

Brückmann/Gerits/Wiens

Het printer- boek voor de commodore printers

Commodore Bibliotheek 15

**DATA BECKER
NEDERLANDS***

Brückmann/Gerits/Wiens

Het printer- boek voor de commodore printers

Commodore Bibliotheek 15

**DATA BECKER
NEDERLANDS***

Oorspronkelijke titel: Das grosse Druckerbuch
Copyright (c) 1984 Data Becker GmbH, Düsseldorf

Voor de Nederlandse vertaling:
Copyright (c) 1985 Uitg. A. W. Bruna & Zoon, Utrecht
Omslagontwerp: Martin van Keulen
Vertaling, produktie en redaktie: ProCread/Informatica, Groningen
Druk: C. Haasbeek, Alphen aan den Rijn
ISBN 90 229 3362 8
D/1986/0939/97

1e druk 1985
2e druk 1986

De keuze en de opbouw van de programma's die in dit boek voorkomen, zijn gebaseerd op hun educatieve waarde. Alle programma's zijn getest op hun juiste werking. De uitgever kan geen enkele verantwoordelijkheid voor eventueel toch optredende fouten aanvaarden.

Uit deze uitgave mag niets worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

Boeken en programma's van

DATA BECKER
NEDERLANDS*

worden in de handel gebracht door;
A.W. Bruna & Zoons Uitgeversmij b.v.,
Postbus 8411, 3503 RK Utrecht
en
A.W. Bruna & Zoon n.v.,
Antwerpsesteenweg 29a, 2630 Aartselaar.

INHOUD

- 1 INLEIDING 7**
- 2 INTERFACES 8**
 - 2.1 De IEC-bus van de Commodore 64 9
 - 2.1.1 Het principe van de werking 9
 - 2.1.2 Adressering 10
 - 2.1.3 Gegevensoverdracht 11
 - 2.2 De Centronics interface 15
 - 2.3 De V24 interface 20
- 3 DE SECUNDAIRE ADRESSEN 25**
 - 3.1 VC-1525 / GP-100VC / MPS-801 26
 - 3.2 VC-1526 27
 - 3.3 Epson printer met DATA BECKER interface 30
- 4 SCHRIJFMACHINE AAN DE USER-PORT 33**
- 5 PROGRAMMA'S VOOR VIC 20 EN COMMODORE 64 39**
 - 5.1 Kleine hulpprogramma's voor dagelijks gebruik 39
 - 5.1.1 DEVICE NOT PRESENT? 40
 - 5.1.2 Het formateren van numerieke en alfanumerieke data 42
 - 5.1.3 Geformateerde programma-listings 48
 - 5.2 Tekstverwerking 56
 - 5.2.1 MINITEKST, een eenvoudige teksteditor 56
 - 5.2.2 Kan het ook groter? 67
 - 5.3 Hardcopy 72
 - 5.3.1 Hardcopy van tekst 73
 - 5.3.2 Hardcopy van grafische voorstellingen 77
 - 5.4 Grafiek met of zonder l-naaldssturing 88
 - 5.4.1 Grafiek zonder l-naaldssturing 89
 - 5.4.2 Grafiek met l-naaldssturing 93
 - 5.4.3 Afbeelding van driedimensionale Hi-Resgrafiek 100
- 6 PRINTERTYPES 106**
 - 6.1 De daisywheel- of margrietwielprinter 106
 - 6.2 De parallelle matrixprinter 107
 - 6.3 De seriële matrixprinter 108
- 7 HET BEDRIJFSSYSTEEM VAN DE MPS-801 111**
 - 7.1 De processor 8039 111
 - 7.2 Blokschema van de MPS-801 115
 - 7.3 De ROM-listing voor de MPS-801 118
- 8 DE PLOTTER COMMODORE 1520 184**
 - 8.1 Werkwijze en gebruiksmogelijkheden van de plotter 184
 - 8.2 De secundaire adressen van de Commodore 1520 190

8.3	Allerlei soorten figuren	208
8.3.1	Vierkant	208
8.3.2	Rechthoek en kubus	209
8.3.3	Parallelogram	214
8.3.4	Driehoeken 1	216
8.3.5	Curven	222
8.3.6	Driehoeken 2	228
8.4	Functies plotten	230
8.5	Bijzondere werkzaamheden met de Commodore 1520	241
8.6	Statistiek met de VC-1520	247
8.7	Hulpprogramma voor de VC-1520	254

1 INLEIDING

Wie voor het eerst over dit boek hoorde vertellen, heeft misschien gedacht: "Hoe kan iemand een heel boek volschrijven over een dergelijk ongecompliceerd apparaat als de printer?". Nu hebt u het boek zelf in handen gekregen en u ziet dat het best omvangrijk geworden is. Wat wordt er nu allemaal in verteld? Het beschrijft alle aspecten, van het stopcontact tot aan het bedrijfssysteem. Daartussen ligt een hele wereld. De meeste lezers zullen zeer geïnteresseerd zijn in de omvangrijke verzameling programma's. Maar ook de lezer die bijvoorbeeld de vierkante haakjes tegen een trema wil ruilen, kan gemakkelijk de plaats vinden waar beschreven staat hoe dit moet gebeuren. We hebben zoveel mogelijk geprobeerd om zowel bij de beschrijvingen als bij de programma's het hele aanbod van printers te bespreken die op de Commodore 64 van toepassing zijn. Voor het gebruik van uw printer zullen er ongetwijfeld ook enige wetenswaardigheden tussen zitten. Hoofdstuk 8 is in zijn geheel aan de plotter VC-1520 gewijd. Als we dit apparaat in de andere paragrafen hadden moeten bespreken, waren ze te lang geworden. Ten slotte wensen we u veel plezier bij het lezen van dit boek en... veel succes.

De schrijvers.

2 INTERFACES

In de homecomputerij bestaat er voor de technici haast geen interessanter thema dan dat van de interfaces. Op dat gebied kan hij zich nu eens echt uitleven en het ene apparaat op het andere proberen aan te sluiten. Een bespreking van de interfaces is onvermijdelijk: er bestaat namelijk geen kant en klare printer die mogelijkheden bezit die u nu juist noodzakelijk acht. Net zoals er een wereld van verschil ligt tussen uw wensen en het apparaat dat in de handel te verkrijgen is, bestaan er verschillen tussen de printers onderling. Wanneer u denkt eindelijk de juiste printer gevonden te hebben, blijkt hij weer net niet bij uw computer te passen. "Dat is toch logisch", zal de vakhandelaar zeggen, "uw computer heeft interface X en de printer de aansluiting Y. Iedereen kan toch op zijn vingers natellen dat zoiets niet werkt". Maar is dat nu werkelijk zo logisch? In dit hoofdstuk hebben we ons ten doel gesteld een uitvoerige uitleg van de aansluitmogelijkheden te geven die in aanmerking komen. En dat zal u uiteindelijk in staat stellen een eigen gefundeerde mening te vormen over de compatibiliteit of de mate van aanpasbaarheid van de verschillende interfaces. Een reeks elektrische verbindingen die bij elkaar horen noemen we in de elektronica een bus. Zo is er een databus om de data van het ene naar het andere onderdeel van de computer te sturen. Er is een adresbus waarop de codering van de adressen te vinden is. Er is ook een interfacebus voor algemeen gebruik, de General Purpose Interface Bus (GPB) of IEEE-488. Een interface is de verbinding van verschillende toestellen die tot het computersysteem en de daarmee verbonden randapparatuur behoren. Bij een verbinding over langere afstand wordt bij voorkeur gebruik gemaakt van een seriële interface. Daarbij vindt de overdracht van gegevens plaats over enkele lijnen waarnaast ook een paar stuurlijnen actief zijn om de overdracht goed te regelen. Serieel wil zeggen dat de gegevenssignalen bit voor bit achter elkaar worden verzonden en ontvangen. Daarvoor is aan weerszijde van de verbinding een ingewikkelde vertaling nodig van bytes naar datastroom en weer naar bytes. Uiteraard kost dat tijd en veel vernuft waarbij een goede norm van groot belang is. Een veel groter aantal lijnen is nodig bij een parallelle interface die vooral voor communicatie over korte afstanden wordt gebruikt. Door de parallelle verbinding hoeven de databytes niet te worden geconverteerd en kunnen de transmissies dus ook veel sneller en directer plaatsvinden. Daar staat weer tegenover dat de besturing van de parallelle interface zeer ingewikkeld en duur wordt als er tweerichtingsverkeer nodig is of zelfs meer toestellen tegelijk op eenzelfde verbinding zijn aangesloten. Bij de Commodore 64 is een compromis gevonden tussen beide

interfaces. We beginnen met deze IEC-bus. Het gaat immers om de Commodore 64.

2.1 De IEC-bus van de Commodore 64

De seriële IEC-bus van de Commodore 64 is niet dezelfde als de IEC-625 1, zoals de Amerikaanse IEEE-488-interface volgens de Duitse Industrie Norm (DIN) wordt genoemd. Ondanks het geringe aantal gebruikte lijnen wordt ongeveer volgens het systeem van de "grote" IEC-bus gewerkt. En dat is beslist niet eenvoudig. Bekijk u het zelf maar eens.

2.1.1 Het principe van de werking

De seriële IEC-bus heeft zes lijnen (ter vergelijking: de parallelle bus heeft er 24):

- 1 SRQ Service ReQuest. Via de SRQ-lijn laat het apparaat de computer weten dat het nieuwe gegevens nodig heeft, dat het gegevens wil aanbieden of dat het een bepaalde handeling verwacht. Bij de Commodore wordt de SRQ niet gebruikt.
- 2 GND GrouND
- 3 ATN ATteNtion. Elke keer als de computer een instructie aan een apparaat wil doorgeven activeert hij deze lijn. Met behulp van ATN wordt bewerkstelligd dat alle apparaten die aan de bus gekoppeld zijn, deze instructie ook ontvangen. Van tevoren staat immers nog niet vast voor welk apparaat de instructie bedoeld is. Dit wordt pas duidelijk wanneer het adres voor het betreffende apparaat wordt doorgegeven. Om die reden moet altijd eerst het device-nummer worden doorgegeven. De andere apparaten kunnen zichzelf dan van de bus "afkoppelen".

- 4 CLK CLoCK. Omdat de data niet byte voor byte (dus per 8 bits tegelijk) worden doorgegeven, maar bit voor bit, zet de zender (in het engels: talker) bij elke bit een klokpuls op de CLK om de geldigheid van de datalijn aan te geven.

- 5 DATA is de datalijn, waarlangs de seriële transmissie van een databyte (met de bit van de laagste waarde voorop) plaatsvindt. Daarnaast heeft deze lijn nog andere taken. Maar daarover leest u later meer.

- 6 RES RESet. Als deze lijn laag is, keren alle aangesloten apparaten in de rusttoestand terug, dus in dezelfde stand als bij het inschakelen.

De volgende uitleg is slechts bedoeld als illustratie van het werkingsprincipe.

```
10 OPEN 1,4
20 PRINT#1,"A"
```

Dit moet resulteren in het printen van de letter A door een printer met device-nummer 4. Acties roept u op door middel van regel 20. Van de instructie OPEN merkt de printer nog niets omdat hiermee (anders dan bij de diskdrive) geen bestandsnaam verbonden is die doorgegeven zou moeten worden. De getallen die in de onderstaande tekst boven de beschrijvingen staan, hebben betrekking op de getallen in de tijdsdiagrammen boven de X-as.

2.1.2 Adressering

1

ATN wordt laag om aan te geven dat er een instructie volgt.

2

Als reactie daarop zet het apparaat de datalijn binnen 1 milliseconde laag. De computer signaleert naar aanleiding daarvan bijvoorbeeld CLK = high. Dat betekent dat er een byte klaarstaat.

3

Zolang het apparaat niet gereed is om data op te nemen, houdt het de DATA-lijn laag. Wanneer het wel gereed is, wordt DATA = hoog.

3-4

Nu komt het device-nummer via de DATA-lijn: bij elke puls verschijnt er op de lijn CLK een bit. De bits die op die manier overgestuurd zijn, vormen samen de bytewaarde 36 (\$24). Hierin is het device-nummer (4) en de verlangde LISTEN-opdracht (op ontvangst overschakelen) opgenomen.

5

De printer moet de DATA-lijn nu weer binnen 1 milliseconde laag maken. Ten eerste is dit nodig om de ontvangst te bevestigen, ten tweede moet het zenden van volgende data worden verhinderd tot het apparaat gereed is voor ontvangst. Zoals U ziet heeft de DATA-lijn drie taken:

- 1 Transmissie van de data
- 2 Het stoppen van de data
- 3 Het bevestigen van de data-ontvangst

2.1.3 Gegevensoverdracht

6

De overdracht van data kan beginnen als de zender (talker) CLK hoog maakt. Daarmee geeft hij aan dat er een byte voor overdracht klaar staat.

7

Door de DATA-lijn hoog te zetten geeft de ontvanger aan dat hij gereed is.

8-9

Bij elke CLK-puls wordt er een bit door de DATA gezonden.

9

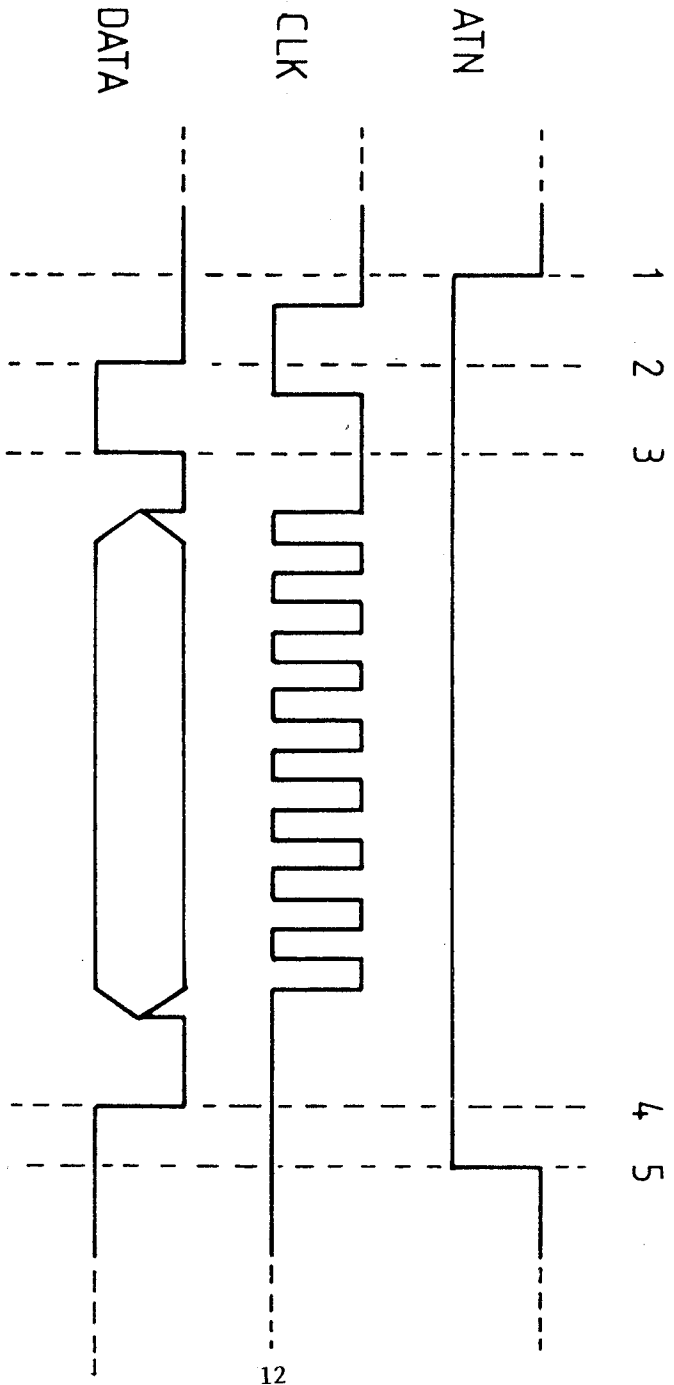
De complete byte wordt door de ontvanger (of listener) met DATA low bevestigd.

11-12

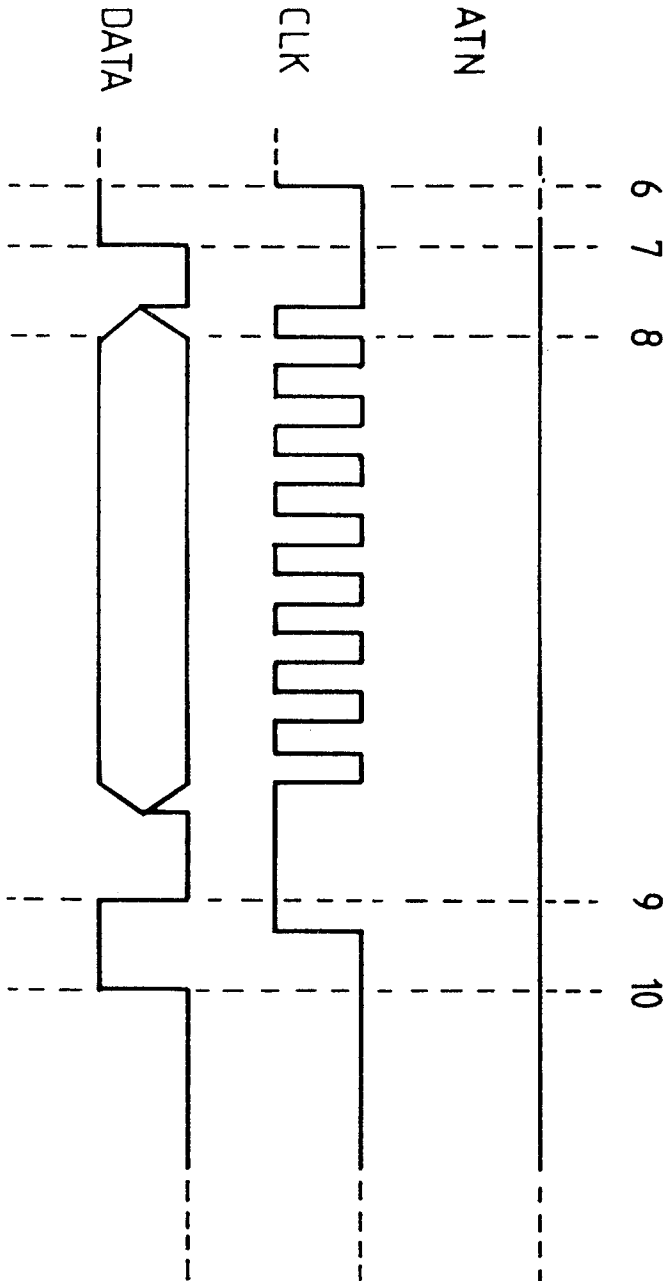
Op dezelfde manier volgt nu de afsluitende CHR\$(13).

Nu krijgt de lijn CLK nog de taak om de laatste byte van een data-overdracht te kenmerken, opdat de ontvanger alvast met de verwerking kan beginnen.

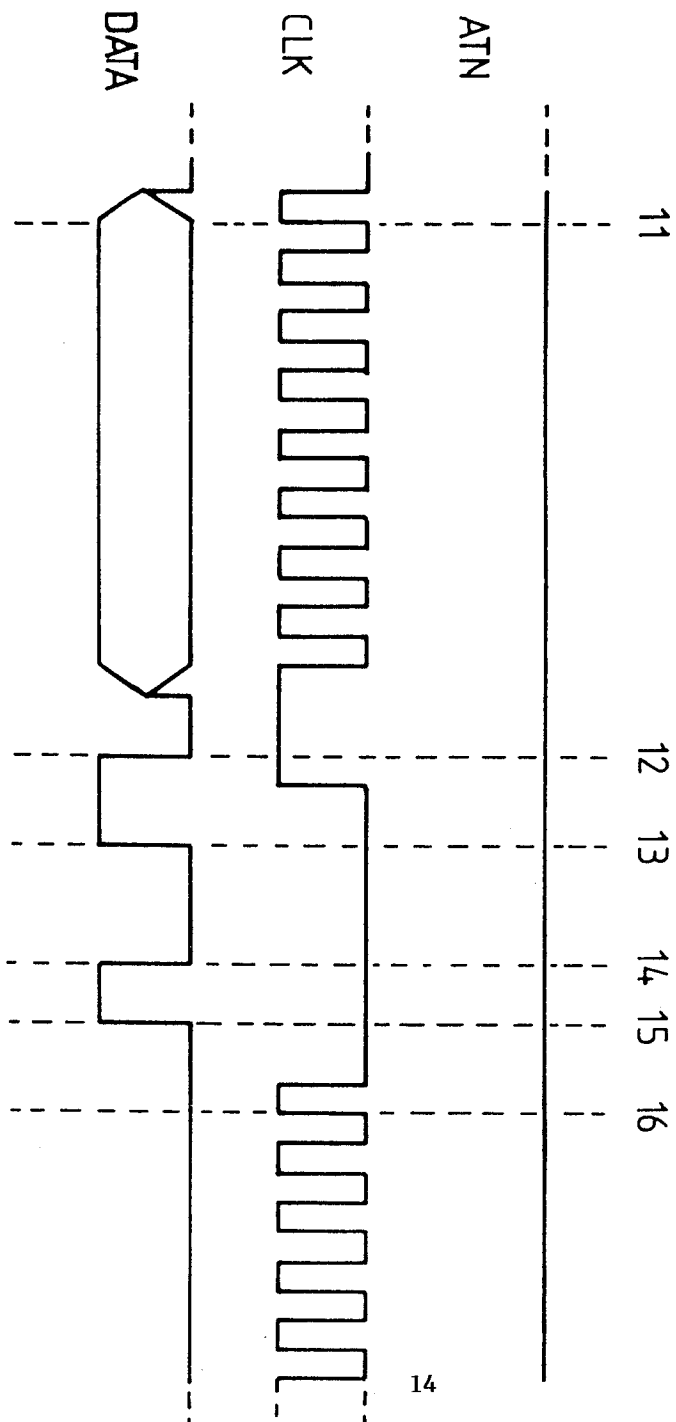
ADRESSERING



OVERDRACHT VAN GEGEVENS



DE LAATSTE DATABYTE



13-14

Tot nu toe gold de volgende afspraak: als de ontvanger (listener) DATA hoog maakt, leidt de zender (talker) binnen 0.2 milliseconde met behulp van CLK = laag de overdracht van gegevens in. Maar in het voorbeeld blijft CLK langer hoog en dat betekent dat de byte die net verzonden werd de laatste was. Ook deze toestand moet de ontvanger bevestigen door DATA = laag te maken.

15

Zodra DATA weer hoog wordt, zorgt de zender voor de transmissie van een lege byte (CHR\$(0) vanaf punt 16). Daarna keert de bus in de rusttoestand terug.

De adressering wordt ongedaan gemaakt zoals in punt 1-5 beschreven, waarbij de databyte de waarde 63 (\$3F) heeft. Dat is de zogenaamde LISTEN-instructie en dat betekent dat alle apparaten die tot nu toe als ontvangers fungeerden weer van de bus kunnen worden losgekoppeld. Er volgen verder geen data meer. Zoals u ziet, werkt dit uitstekend. Een groot nadeel van de seriële bus ten opzichte van de parallelle bus (de veel geringere snelheid) wordt voor een deel opgevangen door de verkorte wachttijden (1 milliseconde in plaats van 64 milliseconden).

2.2 De Centronics interface

De aansluiting op de computer die bij printers het meest wordt toegepast, is gebaseerd op een idee van Centronics. Deze uiterst zekere manier om gegevens over te dragen werd door nagenoeg alle andere fabrikanten van printers overgenomen en min of meer volledig in hun eigen produkten verwerkt. Voor de gebruiker is het natuurlijk een groot voordeel dat printers met deze interface doorgaans het beste en goedkoopste zijn. Het enige probleem is dan nog: hoe sluit ik zo'n apparaat op de Commodore 64 aan? De meest voordelige manier is zeer uitvoerig beschreven in het boek "TIPS & TRUCS voor de Commodore 64". U heeft dan alleen maar een verbindingskabel nodig om de printer op de gebruikerspoort aan te sluiten en verder nog wat software. Met de volgende beschrijving van de aansluiting zult u weinig moeilijkheden ondervinden bij het fabriceren van een dergelijke aansluiting, vooral als u ongeveer weet wat men met de gebruikerspoort kan doen. Een wat duurdere, maar ook meer gerieflijke methode is de toepassing van een interface waarmee men de IEC-bus van de Commodore 64 aan de printer kan aanpassen. Deze oplossing is alleen te gebruiken als

u zich beperkt tot de weergave van alfanumerieke tekens. De talrijke bijzondere tekens en stuurtekens die de Commodore 64 kent, kunt u met een dergelijke configuratie niet reproduceren, althans als die tekens niet tot de karakterset van de printer behoren. Voor de rest is deze methode zeer bruikbaar, vooral omdat u helemaal geen driver-software nodig hebt. Wat het BASIC betreft, kan de printer als een gewoon Commodore apparaat worden aangesproken. Als een verandering van de lettercodes wordt gewenst, moet u in paragraaf 3.3 de uitleg over het secundaire adres 1 nalezen. Een mooie oplossing in het kader van de prijs/prestatieverhouding biedt een inbouwinterface voor de Epson printer. Met zo'n interface (in paragraaf 3.3 beschreven) slaat u meteen meer vliegen in een klap. Ten eerste wordt hij, net als de Commodore printer, gewoon op de IEC-bus aangesloten. Hierdoor wordt het gebruik van BASIC wat eenvoudiger. Ten tweede is zo'n interface in de printer ingebouwd, wat een los slingerend kastje overbodig maakt. Ten derde, en nu komt het: met behulp van deze interface kan de printer - zonder uitzonderingen - alle speciale tekens en stuurtekens weergeven. Bovendien is het apparaat grotendeels op de printer VC-1525 afgestemd zodat de standaard-software die voor deze printer is geschreven zonder problemen zal werken. Om u in de gelegenheid te stellen hierover zelf een oordeel te vormen, komen we tot de kern van de zaak, de Centronics interface. Deze is in de regel als 36-polige Cinch-stekkerverbinding uitgevoerd. Van alle beschikbare lijnen die hij heeft, wordt maar de helft gebruikt om signalen door te geven. Slechts 11 lijnen zijn voor de gegevensoverdracht noodzakelijk. De overige worden gebruikt om bepaalde toestanden in de printer te signaleren en worden ook niet door alle fabrikanten in de stekker verwerkt. De pinconfiguratie van de Centronics bus is als volgt:

- 1 STROBE In rusttoestand is deze lijn hoog. Een lage puls van tenminste 1 milliseconde laat de printer weten dat er zich een geldige databyte op de lijn bevindt.
- 2-9 DATA 0-7. Via deze lijnen wordt aan de printer een byte doorgegeven.
- 10 ACKNLG Met een lage puls van ongeveer 1 milliseconde bevestigt de printer de ontvangst van de databyte. De zender kan nu een nieuwe byte aan de datalijnen doorgeven. (Maar hij kan er nog geen STROBE achteraan sturen!)

Deze lijn is niet echt nodig voor een foutloze sturing van een datastroom. De zogenaamde pure BUSY-sturing is op zich voldoende. Het BUSY-sigitaal dat onder 11 wordt beschreven, is veel belangrijker

dan ACKNLG. Om die reden maken de meeste interfaces geen gebruik van ACKNLG.

- 11 BUSY De printer houdt deze lijn hoog zolang hij niet in staat is een nieuwe databyte op te nemen. Hij kan bijvoorbeeld bezig zijn met het uitprinten van een tevoorren gebufferde regel, het papier kan op zijn, de printer is OFFLINE enzovoort. De zender (talker) zou met STROBE data aan de printer mogen doorgeven als de BUSY-lijn laag is.

- 12 PE Als deze lijn hoog is, signaleert de printer dat het papier op is.

- 16 GND. Dit is het referentiesignaal voor alle signalen.

- 18 +5V Hiermee kunnen externe interfaces van spanning worden voorzien. Niet alle printers hebben deze stroomvoorziening.

- 31 INIT Door een low-puls van tenminste 50 milliseconde op deze lijn wordt de printer in de basistoestand gezet (dezelfde toestand als wanneer hij net ingeschakeld is).

- 32 ERROR Met een lage puls op deze lijn deelt de printer mee dat hij geen papier meer heeft, dat hij OFFLINE is, of dat er een andere fout is opgetreden. Uit de combinatie van de signalen PE en ERROR kunt u de oorzaak voor BUSY afleiden en in de programmatuur verwerken - althans, als u die signalen in uw computer kunt krijgen. Hiervan wordt zelden gebruikt gemaakt.

- 36 SLCT IN Pas nadat er een lage puls op deze lijn is geweest, kan de printer zijn werk doen. Gewoonlijk kan deze puls ook binnen in het apparaat met behulp van een schakelaar opgewekt worden. Mocht uw printer het een keer helemaal laten afweten dan moet u daar maar eens een kijkje gaan nemen. Het is mogelijk dat u - bij het omzetten van een andere schakelaar - ook de genoemde schakelaar per ongeluk hebt omgezet.

De lijnen die we niet genoemd hebben, hebben geen functie, liggen aan massa, of vervullen andere, specifiek voor het apparaat bedoelde taken. De handleiding van printer zal daarover uitsluitsel geven.

Het is zonder meer mogelijk het apparaat alleen met de lijnen 1-9, 11 en 16 te laten werken. De andere lijnen zijn waarschijnlijk niet in alle modellen in dezelfde mate aanwezig.

Het tijdsdiagram aan het einde van deze paragraaf verduidelijkt het verloop van de signalen op de aansluitingen van de interface.

1

Als BUSY = laag kan er een byte op de datalijn worden gezet. Op zijn vroegst na 0.5 milliseconde mag

2

het STROBE-signaal worden geactiveerd. Dit signaal moet tenminste 1 milliseconde duren. BUSY = high geeft aan dat de byte is gearriveerd.

3

Na het STROBE-signaal moet er tot tijdstip

4

nog minstens 0.5 milliseconde verstrijken. Pas nu mogen de datalijnen hun toestand veranderen.

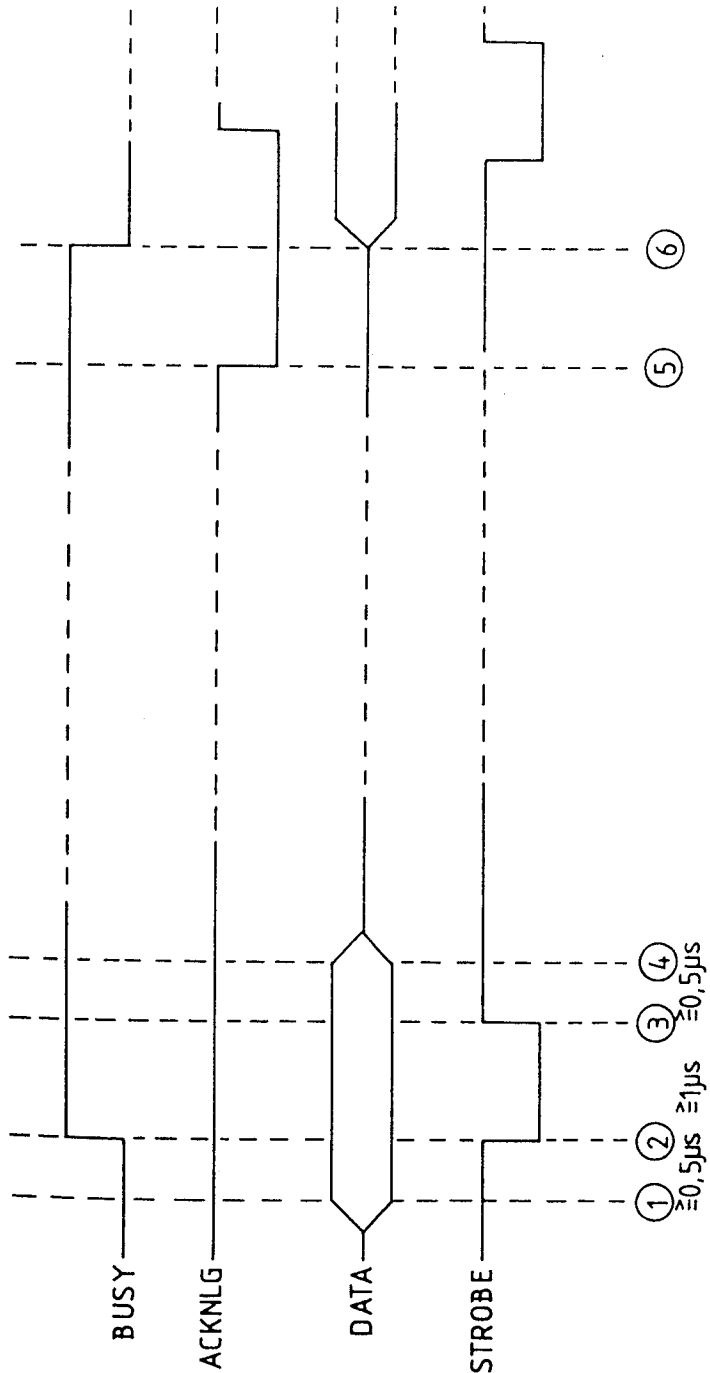
5

De tijd die tot nu toe nodig was, kan sterk variëren. Deze hangt af van wat de printer te doen heeft - of hij alleen het ontvangen teken moet bufferen, of dat dit het laatste teken van een regel was waarna de gehele regel moet worden geprint.

6

Nu kunnen er al nieuwe data op de lijnen worden gezet. Als BUSY = laag kan ook de STROBE-lijn weer geactiveerd worden. Het proces begint dan opnieuw.

TIJDSDIAGRAM VAN EEN CENTRONICS INTERFACE



2.3 De V24 interface

Hierbij gaat het eveneens om een zeer gebruikelijk verbindingssysteem tussen twee apparaten. De interface is universeel voor de datatransmissie over lange afstand omdat zij zo betrouwbaar is. Doorgaans werkt de V24 in twee richtingen, dat wil zeggen dat de data naar beide kanten kunnen stromen. Als u van een printer gebruik maakt, is er natuurlijk slechts transmissie naar buiten noodzakelijk. Toch kunnen bepaalde stuurlijnen ook in tegengestelde richting lopen, net als bij de zojuist besproken Centronics interface. De V 24 is wijdverspreid. De bediening van deze interface levert bij gebruik van de RS-232 cartridge van Commodore in principe geen problemen op, want het bedrijfssysteem van de Commodore 64 steunt deze optie. Maar het kiezen van de parameters en de aansluiting zijn voor de leek vrij moeilijk. Het probleem bestaat uit twee delen. Ten eerste gaat het om het kiezen van de juiste transmissie-parameters (hier komen we later op terug). Een tweede moeilijkheid is dat twee apparaten, voorzien van deze aansluiting, vaak niet met een eenvoudig kabeltje met elkaar kunnen worden verbonden. De oplossing van dit tweede probleem zullen we ook geven. We willen niet dat u het bijltje er bij neergooit als de boel niet werkt en u niet weet of de fout nu in de kabel zit of in de logica. De volgende uiteenzetting heeft betrekking op de RS-232 cartridge VC-1011 van Commodore. U zult waarschijnlijk alleen hiermee te maken krijgen. De aansluiting bestaat uit een 25-polige stekkerverbinding van het type D-subminiatur. De connector is in de winkel verkrijgbaar. De volgende pinbeschrijving van deze verbinding moet u nauwkeurig bestuderen. Het is vooral van belang de richting van de gegevens te onthouden. U moet voorkomen dat u bij het aansluiten van een printer bijvoorbeeld twee signaaluitgangen met elkaar verbindt. Normaal gesproken levert dit geen schade op, maar u zult misschien vergeefs zoeken naar de oorzaak van de mislukte verbinding. Ook daarvoor geven we u enkele tips. Laten we beginnen met de pinbeschrijving van de RS-232 cartridge. Hierbij komen alleen de relevante signalen aan de orde. De pennen die niet besproken worden, moet u bij de bedrading vrijlaten.

- 2 TD. Transmitted Data. Signaaluitgang. Via deze leiding worden de nuttige data bit-serieel uitgevoerd.
- 3 RD. Received Data. Signaalingang. De data die door het aangesloten apparaat worden gezonden, komen hier naar binnen. Bij het aansluiten van een printer wordt deze leiding meestal niet gebruikt.

- 4 RTS. Request To Send. Signaaluitgang. Deze leiding wordt geactiveerd (>3V) wanneer er een adres via de interface wordt verwacht. Meestal is dit bij het gebruik van de printer niet het geval. Als 3 draden worden gebruikt, is dit signaal steeds actief.
- 5 CTS. Clear To Send. Signaalingang. Alleen wanneer deze leiding het aangesloten apparaat activeert (>3V) kan de Commodore 64 data zenden. Bij het gebruik van de printer moet deze leiding beslist door het aangesloten apparaat worden bediend om de datastroom te sturen omdat er anders tekens verloren zouden kunnen gaan. Een uitzondering vormt de 3-draads variant (zie onder) bij extreem lage transmissiesnelheden.
- 6 DSR. Data Set Ready. Signaalingang. Het bedrijfssysteem van de Commodore 64 leest hieraan af dat er gegevens klaarstaan: er wordt gecontroleerd of een apparaat op de interface aangesloten en ingeschakeld is. Een functie, of die nu zenden of ontvangen is, is alleen mogelijk als dit signaal een puls van >3v geeft. Een uitzondering is ook hier weer het gebruik van de 3-draads variant.
- 7 GND. Massa. Referentiesignaal voor alle signalen.
- 8 DCD. Data Carrier Detected. Signaalingang. Dit signaal heeft ongeveer dezelfde betekenis als 6. U kunt de DCD in de stekker het beste met pin 20 verbinden.
- 20 DTR. Data Terminal Ready. Signaaluitgang. Deze leiding is de tegenhanger van 6. Bij de eerste OPEN-instructie door device-nummer 2 wordt dit signaal geactiveerd.

De transmissieparameters worden voor wat de computer betreft in de OPEN-instructie vastlegd. Voorzover ze de printer betreffen, worden ze doorgaans met behulp van schakelaars geregeld. Allereerst moet u erop letten dat deze parameters aan beide kanten met elkaar overeenkomen. Als parameters komen in aanmerking:

- 1 De baudrate

Deze eenheid geeft de transmissiesnelheid in bits per seconde weer. Dit is waarschijnlijk de belangrijkste parameter want als de baudrate niet aan beide kanten overeenstemt, werkt de zaak helemaal niet.

- 2 Het aantal databits

Deze parameter bepaalt uit hoeveel bits een byte bestaat. Bij het aansluiten van een printer kunnen dat er 7 of 8 zijn.

3 Het aantal stopbits

Gewoonlijk komt hier slechts een 1 of een 2 in aanmerking. Deze parameter kunt u naar eigen idee invullen, maar u moet wel rekening blijven houden met wat het aangesloten apparaat vereist.

4 Pariteit

Hier hebt u de keuze tussen geen, even of oneven pariteit. De eerste optie bespaart problemen bij 8 databits en u hoeft dan ook niet meer op zoek te gaan naar fouten in de overdracht. Pariteit betekent het volgende: een databyte bestaat - zoals u weet - uit 0-bits en 1-bits. Ter controle of een byte compleet overgebracht werd, voegt men er nog een extra bit aan toe: de pariteitsbit. Deze geeft aan of het aantal 1-bits even of oneven is, afhankelijk van de instelling van de parameter. Aan de ontvangstkant worden de 1-bits geteld en met de parameter vergeleken. Als er geen overeenstemming bestaat, is er sprake van een transmissiefout die de printer afstraft door een teken te negeren. Werkt de boel niet, hoewel u alles op de juiste wijze hebt ingesteld en met elkaar hebt verbonden, dan speelt de pariteit u wellicht parten. Controleer dat nog maar eens en schakel de pariteitscontrole gewoon uit.

Om de instelling van de computerparameters wat eenvoudiger te maken, hebben we aan enkele variabelen de meest gebruikelijke instellingen gekoppeld en datgene wat overbodig is weggelaten. Meer gedetailleerde informatie vindt u in het boek "64 INTERN" uit de serie van DATA BECKER/NEDERLANDS*.

