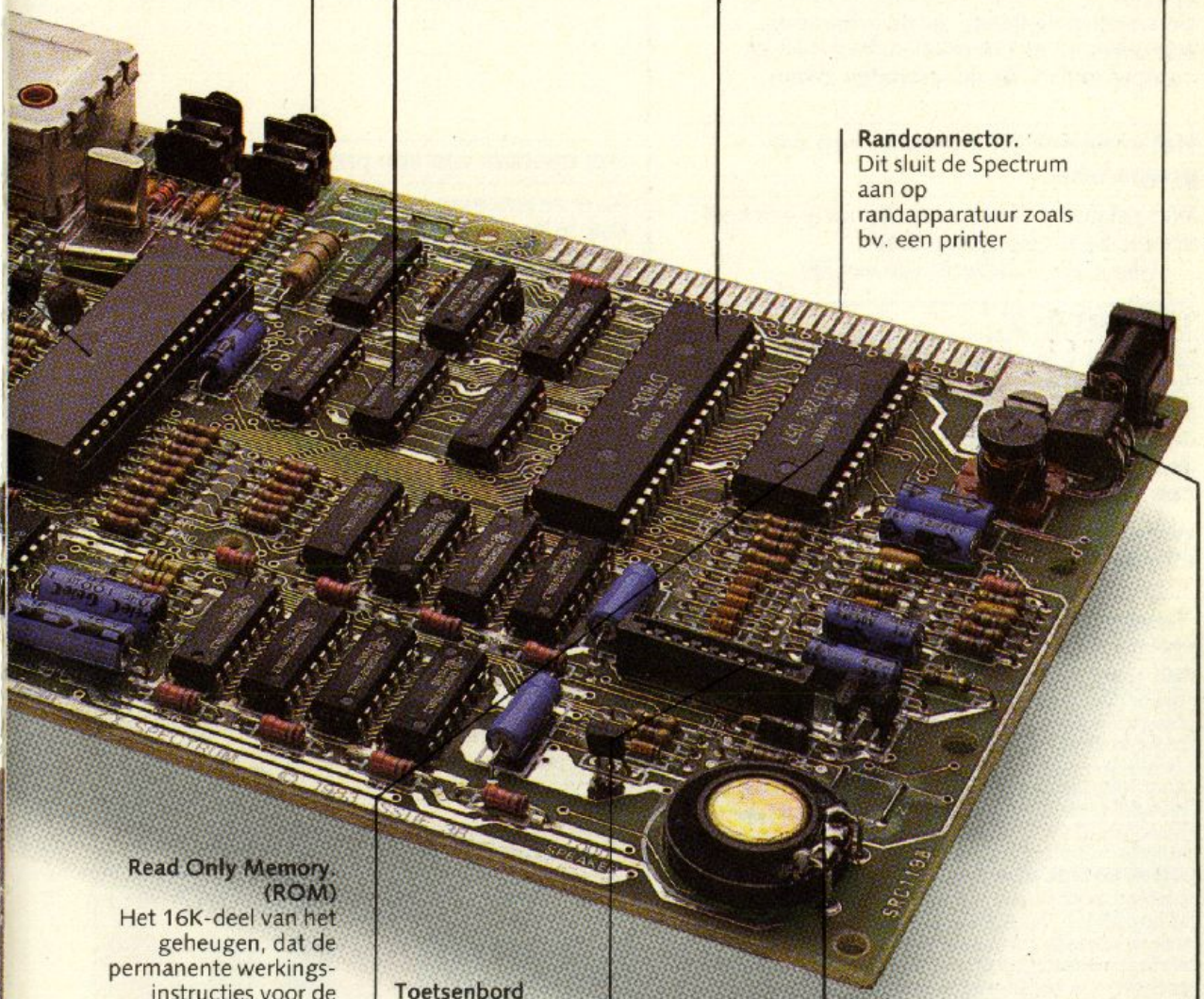
Cassette-contacten
Deze worden gebruikt om informatie en programma's van het geheugen naar de tape te sturen, en om ze van de tape terug te voeren in het geheugen.

Logic chips.
Deze chips werken als een interface bij de uitwisseling van informatie tussen de CPU en de RAM.

Central Processing Unit (CPU)>
Het brein van de computer. De CPU is een Z80 microprocessor. Het voert alle computer calculaties uit en bestuurt de algehele werking van de Spectrum.

9 Volt DC-contact.
Dit is de verbinding naar de stroomvoorziening.

Randconnector.
Dit sluit de Spectrum aan op randapparatuur zoals bv. een printer



Read Only Memory. (ROM)
Het 16K-deel van het geheugen, dat de permanente werkings-instructies voor de CPU bevat. Deze instructies zetten o.a. BASIC programma's om in een vorm, die de CPU kan begrijpen. De inhoud van dit geheugen kan niet vi a het toetsenbord gewijzigd worden.

Toetsenbord connector-punt
Eén van de lintconnectors naar het toetsenbord is hieraan verbonden.

Luidspreker.
Dit produceert geluid, indien vereist.

Voltage regulator.
Deze component zorgt ervoor, dat veranderingen in voltage de computer niet aantasten.

HOE WERKT JE ZX SPECTRUM+?

De werking van de ZX Spectrum+ bestaat zoals bij andere microcomputersystemen uit vier hoofdbestanddelen. Deze zijn de "input-units", zoals het toetsenbord, die informatie en programma's in de computer voeren; de tijdelijke en permanente geheugens, die informatie, programma's en werkingsinstructies opslaan; de Central Processing Unit (Centrale Verwerkingseenheid), die de programma-instructies op de informatie uitvoert en de "output units", die de resultaten geven.

Het intoetsen en afdraaien van een programma

Wat gebeurt er in de computer als je een heel eenvoudig programma afdraait?

Hier is een voorbeeld van een lijn:

1Ø PRINT 6+2

Eerst toets je in. Onder het toetsenbord bevindt zich een raster met een wirwar van draden. Steeds als je een toets indrukt maken twee draden contact en sturen een codesignaal naar de CPU. De CPU op zijn beurt stuurt de code naar RAM, waar het wordt opgeslagen.

Wanneer je het programma afdraait neemt de CPU de opgeslagen codes één voor één uit RAM, in volgorde van het programma. Eerst ontvangt het de code voor PRINT, die het opdraagt een bepaalde werkings-code uit ROM te gebruiken. Deze werkings-code gaat naar de CPU en de CPU bereidt zich voor om de bewerkingen uit te voeren, die een bepaalde waarde op het scherm afbeelden. De CPU krijgt vervolgens de waarde 6 uit RAM. Ook dit is in de vorm van een code en de CPU slaat het op in een klein intern geheugen, dat register heet. Daarna komt de code voor optellen en de CPU krijgt weer de noodzakelijke werkingscode uit ROM. Tenslotte neemt de CPU de code voor 2 uit RAM. Het telt deze code op bij de waarde in het register om het resultaat (8) te krijgen. De CPU zet dan het resultaat om in een andere serie codes, en het getal 8 verschijnt op het scherm.

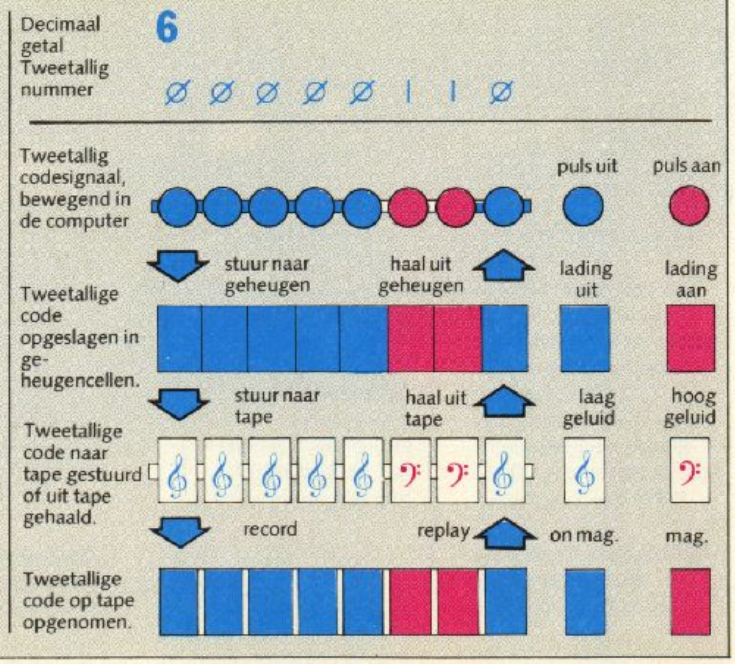
Het opslaan van een programma

Als je de Spectrum vraagt het programma op tape te bewaren, neemt de CPU opnieuw de codes uit RAM. Maar in plaats van ze te bewerken stuurt de CPU de codes naar een omzettingseenheid, die ze verandert in geluidssignalen. Deze signalen worden dan naar de cassetterecorder gestuurd en worden op tape opgenomen. Wanneer je later het programma laad worden de geluidssignalen van de cassetterecorder weer omgezet in computercodes door de omzetter.

De CPU stuurt ze terug naar RAM, waar ze opgeslagen worden totdat ze nodig zijn.

Tweetallige codes

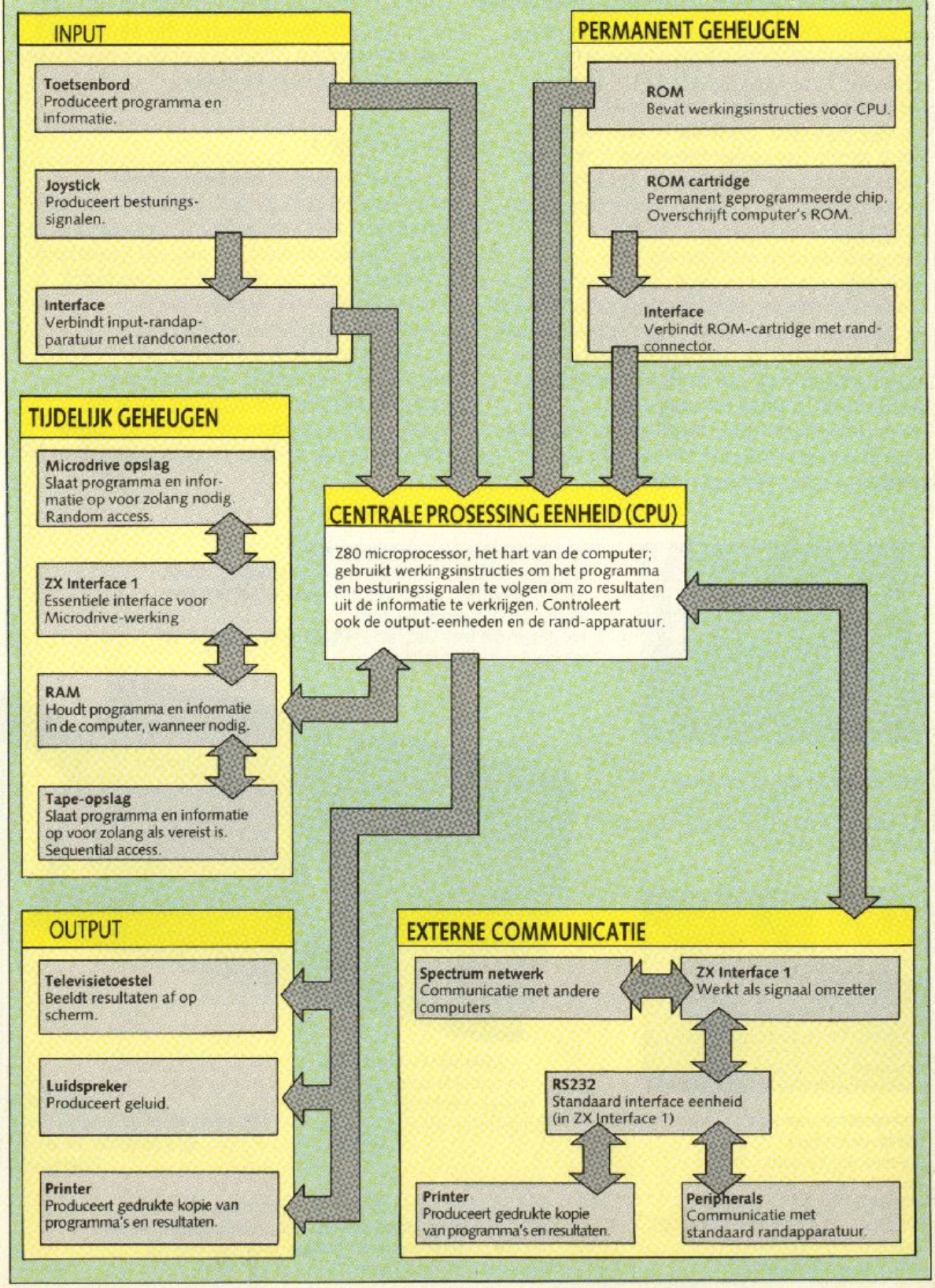
Alle codes, die je Spectrum gebruikt zijn in tweetallige vorm. Ze worden tweetallig genoemd, omdat ze allemaal bestaan uit slechts twee soorten signaal. Ze kunnen vertegenwoordigd worden door tweetallige nummers d.w.z. nummers, die slechts twee cijfers bevatten: Ø en 1. Het tweetallige nummer voor 6 is bijvoorbeeld ØØØØØ11Ø. Binnen in je Spectrum bestaan de codes uit series snelle stroomstoten. Als een stroomstoot bij een willekeurig punt aankomt vertegenwoordigt deze korte elektrische puls een 1 in het tweetallig stelsel. Als een puls niet binnen een bepaalde tijd aankomt, vertegenwoordigt dit een Ø. In computercode is 6 dus uit-uit-uit-uit-uit-aan-aan-uit. De Spectrum behandeld acht coden tegelijk. In dit diagram kun je zien hoe verschillende soorten tweetallige codes door de computer worden gebruikt om informatie van de ene plaats naar de andere te bewegen.



De Spectrum-paden van Input naar Output

Dit diagram laat zien hoe gecodeerde informatie van input-units zoals het toetsenbord door het verwerkingssysteem van de Spectrum loopt, en dan naar output-units zoals het televisiescherm.

Eénhoofdige pijlen geven paden aan, die slechts in een richting lopen. Dubbelhoofdige pijlen geven paden aan, die in in beide richtingen kunnen werken.



HOE SLUIT JE RANDAPPARATUUR AAN?

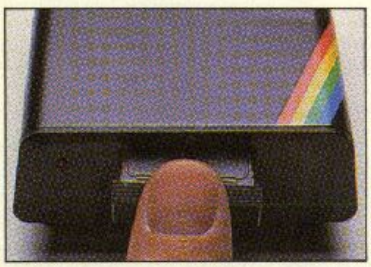
Je kunt je ZX Spectrum+ opwaarderen tot een compleet en krachtig computer-systeem door het gebruik van Sinclair en andere met de Spectrum verenigbare randapparatuur. Centraal in dit systeem staat de ZX Spectrum Interface 1, die je in staat stelt Microdrives aan te sluiten voor een snelle en eenvoudige verwerking van programma's en data en die ook aangesloten kan worden op vele verschillende randapparatuuren, inclusief andere Spectrums. Met deze interface kun je je Spectrum aansluiten op standaardprinters en op een modem. Andere interfaces zijn beschikbaar, die plug-in ROM cartridges aansluiten op de computer voor onmiddellijke programma-lading. Hiermee kun je ook joysticks verbinden om zo spelletjes spannender te maken.

Met de Spectrum verenigbare printers

Sommige printers worden direct aangesloten op de Spectrum's randconnector. Als je al een Sinclair ZX printer hebt, heb je geen interface nodig voor de aansluiting met de computer. Als je echter printers gebruikt, die een RS232 output vereisen, moet je het D-contact van de ZX Interface 1 gebruiken.

ZX Interface 1

De ZX Interface 1-eenheid wordt aangesloten aan de achterkant onder de Spectrum. Het kan je Spectrum aansluiten op acht microdrives, 63 andere Spectrum computers en, via de RS232 standaard interface eenheid, op een grote verscheidenheid van standaard randapparatuur. Microdrives en microdrive cartridges vervangen de cassetterecorder en-bandjes voor de opslag van programma's en data. Door microdrive-cartridges te gebruiken, kun je



Het laden van een Microdrive
Microdrive-cartridges worden ingebracht in de gleuf aan de voorkant van de drive.

Het inzetten van een ROM-cartridge
De cartridge wordt eenvoudigweg op het interface-contact aangesloten. Wanneer de computer aanstaat wordt het programma automatisch geladen.

Het aansluiten van randapparatuur.
De ZX Interface 1 is permanent aangesloten op de rand-connector, zodat het zich aan de achterkant onder de computer bevindt.
De illustratie hier laat het systeem zien, voordat de computer is aangesloten.

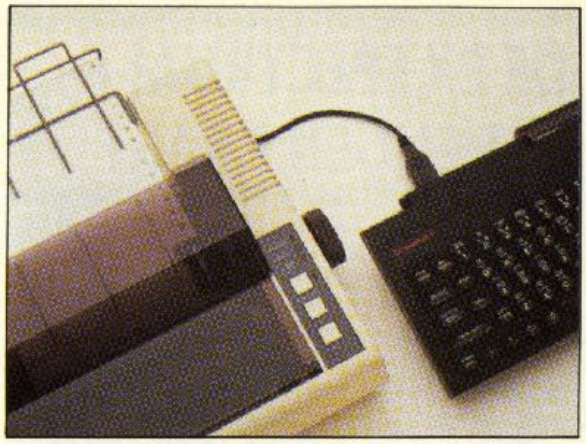
Microdrive-eenheden
Maximaal acht van deze opslag-eenheden kunnen aangesloten worden op één Spectrum.

Lint-kabel
Dit verbindt de Microdrive met de computer via de ZX Interface 1.

Voetnoot
De ZX Spectrum+ heeft twee ingebouwde voetjes die gebruikt kunnen worden om het toetsenbord op te tillen. Deze voetjes zyn niet nodig als een ZX Interface 1 is aangesloten.

binnen seconden programma's bewaren, verifieren en laden. Elke cartridge kan wel 85K aan data opslaan en met gebruik van het maximum van acht microdrives, heeft je Spectrum een opslagcapaciteit van maximaal 680K. Elk programma wordt automatisch opgespoord met een typische access-tijd van 3,5 seconden.

Met gebruik van het netwerk-snoer, dat bij de interface-eenheid geleverd wordt, kun je de computer verbinden met een andere computer: ofwel een ZX Spectrum of een andere ZX Spectrum+. Dit netwerk kan uitgebreid worden tot een maximum van 63 andere Spectrums. De ZX Interface 1-eenheid bevat ook een RS232 interface met D-contact. Een standaard interface-kabel is eveneens beschikbaar.



Standaard printers worden aangesloten via de ZX Interface 1.

ROM cartridges en joysticks

Interfaces zoals de ZX Interface 2 zijn voor de aansluiting van ROM cartridges en joysticks. ROM cartridges laden onmiddellijk, zodra de computer aanstaat. Met een cassetterecorder zou het laden veel langer duren.

ROM cartridge/joystick interface

Randconnector
Randapparatuur wordt via deze connector aangesloten op de computer.

Waarschuwing
Randapparatuur moet altÿd worden aangesloten voordat je de stroom inschakeld.

DE ZX SPECTRUM+ GEHEUGEN KAART

Als je kijkt naar de foto van de binnenkant van de Spectrum op blz. 42-43, zul je zien, dat er een ROM chip is en 16 kleinere RAM chips. Deze chips vormen het geheugen van de Spectrum. Het geheugen bestaat uit 65536 opslageenheden, die elk een byte (een nummer van Ø tot 255) bevatten. Elke eenheid is herkenbaar aan een nummer, dat "address" (adres) genoemd wordt.

ROM betekent Read Only Memory. Dit deel van het geheugen bevat werkings-instructies voor de Central Processing Unit. Het is een 16K ROM d.w.z. het bevat 16x1Ø24 (16384) bytes of adressen. De bytes kunnen alléén vanuit dit geheugen gelezen worden, zodat ze niet veranderd kunnen worden. (Als dit wel kon zou de computer niet meer werken). Je kunt de byte op elk adres verkrijgen door PEEK te gebruiken. RAM betekent Random Access Memory. Het bevat de programma's en informatie die in de computer gevoerd worden. De Spectrum heeft een 48K RAM d.w.z. het bevat 48x1Ø24 (49152) bytes of adressen.. 'Random access' betekent, dat een byte op elk willekeurig adres veranderd kan worden. Dit kan gedaan worden m.g.v. POKE.

De geheugenadressen lopen van Ø tot 65535. Het eerste kwart hiervan zijn ROM adressen, de rest RAM.

Systeem-variabelen

De kolom aan de rechterkant laat zien hoe het Spectrum-geheugen is georganiseerd. Je kunt zien waar de verschillende delen, die de computer besturen zich bevinden. Sommige hiervan kunnen van plaats veranderen, en hun grenzen worden aangegeven door systeem-variabelen.