1 baudrate: variabele br=

50	1
110	3
150	5
300	6
600	7
1200	8
2400	10

2 databits: variabele db=

7	32
8	0

3 stopbits: variabele sb=

1	0
2	128

4 handshake: variabele hs=

nee	0
ja	1

Over de variabele hs zullen we het later nog hebben. Nadat u de passende parameters hebt gekozen, ziet de instructie OPEN er zo uit:

```
OPEN 1,2,0,CHR$(b2+db+sb)+CHR$(hs)
```

Om een kabelverbinding te maken, moet u het volgende doen: u verbindt pin 7 van beide connectors (van de printer en van de computer) met elkaar. Daarna moet u in de computer en de printer de pinnen 4-5 en 6-8-20 met elkaar verbinden. Door de handshake kort te sluiten simuleert u permanent de toestand 'voor bedrijf gereed'. Nu moet u in de printerhandleiding opzoeken welke pin de DATA IN is (2 of 3). Deze moet dan met pin 2 van de computer worden verbonden. Als ook de andere parameters kloppen (denk eraan: geen pariteitscontrole, in Engelse handleidingen vaak 'parity disable' genoemd) kunnen we van start gaan. Nadat de instructies OPEN en PRINT zijn gegeven, moet het apparaat gaan printen. Bij continu printen ziet u vervolgens dat er aan het begin van elke regel een paar tekens ontbreken. Dit is te wijten aan het feit dat we de printer nog niet de mogelijkheid hebben gegeven de datastroom te stoppen als hij even geen tekens kan opnemen. U moet proberen uit te vinden welke lijn de printer gebruikt om deze toestand te signaleren. Stel variabele hs voor de OPEN-instructie in op 1. Open brug 4-5 aan de computer en verbind pin 5 van de computer met brug 4-5 aan de printer en met de brugverbinding 6-8-20. In een van deze gevallen moeten de data in elke regel volledig zijn. Als hs = 0 is, worden de handshake-lijnen niet gebruikt en gaat de computer door met zenden. Deze procedure is alleen mogelijk bij extreem lage baudrates. De printer kan dan nog het uitprinten van tekens gelijke tred laten houden met de invoer. Deze werkwijze moet toegepast worden bij apparaten zonder inputbuffer of bij apparaten die niet de beschikking hebben over signalen die de datastroom kunnen afremmen. Wat er nu nog kan gebeuren is dat de printer in

hoofdletters afdruckt wat u in kleine letters hebt aangeleverd. Dit kunt u in het programma oplossen. Hierover vindt u meer in paragraaf 3.3, waar het secundaire adres 1 wordt uitgelegd. Het is dus niet zo moeilijk als het lijkt. Het enige probleem is eigenlijk de grote hoeveelheid parameters en lijnen.

3 DE SECUNDAIRE ADRESSEN

Er bestaat haast geen dankbaarder onderwerp om over te schrijven dan de secundaire adressen. Ze kunnen u het leven behoorlijk zuur maken als het gaat om de toepasbaarheid (portability) van de printersoftware. Nu moet u niet denken dat met secundair zoiets als tweederangs wordt bedoeld. De aanduiding secundair hangt samen met de terminologie van de IEC-bus. Er is sprake van een primair adres (het device-nummer) en een secundair adres dat aan het primaire adres wordt toegevoegd. Door het secundaire adres krijgt het aangesloten apparaat instructies, bijvoorbeeld over hoe een doorgestuurde serie data geïnterpreteerd moet worden. Zelfs een eenvoudig apparaat als een printer kent veel interpretatiemogelijkheden die menig programmeur bezighouden. Dat zult u na het lezen van dit hoofdstuk wel begrijpen. Denk nu niet dat bij alle printers het secundaire adres dezelfde functies in werking zet, ook al komen ze uit dezelfde fabriek. Het belangrijkste verschil is dat bij het ene apparaat het secundaire adres 7 alleen betekent dat er geen printdata mogen volgen, terwijl bij een ander apparaat juist het tegendeel geldt. Dit betekent dus dat u software alleen mag gebruiken voor de printer waarvoor die software is gemaakt. Als u dezelfde software toepast op een andere printer zou dit tot een mislukking kunnen leiden. Denkt u daar dus aan bij het samenstellen van het programma. Een tip: gebruik een subroutine voor de printeruitvoer, ook al bestaat die slechts uit 1 PRINT-instructie. Voor andere printers hoeft u de PRINT-instructie maar een klein beetje te wijzigen. Dat geldt ook voor de OPEN-instructie. Definieer daarom het logisch filennummer, het device-nummer en het secundaire adres aan het begin van het programma en grijp daar verderop in het programma op terug. Op die manier slaat u twee vliegen in een klap. U kunt het logisch filennummer en het secundair adres gemakkelijk wijzigen. Verandering van het logisch filennummer kan nuttig zijn als u de ene keer gebruik maakt van een printer met Auto-line feed en de andere keer van een printer zonder Auto-line feed. Het bedrijfssysteem van de Commodore 64 voegt bij logische filenummers >127 een extra line feed aan de printdata toe.

Voorbeeld:

```
10 LF = 1:DA = 4: SA = 7
.
.
100 OPEN LF, DA, SA
.
```

1000 PRINT# LF,

Nu kunt u de waarde in regel 10 eventueel variëren zonder het hele programma op OPEN- en PRINT-instructies af te zoeken. In de volgende paragraaf komen de functies van de secundaire adressen van de meest gangbare printers aan de orde. Aan functies die doorgaans veel fouten opleveren, zullen we extra aandacht besteden.

3.1 VC-1525 / GP-100VC / MPS-801

Deze printers hebben slechts 2 secundaire adressen waarmee bepaald wordt uit welke karakterset de printkarakters gehaald moeten worden. Net zoals de Commodore 64 hebben de printers ook twee modi.

Secundair adres 0 of geen secundair adres

```
10 OPEN 1,4,0
of
10 OPEN 1,4
20 PRINT#1,"abcABC"
```

In deze modus zal de printer de kleine letters als hoofdletters afdrukken en de hoofdletters als grafische karakters. Deze modus kunt u tot aan de eerstvolgende <RETURN> veranderen door er het stuurteken voor "cursor down" (CHR\$(17)) aan vooraf te laten gaan:

```
30 PRINT#1,CHR$(17)"abcABC"
```

Met deze regel wordt het schriftbeeld van de hoofdletters en kleine letters op de juiste wijze geprint.

Secundair adres 7

```
10 OPEN 1,4,7
20 PRINT#1,"abcABC"
```

Hoofdletters en kleine letters worden nu juist weergegeven. Ook

deze modus is tijdelijk te veranderen door het stuurteken voor "cursor up" (CHR\$(145)) ervoor te plaatsen:

```
30 PRINT#1,CHR$(145)"abcABC"
```

Hierbij verschijnen de kleine letters als hoofdletters en worden de hoofdletters grafische tekens. Houd er rekening mee dat een wijziging van de modus door CHR\$(17) of CHR\$(145) steeds alleen voor de betreffende PRINT-instructie geldt. Natuurlijk kunnen beide modi ook binnen een regel afgewisseld worden. Dat gaat zo:

```
10 OPEN 1,4  
20 PRINT#1,"abc"CHR$(17)"ABC"
```

of zo:

```
10 OPEN 1,4: OPEN 2,4,7  
20 PRINT#1,"abc";:PRINT#2, "ABC"
```

Beide versies geven hetzelfde resultaat. Rekening houdend met de compatibiliteit geven we echter de voorkeur aan de eerste versie omdat deze op alle tot nu toe bekende Commodore printers toegepast kan worden.

3.2 VC-1526

Deze printer-plotter heeft heel veel in z'n mars, maar er is haast geen andere die zoveel verwarring heeft gesticht: de apparaten die vanaf december '83 werden geleverd, stemmen qua bediening niet overeen met hun eigen handboek maar wel met dat van de VC-1525. Hierbij de kanttekening dat CHR\$(8) niet pixelgrafiek tot gevolg heeft. Met de volgende handeling stelt u de 1526-modus in: na het openen van het apparaat moet u pin 16 van het IC op inbouwpositie U4D aan massa leggen. Nu kunt u uitgaan van de bijgeleverde handleiding. U kunt aan de hand van de zelftest controleren of uw apparaat voor dit doel geschikt is. In de eerste regel van de drukproef vindt u het versienummer van het bedrijfssysteem van de printer. Dit nummer moet tenminste 7 zijn. Voor de 1526-modus zijn er de volgende secundaire adressen:

Secundair adres 0 of geen secundair adres

De printdata worden weergegeven zoals ze via het secundaire adres 7 of 8 (zie aldaar) vastgelegd zijn. De modus kan tijdelijk worden veranderd door CHR\$(17) (kleine letters/hoofdletters) of CHR\$(145) (hoofdletters/grafische karakters) voorop te plaatsen.

Secundair adres 1

Dit adres levert de te formateren gegevens overeenkomstig de formaataanwijzing die daarvoor met het secundaire adres 2 doorgegeven werd. De keuze van de karakterset verloopt zoals daarvoor met het secundaire adres 7 of 8 is vastgelegd. Ook hierbij is de keuze binnen een regellengte te veranderen. Denk eraan dat de gegevens die tot het ene veld behoren door CHR\$(29) van het andere veld gescheiden moeten worden. Voorbeelden hiervan zijn in de handleiding van de printer te vinden.

Secundair adres 2

Met dit secundaire adres worden de formaataanwijzingen doorgegeven. Het bepaalt de plaats op het papier van de gegevens die op grond van secundair adres 1 zijn verzonden. Bijzonder praktisch is de mogelijkheid ook numerieke waarden geformateerd uit te voeren, dat wil zeggen dat decimale punten in dezelfde kolom worden geplaatst. Verdere informatie vindt u in de handleiding.

Secundair adres 3

Met dit adres kunt u aangeven hoeveel regels er per pagina moeten worden geprint voordat er een line feed van zes regels wordt gegeven. Dit kan belangrijk zijn als u wilt voorkomen dat er bij het gebruik van kettingformulieren over de perforatie geprint wordt. Hiermee krijgt u boven- en onderaan een pagina steeds drie blanco regels. Natuurlijk moet u ervoor zorgen dat de gegeven waarde overeenkomt met de maximale regelcapaciteit van het blad minus 6. Concreet: als u kettingformulieren van 12" gebruikt (het meest gangbare formaat) en deze met een dichtheid van 6 regels per inch bedrukt, passen er 72 regels op een blad. Daarvan moet u 6 regels aftrekken: 3 voor de bovenste en 3 voor de onderste rand. De instructie hiervoor is:

```
10 OPEN 1,4,3
20 PRINT#1,CHR$(66)
30 CLOSE 1
```

Houd er rekening mee dat deze maatregel alleen niet het gewenste resultaat oplevert. De paginering moet eerst met

```
10 OPEN 1,4
.
.
.
500 PRINT#1 CHR$(147)CHR$(141);
```

worden geactiveerd.

Secundair adres 4

Secundair adres 4 stelt niet veel voor, want via dit adres wordt de diagnoseroutine van de printer ingeschakeld. Dat werkt zo:

```
10 OPEN 1,4,4:PRINT#1:CLOSE 1
```

Hiermee krijgt u bij programmeerfouten die de printer betreffen een gedrukte foutmelding.

Secundair adres 5

Met dit secundaire adres kunt u zelf een karakter definiëren in een 8x8-matrix. U kunt dit karakter met CHR\$(254) oproepen. Zo kunt u (al gaat het wat moeizaam) ook een grafische voorstelling van pixels (matrixpuntjes) maken. Zie paragraaf 5.3.2.

Secundair adres 6

Hiermee kunt u de afstand tussen twee regels in stappen van 1/216" bepalen. Het bereik van toegestane waarden is 1-127. Denk erom dat een normale puntafstand overeenkomt met drie stappen. Voor een regel zonder gaten moet u de volgende instructie gebruiken:

```
10 OPEN 1,4,6:PRINT#1,CHR$(24):CLOSE 1
```

Secundair adres 7

Dit is de instelling voor kleine letters/hoofdletters. Om deze modus in te kunnen schakelen gebruikt u de instructie

```
10 OPEN 1,4,7:PRINT#1:CLOSE 1
```

Met dit adres mogen geen data worden gestuurd. Gegevens lopen uitsluitend via de secundaire adressen 0 of 1.

Secundair adres 8

Dit adres is de tegenhanger van het secundaire adres 7.

```
10 OPEN 1,4,8:PRINT#1:CLOSE 1
```

schakelt de printer over op de hoofdletter/grafiekmodus.

Secundair adres 9

```
10 OPEN 1,4,9:PRINT#1:CLOSE 1
```

schakelt de diagnoseroutine (die door het secundaire adres 4 was geactiveerd) weer uit.

Secundair adres 10

```
10 OPEN 1,4,10:PRINT#1:CLOSE 1
```

zet de printer in de uitgangspositie terug. Deze mogelijkheid kunt u gebruiken als u in het verloop van een programma meerdere instellingen hebt veranderd en ze niet op dezelfde omslachtige manier ongedaan wilt maken.

3.3 Epson printer met DATA BECKER interface

Omdat deze interface wijdverspreid is, willen we ook hieraan een paar woorden wijden. In principe komt het gebruik van de secundaire adressen met dat van de VC-1525 overeen. Er komen 3 secundaire adressen bij.

Secundair adres 1

Hier moeten we de zaak iets uitvoeriger bespreken: alle gangbare printers met uitzondering van de Commodore hebben een karaktercodering volgens de ASCII-norm. In deze internationaal

erkende standaardcode is de positie van de letters en bijzondere tekens vastgelegd. Zo ligt bijvoorbeeld bij CHR\$(65) de A en bij CHR\$(97) de a. Bij de Commodore (in de kleine letter/hoofdletter-modus) ligt de a echter bij CHR\$(65) en de A bij CHR\$(193). De problemen die hierdoor ongetwijfeld zouden rijzen, zijn opgelost door het invoeren van het secundaire adres 2 (zie aldaar). Het secundaire adres 1 dat hier besproken wordt, laat alle tekens die van de computer komen zonder meer naar de printer doorstromen. U zou daarom nooit in staat zijn kleine letters met een normale tekst op de printer te realiseren. Het secundaire adres 1 is dan ook niet bedoeld om tekst over te brengen maar dient voor de instelling van de vele mogelijkheden die EPSON printers bezitten. U moet daarom nooit die letters gebruiken die in de stuursequenties zijn aangegeven, maar uit de appendix van de handleiding de juiste CHR\$ halen. Als u bijvoorbeeld de kantlijn op positie 10 wilt zetten, vindt u in de handleiding dat daarmee de letter "1" korrespondeert. In de ASCII-tabel vindt u hiervoor de waarde 108. Uw instructie moet er dus zo uitzien:

```
10 OPEN 1,4,1
20 PRINT#1CHR$(27)CHR$(108)CHR$(10);
30 CLOSE 1
```

Als u in plaats van CHR\$(108) de letter "1" invoert, komt u in de grafische modus terecht. Toetst u een 1 in op de Commodore 64 dan leidt dat tot een L op de printer. Dat komt omdat de Commodore 64 van de ASCII-code afwijkt. De pixelgrafiek moet u eveneens over dit secundaire adres sturen. U moet er even aan denken dat u een logisch filenummer >127 moet gebruiken als u toch een normale regelafdruk via dit secundaire adres wilt realiseren. Anders wordt de automatische line feed onderdrukt.

Secundair adres 2

Het hierboven genoemde dilemma is met behulp van dit secundaire adres opgelost. Alle data die via het secundaire adres 2 aankomen, worden met behulp van een tabel automatisch zo veranderd dat een a op het beeldscherm overeenstemt met een a op de printer. Dit geldt ook voor het secundaire adres 7, maar daar wordt het originele schriftbeeld van de Commodore geleverd en niet iedereen is daarover even enthousiast. Bij het secundaire adres 2 wordt het veel mooiere EPSON schriftbeeld gebruikt waarmee ook het trema gemaakt kan worden. Hierbij geldt de beperking dat op het secundaire adres 2 de grafische tekens en stuurtekens van de Commodore niet kunnen worden geprint. U moet bij dit adres niet met de pixelgrafiek werken want ook de puntpatronen worden door de tabel veranderd, waardoor het resultaat misschien niet zo zou zijn als u het wenst.

Secundair adres 3

Dit adres bestaat volgens de informatie van de fabrikant pas sinds juni '84. Hiermee wordt het secundaire adres dat het laatst actief was, gefixeerd totdat het apparaat wordt uitgeschakeld - om het even via welke andere secundaire adressen de data komen. Dat fixeren gaat met de volgende instructie:

```
10 OPEN 1,4,3:PRINT#1:CLOSE 1
```

De bedoeling hiervan is de gebruikelijke werking van tekstverwerkingsprogramma's te omzeilen. Deze programma's zenden de data gewoonlijk via het secundaire adres 7, wat resulteert in een printeruitvoer in het Commodore schriftbeeld. Als u over een Epson beschikt, wilt u ook het Epson schriftbeeld gebruiken. Daarom moet u voor het laden van de tekstverwerkingsroutine de volgende programmaregels intoetsen:

```
10 OPEN 1,4,2:OPEN 2,4,3  
20 PRINT#1:PRINT#2  
30 CLOSE 1:CLOSE 2
```

Nu is het secundaire adres 2 gefixeerd zolang de printer is ingeschakeld.

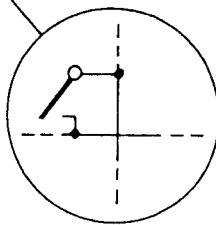
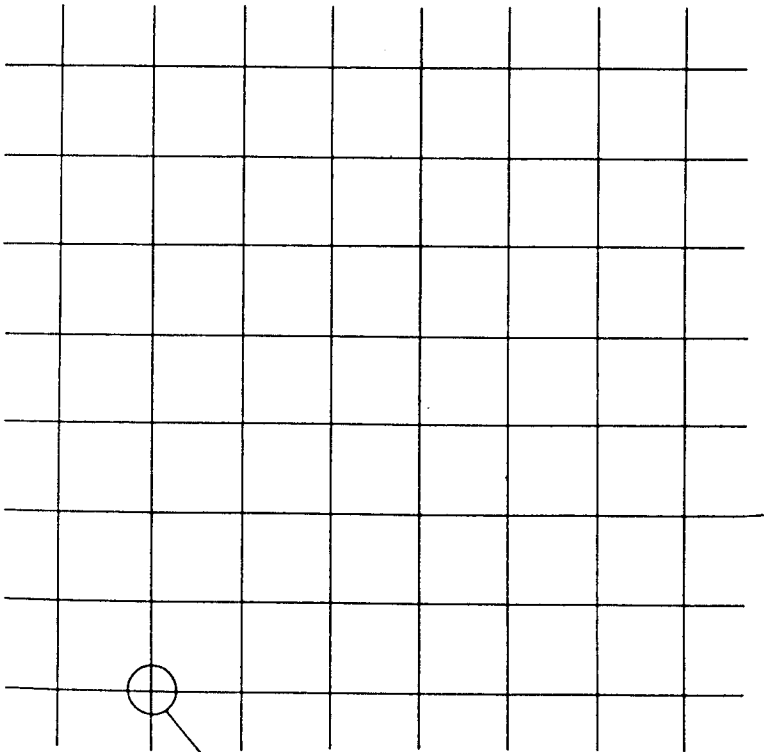
4 SCHRIJFMACHINE AAN DE USER-PORT

Voor alle duidelijkheid: met schrijfmachine bedoelen we een moderne elektrische schrijfmachine. U hoeft niet ontzettend handig te zijn om de hier beschreven schakeling na te bouwen. Het is veel moeilijker de machine aan te sluiten en met een interface te verbinden. We willen deze interessante mogelijkheid toch niet onopgemerkt laten. Allereerst moet u zien uit te vinden of uw machine voor dit doel te gebruiken is. Want ook al is de schakeling nog zo eenvoudig en goedkoop (het materiaal zal niet veel duurder zijn dan f20,-), het is niet geschikt voor elke schrijfmachine. Als u zelf aan de hand van de gebruiksaanwijzing hebt ontdekt dat uw machine van een microprocessor is voorzien, bent u al op de goede weg. U hoeft die microprocessor niet anders te programmeren maar de aanwezigheid ervan duidt erop dat de leidingen van het toetsenbord in de vorm van een X/Y-matrix zijn doorverbonden. En dat hebben we nu juist nodig voor de aansluiting. Hoe komt u er nu achter of dit bij uw machine inderdaad het geval is? Dat is helemaal niet zo moeilijk. U hebt alleen een weerstandsmeter nodig. Dan maakt u de schrijfmachine heel voorzichtig open om bij de contacten van de toetsen te kunnen komen. Voordat u met dit karweitje verder gaat, leggen we u eerst het principe van de toetsenbordmatrix uit. Daarna is het een peuleschil. Een toetsenbordmatrix is opgebouwd als aangegeven op de tekening. Of er een paar lijnen meer of minder zijn, speelt geen enkele rol.

toetsenbordmatrix

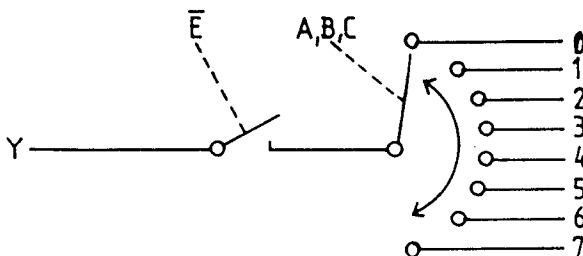
X-lijnen

Y-lijnen



Zoals u ziet, bestaat elk kruispunt op de matrix uit een schakelaar, de toets van de schrijfmachine dus. Het logische systeem of de processor van de schrijfmachine zet een voor een op elke X een spanning. Als er geen toets is ingedrukt, gebeurt er helemaal niets. Als er wel een toets is ingedrukt, gaat de spanning ook op de snijdende Y over. De processor merkt dit op en weet precies welke toets ingedrukt werd. Die toets moet immers op het snijpunt liggen van de X die onder spanning was gezet en de Y waarop de spanning terechtkomt. U moet ervoor zorgen dat de spanning niet hoger dan 5V wordt omdat deze interface daar niet op berekend is. Bij een schrijfmachine die door een microprocessor wordt gestuurd, komen spanningen hoger dan 5V meestal niet voor. Vervolgens moet u de genoemde schakelaars met behulp van de interface simuleren. Het matrixsysteem is daarvoor ideaal want het spaart een heleboel lijnen en andere hardware uit. Stelt u zich eens voor dat u voor elk van de 50 toetsen telkens twee verbindingen moest leggen! Nu hoeven slechts de X- en Y-lijnen afgetapt te worden om erachter te komen waar ze liggen. Zoals op de tekening te zien is, liggen telkens enkele contacten met een kant samen aan een lijn. De verbanden moet u met een Ohmmeter opsporen. Dat betekent dat u elk contact tegelijk met een ander moet meten omdat het niet te verwachten is dat alle toetsen die aan dezelfde lijn liggen ook op het toetsenbord naast elkaar liggen. Dit karweitje zal circa twee uur in beslag nemen. Tenslotte moet u een configuratie hebben gevonden zoals die op de tekening staat afgebeeld. Vergeet niet een notitie van de aangetroffen verbanden te maken. Dat maakt de latere programmering een stuk eenvoudiger. Het is mogelijk dat er geen samenhang bestaat tussen een paar toetsen op dezelfde lijn. Dat zijn de SHIFT-toets of andere stuurtoetsen die niet via de matrix worden opgeroepen. Laten we nu onze schakeling bekijken: de kern bestaat uit twee zogenaamde analoge multiplexers. Op onderstaand schema ziet u hoe u zich de werking dient voor te stellen.

analoge multiplexer: principe van de werking



Bestudeer de afbeelding goed en bekijk dan de volledige schakeling aan het eind van de pragraaf. U ziet nu meteen wat onze bedoeling is. We kunnen namelijk een van de acht X-lijnen met een van acht Y-lijnen verbinden. Welke het zijn, wordt via de userport bepaald. Hierdoor geven we de processor in de schrijfmachine de indruk dat er een toets werd ingedrukt, namelijk degene die op het snijpunt van de lijnen ligt. Om niet voortdurend een paar lijnen onder spanning te laten staan, gebruiken we de ENABLE-ingangen van de multiplexer. Op die manier worden ze slechts voor de duur van een echte druk op de toets gebruikt. Het geheel werkt zo dat de keuze van de X-lijn drie bits van de userport in beslag neemt en de keuze van de Y-lijn drie andere bits. De laatste twee bits bedienen de analoge schakelaar voor de SHIFT-toets en een toets naar keuze. Nadat u alles hebt opgebouwd en aangesloten, kunt u het volgende testprogramma laten draaien:

```

10 PA=56576:PB=56577
20 DA=56578:DB=56579
30 POKE DA,PEEK (DA) OR 4:REM RICHTING BEPALEN
40 POKE DB,255
50 POKE PA,PEEK (PA) OR 4
100 INPUT "X-lijn";X
110 INPUT "Y-lijn";Y
120 INPUT "SHIFT 0=NEE, 1=JA";SH
130 CH=X+8*Y+64*SH:PRINT CH
140 GOSUB 200:GOTO 100
200 POKE PB,CH
210 POKE PA,PEEK (PA) AND 251
220 FOR DL=0 TO 100:NEXT DL
230 POKE PA,PEEK (PA) OR 4
240 RETURN

```

U kunt de gevonden samenhang tussen de verschillende letters en de X-Y-lijnen en SHIFT-toets als volgt noteren: naast de gevonden letter noteert u het getal dat tegelijkertijd op het beeldscherm verschijnt. Met die gegevens en met behulp van het volgende programma maakt u een conversietabel, waarmee u de teksten gemakkelijk kunt printen.

```

10 DIM CH(255):CH=0:FOR I=1 TO 255:CH(I)=0:NEXT I=0
20 GET A$:IF A$="" THEN 20
30 PRINT "BIJ "A$" BEHORENDE GETAL";
40 INPUT A
50 CH(ASC(A$))=A
60 GOTO 20
200 APP=8:REM APP=1 VOOR TAPE
210 IF APP=1 THEN OPEN 1,1,1:GOTO 230
220 OPEN 1,8,8,"KARAKTERS,S,W"

```

```

230 REM GA SCHRIJVEN
240 FOR TEL=0 TO 255
250 PRINT#1,STR$(CH(TEL))
260 NEXT TEL:REM VOLGEND TEKEN
270 CLOSE 1

```

Inlezen

vervang regels

```

210 ... OPEN 1,1,0:GOTO 230
220 OPEN 1,8,8,"KARAKTERS,S,R"
250 INPUT#1,A$:CH(TEL)=ASC(A$)

```

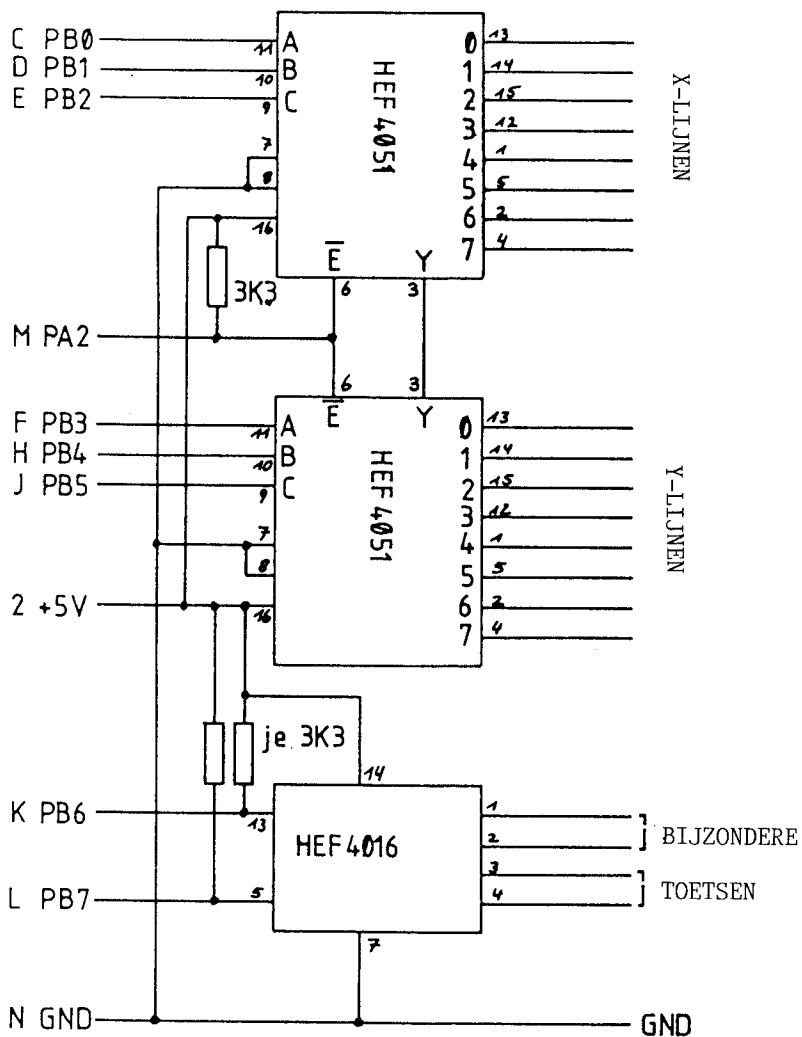
Als u klaar bent, stopt u het programma met STOP en slaat u tabel CH(0-255) op band of schijf op door eenvoudig GOTO 200 in te toetsen. Daarna is het printen van teksten heel eenvoudig: vervang de regels 100-140 uit het eerste programma door de volgende routine:

```

100 FOR L=1 TO LEN(CH$)
110 CH=CH(ASC(MID$(CH$,L,1)))
120 GOSUB 200:NEXT L:RETURN

```

De enige voorwaarde is: uw tekst moet in CHR\$ zijn opgenomen en u moet de routine starten met GOSUB 100. De regels 10 tot 50 moeten een keer aan het begin van uw programma de snelheid trimmen. Zo, dat was dat. We wensen u veel plezier bij het nabouwen. Het schema van de schakelingen is zo eenvoudig dat we er niets meer aan toe te voegen hebben.



5 PROGRAMMA'S VOOR VIC 20 EN COMMODORE 64

In dit hoofdstuk willen we u een paar programma's presenteren die u beslist interessante ideeën aan de hand doen voor de printerprogrammering. Daarbij behandelen we vele toepassingsgebieden. Helaas is het bij zo'n groot aanbod van verschillende printers onmogelijk elk programma voor elke printer te behandelen en hier af te drukken. Hadden we dit wel willen doen, dan had dit boek al snel de omvang van een telefoonboek van een grote stad gekregen. De afgedrukte programma's voor pixelgrafiek leveren over het algemeen voor bezitters van letterwielprinters niet veel op. Maar ook met deze mensen hebben we rekening gehouden en daarom hebben we een paar kleine grafische programma's in dit boek opgenomen, die op letterwielprinters beslist aardige resultaten geven. Op alle belangrijke plaatsen geven we tips hoe u de programma's voor andere printers kunt aanpassen. Als u dus een printer bezit die andere printer-stuurtekens heeft dan in deze paragrafen vermeld staan, kunt u in het bijbehorende printerhandboek een passend stuurinstructie-commando vinden. Of alle programma's met uw printer werken of niet, u kunt in de genoemde programma's vast dingen vinden die voor eigen programmeerdoeleinden geschikt zijn.

5.1 Kleine hulpprogramma's voor dagelijks gebruik

In deze paragraaf worden drie hulpprogramma's besproken, die in deze vorm waarschijnlijk voor alle bezitters van printers interessant zijn. Allereerst laten we u kennis maken met een korte routine die de afbreking van een programma verhindert als de printer niet aanstaat. Dit korte programma kan in elk BASIC-programma worden ingevoegd of daaraan vast worden gekoppeld, waardoor de lastige foutmelding `DEVICE NOT PRESENT` wordt voorkomen. De tweede routine is al wat langer. Het gaat hier om een handig subprogramma voor het formateren van numerieke en alfanumerieke data. Ook het laatste hulpprogramma is een soort geformateerde print-output. Daarbij worden geen getallen of teksten uitgevoerd maar programmalistings, op een overzichtelijke manier. Tussen de instructies worden spaties ingevoegd, bij `FOR-NEXT`-lussen wordt ingesprongen en `IF-THEN`-structuren worden

leesbaar gemaakt. Hiermee worden ook de werkelijk onoverzichtelijke listings leesbaar.

5.1.1 DEVICE NOT PRESENT?

Het BASIC van de Commodore apparaten geeft geen mogelijkheden zoals bijvoorbeeld de ON-ERROR-instructie om een ongewenste afbreking te voorkomen. U kunt syntactische fouten of fouten als delen door nul wel voorkomen door een goede programmeerstijl te gebruiken, maar voor programma's die voor printers of floppy zijn bedoeld, ligt dat anders. Wilt u dan proberen in BASIC fouten te omzeilen (speciaal DEVICE NOT PRESENT) dan moet u uw toevlucht nemen tot trucs. Aan sommige programma's valt heel mooi af te lezen dat deze trucs nog niet algemeen bekend zijn. De samenstellers van de software hebben dan alle moeite gedaan om de programma's te beschermen. Wanneer u bij deze programma's een niet ingeschakelde printer gebruikt, komt u meteen in de directe modus terecht, van waaruit u het programma naar wens kunt listen of veranderen. Een kort BASIC-subprogramma met een paar bytes maakt hier voorgoed een eind aan.

```
100 REM *****PRINTER*****
110 POKE 768,61
120 OPEN 1,4
130 PRINT#1,"";
140 CLOSE 1
150 POKE 768,139
160 IF ST <> -128 THEN 180
170 PRINT"PRINTER INSCHAKELEN"
180 RETURN
200 REM ***** DISKDRIVE *****
210 POKE 768,61
220 OPEN 1,8,15,"I"
230 CLOSE 1
240 POKE 768,139
250 IF ST <> -128 THEN 270
260 PRINT"DISKDRIVE INSCHAKELEN"
270 RETURN
```

Er bestaan weinig verschillen tussen het programma voor de printer en dat voor de floppy. Daarom hebben we de versies in een keer gegeven. Beide routines kunnen met GOSUB worden opgeroepen. Natuurlijk kunt u de uitgevoerde meldingen naar wens aanpassen.

We wijzen erop dat u bij de toepassing van deze routines de BREAK-toets moet uitschakelen, omdat de vector in de regels 150 en 280 betrekking heeft op de inputroutine in de directe modus. De POKE in deze regels zet de vector (die tegelijkertijd ook verantwoordelijk is voor de uitvoer van foutmeldingen) op een RETURN-instructie. Daardoor wordt de uitvoer van foutmeldingen onderdrukt. Bovendien is er na deze POKE geen directe modus meer mogelijk. Als het programma na de eerste POKE wordt onderbroken - door BREAK of door RUN/STOP-RESTORE - laat de computer het afweten. Slechts de "cold reset", dus het uitschakelen en daarna weer inschakelen, kan dan nog enigszins hulp bieden.

5.1.2 Het formateren van numerieke en alfanumerieke data

Iedereen die wel eens heeft geprobeerd een printer-output te programmeren waarbij zowel teksten als getallen mooi geordend onder elkaar afgedrukt moesten worden, zal het BASIC-dialect van de Commodore computer hebben vervloekt. Als gebruikers van andere computers met termen als PRINT USING of andere handige instructies pronken, kan de gebruiker van de Commodore een gevoel van jaloezie niet geheel onderdrukken. Meteen wordt de manual ijverig doorgebladerd, maar het mag niet baten: de zaak ligt niet zo eenvoudig. Maar toen op een goede dag de eerste printers 1526 op de markt verschenen, heerste er grote vreugde en hoopvolle verwachting onder de Commodore-fans. Dit printertype had indrukwekkende en qua bediening zeer eenvoudige mogelijkheden voor geformateerde printeruitvoer. Dat hij toevallig geen grafische optie bezat, nam men op de koop toe. De vreugde was van korte duur. Behalve de positieve eigenschappen die in de manual opgesomd werden, waren er ook negatieve aspecten. Zo kwam het wel eens voor dat de seriële bus compleet geblokkeerd werd. Een zeer lastige zaak, vooral wanneer er juist op dat moment data op de floppy gezet moesten worden. Deze narigheid werd natuurlijk ook door de fabrikant ontdekt, die op korte termijn een uitwisselbare ROM op de markt wilde brengen. Maar dit plan werd eerst weer op de lange baan geschoven om de volgende reden: voor een probleemloze werking van de printer hadden de samenstellers van de software van het printer-bedrijfssysteem wat geheugenruimte nodig. Die was in de bestaande uitvoering blijkbaar niet aanwezig. Dus ging men met de botte bijl te werk om ruimte te creëren. Daarvan werd de geformateerde uitvoer het slachtoffer. Wat dit probleem betreft kunt u kiezen uit de volgende vijf alternatieven. U kunt

- 1 een nieuwe computer met PRINT USING kopen
- 2 een uitbreidingsmogelijkheid voor instructies, bijvoorbeeld MASTER, voor uw computer kopen
- 3 zelf een geschikte routine schrijven
- 4 de afgedrukte routine gebruiken
- 5 helemaal afzien van de geformateerde uitvoer.