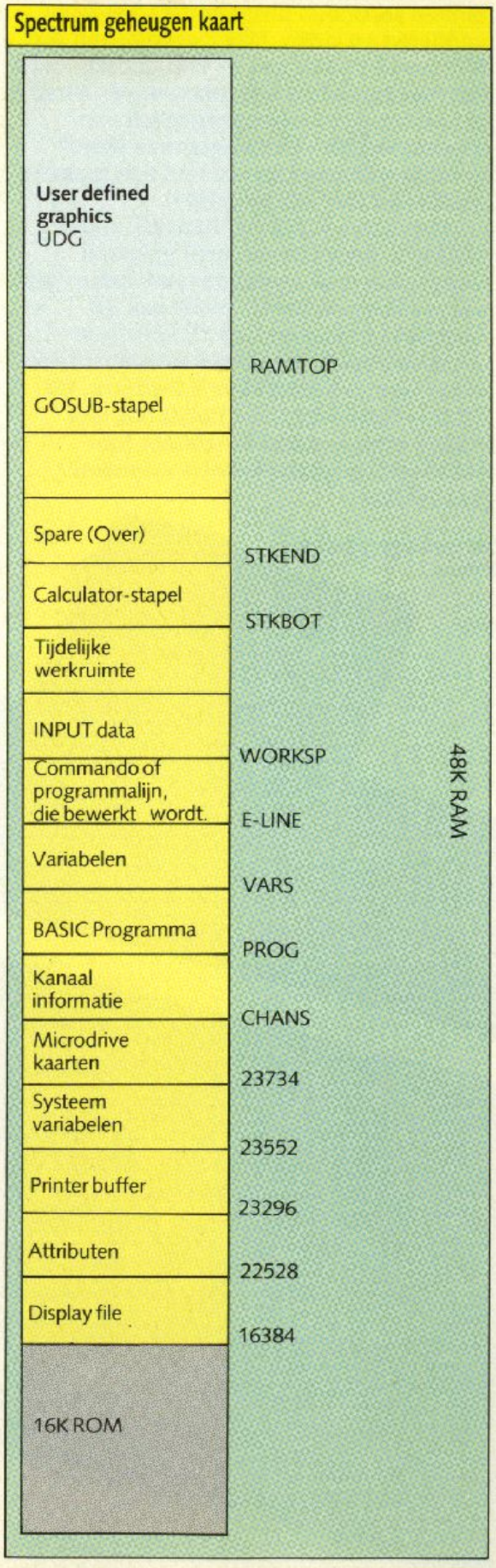
De Spectrum systeem-variabelen zijn geen variabelen zoals die in BASIC gebruikt worden. Het zijn namen voor bepaalde bruikbare waarden, die zich op bepaalde adressen of locaties in het geheugen bevinden. Het doel van de namen is om de betekenis van de bepaalde waarde, opgeslagen op de locatie, te helpen onthouden. Bijvoorbeeld: de systeem-variabele RAMTOP is het bovenste adres in RAM, dat gebruikt kan worden om een BASIC programma en de waarden van zijn variabelen op te slaan. Als je met PEEK de locatie bekijkt, die voor de systeem-variabele wordt gegeven, vind je de waarde, die je zoekt. De manier om dit te doen is om in te toetsen:

```
PRINT PEEK n+256*PEEK(n+1)
```

waarin n de locatie van de systeem-variabele is. Toets, om RAMTOP te vinden in:

```
PRINT PEEK 2373Ø+256*PEEK 23731
```

Je moet nu de waarde 65367 krijgen, het normale adres van RAMTOP. Als je echter RAMTOP verandert met CLEAR, krijg je het nieuwe adres.



LEER SINCLAIR BASIC KENNEN

Dit hoofdstuk beschrijft SINCLAIR BASIC in detail. Je zult hier een overzicht vinden van de manier waarop ieder keyword wordt gebruikt en verdere details m.b.t. de werking van Sinclair BASIC. De gegeven informatie rijkt van het eenvoudigste tot het meest vooruitstrevende BASIC programmering die nodig is. Dit is geen hoofdstuk om van het begin tot het eind door te lezen. Het is eigenlijk een woordenboek voor de programmeur dat je de gelegenheid geeft de mogelijkheden van de Spectrum ten volste te benutten.

REFERENTIE GIDS VAN SINCLAIR BASIC KEYWORDS VOOR PROGRAMMEURS

Deze gids bevat de volledige beschrijving van alle BASIC keywords die beschikbaar zijn op de ZX Spectrum+. Iedere ingang kenmerkt:

- plaats van het keyword
- klasse van het keyword
- doel van het keyword
- gebruik van het keyword
- de programmeer format
- de character bepaalde code

De details met betrekking tot de plaats, het doel, het gebruik en de character bepaalde code verklaren zichzelf. Klasse en format zijn moeilijker, en om het gebruik van de gids zo effectief mogelijk te maken moet je eerst de informatie op deze bladzijde aandachtig lezen.

Keyword klassen

Je kunt keywords over vier klassen verdelen

Opdracht
Een keyword dat er voor zorgt dat er een bepaalde actie plaats vindt kan worden gebruikt om een directe opdracht te vormen. Het wordt uitgevoerd zodra het is ingevoerd.
Bijvoorbeeld: RUN, LOAD.

Statement
Een keyword dat er voor zorgt dat er een bepaalde actie plaats vindt in dat kan worden gebruikt op een programma line. Het wordt alleen uitgevoerd als het programma loopt.
Bijvoorbeeld: DRAW, INPUT

Functie
Een keyword dat een of andere value produceert. Het is een onderdeel van een opdracht of een statement.
Bijvoorbeeld: RND, INT.

Logische operator
Een keyword dat wordt gebruikt om logica in een statement of een opdracht uit te drukken. Het kan de waarheid van bepaalde voorwaarden bepalen of veranderen. De Spectrum heeft drie logische operators: AND, OR en NOT.

Nummers en Variabelen

Nummers
Opgeslagen tot een accuraat van 9 of Ø digits. Nummer handling range is tot ongeveer $1Øu23^{8}$ tot $4 * 10^{39}$.

Variabelen geakkepteerd.
Nummer, elke lengte, beginnend met een letter. Spaties worden niet meegerekend en alle letters worden kleine letters. Hoofdletters en kleine letters worden niet geidentificeerd.

String Elke letter gevolgd door $. Hoofdletters en kleine letters worden niet geidentificeerd.

Array Voor array variablen en subscripties, zie de entry voor DIM.

Keyword format

Het keyword format drukt de syntaxis van ieder keyword uit dat wil zeggen, de juiste combinatie van het keyword en andere factoren zoals values en variabelen. De volgende afkortingen worden gebruikt in de format notatie:

Afkorting	Verklaring	Voorbeeld
num-const	Een numerieke constante (een getal)	24.5
num-var	Een numerieke variabele (een variabele die een getal kan bevatten)	som
num-expr	Een numerieke uitdrukking (iedere geldige combinatie van numerieke constanten en numerieke variabele en keywords die een getal geeft).	som*24.5 **RND***7
int-num-const int-num-var int-num-expr	Een numerieke constante, variabele of uitdrukking waar van de value wordt afgerond naar het dichtst bij zijnde gehele getal.	
string-const	Een rij-constante of rij (iedere combinatie van characters tussen aanhalingstekens)	"ZX Spectrum+"
string-var	Een rij-variabele (een variabele die een rij kan bevatten)	a$
string-expr	En rij-uitdrukking (iedere geldige combinatie van rij constanten en variabelen en keywords die een rij geeft).	a$+"ZX Spectrum+" a$(6 **TO** 8)
letter	Iedere hoofd-letter of gewone-letter	Y x
letter$	Iedere hoofd-letter of gewone-letter gevolgd door $	B$ a$
cond	Een voorwaarde of subvoorwaarde binnen een voorwaarde	x=10 **AND** t<1Ø
statement	Iedere BASIC statement die geldig is als zij wordt gebruikt met een andere statement. Een item naar keuze, dat herhaald kan worden.	**IF** t>1Ø **THEN STOP** **PRINT INK** 2;x
[ ]	An optional item that may be repeated	

Tekens in Sinclair BASIC

Teken	Plaats	Actie/Gebruik
$	4	Rij variabele
'	7	Begint nieuwe line
(	8	Haakje openen
)	9	Haakje sluiten
<=	Q	Kleiner of gelijk aan
<>	W	Niet gelijk aan
>=	E	groter of gelijk aan
<	R	kleiner dan
>	T	groter dan
↑	H	machts verheffen
–	J	aftrekken/neg.getal
+	K	optellen/pos.getal/ aaneenschakeling

Teken	Plaats	Actie/Gebruik
=	L	In gelijk aan
:	Z	Scheidt statements op een programma line
/	V	Delen
*	B	Vermenigvuldigen
.	Own key	Decimaal punt
;	Own key	Beeld de volgende kolom af. Scheidt statements binnen een programma statement.
"	Own key	Openen en sluiten van een rij
,	Own key	Beeld af in kolom 0 of 16. Scheidt values volgend op keywords

ZX Spectrum+ character set

	0	1	2	3	4	5	6	7	8	9
0							**PRINT** komma	**EDIT**	cursor links	cursor rechts
10	cursor beneden	cursor omhoog	**DELETE**	**ENTER**	nummer		**INK** controle	**PAPER** controle	**FLASH** controle	**BRIGHT** controle
20	**INVERSE** controle	**OVER** controle	**AT** controle	**TAB** controle						
30			ruimte	!	"	#	$	%	&	'
40	(	)	*	+	,	–	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	/	]	↑	_	£	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~	©	□	▙
130	▟	▄	▜	▐	▚	▛	▗	▞	▌	▙
140	▄	▟	▜	▐	GRAPHICS A	GRAPHICS B	GRAPHICS C	GRAPHICS D	GRAPHICS E	GRAPHICS F
150	GRAPHICS G	GRAPHICS H	GRAPHICS I	GRAPHICS J	GRAPHICS K	GRAPHICS L	GRAPHICS M	GRAPHICS N	GRAPHICS O	GRAPHICS P
160	GRAPHICS Q	GRAPHICS R	GRAPHICS S	GRAPHICS T	GRAPHICS U	**RND**	**INKEY$**	**PI**	**FN**	**POINT**
170	**SCREEN$**	**ATTR**	**AT**	**TAB**	**VAL$**	**CODE**	**VAL**	**LEN**	**SIN**	**COS**
180	**TAN**	**ASN**	**ACS**	**ATN**	**LN**	**EXP**	**INT**	**SQR**	**SGN**	**ABS**
190	**PEEK**	**IN**	**USR**	**STR$**	**CHR$**	**NOT**	**BIN**	**OR**	**AND**	<=
200	>=	<>	**LINE**	**THEN**	**TO**	**STEP**	**DEF FN**	**CAT**	**FORMAT**	**MOVE**
210	**ERASE**	**OPEN #**	**CLOSE #**	**MERGE**	**VERIFY**	**BEEP**	**CIRCLE**	**INK**	**PAPER**	**FLASH**
220	**BRIGHT**	**INVERSE**	**OVER**	**OUT**	**LPRINT**	**LLIST**	**STOP**	**READ**	**DATA**	**RESTORE**
230	**NEW**	**BORDER**	**CONTINUE**	**DIM**	**REM**	**FOR**	**GO TO**	**GO SUB**	**INPUT**	**LOAD**
240	**LIST**	**LET**	**PAUSE**	**NEXT**	**POKE**	**PRINT**	**PLOT**	**RUN**	**SAVE**	**RANDOMIZE**
250	**IF**	**CLS**	**DRAW**	**CLEAR**	**RETURN**	**COPY**				

ABS ABSolute value

Plaats op het toetsen bord
Extend modus
Symbool G

Functie

ABS geeft de absolute grootte van een numerieke value, dat is de value zonder een + of – teken.

Hoe gebruik je ABS?
ABS wordt gevolgd door een numerieke value. Een uitdrukking moet tussen haakjes staan, bijvoorbeeld:

5Ø LET x = ABS (y – z)

ABS retourneert de absolute value van de numerieke value bijvoorbeeld:

De opdracht:

PRINT ABS – 34,2

Levert op het beeldscherm 34,2 op.

Format
ABS num-const
ABS num-var
ABS (num-expr)

ACS ArcCoSine

Plaats op het toetsenbord
Extend modus
Symbool W

Functie

ACS berekent de value van een hoek uit zijn cosinus.

Hoe gebruik je ACS?
ACS wordt gevolgd door een numerieke value. Een uitdrukking moet tussen haakjes staan, bijvoorbeeld.

6Ø LET x = ACS (y +x)

De value volgend op ACS (y + z boven) is de cosinus van de gewenste hoek en kan varieren van –1 tot 1. ACS retourneerd de value van de hoek dan in radialen. Om radialen in graden om te zetten moet je dan de value die ACS retourneerde met 18Ø/ π vermenigvuldigen. (π = P1) Bijvoorbeeld:

De opdracht:

PRINT 18Ø PI *ASN Ø.5

afgebeeld wordt 6Ø, de hoek in graden, met een cosinus van Ø,5.

Format
ACS num-const
ACS num-var
ACS (num-expr.)

AND

Plaats op het toetsenbord.
Symbool Y

logische werking/functie

AND werkt als een, logical operator om de waarheid van een combinatie van voorwaarden te testen. Alleen als alle afzonderlijke voorwaarden waar zijn, is de combinatie van voorwaarden waar. AND werkt ook als een function om binaire operaties op twee numerieke values of op twee, string values, uit te voeren.

Hoe gebruik je AND?
Als je AND gebruikt als een logical operator dan verbindt het twee voorwaarden in een statement, waar de waarheid van de combinatie getest moet worden, bijvoorbeeld:

9Ø IF x = y + z AND ook < 1Ø dan afgedrukt correct.

Alleen als beide voorwaarden (x = y + z en ook < 1Ø) waar zijn zal de computer correct afbeelden. Als een van de twee voorwaarden (of allebei de voorwaarden) onjuist is, dan is de hele combinatie onjuist en zal bij dit voorbeeld, het programma doorgaan met de volgende regel (line 91).

AND als een, function
Als een function kan AND werken met twee numerieke values bijvoorbeeld:

5Ø LET x = y AND z

AND laat de eerste (y) zien als de tweede (z) niet gelijk is aan ø, en laat ø zien als de tweede, value ø is.

AND kan ook werken op een string value maar dan moet deze voor AND staan. Een numerieke value moet juist altijd ná AND volgen, bijvoorbeeld:

5Ø LET a $ = b $ AND z

AND laat de eerste value (b $) zien indien de tweede (z) niet gelijk is aan Ø. Als de tweede value (z) wel Ø is, dan laat AND een, null-string (" ") zien. Bedenk dat de ZX Spectrum + een value toekend aan een ware of juiste voorwaarde en een value Ø aan een valse of onjuiste voor waarde.

De ZX Spectrum+ herkent iedere niet-nul value als juist en iedere nul (Ø) als onjuist. De ZX Spectrum+ werkt combinaties van numerieke values niet uit in overeenstemming met standaard 'truth-tables' (waarheids tabellen).

Format (Form of Material)
cond AND oond
Num.-expr. AND num.-expr.
String-expr. AND num.-expr.

ASN ArcSiNe

Plaats op het toetsenbord
Extend modus
Symbool Q

Functie:

ASN berekent de value van een hoek uit zijn sinus.

Hoe gebruikje ASN?
ASN wordt gevolgd door een numerieke value. Een uitdrukking moet tussen haakjes worden geplaatst, bijvoorbeeld:

6Ø LET x = ASN (y * z)

De value volgend op ASN (y + z boven) is de sinus van de vereiste hoek. Deze value kan variëren van –1 tot +1. ASN geeft dan de value van de hoek in radialen weer. Om radialen in graden om te zetten moet je de value die ASN retourneerde met 18Ø/ π vermenigvuldigen, bijvoorbeeld:

De opdracht:

PRINT 18Ø PI * ACS Ø.5

afgebeeld wordt 3Ø, de hoek in graden met een sinus van Ø,5.

Format
ASN num-const
ASN num-var
ASN (num-expr.)

AT

Plaats op het toetsenbord
Symbool 1

Zie **Input; L Print; Print**

ATN ArcTaNgent

Plaats op het toetsenbord
Extend modus
Symbool E

Functie

ATN berekend de value van een hoek uit zijn tangens.

Hoe gebuik je ATN?
ATN wordt gevolgd door een numerieke value. Een uitdrukking moet tussen haakjes worden geplaatst, bijvoorbeeld:

6Ø LET x = ATN (y * z)

De value volgend op ATN (y * z hoven) is de tangens van de verlangde hoek. ATN retourneerd de value van de hoek in radialen. Om radialen in graden om te zetten moet je de value die ATN retourneerd vermenigvuldigen met 18Ø/ π.

Bijvoorbeeld:
De opdracht:

PRINT 18Ø/π • ATN 1

afgebeeld wordt dan 45, de hoek in graden die een tangens van 1 heeft.

Format
ATN num-const
ATN num-var
ATN (num-expr.)

ATTR ATTRibutes

Plaats op het toetsenbord
Extend modus
Symbool L

Functie

ATTR levert de bijkomstigheden van een specifieke character positie op het scherm. Dit zijn de kleuren van inkt en papier, de helderheid en de knipper-status van het character in die positie.

Hoe gebruik je ATTR?
ATTR wordt gevolgd door twee numerieke values gescheiden door een komma en tussen haakjes geplaatst, bijvoorbeeld:

15Ø IF ATTR (v,h) = 115 THEN GOSUB 2ØØØ

De eerste value volgend op ATTR (v boven) mag varieren van Ø tot 23 en is het nummer van een line op het beeldscherm. De tweede value (h boven) mag varieren van Ø tot 31 en is het kolom nummer van de positie. ATTR retourneert dan een getal tussen Ø en 255. Dit getal is de som van de bijkomstigheden of attributen bij de specifieke positie (op het beeldscherm), en is als volgt samengesteld:

kleur van de inkt	kleur code (Ø - 7)
kleur van het papier	8x de kleurcode
helderheid	64
knipperen	128

Bijvoorbeeld:
Als een character inpositie 11,16 wordt afgebeeld in inkt-kleur 3 magenta, papierkleur 6 (geel) en is helder maar niet knipperend dan beeld de opdracht:

PRINT ATTR (11,16)

115 af, dat is (3 + 8 x 6 + 64 + Ø).

ATTR in binaire vorm
ATTR retourneert een byte waarin bij bit 7 (erg belangerijk) 1 is voor knipperen in Ø voor normaal; bij bit 6 is 1 voor helder en Ø voor normaal bit 5 tot en met bit 3 geven de papierkleur binair aan. bit 2 tot en met bet Ø geven de kleur van de inkt binair aan.

Format
AATR (num-expr, num-expr)

BEEP

Plaats op het toetsenbord
Extend modus
Symbool Z

Statement/opdracht

BEEP zorgt er voor dat de luidspreker een enkele toon van een bepaalde lengte en hoogte afgeeft.
Hoe gebruik je BEEP?
BEEP kan worden gebruikt om een statement in een programma of een directe opdracht te vormen. Het wordt gevolgd door twee numerieke values gescheiden door een komma, bijvoorbeeld:

8Ø BEEP x,y

De eerste value (x) kan varieren van Ø tot 1Ø en bepaald de duur van de toon is seconden. De tweede value (y) kan varieren van –6Ø tot 69 en bepaald de hoogte van de toon in de semi-tonen lager dan midden C indien negatief en hoger dan midden C indien positief.
Bijvoorbeeld:
De opdracht

BEEP Ø, 5 1

veroorzaakt de toon C# boven midden C in deze toon duurt een halve seconde.

Format
BEEP num-expr, num-expr.

BIN BINary nummer

Plaats op het toetsenbord
Extend modus
Symbool B

Functie

BIN verandert een binair getal in een decimaal getal.
Hoe gebruik je BIN?
BIN wordt gevolgd door een binair getal bestaande uit maximaal zestien enen en nullen, bijvoorbeeld:

5Ø POKE USR "a", BIN 1Ø1Ø 1Ø1Ø

BIN retourneert de decimale value van het binaire getal. Het wordt vooral gebruikt/samen met POKE en USR, zoals hierboven, om zo de door de gebruiker bepaalde, grafics characters te maken. Hierbij is de 1 steeds een pixel inkt-kleur en de steeds een pixel papier kleur.

Bijvoorbeeld:
De opdracht

PRINT BIN 1111111Ø

afgebeeld wordt 254, de decimale value van het hierboven weergegeven binaire getal.

Format
BIN [1] [Ø]

BORDER

Plaats op het toetsenbord
Symbool B

Statement/opdracht

BORDER bepaald de kleur van de grens rond het afbeeldings-gebied op het beeldscherm.
Hoe gebruik je BORDER?
BORDER kan worden gebruikt als een directe opdracht of als een statement in een programma. Het wordt gevolgd door een numerieke value, bijvoorbeeld:

3Ø BORDER RND*7

De value van BORDER wordt afgerond naar het dichtst bij zijnde gehele getal en bepaald de kleur van het grens-gebied als volgt:

Ø zwart
1 blauw
2 rood
3 (magenta) paars
4 groen
5 cyaan
6 geel
7 wit

Merk op dat BORDER ook de papier kleur van het onderste gedeelte van het scherm bepaald. In tegenstelling tot een INK- of PAPER-statement kan de BORDER-statement niet in een PRINT-statement worden opgenomen.
Bijvoorbeeld:
De opdracht

BORDER 2

Levert een rode grens rond het afbeeldings gebied op.

Format
BORDER int-num-expr.

BRIGHT

Plaats op het toetsenbord
Extend modus
Symbool B

Statement/opdracht

BRIGHT zorgt er voor dat characters in helderder kleuren worden afgebeeld dan normaal.
Hoe gebruik je BRIGHT?
BRIGHT kan worden gebruikt als

een directe opdracht, maar wordt gewoonlijk gebruikt om een statement te vormen in een programma. Het wordt gevolgd door een numerieke value, bijvoorbeeld:

8Ø BRIGHT 1

De value die volgt op BRIGHT wordt afgerond naar het dichtst bij gelegen gehele getal. De value mag dan Ø,1 of 8 zijn. Een value van 1 veroorzaakt dat alle characters. achtereenvolgens worden afgebeeld door PRINT of INPUT statements zodat zij verschijnen in helderder inkt- en papier- kleur. Een value van 8 veroorzaakt dat een helder character-positie helder blijft en normale character posities normaal blijven als daar nieuwe characters worden gedrukt. BRIGHT gevolgd door een Ø last zowel BRIGHT, als BRIGHT 8 af zodat alle characters achtereenvolgens normaal worden afgebeeld. BRIGHT kan ook worden ingebracht in afbeeldings-statements gevormd door PRINT, INPUT, PLOT, DRAW en CIRCLE.