De goedkoopste en meest succesvolle methode is het vierde alternatief. Voordat we de routine bespreken, bekijken we welke mogelijkheden het Commodore-BASIC en de printers bieden als het om de sturing van de uitvoer gaat. Alle printers hebben de eigenschap gemeen dat ze karakters zo weergeven als ze ingevoerd worden. De uitvoer van een brief is de meest probleemloze manier

van printeruitvoer, het uitrichten van de kantlijn buiten beschouwing gelaten. Heel wat anders is het printen van rekeningen. Daarbij moeten het tekstgedeelte en het getallengedeelte worden geordend: vaak komt het tekstgedeelte links en komen de getallen rechts te staan. De computer zelf is al voorzien van een mogelijkheid om de weergave te manipuleren: de TAB-functie. Helaas levert deze functie bij een printeruitvoer op het beeldscherm andere resultaten op dan op de printer. Misschien is u al opgevallen dat bijvoorbeeld de regel

```
PRINT"hallo"tab(10)"hoe gaat het"
```

op het beeldscherm heel netjes

```
hallo      hoe gaat het
```

wordt, maar op de printer

```
hallo      hoe gaat het
```

Hoe is dat mogelijk? Heel eenvoudig: het bedrijfssysteem verandert TAB(x) bij de uitvoer op een ander apparaat dan een beeldscherm in SPC(x). TAB betekent immers een absolute positie, gemeten vanaf de linkerrand, terwijl SPC de afstand van de actuele positie van de cursor of printkop bepaalt. Afhankelijk van de lengte van de eerste tekst verschuift het begin van de tweede tekst ook. Wat moet er dus gebeuren? Er bestaat een mogelijkheid, maar die is slechts op een paar Commodore printers van toepassing: door carriage return zonder line feed de absolute positie van de linkerrand bepalen en het tweede tekstgedeelte printen. Bij sommige Commodore printers bewerkstelligt CHR\$(141) het teruglopen van de printkop zonder papiertransport. Een concreet voorbeeld om een regel van een rekening te printen:

```
10 OPEN 1,4
```

```
20 PRINT#1,"VERLENGSNOER"CHR$(141)SPC(40)"FL 6.45"
```

De tekst wordt nu links geprint en de prijs vanaf tabulatorstop 40. Om dit gedaan te krijgen, hebben we de volgende truc toegepast: we hebben eerst de tekst geprint, vervolgens een carriage return zonder line feed gegeven en tenslotte een spatieruimte van 40 tekens. Daarna wordt de prijs geprint. U ziet het: bij een uitdraai kunt u de werking van een echte TAB laten uitvoeren door er CHR\$(141) aan vooraf te laten gaan. Printers die deze mogelijkheid niet hebben (zoals bijvoorbeeld de 1525), hebben een eigen stuurcode om de positie van de printkop te bepalen. De code is CHR\$(16). Met deze stuurcode kunt u de printkop naar elk van de 80 kolommen brengen. Aanwijzingen over

de printpositie moeten als ASCII-teken aan de printer worden doorgegeven. Ook dit zullen we aan de hand van een duidelijk voorbeeld illustreren:

```
10 OPEN 1,4
20 PRINT#1,"AARDBEIENJAM"CHR$(16)"FL 3.12"
```

Ook in dit voorbeeld begint het printen van de prijs bij tabulatorstop 40. De printpositie wordt bepaald door de spatieruimte van 40 tekens voor het teken FL. Zo hebt u bereikt dat de gewenste tekens beginnen op de plaats die SPC(x) of de tekst bepaalt. Maar dat hoeft niet te betekenen dat dit in alle gevallen de meest doelmatige methode is. Om nog even op de rekeningregel van het vorige voorbeeld terug te komen: laten we eens aannemen dat een regel die volgens hetzelfde schema doorgegeven werd, als bedrag de string FL 66,45 zou hebben. Dit bedrag zou wel op dezelfde positie beginnen, maar de komma (hier het decimale teken) zou ten opzichte van de daarboven liggende regel een positie naar rechts opschuiven. Stelt u zich eens voor wat voor resultaat zo'n rekening met een ongelijk aantal cijfers voor de komma zou opleveren. Als documentatie voor uw financiële toestand zou het een ramp zijn. Zoals bijna alle problemen kunnen we ook dit met een programma oplossen. Eerst maakt u van het getal een string. In de string wordt de positie van de decimale komma bekeken en aan de hand daarvan wordt de SPC-parameter of de directe printpositie bepaald (daarbij moeten de verschillende posities binnen een regel natuurlijk ook verschillend worden behandeld). Bij een rekening kan een regel uit verschillende velden bestaan. U begint met het positienummer, daarna komt de **hoeveelheid**, de omschrijving van het artikel, vervolgens de **stukprijs** en tenslotte de totale prijs. Als u al deze gegevens **volgens** bovenstaand schema zou willen verwerken, zou u handen vol **werk** hebben. Het zou eenvoudig zijn als u het formaat van een regel in letter- en cijfervelden kon opbouwen en de uitdraai onder alle omstandigheden met dit formaat zou kloppen. Dit bewerkelijke **karwei** kunt u met het volgende subprogramma omzeilen op de voor u **gemakkelijkste** manier.

```
48000 REM
48001 REM***** FORMATEREN *****
48002 REM
48020 IFFZ%=OTHENFZ%=1:FL%=LEN(FO$):FP$=""
48040 FI%=LEN(FI$):FB$="":FCZ=1
48050 IFMID$(FO$,FZ%,1)=" "THENFZ%=FZ%+1:FB$=FB$+" "
      :GOTO48050
48060 FM$=MID$(FO$,FZ%,1)
48080 FV%=0:FH%=0:FA%=0:FB%=0:FF%=0
48100 IFMID$(FO$,FZ%,1)="."THENFZ%=FZ%+1:FF%=2:GOTO48180
```

```

48120 IFFZ%>FL%THENFA%=FA%+1:FZ%=FZ%+1:GOTO48240
48140 IFMID$(FO$,FZ%,1)=" "THEN48240
48160 FA%=FA%+1:FZ%=FZ%+1:GOTO48100
48180 IFMID$(FO$,FZ%,1)=" "ORFZ%>FL%THEN48240
48200 FB%=FB%+1:FZ%=FZ%+1
48220 GOTO48180
48240 IFMID$(FI$,FC%,1)=" "THENFC%=FC%+1:FF%=FF%+4:GOTO48320
48260 FV%=FV%+1:FC%=FC%+1
48280 IFFC%>FI%THENGOTO48380
48300 GOTO 48240
48320 IFFC%>FI%THEN48380
48340 FH%=FH%+1:FC%=FC%+1
48360 GOTO48320
48380 IFFM$="A"THEN48780
48400 L2=FA%-FV%:IFL2<OTHENFF%=FF%+1:GOTO48720
48420 IFL2=OTHEN48460
48440 FORL1=1TOL2:FB$=FB$+" ":NEXTL1
48460 IFFV%=OTHEN48520
48480 L2=FV%:FORL1=1TOL2
48500 FB$=FB$+MID$(FI$,L1,1):NEXTL1
48520 IFFF%=OTHEN48920
48540 FF%=0:FB$=FB$+" ":L2=FV%+2
48560 IFFB%=OTHEN48920
48580 IFFB%=FH%THENL3=FB%:GOTO48620
48600 L3=-FH%*(FB%>FH%)-FB%*(FH%>FB%)-1
48620 FORL1=L2TOL2+L3
48640 FB$=FB$+MID$(FI$,L1,1):NEXTL1
48660 IFFH%>=FB%THEN48910
48680 L2=FB%-FH%:FORL1=1TOL2
48700 FB$=FB$+"O":NEXTL1:GOTO48910
48720 L2=FA%+FB%:IF(FF%AND2)THENFF%=1:L2=L2+1
48740 FORL1=1TOL2:FB$=FB$+"*"
48760 NEXTL1:GOTO48910
48780 IF(FF%AND4)THENFV%=FV%+FH%+1
48800 L2=FV%:FORL1=1TOL2
48820 IFMID$(FI$,L1,1)<>" "THENL3=L1:L1=L2
48840 NEXTL1
48860 L2=-FV%*(FA%>FV%)-FA%*(FV%>FA%)
48870 IFL2=OTHENL2=FV%
48880 FORL1=L3TOL3+L2
48900 FB$=FB$+MID$(FI$,L1,1):NEXTL1
48902 IFFV%>=FA%THEN48910
48904 L3=FA%-L2:FORL1=1TOL3
48906 FB$=FB$+" ":NEXTL1
48910 IFFZ%>FL%THENFZ%=0:GOTO48960
48920 L2=FZ%:L3=FL%:FORL1=L2TOL3
48930 IFMID$(FO$,L1,1)<>" "THENFZ%=L1:L1=L3:GOTO48950
48940 FB$=FB$+" "
48950 NEXTL1

```

48960 FP\$=FP\$+FB\$:FF%=(FF%AND1)
48970 RETURN

Omdat dit programma veel interne variabelen bevat, beschrijven wij alleen de belangrijkste variabelen.

fo\$ moet het door u gewenste regelformaat bevatten
fi\$ daaraan moet u de data doorgeven die u graag uitgevoerd wilt hebben
fb\$ bevat de zojuist bewerkte deelstring
fp\$ bevat de volledige printstring
fz% is de wijzer op de zojuist bewerkte term in fo\$
ff% signaleert een overschrijding van het aantal plaatsen voor de komma (ff%<>0)
fa% geeft het aantal plaatsen aan voor de komma van het zojuist bewerkte veld in fo\$
fb% als boven, maar nu gaat het om de plaatsen achter de komma.
fv% aantal plaatsen voor de komma in fi\$
fh% aantal plaatsen achter de komma in fi\$

Commentaar:

48020 - Afhankelijk van het eerste inspringen (fz%=0) worden
48080 er enkele variabelen teruggezet.

48100 - De plaatsen voor de komma van het zojuist bewerkte
48160 veld van fo\$ naar fa%.

48180 - Als boven, maar hier gaat het om de plaatsen achter
48220 de komma na fb%.

48240 - Het aantal plaatsen voor de komma in fi\$ na fv%.
48300

48320 - Het aantal plaatsen achter de komma in fi\$ na fh%.
48360

48400 - Hier begint het ordenen van de cijfervelden.
48440 Als fi\$ minder plaatsen voor de komma bevat dan fo\$, wordt er met spaties aangevuld.

48460 - De geldige plaatsen voor de komma komen na fb\$.
48500

48560 - De geldige plaatsen na de komma komen na fb\$.
48640

48660 - Als fi\$ minder plaatsen na de komma bevat dan fo\$,

- 48700 wordt er met nullen aangevuld.
- 48720 - Als fi\$ meer plaatsen voor de komma bevat dan fo\$,
 48760 wordt het hele veld met sterretjes gevuld, zodat het meteen opvalt op de printeruitvoer. In dit geval hebt u het veld te krap bemeten. In het programma wordt deze toestand door ff%<>0 signaleerd.
- 48780 - Hier begint de voorbereiding van de lettervelden.
 48840 Eerst worden leidende spaties onderdrukt zodat de zichtbare tekens ook echt aan het begin van het veld beginnen.
- 48860 - Uit fi\$ worden vervolgens fo\$ tekens naar fb\$
 48900 overgebracht. Overige tekens uit fi\$ worden weggelaten, overige tekens uit fo\$ worden met spaties opgevuld.
- 48910 - Overeenkomstig de afstand tot het volgende veld in
- 48950 fo\$ wordt fb\$ met spaties opgevuld.
- 48960 - fb\$ wordt aan fp\$ toegevoegd en het geheugen vergroot.
 48970

De bediening is eenvoudig: u moet bepalen hoeveel velden zich in de regel bevinden en of deze uit letters of cijfers moet bestaan. Daarna legt u dit formaat in fo\$ vast. Voor de zojuist beschreven rekening zou dat er zo uitzien:

fo\$ = "99 999 aaaaaaaaaa 999.99 9999.99"

U ziet dat in het voorbeeld de cijfervelden met 9 worden aangegeven en de lettervelden met a. Het onderscheid in benaming wordt gemaakt omdat de gegevens in de lettervelden aan de linkerkant moeten komen te staan, en die in de cijfervelden aan de rechterkant, overeenkomstig de ligging van de decimale punt in fo\$. Nu hoeft u alleen nog maar fi\$ van uw gegevens te voorzien en het programma zo vaak te starten als er velden in fo\$ gedefinieerd zijn. Er zitten natuurlijk steeds andere gegevens in fi\$. Aan het einde van dit proces heeft u in fp\$ de complete printregel. In uw programma zou dat er zo uit kunnen zien:

```
10 OPEN 1,4
20 FO$="99 999 AAAAAAAAAA 999.99 9999.99"
30 FI$=STR$(PO):GOSUB 48000:REM POSITIENUMMER
40 FI$=STR$(AN):GOSUB 48000: REM AANTAL
50 FI$=AR$:GOSUB 48000: REM NAAM VAN ARTIKELN
60 FI$=STR$(EP):GOSUB 48000: REM PRIJS PER STUK
70 FI$=STR$(AN*EP):GOSUB 48000: REM TOTALE PRIJS
80 PRINT#1,FP$
```

Zoals u ziet wordt de routine vijf keer gestart, namelijk zo vaak als er velden in fo\$ gedefinieerd zijn. De definitie in fo\$ hoeft u natuurlijk maar een keer door te geven. Deze blijft zolang geldig tot fo\$ een ander formaat krijgt. Een kleine tip: als u bij uitzondering minder velden voor een regel reserveert dan er volgens fo\$ moeten zijn, moet u in de volgende regel fz%=0 zetten, om fo\$ weer van voren af aan te controleren.

5.1.3 Geformateerde programma-listings

Wie kent het probleem niet: u probeert bij het programmeren op elke byte te besparen, zodat het programma tenslotte loopt, maar in het programma zijn geen blanks (spaties) of REM's opgenomen. Als er in dit programma na een paar weken veranderingen aangebracht moeten worden, komt u voor raadsels te staan. Ook al heeft u een listing bij de hand, toch blijft het moeilijk de structuur van het programma te ontdekken. BASIC is nu eenmaal geen goed gestructureerde programmeertaal. We hebben geen pasklare oplossing voor dit probleem. In plaats van achteraf nog met mogelijke oplossingen aan te komen, kunt u beter van tevoren een goede planning en voorbereiding treffen om het programma in de

ontwikkelingsfase al in een overzichtelijke vorm te gieten. Maar met behulp van het afgedrukte BASIC-programma kunt u nog altijd een leesbare listing fabriceren van de BASIC-programma's die op een diskette of cassette zijn opgeslagen. Zo'n listing plaatst spaties tussen alle instructies en maakt door inspringen de verstrengeling van FOR-NEXT-lussen duidelijk. Bovendien begint er na elke dubbele punt een nieuwe printregel. Om de werkwijze van een programma te begrijpen, moet u beschikken over een wat gedetailleerder kennis van de BASIC-interpreter van de VIC 20 en de Commodore 64 en de manier waarop programma's worden opgeslagen. We zullen eerst eens kijken hoe een BASIC-programma in het geheugen ligt opgeslagen. Als u over een machinetaal-monitor beschikt, kunt u de volgende stappen zelf uitvoeren. Allereerst wordt de monitor geladen. Daarna voeren we de programmaregel

```
10 PRINT "HALLO"
```

in en zoeken in het geheugen ons programma op. Eigenaars van een Commodore 64 vinden op de geheugenplaatsen \$0800 tot \$080F de volgende waarden:

0800	00	0E	08	0A	00	99	22	48
0808	41	4C	4C	4F	22	00	00	00

Deze waarden worden bij een VIC 20 echter iets verschoven. Afhankelijk van de aangesloten geheugenuitbreiding krijgen we met de adressen \$0400, \$1000, of \$1200 te maken. Als u eigenaar van een VIC 20 bent, moet u rekening houden met deze veranderde adresruimten, want in de volgende tekstgedeeltes gaan we uit van de adresposities van de Commodore 64. Wat heeft deze serie tekens nu te betekenen? De geheugenplaats \$0800 moet altijd nul zijn: dit geeft het begin van het BASIC-programmageheugen aan. Elke andere waarde in deze geheugenregel leidt bij RUN tot een SYNTAX ERROR. De inhoud van de adressen \$0801 en \$0802 geven de relatie met de volgende programmaregel weer. In dit concrete geval betekenen de waarden dat de volgende BASIC-regel bij adres \$080E begint. De informatie die in deze geheugenplaatsen opgeslagen is, wordt gebruikt voor GOTO's, GOSUB's, maar ook voor de

FOR-NEXT-lussen. Deze adressen worden ook wel aangeduid als linkadressen. De volgende twee geheugenplaatsen bevatten het actuele regelnummer, weer in het formaat lowbyte/highbyte. In ons voorbeeld kregen we \$00A0. In een decimaal getal uitgedrukt wordt dat 10. Nu komen we toe aan de eigenlijke instructie van de regel, de PRINT-instructie. In dit gedeelte van het geheugen zijn de letters voor de instructie niet te vinden. In plaats daarvan treft men een byte aan met de waarde 99. Om de betekenis daarvan te achterhalen, kunnen we deze waarde met de monitor veranderen en onderzoeken of dit eventueel van invloed is op ons BASIC-programma. Bijvoorbeeld: we veranderen 99 in 97 en listen het BASIC-programma. Opeens blijken we een heel andere regel in het geheugen te hebben. De PRINT-instructie is verdwenen en we krijgen de regel:

10 POKE "HALLO"

De instructies worden in een programma omgezet in een code van 1 byte. Daarbij heeft elke instructie haar specifieke en unieke code. Door te experimenteren kunt u heel gemakkelijk controleren voor welke instructie een code geldt. U zult al snel kunnen vaststellen dat de codes (ook wel tokens genoemd) steeds groter dan \$7F zijn. Dit betekent dat bij alle tokens de bit met de hoogste waarde is gezet. Daardoor is het opzoeken van de tokens in een programma heel eenvoudig. In het BASIC-programma komen tekens met een gezette bit van de hoogste waarde alleen maar tussen aanhalingstekens (in strings) voor. Als er dus in het programma zo'n teken met een gezette bit 7 staat, en het aantal aanhalingstekens in deze regel tot aan dat punt nul of even is, moet het wel om een token gaan. Naast de tabel van de tokens is er in het ROM van de computer ook een met alle BASIC-instructies. De computer gebruikt deze bij de invoer van programma's om de syntaxis van de ingevoerde instructies te controleren. Bij de invoer van een regel wordt direct gecontroleerd of er in deze tekst een instructie staat. Als dat het geval is, wordt deze instructie meteen als het overeenkomstige token in het geheugen gezet. De tabel zullen we later in ons programma gebruiken. Na dit verhaal het programma. De afgedrukte listing is in deze vorm geschikt voor de Commodore 64. Voor de VIC 20 (met een uitbreiding van tenminste 3K!), moet in regel 470 een wijziging aangebracht worden.

```

100 DIM BT$(75)
110 LM$=" " :NF$=" " :BL$=" "
120 FORI=1TO8:HZ$=HZ$+LM$:NEXT
130 HZ$=HZ$+"BLZ. "
140 F1$=CHR$(145)+LM$
150 SN=1

```

```

160 PRINTCHR$(147)
170 PRINT
180 PRINT
190 PRINT"          LIST - PROGRAMMA"
200 PRINT
210 PRINT
220 INPUT"          NAAM VAN PROGRAMMA";NA$
230 NP$=NA$+"",P,R"+CHR$(34)
240 PRINT
250 PRINT"          UITVOER NAAR "
260 PRINT"          "CHR$(18)"P"CHR$(146)"RINTER OF "CHR$(18) "D"
          CHR$(146)"ISKDRIVE";
270 INPUT DN$
280 IF DN$="" THEN DN=3:GOTO 360
290 IF DN$="D" THEN DN=8:NL$=NA$+"",S,W"+CHR$(34):GOTO 360
300 IF DN$="P" THEN DN=4
310 PRINT:INPUT"          REGELS PER BLZ. ";RN:
320 RN=(RN=0)*-66+RN
330 PRINT:INPUT"          TITEL          ";TT$
340 HZ$=RIGHT$(HZ$,LEN(HZ$)-LEN(TT$))
350 TT$=TT$+HZ$
360 PRINT
370 INPUT"          LINKER RAND          ";L
380 PRINT
390 PRINT
400 FOR I=0 TO L
410 LM$=LM$+" "
420 NEXT
430 OPEN15,8,15
440 OPEN1,8,4,NP$
450 GOSUB 1150
460 I=0
470 FOR A=41118 TO 41372: REM VIC 20 FOR A=49310 TO 49564
480 X=PEEK(A)
490 IF X>127 THEN 520
500 BT$(I)=BT$(I)+CHR$(X)
510 GOTO 560
520 X=X AND 127
530 BT$(I)=BT$(I)+CHR$(X)
540 PRINTF1$;BT$(I);LM$
550 I=I+1
560 NEXT
570 PRINT
580 PRINT
590 BT$(11)=BT$(11)+" "
600 BT$(39)=BT$(39)+" "
610 REM ***** PROGRAMMA LEZEN *****
620 PRINT CHR$(147)
630 IF DN<8 THEN OPEN4,DN : GOTO 650

```

```

640 OPEN4,DN,3,NA$+".LST,S,W"
650 IF DN<>3 THEN PRINT"UITVOER VOLGT OP DEVICE #";DN
660 PRINT#4
670 GOSUB 1280
680 GOSUB 1180
690 GOSUB 1180
700 REM PRG-START SPRONG
710 REM *****      HOOFDLUS      *****
720 GOSUB 1180
730 GOSUB 1180      : REM SPRONG LINKADRES
740 REM
750 GOSUB 1180
760 LB=A
770 REM REGEL LOW-BYTE
780 GOSUB 1180
790 HB=A
800 REM REGEL HIGH-BYTE
810 ZN$=STR$(HB*256+LB)
820 LN$=LEFT$(LM$,LEN(LM$)-LEN(RN$))+RN$+NF$
830 NN=0
840 REM *****      REGELLUS      *****
850 IF LEN(LN$)<70 THEN 910
860 PRINT#4,LN$;:IF NOT QF THEN PRINT#4
870 IF QF THEN PRINT#4,CHR$(34)
880 LN$=LM$+NF$: IF QF THEN LN$=LN$+CHR$(34)
890 NN=0
900 GOSUB 1220
910 GOSUB 1180
920 REM EEN BYTE HALEN
930 IF A=0 THEN PRINT#4, LN$: LN$="":
    QF=0:RF=0:GOSUB 1220:GOTO720
940 IF NOT RF THEN 970: REM BIJ REM REST ZO LEZEN
950 LN$=LN$+CHR$(A)
960 GOTO 850
970 IF A=32 AND NOT QF THEN 850
980 IF A<>44 THEN 1010
990 IF A=44 AND NN=0 THEN 1010
1000 IF NF$<>" " THEN NF$=LEFT$(NF$,LEN(NF$)-4)
1010 IF A=34 THEN QF=NOT QF
1020 IF A=58 AND NOT QF THEN PRINT#4,LN$; ":" :
    LN$=LM$+NF$: NN=0:GOSUB 1220: GOTO 850
1030 IF A>127 AND NOT QF THEN GOSUB 1060:GOTO 850
1040 LN$=LN$+CHR$(A)
1050 GOTO 850
1060 IF RIGHT$(LN$,1)<>" " THEN LN$=LN$+" "
1070 IF A=128+39 THEN PRINT#4,LN$: LN$=LM$+NF$ :GOSUB1220
1080 IF A=129 THEN NF$=NF$+BL$:
    REM BL$= 4 SPATIES VOOR FOR NEXT

```

```

1090 IF A=130 THEN NN=1: GOSUB 1130
1100 IF A=143 THEN RF=NOT RF: REM REMFLAG AAN
1110 LN$=LN$+BT$(A-128)+" "
1120 RETURN
1130 IF NF$<>" " THEN NF$=LEFT$(NF$,LEN(NF$)-4):
    LN$=LEFT$(LN$,LEN(LN$)-4)
1140 RETURN
1150 INPUT#15,A$,B$,C$,D$
1160 IF A$<>"00" THEN PRINT B$: CLOSE1: CLOSE4: CLOSE15: END
1170 RETURN
1180 GET#1,A$
1190 A=ASC(A$+CHR$(0))
1200 IF ST <>0 THEN CLOSE1: CLOSE4: CLOSE15: END
1210 RETURN
1220 ZZ=ZZ+1
1230 IF ZZ=ZN THEN 1250
1240 RETURN
1250 FOR I=1 TO 72-ZN
1260 PRINT#4
1270 NEXT
1280 PRINT#4,LM$;" ";TT$;SN
1290 PRINT#4
1300 PRINT#4
1310 ZZ=3
1320 SN=SN+1
1330 RETURN

```

Onder andere voor de uitbreiding van het programma is de beschrijving een welkom hulpmiddel. Een mogelijke uitbreiding kan bestaan uit een omzetting van alle reverse tekens in de vorm CHR\$(x). Daardoor zou het programma ook bruikbaar worden voor printers die deze tekens niet zelf kunnen produceren. In de regels 100-150 worden de belangrijke variabelen geïntialiseerd. Array BT\$ bevat later de verschillende BASIC-instructies en LM\$ bepaalt de kantlijn. Het beeldscherm wordt gewist. U voert de programmanaam en het nummer van het uitvoerapparaat in. Als uitvoerapparaat kan de printer of de diskdrive dienen. Om een listing in een lopende tekst op te kunnen nemen, kiezen we het laatste. Hier kunnen we dan commentaren aan toevoegen. Wordt de vraag naar het device-nummer met RETURN beantwoord, dan wordt het beeldscherm aangesproken als uitvoerapparaat. Daarmee kunt u zich van tevoren een beeld vormen van de listings. Bij de keuze tussen beeldscherm of diskdrive worden de regels 310 tot 350 overgeslagen. In deze regels worden bij de printeruitvoer de parameters voor het aantal regels per pagina en de titeltekst gevraagd. De titeltekst verschijnt dan met het paginanummer op elke afgedrukte bladzijde. Wordt de vraag naar het aantal regels per pagina met RETURN beantwoord, dan komen er 66 te staan. In

regel 370 wordt de gewenste kantlijnpositie gevraagd. We adviseren een waarde van 6-10, zodat de perforatie van de kettingformulieren niet door de listing loopt. Ook deze invoer moet met RETURN worden afgesloten. Dan wordt met perforaties geen rekening gehouden. Na het openen van het foutkanaal en de programmapfile springt het programma naar regel 1150. Daar wordt een eventuele foutmelding uitgelezen en aangegeven. De meest voorkomende melding is FILE NOT FOUND. Nadat het systeem heeft vastgesteld of de gewenste file er is, wordt in de FOR-NEXT-lus van de regels 470-560 de reeds genoemde tabel met instructiewoorden uit het ROM gePEEKt. Deze tabel heeft als bijzonderheid dat alle instructies gewoon achter elkaar staan. Om de verschillende instructies van elkaar te kunnen onderscheiden, is elke laatste letter van een instructie met een hoge achtste databit in de genoemde tabel opgenomen. Als regel 490 dat constateert, gaat het programma verder bij regel 520. Nadat de bit is gewist, wordt de instructie op het beeldscherm weergegeven. Vervolgens wordt de index van de variabelen verhoogd en de volgende instructie uit het geheugen gePEEKt. Tenslotte krijgen de instructies IF en THEN in totaal een lengte van 5 tekens; dat wil zeggen dat ze met spaties worden aangevuld.

Het subprogramma in de regels 1180-1210 haalt de bytes van de schijf. Bij elke sprong naar dit subprogramma wordt een teken van de diskette in variabele A ter beschikking gesteld. Eerst wordt deze routine twee keer opgeroepen, waarbij de waarde in A niet wordt gebruikt. Dat komt doordat de eerste twee bytes van elke programmapfile het laadadres aangeven. Voor ons zijn deze waarden niet interessant. Nu komt het programma bij de hoofdlus (regel 710 e.v.). Ook hier worden eerst twee bytes gehaald en verder niet gebruikt. Deze bytes zijn het reeds beschreven linkadres. De volgende twee zijn van groter belang omdat zij het regelnummer bevatten. In de regels 810 en 820 wordt dit berekend en geformateerd. In de hoofdlus wordt eerst gecontroleerd of de string LN\$ niet groter dan 70 tekens is. Als deze hoeveelheid is bereikt, wordt de regel uitgevoerd en het subprogramma in regel 1220 opgeroepen. Daar wordt bekeken of het aantal geprinte regels het maximum al heeft bereikt en als dat nodig is wordt een form-feed toegevoegd. Daarna wordt in regel 910 een byte opgehaald en kan het eigenlijke werk beginnen. Afhankelijk van de waarde van variabele A moeten er nu verschillende handelingen worden gestart. Eerst wordt gecontroleerd of A gelijk aan 0 is. Dat is het geval wanneer het einde van de regel is bereikt. Dan moet LN\$ worden uitgevoerd en de regelteller voor de printregels worden verhoogd. A=32 komt overeen met een spatie. Wanneer de spatie in de programmatekst staat, wordt hij overgeslagen en de volgende byte gehaald. Als de spatie tussen aanhalingstekens is geplaatst, wordt hij later aan de string LN\$ toegevoegd. Het onderscheiden van de "aanhalingstekens"-modus neemt de variabele

QF voor zijn rekening. Vervolgens wordt gecontroleerd of er ergens een komma (44) voorkomt. Deze regel is belangrijk in verband met de NEXT-instructie waarin verschillende, door een komma gescheiden tellers voorkomen. Regel 1010 controleert A op de waarde 34 (aanhalingstekens). Als $A = 34$, wordt aan de variabele QF met behulp van de NOT-bewerking de waarde -1 of 0 toegekend. Bij elk aanhalingsteken wordt deze waarde dus omgekeerd. Zo kunt u zonder grote programmeer-technische ingrepen bepalen of een teken in de programmatekst of tussen aanhalingstekens staat. Daarna wordt bekeken of er een dubbele punt (58) voorkomt. Ook hierbij moet aan de hand van QF worden vastgesteld of de dubbele punt in de programmatekst staat of in een tekststring (literal), die immers altijd tussen aanhalingstekens staat. Als de dubbele punt in het programma staat, wordt de regel LN\$ geprint met de dubbele punt aan het einde. Daarna begint een nieuwe printregel. Zo krijgt u een overzichtelijke indeling. Statements die door een komma van elkaar gescheiden zijn, krijgen hun eigen regel. Maar als het ontvangen teken groter is dan 127 en in de programmatekst staat, gaat het om een token. Staat het tussen aanhalingstekens dan wordt het aan LN\$ toegevoegd. Waarde 167 (regel 1070) staat voor het token THEN. Als het programma een THEN-token signaleert, wordt de tot dan toe opgebouwde regel geprint om de IF-THEN-structuur zichtbaar te maken. In regel 1080 wordt de interpretercode voor de FOR-instructie gezocht. Bij het aantreffen van dit token wordt NF\$ uitgebreid met de serie tekens die in BL\$ gedefinieerd zijn. Op die wijze wordt bereikt dat FOR-NEXT-lussen inspringen en daardoor overzichtelijker worden. Regel 1090 onderzoekt of er een NEXT-instructie aanwezig van. Is dat het geval, dan worden in regel 1130 de tekens verwijderd die bij de FOR-instructie aan NF\$ toegevoegd zijn. De NN-flag wordt gezet om de bijzonderheid te onderstrepen van de NEXT-instructie met de door komma's gescheiden variabelen. Regel 1100 tenslotte controleert of er sprake is van een REM-statement. In dat geval wordt de RF (REM-flag) gezet. Dat heeft tot gevolg dat de tekens na REM zonder verdere controle en eventuele wijzigingen geprint worden. Na het doorlopen van regel 1060-1100 wordt aan de regelstring LN\$ de karakterreeks toegevoegd voor de betreffende interpretercode. Daartoe wordt het betreffende array-element direct via token 128 geadresseerd. Bovendien wordt er een spatie aan toegevoegd om het geheel leesbaarder te maken. Daarna springt het programma in de hoofdloop terug. Op het eerste gezicht lijkt het programma misschien erg ingewikkeld. Maar in feite gebeurt er hetzelfde als bij elke LIST, er zijn alleen wat handigheidjes toegevoegd.

5.2 Tekstverwerking

Tekstverwerking behoort naast het bestandsbeheer beslist tot de belangrijkste werkerreinen van de computer. En omdat bij tekstverwerking altijd een printer is betrokken, hebben we het programma MINITEKST gemaakt. Natuurlijk kan dit programma niet met echte tekstverwerkingsprogramma's, zoals TEXTOMAT, concurreren. Maar als u heel snel een brief of een notitie moet typen, is het wel degelijk bruikbaar. Het programma is in machinetaal geschreven. Het brengt u vast wel op een idee over de manier waarop de printer in machinetaal kan worden gestuurd. Het tweede programma is een tekstverwerker van een heel andere soort. Hiermee kunt u een kleine tekst vergroten. Voor het maken van posters is dat heel aardig.

5.2.1 MINITEKST, een eenvoudige teksteditor

Waar een printer en een computer gebruikt worden, mag een tekstverwerker niet ontbreken. Nu hebben tekstverwerkingsprogramma's met de mogelijkheden van bijvoorbeeld TEXTOMAT zo'n omvang dat alleen al met een assemblerlisting gemakkelijk 300 bladzijden gevuld zouden kunnen worden. Ofschoon we het zelf niet hebben gecontroleerd, denken we dat zo'n programma minstens 40 tot 50 bladzijden DATA-regels zou beslaan. En dan is het nog maar de vraag of er genoeg ruimte in het computergeheugen over is om het programma onder te brengen. Wij hebben een wat bescheidener routine geschreven, maar niet in BASIC. Het intensieve gebruik van stringfuncties in een tekstverwerkingsprogramma in BASIC leidt te vaak tot garbage collection. Bij garbage collection worden de strings uit het geheugen verwijderd die niet meer nodig zijn. Dit kan een paar seconden duren, maar ook 30 minuten. Dat is afhankelijk van het programma en het aantal variabelen. Gedurende die tijd negeert de computer invoer via het toetsenbord. Een tweede punt van kritiek op een teksteditor in BASIC is de omvang, zelfs wanneer alleen de belangrijkste functies worden uitgevoerd. Om in BASIC tekst te verwerken kunt u ook allemaal PRINT#-instructies gebruiken. Maar dan moet wel bij elke regel het regelnummer en de instructie PRINT#<F worden vermeld. Ook moet u er steeds om denken dat de regel niet te lang wordt. Het is dus een beetje overdreven om

zelf een tekst te schrijven in de vorm van een BASIC-programma dat uit PRINT-instructies bestaat. Een voordeel van deze methode is dat u de handige beeldschermeditor kunt gebruiken en de instructies voor het opslaan en laden van teksten. Dat voordeel nemen we over en de nadelen beperken we zoveel mogelijk. Het resultaat is een klein machinetaalprogramma van 508 bytes. In PRINT-instructies moeten voor de tekst altijd aanhalingstekens staan. Ook zonder aanhalingstekens kunt u achter een regelnummer tekst invoeren. Maar dan gebeuren er merkwaardige dingen, als er in de hoofdletter/kleine letter-modus hoofdletters (met SHIFT) worden ingevoerd. Deze tekens verdwijnen zomaar, of er komen plotseling BASIC-instructies in de tekst te staan. Wanneer u een regelnummer en meteen daarna aanhalingstekens invoert, blijven hoofd- en kleine letters bij het listen behouden. De computer bevindt zich in de quote-modus. Maar er is nog een probleem. De cursortoetsen, de toetsen RUN/STOP, CLR/HOME en de functietoetsen corresponderen met reverse grafische tekens of letters. De programmaregel met de cursortoetsen is daardoor niet mogelijk behalve wanneer de quote-modus door invoer van "aanhalingstekens sluiten" weer wordt opgeheven. Het eerste gedeelte van het machinetaalprogramma lost deze problemen op. De computer maakt de regelnummers zelfstandig, zoals gebruikelijk bij de AUTO-instructie. Aanhalingstekens hoeft u ook niet in te voeren; het programma regelt dat zelf. In een speciale geheugenplaats in het RAM (in de zero-page) is vastgelegd dat de quote-modus is ingeschakeld. Nadat de aanhalingstekens op het beeldscherm zijn verschenen, wordt deze geheugencel zo geladen, dat de computer een uitgeschakelde quote-modus voorgetoverd wordt. Nu rest ons nog de controle van de regellengte. Je ziet vaak dat mensen met de ogen aan het toetsenbord gekleefd zitten. Vooral bij een continue tekstinvoer wordt er maar zelden een blik op het beeldscherm geworpen. Wilt u de regellengte toch controleren dan is een akoestisch signaal het geëigende middel. In de zero-page (zo worden de eerste 256 bytes van het RAM aangeduid) bevindt zich een geheugencel, die aangeeft hoeveel tekens in de BASIC-regel staan die op dat moment wordt ingevoerd. Ons kleine programma onderzoekt regelmatig deze geheugencel met een interrupt. Zo gauw een bepaalde waarde bereikt is, klinkt er via de televisieluidspreker een akoestisch signaal. Dan moet u uiterlijk aan het eind van het volgende woord de RETURN-toets bedienen. Bij zeer lange woorden kan het noodzakelijk zijn het woord af te breken en de rest op de volgende regel te schrijven. Zodra de RETURN-toets ingedrukt is, wordt meteen een nieuw regelnummer met aanhalingstekens geschreven. Het tweede deel van het programma is verantwoordelijk voor het printen van de tekst. Natuurlijk zou er ook een uitvoer door middel van LIST mogelijk zijn. Maar wie schrijft er nu een brief waarvan elke regel met een nummer begint? Het tweede programmadeel zorgt ervoor dat de storende regelnummers en de aanhalingstekens niet worden geprint. In

plaats daarvan komt er een marge aan de linkerkant, zodat de letters niet meteen bij de scheurlijn beginnen. Om het gebruik van het programma wat gemakkelijker te maken, is het in de BASIC-interpreter geplaatst. Nadat het programma met SYS is opgeroepen, staan er twee nieuwe instructies ter beschikking: de !GET-instructie om regels in te voeren, en de !PRINT-instructie om de tekst te printen. Bij de laatste is ook de afkorting !? mogelijk. Het vraagteken is de afkorting van de PRINT-instructie. Als u al een beetje met assemblerprogrammering heeft geoefend, kunt u de routine heel eenvoudig zelf wijzigen of uitbreiden.