BRIGHT volgt het keyword maar komt voor de data of afbeeldings parameters: BRIGHT wordt gevolgd door dezelfde values en een punt-komma, bijvoorbeeld:

5Ø PRINT BRIGHT 1
"WARNING"

Het effect van BRIGHT is dan plaatselijk en geldt alleen voor de afgebeelde characters, in puntjes of als een getrokken lijn afgebeeld door de afbeeldings-statement. Merk op dat BRIGHT 1 de kleur van het papier helderder maakt over een oppervlakte zo groot als de hele character positie (8 x 8 pixels) indien ook maar een pixel daar wordt afgebeeld in een inkt-kleur.

Format
BRIGHT int-num-expr. [;]

CAT CATalogue

Microdrive file-handling opdracht. Zie Microdrive en Interface 1 Handboek.

CHR$ CHaRacter string

Plaats op het toetsenbord
Extend modus
Symbool U

Functie

De characters en keywords die beschikbaar zijn op het toetsenbord plus iedere door de gebruiker bepaalde graphics-characters, maken samen de Spectrum character verzameling. Door nu CHR$ en een code nummer te gebruiken kun je ieder character als een rij krijgen. De character verzameling bevat ook diverse contrôle-codes die de afbeelding van de characters beinvloeden. Deze codes kunnen in werking worden gesteld en de characters worden algebeeld, door PRINT te gebruiken voor CHR$. De volledige character verzameling en de code-nummers kunnen worden gevonden op blz. 51.

Hoe gebruik je CHR$
CHR$ wordt gevolgd door een numerieke value, bijvoorbeeld:

8Ø PRINT CHR$ x

Een uitdrukking moet tussen haakjes worden geplaatst. De value volgend op CHR$ (x boven) wordt afgerond naar het dichtst bij zijnde gehele getal. Als de value binnen het bereik 32 tot 255 valt, retourneert CHR$ een toetsenbord character of een door de gebruiker bepaald graphics character of een keyword als een rij. De Spectrum gebruikt de ASCII code voor values van 32 tot 95 en van 97 tot 126. Als x een value van 65 bepaald, dan beeld de bovenstaande statement bij voorbeeld een A af.

CHR$ contrôle codes
Values van 1 tot 31 brengen contrôle-codes terug of worden niet gebruikt. CHR$ 6 (PRINT komma). CHR$ 8 (back-space) en CHR$ 13 (new line of ENTER), beïnvloeden afbeeldingen op het scherm als er een PRINT statement bij betrokken is. CHR$ kan worden gevolgd door de code value en een punt-komma, bijvoorbeeld:

6Ø PRINT "A"; **CHR$**6; "B"

Deze statement laat zien:

A B

Een andere manier waarop je CHR$ contrôle codes kunt gebruiken is het vormen van een samengestelde rij welke de contrôle codes bevat. De statement:

6Ø PRINT "A" + CHR$ 6 + "B"

heeft precies hetzelfde effect als de vorige statement.

De codes 16 tot 23 beïnvloeden kleur en plaats. Elk van deze codes kan in een samengestelde rij worden gebruikt, samen met CHR$, gevolgd door een kleuren code value van Ø tot 7 voor CHR$ 16 (INK contrôle) en CHR$ 17 (PAPER contrôle), of gevolgd door Ø of 1 voor CHR$ 18 tot CHR$ 21 (FLASH, BRIGHT, INVERSE en OVER controles). De opdracht:

PRINT CHR$ 16 + CHR$ 3 + CHR$ 1 + ZX SPECTRUM +

Laat ZX SPECTRUM+ zien, knipperend in rood en geel. In een statement, zoals hierboven, kan ieder + teken worden vervangen door een punt-komma CHR$ 22 (AT Controle) wordt gevolgd door twee CHR$ values die de line en de kolom nummers aangeven. De opdracht:

PRINT CHR$ 22 + CHR$ 11 + CHR$ 16 + CHR$ 42

Laat een ster zien in het midden van het beeldscherm.

CHR$ 23 (TAB controle) wordt ook gevolgd door twee values, op dezelfde manier. De tweede value is gewoonlijk Ø en de eerste value geeft de TAB plaats aan. De opdracht:

PRINT CHR$ 23 + CHR$ 16 + CHR$ Ø + CHR$ 42

Laat een ster zien, halverwege het beeldscherm.

Merk op dat alleen deze controles beschikbaar zijn. Het gebruik van PRINT CHR$ met een keyword value groter dan 164 laat alleen het keyword zien en brengt het niet in werking.

Format
CHR$ int-num-const [;] [+]
CHR$ int-num-var [;] [+]
CHR$ (int-num-expr) [;] [+]

CIRCLE

Plaats op het toetsenbord
Extend modus
Symbool H

Statement/opdracht

CIRCLE TEKENT een cirkel op het beelscherm.

Hoe gebruik je CIRCLE?
CIRCLE wordt gevolgd door drie numerieke values steeds gescheiden door een komma, bijvoorbeeld:

8Ø CIRCLE x, y, z

Elk van deze drie values wordt afgerond naar het dichtst bij zynde gehele getal, als dat nodig is. CIRCLE tekent dan een cirkel op het zeer fijne afbeeldings rooster met de te gebruiken inkt-kleur. De beide eerste values (x, y) bepalen de horizontale en verticale coördinaten van het midden van de cirkel, en de derde value (z) bepaald de lengte van de straal. De afmetingen moeten zodanig zijn dat de cirkel niet

buiten het afbeeldings-gebied komt.

CIRCLE wordt beïnvloed door kleur statements of opdrachten en kan ingebrachte kleur statements bevatten met dezelfde gevolgen als PLOT en DRAW.

Bijvoorbeeld
De opdracht:

CIRCLE 128, 88, 87

beeld een cirkel af die het grootste gedeelte van het afbeeldings gebied gebruikt.

Format
CIRCLE statement; int-num-expr, int-num-expr, int-num-expr.

CLEAR

Plaats op het toetsenbord
Symbool X

Statement/opdracht

CLEAR verwijdert de bestaande values van alle variabelen en maakt de geheugen ruimte vrij die door deze values werd ingenomen, en ruimte tot zover als RAMTOP, het hoogste, address van het BASIC systeem bereikt. CLEAR kan ook worden gebruikt om RAMTOP te herstellen.

Hoe gebruik je CLEAR?
CLEAR kan worden gebruikt als een directe opdracht of het vormt een statement in een programma. CLEAR vereist géén parameters, bijvoorbeeld:

5Ø CLEAR

CLEAR verwijdert de values die waren toegewezen aan alle variabelen, inclusief arrays CLEAR voert ook CLS en RESTORE uit om het beeldscherm leeg te maken en de data-pointer terug te brengen naar het eerste data item. Ook wordt de PLOT positie terug gebracht naar de linker-beneden hoek van het afbeeldings gebied en wordt ook de GOSUB stack uitgewist. Merk op dat CLEAR niet nodig is voor her-dimensionerings arrays omdat DIM een bestaand array met dezelfde naam uitwist. Merk tevens op dat RUN CLEAR uit voerd.

CLEAR en RAMTOP
CLEAR kan ook door een numerieke value worden vervolgd, bijvoorbeeld:

CLEAR 65267

CLEAR voert dan CLEAR uit zoals hierboven en stelt ook nog RAMTOP, het hoogste address van het BASIC-systeem bereik, af op de gegeven value RAMTOP is gesteld op 65267 in de ZX Spectrum+ en ligt onder het gebied dat is gereserveerd voor de door de gebruiker bepaalde graphics. NEW wist het geheugen uit tot aan RAMTOP en gebruikt CLEAR om RAMTOP te verlagen (in het gegeven voorbeeld met 1ØØ bytes) hierdoor levert NEW ook meer geheugen dat immuun is voor NEW. Verhoging van RAMTOP levert meer ruimte voor BASIC ten koste van de door de gebruiker bepaalde grafics. Merk op dat de GOSUB stack dan te vinden is bij RAMTOP, mits RAMTOP boven de calculator stack blijft. Om RAMTOP te vinden moet je invoeren:

PRINT PEEK 23730+ 256***PEEK** 23731

Format
CLEAR [num-expr.]

CLOSE

Plaats op het toetsenbord
Symbool #
Microdrive file handling opdracht. Zie Microdrive en Interface 1 manual.

CLS CLear Screen

Plaats op het toetsenbord
Symbool V

Statement/Command

CLS verwijdert alle tekst en graphics van het afbeeldings gebied en laat het leeg in de (achtergrond) papier kleur van dat moment.

Hoe gebruik je CLS?
CLS kan worden gebruikt als een directe opdracht het kan ook een statement vormen in een programma. CLS vereist geen parameters, bijvoorbeeld:

25Ø IF a$ = "NO" THEN CLS

Het afbeeldings gebied (maar niet de grens) wordt dan uitgewist op de kleur na. Deze kleur was bepaald door de eerder gegeven PAPER statement of de opdracht of bepaald door papier kleur wit Merk/op dat CLS gebruikt moet worden na PAPER en voor PRINT en iedere andere afbeeldings-statement, om een gekleurde achtergrond te produceren over het gehele afbeeldings gebied.

Format
CLS
code entry

CODE

Plaats op het toetsenbord
Extend modus
Symbool 1

Functie
CODE geeft het code nummer van een character in de Spectrum character-verzameling (zie blz 51).
Hoe gebruik je CODE?
CODE wordt gevolgd door een rij, bijvoorbeeld:

9Ø IF CODE a$ < 65 OR CODE a$ > 9Ø THEN GOTO 8Ø

Een rij-uitdrukking moet tussen haakjes worden geplaatst. CODE retourneert het code nummer van het eerste character in de rij. Als dit een nul rij (" ") is, dan retourneert CODE Ø.
CODE geeft ASC11 code values van 32 tot 95 en van 97 tot 126.
Bijvoorbeeld:
De opdracht;

PRINT CODE "ZX SPECTRUM+"

geeft te zien: 9Ø, het code nummer van Z.
SAVE CODE en LOAD CODE
CODE wordt anders gebruikt bij SAVE en LOAD. Zie SAVE CODE en LOAD CODE

Format
CODE string-const
CODE string-var
CODE (string-expr)

CONTINUE

Plaats op het toetsenbord
Symbool C

Opdracht

Als een programma stopt, kan CONTINUE worden gebruikt om het programma opnieuw te starten, van af het punt waar het stopt. Als een fout er de oorzaak van is dat het programma stopte, moet deze eerst worden hersteld, voordat CONTINUE het programma weer kan starten.

Hoe gebruik je CONTINUE?
CONTINUE wordt gebruikt als een directe opdracht als een programma stopt. CONTINUE vereist geen parameters. Na CONTINUE begint het programma gewoonlijk weer waar het stopte. Als de oorzaak een fout was dan kan de opdracht worden ingevoerd om de fout te herstellen en CONTINUE laat het

programma weer beginnen bij de statement waar bij het stopte. Als het programma stopte bij een STOP statement dat report 9 geeft of wanneer het stopte omdat de BREAK toets werd ingedrukt (geeft report L) dan laat CONTINUE het programma bij de eerst volgende statement beginnen. Een correctie opdracht kan pas worden ingevoerd als het nodig is. Als CONTINUE wordt gebruikt om een directe opdracht te hervatten, zal het in een loop gaan als de opdracht stopt bij de eerste statement in de opdracht. De afbeelding verdwijnt, maar controle kan worden teruggekregen door de BREAK toets in te drukken. CONTINUE levert report Ø als de opdracht stopt bij de tweede statement en CONTINUE geeft report N als de opdracht stopt bij de derde (of volgende) statements.

Format
CONTINUE

COPY

Plaats op het toetsenbord
Symbool Z

Opdracht

COPY zorgt er voor dat de Sinclair type-printers een kopie produceren van de afbeelding op het scherm.

Hoe gebruik je COPY?
COPY wordt gebruikt als een direct opdracht wanneer een programma af is, of is gestopt. COPY vereist geen parameters. Na COPY worden de eerste 22 lines van het beeldscherm gekopieerd en afgedrukt, uiteraard alleen als er een printer is aangesloten. Merk op dat alle inkt kleuren van de voor grond zwart worden afgedrukt. De papier kleur van de achtergrond wordt niet gedrukt. De printer kan worden gestopt door de BREAK toets in te drukken.

Als er een programma listing op het scherm verschijnt, kan dit worden afgedrukt als je COPY gebruikt, maar dan moet het geproduceerd zijn door een LIST-opdracht of-statement. Merk op dat een listing op het scherm komt door op de ENTER toets te drukken nadat een programma af is gewerkt of is gestopt. Deze automatische listing kan echter niet worden afgedrukt met COPY.

Format
COPY

COS COSine

Plaats op het toetsenbord
Extend modus
Symbool W

Functie

COS geeft de cosinus van een hoek

Hoe gebruik je COS?
COS wordt gevolgd door een numerieke value, bijvoorbeeld:

```
14Ø LET x = COS y
```

Een uitdrukking moet tussenhaakjes worden geplaatst. De value volgend op COS is de hoek in radialen. COS retourneerd dan de cosinus van de hoek. Graden kunnen worden omgezet in radialen door te vermenigvuldigen met π/18Ø.

Merk op dat COS een negatieve value geeft voor hoeken van 9Ø tot 27Ø graden en een positieve value voor hoeken van Ø tot 9Ø en van 27Ø tot 36Ø graden.

Bijvoorbeeld
De opdracht:

```
PRINT COS (6Ø*PI/18Ø)
```

laat Ø.5 zien, de cosinus van 6Ø graden.

Format
COS num-const
COS num-var
COS (num-expr)

DATA

Plaats op het toetsenbord
Extend modus
Symbool D

Statement

DATA levert een lijst data-items op binnen een programma. Deze items kunnen values of variabelen of rijen zijn bijvoorbeeld om te worden afgebeeld. Ieder item hoort bij een variabele door een READ statement.

De toewijzing van items bij variabelen gebeurt in de volgorde waarin de items of data voorkornen in het programma. Je kunt de RESTORE toets gebruiken om de toewijzing te laten beginnen bij het eerste item in een gegeven DATA statement.

Hoe gebruik je DATA?
DATA kan alleen worden gebruikt om een statement te vormen in een programma. Het wordt gewoonlijk gevolgd door een lijst van numerieke – of rij – constanten steeds gescheiden door een komma, bijvoorbeeld:

```
5Ø DATA 31, "JAN", 28,
    "FEB"
```

Iedere constante wordt dan toegewezen aan een variabele door een READ statement dat de DATA leest. De DATA statement kan overal in het programma plaatsen. Het aantal, soort (numeriek, rij) en de volgorde van de constanten moet overeenkomen met het aantal keren dat de READ statement is uitgevoerd en de soort en volgorde van de variabelen in de READ statement. De lijst van data kan gesplitst zijn in verschillende op elkaar volgende DATA-statements als er teveel items zijn om in één statement te kunnen passen.

Bijvoorbeeld
Het volgende programma:

```
1Ø FOR n=1 TO 2
2Ø READ x,a$
3Ø PRINT a$,x; "days"
4Ø NEXT n
5Ø DATA 31,"JAN", 28, "FEB"
```

laat zien:

JAN	31 days
FEB	28 days

Het gebruik van DATA met variabelen
De items van data in een DATA statement kunnen bestaan uit numerieke – of rij – variabelen of uitdrukkingen, onder de voorwaarde dat er van de variabelen eerder values zijn toegewezen. In het hierboven gegeven voorbeeld kan de DATA statement worden verandert in:

```
5Ø DATA d, m $, d–3, "FEB"
```

Als aan d eerder een value van 31 is toegewezen en aan m $ een value van "JAN", dan krijg je hetzelfde te zien.

LOAD DATA, SAVE DATA en VERIFY DATA
DATA kan ook worden gebruikt met LOAD, SAVE en VERIFY om arrays op band te zetten. Zie LOAD DATA, SAVE DATA en VERIFY DATA.

Format
DATA num-expr [num-expr.] [string-expr.]
DATA string-expr. [num-expr] [string-expr.]

DEF FN DEFine FuNction

Plaats op het toetsenbord
Extend modus
Symbool 1

Statement

DEF FN geeft de gebruiker de mogelijkheid een functie te omschrijven waarvoor geen keyword is. Allerlei parameters kun je aan de functie geven in een FN statement. Deze FN statement roept de functie op en kan zowel numerieke values als rij-values retourneren als resultaat.

Hoe gebruik je DEF FN?
DEF FN kan alleen worden gebruikt als een statement in een programma. Als je een numereike

functie moet beschrijven, wordt DEF FN gevolgd door een enkele letter en dan door een of meer numerieke variabelen. Deze worden allemaal gescheiden door door komma's en tussen haakjes geplaatst.
Bijvoorbeeld: Def Fn r (x,y). Dit wordt gevolgd door een gelijkteken en dan een numerieke uitdrukking die de variabelen bevat, bijvoorbeeld:

```
1ØØØ DEF FN r (x,y) =
        SQR (x ↑ 2 + y ↑ 2)
```

De letter volgend op DEF FN (r boven) is een naam die de functie aanduidt. De variabelen kunnen ook enkele letters zijn. Merk op dat in beide gevallen, de Spectrum geen onderscheidt maakt tussen hoofdletters en gewone letters.

Een DEF FN statement kan overal in een programma worden geplaatst. Om de functie die het omschrijft op te roepen wordt een FN statement gebruikt. Dit wordt dan gevolgd door de letter die de functie aanduidt en een lijst numerieke values, allemaal van elkaar gescheiden door een komma en tussen haakjes geplaatst, bijvoorbeeld:

```
5Ø PRINT FN r (3,4)
```

De values tussen de haakjes worden in dezelfde volgorde aan de functie doorgegeven als de variabelen in de DEF FN statement. Daarom is aan x de value 3 en aan y de value 4 toegekend (in boven genoemd voorbeeld). FN evalueert de uitdrukking en geeft de value.

DEF FN kan ook worden gevolgd door een letter en haakjes, bijvoorbeeld:

```
1ØØØ DEF FN r() = INT (x +
Ø.5)
```

De value die was toegewezen aan x (boven) wordt door gegeven aan de functie als deze door FN wordt opgeroepen. In dit geval zal FN r () de value welke aan x was toegewezen, afgerond naar het dichtst bij zijnde gehele getal retourneren.

DEF FN en rijen.
DEF FN en FN kunnen ook op dezelfde manier worden gebruikt om een rij-functie te omschrijven of op te roepen. In dit geval is de naam van de functie een enkele letter gegolgd door $ en een of meer van de variabelen in de statement is een letter gevolgd door $. Een overeenkomstige rij-uitdrukking vormt de definitie van de functie, bijvoorbeeld:

```
1ØØØ DEF FN a $ ( b$, x, y) =
        b $ (x TO Y)
```

De rij-uitdrukking na het getijkteken is in dit voorbeeld een rij-knipper, en x en y zijn de eerste en de laatste characters van een sectie van b$. FN moet worden gevolgd door de naam van de functie en, tussen haakjes, een rij-value samen met de andere parameters welke aan de functie moeten worden doorgegeven. In dit geval zal de opdracht:

```
PRINT FN a $
("FUNDAMENTAL", 1, 3)
```

FUN te ziengeven, en de opdracht:

```
PRINT FN a $
("FUNDAMENTAL", 5,8)
```

laat AMEN zien.

Format
DEF FN letter ([letter] [letter]) = num-expr.
DEF FN letter $ ([letter $] [letter] [letter] [letter$]) = string-expr
FN letter ([num-expr] [num-expr])
FN letter $ ([string-expr] [num-expr] [num-expr] [string-expr]).

DIM DIMension

Plaats op het toetsenbord
Symbool D

Statement

DIM wordt gebruikt om de dimensie van een array van een gegeven aantal numerieke variabelen of rij variabelen te bepalen. Een array is een lijst van variabelen met dezelfde naam, die worden onderscheiden met behulp van subscripts (values die iedere variabele en elk element in de array onderscheiden van elkaar).

Hoe gebruik je DIM met numerieke arrays?
DIM wordt gebruikt om een statement in een programma te maken. DIM wordt gevolgd door een enkele letter die de betrokken array aanduidt, en een of meer numerieke values, steeds gescheiden door komma's en tussen haakjes geplaatst, bijvoorbeeld:

```
2Ø DIM x (1Ø)
8Ø DIM z (2Ø,5)
```

In het eerste geval wordt een 1-dimensionaal numerieke array gevormd dat 1Ø elementen bevat met subscripts van 1 tot 1Ø. Het array heeft de naam x en de subscripted variabelen zijn x (1) tot x (1Ø). Iedere bestaande array met dezelfde naam wordt verwijdert en aan de variabelen wordt een value Ø toegewezen. Merk op dat bij de dimensionering van een array, de Spectrum geen onderscheid maakt tussen namen met een hoofdletter en die met eengewone letter – variabele x (2) is hetzelfde als X (2). Echter, eenvoudige numerieke variabelen met dezelfde letter als de naam van een array (x of X) kunnen naast elkaar bestaan en kunnen gescheiden gebruikt worden, indien nodig.

Het aantal values tussen haakjes is gelijk aan het aantal dimensies dat wordt gevormd in een numeriek array. Het tweede voorbeeld vormt een twee-dimensionaal array van 1ØØ elementen met 2Ø elementen in de eerste dimensie en 5 elementen in de tweede dimensie. Deze elementen zijn genummerd: z (1,1) tot en met z (2Ø,5).

Je kunt arrays maken met net zoveel dimensies als je wilt. De elementen van een numerieke array kunnen achtereenvolgens worden geïndentificeerd met behulp van de array naam gevolgd door een value tussen haakjes, bijvoorbeeld:

```
7Ø PRINT x (a)
16Ø PRINT x (7,b)
```

DIM en string-arrays (of rij-arrays) DIM wordt op dezelfde manier gebruikt als met numerieke arrays, behalve dan dat een enkele letter die wordt gevolgd door $, wordt gebruikt voor de naam van de array. Verder moet een extra value worden toegevoegd aan de dimensie-values, tussen haakjes, dit is nodig om de lengte van de rij (string) te bepalen, bijvoorbeeld:

```
3Ø DIM a$ (2Ø,5)
9Ø DIM b $ (2Ø,5,1Ø)
```

De eerste statement vormt een array van 2Ø elementen, waarvan elk een rij van 5 characters heeft. De subscripted variabelen worden a $ (1) tot en met a £ 2Ø genoemd, en hen werd aanvankelijk een nul-rij (leeg) ("") toegewezen. Iedere bestaande array met dezelfde naam wordt verwijdert en, integenstelling tot de numerieke arrays, kan een eenvoudige rij-variabele met dezelfde naam niet naast de ander bestaan.

Het tweede voorbeeld vormt een twee-dimensionale rij-array van 1ØØ elementen, met 2Ø elementen in de eerste dimensie en 5 elementen in de tweede dimensie. Alle elementen hebben een lengte van 1Ø characters. Als rij-values achtereenvolgens worden toegewezen aan een rij array, dan worden zij uitgesmeerd met open ruimten aan het eind vande rij of uitgerekt tot de bepaalde lengte indien nodig. De elementen van een rij-array

worden geidentificeerd door de array naam waar achter, tussen haakjes, een of meer numerieke values staan. Deze geven het subscript nummer of de subscript nummers. Bijvoorbeeld, element a $ (2) kan "SMITH" zijn en element b $ (12, 4) kan "DERBYSHIRE" zijn. Er kan echter een extra value worden toegevoegd om een speciaal character te definieren in de rij. In deze voorbeelden, a $ (2,2) zou "M" zijn (het tweede character in "SMITH"), en b $ (12,4,5) zou zijn "Y", (het vijfde character in "DERBYSHIRE").

Num-dimensie rij-arrays.
Het is mogelijk een nul-dimensie rij-array te vormen door slechts een value, tussen haakjes, te gebruiken, bijvoorbeeld:

```
1Ø DIM c $ (15)
```

Deze array heeft slechts een element, dit is c $, en de lengte er van is vastgesteld op de bepaalde value (15 characters).

Format
DIM letter (num-expr [num-expr])
DIM letter $ (num-expr [num-expr]).

DRAW

Plaats op het toetsenbord
Symbool W

Statement/opdracht

DRAW wordt gebruikt om rechtelijnen en krommen op het scherm te tekenen.

Hoe gebruik je DRAW?
DRAW wordt gewoonlijk gebruikt om een statement te vormen in een programma. Als er een rechte lijn nodig is, wordt DRAW gevolgd door twee numerieke values, gescheiden door een komma, bijvoorbeeld:

```
4Ø DRAW x, y
```

Er wordt dan een rechtelijn getekent op het verfijnde graphics rooster, vanuit de positie welke werd bepaald door het voor afgaande PLOT statement, ofwel vanuit de positie bereikt door een voor afgaande DRAW statement, in ieder geval de laatste van de twee. Beide values volgend op DRAW worden naar het dichtst bij zijnde gehele getal afgerond, indien dit nodig is. De eerste value (x boven) bepaald de horizontale afstand vanuit deze positie, en de tweede value (y boven) bepaald de verticale afstand. Deze values zijn negatief als de lijn naar links moet gaan, of naar beneden. De bereikte positie moet binnen het afbeeldingsgebied liggen. Als er geen voorafgaande PLOT of DRAW statement is, begint DRAW op de plaats met coördinaten Ø,Ø (de hoek links onder van het scherm).

DRAW wordt beïnvloedt door kleur statements of kleur opdrachten en kan ingebouwde statements bevatten, die met de zelfde gevolgen als met PLOT en CIRCLE.

DRAWing gebogen lijnen (krommen).
DRAW kan worden gevolgd door een derde value om zo een kromme te produceren, bijvoorbeeld een gedeelte van een cirkel. Bijvoorbeeld:

```
4Ø DRAW x,y,z
```

De derde value (z boven) bepaald de hoek (in radialen) over welke de lijn draait als zij wordt getekend. De lijn draait naar links als de waarde positief is, en naar rechts als de waarde negatief is. Values π of $-\pi$ produceren een cirkel.

Bijvoorbeeld:
Het volgende programma tekent een driehoek:

```
1Ø PLOT 127,15Ø
2Ø DRAW 7Ø, –1ØØ
3Ø DRAW –14Ø, Ø
4Ø DRAW 7Ø, 1ØØ
```

Het toevoegen van 1 of –1 aan de DRAW statement veroorzaakt dat de zijden naar binnen respectievetijk naar buiten uit buigen.

Format
DRAW ([statement:]) int-num-expr, int-num-expr [int-num-expr]

ERASE

Microdrive file-handling opdracht. Zie Microdrive en Interface 1 manual.

EXP EXPonent

Plaats op het toetsenbord
Extend modus
Symbool X

Functie

EXP is een mathematische (wiskundige) functie welke de exponent e tot a kan verheffen.

Hoe gebruik je EXP?
EXP wordt gevolgd door een numerieke value, bijvoorbeeld:

```
6Ø LET y = EXP x
```

Een uitdrukking moet tussen haakjes staan. EXP geeft dan de exponent e verheven tot de macht van het argument (x boven)

Bijvoorbeeld
De opdracht:

```
PRINT EXP 1
```

laat zien 2,7182818, de value van e.

Format
EXP num-const
EXP num-var
EXP (num-expr)

FLASH

Plaats op het toetsenbord
Extend modus
Symbool V

Statement/opdracht

FLASH zorgt er voor dat de plaatsen van de characters knipperen, zodet de inkten papierkleuren afwisselen in een constant tempo.

Hoe gebruik je FLASH?
FLASH kan worden gebruikt als een directe opdracht maar wordt gewoonlijk gebruikt om een statement te vormen in een programma. Het wordt gevolgd door een numerieke value, bijvoorbeeld:

```
5Ø FLASH 1
```

De value volgend op FLASH wordt als dat nodig is afgerond naar het dichtst bij zijnde gehele getal en kan dan Ø ofwel 1 of 8 zijn. Een value van 1 zorgt er voor dat alle characters die achtereenvolgens worden afgebeeld door PRINT of INPUT, gaan knipperen. Een value van 8 laat knipperende characters knipperen en laat andere characters met rust, als er nieuwe characters bij gedrukt worden. FLASH gevolgd door Ø last zowel FLASH 1 als FLASH 8 of zodat alle achtereenvolgens afgebeelde characters normaal zijn.

FLASH kan ook ingebouwd zijn in de afbeeldings statements gevormd door: PRINT, INPUT, PLOT, DRAW en CIRCLE. FLASH volgt het keyword maar gaat voor af aan de data en afbeeldings parameters. FLASH wordt gevolgd door dezelfde values en een punt-komma, bijvoorbeeld;

```
12Ø PRINT FLASH 1; INK 2;
      PAPER 6; "WARNING"
```

Het effect van FLASH is dan plaatselijk en is alleen van toepassing op de afgebeelde characters, puntsgewijs uitgestippeld of als getrokken lijn, door de afbeeldings-statement. Merk op dat FLASH 1 er voor zorgt dat de hele 8 x 8 pixel positie knippert als ook maar een pixel in een inkt kleur is aangestipt.

Format
FLASH int-num-expr [;]

FN FuNction

Plaats op het toetsenbord
Extend modus
Symbool 2

Functie

FN roept een door de gebruiker gedefinieerde functie op. FN wordt altijd gebruikt samen met DEF FN, welke de opgeroepen functie beschrijft.

Hoe gebruik je FN?
Als er een numerieke functie moet worden opgeroepen, wordt FN gevolgd door een letter en een paar haakjes. Als er parameters aan de functie worden gegeven, dan worden deze, gescheiden door een komma, tussen haakjes geplaatst, bijvoorbeeld:

```
17Ø LET x = FN r (3,4)
```

De parameters (3 en 4 boven) worden dan door gegeven aan de functie r. FN geeft dan het resultaat. Als er geen parameters worden door gegeven moeten de haakjes toch worden weergegeven, bijvoorbeeld:

```
7Ø PRINT FN r ()
```

In dit geval gebruikt de functie de values die zijn toegewezen aan zijn variabelen.

FN roept op dezelfde wijze een rijfunctie op, met het verschil dat **$** moet worden toegevoegd na de letter welke de functie een naam geeft. Voor meer details zie DEF FN.

Format
FN letter ([num-expr] [num-expr])
FN letter **$** ([string-expr] [string-expr])

FOR

Plaats op het toetsenbord
Symbool F

Statement/opdracht

FOR wordt altijd gebruik met de keywords TO en NEXT om zodoende een FOR NEXT Loop te vormen. Deze gang van zaken zorgt er voor dat een bepaald gedeelte van het programma een vast-gesteld aantal malen herhaald kan worden.

Hoe gebruik je FOR?
FOR wormt altijd een statement met TO. FOR vordt gevolgd door een letter, een gelijk-teken en dan twee numerieke values, gescheiden door TO, bijvoorbeeld:

```
6Ø FOR a = 1 TO 9
```

De letter (a boven) is een contrôle variabele. De statements die moeten worden herhaald volgen en een of meer van deze statements maakt gewoonlijk gebruik van de contrôle variabele. De loop eindigt dan met een NEXT statement, waarin NEXT wordt gevolgd door een contrôle variabele, bijvoorbeeld:

```
9Ø NEXT a
```

Bij uitvoering verwijdert FOR iedere variabele met dezelfde naam als de contrôle variabele en wijst het er een initiele value, die gelijk is aan de value voor TO (1 boven), aan toe. De statements worden dan uitgevoerd terwijl de contrôle variabele de nieuwe value heeft. Als NEXT wordt bereikt wordt de value van de contrôle variabele met 1 vermeerderd. Als deze value hetzelfde of meer is dan de value na TO (de limiet value is 9 boven), dan keert het programma terug naar de FOR statement en de FOR NEXT loop wordt herhaald. Als de contrôle variabele een grotere value heeft dan de limiet value, dan stopt de loop en gaat het programma verder met de statement na NEXT.

In het voorbeeld hierboven, wordt de loop 9 keer herhaald. De contrôle variabele neemt dan toe van 1 tot 9 en beeft een value van 1Ø bij het verlaten van de laatste, loop. Merk op dat de Spectrum geen onderscheid maakt tussen hoofd letters en gewone letters bij benaming van de contrôle variabele.

Het gebruik van STEP in een FOR NEXT loop.
STEP is een keyword dat kan worden opgenomen in een FOR statement als de contrôle variabele toe-of afneemt met een value niet getijk aan 1. STEP volgt de limiet value en wordt gevolgd door een numerieke value, bijvoorbeeld:

```
6Ø FOR a = 1 TO 9 STEP 2
```

De contrôle variabele is vermeerderd met de step-value (2 boven) tot dat zij groter is dan de limiet value. De controle variabele heeft de op een volgende values 1,3,5,7 en 9 en verlaat de loop met een value van 11.

Een negatieve step value zorgt ervoor dat de value van de contrôle variabele afneemt. In dit geval moet de begin-value groter zijn dan de limiet value en zal de loop eindigen als de value van de contrôle variabele kleiner is dan de limiet value, bijvoorbeeld:

```
6Ø FOR a = 9 TO 1 STEP – 1
```

De value neemt af van 9 tot 1 en verlaat de loop met een value van Ø.

Nesting-loops
Een of meer FOR NEXT loops kunnen in-elkaar worden geplaatst, dit wordt, nesting genoemd. De volgorde van de controle variabelen in de NEXT statements moet tegengesteld zijn aan de volgorde van de controle variabelen in de FOR statements. FOR NEXT loops kunnen tot iedere diepte in elkaar geplaatst worden, Of wel, je kunt net zoveel loops binnen elkaar plaatsen als je wilt.

Format
FOR letter = num-expr **TO** num-expr **STEP** num-expr **NEXT** letter

FORMAT

Microdrive file-handling opdracht. Zie Microdrive en Interface 1 handbook

GOSUB

Plaats op het toetsenbord
Symbool H

Statement/opdracht.

GOSUB zorgt ervoor dat het programma afsplitst naar een subroutine. Dit is een afzonderlijke sectie van het programma. Dit is erg handig als een subroutine diverse malen nodig is in een programma.

Hoe gebruik je GOSUB?
GOSUB kan worden gebruikt als een statement of als een directe opdracht en het wordt gevolgd door een numerieke value, bijvoorbeeld:

```
GOSUB 1ØØØ
```

Bij uitvoering wordt de value volgend na GOSUB (1ØØØ boven) afgerond naar het dichtst bij zijnde gehele getal, en de programma takken naar het line nummer met deze value. Het gebruik van een variabele of een uitdrukking maakt het mogelijk dat het programma splitst naar een subroutine met een berekend line-nummer. Merk op dat indien het line nummer niet bestaat, het programma toch splitst en verder gaat met de eerste statement welke het dan tegenkomt.

Een subroutine eindigt met RETURN, en het programma springt dan terug naar de

statement volgend op de GOSUB statement. Subroutines kunnen zo binnen elkaar zitten dat de een vanuit de ander wordt bereikt, RETURN zal dan het programma terug sturen naar de statement die volgt op de laatst-uitgevoerde GOSUB statement.

De GOSUB-stack (stapelen)
Steeds als GOSUB wordt uitgevoerd, wordt het line-nummer geplaatst in de GOSUB stack in het geheugen. Als er twee of meer GOSUB worden uitgevoerd voor RETURN, dan worden de line-nummers zo opgeslagen dat het laatste nummer boven het eerste is gestapeld (in de stack). RETURN neemt altijd het bovenste line-nummer van de stack en gaat naar deze line om het programma te vervolgen. Merk op dat fout 4 (buiten het geheugen) kan voor komen als er niet voldoende RETURN statements zijn.

Format
GOSUB int-num-expr

GOTO

Plaats op het toetsenbord
Symbool G

Statement/opdracht

GOTO zorgt ervoor det het programma splitst naar een bepaalde line.

Hoe gebruik je GOTO?
GOTO kan worden gebruikt als een directe opdracht, om een programma te laten lopen van af een gegeven line-nummer zonder eerst het beeldscherm leeg te maken. Het kan ook worden gebruikt om een statement te vormen in een programma. GOTO wordt gevolgd door een numerieke value, bijvoorbeeld:

```
6Ø GOTO 35Ø
```

Bij uitvoering wordt de value volgend op GOTO afgerond naar het dichtst bij zijnde gehele getal en het programma splitst af naar de line met dat nummer. Het gebruik van een variabele of een uitdrukking staat het programma toe af te splitsen naar een line met een berekend nummer. Merk op dat als de line niet bestaat het programma toch splitst en verder gaat met de eerste statement die het tegenkomt.

Format
GOTO int-num-expr

IF

Plaats op het toetsenbord
Symbool U

Statement/opdracht

IF wordt altijd gebruikt met het keyword THEN om een beslissing aan te geven welke navolgende actie beinvloed. Om dit te kunnen doen, test de computer iets om uit te vinden of het wel of niet waar is. Als het waar is, dan volgt een bepaalde actie-lijn, als het daar en tegen niet waar is, volgt een andere actie-lijn.

Hoe gebruik je IF en THEN?
IF vormt gewoonlijk een statement met THEN. IF wordt eerst gevolgd door een numerieke value of door een vorwaarde en vervolgens door THEN en een of meer bruikbare BASIC statements, bijvoorbeeld:

```
8Ø IF x THEN GOTO 25Ø
24Ø IF a$ = "NO" THEN PRINT
      "THE END": STOP
```

Een constante variabele af uitdrukking (zoals x hierboven) wordt als waar of juist beschouwd als het een value heeft die niet Ø is. In dit geval zal de statement volgend op THEN en de andere statements op dezelfde line worden uitgevoerd. Het programma gaat dan verder naar de volgende line. Als de value Ø is, dan is de constante, de variabele of de uitdrukking vals of injuist. De er op volgende statements worden dan niet uitgevoerd en het programma springt naar de volgende line. In het voorbeeld zal het programma niet naar line 25Ø gaan als X = Ø.

Als een voorwaarde (a $ = "NO") volgend op IF, waar is, dan worden de statements volgend op THEN uitgevoerd. Als deze voorwaarde echter vals is, dan gaat het programma over op de volgende line. In dit voorbeeld zal, als a $ de value "NO" heeft, "THE END" worden afgebeeld en het programma stopt. Als a $ een andere value heeft, gaat het programma verder met de volgende line.

De Spectrum geeft een juiste voorwaarde een value van 1 en een onjuiste voorwaarde een value Ø. De Spectrum herkent een value niet gelijk aan Ø als onjuist. Aan een variabele kan de value van een voorwaarde worden toegewezen door een statement, zoals bijvoorbeeld:

```
7Ø LET x = a$ = "NO"
```

Merk op dat, in tegenstelling tot sommige andere BASIC's THEN niet voor GOTO geplaatst kan worden.

Format
IF num-expr **THEN** statement [:statement]
IF cond **THEN** statement [:statement]

IN

Plaats op het toetsenbord
Extend modus
Symbool 1

Functie

IN controleerd de status van het toetsenbord en andere input en output instrumenten. IN leest de byte van een gegeven port-addresś dat aangeeft welke status het met de port verbonden instrument heeft.

Hoe gebruik je IN?
IN wordt gevolgd door een numerieke value, bijvoorbeeld:

```
15Ø LET x = IN y
```

De value volgend op IN kan varieren van Ø tot 65535 en bepaald het, port-addresś dat gelezen moet worden. IN retourneert dan de gelezen byte.

Toetsenbord-addresses
Het toetsenbord heeft 8 addresses, waarvan elk een van vyf verschillende bytes bevat, afhankelijk van de toets die je indrukt. De addresses zijn: 65278, 65Ø22, 6451Ø, 63486, 61438, 57342, 4915Ø en 32766. Byte values bij deze addresses kunnen 175, 183, 187, 189 of 19Ø zijn.

Deze line houdt werking tegen tot dat de ENTER toets is ingedrukt en reageert dan zeer snel.

```
8Ø IF IN 4915Ø <> 19Ø
    THEN GOTO 8Ø
```

Iedere andere toets kan worden gevonden door het, port-addresś en de byte value te vervangen door een van de andere gegeven getallen.

Format
IN num-const
IN num-var
IN (num-expr)

INK

Plaats op het toetsenbord
Extend modus
Symbool X

Statement/opdracht

INK bepaald de voorgrond kleur waarin de characters worden afgebeeld, punten gestippeld en lijnen en kurven getrokken.

Hoe gebruik je INK?
INK kan worden gebruikt als een directe opdracht, maar wordt gewoonlijk gebruikt om een statement in een programma te vormen. Het wordt gevolgd door een numerieke value, bijvoorbeeld:

```
7Ø INK x
```

De value volgend op INK wordt afgerond naar het dichtst bij zijnde gehele getal en kan varieren van Ø tot 9. Dr volgende voorgrond kleuren zijn dan gegeven:

Ø	zwart
1	blauw

2	rood
3	paars (magenta)
4	groen
5	blauw-groen (cyaan)
6	geel
7	wit
8	transparent
9	contrasterend zwart of wit

INK 8 bepaald dat de bestaande kleur blijft op iedere positie op het scherm waar INK 8 wordt gebruikt. INK 9 zorgt er voor dat de kleur van de inkt of wit of zwart is zodat het afsteekt tegen het papier, de achtergrond kleur.

Plaatselijke – en niet-plaatselijke inkt kleuren.
Als INK alleen een statement vormt, zoals boven, is de kleur niet plaatselijk met als gevolg dat alle afbeeldingen verschijnen in deze voorgrond kleur. INK kan ook worden ingebouwd in afbeeldings-statements gevormd door: PRINT, INPUT, PLOT, DRAW en CIRCLE. INK volgt op het keyword, maar gaat voor af aan de data of afbeeldings parameters. Het wordt gevolgd door dezelfde values en een punt-komma, bijvoorbeeld:

```
6Ø CIRCLE INK 4; 128, 88, 87
```

Het effect van INK is dan wel plaatselijk en is alleen van toepassing op de afgebeelde characters, gestippeld of getrokken door de afbeeldings statement. In het voorbeeld wordt een groene cirkel getrokken. Daarna keert de inkt kleur terug tot niet-plaatselijk of ontbrekend of zwart.

Format
INK int-num-expr.

INKEY$ INput KEY rij (string)

Plaats op het toetsenbord
Extend modus
Symbool N

Functie

INKEY$ wordt gebruikt om het indrukken van de toetsen op het toetsenbord waar te nemen.