C000		*=	\$C000
C000	BEEPFLG	=	2
C000	FORMFEED	=	6
C000	LM	=	10
C000	CR	=	\$0D
C000	LO	=	\$14
C000	HI	=	\$15
C000	TXNAN	=	\$2B
C000	TXTEND	=	\$2D
C000	FAC	=	\$62
C000	TEXT	=	\$71
C000	CHRGET	=	\$73
C000	CHRGOT	=	\$79
C000	PRTCODE	=	\$99
C000	GETCODE	=	\$A1
C000	FILNM	=	\$B7
C000	LGFNR	=	\$B8
C000	SECAD	=	\$B9
C000	DEVAD	=	\$BA
C000	LINE	=	\$FB
C000	INCR	=	\$FD
C000	REGELS	=	\$FE
C000	BUFFER1	=	\$101
C000	BUFFER2	=	\$200
C000	VECTOR	=	\$0302
C000	VECTOR1	=	\$0308
C000	IRQVEC	=	\$0314
C000	BASLOOP	=	\$A474
C000	OLDVEC	=	\$A483
C000	MAINLOOP	=	\$A486
C000	GOON	=	\$A569
C000	VERDER	=	\$A576
C000	BASSTR	=	\$A68E
C000	EXECOLD	=	\$A7E7
C000	CHKCOM	=	\$AEFD
C000	SYNTAX	=	\$AF08
C000	ILLQUANT	=	\$B248

C000	GETPAR	=	\$B7EB
C000	INTFLOAT	=	\$BC49
C000	FLOATASC	=	\$BDDD
C000	SID	=	\$D400
C000	CLRSCRN	=	\$E544
C000	IRQOLD	=	\$EA31
C000	OPEN	=	\$FFC0
C000	CLOSE	=	\$FFC3
C000	CKOUT	=	\$FFC9
C000	CLRCH	=	\$FFCC
C000	INPUT	=	\$FFCF
C000	BSOUT	=	\$FFD2
C000	STOP	=	\$FFEL

;***** HIER BEGINT HET *****

C000	A9	41	LDA	#<TESTIN	
C002	A0	C0	LDY	#>TESTIN	
C004	8D	08 03	STA	VECTOR1	
C007	8C	09 03	STY	VECTOR1+1	
C00A	A9	80	LDA	#128	; SHFT-COMMODORE DISABLE
C00C	8D	91 02	STA	\$0291	
C00F	A9	17	LDA	#\$17	; GROOT-KLEIN INSCHAKELEN
C011	8D	18 D0	STA	\$D018	
C014	A9	00	LDA	#0	
C016	A8		TAY		
C017	91	2B	STA	(\$2B),Y	; BASIC-PROG. WISSEN
C019	C8		INY		
C01A	91	2B	STA	(\$2B),Y	
C01C	A5	2B	LDA	\$2B	; EN VECTOREN INSTELLEN
C01E	18		CLC		
C01F	69	02	ADC	#\$02	
C021	85	2D	STA	\$2D	
C023	A5	2C	LDA	\$2C	
C025	69	00	ADC	#0	
C027	85	2E	STA	\$2E	
C029	20	8E A6	JSR	BASSTR	
C02C	20	44 E5	JSR	CLRSCRN	; CLEAR SCREEN
C02F	A9	80	LDA	#\$80	
C031	8D	8A 02	STA	650	; REPEAT ALLE TOETSEN
C034	A9	64	LDA	#100	; EERSTE REGELNUMMER
C036	85	FB	STA	LINE	
C038	A9	00	LDA	# 0	
C03A	85	FC	STA	LINE+1	
C03C	A9	0A	LDA	#10	; REGELAFSTAND
C03E	85	FD	STA	INCR	
C040	60		RTS		; EN TERUG
					; TESTEN OP !
C041	20	73 00	TESTIN	JSR	CHRGET

C044	C9	21		CMP	"!"	
C046	F0	06		BEQ	TEST2	; ALS ! DAN VERDER TESTEN
C048	20	79	00	JSR	CHRGOT	
C04B	4C	E7	A7	JMP	EXECOLD	; ANDERS NORMALE INSTR.
C04E	20	73	00	JSR	CHRGET	; VOLGENDE TEKEN HALEN
C051	C9	A1		CMP	#GETCODE	; IS HET !GET?
C053	F0	0A		BEQ	GET	
C055	C9	99		CMP	#PRTCODE	; IS HET !PRINT?
C057	F0	03		BEQ	PRT	
C059	4C	08	AF	JMP	SYNTAX	; GEEN VAN BEIDE DAN
C05C	4C	41	C1	JMP	PRT1	; SYNTAX ERROR
C05F	20	73	00	JSR	CHRGET	
C062	F0	0D		BEQ	LO	; MEER PARAMETERS?
C064	20	EB	B7	JSR	GETPAR	; DAN HALEN
C067	86	FD		STX	INCR	; EN ONTHOUDEN
C069	A5	14		LDA	LO	
C06B	85	FB		STA	LINE	
C06D	A5	15		LDA	HI	
C06F	85	FC		STA	LINE +1	
C071	A9	80	LO	LDA	#128	
C073	8D	91	02	STA	\$0291	
C076	A9	17		LDA	#\$17	
C078	8D	18	D0	STA	\$D018	
C07B	A9	85		LDA	#<AUTO	; VECTOR INVOER REGEL
C07D	8D	02	03	STA	VECTOR	; NAAR AUTO-ROUTINE
C080	A9	C0		LDA	#>AUTO	; OMBUIGEN
C082	8D	03	03	STA	VECTOR+1	
C085	20	8B	C0	JSR	AUTONUM	
C088	4C	86	A4	JMP	MAINLOOP	
C08B	AD	14	03	LDA	IRQVEC	; BEEP INGESCHAKELD?
C08E	C9	03		CMP	#<BEEPON	
C090	F0	0C		BEQ	WO	
C092	78			SEI		
C093	A9	03		LDA	#<BEEPON	; INTERRUPT VECTOR
C095	8D	14	03	STA	IRQVEC	; NAAR BEEP-ROUTINE
C098	A9	C1		LDA	#>BEEPON	; OMBUIGEN
C09A	8D	15	03	STA	IRQVEC+1	
C09D	58			CLI		
C09E	A5	FB	WO	LDA	LINE	; REGELNUMMER HALEN
COA0	A6	FC		LDX	LINE+1	; EN IN ASCII-FORMAAT
COA2	85	63		STA	FAC+1	; VERANDEREN
COA4	86	62		STX	FAC	
COA6	A2	90		LDX	#\$90	
COA8	38			SEC		
COA9	20	49	BC	JSR	INTFLOAT	
COAC	20	DD	BD	JSR	FLOATASC	
COAF	A2	00		LDX	#0	
COB1	BD	01	01	LDA	BUFFER1,X	
COB4	F0	09		BEQ	L2	

COB6 9D 00 02		STA BUFFER2,X	; REGELNUMMER IN INVOERBUFFER
COB9 20 D2 FF		JSR BSOUT	; EN OP BEELDSCHERM
COBC E8		INX	
COBD D0 F2		BNE L1	
COBF A5 FB	L2	LDA LINE	; REGELNUMMER VERHOGEN
COC1 18		CLC	
COC2 65 FD		ADC INCR	
COC4 85 FB		STA LINE	
COC6 90 02		BCC L3	
COC8 E6 FC		INC LINE +1	
COCA A9 20	L3	LDA #" "	; SPATIE
COCC 9D 00 02		STA BUFFER2,X	; IN INVOERBUFFER
COCF 20 D2 FF		JSR BSOUT	; EN OP BEELDSCHERM
COD2 E8		INX	
COD3 A9 22		LDA #34	; AANHALINGSTEKENS
COD5 9D 00 02		STA BUFFER2,X	; IN INVOERBUFFER
COD8 E8		INX	
COD9 20 D2 FF		JSR BSOUT	; EN OP BEELDSCHERM
CODC A9 00		LDA #0	
CODE 85 D4		STA \$D4	; QUOTE-FLAG WISSEN
COEO 20 CF FF		JSR INPUT	
COE3 C9 0D		CMP #CR	; ALLEEN RETURN?
COE5 F0 03		BEQ L4	
COE7 4C 69 A5		JMP GOON	; ZONIET, DAN VERDER
COEA A5 FB	L4	LDA LINE	; ANDERS REGELNUMMER WEER
COEC E5 FD		SBC INCR	; VERLAGEN
COEE 85 FB		STA LINE	
COFO B0 02		BCS L5	
COF2 C6 FC		DEC LINE+1	
COF4 A9 83	L5	LDA #<OLDVEC	; EN INVOERVECTOR
COF6 A0 A4		LDY #>OLDVEC	; RESETTEN
COF8 8D 02 03		STA VECTOR	
COFB 8C 03 03		STY VECTOR+1	
COFE A2 00		LDX #0	; INVOERBUFFER WISSEN
C100 4C 76 A5		JMP WEITER	
C103 A5 D3	BEEPON	LDA \$D3	; CURSORKOLOM
C105 C9 40		CMP #64	; 64 TEKENS?
C107 F0 07		BEQ BEEP	; DAN BEEPEN
C109 A9 00		LDA #0	; ANDERS FLAG WISSEN
C10B 85 02		STA BEEPFLG	; MAG WEER BEEPEN
C10D 4C 31 EA		JMP IRQOLD	; OUDE IRQ-ROUTINE
C110 A5 02	BEEP	LDA BEEPFLG	; MAG BEEPEN?
C112 D0 2A		BNE END	; ZONIET, DAN VERDER
C114 A9 08		LDA # 8	
C116 8D 18 D4		STA SID+24	; VOLUME
C119 A9 08		LDA # 8	
C11B 8D 03 D4		STA SID+3	
C11E A9 00		LDA # 0	
C120 8D 00 D4		STA SID	; LOWBYTE FREQ

```

C123 A9 78          LDA # 120
C125 8D 01 D4       STA SID+1          ; HIGHBYTE FREQ
C128 A9 FF          LDA # 255
C12A 8D 05 D4       STA SID+5          ; ATTACK/DECAY
C12D A9 F9          LDA # 249
C12F 8D 06 D4       STA SID+6          ; SUST/RELEASE
C132 A9 41          LDA # 65
C134 8D 04 D4       STA SID+4          ; STUURREGISTER
C137 A9 40          LDA # 64
C139 8D 04 D4       STA SID+4          ; STUURREGISTER
C13C 85 02          STA BEEPFLG        ; EENS EN NOOIT WEER
C13E 4C 31 EA END   JMP IRQOLD         ; IRQ EXIT

```

;***** PRINT ROUTINE *****

```

C141 A9 04   PRT1   LDA #4              ; DEVICENUMMER
C143 85 BA           STA DEVAD
C145 A9 7E           LDA #126           ; FILENUMBER
C147 85 B8           STA LGFNR
C149 A9 00           LDA #0             ; FILENAAM
C14B 85 B7           STA FILNM
C14D A9 00           LDA #0             ; SECUNDAIR ADRES
C14F 85 B9           STA SECAD
C151 20 C0 FF        JSR OPEN           ; FILE OPENEN
C154 A6 B8           LDX LGFNR
C156 20 C9 FF        JSR CKOUT          ; UITVOER OP PRINTER
C159 A5 2D           LDA TXTAN+2        ; = VARIABLENSTART
C15B 38              SEC
C15C E9 02           SBC #2             ; MINUS 2
C15E C5 2B           CMP TXTAN          ; ALS BASIC-START?
C160 D0 0B           BNE PRINT1
C162 A5 2E           LDA TXTAN+3
C164 E9 00           SBC #0
C166 C5 2C           CMP TXTAN+1
C168 D0 03           BNE PRINT1
C16A 4C FA C1        JMP END1           ; GEEN TEKST IN GEHEUGEN
C16D A5 2B   PRINT1  LDA TXTAN          ; SKIP LINK,
C16F 18              CLC                ; REGELNUMMER EN
C170 69 05           ADC #5             ; AANHALINGSTEKENS
C172 85 71           STA TEXT           ; PLAATS IN TEKST ONTHOUDEN
C174 A5 2C           LDA TXTAN+1        ; HIGHBYTE OOK ONTHOUDEN
C176 85 72           STA TEXT+1
C178 A9 42           LDA #66            ; REGELS PER PAGINA
C17A 85 FE           STA REGELS
C17C A0 0A           LDY #LM
C17E A9 20           LDA #" "          ; KANTLIJN PRINTEN
C180 20 D2 FF LMLoop1 JSR BSOUT
C183 88              DEY
C184 D0 FA           BNE LMLoop1

```

C186 A9 11	LDA #17	; CURSOR DOWN
C188 20 D2 FF	JSR BSOUT	; VOOR GROOT-KLEINSCHRIFT
C18B A0 00	LDY #0	; ***** LOOP *****
C18D B1 71 LOOP	LDA (TEXT),Y	
C18F D0 03	BNE W1	; ALS 0 DAN EIND BASIC-REGL
C191 4C A4 C1	JMP NUL	
C194 20 D2 FF W1	JSR BSOUT	; TEKENS STUREN
C197 20 E1 FF	JSR STOP	; OP STOP GEDRUKT?
C19A F0 51	BEQ EINDE	; DAN EINDE PRINTEN
C19C E6 71	INC TEXT	; WIJZER OP TEKST VERHOGEN
C19E D0 02	BNE W2	
C1A0 E6 72	INC TEXT+1	
C1A2 D0 E9 W2	BNE LOOP	
C1A4 A9 0D NUL	LDA #CR	
C1A6 20 D2 FF	JSR BSOUT	; LINE FEED STUREN
C1A9 C6 FE	DEC REGELS	; REGELCOUNTER -1
C1AB D0 0C	BNE W5	; 66 REGELS GEPRINT?
C1AD A0 06	LDY #FORMFEED	; DAN VOLGENDE PAGINA
C1AF 20 D2 FF WE	JSR BSOUT	
C1B2 88	DEY	
C1B3 D0 FA	BNE WE	
C1B5 A9 42	LDA #66	; REGELCOUNTER WEER OP 66
C1B7 85 FE	STA REGELS	
C1B9 A0 0A W5	LDY #LM	; KANTLIJN VOLGENDE REGEL
C1BB A9 20	LDA #" "	
C1BD 20 D2 FF LMLoop	JSR BSOUT	
C1C0 88	DEY	
C1C1 D0 FA	BNE LMLoop	
C1C3 A9 11	LDA #17	; CURSOR DOWN
C1C5 20 D2 FF	JSR BSOUT	
C1C8 A0 03	LDY #3	
C1CA E6 71 L6	INC TEXT	; SKIP 0 EN LINK-ADRES
C1CC D0 02	BNE W3	
C1CE E6 72	INC TEXT+1	
C1D0 88 W3	DEY	
C1D1 D0 F7	BNE L6	
C1D3 A5 71	LDA TEXT	
C1D5 C5 2D	CMP TXTEND	; EINDE BASIC-PROGRAMMA?
C1D7 D0 06	BNE W4	
C1D9 A5 72	LDA TEXT+1	
C1DB C5 2E	CMP TXTEND+1	
C1DD F0 0E	BEQ EINDE	
C1DF A0 03 W4	LDY #3	
C1E1 E6 71 L7	INC TEXT	; SKIP REGELNUMMER
C1E3 D0 02	BNE W6	; EN AANHALINGSTEKENS
C1E5 E6 72	INC TEXT+1	
C1E7 88 W6	DEY	
C1E8 D0 F7	BNE L7	
C1EA 4C 8D C1	JMP LOOP	; VOLGENDE REGEL PRINTEN

C1ED A9 OD	EINDE	LDA #CR	
C1EF 20 D2 FF		JSR BSOUT	; LAATSTE LINE FEED
C1F2 20 CC FF		JSR CLRCH	; UITVOER OP BEELDSCHERM
C1F5 A5 B8		LDA LGFNR	
C1F7 20 C3 FF		JSR CLOSE	; FILE SLUITEN
C1FA 4C 74 A4 END1		JMP BASLOOP	; TERUG IN BASIC

Een BASIC-lader is voor dit soort programma's nooit weg. Daarvoor zijn geen assembler- of monitorprogramma's nodig. Als u dit programma hebt overgetypt, moet u het op diskette of cassette opslaan voordat u het start. Na RUN wordt het automatisch gewist!

```

100 FOR I = 49152 TO 49660
110 READ X :POKE I,X : S=S+X : NEXT
120 DATA 169, 65,160,192,141, 8, 3,140, 9, 3,169,128
130 DATA 141,145, 2,169, 23,141, 24,208,169, 0,168,145
140 DATA 43,200,145, 43,165, 43, 24,105, 2,133, 45,165
150 DATA 44,105, 0,133, 46, 32,142,166, 32, 68,229,169
160 DATA 128,141,138, 2,169,100,133,251,169, 0,133,252
170 DATA 169, 10,133,253, 96, 32,115, 0,201, 33,240, 6
180 DATA 32,121, 0, 76,231,167, 32,115, 0,201,161,240
190 DATA 10,201,153,240, 3, 76, 8,175, 76, 65,193, 32
200 DATA 115, 0,240, 13, 32,235,183,134,253,165, 20,133
210 DATA 251,165, 21,133,252,169,128,141,145, 2,169, 23
220 DATA 141, 24,208,169,133,141, 2, 3,169,192,141, 3
230 DATA 3, 32,139,192, 76,134,164,173, 20, 3,201, 3
240 DATA 240, 12,120,169, 3,141, 20, 3,169,193,141, 21
250 DATA 3, 88,165,251,166,252,133, 99,134, 98,162,144
260 DATA 56, 32, 73,188, 32,221,189,162, 0,189, 1, 1
270 DATA 240, 9,157, 0, 2, 32,210,255,232,208,242,165
280 DATA 251, 24,101,253,133,251,144, 2,230,252,169, 32
290 DATA 157, 0, 2, 32,210,255,232,169, 34,157, 0, 2
300 DATA 232, 32,210,255,169, 0,133,212, 32,207,255,201
310 DATA 13,240, 3, 76,105,165,165,251,229,253,133,251
320 DATA 176, 2,198,252,169,131,160,164,141, 2, 3,140
330 DATA 3, 3,162, 0, 76,118,165,165,211,201, 64,240
340 DATA 7,169, 0,133, 2, 76, 49,234,165, 2,208, 42
350 DATA 169, 8,141, 24,212,169, 8,141, 3,212,169, 0
360 DATA 141, 0,212,169,120,141, 1,212,169,255,141, 5
370 DATA 212,169,249,141, 6,212,169, 65,141, 4,212,169
380 DATA 64,141, 4,212,133, 2, 76, 49,234,169, 4,133
390 DATA 186,169,126,133,184,169, 0,133,183,169, 0,133
400 DATA 185, 32,192,255,166,184, 32,201,255,165, 45, 56
410 DATA 233, 2,197, 43,208, 11,165, 46,233, 0,197, 44
420 DATA 208, 3, 76,250,193,165, 43, 24,105, 5,133,113
430 DATA 165, 44,133,114,169, 66,133,254,169, 17, 32,210
440 DATA 255,160, 10,169, 32, 32,210,255,136,208,250,160

```

```

450 DATA 0,177,113,208, 3, 76,164,193, 32,210,255, 32
460 DATA 225,255,240, 81,230,113,208, 2,230,114,208,233
470 DATA 169, 13, 32,210,255,198,254,208, 12,160, 6, 32
480 DATA 210,255,136,208,250,169, 66,133,254,169, 17, 32
490 DATA 210,255,160, 10,169, 32, 32,210,255,136,208,250
500 DATA 160, 3,230,113,208, 2,230,114,136,208,247,165
510 DATA 113,197, 45,208, 6,165,114,197, 46,240, 14,160
520 DATA 3,230,113,208, 2,230,114,136,208,247, 76,141
530 DATA 193,169, 13, 32,210,255, 32,204,255,165,184, 32
540 DATA 195,255, 76,116,164
550 IF S <> 64452 THEN PRINT "FOUT IN DATA !!" : END
560 PRINT "OK"
570 SYS .12 * 4096

```

We gaan nu wat meer gedetailleerd in op het gebruik. Het programma is in deze vorm direct te gebruiken op de Commodore 64 en alle Commodore printers kunnen het via de seriële IEC-bus opvangen. Na het programma met de SYS-instructie te hebben gestart, worden verschillende geheugenlocaties geïnitialiseerd alsof een NEW-instructie werd uitgevoerd. Het laadprogramma is gewist en de tekstinvoer kan beginnen. !GET opent de invoermodus; toets dat in. Meteen zullen op het beeldscherm het regelnummer 100 en aanhalingstekens verschijnen. Daarna voert u uw tekst in. Als de regel langer wordt dan 64 tekens, hoort u een piep. Let wel: de hele regel wordt geteld, inclusief regelnummer. Na de piep moet u bij de eerstvolgende gelegenheid de RETURN-toets indrukken, omdat anders de printeruitvoer in de war raakt. Omdat de linkermarge 10 spaties breed is, passen er 70 tekens in een regel. Als dit aantal wordt overschreden, komen de overvloedige tekens op de volgende regel te staan, waarbij ook de linker marge wordt gebruikt. U kunt de invoermodus onmiddellijk na het regelnummer verlaten met een druk op RETURN. Daardoor komt u in de normale invoermodus van de computer. Nu kunt u alle instructies invoeren die de computer kent. Het gaat vooral om de LIST-, LOAD- en SAVE-instructie. Wanneer u de invoermodus hebt verlaten, kunt u de tekst listen en als een normaal BASIC-programma behandelen. Als u weer !GET intikt, gaat het programma verder met de laatste regel, waar alleen een RETURN in voorkomt. Vanzelfsprekend kunt u ook regels met de hand invoeren. Schrijf ze zoals ze in de invoermodus op het beeldscherm verschijnen. De !GET-instructie heeft ook de mogelijkheid twee parameters aan te geven. De syntax luidt dan als volgt: !GET regelnummer, regelafstand. !GET 200,5 start dus bij regel 200; de volgende regel is dan nummer 205. Dit is vooral nuttig als u in een oude tekst verder wilt werken. In dat geval geeft u een regelnummer aan dat na het laatst in de tekst gebruikte regelnummer ligt. De !PRINT-instructie kent verder geen parameters. Zodra hij wordt ingevoerd, opent het programma een kanaal met het logisch filennummer 126 en stuurt het

de tekst in hoofdletters naar de printer. Bij het printen wordt de STOP-toets gecheckt, zodat u het printen op elk gewenst moment kunt stoppen. Als u in de tekst de verschillende mogelijkheden van de printer wilt gebruiken, kan dat ook. Laten we de volgende tekst als voorbeeld nemen:

XY-diskettes in gezinsverpakking
2000 stuks voor slechts 15,34

De onderste regel van de tekst moet in een verbreed lettertype geprint worden. Het inschakelen van dit brede lettertype gebeurt bij de meeste printers met behulp van de stuurcode CHR\$(14). Het probleem hierbij is: hoe moet ik de printer deze stuurcode meedelen? De Commodore 64 heeft een CONTROL-toets. Wanneer u deze toets samen met een lettertoets indrukt, geeft de Commodore 64 een teken met waarde 1 bij CONTROL-A tot waarde 26 bij CONTROL-Z. Waarde 14 krijgt men bij CONTROL-N. Dit teken kunnen we alleen als teken in een tekst invoeren in de quote-modus. Daarom is het proces enigszins gecompliceerd, want de quote-modus is uitgeschakeld tijdens de invoer. Gecompliceerd of niet, het werkt. Eerst schrijft u de tekst zoals normaal. Dan verlaat u de invoermodus door RETURN in te drukken na het volgende regelnummer. Daarna zet u de cursor op de 2 in de tekst. Door SHIFT-INST/DEL in te drukken, maakt u ruimte voor een volgend teken en nu kunt u CONTROL-N indrukken. Op het beeldscherm verschijnt een reverse n die ervoor zorgt dat bij het printen op het brede lettertype wordt overgeschakeld. Zo kunt u alle codes van 1 tot 26 in de tekst invoegen. Code 27 krijgt u door het indrukken van CONTROL-;. Door die handeling verschijnt een reverse rechte haak openen. Code 29 krijgt u door CONTROL-; wat overeenkomt met een reverse rechte haak sluiten. In ieder geval moet u de combinatie van CONTROL en @ vermijden. Deze toetsencombinatie heeft de waarde 0 en geeft het einde van de BASIC-regel aan. Alles wat in een regel op de nul volgt, wordt bij het printen weggelaten. Wanneer u uw computer vaker als tekstverwerkingsprogramma moeten aanschaffen. Met MINITEKST zijn dingen als het verschuiven van blokken, het verbinden van teksten tot een lange tekst die in zijn geheel geprint moet worden, of het schrijven van circulares niet of nauwelijks mogelijk. Maar voor wie slechts af en toe een brief of een korte notitie moet tikken, biedt het programma uitkomst.

5.2.2 Kan het ook groter?

Een matrixprinter heeft het juiste lettertype voor brieven, manuscripten of listings. Maar wat te doen als u grotere tekens wilt? Bezitters van een 7-naalds matrixprinter (tot die categorie behoren de MPS-801, de VC-1515, de VC-1525, maar ook de GP-700 of de Epson met de bijbehorende interfaces) kunnen de programma's in deze paragraaf zonder meer gebruiken voor grotere tekens. Het probleem bij het printen van eigen tekens is altijd de samenstelling van de karakterset. De meeste mensen zinkt de moed in de schoenen bij de gedachte aan het aantal gegevens dat moet worden ingevoerd. Een veelbelovend programmeerobject verdwijnt maar al te vaak in de la omdat niet iedereen zin heeft zoveel Kbyte in te tikken. Maar waarom zou u een karakterset ontwerpen als die al in de computer aanwezig is? Zowel bij de VIC 20 als bij de Commodore 64 bestaat de mogelijkheid de ingebouwde karakterset uit te lezen. Stop! zullen sommigen nu roepen, je kunt de karakterset van de Commodore 64 in BASIC niet uitlezen zonder dat de computer uitvalt. Dat klopt maar gedeeltelijk. Er bestaat een eenvoudige methode waardoor het uitlezen toch mogelijk wordt. Omdat het adresbereik van de Commodore 64 ten dele door verschillende componenten tegelijk wordt gebruikt, nemen de karaktergenerator en de I/O-IC's (de VIC, de CIA's en de SID) dezelfde adresruimte in beslag. Bit 2 van de processorport bepaalt met -CHAREN of de processor zich richt op de karaktergenerator of de perifere IC's. Als deze bit laag is, is de karaktergenerator ingeschakeld. Bit 2 kan in BASIC worden beïnvloed door een POKE op adres 1. Zonder verdere maatregelen zal de Commodore 64 na POKE 1,51 (-CHAREN = low) geen activiteit meer vertonen na de READY-melding. Dat komt omdat door de POKE-instructie de processor ook geen vat meer heeft op CIA1. Dit IC zorgt 60 keer per seconde voor een interrupt. Die interrupt hebben we nodig maar dan moet de processor deze CIA wel kunnen aanspreken. Wanneer de karakterset moet worden uitgelezen, moet deze interrupt worden uitgeschakeld. De machinetaalinstructie SEI schakelt de interrupt uit. In BASIC bereiken we hetzelfde als we de oorzaak van de interrupt elimineren. Dat kan door het zogenaamde ICR (= Interrupt Control Register) in CIA1 te wissen. Een meer gedetailleerde beschrijving zou te ver voeren. Belangrijk is dat POKE 56333,127 de interrupt uitschakelt en dat POKE 56333,129 de interrupt weer inschakelt. Bij de VIC 20 is de zaak veel eenvoudiger. Hier ligt de karakterset immers in de invloedssfeer van de processor zodat die zonder speciale maatregelen uitgelezen kan worden. We weten nu hoe we bij de karaktergenerator kunnen komen. Voordat we de volgende stap nemen, bekijken we eerst hoe karakters tot stand komen. Op het

beeldscherm worden ze lijn voor lijn opgebouwd en niet met 8 lijnen tegelijk zoals bij een matrixprinter. Dat geldt zowel voor televisies als monitoren. Net als bij een matrixprinter bestaan de karakters op het beeldscherm dus uit een aantal punten. Per teken staan er 8x8 punten ter beschikking. De acht punten van zo'n horizontale puntenrij worden in het beeldschermgeheugen in 8 bits (= 1 byte) opgeslagen. Is er een bit gezet, dan verschijnt er op het beeldscherm een punt. Is het karakter gewist, dan verschijnt op die plaats de achtergrondkleur. Een compleet karakter bestaat uit 8 bytes, die in het beeldschermgeheugen achter elkaar zijn opgeslagen in de volgorde van de beeldschermcode en niet volgens hun ASCII-waarden. De eerste 8 bytes van de karakterset bevatten de bitpatronen voor het apestaartje (@) en de volgende 8 bytes leveren de vorm van de hoofdletter A. Wanneer we een groot karakter op de printer willen maken, kunnen we hiervoor de 1-naaldsmodus gebruiken. We willen elke gezette bit afbeelden als een zwart veld van 7 punten verticaal en 6 punten horizontaal. Omdat er in een grafiek met hoog oplossend vermogen 480 punten op een regel geprint kunnen worden, komen we bij deze puntgrootte op een karaktermatrix van 64 punten verticaal en 48 punten horizontaal. Daarmee komen we uit op precies 10 tekens per regel. En nu het programma. Omdat het vervaardigen van een karaktergenerator veel tijd kost, hebben we het programma opgedeeld. Het eerste programma vervaardigt uit de eerste 64 tekens van de karaktergenerator de gegevensreeksen die de printer aansturen. Ze worden voor later gebruik op de floppy geschreven in een sequentieel bestand. Het tweede programma haalt die gegevens van de schijf, brengt ze onder in een array en vraagt naar de te printen tekst. Deze tekst wordt onderzocht en vervolgens geprint. De hele karakterset kan in deze vorm worden uitgeprint. Maar waarschijnlijk is het niet uw bedoeling grafische tekens of cursor-stuurtekens zo op papier te zetten. Bovendien wordt het bestand op de diskette dan vier keer zo groot en het inleesproces wordt evenredig langer. In de actuele vorm beslaat de file 67 blokken. Lezers die niet in het bezit zijn van een diskdrive kunnen gemakkelijk beide programma's samenvoegen. U moet in dat geval wel iedere keer wachten tot de printgegevens klaar zijn. Het programma dat de printgegevens maakt:

```

100 DIM A$( 63,7)
110 POKE56320+13,127      : REM VERVALT BIJ VIC 20
120 POKE 1,51             : REM VERVALT BIJ VIC 20
130 A=13*4096             : REM 8*4096 BIJ VIC 20
140 FORK=0 TO 63
150 PRINTK,
160 FORI=0 TO 7
170 A$(K,I)=PEEK(A+I)
180 NEXTI

```

```

190 A=A+8
200 NEXTK
210 POKE1,55 : REM VervalT BIJ VIC 20
220 POKE56320+13,129 : REM VervalT BIJ VIC 20
230 OPEN15,8,15
240 OPEN1,8,4,"NORM CHR DATA,S,W"
250 GOSUB460
260 PRINTCHR$(147);
270 FORK=OTO 63
280 PRINTCHR$(K+((K<32)*-64));
290 FORJ=OTO7
300 A=A%(K,J)
310 FORI=7TOOSTEP-1
320 IF(2*I=(AAND2*I))THENGOSUB420:GOTO340
330 GOSUB440
340 NEXTI
350 PRINT#1,X$
360 X$=""
370 NEXTJ
380 NEXTK
390 CLOSE1
400 CLOSE15
410 END
420 X$=X$+CHR$(8)+CHR$(26)+CHR$(5)+CHR$(255)
430 RETURN
440 X$=X$+CHR$(8)+CHR$(26)+CHR$(5)+CHR$(128)
450 RETURN
460 INPUT#15,EN,ET$,ET,ES
470 IFEN=OTHENRETURN
480 PRINTEN;ET$;ET,ES
490 CLOSE15:CLOSE1

```

Met het volgende programma kunnen grote karakters worden geprint:

```

100 DIM A$( 63,7)
110 PRINTCHR$(147)
120 OPEN15,8,15
130 OPEN4,8,4,"NORM CHR DATA,S,R"
140 GOSUB380
150 FORK=OTO 63
160 PRINTK,
170 FORI=OTO7
180 INPUT#4,A$(K,I)
190 NEXTI,K
200 CLOSE4
210 CLOSE15
220 PRINT
230 A$=""
240 INPUT"TEKST (MAX.10 TEKENS)";A$

```

```

250 IFA$<>""THEN280
260 PRINT#1,CHR$(15)
270 END
280 PRINTCHR$(147);A$
290 OPEN1,4
300 FORI=0TO7
310 FORL=1TOLEN(A$)
320 K=ASC(MID$(A$,L,1))
330 K=K+((K>63)*64)
340 PRINT#1,A$(K,I);
350 NEXTL:PRINT#1:NEXTI
360 CLOSE1
370 GOTO220
380 INPUT#15,EN,ET$,ES,ET
390 IFEN=0THENRETURN
400 PRINTEN;ET$;ES;ET
410 CLOSE15
420 CLOSE4

```

10 tekens per regel is niet optimaal. Gaat het om langere teksten, dan kunt u ze met dit programma na elkaar afdrukken, uitknippen en naast elkaar plakken. Ook is de grootte van de karakters misschien niet in alle gevallen toereikend. De programma's die we nu introduceren, ruimen deze problemen uit de weg. Als u langere teksten wilt printen, hoeft u alleen maar de tekst 90 graden te draaien, waardoor het papier overdwars wordt bedrukt. De lengte van de regels wordt dan alleen nog beperkt door de lengte van het papier (altijd nog ruim 600 meter bij een doos van 2000 bladen). Als u de tekst draait, kunnen de tekens ook groter afgebeeld worden. Het programma vraagt bij het samenstellen van de te printen gegevens naar de gewenste grootte. U kunt uit verschillende gradaties (1 tot en met 5) kiezen. Grootte 1 levert karakters van hetzelfde formaat als het vorige programma. Een karakter in grootte 5 beslaat de hele breedte van het papier. Zo krijgt u karakters die ook op enkele tientallen meters afstand nog goed te lezen zijn. Dit levert aardige mogelijkheden op voor reclamedoeleinden of school- en sportmanifestaties. Wanneer u langere teksten wilt printen, moet u wel wat geduld hebben. De printersnelheid wordt nu niet meer in karakters per seconde, maar in karakters per minuut uitgedrukt. De karakters worden als zwarte blokconstructies geprint; daarom is het aanbevelenswaardig de printer na circa 10 karakters in grootte 5 een korte rustpauze te geven. De printerkop zou anders te heet worden.

```

100 DIM A$( 63,7)
110 PRINT CHR$(147)
120 NA$="TEKENGEGEVENS"
130 INPUT "GROOTTE (1-5)";V

```

```

140 IF V<1 OR V>5 THEN 110
150 PRINT CHR$(147);
160 NA$=NA$+STR$(V)+",S,W"
170 POKE 56320+13,127 : REM INTERRUPT VERBODEN VOOR C64
180 POKE 1,51 : REM CHAREN LOW, ALLEEN VOOR C64
190 REM KARAKTERGENERATOR NAAR ARRAY AZ(K,I)
200 A=13*4096 : REM A=8*4096 BIJ VIC 20
210 FOR K=0 TO 63:PRINT K,:FOR I=0 TO 7
220 AZ(K,I)=PEEK(A+I)
230 NEXT:A=A+8:NEXT
240 POKE1,55:POKE56320+13,129 : REM INTERRUPT WEER TOELATEN
250 REM *!!!!!!!!!!!!!!!!!!!!* REGEL 135 VERVALT VOOR VIC 20
260 OPEN15,8,15
270 OPEN1,8,4,NA$
280 GOSUB 440
290 PRINT CHR$(147);
300 FOR K=0 TO 63 : PRINT CHR$(K+((K<32)* -64));
310 FOR I=7 TO 0 STEP -1
320 FOR J=7 TO 0 STEP-1
330 A=AZ(K,J)
340 IF (2^I=(A AND 2^I)) THEN GOSUB 400 : GOTO 360
350 GOSUB 420
360 NEXT J:PRINT#1,X$:X$=""
370 NEXT I,K
380 CLOSE 1:CLOSE 15
390 END
400 X$=X$+CHR$(8)+CHR$(26)+CHR$(12*V)+CHR$(255)
410 RETURN
420 X$=X$+CHR$(8)+CHR$(26)+CHR$(12*V)+CHR$(128)
430 RETURN
440 INPUT#1, EN, ET$, ES, ET
450 IF EN=0 THEN RETURN
460 PRINT EN; ET$; ES; ET
470 CLOSE 1 : CLOSE 15

```

Meteen hierna het printprogramma:

```

100 INPUT "GROOTTE (1-5)";V
110 POKE 56320+13,127 : REM INTERRUPT VERBIEDEN
120 POKE 1,51 : REM CHAREN UIT LOW
130 REM KARAKTERGENERATOR NAAR ARRAY AZ(K,I)
140 DIM AZ( 63,7),A$( 63,7)
150 A=13*4096
160 FOR K=0 TO 63
170 : PRINT K
180 : FOR I=0 TO 7
190 : : AZ(K,I)=PEEK(A+I)
200 : NEXT
210 : A=A+8

```

```

220 NEXT
230 POKE1,55:POKE56320+13,129 : REM INTERRUPT WEER TOELATEN
240 FOR K=0 TO 63
250 : PRINT K
260 : FOR I=7 TO OSTEP-1
270 : : FOR J=7 TO 0 STEP-1
280 : : : A=A%(K,J)
290 : : : IF (2^I=(A AND 2^I))THEN GOSUB 480: GOTO 310
300 : : : GOSUB 500
310 : : NEXT J
320 : : A$(K,7-I)=X$
330 : : X$=""
340 : NEXT I
350 NEXT K
360 INPUT"TEKST";A$
370 OPEN1,4
380 FOR L=1 TO LEN(A$)
390 : PRINT CHR$(19); MID$(A$,L,1)
400 : K=PEEK(1024)
410 : FOR I=0 TO 7
420 : : FOR J=1 TO V
430 : : PRINT#1,A$(K,I)
440 : : NEXT J
450 : NEXT I
460 NEXT L
470 CLOSE 1:GOTO360
480 X$=X$+CHR$(8)+CHR$(26)+CHR$(12*V)+CHR$(255)
490 RETURN
500 X$=X$+CHR$(8)+CHR$(26)+CHR$(12*V)+CHR$(128)
510 RETURN

```

5.3 Hardcopy

Nog maar een paar jaar geleden waren printers nauwelijks te betalen. Zelfs voor eenvoudige matrixprinters zonder speciale mogelijkheden moest de consument al astronomische bedragen neertellen. En dat in een tijd waarin de prijzen van de hardware ook nog een aanslag op de portemonnee betekenden. Dus werd voor het printen alles gebruikt wat maar enigszins geschikt was. Zo werden er ondanks tekortkomingen aardige bedragen betaald voor afgedankte telexmachines van de PTT. De vreugde was groot als men met zelfgebouwde speciale interfaces en een zelf samengesteld driverprogramma programmalistings kon printen. Men nam zelfs op

de koop toe dat de karakterset van deze monsterachtige apparaten niet aan kritische eisen voldeed. In die tijd kon men in sommige vaktijdschriften programmalistings van een speciaal soort bewonderen. Knappe koppen waren op het idee gekomen videohardcopies van het beeldscherm te maken. Zo kon een complete listing geprint worden, bestaande uit beeldschermfoto's. Nu is dit allemaal veel eenvoudiger en beter. Maar nog steeds is het idee van de beeldschermfoto's actueel. Het is alleen niet meer noodzakelijk een donkere kamer naast de computer te hebben om een grafische voorstelling of een tekst van het beeldscherm op papier te krijgen. Om de beeldscherm inhoud af te drukken, is behalve de printer alleen nog een klein programma nodig.

5.3.1 Hardcopy van tekst

De karakters op het beeldscherm zijn de weergave van een bepaalde geheugenruimte in uw computer. Het RAM is zonder meer te lezen en te beschrijven. Met POKE kan een karakter op het beeldscherm worden geschreven en met PEEK kunnen karakters uit het beeldschermgeheugen gelezen worden. Als bijvoorbeeld in de linker bovenhoek van het beeldscherm de letter A staat, krijgen we door Z=PEEK (adres) de waarde 1 in variabele Z. Het adres is het adres van de geheugenplaats van de eerste beeldschermpositie. Het eerste adres van het beeldschermgeheugen ligt bij de Commodore 64 gewoonlijk bij 1024 (hex \$0400). Bij de VIC 20 ligt het startadres (afhankelijk van de geheugenuitbreiding) bij 7680 (\$1E00) of bij 4096 (\$1000). De informatie op het beeldscherm begint vanaf dit adres. Helaas heeft Commodore een ongebruikelijke codering voor de karakters toegepast. In het beeldscherm-RAM komen slechts enkele karakters overeen met de ASCII-waarden. De waarden die in het video-RAM opgeslagen zijn, moeten in de juiste ASCII-waarden worden omgezet. We hebben dus een universele hardcopyroutine nodig, die zowel in het programma opgeroepen kan worden als door een toetsdruk een beeldschermafdruk kan geven. Beide mogelijkheden biedt het volgende programma. Een SYS-instructie roept de HC-routine op; gelijktijdig indrukken van CTRL en F1 heeft hetzelfde effect. De afgedrukte assemblerlisting is alleen op de Commodore 64 van toepassing. Van de BASIC-laders die daarna staan afgedrukt, is de tweede aangepast voor de VIC 20. Bezitters van een VIC 20 die de assemblerlisting willen gebruiken, kunnen voor de aanpassing het beste gebruik maken van een vergelijkingstabel van de routines in het bedrijfsstelsel. Die is te vinden in het boek 64 INTERN uit deze reeks.