Hoe gebruik je INKEY$?
INKEY$ heeft geen argument nodig en wordt gewoonlijk gebruikt om een character toe te wijzen aan een rij-variabele of een bepaald character te proberen, bijvoorbeeld:

```
  7Ø LET a$=INKEY$
13Ø IF INKEY$="N" THEN
STOP
```

Bij uitvoering retourneert INKEY$ het character dat wordt gegeven door de toets die je indrukt, op dat moment. Als er geen toets wordt ingedrukt dan geeft INKEY$ een nul-rij (""). Merk op dat INKEY$ onderscheidt maakt tussen hoofd letters en gewone letters en andere verschoven en niet-verschoven characters. (gebruik IN om de toetsen te vinden zonder onderscheid te maken tussen de characters).

In tegenstelling tot INPUT, zal INKEY$ niet wachten maar direct door gaan naar de volgende statement. INKEY$ wordt daarom gewoonlijk binnen een loop geplaatst welke zich herhaald tot de gewenste toets in ingedrukt. Bijvoorbeeld:

```
6Ø IF INKEY$ <> "y" THEN
GOTO 6Ø
```

Deze line houdt verdere afwerking tegen tot dat je de y toets in drukt (zonder CAPS SHIFT of CAPS LOCK).

Format
INKEY$

INPUT

Plaats op het toetsenbord
Symbool 1

Statement/opdracht

INPUT maakt het mogeltijk data in te voeren terwijl het programma loopt.

Hoe gebruik je INPUT?
INPUT vormt gewoonlijk een statement in een programma en wordt op dezelfde manier gebruikt als PRINT. In zijn eenvoudigste vorm wordt het gevolgd door een numerieke variabele of een rij-variabele, bijvoorbeeld:

```
6Ø INPUT x
9Ø INPUT a$
```

De computer wacht dan tot dat er of wel een getal of een rij is ingevoerd. De value wordt afgebeeld aan het begin van de onderste line zodra het ingetoetst wordt. Als je dan op de enter toets drukt, wordt de value toegewezen aan de bepaalde variabele en het programma gaat verder.

Een INPUT statement kan meer dan een variabele omvatten en zal characters afbeelden om een aanzet te vormen. Dit wordt op precies dezelfde manier gedaan als bij PRINT; gebruikmakend van aanhalings tekens om de aanzet characters in te sluiten en puntkomma's of komma's om items te scheiden. Afbeeldings statements zoals INK, FLASH en PAPER kunnen worden ingebouwd, bijvoorbeeld:

```
8Ø INPUT INK 2; "What is your
    name?" :n$ ("How old are
    you, "+n$+"?"; age
```

Merk de volgende verschillen op met PRINT: INPUT wacht als het bij een variabele komt, dus alle variabelen en uitdrukkingen (zoals die die n$ bevat, boven) die in de aan zet moeten worden betrokken moeten tussen haakjes worden geplaatst. Afbeelding begint bij de start van de onderste line en schuift door als meer dan een line is gebruikt. AT kan worden gebruikt in een INPUT statement, op dezelfde manier als bij PRINT. AT ØØ beeld de start van de onderste line uit en de afbeelding schuift op als meer dan twee (regels) lines worden afgebeeld.

Hoe stop je INPUT?
Als INPUT wordt gevolgd door een numerieke variabele en STOP wordt ingevoerd, dan stopt het programma. Met een rij-variabele volgend op INPUT, kan het eerste aanhalings teken worden verwijdert en dan kan STOP worden ingevoerd om het programma te stoppen.

Het gebruik van INPUT met LINE.
INPUT LINE kan alleen met rij variabelen worden gebruikt. Gewoonlijk veroorzaakt INPUT met een rij-variabele dat een paar aanhalings tekens wordt afgebeeld. Als de rij wordt ingetoetst, verschijnt deze tussen de aanhalings tekens. Om deze aanhalings tekens te verwijderen moet je INPUT LINE gebruiken gevolgd door een rij variabele. Als een aanzet vereist is, wordt deze tussen INPUT en LINE geplaatst, bijvoorbeeld:

```
7Ø INPUT "What is your
name?";
    LINE n$
```

Format
INPUT [prompt] [;] [,] [']
num-var
INPUT [prompt] [;] [,] [']
string-var
INPUT [prompt] [;] [,] [']
LINE string-var
[prompt] = [string-const]
[(string-expr)] [AT int-num-expr, int-num-expr]
[statement] [;] [,] [']

INT INTeger

Plaats op het toetsenbord
Extend modus
Symbool R

Functie

INT verandert niet-gehele getallen in gehele getallen.

Hoe gebruik je INT?
INT wordt gevolgd door een numerieke value, bijvoorbeeld:

7Ø LET x = INT y

Een uitdrukking moet tussen haakjes worden geplaatst. INT geeft dan de value afgerond tot een geheel getal.
Bijvoorbeeld:
de opdracht:

PRINT INT 45, 67, **INT**-7, 66

laat zien:

45 –8

Format
INT num-const
INT num-var
INT (num-expr)

INVERSE

Plaats op het toetsenbord
Extend modus
Symbool M

Statement/opdracht

INVERSE zorgt er voor dat de kleuren worden aangebracht op de character posities zodat de inkt kleur de papier kleur wordt en anders om.

Hoe gebruik je INVERSE?
INVERSE wordt gewoonlijk gebruikt om een statement te vormen in een programma. Het wordt gevolgd door een numerieke value, bijvoorbeeld:

7Ø INVERSE 1

De value volgend op INVERSE wordt afgerond naar de dichtst bij zijnde integer en kan dan ofwel Ø ofwel 1 zijn. INVERSE 1 zorgt er voor dat alle opeen volgende afbeeldingen gemaakt door PRINT en INPUT in inverse kleuren verschijnen (inkt en papierkleur verwisseld) INVERSE Ø brengt de inkt en papierkleur terug naar normaal.

Merk op dat INVERSE kan worden ingebouwd in de afbeeldings-statements net zoals INK. Echter, indien het wordt gebruikt met CIRCLE, PLOT of DRAW, zorgt INVERSE, ervoor dat er een lijn of punt wordt gestippeld in de papier kleur, zodat het verdwijnt.

Format
INVERSE int-num-expr

LEN LENgth van de rij

Plaats op het toetsenbord
Extend modus
Symbool K

Functie

LEN geeft de lengte van een rij.

Hoe gebruik je LEN?
LEN wordt gevolgd door een rij-value, bijvoorbeeld:

5Ø LET x = LEN a$

Een uitdrukking moet tussen haakjes worden geplaatst. LEN geeft het aantal characters in de rij, bijvoorbeeld:

```
12Ø INPUT a$: IF LEN a$ > 9
     THEN GOTO 12Ø
```

Deze line laat alleen rijen door met 9 of minder characters.

Format
LEN string-const
LEN string-var
LEN (string-expr.)

LET

Plaats op het toetsenbord
Symbool L

Statement/opdracht

LET wordt gebruikt om een value toe te wijzen aan een variabele. In Sinclair-BASIC, kan LET niet worden geomiteerd in een toewijzings statement.

Hoe gebruik je LET?
LET vormt gewoonlijk een statement in een programma maar kan ook worden gebruikt als een directe opdracht. Het wordt gevolgd door een numerieke variabele of een rij-variabele, een getijkteken en daar na een value. De value kan zowel een numerieke value zijn als een rij-value, dit is afhankelijk van de variabele die aan LET voor af gaat. Bijvoorbeeld:

```
6Ø LET x=x+1
8Ø LET a$="Correct"
```

De value is dan toegewezen aan de variabele.

Merk op dat eenvoudige variabelen niet-gedefinieerd zijn tot dat er toegewezen values zijn door LET, READ of INPUT. Array variabelen worden (geïnitialiseerd) terug gebracht tot Ø of een nul-rij (" "), (Zie DIM).

Format
LET num-var=num-expr
LET string-var = string-expr

LINE

Plaats op het toetsenbord
Extend modus
Symbool 3

zie **SAVE**

LIST

Plaats op het toetsenbord
Symbool K

Statement/opdracht

LIST produceerd een listing van het lopende programma in het geheugen.

Hoe gebruik je LIST?
LIST wordt gewoonlijk gebruikt als een directe opdracht maar kan ook een statement vormen in een programma. Om een volledig programma te listeń wordt het alleen gebruikt. Na een directe opdracht;

LIST

verschijnt de eerste bladzijde van de listing en de er op volgende bladzijden zullen over het beeldscherm schuiven als je een toets aan raakt, welke dan ook, behalve N, de spatie-balk of STOP.

LIST kan ook worden gevolgd door een line nummer, in de vorm van een numerieke value, bijvoorbeeld:

LIST 1ØØ

De value die op LIST volgt wordt dan afgerond naar het dichtst bij zijnde gehele getal, en de listing gaat dan vanaf deze line verder. Als er geen line is met dit nummer gaat de listing bij de volgende line verder.

Format
LIST [int-num-expr]

LLIST Line printer, LIST

Plaats op het toetsenbord
Extend modus
Symbool V

Statement/opdracht

LLIST zorgt er voor dat Sinclair printers een afdruk-listing produceren van het lopende programma.

Hoe gebruik je LLIST?
LLIST wordt op dezelfde wijze gebruikt als LIST (zie LIST voor meer details). Merk op dat de afbeelding op het scherm niet verandert als de Listing wordt afgedrukt.

Format
LLIST [int-num-expr]

LN (Natuurlijke) logaritme

Plaats op het toetsenbord
Extend modus
Symbool Z

Functie

LN geeft de natuurlijke logaritme (de logaritme met e als basis) van een value. Het werkt als de inverse (omgekeerde) van EXP.

Hoe gebruik je LN?
LN wordt gevolgd door een numerieke value, bijvoorbeeld:

```
6Ø LET x = LN y
```

Een uitdrukking moet tussen haakjes worden geplaatst. De value volgend op LN moet groter zijn dan Ø. LN geeft dan de natuurlijke logaritme van deze value.

Format
LN num-const
LN num-var
LN (num-expr)

LOAD

Plaats op het toetsenbord
Symbool J

Statement/opdracht

LOAD laad een volledig programma in het geheugen van een band.

Hoe gebruik je LOAD?
LOAD wordt gewoonlijk gebruikt als een directe opdracht, maar het kan ook een statement in het programma vormen, zodat een nieuw programma geladen kan worden. LOAD wordt gevolgd door een archief naam. Dit is een rij value met een lengte tot 1Ø characters, bijvoorbeeld:

```
LOAD "filename"
```

Bij de uitvoering van het lopende programma, (uit het geheugen) en alle values van zijn variabelen zijn verwijdert. De Spectrum zoekt dan naar het genoemde programma en laad het waar het hoort. Merk op dat de computer bij de programma namen onderscheidt maakt tussen hoofd-letters en gewone letters.

Als een nul-rij ("") LOAD volgt, zoals in deze opdracht:

```
LOAD " "
```

dan laad de Spectrum het eerste volledige programma dat het vindt.

Merk op dat LOAD anders wordt gebruikt als er een Microdrive is aangesloten (Zie het Microdrive en Interface 1 handboek).

Format
LOAD string-expr

LOAD CODE

Plaats op het toetsenbord J
Extend modus
Symbool I

Statement/opdracht

LOAD CODE wordt gebruikt om een gedeelte van het geheugen met informatie te laden, van af een band. De informatie bestaat uit een set bytes en deze worden naar een set addresses in het geheugen gestuurd. LOAD CODE kan worden gebruikt om een afbeelding te laden of informatie in het geheugen te laden over door de gebruiker bepaalde graphics, bijvoorbeeld:

Hoe gebruik je LOAD CODE?
LOAD CODE kan worden gebruikt als een directe opdracht. Het kan ook een statement in een programma vormen. LOAD wordt gevolgd door de archiefnaam, dit is een rij-value, en dan pas CODE, bijvoorbeeld:

```
LOAD "data" CODE
```

De archiefnaam die op LOAD volgt is de naam van de informatie die moet worden geladen en is onderhevig aan dezelfde beperkingen als programma namen (zie LOAD). LOAD CODE zoekt vervolgens naar de genoemde informatie en als deze is gevonden, beeld zij af: Bytes: gevolgd door de naam. De Spectrum laad dan de bytes in het geheugen op de addresses van waar zij werden gehaald. Iedere bestaande informatie wordt onder geschreven.

CODE kan ook worden gevolgd door een of twee numerieke values gescheiden door een komma, bijvoorbeeld:

```
LOAD "picture" CODE
16384, 6912
```

De values volgend op CODE worden afgerond naar het dichtst bij zijnde gehele getal en bepalen dan het start address (16384 boven) waar de genoemde informatie geladen moet worden, en het aantal bytes (6912 boven) dat naar locaties gestuurd moet worden, beginnend bij 16384. Als het nummer verkeerd is wordt er een tape loading error report gegeven. Als slechts een value volgt op CODE, bepaald het start address van waar uit alle bytes gevonden moeten worden.

Het bovengenoemde voorbeeld kan ook worden uitgvoerd door de keywords LOAD SCREEN $.

Voor details m.b.t. het opslaan van bytes, zie SAVE CODE.

Format
LOAD string-expr CODE
[int-num-expr] [int-num-expr]

LOAD DATA

Plaats op het toetsenbord
J
Symbool D

Statement/opdracht

LOAD DATA wordt gebruikt om arrays te laden van de band. De arrays zijn opgenomen m.b.v. SAVE DATA.

Hoe gebruik je LOAD DATA?
LOAD DATA kunnen worden gebruikt om een statement in een programma te vormen, of als een directe opdracht. LOAD wordt eerst gevolgd door een archiefnaam, dit is een rij-value. Hier op volgt DATA en een letter of een letter en $, en tot slot een paar lege haakjes, bijvoorbeeld:

```
27Ø LOAD "numbers" DATA n()
3ØØ LOAD "names" DATA n$()
```

De archiefnaam volgend op LOAD is de naam die wordt gegeven aan de array op de band. Deze naam is onderhevig aan dezelfde beperkingen als programma namen gebruikt met LOAD. De letter of letter $ volgend op DATA is de naam die het array krijgt in het programma als het is geladen en gebruikt.

Bij uitvoering zoekt de Spectrum naar het genoerde array op de band. Als het is gevonden verschijnt het bericht: Number array of Character array; gevolgd door de naam, en het array wordt geladen. Ieder array dat op dat moment in het geheugen is, met dezelfde letter-naam (n of n $ boven) wordt verwijdert en een nieuw array met deze letter en de opgeslagen values op de band, wordt gevormd. Merk op dat by character arrays, iedere rij-variabele die op dat moment in het geheugen is, met dezelfde letter wordt ook verwijdert.

Format
LOAD string-expr **DATA**
letter [$] ()

LOAD SCREEN$

Plaats op het toetsenbord
J
Symbool K

Statement/opdracht

LOAD SCREEN$ maakt het mogelijk een afbeelding direct van de band op het beeldscherm te brengen. Het zend informatie van de band naar dat gedeelte van het geheugen dat afbeelding op het scherm controleerd, zodat een beeld geproduceerd kan worden.

Hoe gebruik je LOAD SCREEN $
LOAD SCREEN $ kan worden gebruikt om een statement te vormen in een programma of als een directe opdracht. LOAD wordt gevolgd door een archief naam (dit is een rij value) en dan SCREEN$, bijvoorbeeld:

LOAD "Picture" **SCREEN$**

De archiefnaam volgend op LOAD is dezelfde als de naam gegeven aan de beeld informatie op de band. Deze naam is onderhevig aan dezelfde restricties als programma namen, gebruikt met LOAD. De Spectrum zoekt vervolgens naar de genoemde informatie en als deze gevonden is wordt deze eerst in het afbeeld-archief en dan in de attributen afdeling van het geheugen opgeslagen. Het beeld wordt geleidelijk opgebouwd in de betrokken inkt en papier kleuren. Hierna worden de attributen (echte kleuren ed.) toegevoegd.

(Voor meer details mbt. het opslaan van beeldscherm-informatie, zie SAVE SCREEN$).

Format
LOAD string-expr **SCREEN$**

LPRINT Line printer PRINT

Plaats op het toetsenbord
Extend modus
Symbool C

Statement/opdracht

LPRINT zorgt er voor dat printers van het Sinclair-type een item van data afdrukken, op dezelfde wijze als PRINT het item op het beeldscherm laat komen.

Hoe gebruik je LPRINT?
LPRINT kan een statement in een programma vormen of worden gebruikt als een directe opdracht. Het wordt gevolgd door items van data die kunnen worden gescheiden door een puntkomma (;), komma's of apostrophes, bijvoorbeeld:

```
6Ø LPRINT "Number"; x
    "Name";n$ "Age";a
```

Als de items output zijn voor de printer, worden zij op dezelfde grootte afgedrukt, als PRINT hen afbeeld op het scherm. Een LPRINT statement of opdracht kan ook TAB statements, bepaalde CHR$ controles, INVERSE en OVER statements en control codes met hetzelfde gevolg als PRINT, bevatten. Een AT statement kan ook in begrepen zijn, maar het line-nummer wordt terzijde gelaten en het item van data wordt gedrukt op de gegeven kolom positie en dezelfde line.

Format
LPRINT [**TAB** (int-num-expr;)]
[**AT** int-num-expr, int-num-expr;]
[**CHR$** (int-num-expr):]
[statement;] [num-expr]
[string-expr] [;] [,] ['']

MERGE

Plaats op het toetsenbord
Extend modus
Symbool T

Statement/opdracht

MERGE maakt het mogelijk twee programma's te versmelten, tot een.

Hoe gebruik je MERGE?
MERGE kan worden gebruikt om een statement in een programma te vormen of als een directe opdracht. Het wordt gevolgd door een archief naam in de vorm van een rij-value, bijvoorbeeld:

5ØØ MERG "prog2"

De archief naam volgend op MERGE is de naam van het programma dat moet worden versmolten met het programma dat nu in het geheugen zit. Deze naam is onderhevig aan dezelfde beperkingen als programma – namen gebruikt met LOAD. MERGE laad dan het nieuwe programma in het geheugen zonder het eerste te verwijderen. Echter; het nieuwe programme schrijft over die lines in het bestaande programma met dezelfde line nummers als de lines in het nieuwe programma hebben, en variabelen met dezelfde naam worden ook onder geschreven.

Format
MERGE string-expr

MOVE

Microdrive file-handling opdracht. Zie het Microdrive en Interface 1 handboek.

NEW

Plaats op het toetsenbord
Symbool A

Statement/opdracht

NEW maakt het BASIC geheugen gebied leeg (dit is het gebied tot aan RAMTOP) Hierbij wordt het programma dat op dat moment in dat geheugen gedeelte is, verwijdert.

Hoe gebruik je NEW?
NEW wordt gewoonlijk gebruikt als een directe opdracht, maar het kan ook een statement in een programma vormen. Het wordt alleen gebruikt. Bij uitvoering worden programma en variabelen verwijdert. Het geheugen wordt uitgewist tot aan RAMTOP zodat de door de gebruiker bepaalde graphics characters niet worden aangetast, om dat deze boven RAMTOP zijn opgeslagen.

Format
NEW

NEXT

Plaats op het toetsenbord
Symbool N

Statement/opdracht

NEXT wordt altijd gebruikt samen met for om zo een FOR NEXT loop te vormen.

Hoe gebruik je NEXT?
NEXT wordt gewoonlijk gebruikt om een statement te vormen in een programma om zo een FOR NEXT loop de vormen. Het wordt gevolgd door een letter, dat is de controle variabele in de loop, bijvoorbeeld:

9Ø NEXT a

In Sinclair BASIC moet er een controle – variabele zijn.

(voor meer details m.b.t. de FOR NEXT loops zie FOR).

Format
NEXT letter

NOT

Plaats op het toetsenbord
Symbool S

Logische werking/functie

NOT wordt gebruikt om de juistheid van een voorwaarde om te keren. Zodoende wordt een onjuiste voorwaarde juist en omgekeerd een juiste voorwaarde onjuist.

Hoe gebruik je NOT?
NOT wordt gevolgd door een voorwaarde of door een numerieke value, bijvoorbeeld:

```
9Ø IF NOT x = y + z THEN
    PRINT "Wrong"
```

```
9Ø LET correct = x = y + z IF
   NOT correct THEN PRINT
   "wrong"
```

Wanneer NOT wordt gevolgd door een voorwaarde (x = y + z– boven). dan zal de Spectrum eerst een value van 1 toekennen aan de voorwaarde, als deze juist is, en ø als deze voorwaarde onjuist is. NOT zal daarna als een functie optreden en de geproduceerde values omdraaien, zodat het tegengestelde van de voorwaarde getest kan worden. Bedenk dat een voorwaarde tusson haakjes geplaatst moet worden als hij AND of OR bevat.

Als NOT wordt gevolgd door een numerieke-value, geeft het een Ø berug, als de volgende value niet nul is. Als NOT wordt gevolgd door een numerieke-value, geeft het een 1 terug, als de volgende value Ø is. Zodoende zal in de bovengenoemde voorbeelden de Spectrum "wrong" afdrukken als: x <> y + z, of als "correct" een value Ø heeft.

Format
NOT cond
NOT num. expr.

OPEN#

Microdrive file-handling opdracht. Zie Microdrive en Interface 1 handboek

OR

Plaats op het toetsenbord
Symbool U

logische operator/functie

OR treedt op als een logische operator, om de juistheid van een combinatie van voorwaarden te testen. Als een of meer van de voorwaarden juist is, dan is het geheel juist. OR treedt ook op als een functie om binaire operaties te verrichten op twee numerieke values.

Hoe gebruik je OR?
Als logische operator verbindt OR twee voorwaarden on een statement waar van de juistheid van het geheel getest moet worden, bijvoorbeeld:

```
7Ø IF INKEY$ = "N" OR
   INKEY$ = "n" THEN STOP
```

Als een van beide of beide voorwaarden juist zijn dan is het geheel juist. In de combinatie (hierboven) een van de voorwaarden (INKEY$ = "N" in INKEY$ = "n") wordt juist zodra ie de N toets indrukt. Het doet er dan niet toe of de CAPS SHIFT of de CAPS LOCK werkt of niet. De gehele combinatie is dan waar en het programma stopt.

OR als een functie.
De ZX Spectrum+ wijst een numerieke value van 1 toe aan een juiste voorwaarde en Ø aan een onjuiste voorwaarde. De Spectrum herkent iedere niet-nul value als juist en de Ø value als onjuist. OR kan daarom voor af gegaan worden door een numerieke value, of gevolgd worden door een numerieke value, bijvoorbeeld:

```
4Ø LET x = y OR z
```

De variabele x wordt een value 1 toegewezen als z is niet-nul of wel juist. De variabele x wordt een value y toegewezen als z is nul ofwel een onjuiste voorwaarde.

Merk op dat de Spectrum de combinaties van numerieke values niet evalueert in overeenkomst met de standaard-waarheids-tabellen.

Format
cond **OR** cond
num-expr **OR** num-expr

OUT

Plaats op het toetsenbord
Extend modus
Symbool O

Statement/opdracht

OUT stuurt een byte naar een gegeven input/output port address om zo een output instrument te sturen.

Hoe gebruik je OUT?
OUT ken gebruikt worden om een statement in een programma te vormen, of als een directe opdracht. OUT wordt gevolgd door twee numerieke values, welke worden gescheiden door een komma, bijvoorbeeld:

```
4Ø OUT 254,3
```

Beide values worden afgerond naar het dichtst bij zijnde gehele getal. De eerste value (254 boven) kan dan variëren van Ø tot 65535 en is het port-address. De tweede value (3 boven) kan variëren van Ø tot 255 en is de byte die naar zijn address gezonden wordt.

Bits Ø tot 2 van de byte output naar port-address 254 bepalen de grens kleur. Het boven gegeven voorbeeld levert een paarse grenskleur op. Bit 3 op dit address stuurt de MIC socket in bib 4 de luidspreker. Port address 251 stuurt de printer en ports 254, 247 en 239 worden gebruikt met andere rand-apparaten.

Format
OUT int-num-expr, int-num-expr

OVER

Plaats op het toetsenbord
Extend modus
Symbool N

Statement/opdracht

OVER wordt gebruikt om het ene character over het andere te drukken. Het kan ook worden gebruikt om punten te stippelen of lijnen en krommen te tekenen in een papier kleur in plaats van een inkt kleur.

Hoe gebruik je OVER?
OVER wordt gewoonlijk gebruikt om een statement in een programma te vormen. Het wordt gevolgd door een numerieke value, bijvoorbeeld:

```
8Ø OVER 1
```

De value volgend op OVER wordt algerond naar het dichtst bij zijnde gehele getal en kan Ø of 1 zijn. OVER Ø, welke is de defecte toestand, veroorzaakt dat ieder character het character dat eerder op die positie werd geplaatst, verwijdert en zijn plaats in neemt. OVER 1 zorgt er voor dat iedene twee characters, afgebeeld op dezelfde plaats, gecombineerd worden.

OVER kan worden ingebouwd in een PRINT of een INPUT statement op dezelfde manier als INK zod at het alleen die characters beinvlood die worden afgebeeld door de statement. Dit statement bijvoorbeeld onderstreept een woord:

```
6Ø PRINT AT 11, 15; "YES";
   OVER 1; AT 11, 15; "   "
```

Echter, merk op dat characters zijn gecombineerd zodat de papier kleur is gegeven, waar de inkt kleuren elkaar overlappen.

OVER in high-resolution
OVER kan worden gebruikt met PLOT, DRAW en CIRCLE. Zonder OVER, kunnen lijnen in krommen elkaar overlappen, maar zij moeten dezelfde inkt kleur hebben anders verandert de inkt kleur in de hele character positie, daar waar zij kruisen. Als OVER, wordt gebruikt produceren lijnen en krommen de papier kleur, daar waar zij overlappen of characters

tegenkomen. Punten stippelen of lijnen trekken of krommen op precies dezelfde plaats waar zij al getekend waren, laat hen bij gebruik van OVER 1, verdwijnen.

Format
OVER int-num-expr

PAPER

Plaats op het toetsenbord
Extend modus
Symbool C

Statement/opdracht

PAPER wordt gebruikt om de papier of achtergrond kleur te bepalen, die wordt gebruikt op het beeldscherm. Dit kan ofwel de kleur van de achtergrond over het gehele afbeeldings gebied zijn, ofwel de kleur achter de individuele characters, punten of lijnen, die verschijnen in enkele character posities.

Hoe gebruik je PAPER?
PAPER kan worden gebruikt om een statement in een programma te vormen, of als een directe opdracht. Het wordt gevolgd door een numerieke value, bijvoorbeeld:

8Ø PAPER x

De value volgend op PAPER wordt afgerond naar het dichtst bij zijnde gehele getal. Deze value kan varieren van Ø tot 9. De dan gegeven papier kleuren zijn dezelfde als die gegeven bij INK (zie INK). Paper kleuren kunnen algemeen of Lokaal (plaatselijk) zijn. Papier kleuren zijn plaatselijk als PAPER wordt ingebouwd in statements op precies dezelfde wijze als de inkt kleuren (zie INK). Als characters worden gedrukt volgens een PAPER statement, algemeen of plaatselijk, zal de achtergrond achter de gehele bedoelde character positie naar de bepaalde kleur toe veranderen. Dit gebeurt ook wanneer er punten worden gestippeld of lijnen en cirkels getekend, met een ingebouwde statement. Det gebeurt niet als gevolg van een algemene opdracht of een algemene statement ((dwz met betrekking tot de gehele afbeeldingsruimte)). Om een gekleurde achtergrond te produceren over het gehele afbeeldingsgebied, is het nodig CLS te gebruiken na een PAPER statement. De gehele afbeelding wordt in die kleur omgezet, en deze kleur blijft dan de achtergrond kleur over het gehele gebied.

Format
PAPER int-num-expr [;]

PAUSE

Plaats op het toetsenbord
Symbool M

Statement/opdracht

PAUSE kan worden gebruikt om een programma op te houden voor een bepaalde of on bepaalde tijd.

Hoe gebruik je PAUSE?
PAUSE wordt gewoonlijk gebruikt om een statement te vormen in een programma. Het wordt gevolgd door een numerieke value, bijvoorbeeld:

13Ø PAUSE 1ØØ

De value volgend op PAUSE wordt afgerond naar het dichtst bij zijnde gehele getal. Deze value kan variëren van Ø tot 65535. Het bepaald het oponthoud dat optreedt als dit nummer het televisie beeld omlÿst, zodat een value van 5Ø een pause veroorzaakt van 1 seconde in UK en Europa waar de frequentie van het elektrisch net 5Ø Hz is.

Merk op dat de pauze altijd afgebroken kan worden door willekeurig een toets in te drukken. Verder levert PAUSE Ø een on begrensde pauze op welke wordt afgebroken als je een toets in drukt.

Format
PAUSE int-num-expr

PEEK

Plaats op het toetsenbord
Extend modus
Symbool O

Functie

PEEK geeft de value van de byte die is opgeslagen op een speciaal address in het geheugen.

Hoe gebruik je PEEK?
PEEK wordt gevolgd door een numerieke value, bijvoorbeeld:

8Ø LET x=**PEEK**(256*y)

Merk op dat de uitdrukking tussen haakjes moet zijn geplaatst. De value volgend op PEEK wordt afgerond naar het dichtst bij zijnde gehele getal. De value kan dan varieren van Ø tot 65535 en geeft een address in het geheugen. PEEK geeft dan de value van de byte (en getal tussen Ø en 255) op het bepaalde address.

Bijvoorbeeld:
Het aantal frames van de televisie beelden, dat heeft plaats gevonden sinds de Spectrum word aangezet, is opgeslagen op de addresses 23672 tot 23674. Als de frames in een constant tempo worden geproduceerd, dan kan m.b.v. PEEK de tijd worden gemeten. De hieronder gegeven line laat de tijd in seconden zien (sinds de Spectrum werd aangezet), vermindert met de tijd gebruikt voor het produceren van geluid en het besturen van rand-apparatuur zoals de cassette speler en de printer).

1Ø PRINT (PEEK
23672+256***PEEK**
23673+65536***PEEK**
23674)/5Ø

Merk op dat 5Ø (hierboven) de frequentie van het elektrisch is. Als deze 6Ø Hz zou zign moet je 5Ø door te vervangen.

Format
PEEK int-num-const
PEEK int-num-var
PEEK (int-num-expr)

PI

Plaats op het toetsenbord
Extend modus
Symbool M

Functie

PI geeft de value van pi (π) voor gebruik in berekeningen.

Hoe gebruik je PI?
PI vereist geen values of variabelen als het wordt gebruikt in een statement of in een opdracht, bijvoorbeeld:

DRAW 225, Ø, –**PI**

PI levert een value van 3.1415927 zodat de opdracht hierboven een grote halve cirkel op het beeldscherm.

Format
PI

PLOT

Plaats op het toetsenbord
Symbool Q

Statement/opdracht

PLOT wordt gebruikt bij high-resolution-graphics om een pixel of punt, in kleur, op een bepaalde plaats op het scherm te stippelen.

Hoe gebruik je PLOT?
PLOT wordt gebruikt om een statement in een programma te vormen, of als een opdracht. Het wordt gewoonlijk gevolgd door

twee numerieke values, bijvoorbeeld:

```
5Ø PLOT 128, 87
```

Beide values volgend op PLOT worden afgerond naar het dichtst bij zijnde gehele getal. De eerste value kan varieren van Ø tot 255 en bepaald de horizontale coördinaat van de positie op het scherm. De tweede value kan variëren van Ø tot 175 en bepaald de vertikale coördinaat. Een pixel wordt gewoonlijk gestippeld in de op dat moment gebruikte inkt kleur, en op de bepaalde plaats.

Merk de volgende gevolgen van kleur-statements of opdrachten op PLOT op: Na OVER 1 zal een bestaandestip op dezelfde plaats worden verandert in de papierkleur. Na INVERSE 1, wordt de stip aangebracht in de papier kleur. Na BRIGHT 1 of FLASH 1, zal de gehele character positie op het low-resolution-scherm, waarin de pixel is gestipt, helder zijn en knipperen.

Deze 4 keywords en INK kunnen ook worden ingebouwd in een PLOT statement, op dezelfde wijze als met PRINT, bijvoorbeeld:

```
16Ø PLOT INK 2;x,y
```

Hun effect is hetzelfde, maar plaatselijk en gelimiteerd tot de door de statement bepaalde pixel. Als PAPER is ingebouwd in een PLOT statement, verandert de papierkleur van de gehele character positie rond de pixel, naar de gegeven kleur.

Format
PLOT [statement] int-num-expr, int-num-expr

POINT

Plaats op het toetsenbord
Extend modus
Symbool 8

Functie

POINT wordt gebruikt om uit te vinden of een kleur op een bepaalde plaats op het high-resolution-scherm de inkt kleur of de papierkleur is.

Hoe gebruik je POINT?
POINT wordt gevolgd door twee numerieke waarden, gescheiden door een komma en tussenhaakjes geplaatst, bijvoorbeeld:

```
24Ø IF POINT (x,y) = 1 THEN
     GOSUB 6ØØ
```

De twee values volgend op point worden naar gehele getallen afgerond. De eerste value kan varieren van Ø tot 255 en bepaald de horizontale coordinaat van een pixel op het scherm. De tweede value kan varieren van Ø tot 175 en bepaald de verticale coordinaat POINT levert 1 op als de pixel op de bepaalde plaats inkt kleur heeft en Ø als het papierkleur is.

Format
POINT (int-num-expr, int-num-expr)

POKE

Plaats op het toetsenbord
Symbool O

Statement/opdracht

POKE wordt gebruikt om de value van de byte op een bepaald address in het geheugen, te veranderen.

Hoe gebruik je POKE?
POKE wordt gebruikt om een statement in een programma te vormen, of als een opdracht. Het wordt gevolgd door twee numerieke values, gescheiden door een komma, bijvoorbeeld:

```
POKE 236Ø9, 255
```

De beide values volgend op POKE worden afgerond naar het dichtst bij zijnde gehele getal. De eerste value kan varieren van 16384 tot 65535 en in een address in RAM. De tweede value kan varieren van Ø tot 255 en bepaald de byte die geschreven moet worden naar het bepaalde address.

Format
POKE int-num-expr, int-num-expr

PRINT

Plaats op het toetsenbord
Symbool P

Statement/opdracht

PRINT beeld data af op het scherm. Ieder enkel character of serve characters kan data vormen. Een PRINT statement kan andere keywords bevatten om 2Ø de positie en de kleur van de data te bepalen.

Hoe gebruik je PRINT?
PRINT kan alleen worden gebruikt, of het kan worden gevolgd door data. Deze data kan de vorm van elke numerieke uitdrukking of rij-uitdrukking hebben en zelfs kan het een mengeling van beiden zijn.

Als je PRINT met data gebruikt moeten twee of meer gescheiden items gescheiden worden gescheiden items gescheiden worden door een punt-komma, komma of apostrophe.

Bepaalde andere keywords kunnen worden ingebracht, in iedere volgorde, tussen PRINT en de data. Dit onder de voorwaazrde dat elke statement, gevormd door een keyword, eindigt met punt-komma. Deze keywords zijn: CHR$, TABN, AT, INK, PAPER, FLASH, BRIGHT, INVERSE en OVER.

PRINT met rijen.
PRINT alleen of gevolgd door een nul-rij ("") beeld een lege line of en beweegt de positie-wijzer naar het begin van de volgende line.

PRINT gevolgd door een rij-constante (ieder character tussen dubbele aanhalings tekens) beeld de characters of zoals zij verschijnen tussen de aanhalings tekens. De opdracht:

```
PRINT "3/542/76/21"
```

beeld af: 3/542/76/21

PRINT gevolgd door een rij variabele of uitdrukking beeld de rij of rijen welke zij voorstellen, af.

PRINT met getallen
PRINT gevolgd door een numerieke uitdrukking beeld de value van die uitdrukking af. Getallen worden in decimale notatie afgebeeld met tot 8 digits een geen rij nullen achter het decimale punt.

Erg grote en erg kleine getallen worden afgebeeld in een kortere wetenschappetijke notatie, d.w.z. als twee cyfers gescheiden door de letter E. Dit

geeft een getal aan waarin het eerste gedeelte (de mantisse) wordt verheven tot de macht van het tweede gedeelte (de exponent). De opdracht:

PRINT 3/542/76/21

geeft als beeld:

3.468Ø798E-6

PRINT geformeerd met punt tekens
PRINT gevolgd door data items gescheiden door puntkomma, beeld de items naast elkaar af zonder tussenruimte.
De opdracht:

PRINT 1;2;3

beeld af

123

PRINT gevolgd door data items gescheiden door een komma beeld ieder item af aan het begin of in het midden van een line, afhanketijk van de positie van het eerste item. De opdracht:

PRINT 1,2,3

beeld af:

1 2
3

PRINT gevolgd door data items gescheiden door een apastrophe beeld het item na de apastrophe af aan het begin van de volgende line. De opdracht:

PRINT 1'2'3

beeld af:

1
2
3

Als een PRINT statement of opdracht eindigt met puntkomma, komma af apostrophe, dan wordt het item dat door de volgende PRINT wordt afbeeld, op dezelfde wijze behandeld.

PRINT en andere keywords.
PRINT kan ook worden gevolgd door TAB, een numerieke value, een puntkomma en dan een data item, bijvoorbeeld:

6Ø PRINT TAB x; a$

De value volgend op TAB (x boven) wordt algerond naar het dichtst bij zijnde gehele getal en wordt dan gedeeld door 32. Het resultaat wordt geretourneerd om een value tussen Ø en 31 tegeven. Dan wordt de data item afgebeeld op dezelfde kolom positie op dezelfde of de volgende line.

PRINT kan worden gevolgd door AT en dan twee numerieke values gescheiden door een komma, een puntkomma en een data item, bijvoorbeeld:

5Ø PRINT AT 1,c; "Data"

De eerste value (1 boven) dan varieren van Ø tot 21 en bepaald het nummer van de line of de rij waarin de data worden afgebeeld. De tweede value (c boven) kan varieren van Ø tot 31 en bepaald het nummer van de kolom waarin het eerste character (of digit of de data) zullen worden afgebeeld. Values welke geen gehele getallen zijn worden geacceptoerd en afgerond naar het dichtst bij zijnde gehele getal. De opdracht: **PRINT AT** 11, 16; "*" beeld een ster af in het midden van het scherm.

PRINT kan ook worden gevolgd door een of meer CHR$ functies (zie CHR$ voor details).

PRINT en kleur keywords
De afbeelding geproduceerd door PRINT wordt beinvloed door kleur statements en kleur opdrachten gegeven door: INK, PAPER, FLASH, BRIGHT, INVERSE en OVER. PRINT kan ook worden gevolgd door een of meer van deze statements (ieder gevolgd door een puntkomma) voor de data items, bijvoorbeeld:

5Ø PRINT AT 11,16; INK 2;
FLASH 1; "*"

De data item wordt dan afgebeeld met de bijdragen gespecificeerd door de kleur keywords. Deze bijdragen zijn plaatselijk en zijn alleen van toepassing op het afgebeelde item. Na de uitvoering van de PRINT statement, gaan de keywords terug naar algemene values. PRINT zal ook gehoor geven aan plaatsetijke kleur-controle-codes, in gebracht met de data (zie blz. 33).

Format
PRINT [**TAB** int-num-expr;]
[**AT** int-num-expr, int-num-expr;]
[**CHR$** (int-num-expr);]
[statement;] [num-expr]
[string-expr] [;] [,] ['']

RANDOMIZE

Plaats op het toetsenbord
Symbool T

Statement/opdracht

RANDOMIZE verschijnt op het toetsenbord als RAND. Het wordt samen met RND gebruikt om volgorden van getallen te produceren die ofwel willekeurig ofwel voorspelbaar zijn.

Hoe gebruik je RANDOMIZE?
RANDOMIZE wordt gebruikt om een statement te vormen in een programma of als een opdracht. Het wordt gevolgd door een numerieke value, bijvoorbeeld:

RANDOMIZE 1
1Ø RANDOMIZE

De value volgend op RANDOMIZE wordt afgerond naar het dichtst bij zijnde gehele getal en kan variëren van Ø tot 65535. Een value groter dan Ø voegt de systeem-variabele SEED toe aan deze value. Als hier RND op volgt levert het altijd dezelfde volgorde van getallen op. (zie blz 48 voor informatie m.b.t. systeem variabelen). De eigenlijke volgorde hangt of van de value van RANDOMIZE.

Als RANDOMIZE wordt gevolgd door Ø of geen value, dan wordt aan SEED de value van de systeem-variabele FRAMES gegeven. Deze telt de frames afgebeeld van af het moment dat de Spectrum word aan gezet. Omd at SEED 5Ø (of 6Ø afhankelijk van de frequentie van het elektrisch net) keer per seconde verandert, zal de volgorde van de getallen, opgewekt door RND, volgend op RANDOMIZE of RANDOMIZE Ø uiterst willekeurig zijn.

Format
RANDOMIZE [int-num-expr]

READ

Plaats op het toetsenbord
Extend modus
Symbool A

Statement/opdracht

READ wordt gebruikt samen met DATA om values toe te wijzen aan variabelen gebruik makend van de values in een DATA statement.

Hoe gebruik je READ?
READ wordt gewoonlyk gebruikt om een statement in een programma te vormen. Het wordt

gevolgd door een of meer numerieke variabelen of rij-variabelen, gescheiden door een, komma, bijvoorbeeld:

2Ø READ a$, x

Als READ voor het eerst wordt uitgevoerd, neemt het evenveel values als er variabelen zijn van af de start van de eerste DATA tijst en wijst de values toe aan de variabelen in volgorde. Als READ weer wordt uitgevoerd, wordt de volgende set DATA values toe gewezen aan de variabelen die genoemd zijn in de READ-statement, enz.

(Voor meer details, zie DATA).

Format
READ num-var [,num-var] [,string-var]
READ string-var [num-var] [,string-var]

REM REMark

Plaats op het toetsenbord
Symbool E

Statement

REM wordt gebruikt om opmerkingen of herkennings punten in een programma te stoppen. Dit kan de titel en de auteur zijn van het programma, of verklaringen van lines in het programma zoals het nut van een variabele. De opmerkingen spelen geen rol in het aflopen van het programma en kunnen worden opgevat als, listing.

Hoe gebruik je REM?
REM vormt ofwel een eigen line in een programma of het vormt de laatste statement op een line. Het wordt gevolgd door iedere opmerking die je in kunt toetsen, zoals jij wil, bijvoorbeeld:

8Ø INPUT n$: REM n$ is name

Als de computer REM tegenkomt, negeert hij alles wat na REM komt op die line.

Format
REM ieder character

RESTORE

Plaats op het toetsenbord
Extend modus
Symbool S

Statement/opdracht

RESTORE wordt gebruikt samen met READ en DATA om ervoor te zorgen dat READ values neemt van een bepaalde DATA statement, in plaats van de eerste of de volgende DATA-statement in het programma.