033C			*=	\$033C
033C	IRQ		=	\$314
033C	KEY		=	\$EA87
033C	CTRL		=	\$028D
033C	INTER		=	\$EA31
033C	IRQFLAG		=	\$02
033C	TOETS		=	\$C5
033C	FA		=	\$BA
033C	LF		=	\$B8
033C	TEMP		=	\$71
033C	FNLEN		=	\$B7
033C	AANTAL		=	\$C6
033C	SA		=	\$B9
033C	OPEN		=	\$FFC0
033C	CKOUT		=	\$FFC9
033C	BSOUT		=	\$FFD2
033C	STOP		=	\$FFE1
033C	STORE		=	\$67
033C	CLRCH		=	FFCC
033C	CLOSE		=	FFC3
033C	78		SEI	; INTERRUPT VERBIEDEN
033D	A9 49		LDA #<NEW	; INTERRUPTVECTOR NAAR
033F	A0 03		LDY #>NEW	; HARDCOPY OMBUIGEN
0341	8D 14 03		STA IRQ	
0344	8C 15 03		STY IRQ+1	
0347	58		CLI	
0348	60		RTS	; EN TERUG
				; NIEUWE INTERRUPTROUTINE
0349	20 87 EA NEW		JSR KEY	; TOETSENBOORD LEZEN
034C	AD 8D 02		LDA CTRL	; CTRL-TOETS?
034F	C9 04		CMP #\$04	; DAN BIT2 GEZET
0351	D0 79		BNE EXIT1	; ANDERS GEEN HC
0353	A5 C5		LDA TOETS	
0355	C9 04		CMP #\$04	
0357	D0 73		BNE EXIT1	
0359	A9 00		LDA #\$00	
035B	85 C6		STA AANTAL	
035D	F0 04		BEQ HCOPY	; ABSOLUTE SPRONG
035F	A9 FF	SYSIN	LDA #\$FF	; INSPRONG VOOR SYS
0361	85 02		STA IRQFLAG	; MARKEREN
0363	A9 04	HCOPY	LDA #\$04	; HARDCOPY
0365	85 BA		STA FA	; DEVICE-NUMMER
0367	A9 7E		LDA #\$7E	
0369	85 B8		STA LF	; LOG.FILENUMMER
036B	A9 00		LDA #\$00	
036D	A0 04		LDY #\$04	; VIDEORAM OP \$0400
036F	85 71		STA TEMP	
0371	84 72		STY TEMP+1	
0373	85 B7		STA FNLEN	; GEEN FILENAAM

0375	A9 00	LDA #\$00
0377	85 B9	STA SA ;SECUNDAIR ADRES
0379	20 C0 FF	JSR OPEN
037C	A6 B8	LDX LF
037E	20 C9 FF	JSR CKOUT
0381	A2 19	LDX #\$19 ;25 REGELS
0383	A9 0D LOOP	LDA #\$0D ;LINE FEED
0385	20 D2 FF	JSR BSOUT ;UITVOEREN
0388	20 E1 FF	JSR STP ;STOP-TOETS ?
038B	F0 2E	BEQ EXIT ;DAN KLAAR
038D	A0 00	LDY #\$00
038F	B1 71 LOOP2	LDA (TEMP),Y;TEKEN UIT VIDEORAM
0391	85 67	STA STORE ;HALEN EN MARKEREN
0393	29 3F	AND #\$3F
0395	06 67	ASL STORE
0397	24 67	BIT STORE ;BEELDSCHERMCODE
0399	10 02	BPL * + 4 ;IN ASCII VERANDEREN
039B	09 80	ORA #\$80
039D	70 02	BVS \$03A1
039F	09 40	ORA #\$40
03A1	20 D2 FF	JSR BSOUT ;EN UITVOEREN
03A4	C8	INY
03A5	C0 28	CPY #\$28 ;40 TEKENS AFGEDRUKT ?
03A7	D0 E6	BNE LOOP2
03A9	98	TYA
03AA	18	CLC
03AB	65 71	ADC TEMP ;WIJZER OP VIDEORAM
03AD	85 71	STA TEMP ;MET 40 VERHOGEN
03AF	90 02	BCC * + 4
03B1	E6 72	INC TEMP+1
03B3	CA	DEX
03B4	D0 CD	BNE LOOP ;ALLE REGELS AFGEDRUKT ?
03B6	A9 0D	LDA #\$0D ;NIEUWE REGEL UITVOEREN
03B8	20 D2 FF	JSR BSOUT
03BB	20 CC FF EXIT	JSR CLRCH ;UITVOER WEER OP BEELD-
03BE	A9 7E	LDA #\$7E ;SCHERM
03C0	20 C3 FF	JSR CLOSE ;FILE SLUITEN
03C3	A5 02	LDA IRQFLAG;HC OPGEROEPEN MET CTRL-F1?
03C5	F0 05	BEQ EXIT1 ;DAN NAAR EXIT1
03C7	A9 00	LDA #\$00 ;FLAG WISSEN
03C9	85 02	STA IRQFLAG
03CB	60	RTS ;RETURN VOOR SYS-OPROEP
03CC	4C 31 EA EXIT1	JMP INTER ;OUDE INTERRUPT OPROEPEN

Het programma begint in de cassettebuffer op adres 828. Dit adres moet u ook bij de SYS-instructie aangeven om de routine te activeren. Het logische filenummer is 126. Als u in uw eigen programma gebruik maakt van hetzelfde filenummer moet u de file na gebruik van de hardcopy opnieuw openen omdat die aan het eind van

de routine natuurlijk gesloten wordt. Om fouten te voorkomen, kunt u dit filenummer reserveren voor de hardcopy. Druk voor een hardcopy tegelijk op CTRL en F1. In een programma roept u een hardcopy op met SYS 863. De STOP-toets onderbreekt het printproces. Daardoor wordt natuurlijk ook een eventueel lopend programma onderbroken. Nu volgen de BASIC-laders voor de Commodore 64 en de VIC 20. We beginnen met die voor de Commodore 64:

```
100 FOR I = 828 TO 974
110 READ X : POKE I,X : S=S+X : NEXT
120 DATA 120,169, 73,160, 3,141, 20, 3,140, 21, 3, 88
130 DATA 96, 32, 30,235,173,141, 2,201, 4,208,121,165
140 DATA 197,201, 4,208,115,169, 0,133,198,240, 4,169
150 DATA 255,133, 2,169, 4,133,186,169,126,133,184,169
160 DATA 0,160, 4,133,113,132,114,133,183,169, 0,133
170 DATA 185, 32,192,255,166,184, 32,201,255,162, 23,169
180 DATA 13, 32,210,255, 32,225,255,240, 46,160, 0,177
190 DATA 113,133,103, 41, 63, 6,103, 36,103, 16, 2, 9
200 DATA 128,112, 2, 9, 64, 32,210,255,200,192, 22,208
210 DATA 230,152, 24,101,113,133,113,144, 2,230,114,202
220 DATA 208,205,169, 13, 32,210,255, 32,204,255,169,126
230 DATA 32,195,255,165, 2,240, 5,169, 0,133, 2, 96
240 DATA 76,191,234
250 IF S <> 17844 THEN PRINT "FOUT IN DATA !!" : END
260 PRINT "OK"
270 SYS 828
```

In de listing voor de VIC 20 is de highbyte van het startadres van het video-RAM vet gedrukt. Er is een waarde ingezet die van toepassing is op de standaarduitvoering. Als u een geheugenuitbreiding van tenminste 8K hebt geïnstalleerd, moet u de waarde 30 vervangen door 16.

```
100 FOR I = 828 TO 974
110 READ X : POKE I,X : S=S+X : NEXT
120 DATA 120,169, 73,160, 3,141, 20, 3,140, 21, 3, 88
130 DATA 96, 32, 30,235,173,141, 2,201, 4,208,121,165
140 DATA 197,201, 4,208,115,169, 0,133,198,240, 4,169
150 DATA 255,133, 2,169, 4,133,186,169,126,133,184,169
160 DATA 0,160, 30,133,113,132,114,133,183,169, 0,133
170 DATA 185, 32,192,255,166,184, 32,201,255,162, 23,169
180 DATA 13, 32,210,255, 32,225,255,240, 46,160, 0,177
190 DATA 113,133,103, 41, 63, 6,103, 36,103, 16, 2, 9
200 DATA 128,112, 2, 9, 64, 32,210,255,200,192, 22,208
210 DATA 230,152, 24,101,113,133,113,144, 2,230,114,202
220 DATA 208,205,169, 13, 32,210,255, 32,204,255,169,126
230 DATA 32,195,255,165, 2,240, 5,169, 0,133, 2, 96
240 DATA 76,191,234
```

```

250 IF S <> 17844 THEN PRINT "FOUT IN DATA !!" : END
260 PRINT "OK"
270 SYS 828

```

5.3.2 Hardcopy van grafische voorstellingen

We kunnen nu een afdruk maken van een tekst op het beeldscherm. Laten we eens bekijken of we op dezelfde manier een afdruk kunnen maken van een grafische voorstelling. Goede grafische uitbreidingssets, zoals SUPERGRAFIEK 64, hebben de noodzakelijke routines en instructies om van het grafische beeldscherm een afdruk te maken (ook op verschillende typen printers). De grafiekmodus van de Commodore 64 is ook in BASIC met PEEK en POKE te bedienen. Daardoor ontbreken er vaak geschikte programma's om grafische voorstellingen op papier te krijgen. Zulke programma's zijn juist interessant omdat u bij programma's in BASIC lang moet wachten totdat de afbeelding klaar is. We introduceren nu twee hardcopy-routines voor grafische voorstellingen. Ze draaien beide op de Commodore 64. De eerste is bedoeld voor de bediening van de 7-naaldsprinter. Met de tweede routine kunt u grafische plaatjes uitdraaien op een Commodore 1526. En dat terwijl deze printer eigenlijk geen grafische mogelijkheden heeft. Beide routines worden in de adresruimtes zo aangepast dat bediening met de grafiekhulp uit het boek 64 INTERN mogelijk is. Eerst geven we u de assemblerlistings en daarna volgen weer de BASIC-laders. Om te beginnen de hardcopy-routine voor de 7-naaldsprinter:

```

C200          *= $C200
C200          A  = $AC
C200          B  = A+1
C200          USE = $FD
C200          XK  = $14
C200          FLG = $97
C200          OFFX = $63
C200          GRMEM = $2000
C200 20 F1 B7   JSR $B7F1
C203 86 67     STX $67
C205 20 0F F3   JSR $F30F ;ZOEKT LOGISCH FILENUMMER
C208 20 1F F3   JSR $F31F ;ZET FILEPARAMETER
C20B A6 67     LDX $67
C20D 20 C9 FF   JSR $FFC9 ;KANAAL OPENEN
C210 A9 FF     LDA #$FF
C212 85 61     STA $61 ;MASKER

```

C214	A9	07		LDA	#\$07	
C216	85	FD		STA	\$FD	;PAKKETGROOTTE
C218	A9	1C		LDA	#\$1C	
C21A	85	97		STA	\$97	;REGELTELLER
C21C	A9	00		LDA	#\$00	
C21E	8D	0D	C3	STA	\$C30D	;YK-FLAG
C221	A9	28	SEV1	LDA	#\$28	
C223	8D	0F	C3	STA	\$C30F	;PAKKETTELLER
C226	A2	04		LDX	#\$04	
C228	BD	B1	C2 SEV11	LDA	\$C2B1,X	;GECENTREERD
C22B	20	D2	FF	JSR	\$FFD2	
C22E	CA			DEX		
C22F	10	F7		BPL	\$C228	
C231	A9	00		LDA	#\$00	
C233	85	63		STA	\$63	
C235	85	64		STA	\$64	;XK=0
C237	AD	0D	C3 SEV2	LDA	\$C30D	
C23A	85	65		STA	\$65	;YK
C23C	A9	00		LDA	#\$00	
C23E	85	FE		STA	\$FE	
C240	A5	63	SEV3	LDA	\$63	;XK-L
C242	A6	64		LDX	\$64	;XK-H
C244	A4	65		LDY	\$65	;YK
C246	20	B6	C2	JSR	\$C2B6	;POSITIE BEREKENEN
C249	A0	00		LDY	#\$00	
C24B	B1	AC		LDA	(\$AC),Y	
C24D	A6	FE		LDX	\$FE	
C24F	9D	11	C3	STA	\$C311,X	;IN BUFFER
C252	E6	65		INC	\$65	;YK
C254	E8			INX		
C255	86	FE		STX	\$FE	
C257	E4	FD		CPX	\$FD	
C259	D0	E5		BNE	\$C240	
C25B	A9	00		LDA	#\$00	
C25D	A0	07		LDY	#\$07	
C25F	A6	FD	SEV4	LDX	\$FD	
C261	1E	11	C3 SEV5	ASL	\$C311,X	
C264	2A			ROL		
C265	CA			DEX		
C266	10	F9		BPL	\$C261	
C268	25	61		AND	\$61	;BIJ LAATSTE REGEL=#\$OF
C26A	09	80		ORA	#\$80	;HIGH-BYTE
C26C	20	D2	FF	JSR	\$FFD2	;STUREN
C26F	88			DEY		
C270	10	ED		BPL	\$C25F	
C272	A5	63		LDA	\$63	;XK-L
C274	18			CLC		
C275	69	08		ADC	#\$08	
C277	85	63		STA	\$63	;XK+8

C279	90	02		BCC	\$C27D	
C27B	E6	64		INC	\$64	;XK-H
C27D	CE	0F	C3 SEV6	DEC	\$C30F	
C280	DO	B5		BNE	\$C237	
C282	A9	0D		LDA	#\$0D	
C284	20	D2	FF	JSR	\$FFD2	
C287	AD	0D	C3	LDA	\$C30D	
C28A	18			CLC		
C28B	69	07		ADC	#\$07	
C28D	8D	0D	C3	STA	\$C30D	;FLAG+7
C290	C6	97		DEC	\$97	
C292	FO	03		BEQ	\$C297	
C294	4C	21	C2 SEV8	JMP	\$C221	
C297	A9	04	SEV81	LDA	#\$04	
C299	C5	FD		CMP	\$FD	
C29B	FO	0C		BEQ	\$C2A9	
C29D	85	FD		STA	\$FD	;LAATSTE REGEL
C29F	A9	01		LDA	#\$01	
C2A1	85	97		STA	\$97	
C2A3	A9	0F		LDA	#\$0F	
C2A5	85	61		STA	\$61	;MASKER
C2A7	DO	EB		BNE	\$C294	
C2A9	A9	0F	SEV7	LDA	#\$0F	;NORMALE MODUS
C2AB	20	D2	FF	JSR	\$FFD2	
C2AE	4C	CC	FF	JMP	\$FFCC	;SLUIT KANAAL
C2B1	50	00	10	SAVTAB.BYT	80,0,16,27,8	
C2B6						
C2B6	85	14	HPOSN	STA	\$14	
C2B8	86	15		STX	\$15	
C2BA	98			TYA		
C2BB	4A			LSR	A	
C2BC	4A			LSR	A	
C2BD	4A			LSR	A	
C2BE	AA			TAX		
C2BF	BD	EF	C2	LDA	\$C2EF,X	
C2C2	85	AD		STA	\$AD	
C2C4	8A			TXA		
C2C5	29	03		AND	#\$03	
C2C7	AA			TAX		
C2C8	BD	09	C3	LDA	\$C309,X	
C2CB	85	AC		STA	\$AC	
C2CD	98			TYA		
C2CE	29	07		AND	#\$07	
C2D0	18			CLC		
C2D1	65	AC		ADC	\$AC	
C2D3	85	AC		STA	\$AC	
C2D5	A5	14		LDA	\$14	
C2D7	29	F8		AND	#\$F8	
C2D9	85	63		STA	\$63	

```

C2DB A9 20          LDA #$20      ;ACTUELE GRAFIEKPAGINA
C2DD 05 AD          ORA $AD
C2DF 85 AD          STA $AD
C2E1 18             CLC
C2E2 A5 AC          LDA $AC
C2E4 65 63          ADC $63
C2E6 85 AC          STA $AC
C2E8 A5 AD          LDA $AD
C2EA 65 15          ADC $15
C2EC 85 AD          STA $AD
C2EE 60             RTS
C2EF 00 01 02 MULH .BYT 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
C2FD 11 12 14       .BYT 17,18,20,21,22,23,25,26,27,28,30,31
C309 00 40 80 MULL .BYT 0,$40,$80,$C0
C30D 00 00          TUSS .WOR 0000
C30F 00 00          FLG2 .WOR 0000
C311 00             BUFFER .BYT 00

```

Nu de listing voor de printer VC 1526:

```

C200             .OPT P1
C200             *= $C200
C200             ZF = $68
C200             Z3 = $69      ;BUFFER
C200             A  = $AC
C200             B  = A+1
C200             GRMEM = $2000
C200 20 F1 B7     JSR $B7F1
C203 86 67        STX $67
C205 20 0F F3     JSR $F30F    ;ZOEKT LOGISCH FILENUMMER
C208 20 1F F3     JSR $F31F    ;ZET FILEPARAMETER
C20B A9 05 EIGHT LDA #$05
C20D 20 2F C3     JSR $C32F    ;OPEN VOOR SA=5
C210 A9 06        LDA #$06
C212 20 2F C3     JSR $C32F    ;OPEN VOOR SA=6
C215 A2 16        LDX #$16
C217 20 C9 FF     JSR $FFC9
C21A A9 14        LDA #$14    ;REGELAFSTAND VOOR GRAFIEK
C21C 20 D2 FF     JSR $FFD2
C21F A9 0D        LDA #$0D    ;TEKEN VOOR EINDE
C221 20 D2 FF     JSR $FFD2
C224 20 CC FF     JSR $FFCC    ;KANAAL WISSEN
C227 A0 19        LDY #$19    ;AANTAL REGELS
C229 84 64        STY $64     ;ZETTEN
C22B A9 02 HA10   LDA #$02    ;DUBBELE HOOGTE
C22D 85 68        STA $68     ;HERHALINGSTELLER
C22F A9 00 HA10A  LDA #$00    ;TELLER VOOR SPATIES
C231 85 65        STA $65     ;INSTELLEN

```

C233	A9	28		LDA #\$28	;AANTAL KOLOMMEN
C235	85	63		STA \$63	;INSTELLEN
C237	A9	19		LDA #\$19	
C239	38			SEC	
C23A	E5	64		SBC \$64	;NOG TE PRINTEN REGELS
C23C	AA			TAX	
C23D	BD	38	C3	LDA \$C338,X	
C240	09	20		ORA #\$20	
C242	85	AD		STA \$AD	
C244	8A			TXA	
C245	29	03		AND #\$03	
C247	AA			TAX	
C248	BD	52	C3	LDA \$C352,X	
C24B	85	AC		STA \$AC	
C24D	A2	00	HA11	LDX #\$00	;GRAFIEKGEGEVENS HALEN
C24F	A0	08		LDY #\$08	
C251	A1	AC	HA1L2	LDA (\$AC,X)	
C253	99	56	C3	STA \$C356,Y	
C256	E6	AC		INC \$AC	
C258	D0	02		BNE \$C25C	
C25A	E6	AD		INC \$AD	
C25C	88		HA1L01	DEY	
C25D	D0	F2		BNE \$C251	
C25F	A0	08		LDY #\$08	
C261	A2	08	HA1L4	LDX #\$08	
C263	1E	56	C3 HA1L3	ASL \$C356,X	
C266	2A			ROL	
C267	CA			DEX	
C268	D0	F9		BNE \$C263	
C26A	20	0E	C3	JSR \$C30E	
C26D	99	5E	C3	STA \$C35E,Y	
C270	88			DEY	
C271	D0	EE		BNE \$C261	
C273	A0	08	HA1L02	LDY #\$08	;GRAFIEK PRINTEN
C275	20	A6	C2	JSR \$C2A6	
C278	A0	04		LDY #\$04	
C27A	20	A6	C2	JSR \$C2A6	
C27D	C6	63		DEC \$63	
C27F	D0	CC		BNE \$C24D	
C281	C6	68		DEC \$68	
C283	D0	AA		BNE \$C22F	
C285	C6	64	HA1L50	DEC \$64	;REGELTELLER VERLAGEN
C287	D0	A2		BNE \$C22B	;NOG NIET KLAAR
C289	A2	16		LDX #\$16	
C28B	20	C9	FF	JSR \$FFC9	;SA=6
C28E	A9	24		LDA #\$24	
C290	20	D2	FF	JSR \$FFD2	;REGELAFSTAND TEKST
C293	A9	0D		LDA #\$0D	
C295	20	D2	FF	JSR \$FFD2	;TEKEN VOOR EINDE

C298	20	CC	FF	JSR	\$FFCC	
C29B	A9	16		LDA	#\$16	:SA=6
C29D	20	C3	FF	JSR	\$FFC3	
C2A0	A9	19		LDA	#\$19	
C2A2	20	C3	FF	JSR	\$FFC3	;CLOSE COMMANDOKANAAL
C2A5	60			RTS		;KLAAR
C2A6	A9	00	PRDFCH	LDA	#\$00	
C2A8	85	66		STA	\$66	
C2AA	A2	15		LDX	#\$15	
C2AC	20	C9	FF	JSR	\$FFC9	;KANAAL VOOR TEKENDEFINITIE
C2AF	98			TYA		
C2B0	48			PHA		
C2B1	A2	04		LDX	#\$04	
C2B3	B9	5E	C3 HA62	LDA	\$C35E,Y	
C2B6	20	D2	FF	JSR	\$FFD2	
C2B9	20	D2	FF	JSR	\$FFD2	
C2BC	05	66		ORA	\$66	
C2BE	85	66		STA	\$66	
C2C0	88			DEY		
C2C1	CA			DEX		
C2C2	D0	EF		BNE	\$C2B3	
C2C4	20	CC	FF	JSR	\$FFCC	;KANAAL SLUITEN
C2C7	A5	66		LDA	\$66	
C2C9	F0	31		BEQ	\$C2FC	
C2CB	A6	67		LDX	\$67	
C2CD	20	C9	FF	JSR	\$FFC9	;GEGEVENSKANAAL
C2D0	A5	65		LDA	\$65	;AANTAL SPATIES
C2D2	F0	09		BEQ	\$C2DD	;GEEN
C2D4	AA			TAX		
C2D5	A9	20		LDA	#\$20	
C2D7	20	D2	FF HA64	JSR	\$FFD2	
C2DA	CA			DEX		
C2DB	D0	FA		BNE	\$C2D7	
C2DD	A9	FE	HA63	LDA	#\$FE	;STUURTEKENS
C2DF	20	D2	FF	JSR	\$FFD2	
C2E2	68		HA61A	PLA		
C2E3	C9	04		CMP	#\$04	
C2E5	D0	0A		BNE	\$C2F1	
C2E7	A5	63		LDA	\$63	
C2E9	C9	01		CMP	#\$01	
C2EB	D0	04		BNE	\$C2F1	
C2ED	A9	0D		LDA	#\$0D	
C2EF	D0	02		BNE	\$C2F3	
C2F1	A9	8D	HA75A	LDA	#\$8D	
C2F3	20	D2	FF HA75	JSR	\$FFD2	
C2F6	20	CC	FF	JSR	\$FFCC	;KANAAL SLUITEN
C2F9	E6	65		INC	\$65	
C2FB	60			RTS		
C2FC	A5	63	HA61	LDA	\$63	

```

C2FE C9 01      CMP #$01
C300 DO 08      BNE $C30A
C302 A6 67      LDX $67
C304 20 C9 FF    JSR $FFC9
C307 4C E2 C2    JMP $C2E2
C30A 68          HA61B PLA
C30B E6 65      INC $65
C30D 60          RTS
C30E A6 68      BSPLIT LDX $68      ;REGELFLAG
C310 EO 01      CPX #1      ;TWEDE REGEL AFDRUKKEN?
C312 FO 04      BEQ $C318      ;JA
C314 4A          LSR A
C315 4A          LSR A
C316 4A          LSR A
C317 4A          LSR A      ;BOVENSTE NIBBLE NAAR BENEDEN
C318 29 OF      HA90 AND #$OF      ;INDEX VAN TABEL OPBOUWEN
C31A AA          TAX
C31B BD 1F C3    LDA $C31F,X;
C31E 60          RTS
C31F 00          HATAB1 .BYT %00000000
C320 00          .BYT %00000011
C321 00          .BYT %00001100
C322 00          .BYT %00001111
C323 00          .BYT %00110000
C324 00          .BYT %00110011
C325 00          .BYT %00111100
C326 00          .BYT %00111111
C327 00          .BYT %11000000
C328 00          .BYT %11000011
C329 00          .BYT %11001100
C32A 00          .BYT %11001111
C32B 00          .BYT %11110000
C32C 00          .BYT %11110011
C32D 00          .BYT %11111100
C32E 00          .BYT %11111111
C32F 85 B9      OPENCH STA $B9
C331 09 10      ORA #$10
C333 85 B8      STA $B8
C335 4C C0 FF    JMP $FFC0
C338 00 01 02    MULH .BYT 0,1,2,3,4,5,6,7,8,10,11,12,13,15,16
C346 11 12 14    .BYT 17,18,20,21,22,23,25,26,27,28,30,31
C352 00 40 80    MULL .BYT 0,$40,$80,$C0
C356 00          BUFFER.BYT 00

```

Beide programma's gaan uit van een grafiekpagina vanaf \$2000. In deze geheugenruimte zijn de meeste grafische voorstellingen opgeslagen. Ook het al genoemde hulpprogramma uit 64 INTERN slaat ze daar op. Als een te printen grafische voorstelling zich in een andere adresruimte bevindt, kunt u zonder meer door een

verandering van GRMEM in de variabelendeclaratie een grafische voorstelling uit een andere adresruimte printen. Het grafiekgeheugen is zo georganiseerd dat de byte op adres \$2000 (=8192 decimaal) correspondeert met de acht horizontaal naast elkaar liggende grafiekpunten in de linker bovenhoek. Als er in deze byte een bit wordt gezet, verschijnt er op de aangegeven plaats een punt. Wanneer de bit laag is, verschijnt de achtergrondkleur. De volgende byte (\$2001) betreft de 8 punten daaronder. Tot en met de byte op adres \$2007 worden de bijbehorende bytes onder elkaar op het beeldscherm afgebeeld als horizontale puntenrijen. De byte met adres \$2008 levert bij gezette bits punten op in de bovenste puntenrij, naast die van adres \$2000:

\$2000	\$2008
\$2001	\$2009
\$2002	\$2010
\$2003	enzovoort
\$2004	
\$2005	
\$2006	
\$2007	

Telkens 8 bytes van 8 bits vormen een blok van 64 bits. De printer kan alleen verticale puntenrijen printen, dus moet een hardcopy-programma bits isoleren en tot een byte samenvoegen. Het probleem van de 7-naaldsprinter is dat per puntenreeks slechts 7 bytes behandeld kunnen worden. De achtste bit komt pas bij de volgende reeks aan de beurt. Nog een probleem is het feit dat de aanwezige 25 regels (200 puntenreeksen) niet door 7 deelbaar zijn zonder dat er een rest overblijft. Als de grafische voorstelling geprint wordt, blijven er aan het einde 4 puntenreeksen over, die afzonderlijk behandeld moeten worden. Deze problemen heeft een 8-naaldsprinter (zoals de Commodore 1526) natuurlijk niet. Hiermee kan men 200 puntenrijen prima in 25 printregels afdrukken, althans als de printer de mogelijkheid heeft om grafische plaatjes te printen. Nu heeft de Commodore 1526 een tamelijk onbekende mogelijkheid: hij kan een karakter in een 8x8-matrix printen. In de manual wordt daarvan een zeer oninteressant voorbeeld gegeven. De printer produceert 10 keer het Commodore-teken: geen toepassingsmogelijkheid die dagelijks nodig is. Het werken met vrij te definiëren karakters wordt bemoeilijkt omdat ze per regel maar 1 keer mogen voorkomen. Maar daarvoor bestaat een oplossing! De Commodore 1526 kent een speciaal teken dat een regel beëindigt maar geen line feed in werking stelt. Dat is het teken CHR\$(141) en het gaat hier om de waarde 13 + 128. CHR\$(13) is de waarde die anders een regel met line feed beëindigt. Maar CHR\$(141) is een CHR\$(13) met SHIFT, dat wil zeggen: het hoogste bit is gezet. Na dit teken kan het definieerbare teken opnieuw

worden gedefinieerd en geprint. Maar dan moet het programma wel een teller bezitten om aan te geven waar dit nieuwe teken geprint moet worden, want de printerkop staat weer op de eerste positie, ook als hij niet naar een nieuwe regel is gegaan. Om het teken op de juiste positie te krijgen, moet u er het juiste aantal spaties aan vooraf laten gaan. Deze uitvoer van grafiekgegevens is natuurlijk zeer tijdrovend: de data kunnen niet gewoon achter elkaar uitgevoerd worden. Ook de spaties (blanks), die in aantal toenemen naarmate u dichter bij het regeleinde komt en die u ook moet overbrengen, nemen nogal wat tijd in beslag. Maar in machinetaal is een redelijke snelheid te bereiken, in tegenstelling tot BASIC, waar een hardcopy veel tijd kost. Het programma regelt alle beschreven processen. Bovendien heeft het programma nog een extraatje voor de Commodore 1526: het vergroot grafische afbeeldingen met factor 2. Rekent u maar na: bij een matrix van 8x8 punten komt u uit op maximaal 640 punten (8 punten x 80 karakters) per printregel. De Commodore 64 geeft slechts 40 karakters, maar beschikt wel over de 8x8-matrix. De printer heeft dus het dubbele aantal punten in horizontale richting. Zou er een exacte copy gemaakt worden, dan zou het blad maar voor de helft worden benut. Maar door verdubbeling van de punten in horizontale en verticale richting kan het papier goed worden benut. Maar laten we ons met de BASIC-loaders gaan bezighouden. We geven u eerst het programma voor de 7-naaldsprinter en daarna het programma voor de bediening van de Commodore 1526.

```

100 FOR I = 49664 TO 49933
110 READ X : POKE I,X : S=S+X : NEXT
120 DATA 32,241,183,134,103,32,15,243,32,31,243,166
130 DATA 103,32,201,255,169,255,133,97,169,7,133,253
140 DATA 169,28,133,151,169,0,141,13,195,169,40,141
150 DATA 15,195,162,4,189,177,194,32,210,255,202,16
160 DATA 247,169,0,133,99,133,100,173,13,195,133,101
170 DATA 169,0,133,254,165,99,166,100,164,101,32,182
180 DATA 194,160,0,177,172,166,254,157,17,195,230,101
190 DATA 232,134,254,228,253,208,229,169,0,160,7,166
200 DATA 253,30,17,195,42,202,16,249,37,97,9,128
210 DATA 32,210,255,136,16,237,165,99,24,105,8,133
220 DATA 99,144,2,230,100,206,15,195,208,181,169,13
230 DATA 32,210,255,173,13,195,24,105,7,141,13,195
240 DATA 198,151,240,3,76,33,194,169,4,197,253,240
250 DATA 12,133,253,169,1,133,151,169,15,133,97,208
260 DATA 235,169,15,32,210,255,76,204,255,80,0,16
270 DATA 27,8,133,20,134,21,152,74,74,74,170,189
280 DATA 239,194,133,173,138,41,3,170,189,9,195,133
290 DATA 172,152,41,7,24,101,172,133,172,165,20,41
300 DATA 248,133,99,169,32,5,173,133,173,24,165,172
310 DATA 101,99,133,172,165,173,101,21,133,173,96,0

```

```

320 DATA 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13, 15
330 DATA 16, 17, 18, 20, 21, 22, 23, 25, 26, 27, 28, 30
340 DATA 31, 0, 64,128,192,203
350 IF S <> 31355 THEN PRINT "FOUT IN DATA !!" : END
360 PRINT "OK"

```

U ziet dat er nogal wat gegevens ingetikt moeten worden, maar met de proefsom worden fouten bij de invoer zeker vastgesteld. Men vraagt ons wel eens of de proefsom van de verschillende programma's wel klopt, omdat er tijdens het programma steeds fouten optreden. Ons antwoord is dat we de proefsom niet met pen en papier, maar met de Commodore 64 berekenen. Als je eenmaal zo'n apparaat hebt, moet je er ook gebruik van maken! Een klein programma zet ook het machinetaalprogramma om. U moet dus niet meteen de proefsom veranderen als u een fout in de gegevens signaleert. In 9 van de 10 gevallen zit er dan waarschijnlijk een fout in het programma dat u overgetypt hebt. Hier volgt de iets langere listing voor de printer Commodore 1526:

```

100 FOR I = 49664 TO 50006
110 READ X : POKE I,X : S=S+X : NEXT
120 DATA 32,241,183,134,103, 32, 15,243, 32, 31,243,169
130 DATA 5, 32, 47,195,169, 6, 32, 47,195,162, 22, 32
140 DATA 201,255,169, 20, 32,210,255,169, 13, 32,210,255
150 DATA 32,204,255,160, 25,132,100,169, 2,133,104,169
160 DATA 0,133,101,169, 40,133, 99,169, 25, 56,229,100
170 DATA 170,189, 56,195, 9, 32,133,173,138, 41, 3,170
180 DATA 189, 82,195,133,172,162, 0,160, 8,161,172,153
190 DATA 86,195,230,172,208, 2,230,173,136,208,242,160
200 DATA 8,162, 8, 30, 86,195, 42,202,208,249, 32, 14
210 DATA 195,153, 94,195,136,208,238,160, 8, 32,166,194
220 DATA 160, 4, 32,166,194,198, 99,208,204,198,104,208
230 DATA 170,198,100,208,162,162, 22, 32,201,255,169, 36
240 DATA 32,210,255,169, 13, 32,210,255, 32,204,255,169
250 DATA 22, 32,195,255,169, 25, 32,195,255, 96,169, 0
260 DATA 133,102,162, 21, 32,201,255,152, 72,162, 4,185
270 DATA 94,195, 32,210,255, 32,210,255, 5,102,133,102
280 DATA 136,202,208,239, 32,204,255,165,102,240, 49,166
290 DATA 103, 32,201,255,165,101,240, 9,170,169, 32, 32
300 DATA 210,255,202,208,250,169,254, 32,210,255,104,201
310 DATA 4,208, 10,165, 99,201, 1,208, 4,169, 13,208
320 DATA 2,169,141, 32,210,255, 32,204,255,230,101, 96
330 DATA 165, 99,201, 1,208, 8,166,103, 32,201,255, 76
340 DATA 226,194,104,230,101, 96,166,104,224, 1,240, 4
350 DATA 74, 74, 74, 74, 41, 15,170,189, 31,195, 96, 0
360 DATA 3, 12, 15, 48, 51, 60, 63,192,195,204,207,240
370 DATA 243,252,255,133,185, 9, 16,133,184, 76,192,255

```

```

380 DATA 0, 1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 13
390 DATA 15, 16, 17, 18, 20, 21, 22, 23, 25, 26, 27, 28
400 DATA 30, 31, 0, 64, 128, 192, 0
410 IF S <> 41908 THEN PRINT "FOUT IN DATA !!" : END
420 PRINT "OK"

```

Tenslotte nog een paar aanwijzingen voor het gebruik van de programma's en de printer Commodore 1526. Na het laden en starten van het programma dat bij uw printer hoort, kunt u elk gewenst programma starten. Als de grafische voorstelling op het beeldscherm staat en u wilt een hardcopy, dan kunt u de routine met:

```
SYS 12*4096+2*256,1f
```

oproepen. Het logisch filennummer, dat als extra parameter nodig is, moet van te voren in het programma aan de printer worden meegedeeld in een OPEN-instructie met device-nummer 4:

```

10
20
30 EEN WILLEKEURIG GRAFIEKPROGRAMMA
40 GOSUB 1000: REM HARDCOPY
50 HIER VERDER
.
.
.
999 END
1000 OPEN 1,4
1010 SYS 49664,1
1020 CLOSE 1
1030 RETURN

```

Als u het juiste programma hebt ingevoerd en de hardcopy-functie wordt niet uitgevoerd, kan dat komen doordat uw printer met het verwisselbare ROM is uitgerust. Dit verwisselbare ROM heeft naast de in de manual beschreven mogelijkheden een nieuwe modus. Door een kleine verandering in de hardware kan deze printer als een Commodore 1525 functioneren. Bepaalde stuurtekens en secundaire adressen van de Commodore 1526 worden dan genegeerd. Maar ook in de 1525-modus is het printen van grafische afbeeldingen niet mogelijk. Hoe kunt u nu bepalen welk bedrijfssysteem in de printer is toegepast? Dat kan op twee manieren. U kunt bijvoorbeeld de printer openmaken, de blikken afdekplaat verwijderen en daarna het EPROM van het bedrijfssysteem localiseren (een 28-polig IC met een sticker). Als er op dit IC 1526/C staat, dan is het vast en zeker de oude versie van het bedrijfssysteem. Deze kan kuren vertonen, maar zal de hardcopy

met het hier beschreven programma wel verwerken. Staat er 325341-06H, dan is dat een nieuwe versie van het bedrijfssysteem. U moet in dat geval controleren of het IC U4D (een 6532) met alle pennen goed in het voetje zit. De aanduiding U4D vindt u op de printplaat, aan de smalle kant van het IC. Als pootje 16 (vanaf links boven gerekend) niet in de sokkel steekt maar met een stukje draad aan de printplaat is verbonden, bevindt de printer zich in de Commodore 1525-modus. In dit geval werkt het programma niet. Maar u hoeft niet per se de printer open te maken om te ontdekken welke versie en welk bedrijfssysteem in de printer ingesteld zijn. Voert u eerst maar eens de self-test uit. Daartoe moet u het apparaat inschakelen en het mechanisme van het papiertransport vasthouden. In de eerste regel die dan geprint wordt, verschijnt onder andere de aanduiding "REV." en een nummer. Als dat 1.0 is, gaat het om het oude bedrijfssysteem. 7.0 is het nieuwe. U kunt met een simpele truc vaststellen of de printer in de 1526- of de 1525-modus werkt. Stuur via secundair adres 7 een tekst naar de printer. Als de printer reageert en de tekst uitvoert, is de Commodore 1525-modus door de zojuist beschreven brug aan IC U4D ingesteld. Als de printer niet print, functioneert hij in de Commodore 1526-modus en werkt onze hardcopy-routine. Natuurlijk kunnen er nog verschillende andere versies van het bedrijfssysteem in omloop zijn. Wij hebben ons hier beperkt tot de twee versies die het meest voorkomen.

5.4 Grafiek met of zonder l-naaldssturing

Grafische afbeeldingen programmeren is heel interessant, maar erg moeilijk. Het beste voorbeeld is het programmeren van grafische plaatjes met een hoge oplossing op de Commodore 64. Hiermee kunt u de meest geweldige dingen op het beeldscherm van de Commodore 64 toveren, als u maar weet hoe dat moet. Hetzelfde geldt voor de grafiekprogrammering van de printers. Een keurig geprinte grafische voorstelling als resultaat van enige uren programmeerwerk is altijd een bijzondere belevenis. In de volgende paragrafen willen we proberen u stap voor stap vertrouwd te maken met de mogelijkheden van het programmeren van grafische voorstellingen. Ook als u niet beschikt over een printer met speciale grafische mogelijkheden, kunt u veel van uw gading vinden in dit hoofdstuk. Het zal u verbazen wat met eenvoudige middelen allemaal bereikt kan worden. Eerst proberen we met de normale- of tekstmodus een paar keurige plaatjes te vervaardigen. We beginnen met tweedimensionale beelden, later zullen ook driedimensionale

afbeeldingen volgen. Daarna gaan we bezig met grafiek met hoog oplossend vermogen. De volgende programma's lopen op printers die geen bijzondere mogelijkheden hebben om de afbeelding te veranderen. Deze programma's lopen op alle printers die minstens 80 karakters per regel kunnen realiseren. Daardoor hebben ook de gebruikers van margrietwielprinters de gelegenheid de programma's te proberen en ermee te experimenteren. Ook willen we in de programma's ingaan op enkele bijzonderheden die bij printers vaker optreden, om de resultaten te verbeteren. U moet dus per geval bekijken of de gebruikte stuurtekens op uw printer beschikbaar zijn. Vaak zijn bij verschillende printers identieke modi door verschillende stuurtekens te bereiken.

5.4.1 Grafiek zonder 1-naaldssturing

Het eerste programma dat hier aan u wordt voorgesteld, is zo ongeveer de meest eenvoudige vorm. Het vervaardigt een sinuscurve, bestaande uit sterretjes, die verticaal over het papier loopt.

```
10 PI= 3.14159 : OPEN 4,4
20 FOR A= 0 TO 2*PI STEP .25
30 B=SIN(A)
40 C=B*39+40
50 FOR I=1 TO C :PRINT#4,CHR$(32);:NEXT
60 PRINT#4,"*"
70 NEXT A
80 CLOSE 4
```

Als u dit voorbeeld test, zal het resultaat u misschien wat teleurstellen. De gemaakte curve is alleen met wat fantasie als sinuscurve te herkennen. Maar als u een beetje met de parameters goochelt, kunt u vast wel een betere curve produceren. Vermindering van de stapgrootte in regel 20 leidt al tot een aanmerkelijk beter resultaat, hoewel de curve wel iets in de lengterichting wordt uitgerekt. Dit effect kan men bij veel printers verminderen door de regelafstand te verkleinen. Een verandering van de regelafstand bij 7-naaldsprinters wordt bereikt door regel 60 te veranderen:

```
60 PRINT#4,"*"; CHR$(8): PRINT#4,CHR$(15)
```

Deze printers kennen maar twee verschillende regelafstanden. Ten eerste de in de tekstmodus gebruikelijke afstand van 6 regels per inch. En ten tweede de afstand van 9 regels per inch die in de grafiekmodus wordt toegepast. In ons voorbeeld gebruiken we de tweede optie. Daartoe schakelen we de grafiekmodus in aan het einde van de regel, geven de instructie voor een line feed, en schakelen terug in de tekstmodus. De Commodore 1526 kan de regelafstand in stapjes van .1176 mm veranderen. De regelafstand wordt door een waarde gespecificeerd die via secundair adres 6 aan de printer wordt meegedeeld. Als u de regelafstand wilt verkleinen, moet u een waarde kleiner dan 36 kiezen. 36 is de defaultwaarde. Om de printregels meteen onder elkaar te kunnen printen, moet de regelafstand met

```
5 OPEN 6,4,4: PRINT#6,CHR$(24): CLOSE 6
```

worden veranderd. Helaas gebruiken printerfabrikanten verschillende stuurtekens om de regelafstand in te stellen. Als u een printer heeft die hier niet is beschreven, moet u het manual er maar eens op nalezen. Er is vast wel een geschikte mogelijkheid te vinden. Een hele stap in de goede richting is een printer die niet alleen de printpositie in complete karakters geeft, maar die ook posities van punten kan bepalen. Deze mogelijkheid vindt u doorgaans alleen bij matrixprinters, maar ook margrietwielprinters met proportioneel schriftbeeld hebben soms dergelijke mogelijkheden. Ook de Commodore printers 1515, 1525, enzovoort (de 7-naaldstypes dus) bezitten deze mogelijkheid, die maar zelden wordt benut. Daar deze printers 80 karakters in een 7x6 matrix kunnen realiseren, krijgen we een oplossend vermogen van $6 \times 80 = 480$ posities. Daarbij duikt het volgende probleempje op. Omdat we aan de printer alleen waarden van nul tot 255 in de vorm CHR\$(x) kunnen doorgeven, zouden we niet in staat zijn alle printposities te specificeren. Daarom moet de aanduiding van de positie in twee bytes plaatsvinden. Eerst wordt de printer door een CHR\$(27);CHR\$(16) op de ontvangst van printpositiecodes voorbereid. Daarna worden deze beide bytes in het formaat CHR\$(HIGHBYTE);CHR\$(LOWBYTE) aan de printer meegedeeld. Om op puntpositie 200 een * te realiseren, moet na een OPEN-instructie de regel

```
PRINT#1,CHR$(27);CHR$(16);CHR$(0);CHR$(200);"*"
```

aan de printer worden meegedeeld. Voor puntpositie 300 luidt de regel als volgt:

```
PRINT#1,CHR$(27);CHR$(16);CHR$(1);CHR$(44);"*"
```

In de laatste combinatie is het derde teken 1 en het vierde teken 44 (300-256). Zo kunnen alle 480 printposities worden gekozen. In

het volgende programma (een uitbreiding van de functieplotter) hebben we gebruik gemaakt van deze mogelijkheid. Bovendien heeft dit programma de mogelijkheid om afbeeldingen op schaal weer te geven en de mogelijkheid om de schaalverdeling zichtbaar te maken. Zo kunt u ook functies afbeelden waarvan de waarden buiten de grenzen van -1 tot 1 liggen, zonder van tevoren de plaats van de resultaten te weten. Met onderstaand programma kunt u al heel aardige grafische plaatjes maken.

```

100 REM *****
110 REM ***                                     ***
120 REM ***          FUNCTIEPLOTTER          ***
130 REM ***                                     ***
140 REM ***      MET SCHAAALAANDUIDING        ***
150 REM ***                                     ***
160 REM ***      VOOR 1515, 1525 ETC.        ***
170 REM ***                                     ***
180 REM *****
190 PI=3.1415926
200 GOSUB 450 : REM GRENZEN BEPALEN
210 OPEN4,4:OPEN1,4
220 FOR I=1 TO 80
230 PRINT#4,CHR$(249);
240 NEXT
250 PRINT#4,CHR$(8)
260 FOR A=0 TO 2*PI STEP .1
270 GOSUB 530
280 C=B*MY+240
290 HB=INT(C/256)
300 LB=C AND 255
310 IF C>=235 AND C<=245 THEN 390
320 IF C<=240 THEN 360
330 PRINT#4,CHR$(15)CHR$(27)CHR$(16)CHR$(0)CHR$(240)
    CHR$(245);
340 PRINT#4,CHR$(27)CHR$(16)CHR$(HB)CHR$(LB)"* "CHR$(8)
350 GOTO400
360 PRINT#4,CHR$(15)CHR$(27)CHR$(16)CHR$(HB)CHR$(LB)"* ";
370 PRINT#4,CHR$(27)CHR$(16)CHR$(0)CHR$(240)CHR$(245)
    CHR$(8)
380 GOTO 400
390 PRINT#4,CHR$(15)CHR$(27)CHR$(16)CHR$(HB)CHR$(LB)"* "
    CHR$(8)
400 NEXT A
410 PRINT#1,CHR$(15)
420 CLOSE1
430 END
440 REM SCALING
450 FOR A=0 TO 2*PI STEP .1
460 GOSUB 530

```

```

470 IF B>MA THEN MA=B
480 IF B<MI THEN MI=B
490 NEXT
500 IF ABS(MI)>MA THEN MY=233/ABS(MI):RETURN
510 MY=233/MA:RETURN
520 REM FUNCTIE IN REGEL 530 PLAATSEN
530 B=TAN(A)
540 RETURN

```

Het teken van de nullijn brengt nogal wat werk met zich mee. Afhankelijk van de berekende puntpositie moet beslist worden of de te printen punt rechts of links van de nullijn moet liggen, of dat deze punt de nullijn moet snijden. Als de nullijn en het sterretje samenvallen, verschijnt er geen nullijn. Voor het printproces worden alle waarden van de functie berekend. De resultaten daarvan zijn nodig om de afmetingen van de afbeelding vast te leggen. Omvangrijkere functies zullen nogal wat tijd in beslag nemen. Bij BASIC-programma's met veel rekenwerk moet u geduld hebben. Wie nog steeds niet tevreden is over de uitdraai moet maar eens in het hoofdstuk over hoogoplossende grafiek kijken. Daarin wordt een functieplotter beschreven die de mogelijkheden van de grafiekmodus benut en goede resultaten laat zien. Tot slot geven we een laatste tip voor het maken van grafieken met lage oplossingsgraad. Veel printers zijn in staat karakters in de zogenaamde compressed mode te printen. In deze modus worden in plaats van 80 meestal 132 tekens per regel uitgevoerd. Het is een feit dat de grafische afbeelding beter lukt naarmate er meer punten op een vlak worden geprint. Er is nog een methode om een beter resultaat te krijgen. Daartoe neemt u de vorige voorbeelden en vermenigvuldigt de betreffende waarden met de factor $132/80 = 1.65$. Deze tip geldt ook voor het volgende programma, waarmee we een driedimensionale grafiek proberen te maken. Als u het eerste programma van dit hoofdstuk hebt getest, denkt u misschien dat de afbeelding van een driedimensionale grafiek met sterretjes helemaal niets wordt. Dit programma leert anders.

```

100 DIM A$(80)
110 OPEN4,4
120 FOR X=1 TO 72
130 : H=0:L=120
140 : FOR I=1 TO 80:A$(I)=" ":NEXT
150 : FOR Y=1 TO 48
160 : : XT=(X-Y)*2
170 : : IF XT<1 THEN 290
180 : : IF XT>48 THEN 280
190 : : C=SQR((XT-24)*(XT-24)+(Y-24)*(Y-24))+1
200 : : W=XT+EXP(-.01*C*C)*45+3

```

```

210 : : IF W>H THEN 240
220 : : IF W<L THEN 260
230 : : GOTO 280
240 : : H=W
250 : : IF W>L THEN 270
260 : : L=W
270 : : A$(W)="*"
280 : NEXT Y
290 : FOR I=1 TO 80:PRINT#4,A$(I);:NEXT
300 NEXT X

```

Dit programma mag dan kort zijn, het is beslist niet eenvoudig! We zijn niet van plan het regel voor regel door te nemen. We geven u alleen aan waar u eventueel veranderingen kunt aanbrengen. Alles wat bij de vorige programma's is gezegd, geldt natuurlijk ook voor dit programma. Bij grotere regellengte (bijvoorbeeld 132 karakters per regel) moet u alle waarden met de genoemde factor 1.65 vermenigvuldigen. Dit geldt in het bijzonder voor de DIM-instructie aan het begin van het programma. Een uitzondering zijn waarde 2 in regel 160 en 0.01 in regel 200. Ook de beschreven maatregelen om de regelafstand te reduceren moet u in dit programma maar eens proberen. Om een andere oppervlakte af te beelden, kan de functie in regel 200 veranderd worden. Probeert u het eens met

```
200 W=XT-EXP(-.01*C*C)*45+25
```

Natuurlijk halen deze grafieken het lang niet bij Hi-Res-grafieken die met de l-naaldssturing zijn vervaardigd. Het aantal punten dat kan worden afgebeeld is daarvoor te klein. Een voordeel is natuurlijk dat je bij een geringer aantal punten ook minder berekeningen hoeft te maken. Daardoor wordt het programma alleen maar sneller. Vooral grote afbeeldingen kosten met de l-naaldsgrafiek verscheidene uren rekenwerk.

5.4.2 Grafiek met l-naaldssturing

In de vorige paragraaf hebben we geprobeerd met een paar trucs toch grafieken te produceren met printers die niet over bijzondere mogelijkheden beschikken. In deze paragraaf komen de printers aan bod die wel over een grafiekmodus beschikken. Door de drastische stijging van het aantal punten wordt de afbeelding een stuk fraaier. Deze paragraaf behandelt de programmering van dit soort

plaatjes. Hierna kunt u zelf ook zulke complexe driedimensionale afbeeldingen produceren als uw printer 1-naaldsprint kan leveren. De voorbeelden hebben betrekking op de printers 1515, 1525, GP 100 VC, MPS-801 en Epson met interface. Als u een andere printer bezit (bijvoorbeeld een Itoh), kunt u met behulp van uw manual zelf een aanpassing construeren. Eerst behandelen we de principes van de 1-naaldssturing. In de manuals wordt dit thema meestal oppervlakkig behandeld, zodat men met een heleboel vragen blijft zitten. En dan te bedenken dat het vaak de 1-naaldssturing is die de doorslag geeft bij de aankoop van een bepaalde printer. Om een printer in de grafiekmodus te kunnen hanteren, moet u enige elementaire kennis over de computer hebben. De VIC 20 en de Commodore 64 zijn zogenaamde 8-bits computers. Zoals u weet, is een bit de kleinste informatiedragende eenheid. Een bit kent maar twee toestanden: hij is gezet (aan) of hij is uit. Deze toestanden worden ook met waar-onwaar of met ja-nee aangeduid. Technici gebruiken graag de termen high (hoog) en low (laag), overeenkomstig de spanningspulsen die te meten zijn op de lijn van de bit. Tenslotte bestaat er nog de aanduiding 1 voor high en 0 voor low. Die cijfers worden toegepast bij het berekenen van de bitwaarden. Dat gaan wij nu ook doen, om de achtergronden van de grafische programmering beter te kunnen begrijpen. Laten we eens twee bits als voorbeeld nemen. Daarmee kun je al vier combinaties maken:

Bit 1	Bit 2
NEE	NEE
NEE	JA
JA	NEE
JA	JA

Als er nog een bit bijkomt, zijn er al 8 mogelijkheden. Vier bits leveren 16 combinaties op en 5 bits 32. Bij elke extra bit verdubbelen de mogelijkheden. Wanneer u de getallen precies bestudeert, kunt u vaststellen dat het aantal combinaties gemakkelijk berekend kan worden. U hoeft alleen het getal 2 met zichzelf te vermenigvuldigen en wel per beschikbare bit 1 keer. Anders gezegd: bij twee bits rekent u 2×2 , bij vier bits $2 \times 2 \times 2 \times 2$. U kunt deze bewerking ook beschrijven als twee tot de macht twee, respectievelijk twee tot de macht vier. Net als de computer gebruiken we het teken ^ (pijl omhoog) om machtsverheffen uit te drukken. Omdat uw computer met 8-bits waarden werkt, hebben we de beschikking over $2^6 = 256$ verschillende combinaties van gezette en gewiste bits. Dit zijn de waarden 0 tot 255. De nul wordt dus wel degelijk als waarde meegeteld. We krijgen nul als geen van de 8 bits is gezet. We nummeren deze bits van 0-7. Deze aanduiding is niet willekeurig gekozen. Laten we voor een beter begrip de

kleine tabel er nog eens bijhalen. We veranderen alleen de aanduiding van de bits:

Bit 1	Bit 0	
NEE	NEE	= 0
NEE	JA	= 1
JA	NEE	= 2
JA	JA	= 3

Belangrijk is dat elke gezette bit een bepaalde waarde vertegenwoordigt. Zo heeft een gezette bit 0 de waarde 1, een gezette bit 1 de waarde 2. Als beide bits zijn gezet, moet u hun waarden bij elkaar optellen. Als u aan de tabel nog een bit zou toevoegen, zou deze de waarde 4 krijgen. Ook hierbij geldt: de waarde van een bit wordt berekend door het getal 2 te verheffen met zijn aanduiding. Bijvoorbeeld: een gezette bit 2 betekent $2^2 = 4$. U berekent de waarde van een gezette bit 7 dus zo: $2^7 = 128$. In de volgende tabel worden alle mogelijke waarden opgesomd:

bit	7	6	5	4	3	2	1	0	
									----- $2^0 = 1$
								-----	$2^1 = 2$
								-----	$2^2 = 4$
								-----	$2^3 = 8$
								-----	$2^4 = 16$
								-----	$2^5 = 32$
								-----	$2^6 = 64$
								-----	$2^7 = 128$

								-----	255

U vraagt zich misschien af wat die hele verhandeling met de l-naaldssturing te maken heeft. Het antwoord is eenvoudig. Puntenreeksen worden onder elkaar geprint. Daarbij worden de verschillende naalden of punten net zo van waarden voorzien als de bits uit ons voorbeeld. We moeten hierbij de kanttekening plaatsen dat de getoonde volgorde van boven naar beneden geldt voor de printer 1515 en de typenummers daarboven. Bij Epson printers en enkele andere moet de volgorde omgedraaid worden: de onderste naald heeft waarde 1 en de bovenste naald waarde 128. Nu we de waardetoekenning van de verschillende punten duidelijk hebben gemaakt, gaan we iets printen. Om eenvoudig te beginnen, gaan we uit van een hypothetische printer, die beschikt over alle denkbare instructies en mogelijkheden. Met die ideale printer maken we een O-achtig karakter. We gaan ervan uit dat de bovenste puntenrij waarde 1 heeft.

***	1
* *	2
* *	4
* *	8
* *	16
* *	32
***	64

Gemakshalve hebben we de waarden van de punten meteen in de juiste regels geplaatst. Nu hoeven we alleen nog maar de waarden van elke kolom op te tellen. Omdat het uitgebeelde karakter zo mooi symmetrisch is, hoeven we alleen de waarden voor de eerste 3 kolommen te berekenen. De waarden herhalen zich namelijk. In de eerste kolom krijgen we de volgende som: $4+8+16=28$. In de tweede kolom komen we op: $2+32=34$ en in de derde kolom is het resultaat: $1+64=65$. Beide volgende kolommen krijgen ook 65, daarna volgt weer 34 en tenslotte 28. Nadat we zo de waarden hebben bepaald, komt de vraag wat we er mee moeten. Daar deze waarden als grafische afbeelding geprint moeten worden, moeten we de printer naar de grafiekmodus overschakelen. Bij onze ideale printer kon dat met de waarde 128:

```
PRINT#1,CHR$(128)
```

Nu kunnen we na elkaar van tevoren bepaalde waarden met PRINT-instructies naar de printer sturen. We zetten de waarden voor de grafiekbytes gemakshalve in DATA-regels. Een volledig programma zou er als volgt uit kunnen zien:

```
10 OPEN4,4
20 PRINT#4,CHR$(128): REM GRAFIEKMODUS INSCHAKELEN
30 FOR I= 1 TO 7 : REM AANTAL OVER TE DRAGEN GEGEVENS
40 READ A : REM HAALT 1 BYTE
50 PRINT#4,CHR$(A) : REM BYTE NAAR PRINTER
60 NEXT I
70 PRINT#4,CHR$(128): REM GRAFIEKMODUS UITSCHAKELEN
80 CLOSE 4
90 DATA28,34,65,65,65,34,28
```

Het hier beschreven programma functioneert alleen als de hypothetische printer 7 naalden in de grafiekmodus kan aansturen. Als er een achtste naald bij zou komen, zou het niet mogelijk zijn de grafiekmodus in- en uit te schakelen met code 128. Met deze waarde zou de achtste naald dan aangestuurd moeten worden. Om het probleem te omzeilen kunnen we gebruik maken van de volgende werkwijze, waarbij we 128 reserveren om de grafiekmodus in- en uit

te schakelen. In de grafiekmodus informeert het eerste teken de printer over het aantal bytes dat volgt. Als dit aantal bereikt is, komt de printer in de tekstmodus, als er geen maatregelen worden getroffen. In ons voorbeeldprogramma zou regel 70 daardoor in zijn geheel wegvallen. Regel 20 zou luiden:

```
20 PRINT#4,CHR$(128);CHR$(7)
```

U ziet dat er een behoorlijk verschil bestaat tussen de grafische modi van de 7- en de 8-naalds printers. Dit verschil is vooral interessant voor de eigenaars van 8-naaldsprinters (zoals Epson), die een 7-naaldsgrafiekprogramma willen aanpassen. We gaan ons nu bezighouden met de grafische modi van echte printers. We richten onze aandacht eerst op de 7-naaldsprinters. Bij deze printers bestaan er twee stuurtekens om de grafiek aan te sturen. Met code 8 wordt grafiek ingeschakeld en code 15 schakelt deze weer uit. Dat levert problemen op want bij ons hypothetische model waren de getallen 0-127 allemaal grafische tekens. De constructeurs hebben de volgende truc bedacht: bij elke grafiekbyte moet 128 worden opgeteld. Daardoor komen de grafische bytes tussen 128 en 255 te liggen. Code 15 kan dus ongestraft de tekstmodus herstellen. Code 8 vervult nog een functie. Hij vergroot het aantal te printen regels van 6 regels per inch (tekstmodus) tot 9 regels per inch. Hierdoor zijn we in staat grafische voorstellingen naadloos aan elkaar te plakken. Als u een Epson printer met de VIC 20 of Commodore 64 (al dan niet met een speciale interface) bedient, beschikt u over zeer produktieve mogelijkheden voor de aansturing van de 1-naaldsgrafiek. Omdat dit type printers in totaal 9 naalden bezit, hoeft u geen rekening te houden met de beperkingen van een 7-naaldsprinter. De grafische bytes worden in hun geheel (8 bits tegelijk) doorgestuurd en geprint. Een naald wordt hierbij niet gebruikt. Hier moeten we het volgende opmerken: de FX-80 kent een zogenaamde 9-naalds grafiekmodus, waarbij telkens twee grafiekbytes moeten worden overgedragen. Het paradepaardje onder de Epson printers, de LQ 1500, heeft een printkop met 24 naalden! Deze 24 naalden kunnen in speciale grafische modi allemaal aangestuurd worden. Daarmee kunnen grafische afbeeldingen gemaakt worden die er haast als foto's uitzien. De waardeverdeling bij Epson printers is precies tegengesteld aan die van Commodore printers. De naald met waarde 1 is de onderste en de naald met waarde 128 is de bovenste. Ook het aantal over te dragen grafiekbytes moet bekend zijn. Dat hoeft normaal gesproken geen problemen op te leveren, omdat de grafische gegevens berekend worden en ook het aantal resultaten in het programma bekend is. We gaan nu het eerste kleine programma bespreken. Het betreft weer een functieplotter. Functieplotters hebben het voordeel dat ze snel geprogrammeerd kunnen worden; de resultaten zijn heel behoorlijk. We tonen u twee versies van het programma, namelijk de versie voor Commodore printers en een tweede voor Epson

printers. Dankzij de vrij uitvoerige toelichting kunnen deze programma's gemakkelijk worden aangepast op grafische printers van bijvoorbeeld de merken Itoh of Seikosha. De nu volgende functieplotter is opzettelijk zo eenvoudig mogelijk gehouden om het meest wezenlijke, namelijk de aansturing van grafiek, de volle aandacht te geven.

```

10 REM
20 REM FUNCTIEPLOTTER
30 REM
40 REM PER NAALD GESTUURD
50 REM
60 REM VOOR COMMODOREPRINTERS MET 7 NAALDEN
70 REM
80 REM
90 PI = 3.14159
100 DEFFN A(X)=SIN(X)
110 INPUT"HORIZONTALE OPLOSSING";X%
120 IF X%<1 OR X%>479 THEN 110
130 INPUT"VERTICALE OPLOSSING";Y%
140 IF Y%<1 THEN 130
150 INPUT"DEFINITIEBEREIK ";A,B
160 A=A*PI:B=B*PI:IF A>=B THEN 150
170 INPUT"WAARDENBEREIK ";C,D
180 IF C>=D THEN 170
190 DIM Y%(X%)
200 VF=Y%/(D-C)
210 HF=(B-A)/X%
220 H=1
230 FOR I=6 TO 0 STEP -1
240 Y$(I)=CHR$(H+128):PRINT ASC(Y$(I))
250 H=H+H
260 NEXT
270 FOR I=1 TO X%
280 Y=FNA(A+HF*I)
290 Y%(I)=(Y-C)*VF+.5
300 NEXT
310 OPEN1,4
320 PRINT#1,CHR$(8);
330 A$=CHR$(128)
340 FOR H=Y%-6 TO 0 STEP -7
350 REM
360 FOR I=1 TO X%
370 Y=Y%(I)-H
380 IF Y<7 AND Y>=0 THEN PRINT#1,Y$(Y);:NEXT:GOTO410
390 PRINT#1,A$;
400 NEXT
410 PRINT#1
420 NEXT

```

```
430 PRINT#1,CHR$(15):CLOSE1
```

Hierna volgen nog de regels die bij gebruik van een Epson printer veranderd moeten worden. In ons voorbeeld zijn we uitgegaan van een Epson printer met DATA BECKER interface. Als u de printer bedient via de userport als Centronics printer, of als u een andere interface gebruikt, moet u de OPEN-instructie aanpassen.

```
10 REM
20 REM FUNCTIEPLOTTER VOOR MX/RX/FX
30 REM
120 IF X%<1 OR X%>960 THEN 110
230 FOR I=0 TO 7
310 OPEN1,4,1
320 PRINT#1,CHR$(27)CHR$(65)CHR$(8);
330 A$=CHR$(0)
340 FOR H=Y%-7 TO-7 STEP -8
350 PRINT#1,CHR$(27)CHR$(76)CHR$(X% AND 255) CHR$(X%/256);
380 IF Y<8 AND Y>=0 THEN PRINT#1,Y$(I);:NEXT:GOTO 410
430 PRINT#1: CLOSE1
```

In 100 wordt de functie gedefinieerd die geprint^e moet worden. U mag die functie naar eigen inzicht definiëren. Experimenteer er maar mee. Daarna volgen de belangrijkste parameters. De horizontale oplossing bepaalt de breedte van de grafische afbeelding op het papier. Als u op een snelle manier dingen wilt proberen, moet u hier kleine waarden (bijvoorbeeld 100) invoeren. Daardoor krijgt u een smalle grafische voorstelling, die vrij snel wordt uitgevoerd. Het maximum is 479, dat is namelijk het aantal mogelijke grafiekpunten. De verticale oplossing is niet begrensd. Toch moet u geen waarden invoeren die groter zijn dan de horizontale oplossing, omdat de grafiek daardoor in verticale richting een behoorlijke vertekening zou krijgen. Het definitiebereik bevat de waarden waarvoor de functie wordt berekend. Het waardenbereik bepaalt welk bereik in verticale richting wordt weergegeven. Als bij het uitprinten van bijvoorbeeld een sinuscurve de toppen niet meegeprint worden, moet u het waardenbereik iets ruimer maken. Na het invoeren van deze waarden worden de variabelen HF en VF berekend. Deze variabelen zijn nodig om de te plotten punten te bepalen. In regel 230 tot 260 worden de mogelijke bitpatronen berekend en in array Y\$(I) opgeslagen. Daarbij wordt bij 7-naaldsprinters meteen de hoogste bit gezet (+128), omdat de grafische gegevens bij deze printers tussen 128 en 255 liggen. Bij Epson printers treedt hier het eerste grote verschil op. Ten eerste zijn de naalden van beneden naar boven genummerd en bovendien zijn er 8 naalden in plaats van 7. Daarom zal de lus in regel 230 tot 260 bij Commodore printers 6 keer lopen en bij Epson printers 7 keer. Vanwege dit verschil in volgorde en aantal gaat de lus de ene keer van 6 naar 0 en de

andere keer van 0 naar 7. Bij een Itoh printer geldt dezelfde volgorde van de naalden als bij Commodore. De bovenste naald heeft waarde 1 en de onderste naald 128. Net als bij Commodore gaat de lus van hoog naar laag, in dit geval van 7 naar 0. Na de lus wordt de functie voor alle waarden berekend en in array Y% opgeslagen. Dat gebeurt in regel 270 tot 300. Daarna begint de printeruitvoer. Eerst wordt het printerkanaal geopend. De gegevens in de OPEN-instructie moeten natuurlijk aan de omstandigheden worden aangepast. Het printen gebeurt in twee lussen. De eerste lus is de regellus. Deze loopt vanaf de waarde van de verticale oplossing in Y% -6 naar -6 in stappen van 7, overeenkomstig de 7 naalden. Bij gebruik van een 8-naaldsprinter worden de betreffende waarden veranderd in regel 340. De tweede lus loopt van 1 tot aan de aangegeven waarde van de horizontale oplossing. Variabele I van deze tweede lus dient om de van tevoren berekende waarde van de functie te krijgen voor de punt op de X-as. Van deze functiewaarde wordt de waarde van de regellus afgetrokken. Als het resultaat binnen het bereik van 0 tot 6 ligt (of 0 tot 7 bij Epson), wordt deze waarde gebruikt om het karakter te printen dat de punt op de juiste positie plaatst. Deze puntpatronen werden immers van te voren in array Y\$ gedefinieerd. In alle andere gevallen wordt alleen waarde 128 naar de printer gestuurd om een lege grafische puntenrij aan te duiden. Bij Epson printers is dat CHR\$(0). In de hoop dat u de eerste experimenten met de 1-naaldsgrafiek hebt begrepen, willen we ons op de wat omvangrijkere grafische afbeeldingen storten. Het meest boeiend zijn driedimensionale grafieken. De programmering is wat moeilijker, maar door het groter aantal punten wordt het resultaat zeer fraai. Met de informatie die u in dit hoofdstuk hebt gekregen, moet u de programma's in de volgende paragraaf ook aan andere printers aan kunnen passen. We behandelen alleen nog Commodore printers.

5.4.3 Afbeelding van driedimensionale Hi-Resgrafiek

De weergave van ruimtelijke vormen is een uitdaging voor de programmeur. Heeft u een geschikte printer, dan moet u eigenlijk meteen aan de slag gaan. U moet wel wat elementaire wiskundige kennis bezitten om een geslaagd programma te kunnen maken. Het probleem bij een driedimensionale figuur is dat het papier maar twee dimensies heeft. De derde dimensie (de diepte) kan alleen met verschillende tinten zwart en grijs geproduceerd worden. De oplossing die daarvoor nodig is, kunt u alleen in de Hi-Res-modus,

dus met de 1-naaldssturing, op enigszins overtuigende wijze creëren. We hebben twee programma's ontwikkeld die op de printer heel aardige resultaten leveren. Om te beginnen een programma dat een kogel produceert. Als u de afbeelding op een afstand bekijkt, lijkt die al aardig op een echte driedimensionale kogel. Het tweede programma is te vergelijken met een driedimensionale functieplotter. U kunt er gekromde, golvende vlakken mee maken die een heel plastische indruk geven. Laten we eerst de kogel eens bekijken.

```

100 REM *****
110 REM **                               **
120 REM **      3-D  KOGEL              **
130 REM **                               **
140 REM *****
150 PRINTCHR$(147)
160 PRINT"KOGEL"
170 INPUT "DOORSNEDE IN PUNTEN (480 MAX.)";D
180 HD=D/2:VF=D/10
190 INPUT "PRINTER OF FLOPPY";D$
200 IF D$="P" THEN OPEN1,4:PRINT#1:PRINT#1,CHR$(8);
210 IF D$="F" THEN OPEN1,8,4,"O:KOGELGEGEVENS,S,W"
220 ZK=1/10000:R=1
230 FOR I=0 TO 6
240 MK(I)=R
250 R=R+R
260 NEXT
270 R=0
280 DIM WE%(D)
290 FOR Z=-1 TO 1 STEP 1/HD
300 ZZ=ZZ+1
310 PRINT ZZ;
320 R=R+1
330 MY=SQR(1-Z*Z)-ZK
340 Y=-MY
350 X=SQR(ABS(1-Y*Y-Z*Z))+ZK
360 A=SQR(ABS((1+Y*Y/X/X)*(1+Z*Z/X/X)))
370 S=1/A/VF
380 IF S<0 THEN S=0
390 Y=Y+S
400 IF Y>MY THEN WE%(HD+MY*HD)=WE%(HD+MY*HD)ORMK(R-1):GOTO430
410 WE%(HD+Y*HD)=WE%(HD+Y*HD) OR MK(R-1)
420 GOTO 350
430 IF R>6 THEN GOSUB 490
440 NEXT Z
450 IF R>0 THEN GOSUB 490
460 PRINT#1,CHR$(15)
470 CLOSE1
480 END

```

```

490 FOR Q=1 TO D
500 PRINT#1,CHR$(WE%(Q)+128);
510 WE%(Q)=0
520 NEXT:PRINT#1:R=0
530 RETURN

```

Na het starten van het programma wordt u gevraagd welke diameter de kogel moet hebben. Het maximum ligt bij 480, dat is het aantal punten per regel. Hoe kleiner de diameter, des te sneller is de afbeelding klaar. Bij een maximale diameter is de computer wel een paar uur bezig. In dat geval is het misschien beter uw computer 's avonds in te schakelen, dan hebt u de volgende ochtend het mooie resultaat. Niet iedereen is even gelukkig met het nachtelijk geratel van de printer. Een andere mogelijkheid is de berekende gegevens op schijf te zetten. Aldus opgeslagen data kunnen in een betrekkelijk korte tijd op de printer worden uitgevoerd. Zo kunt u ook meerdere identieke kogels achter elkaar laten printen, zonder er steeds tijdrovend rekenwerk aan te spenderen. Om de gegevens te printen kunt u het volgende kleine programma gebruiken:

```

100 OPEN 4,4
110 PRINT#4,CHR$(8)
120 OPEN1,8,4,"KOGELGEGEVENS,S,R"
130 GET#1,A$:IF ST=64 THEN 150
140 PRINT#4,A$::GOTO 130
150 PRINT#4,CHR$(15)
160 CLOSE4
170 CLOSE1

```

Laten we nog even het eigenlijke programma bekijken. Na de invoer van de diameter en het uitvoerapparaat worden uit de diameter enige benodigde gegevens berekend. Ten eerste de straal (= radius). Die bepaalt het middelpunt. Vervolgens wordt de variabele VF (voor VulFactor) berekend. Die bepaalt het aantal punten in het plaatje. Experimenteer maar eens met deze variabele. Daarna volgt het openen van het printer- of floppykanaal en de initialisering van de variabelen ZK (voor Zeer Klein) en R. ZK verhindert dat in regel 360 DIVISION BY ZERO optreedt. Variabele R staat voor de naaldenrij en is nodig aan het begin om de bitpatronen op de bekende wijze in array MK op te slaan. De eigenlijke berekening vindt plaats in de FOR-NEXT-lus in regel 290 tot 440. De regels 350 en 360 zijn verantwoordelijk voor de vorm van de afbeelding. Ook hier wordt variabele R weer meegenomen, die de actuele naaldenrij moet bepalen. Als alle punten voor de 7 naalden zijn berekend, worden ze naar het uitvoerapparaat gestuurd. Daarna worden de volgende punten berekend. Alleen variabele HD is belangrijk bij de aanpassing aan andere printers. HD bepaalt het aantal puntenrijen in de

FOR-NEXT-lus. Omdat niet alle printers een vierkant maken van een veld van 100x100 punten, wordt de afbeelding bij sommige printers eivormig. Afhankelijk van de verhouding tussen verticale en horizontale punten staat het ei op de punt of ligt het op zijn kant. Als dit verschijnsel optreedt, moet u variabele HD groter of kleiner maken, afhankelijk van de aard van de vervorming. Als HD kleiner wordt, wordt de kogel platter. Wordt HD groter, dan staat het ei op zijn punt. Alle andere aanpassingen hebben zoals bekend betrekking op de stuurtekens voor het inschakelen van de grafiek. Als bij de 8-naaldsprinter informatie over het aantal grafiekgegevens berekend moet worden, kunt u die informatie in variabele D (de diameter) terugvinden. U hebt immers het aantal punten in horizontale richting in deze variabele ondergebracht. Op den duur zullen kogels wel gaan vervelen, dus presenteren we ook het programma waarmee u elk gewenst object kunt maken. U kunt maximaal vier punten neerzetten, van waaruit golvende lijnen met verschillende lengtes en hoogtes lopen. Het aantal grafische mogelijkheden is onbegrensd. Net als alle andere BASIC-programma's waarin een aantal berekeningen staan die steeds weer herhaald worden, loopt dit programma tamelijk langzaam. Met een goede compiler kan de rekentijd aanzienlijk worden verkort. Overigens duurt een vol vlak met een spits iets langer dan twee uur, wanneer de gegevens op schijf staan. Net als het eerder besproken programma is ook dit programma geschikt voor alle 7-naaldsprinters. Maar het is niet moeilijk om dit programma aan te passen aan andere printertypes.

```

100 PI=3.14159
110 DIML%(480),NS(6),HM(4),XP(4),YP(4),WX(4),WY(4)
120 X=1
130 FOR I=0 TO 6
140 NS(I)=X
150 X=X+X
160 NEXT
170 PRINTCHR$(147)
180 PRINT:PRINT"          3D-FUNCTIEPLOTTER"
190 FOR I=0 TO 480
200 L%(I)=128
210 NEXT
220 I=0
230 PRINT:PRINT"          MEEST GESCHIKTE WAARDEN"
240 PRINT" WORDEN TUSSEN HAAKJES AANGEGEVEN"
250 PRINT:PRINT"          WILT U DE UITVOER"
260 PRINT:PRINT"          OP PRINTER OF FLOPPY DISK"
270 INPUT"          ";A$
280 IF A$="F" THEN OPEN1,8,4,"3D-DATEN,S,W" :GOTO310
290 IF A$="P" THEN OPEN 1,4:GOTO310
300 GOTO 270
310 PRINT#1:PRINT#1,CHR$(8);

```

```

320 PRINT:PRINT"          VOER DE PARAMETERS IN"
330 PRINT:PRINT
340 INPUT"AANTAL LIJNEN VERTICAAL          ";A$:LN%=VAL(A$)
350 MF%=(LN%*2/3):IF LN%<1 OR LN%>360 THEN 340
360 PRINT:INPUT"AANTAL TOPPEN  (1-4)";A$:NP%=VAL(A$)
370 IF NP%<1 OR NP%>4 THEN 360
380 PRINT:INPUT"LINKER RAND          (150)";A$:LM%=VAL(A$)
390 PRINTCHR$(147):GOTO 420
400 FOR I=1 TO NP%
410 PRINT CHR$(19)
420 PRINT:PRINT"  TOP NR."I:PRINT"  X POSITIE (0 TOT"MF%")";
430 IF FF=1 THEN INPUT A$:XP(I)=VAL(A$)
440 IF FF=0 THEN PRINT
450 PRINT:PRINT"  TOP NR."I:PRINT"  Y POSITIE (0 TOT"MF%")";
460 IF FF=1 THEN INPUT A$:YP(I)=VAL(A$):M%=MF%/7
470 IF FF=0 THEN PRINT
480 PRINT:PRINT"  TOP NR."I:PRINT"  X GOLFLENGTE  ("M%")";
490 IF FF=1 THEN INPUT A$:WX(I)=2*PI/VAL(A$)
500 IF FF=0 THEN PRINT
510 PRINT:PRINT"  TOP NR."I:PRINT"  Y GOLFLENGTE  ("M%")";
520 IF FF=1 THEN INPUT A$:WY(I)=2*PI/VAL(A$):M%=LM%+XP(I)-20
530 IF FF=0 THEN PRINT
540 PRINT:PRINT"  TOP NR."I:PRINT"  HOOGTE      (20 BIS"M%")";
550 IF FF=1 THEN INPUTA$:HM(I)=VAL(A$)
560 IF FF=0 THEN FF=1:GOTO400
570 NEXTI:PRINT
580 GN=1/10000:K=0:MM%=480
590 FOR I=1 TO LN%*1.25:PRINTI;
600 IL=2000:IH=-IL
610 FOR J=1 TO MF%
620 IF J>I THEN J=MF%:GOTO740
630 IX=(I-J)*2:IF IX>MF% THEN 740
640 X=IX:Y=J:Z=IX+LM%
650 FOR N=1 TO NP%
660 X1=(X-XP(N))*WX(N)+GN
670 Y1=(Y-YP(N))*WY(N)+GN
680 W=SQR(X1*X1+Y1*Y1)+GN:Z=Z-SIN( W)/W*HM(N)
690 NEXT N
700 IZ=Z
710 IF IZ<0 OR IZ>MM% GOTO 740
720 IF IZ>IH THEN IH=IZ:L%(IZ)=L%(IZ) OR NS(K)
730 IF IZ<IL THEN IL=IZ:L%(IZ)=L%(IZ) OR NS(K)
740 NEXT J
750 K=K+1
760 IF K>6 THEN GOSUB 820
770 NEXT I
780 IF K>0 THEN GOSUB 820
790 GOSUB 820

```

```

800 PRINT#1,CHR$(15):CLOSE1
810 END
820 FOR J=1 TO MM%
830 PRINT#1,CHR$(L%(J));:L%(J)=128
840 NEXT J:K=0:PRINT:PRINT#1
850 RETURN

```

Ook bij dit programma kunt u kiezen of de uitvoer meteen via de printer moet lopen, of dat die eerst op floppy wordt gezet. Als de gegevens eerst op floppy worden opgeslagen, kunnen ze met behulp van het korte laadprogramma (dat u nog van de kogel kent) worden geprint. U moet niet vergeten de juiste filenaam in te voeren, anders krijgt u de reactie FILE NOT FOUND. Bij dit programma moet u enkele parameters invoeren. We hebben de waarden en berekenen die het meest gunstig zijn. Eerst wordt u naar het aantal verticale printpunten gevraagd. Begin ter oefening met kleine waarden. Vijftig tot tachtig punten zijn voldoende om het resultaat te kunnen beoordelen. Bij de volgende invoer gaat het om het aantal toppen. Een top is het middelpunt van golflijnen, te vergelijken met een vijver waar een steen in is gevallen. Na de invoer van deze drie waarden verschijnt er een masker op het beeldscherm, waarin de parameters ingevoerd kunnen worden voor de posities van de toppen, de lengte van de golflijn en de hoogte van de toppen. Misschien staat u wat vreemd te kijken bij de vraag naar de golflengte in X- en Y-richting. Maar u kunt behalve cirkelvormige golflijnen ook ovale golflijnen uitvoeren. Als alle parameters zijn ingevoerd, beginnen we met het grote rekenkarwei. De factor 1.25 in regel 590 kan per printer variëren, afhankelijk van de relatie tussen horizontale en verticale printposities. In het minst gunstige geval worden de golven ovaal geprint, ondanks gelijke golflengten in X- en Y-richting. In regel 620 tot 670 worden de punten berekend. In regel 690 tot 720 wordt beslist of het om een zichtbare punt gaat of niet. Zo wordt ook het juiste bitpatroon uit array NS door OR met de eventueel al aanwezige waarde verbonden. Het printen van de grafiek in regel 810 tot 830 lijkt heel veel op de routine die bij de kogel werd gebruikt. Hiermee sluiten we het hoofdstuk over grafiek af. Met de beschreven methodes en een beetje fantasie kunt u vast ook zulke grafieken programmeren. Geef de moed niet te snel op als er een keer SYNTAX ERROR op het beeldscherm verschijnt.

6 PRINTERTYPES

Het was een lange weg van de eerste schrijfmachine naar de regelprinter. De opmars van de computer betekende dat berekeningen bliksemsnel konden worden uitgevoerd. Maar dat heeft weinig nut als de resultaten niet net zo snel op papier kunnen worden gezet. Papier is nog steeds de belangrijkste informatiebron. Nog niet eens zo lang geleden gebruikte men de telex om te printen. Veel computerfreaks doen dat nu nog vanwege de lage kosten. Een groot nadeel is de geringe snelheid: maar zes karakters per seconde. Er werden nieuwe printers ontworpen geschikt voor grote bedrijven en universiteiten. Het werk dat deze kasten van printers verzetten, was niet mis: maar liefst 600 regels per minuut. De printer stond op een speciale plaats en elke gebruiker moest zijn printwerk daar afhalen. De ontwikkeling waarvan u nu als gebruiker van een homecomputer profiteert, begon pas toen men de computer naar de arbeidsplaats toebracht door meerdere beeldschermen aan een centrale computer te koppelen. En waar een terminal staat, mag een printer niet ontbreken. Dus moesten er kleine, maar productieve printers komen. De trend zette door en de computer deed zijn intrede in de huiskamer. Daling van de prijs was noodzakelijk. De printer is daarbij niet achtergebleven.