Hoe gebruik je RESTORE?
Restore vormt gewoonlijk een statement in een programma. Het wordt gevolgd door een numerieke value, bijvoorbeeld:

16Ø RESTORE 8ØØ

De value volgend op RESTORE wordt afgerond naar het dichtst bij zijnde gehele getal. Het is dan het nummer van een line in het programma die een DATA statement bevat. Volgend op RESTORE zal de volgende READ statement de values toe wijzen welke behoren tot deze DATA statement. Als het nummer van de line niet bestaat of geen DATA statement bevat, dan zal READ doorgaan naar de eerst volgende DATA-statement na deze line.

Als RESTORE wordt gevolgd door Ø of geen value, dan zal de volgende read-statement naar de eerste DATA statement in het programma gaan.

Format
RESTORE [int-num-expr]

RETURN

Plaats op het toetsenbord
Symbool Y

Statement/opdracht

RETURN wordt gewoonlijk gebruikt om een statement in een programma te vormen. Het wordt alleen gebruikt aan het einde van een subroutine, bijvoorbeeld:

1ØØØ RETURN

Bij uitvoering vertakt het programma naar de statement volgend op de laatste GOSUB statement die werd uitgevoerd.

(Zie GOSUB poor meer details).

Format
RETURN

RND RaNDom number

Plaats op het toetsenbord
Extend modus
Symbool T

Functie

RND wordt gebruikt om een willekeurig getal optewekken.

Hoe gebruik je RND?
RND wordt alleen gebruikt in een statement of in een opdracht, bijvoorbeeld:

6Ø LET x = RND

RND retourneerd een willekeurig getal dat kleiner is dan 1 en groter of gelijk aan Ø.

Als de Spectrum naar reset of NEW is geschakeld, worden getallen achter een volgens geretourneerd door RND in dezelfde volgorde. De volgorde wordt op gewekt door de machten van 75 te nemen (75, 75*75, 75*75*75, . . .) en ieder macht te delen door 65537 en dan alleen het resultaat te gebruiken, hierna 1 van het resultaat af te trekken en het resultaat hiervan weer door 65536 te delen.

Als een nog willekeuriger of een andere bepaalde volgorde vereist is dan moet je RANDOMIZE voor RND gebruiken.

Random gehele getallen
((willekeurig nemen van . . .))
Ieder geheel getal van 1 tot x wordt gegeven door INT (RND*x) + 1. Om nu een willekeurig geheel getal tussen Ø en x op te wekken, moet je INT/RND*x + Ø.5 gebruiken.

Format
RND

RUN

Plaats op het toetsenbord
Symbool R

Statement/opdracht

RUN zorgt er voor dat een programma gaat lopen, gewoonlijk vanaf de eerste line.

Hoe gebruik je RUN?
RUN kan worden gebruikt als een directe opdracht, maar ook om een statement te vormen in een programma. Het wordt gevolgd door een numerieke value, bijvoorbeeld:

RUN 5Ø

Als er geen value volgt op RUN, dan loopt het programma vanaf de eerste line. Als er een value bij is betrokken, wordt deze afgerond naar het dichtst bij

zijnde gehele getal. Het programma loopt dan verder vanaf de line met dat nummer. Als die line niet bestaat, loopt het programma vanaf de eerst volgende line.

Als een programma, wordt opgeslagen met gebruik van LINE, dan loopt het automatische als het laad en is RUN niet nodig.

Format
RUN int-num-expr

SAVE

Plaats op het toetsenbord
Symbool S

Statement/opdracht

SAVE stuurt een programma naar de cassette-speler zodat het op band gezet kan worden.

Hoe gebruik je SAVE?
SAVE wordt gewoonlijk gebruikt als een directe opdracht maar het han ook een statement vormen in een programma. Het wordt gevolgd door een archief naam. Dit is dan een rij-value, bijvoorbeeld:

SAVE "archief naam"

De archief naam kan tot 1Ø characters bevatten. Bij uitvoering wordt het volgende afgebeeld:

Start tape, then press any key

((start band, dan druk een toets in))

Na het indrukken van, willekeurig welke, toets, wordt het programma naar de cassette recorder gestuurd.

Als het hele programma op de band staat verschijnt tot slot **Ø OK, Ø, 1**.

Automatic running
Als het opgeslagen programma bij het laden automatisch moet lopen dan moet SAVE worden gebruikt samen met LINE. De programma naam wordt dan gevolgd door LINE en een numerieke value, bijvoorbeeld:

SAVE "archief naam" **LINE 1**

De value volgend op LINE wordt afgerond naar het dichtst bij zijnde gehele getal en moet dan 1 zijn of het nummer van de line in het programma. Het programma wordt dan naar de band gezonden, net als bij SAVE. Bij het laden loopt het programma automatisch vanaf de line met het bepaalde nummer of, als die line niet bestaat, vanaf de volgend line in het programma.

Format
SAVE string-expr [**LINE** int-num-expr]

SAVE CODE

Plaats op het toetsenbord
Symbool S
Extend modus I

Statement/opdracht

SAVE CODE zendt een sectie van informatie van het geheugen naar de cassett recorder, om het op band op te slaan. De informatie kan dan weer worden terug geplaatst in het geheugen door LOAD CODE te gebruiken.

Hoe gebruik je SAVE CODE?
SAVE CODE kan als een directe opdracht worden gebruikt of om een statement te vormen in een programma. SAVE wordt gevolgd door de archief naam, dit is een rij value. Hierna volgt dan CODE. CODE wordt op zijn beurt gevolgd door twee numerieke values, gescheiden door een komma, bijvoorbeeld:

SAVE "picture" **CODE**
16384, 6912

De archiefnaam volgend op SAVE kan tot 1Ø characters bevatten. De twee values volgend op CODE worden beiden afgerond naar het dichtst bij zijnde gehele getal. De eerste value goeft het start address (16384 boven) van de informatie in het geheugen, en de tweede value (6912) geeft het aantal bytes dat moet worden opgeslagen.

De informatie wordt dan naar de band gezonden op dezelfde wijze als bij een programma met SAVE.

Format
SAVE string-expr **CODE** int-num-expr, int-num-expr

SAVE DATA

Plaats op het toetsenbord S
Extend modus
Symbool D

Statement/opdracht

SAVE DATA slaat een array op band op. Het array kan dan worden geladen als je LOAD DATA gebruikt.

Hoe gebruik je SAVE DATA?
SAVE DATA kan worden gebruikt om een statement te vormen on een programma of als een directe opdracht. SAVE wordt gevolgd door een archief naam, dan door DATA, een letter of een letter met $ en tot slot een paar lege haakjes, bijvoorbeeld:

4ØØ SAVE "nummers" **DATA** n ()
75Ø SAVE 'names" **DATA** n$ ()

De archief naam van het array kan tot 1Ø characters bevatten. De letter (of letter met $) volgend op DATA is de naam van het array in het programma dat op de band moet worden opgeslagen. Het array wordt dan op dezelfde wijze naar de band gestuurd als bij een programma met SAVE.

Format
SAVE string-expr **DATA** letter [$] ()

SAVE SCREEN$

Plaats op het toetsenbord
Extend modus
Symbool K

Statement/opdracht

SAVE SCREEN$ slaat de afbeeldingen van het beeldscherm op band op. Deze kunnen worden terug gebracht in de computer door LOAD-SCREEN$ te gebruiken.

Hoe gebruik je SAVE SCREEN$?
SAVE SCREEN$ kan worden gebruikt als een directe opdracht of om een statement in een programma te vormen. SAVE wordt gevolgd door de archief naam, dit is een rij-value en dan door SCREEN$, bijvoorbeeld:

SAVE "picture" **SCREEN$**

De archiefnaam kan tot 1Ø characters bevatten. De afbeelding wordt dan naar de band gezonden op dezelfde wijze als bij een programma met SAVE.

Format
SAVE string-expr **SCREEN$**

SCREEN$

Plaats op het toetsenbord
Extend modus
Symbool K

Functie

SCREEN$ neemt waar welk

character op een bepaalde plaats op het scherm verschijnt.

Hoe gebruik je SCREEN$?
SCREEN$ wordt gevolgd door twee numerieke values, gescheiden door een komma en tussen haakjes geplaatst, bijvoorbeeld:

```
16Ø IF SCREEN$ (l,c) = "X"
    THEN PRINT "CRASH"
```

De values volgend op SCREEN$ worden naar het dichtst bij zijnde gehele getal afgerond, als dit nodig is. De eerste value (1 boven) kan varieren van Ø tot 21 en geeft het nummer van de line van een positie op het beeldscherm. De tweede value (c boven) kan varieren van Ø tot 31 en geeft het kolom nummer van een positie op het scherm. SCREEN$ retourneert dan het afgebeelde character op deze positie als een rij constant (het character tussen aan halings tekens "X" boven).

Als er geen character op die positie is, dan geeft SCREEN$ een nul-rij ("").

Format
SCREEN$ (int-num-expr, int-num-expr)

SGN SIGN

Plaats op het toetsenbord
Extend modus
Symbool F

Functie

SGN geeft aan wanneer een getal positief, negatief of nullis.

Hoe gebruik je SGN?
SGN wordt gevolgd door een numerieke value, bijvoorbeeld:

```
5Ø LET x = SGN y
```

Een uitdrukking moet tussen haakjes worden geplaatst. SGN geeft dan 1 als de value van argument (y boven) positief is, SGN geeft-1 als het negatief is en Ø als het nul is.

Format
SGN num-const
SGN num-var
SGN (num-expr)

SIN SINas

Plaats op het toetsenbord
Extend modus
Symbool Q

Functie

SIN geeft de sinus van een hoek.

Hoe gebruik je SIN?
SIN wordt gevolgd door een numerieke value, bijvoorbeeld:

```
8Ø LET x = SIN y
```

Een uitdrukking moet tussen haakjes worden geplaatst. De value volgend op SIN is de hoek in radialen, en SIN geeft de sinus van de hoek. Graden kunnen in radialen worden omgezet door te vermenigvuldigen met π/18Ø.

Merk op dat SIN een positieve value geeft voor hoeken tussen Ø en 18Ø graden en een negatieve value voor hoeken tussen 18Ø en 36Ø graden.

Format
SIN num-const
SIN num-var
SIN (num-expr)

SQR

Plaats op het toetsenbord
Extend modus
Symbool H

Functie

SQR geeft de vierkants wortel van een getal.

Hoe gebruik je SQR?
SQR wordt gevolgd door een numerieke value, bijvoorbeeld:

```
7Ø LET x = SQR y
```

Een uitdrukking moet tussen haakjes worden geplaatst. De value volgend op SQR (y boven) moet groter zijn dan nul. SQR geeft dan de vierkants wortel.

Format
SIN num-const
SIN num-var
SIN (num-expr)

STEP

Plaats op het toetsenbord
Symbool D

Zie **FOR**

STOP

Plaats op het toetsenbord
Symbool A

Statement/opdracht

STOP stopt het programma op een bepaald punt. Het kan nodig zijn STOP te gebruiken om de hoofd-sectie van een programma te beëindigen om zodoende de sub-programmas te beperken tot een aparte sectie. STOP kan ook worden gebruikt om een programma te zuiveren van fouten.

Hoe gebruik je STOP?
STOP wordt gewoonlijk gebruikt om een statement om een programma te vormen. Het wordt alleen gebruikt, bijvoorbeeld:

```
65Ø STOP
```

Bij uitvoering stopt het programma en het verslag,

```
9 STOP statement
```

verschijnt met het line en statement nummer waar het programma stopte. Zuiverings procedures, zoals het afbeelden en veranderen van de values, kunnen dan worden ondernomen. Het invoeren van CONTINUE laat het programma weer starten en wel bij de volgende statement met de nieuve values.

Format
STOP

STR$

Plaats op het toetsenbord
Extend modus
Symbool Y

Functie

STR$ zet een getal om in een rij.

STR$ wordt gevolgd door een numerieke value, bijvoorbeeld:

```
9Ø LET a$ = STR$ x
```

Een uitdrukking moet tussen haakjes worden geplaatst. STR$ geeft dan de value van zijn argument (x boven) als een rij-constant. Als aan x de value 65 was gegeven dan geeft de statement hierboven aan a$ de value "65".

Format
STR$ num-const
STR$ num-var
STR$ (num-expr)

TAB

Zie **LPRINT, PRINT**

TAN TANgent

Plaats op het toetsenbord
Extend modus
Symbool E

Functie

TAN geeft de tangens van een hoek.

Hoe gebruik je TAN?
TAN wordt gevolgd door een numerieke value, bijvoorbeeld:

13Ø LET = TAN y

Een uitdrukking moet tussen haakjes staan. De value volgend op TAN is de hoek in radialen en TAN geeft de tangens van de hoek. Graden kunnen worden omgezet in radialen door te vermenigvuldigen met $\pi/18Ø$.

Merk op dat TAN een positieve waarde geeft voor hoeken tussen Ø en gradenen tussen 18Ø en 27Ø graden. Voor hoeken tussen 9Ø en 18Ø graden en 27Ø en 36Ø geeft TAN een negatieve waarde.

Format
TAN num-const
TAN num-var
TAN (num-expr)

THEN

Plaats op het toetsenbord
Symbool G

Zie **IF**

TO

Plaats op het toetsenbord
Symbool F

Functie

TO heeft twee verschillende toepassingen in Sinclair BASIC. Het wordt gebruikt samen met FOR om zo een FOR NEXT loop te vormen (zie FOR voor details) en het wordt ook gebruikt om rijen in kleinere sub-rijen te knippen.

Hoe gebruik je TO bij het, rij-knippen?
TO wordt gebruikt om de eerste en de laatste character van een sub-rij te bepalen, binnen de hoofd-rij. TO wordt vooraf gegaan door een rij-value en een haakje-openen, en dan een numerieke value naar keuze. Het wordt gevolgd door een andere numerieke value naar keuze en dan een haakje sluiten, bijvoorbeeld:

8Ø PRINT a$ (4 TO 7)

Een rij uitdrukking moet ook tussen haakjes worden geplaatst. De rij-value (a$ boven) geeft de rij doe geknipt moet worden. De twee numerieke values (4 en 7) bepalen de posities van de eerste en de laatste characters van de sub-rij, binnen de rij. TO geeft dan de sub-rij, (characters 4 tot 7 van a$).

De eerste numerieke value heeft een default-value van 1 en de tweede heeft een default value gelijk aan de positie van het laatste character in de rij.

Format
string-const ([num-expr] **TO** [num-expr])
string-var ([num-expr] **TO** [num-expr])
[string-expr) ([num-expr] **TO** [num-expr])

USR User Sub Routine (gebruikers subprogramma)

Plaats op het toetsenbord
Extend modus
Symbool L

Functie

USR wordt gebruikt om een sub programma in machine code op te roepen. Dit sub programma is op een bepaald address in het geheugen geplaatst. Het wordt ook gebruikt om de data, van de door de gebruiker bepaalde graphics, in de gereserveerde plaats en boven in het geheugen te plaatsen.

USR in machine code
Om machine code be gebruiken, wordt USR gevolgd door een numerieke waarde, bijvoorbeeld:

8Ø PRINT USR 65ØØØ
1ØØ RANDOMIZE USR 65ØØØ

Een uitdrukking moet tussen haakjes worden geplaatst. De value volgend op USR wordt afgerond naar het dichtst bij zijnde gehele getal, en is dan het start address in het geheugen, waar een sub-programma in machine-code is geplaatst. Iedere statement met USR roept dan het sub-programma van dit address op en USR geeft de value van de inhoud van het bc-register-pair.

USR en door de gebruiker bepaalde graphics.
Om door de gebruiker bepaalde graphics te vormen wordt USR met POKE gebruikt. Het wordt gevolgd door een rij constant of een rij-variabele om een address te geven voor een POKE statement, bijvoorbeeld:

5Ø POKE USR "a", 255

De rij-value volgend op USR kan een enkele letter zijn, variërend van A tot U of van a tot u. Hoofd letters worden niet onderscheiden van gewone letters.

USR geeft dan het start address van een van de 21 secties van het geheugen die zijn gereserveerd voor door de gebruiker bepaalde graphics. Iedere sectie bevat 8 addresses waaraan m.b.v. POKE 8 bytes aan verbonden zijn, om zo een graphics character te vormen.

Format
USR int-num-const
USR int-num-var
USR (int-num-expr)
USR string-const
USR string-var

VAL VALue

Plaats op het toetsenbord
Extend modus
Symbool J

Functie

VAL verandert een rij met een numerieke value in een getal.

Hoe gebruik je VAL?
VAL wordt gevolgd door een rij-constante of een rij-variabele, bijvoorbeeld:

7Ø LET x = VAL a$

De value van een rij-constante of rij-variabele, wordt ontdaan van zijn aanhalingstekens, en moet dan een numerieke value zijn. Vals bewerkstelligt dit, door het te retourneren als een numerieke

constante, bijvoorbeeld:
Als a$ de value "435" heeft, dan wijst de boven gegeven statement een value van 435 toe aan X. VAL kan echter ook de waarde bepalen van uitdrukkingen, bijvoorbeeld:

```
1Ø INPUT a$, x
2Ø PRINT VAL a$
```

De rij value die is tooegewezen aan a$ zou een uitdrukking moeten zijn die x gebruikt, bijvoorbeeld "x*x". Er wordt dan een numerieke value toegekend aan x, bijvoorbeeld 5. VAL neemt de aanhalingstekens weg van de rij-value om x*x te krijgen, en bepaalt er de waarde van door de aan x toegewezen value te gebruiken. Het resultaat is 25.

Format
VAL string-const
VAL string-var

VAL$ VALue (string)

Plaats op het toetsenbord
Extend modus
Symbool J

Functie

VAL$ bepaald de waarde van een rij als een rij-uitdrukking.

Hoe gebruik je VAL$?.
VAL$ wordt gevolgd door een rij-variabele, bijvoorbeeld:

```
13Ø PRINT VAL$ z $
```

De value van de rij-variabele wordt ontdaan van zijn aanhalingstekens en moet dan een rij-uitdrukking zijn. VAL$ bepaalt de waarde van de uitdrukking en geeft de value als een rij-constant.

Probeer bijvoorbeeld dit programma:

```
1Ø INPUT a$³ x $
2Ø PRINT VAL$ z $
```

De rij-value die hoort bij a$ zou een uitdrukking moeten zijn die x $ gebruikt, bijvoorbeeld: "x$ + x$". Er wordt dan een rij-value toe gewezen aan x$, bijvoorbeeld "DO". VAL$ neemt de aanhalings tekens weg dan de value van a$ om zo x$ + x$ te karijgen. VAL$ bepaalt dan de waarde van deze uitdrukking door gebruik te maken van de value die was toegewezen aan x$³. Het resultaat is DODO

Format
VAL$ string-var

VERIFY

Plaats op het toetsenbord
Extend modus
Symbool R

Statement/opdracht

VERIFY controleerd of een programma op de juiste wijze op de band is opgeslagen, na gebruik van SAVE.

Hoe gebruik je VERIFY?
VERIFY wordt gewoonlijk als een directe opdracht gebruikt op precies dezelfde wijze als LOAD. Het wordt gevolgd door de naam van het programma, bijvoorbeeld:

```
VERIFY "archiefnaam"
```

Als de band wordt aangezet wordt de naam van ieder programma dat wordt gevonden, afgebeeld en ieder programma dat op de band staat met dezelfde naam wordt vergeleken met het programma in het geheugen. Als blijkt dat de twee hetzelfde zijn dan wordt het volgende verslag gegevan:

```
Ø OK, Ø:1
```

Merk op dat VERIFY "het volgende programma op de band vergelijkt met dat in het geheugen.

VERIFY CODE en VERIFY DATA
VERIFY CODE kan op precies dezelfde wijze worden gebruikt als LOAD CODE en wel om bevestigen dat een sectie can de geheugen-informatie op band is opgeslagen. VERIFY DATA werkt op dezelfde wijze als LOAD DATA en wel om te controleren dat een array is opgeslagen op band. (Zie LOAD CODE en LOAD DATA voor details).

Format
VERIFY string-expr
VERIFY string-expr **CODE** [int-num-expr] [,int-num-expr]
VERIFY string-expr **DATA** letter [$] ()

VERIFY CODE

zie **VERIFY**

Plaats op het toetsenbord
Extend modus
Symbool R
Extend modus
I

VERIFY DATA

Zie **VERIFY**

Plaats op het toetsenbord
Extend modus
Symbool R
Extend modus
D

ZX SPECTRUM+ BEELDSCHERM VERSLAG

Een command verschijnt als Ø en in een line, statement 1 is aan het begin van de line, statement 2 komt na de eerste colon of THEN enz. CONTINUE het programma te continueren bij de statement in het verslag aangegeven.

Ø OK
Succesvolle aanvulling, of succesvolle sprong naar een line-nummer groter dan enig ander in het programma. CONTINUE negeert dit verslag en hervat bij de statement omschreven in het voorofgaande verslag.

1 NEXT without FOR
NEXT wordt aangetroffen zonder FOR in er bestaat een variabele met dezelfde naam als de controle variabele.

2 Variable not found
Een eenvoudige variabele word gebruikt zonder er een value aan toe te wijzen of de value van de band te laden. Een controle variabele werd gebruikt met NEXT zonder het eerst op te stellen in een FOR statement. Een ingetekende variabele werd gebruikt voordat de omvang van de array met DIM was bepaald of voordat een array van de band was geladen.

3 Subscript wrong
Een intekening valt buiten de proporties van het array.

4 Out of memory
Er is niet genoeg geheugenruimte over om de statement of de opdracht af de maken.

5 Out of screen
INPUT heeft meer dan 23 lines opgewekt in het onderste gedeelte van het scherm, of er is een line-nummer van 22 of meer gebruikt met PRINTAT.