6.1 De daisywheel- of margrietwielprinter

De werkwijze van deze printer is afgeleid van de bekende schrijfmachine met de kogelkop. Voorwaarde voor het gebruik van kettingformulieren is een transportmechanisme dat niet kan bewegen in horizontale richting. Maar het kogelkopmechanisme is zo gecompliceerd dat het meestal op een vaste plaats zit. De wagen met het papier gaat er langs. Bij uitzondering beweegt het bolletje, maar dat vereist een kostbaar geleidingsmechanisme. Bovendien is zo'n bolletje vrij duur. Verschillende lettertypes jagen u dan ook behoorlijk op kosten. Om die reden moest er een lichte en goedkope pendant worden ontwikkeld die gemakkelijk over een transportslede langs het papier kon lopen. De margrietwielprinter was geboren. Het principe is eenvoudig en doeltreffend: men neme een schijf met alle noodzakelijke letters, getallen en bijzondere karakters en monteert die op de as van een

kleine motor. Zo kunnen de segmenten met de karakters afzonderlijk tegen het papier worden gedrukt. De karakterblokjes zijn met verende spaken in een cirkel aan de as bevestigd. Ter hoogte van de printpositie bevindt zich een elektromagnetisch bewogen hamertje, dat het gewenste karakter samen met het lint tegen het papier drukt. Een stappenmotor kiest de karakters. Op de as van die motor is het dus bevestigd. Een gewone elektromotor begint te draaien zodra hij spanning krijgt aangevoerd. Als de stroom uitgeschakeld wordt, blijft hij een tijdje doorlopen voordat hij tot stilstand komt. Maar wij hebben een motortje nodig dat snel en exact het gewenste teken voor de hamer plaatst. Een stappenmotor beweegt zich bij elke stroompuls onder een precies gedefinieerde hoek. Als u de verdeling van de karakters op het margrietwiel eenmaal kent, kunt u precies berekenen na hoeveel pulsen het gewenste teken voor de hamer staat. Het geheel is op een slede gemonteerd, die ook door een stappenmotor wordt aangedreven. Zo kunt u de karakters in horizontale richting precies op hun plaats brengen. De kwaliteit van het schriftbeeld van een dergelijke printer is zeer goed. Het lettertype kunt u veranderen door een ander margrietwiel te gebruiken. Omdat het ronddraaien van het wiel toch nog wat tijd kost, is de margrietwielprinter niet het snelste systeem. Er is een duidelijke samenhang tussen printsnelheid en prijs. De prijs van een printer varieert van 1400 tot 8000 gulden.

6.2 De parallele matrixprinter

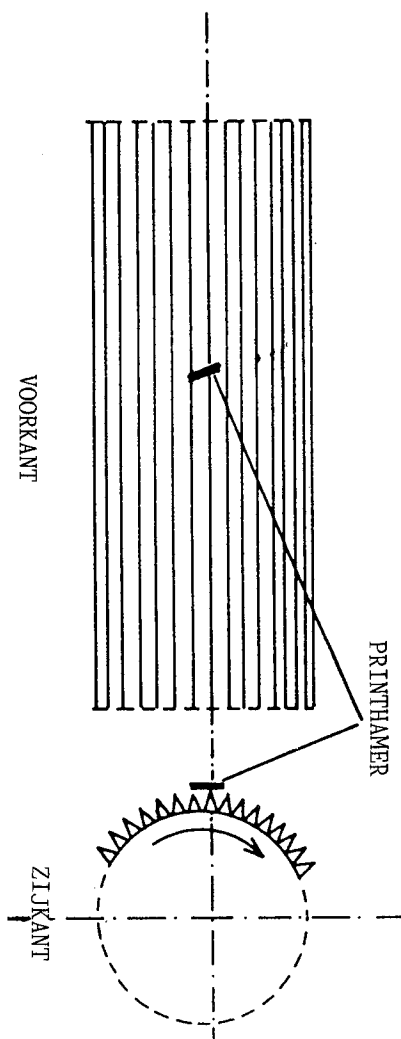
Er was behoefte aan een hogere printsnelheid en een lagere prijs. Het principe van de karakterblokjes stond vanwege het gewicht een grote snelheid in de weg. Een nieuwe weg moest worden bewandeld om zoveel mogelijk karakters met zo weinig mogelijk beweging te produceren. Het lag voor de hand de karakters samen te stellen uit zo min mogelijk segmenten. Daar de printerkop toch al over het papier beweegt, zijn alleen segmenten nodig voor de verticale doorsnede van een karakter. Dezelfde segmenten kunnen immers ook de volgende doorsnede(n) printen. Punten zijn in dit geval de meest efficiënte oplossing. Door een naald tegen lint en papier te drukken, krijg je een punt. Een naald is klein en licht en heeft dus een geringe massa traagheid waardoor snelle bewegingen mogelijk zijn. Het voordeel van deze methode is dat men niet gebonden is aan een vaste karakterset. U kunt nu elk gewenst karakter produceren, zoals u aan de l-naaldsgrafiek kunt zien, waarmee het merendeel van deze printers is uitgerust. De

karacterset is in het bedrijfssysteem van het apparaat verankerd. Hoe zoiets eruitziet, kunt u in paragraaf 7.3 nalezen. De karacterset wordt gewoon door de waarde van een teken geadresseerd, het puntpatroon wordt er uitgehaald en daarmee worden de naalden van de printerkop gestuurd. Op het papier verschijnt dan het bitpatroon uit het karaktergeheugen. Het aantal printnaalden per karakter ligt gewoonlijk tussen zeven en negen. Hoe meer naalden er worden gebruikt, des te fijner is het schriftbeeld. De printsnelheid ligt tussen 100 en 400 karakters per seconde. In de schoonschrijfmodus daalt de snelheid tot circa 40 karakters, omdat de printerkop dan twee keer over het papier beweegt. Bij de tweede ronde wordt het papier fracties van een millimeter verschoven. Het schriftbeeld wordt daardoor dichter. Bij enkele apparaten is dit systeem zo geperfectioneerd dat het resultaat nauwelijks van dat van een margrietwielprinter is te onderscheiden. De prijzen voor de parallelle matrixprinter schommelen tussen de 1000 en 4000 gulden. Dit type printer is zeer geschikt voor mensen die veel te printen hebben.

6.3 De seriële matrixprinter

Deze term was tot voor kort nog volkomen onbekend. Pas toen de prijs beneden de 1000 gulden kwam te liggen, kreeg dit apparaat meer bekendheid. De lage prijs (600-900 gulden) is te danken aan de vereenvoudiging van het apparaat. Ook in dit geval is het aantal segmenten gereduceerd. Als u niet per se een hele kolom in een keer wilt afdrukken, is 1 naald voldoende. Als u met 1 naald een hele kolom zou willen printen, zou of de kop of het papier in verticale richting bewogen moeten worden. Daardoor wordt het allemaal niet goedkoper. Maar bij een seriële matrixprinter gebeurt dat ook niet. Om de werking snel en duidelijk uit te leggen, schieten woorden tekort. Maar met het volgende plaatje moet het lukken.

Schema van de seriële matrixprinter



Seriële matrixprinters hebben een ribbelrol ter hoogte van de printhamer. Met papierloop heeft deze rol niets te maken. De printhamer is geen naald maar een metalen stripje met de lengte van een kolom. Er wordt een punt geprint als de hamer een ribbel van de rol raakt. Het lint en het papier zitten daartussen. De punt is het raakvlak van ribbel en hamer. Ze staan namelijk bijna haaks op elkaar. Er wordt per kolom geprint en wel van onder naar boven. De ribbelrol is gekoppeld aan de printerkop. Bij elke slag draait de rol van onder naar boven en schuift de hamer een klein stukje naar rechts. Als de hamer niet scheef stond (van rechtsonder naar linksboven), zouden de punten in een kolom niet netjes onder elkaar komen te staan. Bij deze methode hoeft noch de printerkop, noch het papier in verticale richting te worden bewogen om een kolom te printen. Het papier wordt pas doorgeschoven als de volgende regel aan de beurt is. De ribbelrol drijft een klokschijf aan, waarvan de sleuven bewegen in een lichtbundel. De op die manier verkregen puls is nodig om de processor te laten weten wanneer hij de hamer in werking kan zetten. Bij elke lichtpuls mag de hamer 1 keer slaan: de rol is dan een punt verder gedraaid. De printsnelheid is natuurlijk niet erg hoog omdat de printhamer veel vaker moet bewegen dan een naald van de parallelle printer. De snelheid ligt bij 50 tot 80 karakters per seconde. Voor mensen die af en toe printwerk hebben, is dit een apparaat waarvan de aanschaf het overwegen waard is.

7 HET BEDRIJFSSYSTEEM VAN DE MPS-801

De MPS-801 is een voorbeeld van een seriële matrixprinter. Het viel niet mee het bedrijfssysteem te bespreken. We moesten een disassembler schrijven voor de processor 8039 en de interpretatie van de instructies stelde ons voor problemen omdat we ons tot nu toe alleen maar hadden verdiept in de 6502/6510 en de Z80. Daarom is het commentaar af en toe wat cryptisch, waarvoor we bij voorbaat onze verontschuldiging aanbieden. Om het de 6502-specialist iets gemakkelijker te maken, behandelen we eerst wat bijzonderheden van de processor 8039. Ook gaan we in op instructies die u niet zonder meer uit de machinetaal van de 6502 kunt afleiden. Vervolgens vertellen we iets over de hardware van de printer aan de hand van het blokschema en bespreken de instructies die nodig zijn om de hardware te bedienen.

7.1 De processor 8039

De 8039 is een single chip processor. Hij bevat praktisch alles wat nodig is om eenvoudige hardwarecomponenten te sturen. Ondersteunende bouwstenen zoals RAM of I/O-ports zijn overbodig. Het enige wat hij echt nodig heeft, is het externe programmeergeugen en driverbouwstenen, om motoren en dergelijke te kunnen sturen. We geven u in 't kort een paar gegevens:

- * 128 bytes RAM
- * teller/tijdsaanduiding
- * 16 I/O-lijnen
- * 2 gescheiden testingangen

Behalve de compactheid zijn er ook wat eigenaardigheden. Een probleem is dat maar 2K programmeergeugen geadresseerd kan worden. Als er nog een ruimte van 2K geadresseerd moet worden, moet men al overschakelen. Het maximum is 4K. Een andere beperking is dat de stack (stapel) slechts 8 nivo's kent. Het nesten van routines is daardoor slechts in beperkte mate mogelijk. Zoals u zult zien, wordt daar ook weinig gebruik van gemaakt. We introduceren nu een paar instructies die u vast nog niet kent:

Spronginstructies:

jt0 of jnt0: springen, als lijn T0 hoog of laag is
jtl of jntl: als boven, maar voor lijn T1
jni : springen, als lijn INT=laag
jf0 : springen, als flag 0 gezet is
jfl : als boven, maar nu voor flag 1

De flags worden als merktekens in het programma gebruikt. Ze worden gewist met clr f0 of met clr fl. Ze worden gezet met cpl f0 of cpl fl. Alle spronginstructies, behalve jmp en call, kunnen alleen binnen een page springen. Daarom zijn hun operanden ook maar 1 byte lang. Stel, u hebt de volgende instructie:

```
OABF jnz 00
```

De sprong leidt niet naar 0000, maar naar 0A00, als de inhoud van de accu niet gelijk is aan nul. Page 0A van ons voorbeeld kan met dergelijke sprongen niet verlaten worden. Dit ligt anders bij jmp en call. Die kunt u binnen een bereik van 2K gebruiken. Als u dat bereik wilt verlaten, moet u het geheugen van tevoren op de volgende wijze omschakelen:

```
0000 sel mbl  
0001 jmp 000
```

Deze sprong gaat niet naar 0000, maar naar 0800, want het geheugen werd van tevoren omgeschakeld. Met mb0 kiest u het bovenste geheugenbereik, met mbl het onderste. De omschakeling werkt pas bij de volgende instructie, omdat anders de jmp uit ons voorbeeld niet meteen uitgevoerd zou worden. We hebben de doeladressen aangepast, zodat u bij het lezen van de ROM-listings niet bij dergelijke instructies hoeft te controleren of er misschien een sel mb0 of sel mbl voor staat.

Transferinstructies:

U kunt alles uit de accumulator halen en er naartoe brengen. Immediate-operanden (altijd voorafgegaan door #) kunnen overal naar toe worden gebracht. Als aan een variabele in de vorm van een operand een apestaart (@) voorafgaat, is hij niet de operand zelf, maar verwijst hij naar het adres van de operand. Dit is noodzakelijk als het RAM wordt aangesproken behalve wanneer het om het registerbereik gaat.

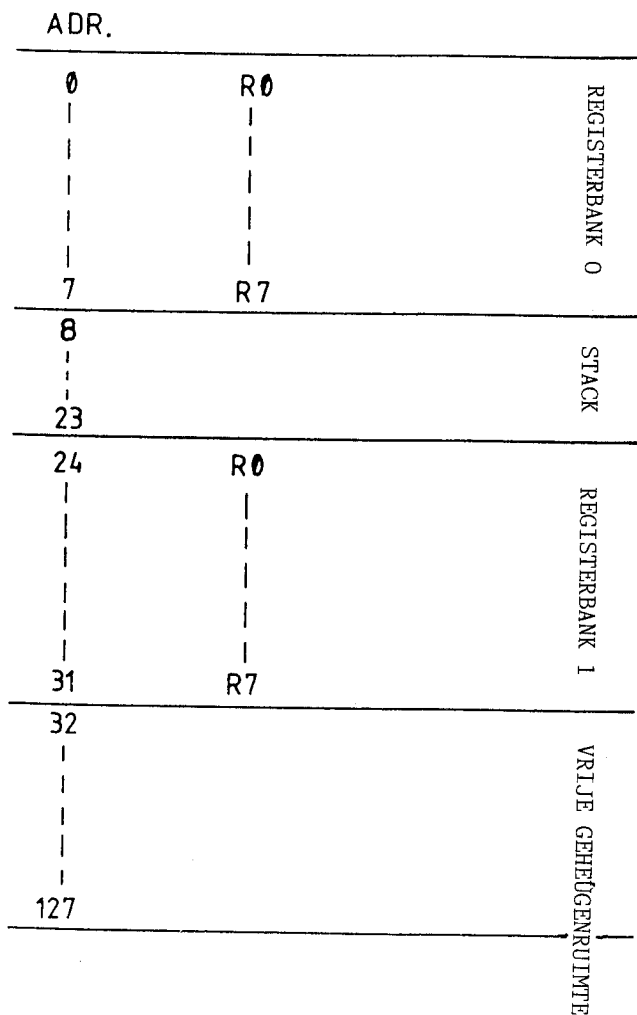
De organisatie van het geheugen:

Zoals u aan de volgende memory-map kunt aflezen, is het geheugen opgedeeld in twee registerbanks, een stack en het overige geheugen. Het geheugen kan op alle posities indirect worden geadresseerd. In het registerbereik is dat ook direct mogelijk.

```
0000 mov r1,#03
0002 sel rbl
0003 mov r1,#04
```

De eerste instructie laadt geheugencel 1. De derde laadt geheugencel 25, omdat er van tevoren op registerbank 1 werd overgeschakeld. Met sel rb0 kan weer worden teruggesprongen.

Memory-map van de 8039



De ruimte 34...123 wordt in de MPS-801 als regelbuffer gebruikt.

Vergelijkingsinstructies kent de 8039 niet. Er kan alleen maar indirect vergeleken worden en wel als volgt. Bij de vergelijkingsoperand wordt het complement van de eenheden opgeteld en er wordt gecontroleerd of het resultaat nul is. Stel, u hebt in de accu een waarde staan waarvan u wilt weten of die gelijk is aan 3. Dat doet u op deze manier:

```
0123 add a,#FD
0125 jz 40
```

Als in de accu nu de waarde 3 had gestaan, zou er gesprongen zijn. Deze methode is wat omslachtig, maar werkt wel. Daarbij willen we het laten. Voorbeelden en toelichtingen voor de I/O-programmering staan in de volgende paragraaf.

7.2 Blokschema van de MPS-801

De getoonde afbeelding is praktisch compleet. Voor de overzichtelijkheid hebben we alleen het programmeergeheugen en de stuurschakelingen voor de motoren en dergelijke weggelaten. De signaalnamen worden nu kort uitgelegd. De signalen aan de linkerkant behoren tot de IEC-bus, die we in paragraaf 2.1 behandelden. De signalen aan de rechterkant sturen de hardwarecomponenten naar de printer.

TEST Dit signaal is afkomstig van een schakelaar aan de achterkant van de printer. Als die in de positie TEST staat, is deze lijn=laag.

4/5 Ook dit is een signaal van een schakelaar. Het is laag als de schakelaar op device-nummer 5 staat. In andere gevallen is het hoog.

LICHTPULS Dit is het signaal van de lichtpuls van de klokschijf (zie paragraaf 6.3).

BASISPOSITIE Als de printkop op de linkerkantlijn staat, onderbreekt hij daar een lichtstraal. Als dat het geval is, is dit signaal=hoog.

LF-TOETS Als de LF-toets op de bovenkant van het apparaat wordt ingedrukt, is deze lijn=laag.

KOPMOTOR AAN Is deze uitgang low dan wordt het transportmechanisme voor de printkop in beweging gezet. Tegelijkertijd gaan ook de ribbelrol en de klokschijf draaien.

TERUGLOOP VAN DE WAGEN AAN Als deze lijn low is, wordt de printkop van het aandrijfmechanisme losgekoppeld. Een veer herstelt de basispositie.

LINE FEED AAN Dit is de hoofdstroomvoorziening van de stappenmotor voor het regeltransport. De motor gaat pas echt bewegen door een rondlopend signaal op de lijnen LF1, LF2, LF3 en LF4.

PRINTHAMER AAN Is deze lijn low dan staat de printhamer op het papier.

Programmering van de portlijnen:

de signaaluitgangen worden doorgaans direct gestuurd. Een laag (low) wordt door een logische AND opgewekt en een hoog (high) door een logische OR. Bijvoorbeeld:

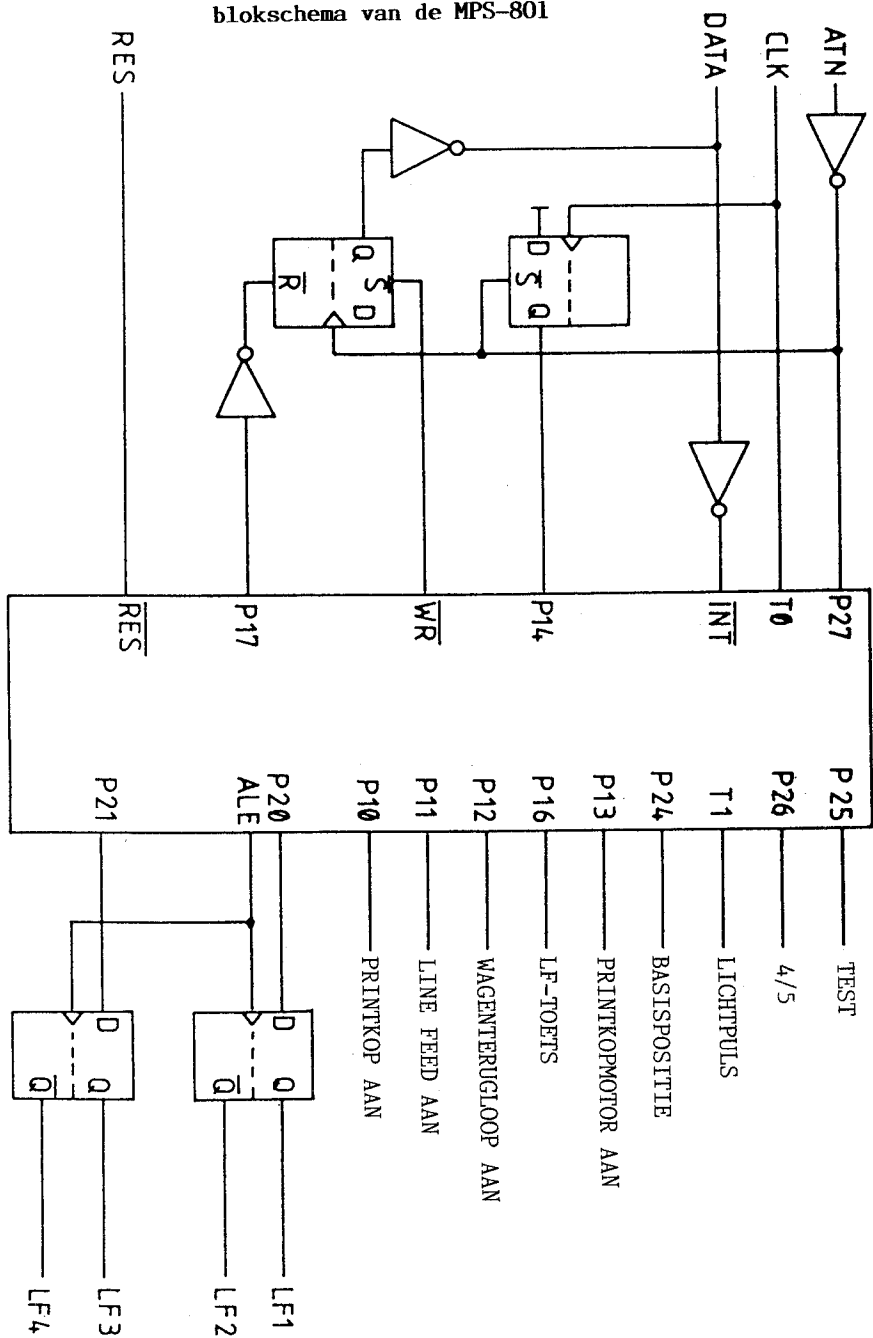
```
0011 anl pl,#FE
0013 orl pl,#01
```

Door deze serie instructies zou de printhamer kort in werking worden gesteld. De toestand van de portlijnen kan worden vastgesteld door ze in de accu te halen en vervolgens de gewenste bits te testen.

```
0025 mov a,p2
0026 jb6 47
```

Met deze sequentie wordt een sprong naar 0047 uitgevoerd als de adresschakelaar op 4 staat. De lijnen T0 en T1 worden met de reeds beschreven sprongen jt0, jtl, jnt0 en jntl gecontroleerd.

blokschema van de MPS-801



7.3 De ROM-listing voor de MPS-801

```

0001 00    NOP
0002 00    NOP
0003 00    NOP
0004 00    NOP
0005 00    NOP
0006 00    NOP
***** FOUTENSPRONG
0007 65    STOP TCNT
0008 35    DIS TCNTI
0009 15    DIS I
000A 89FF  ORL P1,#FF (255)
000C 8AFC  ORL P2,#FC (252)
000E 040E  JMP OOE      DEAD LOCK
0010 89FF  ORL P1,#FF (255)
0012 8AFC  ORL P2,#FC (252)
0014 15    DIS I
0015 E5    SEL MBO
0016 C5    SEL RBO
0017 65    STOP TCNT
0018 35    DIS TCNTI
0019 99FD  ANL P1,#FD (253)
001B F5    SEL MB1
001C D437  CALL 637      DELAY
001E E5    SEL MBO
001F 89FF  ORL P1,#FF (255)
*****
0021 27    CLR A      REGISTERBLOK 0
0022 B802  MOV RO,#02 (2) EN
0024 B916  MOV R1,#16 (22) GEHEUGEN
0026 A5    CLR F1      WISSEN
0027 A0    MOV @RO,A
0028 18    INC RO
0029 E927  DJNZ R1,27
*****
002B 0A    IN A,P2      ATN?
002C F22B  JB7 2B      JA, WACHTEN
002E 997F  ANL P1,#7F (127) DATA VRIJGEVEN
0030 BC01  MOV R4,#01 (1)
0032 5400  CALL 200      TRANSPORT VAN DE PRINTKOP EN
0034 B811  MOV RO,#11 (17) LICHTPULS TESTEN
0036 F0    MOV A,@RO
0037 53FC  ANL A,#FC (252)
0039 A0    MOV @RO,A
003A 5402  CALL 202      PRINTKOP IN
003C 99FB  ANL P1,#FB (251) BASISPOSITIE
003E B4CD  CALL 5CD      BRENGEN

```

```

0040 B810 MOV R0,#10 (16)
0042 B018 MOV @R0,#18 (24)
*****
0044 27 CLR A REGISTERBLOK 1
0045 B816 MOV R0,#16 (22)
0047 A0 MOV @R0,A EN
0048 18 INC R0
0049 A0 MOV @R0,A GEHEUGEN
004A B81A MOV R0,#1A (26)
004C B966 MOV R1,#66 (102) WISSEN
004E A0 MOV @R0,A
004F 18 INC R0
0050 E94E DJNZ R1,4E
*****
0052 85 CLR FO
0053 AA MOV R2,A REGISTERBLOK 0
0054 AB MOV R3,A
0055 AD MOV R5,A TERUGZETTEN
0056 AE MOV R6,A
0057 B821 MOV R0,#21 (33)
0059 B002 MOV @R0,#02 (2)
005B 18 INC R0
005C FC MOV A,R4
005D 37 CPL A
005E B008 MOV @R0,#08 (8)
0060 1268 JBO 68
0062 B00F MOV @R0,#0F (15)
0064 F268 JB7 68
0066 B00E MOV @R0,#0E (14)
0068 18 INC R0
0069 B918 MOV R1,#18 (24)
006B B100 MOV @R1,#00 (0)
006D 2379 MOV A,#79 (121)
006F 5C ANL A,R4
0070 AC MOV R4,A
0071 F8 MOV A,R0
0072 AF MOV R7,A
0073 E400 JMP 700 EINDE RESET
***** SPRONG VANUIT LISTEN MET
DATA BYTE IN R2
0075 FE MOV A,R6
0076 C67A JZ 7A
0078 4427 JMP 227
007A FC MOV A,R4
007B 128E JBO 8E
007D FA MOV A,R2
007E F282 JB7 82 DATA BYTE > 127? JA -->
0080 2400 JMP 100
0082 B821 MOV R0,#21 (33)

```

```

0084 B000 MOV @R0,#00 (0)
0086 BB01 MOV R3,#01 (1)
0088 B918 MOV R1,#18 (24)
008A B100 MOV @R1,#00 (0)
008C 04B5 JMP 0B5
*****
008E FA MOV A,R2
008F 5360 ANL A,#60 (96) AFDrukbaar Tekens?
0091 9695 JNZ 95 JA -->
0093 2400 JMP 100
0095 FA MOV A,R2
0096 03BF ADD A,#BF (191) >64?
0098 F6A4 JC A4 JA -->
009A 031F ADD A,#1F (31) Aanhalingstekens?
009C 96A2 JNZ A2 NEE -->
009E FC MOV A,R4 QUOTE-
009F D302 XRL A,#02 (2) FLAG
00A1 AC MOV R4,A FLIPPEN
00A2 0482 JMP 082
00A4 FC MOV A,R4
00A5 72AE JB3 AE
00A7 FA MOV A,R2
00A8 F5 SEL MB1
00A9 9400 CALL COO
00AB E5 SEL MBO
00AC 0482 JMP 082
00AE FA MOV A,R2
00AF F5 SEL MB1
00B0 34D8 CALL 9D8
00B2 E5 SEL MBO
00B3 0482 JMP 082
***** BYTE IN BUFFER OPSLAAN
00B5 FF MOV A,R7
00B6 A8 MOV R0,A
00B7 FA MOV A,R2
00B8 A0 MOV @R0,A
00B9 1F INC R7 BUFFERTELLER +1
00BA FE MOV A,R6
00BB 96CC JNZ CC
00BD FB MOV A,R3
00BE F2CE JB7 CE
00C0 2384 MOV A,#84 (132)
00C2 6F ADD A,R7 BUFFERTELLER > 123?
00C3 F6D4 JC D4

```

```

00C5 D5 SEL RB1
00C6 F8 MOV A,RO
00C7 C5 SEL RBO
00C8 03FB ADD A,#FB (251)
00CA F6D4 JC D4
00CC E400 JMP 700 TERUG NAAR WACHTLUS
00CE B820 MOV RO,#20 (32)
00D0 FB MOV A,R3
00D1 A0 MOV @RO,A
00D2 04E9 JMP OE9
*****
00D4 D5 SEL RB1
00D5 F8 MOV A,RO
00D6 C5 SEL RBO
00D7 E7 R1 A
00D8 E7 R1 A
00D9 E7 R1 A
00DA E7 R1 A
00DB 5370 ANL A,#70 (112)
00DD A8 MOV RO,A
00DE FC MOV A,R4
00DF 538F ANL A,#8F (143)
00E1 48B8 JNT1 B8
00E3 20 XCH A,@RO
00E4 A0 MOV @RO,A
00E5 FF MOV A,R7 LINE FEED
00E6 A8 MOV RO,A ACHTER BUFFER
00E7 B00A MOV @RO,#0A (10) ZETTEN
00E9 2378 MOV A,#78 (120)
00EB 5C ANL A,R4
00EC 4301 ORL A,#01 (1)
00EE AC MOV R4,A
00EF BF22 MOV R7,#22 (34) BUFFERWIJZER TERUGZETTEN
00F1 44AD JMP 2AD BUFFER PRINTEN

```

00F3 00 00 00 00 00 00 00 00 00 00 00 00 00

***** INTERPRETATIE VAN STUURTEKENS

```

0100 FC MOV A,R4
0101 37 CPL A QUOTE MODUS?
0102 3207 JB1 07 NEE -->
0104 F5 SEL MB1
0105 04C3 JMP 8C3
0107 23F2 MOV A,#F2 (242) WIJD GEDRUKT AAN?
0109 6A ADD A,R2
010A C638 JZ 38 JA -->
010C 07 DEC A WIJD GEDRUKT UIT?
010D C638 JZ 38 JA -->
010F 07 DEC A PRINTPOSITIEBESTEMMING?

```

0110	C669	JZ	69		JA -->
0112	07	DEC	A		TEKSTMODUS AAN?
0113	C673	JZ	73		JA -->
0115	07	DEC	A		REVERSE AAN?
0116	C682	JZ	82		JA -->
0118	23F8	MOV	A,#F8	(248)	
011A	6A	ADD	A,R2		BIT IMAGE MODE?
011B	C65F	JZ	5F		JA -->
011D	23E5	MOV	A,#E5	(229)	
011F	6A	ADD	A,R2		PRINTSTARTPOSITIE (27)?
0120	C69C	JZ	9C		JA -->
0122	17	DEC	A		
0123	C6A1	JZ	A1		
0125	FA	MOV	A,R2		
0126	036F	ADD	A,#6F	(111)	GRAFIEKMODUS?
0128	C6A8	JZ	A8		JA -->
012A	07	DEC	A		REVERSE UIT?
012B	C6B2	JZ	B2		JA -->
012D	23F6	MOV	A,#F6	(246)	
012F	6A	ADD	A,R2		LINE FEED?
0130	C6BE	JZ	BE		JA -->
0132	03FD	ADD	A,#FD	(253)	CR?
0134	C6BE	JZ	BE		JA -->
0136	E400	JMP	700		TERUG NAAR WACHTLUS
*****					STUURTEKENS 14/15
0138	B810	MOV	R0,#10	(16)	
013A	B018	MOV	@R0,#18	(24)	
013C	2301	MOV	A,#01	(1)	
013E	4C	ORL	A,R4		
013F	AC	MOV	R4,A		
0140	B921	MOV	R1,#21	(33)	
0142	F1	MOV	A,@R1		
0143	325A	JB1	5A		
0145	124C	JBO	4C		
0147	D5	SEL	RB1		
0148	18	INC	R0		
0149	C5	SEL	RBO		
014A	245B	JMP	15B		
014C	B4F0	CALL	5F0		DE LAATSTE DRIE
014E	B4F0	CALL	5F0		BYTES VAN DE BUFFER
0150	B4F0	CALL	5F0		UITVULLEN MET NULLEN
0152	F1	MOV	A,@R1		
0153	F257	JB7	57		
0155	245B	JMP	15B		
0157	D5	SEL	RB1		
0158	C8	DEC	R0		
0159	C5	SEL	RBO		
015A	CF	DEC	R7		
015B	B102	MOV	@R1,#02	(2)	

```

015D 04B5 JMP OB5
***** BIT IMAGE MODE AAN
015F B810 MOV RO,#10 (16)
0161 B010 MOV @RO,#10 (16)
0163 FC MOV A,R4
0164 537C ANL A,#7C (124)
0166 AC MOV R4,A
0167 2440 JMP 140
***** STUURTEKEN 16
0169 BD02 MOV R5,#02 (2)
016B BB01 MOV R3,#01 (1)
016D D5 SEL RB1
016E 18 INC RO
016F C5 SEL RBO
0170 1E INC R6
0171 04B5 JMP OB5
***** TEKSTMODUS AAN
0173 FC MOV A,R4
0174 1278 JBO 78
0176 E400 JMP 700
0178 B814 MOV RO,#14 (20)
017A 2301 MOV A,#01 (1)
017C 40 ORL A,@RO
017D 53FD ANL A,#FD (253)
017F A0 MOV @RO,A
0180 E400 JMP 700
***** REVERSE AAN
0182 FC MOV A,R4
0183 1287 JBO 87
0185 2476 JMP 176
0187 5276 JB2 76
0189 5378 ANL A,#78 (120)
018B 4305 ORL A,#05 (5)
018D AC MOV R4,A
018E B921 MOV R1,#21 (33)
0190 F1 MOV A,@R1
0191 CF DEC R7
0192 5298 JB2 98
0194 D5 SEL RB1
0195 18 INC RO
0196 C5 SEL RBO
0197 1F INC R7
0198 B104 MOV @R1,#04 (4)
019A 04B5 JMP OB5
***** PRINTSTART 27
019C BD01 MOV R5,#01 (1)
019E 1E INC R6
019F E400 JMP 700
***** STUURTEKEN 28

```

```

01A1 FC      MOV  A,R4
01A2 129F    JBO  9F
01A4 BD04    MOV  R5,#04 (4)
01A6 246D    JMP  16D
***** GRAFIEKMODUS AAN
01A8 B814    MOV  R0,#14 (20)
01AA 2302    MOV  A,#02 (2)
01AC 40      ORL  A,@R0
01AD 53FE    ANL  A,#FE (254)
01AF A0      MOV  @R0,A
01B0 E400    JMP  700
***** REVERSE UIT
01B2 FC      MOV  A,R4
01B3 52B7    JB2  B7
01B5 E400    JMP  700
01B7 5378    ANL  A,#78 (120)
01B9 4301    ORL  A,#01 (1)
01BB AC      MOV  R4,A
01BC 248E    JMP  18E
***** LINE FEED OF CR
01BE FC      MOV  A,R4
01BF 53FD    ANL  A,#FD (253)
01C1 AC      MOV  R4,A
01C2 BAOA    MOV  R2,#0A (10)
01C4 FB      MOV  A,R3
01C5 C6CB    JZ   CB
01C7 BB81    MOV  R3,#81 (129)
01C9 04B5    JMP  0B5
01CB BB80    MOV  R3,#80 (128)
01CD 04B5    JMP  0B5

01CF 00
01D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0200 99F3    ANL  P1,#F3 (243) CR + PRINTKOPMOTOR AAN
0202 99FB    ANL  P1,#FB (251) CR AAN
0204 B4CD    CALL 5CD      PRINTKOP IN BASISPOSITIE BRENGEN
0206 FC      MOV  A,R4
0207 538F    ANL  A,#8F (143)
0209 AC      MOV  R4,A
020A D46D    CALL 66D      PRINTKOPBEWEGING EN LICHTPULS TESTEN
020C F9      MOV  A,R1
020D 03E7    ADD  A,#E7 (231)
020F F618    JC   18
0211 B001    MOV  @R0,#01 (1)
0213 030C    ADD  A,#0C (12)
0215 F618    JC   18

```

```

0217 10    INC  @RO
0218 B810  MOV  RO,#10 (16)
021A 27    CLR  A
021B 62    MOV  T,A
021C 55    STRT T
021D 1621  JTF  21
021F 441D  JMP  21D
0221 E81D  DJNZ RO,1D
0223 65    STOP TCNT
0224 8908  ORL  P1,#08 (8)  PRINTKOPMOTOR UIT
0226 83    RET

```

```

0227 FD    MOV  A,R5
0228 1247  JBO  47
022A 3282  JB1  82
022C FE    MOV  A,R6
022D 07    DEC  A
022E 9632  JNZ  32
0230 246D  JMP  16D
0232 FA    MOV  A,R2
0233 F243  JB7  43
0235 B4F0  CALL 5F0
0237 B4F0  CALL 5F0
0239 D5    SEL  RB1
023A C8    DEC  RO
023B C5    SEL  RBO
023C B4C9  CALL 5C9
023E B917  MOV  R1,#17 (23)
0240 A1    MOV  @R1,A
0241 E400  JMP  700
0243 B4C9  CALL 5C9
0245 0482  JMP  082
0247 FE    MOV  A,R6
0248 07    DEC  A
0249 C672  JZ   72
024B 07    DEC  A
024C C679  JZ   79
024E B917  MOV  R1,#17 (23)
0250 F1    MOV  A,@R1
0251 A9    MOV  R1,A
0252 C65C  JZ   5C
0254 07    DEC  A
0255 9637  JNZ  37
0257 FA    MOV  A,R2
0258 0320  ADD  A,#20 (32)
025A F637  JC   37
025C B821  MOV  RO,#21 (33)
025E F0    MOV  A,@RO
025F 12A7  JBO  A7

```

0261	B081	MOV	@R0,#81	(129)
0263	3267	JB1	67	
0265	B001	MOV	@R0,#01	(1)
0267	FF	MOV	A,R7	
0268	29	XCH	A,R1	
0269	A1	MOV	@R1,A	
026A	1F	INC	R7	
026B	B4C9	CALL	5C9	
026D	B917	MOV	R1,#17	(23)
026F	A1	MOV	@R1,A	
0270	04B5	JMP	0B5	
0272	23F0	MOV	A,#F0	(240)
0274	6A	ADD	A,R2	
0275	963C	JNZ	3C	
0277	246D	JMP	16D	
0279	FA	MOV	A,R2	
027A	5301	ANL	A,#01	(1)
027C	B917	MOV	R1,#17	(23)
027E	A1	MOV	@R1,A	
027F	1E	INC	R6	
0280	E400	JMP	700	
0282	FE	MOV	A,R6	
0283	07	DEC	A	
0284	9689	JNZ	89	
0286	FA	MOV	A,R2	
0287	447C	JMP	27C	
0289	B917	MOV	R1,#17	(23)
028B	F1	MOV	A,@R1	
028C	530F	ANL	A,#0F	(15)
028E	E7	RL	A	
028F	A8	MOV	RO,A	
0290	B904	MOV	R1,#04	(4)
0292	68	ADD	A,RO	
0293	E992	DJNZ	R1,92	
0295	A8	MOV	RO,A	
0296	FA	MOV	A,R2	
0297	530F	ANL	A,#0F	(15)
0299	68	ADD	A,RO	
029A	A8	MOV	RO,A	
029B	BE05	MOV	R6,#05	(5)
029D	68	ADD	A,RO	
029E	E6A1	JNC	A1	
02A0	19	INC	R1	
02A1	EE9D	DJNZ	R6,9D	
02A3	AA	MOV	R2,A	
02A4	F9	MOV	A,R1	
02A5	4452	JMP	252	
02A7	B4F0	CALL	5F0	
02A9	CF	DEC	R7	

```

02AA CF DEC R7
02AB 4467 JMP 267
***** BUFFER PRINTEN
02AD 27 CLR A
02AE B815 MOV R0,#15 (21)
02B0 B003 MOV @R0,#03 (3)
02B2 85 CLR F0
02B3 AB MOV R3,A
02B4 BE18 MOV R6,#18 (24)
02B6 B81E MOV R0,#1E (30) KARAKTERTELLER TERUGZETTEN
02B8 A0 MOV @R0,A
02B9 18 INC R0
02BA A0 MOV @R0,A
02BB BD80 MOV R5,#80 (128)
02BD 6401 JMP 301 UITDRAAI STARTEN
*****
02BF FD MOV A,R5
02C0 F2C4 JB7 C4
02C2 8400 JMP 400 KARAKTERS PRINTEN
02C4 65 STOP TCNT
02C5 99FB ANL P1,#FB (251) CR AAN, REST UIT
02C7 FC MOV A,R4
02C8 538F ANL A,#8F (143)
02CA AC MOV R4,A
02CB 27 CLR A
02CC AD MOV R5,A
02CD A8 MOV R0,A
02CE 0A IN A,P2
02CF 92D7 JB4 D7 PRINTKOP IN BASISPOSITIE?
02D1 D410 CALL 610 JA -->
02D3 E8CE DJNZ R0,CE TIJDCONTROLE
02D5 0407 JMP 007 DEAD LOCK
02D7 D46D CALL 66D WACHTEN TOT PRINTKOP DE BASIS-
02D9 F0 MOV A,@R0 POSITIE HEEFT VERLATEN
02DA C6E0 JZ E0
02DC 32E9 JB1 E9
02DE 8400 JMP 400 KARAKTERS PRINTEN
02E0 F9 MOV A,R1
02E1 03F3 ADD A,#F3 (243)
02E3 F6DE JC DE
02E5 D4A1 CALL 6A1 WACHTEN OP PULS
02E7 8400 JMP 400 KARAKTERS PRINTEN
02E9 F9 MOV A,R1
02EA 03E7 ADD A,#E7 (231)
02EC F6DE JC DE
02EE 44E5 JMP 2E5
02F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