6 Number too big
De computer heeft geprobeerd een getal te produceren dat groter is dan 10^{38}.

7 RETURN without GOSUB
Het aantal RETURN statements is een groter dan het aantal GOSUB statements.

8 End of file
Microdrive file-handling raport.

9 STOP statement
STOP werd gebruikt om het programma stil te zetten. CONTINUE zal het programma hervatten bij de volgende statement.

A Invalid argument
Een verkeerd argument of een verkeerde value werd aan een functie gegeven.

B Integer out of range
Een value werd afgerond naar het dichtst bij zijnde gehele getal en valt buiten het bereik waarbinnen het geaccepteerd kan worden.

C Nonsense in BASIC
De statement is onzin in BASIC, in de context van het programma. Bijvoorbeeld een ongeldige uitdrukking die wordt gebruikt met VAL or VAL$.

D BREAK – CONT repeats
BREAK werd ingedrukt. CONTINUE zal de statement, waar de computer stopte, herhalen.

E Out of DATA
READ heeft geprobeerd om verder te lezen dan het einde van de laatste DATA statement in het programma RESTORE kan nodig zijn.

F Invalid file name
SAVE werd gebruikt met een naam welke geen of meer dan 1Ø characters bevat.

G No room for line
Er is niet genoeg geheugen ruimte over gebleven om een nieuwe programma line intevoeren.

H STOP in INPUT
STOP werd ingevoerd als reactie op INPUT of begon aan de ingevoerde data. CONTINUE herhaald de INPUT statement.

I FOR without NEXT
Een FOR NEXT loop werd niet uitgevoerd omdat de limiet of de STEP value verkeerd waren, (Bijvoorbeeld, FOR x = 5 TO Ø) en de overeenkomstige NEXT niet werd gevonden.

J Invalid I/O device
Microdrive file-handling rapport

K Invalid colour
De value specifiek voor INK, PAPER, FLASH, BRIGHT, INVERSE of OVER of het overeenkomstige controle character is buiten bereik.

L BREAK into program
BREAK werd ingedrukt. Het rapport bepaalt de volgende statement nader en CONTINUE hervat het programma met de volgende statement.

M RAMTOP no good
De value die wordt bepaald voor RAMTOP is of te groot of te klein.

N Statement lost
Er werd een sprong ondernomen naar een statement dat niet langer bestaat.

O Invalid stream
Microdrive file-handling rapport

P FN without DEF
Er werd een FN statement gebruikt zonder de overeenkomstige DET FN statement.

Q Parameter error
Een FN statement bevat het verkeerde aantal values dat moet worden door gegeven aan de functie, of een van de values is van het verkeerde type (een rij-value in plaats van een nummer of omgekeerd).

R Tape loading error
De laad procedure of de verificatie procedure heeft gefaald. Als de procedure van het samengaan (van programm's) heeft gefaald.

VERDER DAN BASIC

BASIC is een algemeen toepasbare computertaal die erg goed werkt voor de meeste toepassingen. Het is echter niet de enige computertaal die je op de Spectrum kunt gebruiken. Je kunt ook Software krijgen dat andere talen levert zoals FORTH, micro-PROLOS en LOGO. Deze talen werken op een heel andere manier als BASIC en geven nieuwe mogelijkheden voor je computer.

Omdat BASIC een algemeen toepasbare taal is kan het voor sommige toepassingen nogal omslachtig zijn. Het is ook redelijk langzaam. Andere talen kunnen een grotere so epelheid geven, gecombineerd met eenvoud van de programmeerbaarheid en een grotere snelheid. FORTH bijvoorbeeld geeft je de mogelijkheid je eigen woorden te bepalen en ze te gebruiken in de instructies die de computer begrijpt en die hij ongeveer tien keer zo snel uitvoerd als dezelfde opdrachten in BASIC. Met micro-PROLOB, zal de computer eenvoudige Engelse zinnen begrijpen, die hij dan in zijn geheugen opslaat als een basis voor een, gesprek met de gebruiker. LOGO is een computertaal die is ontwikkeld voor educatief gebruik. Als je echter echt snelle programma's wilt schrijven voor je ZX Spectrum+ dan moet je begrijpen hoe je in machine code kunt programmeren.

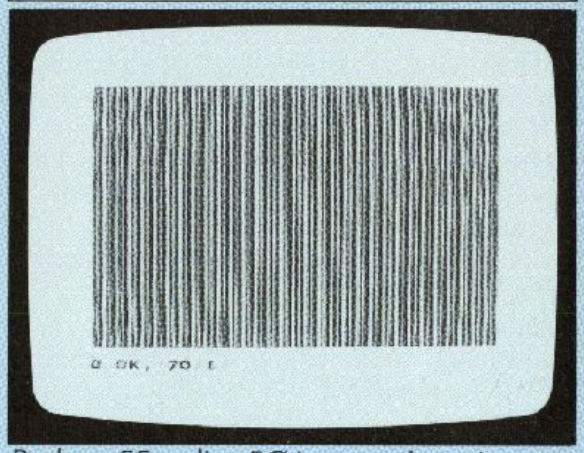
Machine Code

BASIC wordt gebruikt om jouw instructies aan de computer te kunnen geven in een vorm die gemakkelijk te begrijpen is. De Spectrum CPU – de machtige Z8ØA chip begrijpt BASIC niet echt. Een gedeelte van het geheugen bevat een permanent programma dat, BASIC vertaler wordt genoemd en jouw BASIC instructies omzet in een serie code-signalen.

De vertaler heeft enige tijd nodig om jouw BASIC instructies te vertalen in de Z8ØA code ofwel de machine code. Je kunt echter BASIC, overslaan als je dat wil en de machine code direct naar de Z8ØA sturen. Je programma wordt dan heel snel uitgevoerd. De prijs die je moet betalen voor deze tijd-winst is de tijd die je nodig hebt om een programma in machine code te schrijren. In begenstelling tot BASIC is het een erg lastige taal. Het neemt veel tijd om hem te leren. Hoeveel we in deze gids de machine-code niet bespreken, zijn er veel boeken te krijgen waaruit je de Spectrum machine-code tot een gevorderd niveau kunt leren. Om een idee van de snelheid van machine-code te krijgen kun je het volgende programma eens uitproberen.

SNELLE-STREPEN

```
  10 FOR x=0 TO 15
  20 READ n: POKE 65000+x,n
  30 NEXT x
  40 DATA 33,255,63,1,1,24,22
  50 DATA 55
  60 DATA 35,11,120,177,200,114,
24,248
  70 RANDOMIZE USR 65000
```



Probeer 55 op line 5Ø te veranderen in een value van 1 tot 2155 en zie hoe de strepen veranderen. Je kunt ook gekleurde strepen produceren door eerst een INK opdracht te gebruiken. Dit is echter niet de bedoeling van het programma. Zie hoe snel de afbeelding wordt geproduceerd als je de machine-code gebruikt – het is praktische direct.

Dit programma werkt omdat de DATA statements 16 codes bevatten welke zijn opgeslagen op addresses 65ØØØ en verder, in het geheugen door de lines 1Ø tot 3Ø. Line 7Ø zendt de codes naar de Z8ØA en de afbeelding wordt direct gegeven.

Veel van de spelletjes die je voor je Spectrum kunt krijgen zijn in machine-code geschreven om zo super-snelle-actie te krijgen. Om je te helpen om in machine-code te schrijven zijn er speciale programma's te krijgen, de assemblers. Deze geven instructies die je intoetst inplaats van alleen getallen, en dat is wat de machine-code zelf nodig heeft. De instructies zijn geen Engelse woorden zoals de keywords in BASIC, maar afkortingen of mnemonics welke de operaties vertegenwordigen die de computer moet verrichten. Je moet daarom begrijpen hoe de computer werkt, stap-voor-stap, voor dat je assembly taal kunt gebruiken.

COMPUTERTAAL – WAT HET BETEKEND

Veel worden die worden gebruikt in de computer-wereld worden ook in het dagelijks leven gebruikt, maar met een andere betekenis. Hier volgen de verklaringen van enkele van deze woorden die in dit boek worden gebruikt, samen met speciale computer-termen. De schuin-gedrukte woorden hebben een eigen betekenis. Als er een woord of term is in het boek, dat je niet begrijpt en het is hier niet verklaard, probeer het dan in de index te vinden.

Address Een enkele geheugen eenheid. Er zyn 65536 adressen in het ZX Spectrum+.

Argument Een waarde welke wordt gebruikt door een functie om een uitkomst te krijgen.

Array Een groep bij eklaar behorende gegevens (data) welke samen in één gedeelte van het geheugen zitten.

Attributen Codes die de kleur van de, characters geven.

BASIC De computer taal welke wordt gebruikt door de ZX Spectrum+ en de meeste andere micro computers voor thuis.

Binary code Het soort code dat door computers wordt gebruikt. Deze code bestaat uit volgorden van twee verschillende tekens, gewoonlijk een aan – of uit signaal bijvoorbeeld aan-uit elektrische signalen.

Bit In een computer duidt het op een aan – of uit signaal. Het is een samentrekking van binary digit.

Byte Een groep van 8 bits. Deze groep vertegenwoordigd een getal met een waarde van Ø tot 255. Elk addres in het geheugen bevan één byte.

Character Ieder los teken: een getal (Ø tot 9) een letter of een grafisch teken dat kan worden weergegeven of gedrukt.

Character set De volledige verzameling vaste characters en bepaalde controle codes gebruikt door de computer. ((bij voorbeeld de tekens op een toetsen bord)).

Command Een enkele instructie welke door de computer wordt uitgevoerd, ook wel direct-command genoemd.

Constant Een getal of een groep van een of meer letters of andere characters.

CPU Central Processing Unit – Het centrale gedeelte van de computer dat het rekenwerk verricht en de andere gedeelten controleerd. De ZX Spectrum+ gebruikt hiervoor een Z8Ø microprocessor.

Cursor De positie op het scherm waar iets nieuws zal worden afgebeeld. Het kan worden aangegeven door een knipper signaal dat de modus aangeeft waarin de computer is.

Data Informatie of gegevens welke de computer of uit een programma krijgt of ingevoerd van buiten in het programma, zodat de computer resultaten kan produceren.

Direct Command Een samenstel van een of meer instructies welke direct worden uitgevoerd door de computer.

Edit Om kleine veranderingen in een programma aan te brengen.

Enter Om een volledige instructie of informatie aan de computer te geven.

Expression Een combinatie van constants, variables en keywords.

Function Een operatie waar bij de computer een of meer values (of arguments) neemt en deze gebruikt om een andere value (of argument) als uitkomst te geven.

Graphics De productie van beelden in de vorm van tekeningen, kaarten of diagrammen, door de computer.

Hardware De computer zelf en alle overeenkomstige apparaten of machines, zoals, peripherals.

Information Woorden getallen en tekens in iedere combinatie die de computer kan verwerken.

Input Programs en data ingevoerd in de computer.

Interface Een eenheid welke de computer en of periperals verbindt en welke garandeert dat zij met elkaar kunnen communiceren.

K Een eenheid van geheugen capaciteit van een computer. 1 K is gelijk aan 1 kilobyte of wel 1Ø24 bytes. De geheugen-capaciteit in K is gelijk aan het totale aantal addresses in het geheugen, waar van elk een byte kan opslaan. De ZX Spectrum+ heeft een 48K RAM en een 16K ROM, of wel een totaal van 64K.

Keyword Een computer instructie in BASIC. Het kan enige values vereisen om te werken. ((Met behulp van van een keyword wordt toegang tot een computer systeem verkregen)).

Line Een instructie of set van instructies in een programma. De line heeft een nummer zodat zij wordt uitgevoerd op de juiste plaats in de juiste volgorde met andere lines.

Listing De lines van een programma in de juiste volgorde geregistreerd.

Load Het invoeren van een programma of data in de computer vanuit een opslag-apparaat zoals een cartridge of cassette.

Logic Het proces volgens het welk de computer beslist of resultaken juist of onjuist zijn en of de states ((Basic acties)) waar of vals zijn.

Loop Een gedeelte van een programma dat een of meer malen wordt herhaald. Een lus.

Machine Code De taal welke de ZX Spectrum+ begrijpt. Programma's in BASIC worden in de machine code vertaald door de computer, terwijl deze het programma volgt.

Memory ((Het geheugen)). Dat gedeelte van de computer dat het programma en de, data bevat indien vereist, en ook zijn hier de permanent operating instructions ((vaste gebruiks aanwijzingen)) te vinden.

Mode In de Spectrum, een van de 5 states welke aangeven welk keyword en character kan worden geproduceerd door elk van de toetsen op het toetsenbord. Gedurende het programmeren wordt de modus aangegeven door een knipperende letter binnen de cursor.

Nesting De ordening van loops binnen een programma zodat een of meer loops worden uitgevoerd binnen elkaar.

Numeric variable Een variabele welke uit een of meer cijfers bestaat. Deze variabele omvat een getal.

Operator Een instructie welke rekenen of logica uitvoerd.

Output De resultaten van de computer.

Peripheral Ieder apparaat dat met een computer is verbonden. ((bijvoorbeeld: Magneet banden, diskdrives, kaartlezers, printers ed.)).

Pixel De kleinste kleurvlek welke op het scherm kan verschijnen. Afkorting voor, picture cell.

Print Ofwel om de resultaten of grafics op het beeldscherm te vertonen, danwel deze aftedrukken op een printer.

Program ((Programma)). Een volgorde van instructies welke de computer moet uitvoeren.

RAM Random Access Memory Dat gedeelte van het geheugen waarin programma's (en gegevens) kunnen worden ingevoerd, evenals andere veranderende values. Ookwel het toegankelijk geheugen genoemd. De inhoud van RAM wordt uitgewist als de computer wordt uitgezet. De ZX Spectrum+ heeft een 48K RAM.

Register Een kleine geheugen-eenheid afzondertijk van het hoofd geheugen. Registers ((of hulpgeheugens)) binnen de CPU worden gebruikt voor het uitvoeren van het reken-proces.

Report Een bericht dat de computer afbeeld om zijn acties weer te geven.

Resolution De graad van gedetaileerdheid mogelijk in de computer-afbeeldingen.

ROM (Read Only Memory). Dat gedeelte van het geheugen dat de permanente programma's en instructies voor de computer bevat. De ZX Spectrum+ heeft een 16K ROM.

Save Om een programma of data in een opslag-apparaat (zo als een cartridge of een cassette) op te slaan.

Scroll De vloeiende beweging van characters op een beeldscherm welke het mogelijk maakt meer af te beelden dan wanneer het beeld niet door zou (open).

Statement Ofwel een keyword dat wordt gebruikt om een instructie te vormen in een programma line, of de opdracht zelf.

String Een groep van een of meer characters ingesloten in rijen om hen van andere getallen en numerieke variabelen te onderscheiden.

String variable Een variabele die een rij bevat. Zij bestaan altijd uit een enkele letter en het teken.

Syntax De juiste volgorde van keywords, constants, variabelen en expressions welke vereist zijn om een bruikbare BASIC – opdracht te vormen.

Value Ieder getal of rij dat gegeven wordt door of vertegenwoordigd door een constante, variabele of uitdrukking.

Variable Een of meer geheugen-eenheden welke een specifieke constant bevatten, voor gebruik door de computer. Iedere eenheid krijgt een naam of een letter zodat je hem gemakkelyk terug vindt. De ZX Spectrum+ maakt een onderscheidt tussen, numeric variables en string variables.

INDEX

De schuingedrukte bladzijde nummers verwijzen naar illustraties en titels

Voor het eerst uitgegeven in 1984 door Dorling Kindersley Ltd, 9 Henrietta Street, Londen WC2E 8PS in samenwerking met Sinclair Research Ltd, 25 Willis Road, Cambridge.

Redacteur David Burnie
Kunstredacteur Peter Luff
Ontwerper Debra Lee
Fotograaf Trevor Melton
Screen-shot fotograaf Vincent Oliver
Hoofdredacteur Alan Buckingham

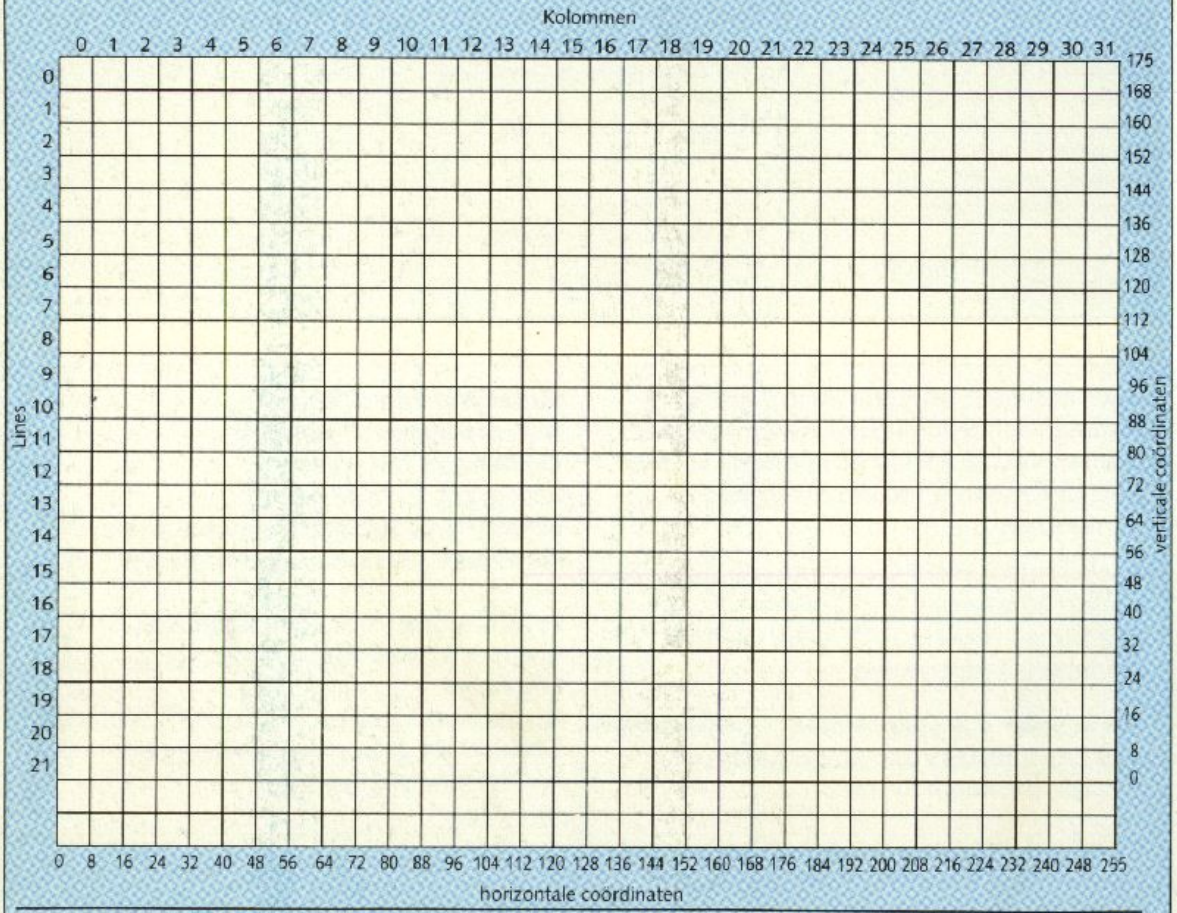
Letterzetting door The Letter Box Company (Woking) Ltd, Woking, Engeland
Reproductie door A Mondadori, Verona
Gedrukt en gebonden in Italie door A. Mondadori, Verona.

Beeld afbeeldings rooster.

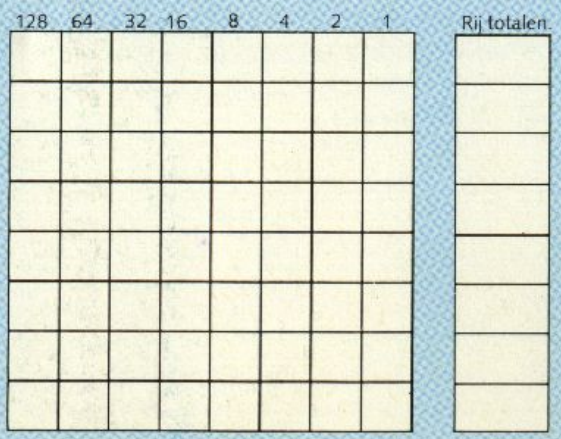
Het rooster hieronder laat de coördinaten zien van zowel low-resolution als high-resolution graphics. Hoewel individuele graphics keywords alleen werken met ofwel low-ofwel high resolution coordinaten, kun je zowel low- als high resolution keywords gebruiken als je een beeld produceert.

Het low-resolution rooster is verdeeld in twee delen, het hoofd-afbeeldings gebied en dan twee lines onder aan het scherm. PRINT AT zal characters in het bovenste gedeelte produceren. INPUT AT zal characters on het onderste gedeelte produceren. De coördinaten van het low-resolution scherm kun je zien aan de linker boven zijde van het rooster.

Het high-resolution rooster betreft alleen het hoofd-afbeeldings gebied. PLOT, DRAW en CIRCLE worden gebruikt om high-resolution graphics te produceren. De coördinaten voor high-resolution kun je zien aan de rechter zijde (beneden) van het rooster.



Door de gebruiker gedef iniëerd graphics rooster
Tel de gegeven getallen op om de decimale value van iedere rij van het rooster te krijgen. Gebruik dan POKE USR om het character te programmeren.



SPECTRUM SOFTWARE

De gehele Spectrum computer software range (inclusief alle bestaande titels) is geheel compatibel met je nieuwe ZX Spectrum +

DORLING KINDERSLEY LTD
in samenwerking met
SINCLAIR RESEARCH LTD

ZX Spectrum+ Gebruikshandleiding