```

***** BUFFER PRINTEN
0300 AC      MOV  R4,A
0301 FF      MOV  A,R7
0302 A8      MOV  R0,A
0303 FO      MOV  A,@RO      BYTE HALEN
0304 1F      INC  R7          BUFFERWIJZER + 1
0305 AA      MOV  R2,A        BYTE --> R2
0306 03FO    ADD  A,#FO      (240) PRINTPOSITIE BEPALEN?
0308 961A    JNZ  1A          NEE -->
030A 85      CLR  FO
030B 95      CPL  FO
030C FF      MOV  A,R7
030D A8      MOV  R0,A
030E FO      MOV  A,@RO
030F 1F      INC  R7
0310 B917    MOV  R1,#17     (23)
0312 A1      MOV  @R1,A
0313 C9      DEC  R1
0314 18      INC  R0
0315 FO      MOV  A,@RO
0316 A1      MOV  @R1,A
0317 1F      INC  R7
0318 8472    JMP  472          STARTPOSITIE BEREKENEN
*****
031A 17      DEC  A          TEKSTMODUS AAN?
031B 9623    JNZ  23          NEE -->
031D FC      MOV  A,R4
031E 537C    ANL  A,#7C      (124)
0320 17      DEC  A
0321 6400    JMP  300
*****
0323 17      DEC  A          REVERSE AAN?
0324 962D    JNZ  2D          NEE -->
0326 FC      MOV  A,R4
0327 537C    ANL  A,#7C      (124)
0329 4381    ORL  A,#81      (129)
032B 6400    JMP  300
*****
032D 03FC    ADD  A,#FC      (252) =22?
032F 9636    JNZ  36          NEE -->
0331 FC      MOV  A,R4
0332 4304    ORL  A,#04      (4)
0334 6400    JMP  300
*****
0336 0380    ADD  A,#80      (128) =150?
0338 9643    JNZ  43          NEE
033A FC      MOV  A,R4
033B 123F    JBO  3F
033D 44BF    JMP  2BF

```

033F 53F9 ANL A,#F9 (249)

0341 6400 JMP 300

0343 038A ADD A,#8A (138) =12?

0345 964D JNZ 4D NEE -->

0347 85 CLR FO

0348 FC MOV A,R4

0349 537C ANL A,#7C (124) QUOTE FLAG RESET

034B 6400 JMP 300

034D 03EE ADD A,#EE (238) =30?

034F 9660 JNZ 60 NEE -->

0351 FF MOV A,R7

0352 A8 MOV RO,A

0353 FO MOV A,@RO

0354 AB MOV R3,A

0355 1F INC R7

0356 18 INC RO

0357 FO MOV A,@RO

0358 AA MOV R2,A

0359 1F INC R7

035A FC MOV A,R4

035B 4302 ORL A,#02 (2)

035D AC MOV R4,A

035E 44BF JMP 2BF

0360 0310 ADD A,#10 (16) WIJD GEDRUKT AAN?

0362 9666 JNZ 66 NEE -->

0364 84FE JMP 4FE

0366 FC MOV A,R4

0367 126B JBO 6B

0369 44BF JMP 2BF

***** KARAKTERS KLAARMAKEN VOOR PRINT
EN PER KOLOM VANAF
ADRES 24 WEGZETTEN

036B B81F MOV RO,#1F (31)

036D FO MOV A,@RO

036E C684 JZ 84

0370 C8 DEC RO

0371 FC MOV A,R4

0372 F27B JB7 7B

0374 FO MOV A,@RO

0375 0325 ADD A,#25 (37)

0377 F682 JC 82

0379 6484 JMP 384

037B FO MOV A,@RO

037C 032B ADD A,#2B (43)

037E F682 JC 82

0380	6484	JMP	384		
0382	849A	JMP	49A		LINE FEED
0384	FA	MOV	A,R2		
0385	0340	ADD	A,#40	(64)	TEKEN > 191?
0387	E696	JNC	96		NEE -->
0389	D461	CALL	661		POSITIE IN KAR.SET BEREKENEN
038B	F8	MOV	A,R0		
038C	F5	SEL	MB1		
038D	74D8	CALL	BD8		MATRIXKOLOM HALEN
038F	E5	SEL	MBO		
0390	18	INC	R0		
0391	19	INC	R1		
0392	EB8B	DJNZ	R3,8B		HERHALEN TOT 6 KOLOMMEN KLAAR ZIJN
0394	64D6	JMP	3D6		PRINTEN
0396	0320	ADD	A,#20	(32)	TEKEN > 159?
0398	E6A7	JNC	A7		NEE -->
039A	D461	CALL	661		POSITIE IN KAR.SET BEREKENEN
039C	F8	MOV	A,R0		
039D	F5	SEL	MB1		
039E	54C0	CALL	ACO		MATRIXKOLOM HALEN
03A0	E5	SEL	MBO		
0A31	18	INC	R0		
03A2	19	INC	R1		
03A3	EB9C	DJNZ	R3,9C		HERHALEN TOT 6 KOLOMMEN KLAAR ZIJN
03A5	64D6	JMP	3D6		PRINTEN
03A7	0340	ADD	A,#40	(64)	TEKEN > 95?
03A9	E6B8	JNC	B8		NEE -->
03AB	D461	CALL	661		POSITIE IN KAR.SET BEREKENEN
03AD	F8	MOV	A,R0		
03AE	F5	SEL	MB1		
03AF	B4C0	CALL	DC0		MATRIXKOLOM HALEN
03B1	E5	SEL	MBO		
03B2	18	INC	R0		
03B3	19	INC	R1		
03B4	EBAD	DJNZ	R3,AD		HERHALEN TOT 6 KOLOMMEN KLAAR ZIJN
03B6	64D6	JMP	3D6		PRINTEN
03B8	0320	ADD	A,#20	(32)	TEKEN > 63?
03BA	E6C9	JNC	C9		NEE -->
03BC	D461	CALL	661		POSITIE IN KAR.SET BEREKENEN
03BE	F8	MOV	A,R0		
03BF	F5	SEL	MB1		
03C0	34C0	CALL	9C0		MATRIXKOLOM HALEN
03C2	E5	SEL	MBO		
03C3	18	INC	R0		
03C4	19	INC	R1		
03C5	EBBE	DJNZ	R3,BE		HERHALEN TOT 6 KOLOMMEN KLAAR

03C7	64D6	JMP	3D6		ZIJN
03C9	0320	ADD	A,20	(32)	PRINTEN
03CB	D461	CALL	661		POSITIE IN KAR.SET BEREKENEN
03CD	F8	MOV	A,R0		
03CE	F5	SEL	MB1		
03CF	14C0	CALL	8C0		MATRIXKOLOM HALEN
03D1	E5	SEL	MBO		
03D2	18	INC	R0		
03D3	19	INC	R1		
03D4	EBCD	DJNZ	R3,CD		HERHALEN TOT 6 KOLOMMEN KLAAR ZIJN
03D6	FE	MOV	A,R6		
03D7	A9	MOV	R1,A		
03D8	F1	MOV	A,@R1		
03D9	AA	MOV	R2,A		
03DA	FC	MOV	A,R4		
03DB	52DF	JB2	DF		
03DD	44BF	JMP	2BF		PRINTEN
03DF	FA	MOV	A,R2		
03E0	37	CPL	A		
03E1	AA	MOV	R2,A		
03E2	44BF	JMP	2BF		PRINTEN

03E4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
03F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

***** KARAKTERS PRINTEN
0400 D4E8 CALL 6E8 REGELEINDE BEREIKT?
0402 F690 JC 90 JA -->
0404 FA MOV A,R2
0405 537F ANL A,#7F (127)
0407 A9 MOV R1,A
0408 B815 MOV R0,#15 (21)
040A 10 INC @R0
040B F0 MOV A,@R0
040C D304 XRL A,#04 (4)
040E C61B JZ 1B
0410 B8F0 MOV R0,#F0 (240)
0412 42 MOV A,T
0413 D302 XRL A,#02 (2)
0415 C61E JZ 1E
0417 E812 DJNZ R0,12
0419 0407 JMP 007 DEAD LOCK
041B A0 MOV @R0,A
041C 8420 JMP 420
041E 461E JNT1 1E PULSONDERBREKING AFWACHTEN
0420 65 STOP TCNT

0421	23C0	MOV	A,#C0	(192)	
0423	62	MOV	T,A		PULSBEWAKING
0424	55	STR	T		INSCHAKELEN
0425	25	EN	TCNTI		
0426	F9	MOV	A,R1		VOLGENDE PUNT VAN DE
0427	77	RR	A		MATRIXKOLOM HALEN
0428	A9	MOV	R1,A		
0429	37	CPL	A		
042A	562A	JT1	2A		PULS AFWACHTEN
042C	F23D	JB7	3D		PUNT PRINTEN? NEE -->
042E	99FE	ANL	P1,#FE	(254)	AANSLAG VAN DE PRINTKOP
0430	27	CLR	A		
0431	B842	MOV	RO,#42	(66)	
0433	F9	MOV	A,R1		
0434	D239	JB6	39		
0436	00	NOP			
0437	B84E	MOV	RO,#4E	(78)	
0439	E839	DJNZ	RO,39		
043B	8901	ORL	P1,#01	(1)	PRINTKOP UIT
043D	1D	INC	R5		PUNTTELLER + 1
043E	463E	JNT1	3E		PULSONDERBREKING AFWACHTEN
0440	65	STOP	TCNT		CONTROLE UIT
0441	35	DIS	TCNTI		
0442	23F9	MOV	A,#F9	(249)	7 PUNTEN GEPRINT?
0444	6D	ADD	A,R5		
0445	E621	JNC	21		NEE --> VOLGENDE PUNT
0447	C651	JZ	51		JA -->
0449	03F9	ADD	A,#F9	(249)	WIJD GEDRUKT KLAAR?
044B	9621	JNZ	21		NEE -->
044D	F4CD	CALL	7CD		KOLOMTELLER + 1 & PULSTELLER START
044F	8464	JMP	464		
0451	F4CD	CALL	7CD		----- " ----- " -----
0453	B672	JFO	72		
0455	FC	MOV	A,R4		
0456	1262	JBO	62		
0458	37	CPL	A		
0459	325D	JB1	5D		
045B	EB00	DJNZ	R3,00		
045D	4302	ORL	A,#02	(2)	
045F	37	CPL	A		
0460	6400	JMP	300		
0462	F26E	JB7	6E		
0464	1E	INC	R6		
0465	FE	MOV	A,R6		
0466	03E2	ADD	A,#E2	(226)	
0468	9670	JNZ	70		
046A	BE18	MOV	R6,#18	(24)	
046C	6401	JMP	301		VOLGENDE BYTE HALEN
046E	BD07	MOV	R5,#07	(7)	

0470 64D6 JMP 3D6

***** PRINTSTARTPOSITIE BEREKENEN

0472 B81F MOV RO,#1F (31)

0474 B917 MOV R1,#17 (23)

0476 F1 MOV A,@R1

0477 AA MOV R2,A

0478 FO MOV A,@RO

0479 37 CPL A

047A 6A ADD A,R2

047B F68C JC 8C

047D 37 CPL A

047E 9689 JNZ 89

0480 C8 DEC RO

0481 C9 DEC R1

0482 F1 MOV A,@R1

0483 AA MOV R2,A

0484 FO MOV A,@RO

0485 37 CPL A

0486 6A ADD A,R2

0487 F68C JC 8C

0489 85 CLR FO

048A 6401 JMP 301

048C 27 CLR A

048D AA MOV R2,A

048E 44BF JMP 2BF

0490 FC MOV A,R4

0491 37 CPL A

0492 329A JB1 9A

0494 CF DEC R7

0495 CF DEC R7

0496 FF MOV A,R7

0497 A8 MOV RO,A

0498 FB MOV A,R3

0499 A0 MOV @RO,A

049A CF DEC R7

049B 65 STOP TCNT

049C B4FC CALL 5FC

LINE FEED

049E B820 MOV RO,#20 (32)

04A0 FO MOV A,@RO

04A1 37 CPL A

04A2 F2AA JB7 AA

04A4 FC MOV A,R4

04A5 538F ANL A,#8F (143)

04A7 AC MOV R4,A

04A8 44AD JMP 2AD

BUFFER PRINTEN

04AA B822 MOV RO,#22 (34)

04AC FC MOV A,R4

04AD 37 CPL A

04AE	52B3	JB2	B3	
04B0	B012	MOV	@R0,#12	(18)
04B2	18	INC	R0	
04B3	B008	MOV	@R0,#08	(8)
04B5	12BD	JB0	BD	
04B7	B00F	MOV	@R0,#0F	(15)
04B9	F2BD	JB7	BD	
04BB	B00E	MOV	@R0,#0E	(14)
04BD	FF	MOV	A,R7	
04BE	A9	MOV	R1,A	
04BF	F1	MOV	A,@R1	
04C0	AA	MOV	R2,A	
04C1	18	INC	R0	
04C2	F9	MOV	A,R1	
04C3	0381	ADD	A,#81	(129)
04C5	F6CC	JC	CC	
04C7	19	INC	R1	
04C8	FA	MOV	A,R2	
04C9	A0	MOV	@R0,A	
04CA	84BF	JMP	4BF	
04CC	C8	DEC	R0	
04CD	F0	MOV	A,@R0	
04CE	A9	MOV	R1,A	
04CF	27	CLR	A	
04D0	A0	MOV	@R0,A	
04D1	F9	MOV	A,R1	
04D2	03F6	ADD	A,#F6	(246)
04D4	96CC	JNZ	CC	
04D6	F8	MOV	A,R0	
04D7	AF	MOV	R7,A	
04D8	27	CLR	A	
04D9	A0	MOV	@R0,A	
04DA	18	INC	R0	
04DB	F8	MOV	A,R0	
04DC	0380	ADD	A,#80	(128)
04DE	E6D8	JNC	D8	
04E0	B4C9	CALL	5C9	
04E2	B820	MOV	R0,#20	(32)
04E4	F0	MOV	A,@R0	
04E5	77	RR	A	
04E6	77	RR	A	
04E7	77	RR	A	
04E8	77	RR	A	
04E9	5307	ANL	A,#07	(7)
04EB	D5	SEL	RB1	
04EC	A8	MOV	R0,A	
04ED	C5	SEL	RBO	
04EE	D305	XRL	A,#05	(5)

```

04F0 96F4 JNZ F4
04F2 04E5 JMP OE5
04F4 F0 MOV A,@R0
04F5 4320 ORL A,#20 (32)
04F7 A44A JMP 54A
04F9 00 NOP
04FA 00 NOP
04FB 00 NOP
04FC 00 NOP
04FD 00 NOP

```

***** WIJD GEDRUKT

```

04FE B820 MOV R0,#20 (32)
0500 F0 MOV A,@R0
0501 F251 JB7 51
0503 8908 ORL P1,#08 (8)
0505 FD MOV A,R5
0506 D380 XRL A,#80 (128)
0508 C60F JZ OF
050A 99FB ANL P1,#FB (251)
050C 65 STOP TCNT
050D B4CD CALL 5CD
050F B821 MOV R0,#21 (33)
0511 B000 MOV @R0,#00 (0)
0513 18 INC R0
0514 B918 MOV R1,#18 (24)
0516 B100 MOV @R1,#00 (0)
0518 FC MOV A,R4
0519 37 CPL A
051A 521F JB2 1F
051C B012 MOV @R0,#12 (18)
051E 18 INC R0
051F B008 MOV @R0,#08 (8)
0521 1229 JB0 29
0523 B00F MOV @R0,#0F (15)
0525 F229 JB7 29
0527 B00E MOV @R0,#0E (14)
0529 18 INC R0
052A B010 MOV @R0,#10 (16)
052C 18 INC R0
052D B91F MOV R1,#1F (31)
052F F1 MOV A,@R1
0530 A0 MOV @R0,A
0531 C9 DEC R1
0532 18 INC R0
0533 F1 MOV A,@R1
0534 A0 MOV @R0,A
0535 18 INC R0
0536 F8 MOV A,R0
0537 AF MOV R7,A

```

0538	B4C9	CALL	5C9	
053A	F8	MOV	A,R0	
053B	37	CPL	A	
053C	0381	ADD	A,#81	(129)
053E	A9	MOV	R1,A	
053F	27	CLR	A	
0540	A0	MOV	@R0,A	
0541	18	INC	R0	
0542	E940	DJNZ	R1,40	
0544	B820	MOV	R0,#20	(32)
0546	F0	MOV	A,@R0	
0547	5302	ANL	A,#02	(2)
0549	4C	ORL	A,R4	
054A	532F	ANL	A,#2F	(47)
054C	AC	MOV	R4,A	
054D	BB01	MOV	R3,#01	(1)
054F	E400	JMP	700	
0551	B4FC	CALL	5FC	
0553	B87F	MOV	R0,#7F	(127)
0555	F0	MOV	A,@R0	
0556	C662	JZ	62	
0558	0A	IN	A,P2	
0559	37	CPL	A	
055A	B264	JB5	64	
055C	FC	MOV	A,R4	
055D	5371	ANL	A,#71	(113)
055F	4301	ORL	A,#01	(1)
0561	AC	MOV	R4,A	
0562	0444	JMP	044	
0564	B87F	MOV	R0,#7F	(127)
0566	F0	MOV	A,@R0	
0567	966F	JNZ	6F	
0569	B020	MOV	@R0,#20	(32)
056B	23F7	MOV	A,#F7	(247)
056D	5C	ANL	A,R4	
056E	AC	MOV	R4,A	
056F	B922	MOV	R1,#22	(34)
0571	B10F	MOV	@R1,#0F	(15)
0573	19	INC	R1	
0574	FC	MOV	A,R4	
0575	BC00	MOV	R4,#00	(0)
0577	37	CPL	A	
0578	727C	JB3	7C	
057A	BC01	MOV	R4,#01	(1)
057C	BB50	MOV	R3,#50	(80)
057E	F0	MOV	A,@R0	
057F	AA	MOV	R2,A	
0580	03BF	ADD	A,#BF	(191)
0582	F6AD	JC	AD	

```

0584 0341 ADD A,#41 (65)
0586 A1 MOV @R1,A
0587 10 INC @R0
0588 19 INC R1
0589 F0 MOV A,@R0
058A C6BF JZ BF
058C 0380 ADD A,#80 (128)
058E 9692 JNZ 92
0590 B0A0 MOV @R0,#A0 (160)
0592 EB7E DJNZ R3,7E
0594 B10A MOV @R1,#0A (10)
0596 BB81 MOV R3,#81 (129)
0598 BF22 MOV R7,#22 (34)
059A FC MOV A,R4
059B C6A1 JZ A1
059D BC09 MOV R4,#09 (9)
059F A4A3 JMP 5A3
05A1 BC01 MOV R4,#01 (1)
05A3 B820 MOV R0,#20 (32)
05A5 B081 MOV @R0,#81 (129)
05A7 B810 MOV R0,#10 (16)
05A9 B018 MOV @R0,#18 (24)
05AB 44AD JMP 2AD
05AD FC MOV A,R4
05AE C6B6 JZ B6
05B0 F5 SEL MB1
05B1 34D8 CALL 9D8
05B3 E5 SEL MB0
05B4 A4BA JMP 5BA
05B6 F5 SEL MB1
05B7 9400 CALL C00
05B9 E5 SEL MB0
05BA FA MOV A,R2
05BB B87F MOV R0,#7F (127)
05BD A486 JMP 586
05BF FC MOV A,R4
05C0 BC01 MOV R4,#01 (1)
05C2 C6C5 JZ C5
05C4 CC DEC R4
05C5 B020 MOV @R0,#20 (32)
05C7 A492 JMP 592
05C9 27 CLR A
05CA AD MOV R5,A
05CB AE MOV R6,A
05CC 83 RET

```

```

05CD FC MOV A,R4
05CE B2E6 JB5 E6
05D0 D425 CALL 625

```

TIMER INSTELLEN

```

05D2 0A    IN    A,P2
05D3 37    CPL    A                PRINTKOP IN BASISPOSITIE?
05D4 92E7  JB4    E7                NEE -->
05D6 D425  CALL   625
05D8 4320  ORL    A,#20    (32)
05DA AC    MOV    R4,A
05DB FC    MOV    A,R4
05DC 37    CPL    A
05DD B2E6  JB5     E6
05DF D430  CALL   630
05E1 0A    IN    A,P2
05E2 92DB  JB4     DB
05E4 A4D2  JMP     5D2
05E6 83    RET
05E7 D430  CALL   630                TIMER TESTEN
05E9 A4D2  JMP     5D2
***** LINE FEED
05EB F5    SEL    MB1
05EC C400  JMP     EO0
05EE E5    SEL    MB0
05EF 83    RET
*****
05F0 CF    DEC    R7                LAATSTE BYTE
05F1 FF    MOV    A,R7                VAN DE BUFFER
05F2 A8    MOV    RO,A                OP O ZETTEN
05F3 27    CLR    A
05F4 A0    MOV    @RO,A
05F5 83    RET
*****
05F6 D410  CALL   610
05F8 D410  CALL   610
05FA C410  JMP     610
***** LINE FEED
05FC 8908  ORL    P1,#08    (8)    PRINTKOPMOTOR UIT
05FE B820  MOV    RO,#20    (32)
0600 F0    MOV    A,@RO
0601 37    CPL    A
0602 F206  JB7     06
0604 120D  JBO     OD
0606 99FB  ANL    P1,#FB    (251) CR AAN
0608 B4EB  CALL   5EB                LINE FEED
060A B4CD  CALL   5CD                WACHTEN TOT PRINTKOP IN
060C 83    RET                        BASISPOSITIE IS
060D B4EB  CALL   5EB                LINE FEED
060F 83    RET
***** TIJDCONTROLE
0610 234F  MOV    A,#4F    (79)
0612 62    MOV    T,A
0613 55    STRT   T

```

```

0614 1618 JTF 18
0616 C414 JMP 614
0618 161C JTF 1C
061A C418 JMP 618
061C 1620 JTF 20
061E C41C JMP 61C
0620 FC MOV A,R4
0621 B233 JB5 33
0623 65 STOP TCNT
0624 83 RET
*****
0625 27 CLR A
0626 D5 SEL RB1
0627 A9 MOV R1,A
0628 C5 SEL RBO
0629 62 MOV T,A
062A 55 STRT T
062B FC MOV A,R4
062C 53AF ANL A,#AF (175)
062E AC MOV R4,A
062F 83 RET
*****
0630 1635 JTF 35
0632 83 RET
0633 D5 SEL RB1
0634 19 INC R1
0635 D5 SEL RB1
0636 19 INC R1
0637 F9 MOV A,R1
0638 C5 SEL RBO
0639 9648 JNZ 48
063B FC MOV A,R4
063C 37 CPL A
063D 9243 JB4 43
063F D25C JB6 5C
0641 0407 JMP 007
0643 37 CPL A
0644 4310 ORL A,#10 (16)
0646 AC MOV R4,A
0647 83 RET
0648 03F3 ADD A,#F3 (243)
064A FC MOV A,R4
064B 37 CPL A
064C B232 JB5 32
064E E632 JNC 32
0650 FC MOV A,R4
0651 538F ANL A,#8F (143)
0653 AC MOV R4,A
0654 8904 ORL P1,#04 (4) CR UIT

```

```

0656 27 CLR A
0657 D5 SEL RB1
0658 A9 MOV R1,A
0659 C5 SEL RBO
065A 65 STOP TCNT
065B 83 RET
065C 37 CPL A
065D 4340 ORL A,#40 (64)
065F AC MOV R4,A
0660 83 RET
*****
0661 AA MOV R2,A POSITIE
0662 97 CLR C IN
0663 F7 RLC A KARAKTER-
0664 AA MOV R2,A GENERATOR
0665 F7 RLC A BEREKENEN
0666 6A ADD A,R2
0667 A8 MOV R0,A
0668 B918 MOV R1,#18 (24)
066A BB06 MOV R3,#06 (6)
066C 83 RET
*****
066D 99F3 ANL P1,#F3 (243) PRINTKOPMOTOR EN CR AAN
066F 27 CLR A
0670 62 MOV T,A
0671 B80D MOV R0,#0D (13)
0673 55 STRT T
0674 1678 JTF 78
0676 C474 JMP 674
0678 E874 DJNZ R0,74
067A 65 STOP TCNT
067B 23C0 MOV A,#C0 (192)
067D 62 MOV T,A
067E 55 STRT T
067F 25 EN TCNTI DEAD LOCK, ALS ER GEEN PULS KOMT
0680 4684 JNT1 84 TWEE PULSEN
0682 5682 JT1 82 VAN DE
0684 4684 JNT1 84 KLOKSCHIJF
0686 5686 JT1 86 AFWACHTEN
0688 65 STOP TCNT
0689 35 DIS TCNTI
068A 8904 ORL P1,#04 (4) CR UIT
068C 27 CLR A
068D 62 MOV T,A
068E 55 STRT T
068F B920 MOV R1,#20 (32)
0691 0A IN A,P2 HEEFT DE PRINTKOP
0692 92BD JB4 BD DE BASISPOSITIE
0694 0A IN A,P2 VERLATEN?

```

```

0695 92BD JB4 BD NEEN -->
0697 B900 MOV R1,#00 (0)
0699 65 STOP TCNT
069A 169C JTF 9C
069C 23C0 MOV A,#C0 (192)
069E 62 MOV T,A
069F 55 STRT T
06A0 25 EN TCNTI
***** PULS AFWACHTEN
06A1 56A1 JT1 A1
06A3 46A3 JNT1 A3
06A5 65 STOP TCNT
06A6 35 DIS TCNTI
06A7 19 INC R1
06A8 27 CLR A
06A9 62 MOV T,A
06AA B81E MOV RO,#1E (30)
06AC A0 MOV @RO,A
06AD 18 INC RO
06AE A0 MOV @RO,A
06AF 55 STRT T
06B0 46A3 JNT1 A3
06B2 42 MOV A,T
06B3 37 CPL A
06B4 72B0 JB3 B0
06B6 12B0 JBO B0
06B8 65 STOP TCNT
06B9 B811 MOV RO,#11 (17)
06BB 97 CLR C
06BC 83 RET
*****
06BD 16C1 JTF C1 TIMER AFGELOPEN? JA, -->
06BF C491 JMP 691
06C1 E991 DJNZ R1,91 HERHALEN TOT R1=0, DAN
06C3 0407 JMP 007 DEAD LOCK
***** LINE FEED
06C5 8908 ORL P1,#08 (8) PRINTKOPMOTOR UIT
06C7 B810 MOV RO,#10 (16)
06C9 B018 MOV @RO,#18 (24)
06CB 0A IN A,P2
06CC 92D2 JB4 D2
06CE 99FB ANL P1,#FB (251) CR AAN
06D0 B4CD CALL 5CD
06D2 09 IN A,P1 LINE FEED--TOETS GEPRINT?
06D3 D2DD JB6 DD NEE -->
06D5 B4EB CALL 5EB
06D7 B4F6 CALL 5F6
06D9 B4F6 CALL 5F6
06DB C4D2 JMP 6D2

```

```

06DD E400 JMP 700 WACHTLUS
*****
06DF 997F ANL P1,#7F (127) DATA FLIPFLOP VRIJGEVEN
06E1 90 MOVX @RO,A DATA LOW MAKEN
06E2 83 RET
*****
06E3 8980 ORL P1,#80 (128) DATA HIGH MAKEN
06E5 997F ANL P1,#7F (127)
06E7 83 RET
***** REGELEINDE CONTROLEREN
06E8 97 CLR C
06E9 B81F MOV RO,#1F (31)
06EB F0 MOV A,@RO
06EC C6F6 JZ F6
06EE 03FE ADD A,#FE (254)
06F0 F6F6 JC F6
06F2 C8 DEC RO
06F3 F0 MOV A,@RO
06F4 0320 ADD A,#20 (32)
06F6 83 RET

06F7 00 00 00 00 00 00 00 00 00

***** WACHTLUS
0700 7613 JF1 13 ALS LISTENER GEADRESSEERD? JA -->
0702 0A IN A,P2 ATN?
0703 F21A JB7 1A JA -->
0705 D4E3 CALL 6E3 DATA HIGH MAKEN
0707 0A IN A,P2 ATN?
0708 F21A JB7 1A JA -->
070A B20E JB5 0E TESTMODUS?
070C A464 JMP 564 JA -->
070E 09 IN A,P1 LINE FEED-TOETS GEPRINT?
070F D200 JB6 00 NEE -->
0711 C4C5 JMP 6C5 LINE FEED UITWISSEN
0713 0A IN A,P2 ATN?
0714 F21A JB7 1A JA -->
0716 364C JTO 4C CLK HIGH? JA -->
0718 E400 JMP 700 WACHTEN
***** INSPRINGEN BIJ ATN
071A D4DF CALL 6DF DATA LOW MAKEN
071C B8DC MOV RO,#DC (220) TELLER ZETTEN
071E D4DF CALL 6DF DATA LOW MAKEN
0720 E81E DJNZ RO,1E WACHTEN
0722 09 IN A,P1 CLK LOW?
0723 37 CPL A
0724 9200 JB4 00 NEE -->

```

```

0726 B814 MOV RO,#14 (20)
0728 23CF MOV A,#CF (207)
072A 50 ANL A,@RO
072B A0 MOV @RO,A
072C B814 MOV RO,#14 (20) TELLER VOOR WACHTEN OP CLK ZETTEN
072E B908 MOV R1,#08 (8) BITTELLER ZETTEN
0730 8980 ORL P1,#80 (128) DATA VOOR INVOER VRIJGEVEN
0732 BA01 MOV R2,#01 (1)
0734 863C JNI 3C DATA HIGH? JA -->
0736 0A IN A,P2 ATN NOG LOW?
0737 37 CPL A
0738 F200 JB7 00 NEE, VALS ALARM -->
073A E434 JMP 734
073C 2643 JNTO 43 CLK NOG LOW? JA -->
073E 0A IN A,P2 ATN NOG LOW?
073F F248 JB7 48 JA -->
0741 E400 JMP 700 VALS ALARM -->
0743 0A IN A,P2 ATN NOG LOW?
0744 F27D JB7 7D JA -->
0746 E400 JMP 700 VALS ALARM -->
0748 E83C DJNZ RO,3C WACHTEN OP CLK=LOW
074A E469 JMP 769
***** OP BYTE WACHTEN
074C B814 MOV RO,#14 (20)
074E B908 MOV R1,#08 (8) BITTELLER ZETTEN
0750 8980 ORL P1,#80 (128) DATA HIGH MAKEN
0752 BA14 MOV R2,#14 (20)
0754 0A IN A,P2 ATN LOW?
0755 F21A JB7 1A JA -->
0757 865B JNI 5B DATA HIGH? JA -->
0759 E452 JMP 752 ANDERS WACHTEN
075B 2664 JNTO 64 CLK NOG LOW? JA -->
075D 0A IN A,P2 ATN LOW?
075E F21A JB7 1A JA -->
0760 E85B DJNZ RO,5B
0762 E469 JMP 769
0764 0A IN A,P2 ATN LOW?
0765 F21A JB7 1A JA -->
0767 E47D JMP 77D
0769 D4DF CALL 6DF DATA LOW MAKEN
076B B821 MOV RO,#21 (33)
076D E86D DJNZ RO,6D WACHTEN
076F 8980 ORL P1,#80 (128) DATA HIGH MAKEN
0771 0A IN A,P2
0772 EA78 DJNZ R2,78
0774 F27A JB7 7A
0776 E400 JMP 700
0778 F21A JB7 1A

```

077A	1A	INC	R2	
077B	3671	JTO	71	
*****				BYTE HALEN
077D	267D	JNTO	7D	WACHTEN OP CLK=HIGH
077F	97	CLR	C	BITS
0780	8683	JNI	83	NAAR
0782	A7	CPL	C	RO
0783	F8	MOV	A,R0	SCHUIVEN
0784	67	RRC	A	
0785	A8	MOV	RO,A	
0786	E971	DJNZ	R1,71	HERHALEN, TOT BITTELLER=0
0788	0A	IN	A,P2	ATN LOW?
0789	EA8F	DJNZ	R2,8F	
078B	F291	JB7	91	JA -->
078D	E400	JMP	700	WACHTLUS
078F	F21A	JB7	1A	ATN LOW? JA -->
0791	1A	INC	R2	
0792	3688	JTO	88	WACHTEN TOT CLK=LOW
0794	F8	MOV	A,R0	ONTVANGEN BYTE
0795	37	CPL	A	INVERTEREN EN
0796	AA	MOV	R2,A	NAAR R2 BRENGEN
0797	0A	IN	A,P2	ATN LOW?
0798	F2A7	JB7	A7	JA -->
079A	769E	JF1	9E	ALS LISTENER GEADRESSEERD? JA -->
079C	E400	JMP	700	WACHTLUS
079E	D4DF	CALL	6DF	DATA LOW MAKEN (BEVESTIGEN)
07A0	B814	MOV	RO,#14 (20)	
07A2	F0	MOV	A,@R0	
07A3	F200	JB7	00	
07A5	0475	JMP	075	
07A7	237F	MOV	A,#7F (127)	TIJDENS ATN
07A9	5A	ANL	A,R2	ONTVANGEN ADRESSEN
07AA	AA	MOV	R2,A	NAAR R2 BRENGEN
07AB	B814	MOV	RO,#14 (20)	
07AD	23C1	MOV	A,#C1 (193)	UNLISTEN?
07AF	6A	ADD	A,R2	
07B0	96B7	JNZ	B7	NEE -->
07B2	A5	CLR	F1	LISTEN FLAG TERUGZETTEN
07B3	B000	MOV	@R0,#00 (0)	
07B5	E4BA	JMP	7BA	
07B7	F5	SEL	MB1	
07B8	C478	JMP	E78	ADRESSEN CONTROLEREN
07BA	D4DF	CALL	6DF	DATA LOW MAKEN
07BC	36C3	JTO	C3	CLK HIGH? JA -->
07BE	0A	IN	A,P2	ATN LOW?
07BF	F2BC	JB7	BC	JA -->
07C1	E400	JMP	700	WACHTLUS
07C3	0A	IN	A,P2	ATN LOW?
07C4	F22C	JB7	2C	JA -->

```

07C6 E4C1 JMP 7C1
***** WACHTEN TOT ADRESSERINGS-
CYCLUS BEËINDIGD IS
07C8 0A IN A,P2 ATN LOW?
07C9 F2C8 JB7 C8 JA -->
07CB E400 JMP 700 WACHTLUS
*****
07CD 45 STRT CNT PULSTELLER
07CE 27 CLR A STARTEN
07CF 62 MOV T,A &
07D0 AD MOV R5,A KARAKTER-
07D1 B81E MOV R0,#1E (30) TELLER
07D3 10 INC @R0 + 1
07D4 F0 MOV A,@R0
07D5 18 INC R0
07D6 96D9 JNZ D9
07D8 10 INC @R0
07D9 83 RET
07DA 00 00 00 00 00 00
07E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
07F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

0800 00 00 00 00 00 00

0806 00 00 5F 00 00 00

*

*

*

*

*

*

080C 00 07 00 07 00 00

* *

* *

* *

0812 14 7F 14 7F 14 00

* *

* *

* *

* *

* *

0818 24 2A 7F 2A 12 00

*

* *

* *

*

081E 63 13 08 64 63 00

** *

** *

*

*

*

* **

* **

0824 30 4E 59 26 50 00

*
* *
* *
**
* * *
* *
** *

082A 00 04 02 01 00 00

*
*
*

0830 00 1C 22 41 00 00

*
*
*
*
*
*
*

0836 00 41 22 1C 00 00

*
*
*
*
*
*
*

083C 2A 1C 7F 1C 2A 00

*
* * *

* * *
*

0842 08 08 3E 08 08 00

*
*

*
*

0848 00 40 30 00 00 00

*
*
*

084E 08 08 08 08 08 00

0854 00 60 60 00 00 00

**
**

085A 60 10 08 04 03 00

*
*
*
*
*
*
*
*
*

0860 3E 51 49 45 3E 00

* *
* **
* * *
** *
* *

0866 00 42 7F 40 00 00

*
**
*
*
*
*

```

086C      62 51 51 49 46 00
          ***
          *   *
            *
           *
          **
          *
          *****

0872      22 41 49 49 36 00
          ***
          *   *
            *
           **
            *
          *   *
          ***

0878      18 14 12 7F 10 00
          *
          **
          * *
          * *
          *****
          *
          *

087E      27 45 45 45 39 00
          *****
          *
          ****
            *
            *
          *   *
          ***

0884      3C 4A 49 49 31 00
          ***
          *
          *
          *****
          *   *
          *   *
          ***

088A      01 71 09 05 03 00
          *****
          *
          *
          *
          *
          *
          *

```

0890 36 49 49 49 36 00

 * *
 * *

 * *
 * *

0896 46 49 49 29 1E 00

 * *
 * *

 *
 *

089C 00 00 12 00 00 00
 *
 *

08A2 00 40 32 00 00 00
 *
 *
 *
 *

08A8 08 14 22 41 41 00
 **
 *
 *
 *
 *
 *
 **

08AE 14 14 14 14 14 00

08B4 41 41 22 14 08 00

**

*

*

*

*

*

**

08BA 02 01 51 09 06 00

*

*

*

*

*

*

```
08C0 A3 .MOVP A,@A      MATRIKKOLOM
08C1 A1  MOV  @R1,A      HALEN
08C2 83  RET
```

```
08C3 FA  MOV  A,R2
08C4 F2D7 JB7  D7
08C6 03E0 ADD  A,#E0      (224) TEKEN>31?
08C8 F6D1 JC   D1         JA -->
08CA 0313 ADD  A,#13      (19)  CR?
08CC C6D4 JZ   D4         JA -->
08CE FC  MOV  A,R4
08CF 12D7 JBO  D7
08D1 E5  SEL  MBO
08D2 E400 JMP  700        WACHTLUS
08D4 E5  SEL  MBO
08D5 2407 JMP  107
08D7 34C3 CALL 9C3
08D9 FA  MOV  A,R2
08DA 0340 ADD  A,#40      (64) STUURTEKEN INTERPRETEREN
08DC AA  MOV  R2,A        REVERSE ON INVIOEGEN
08DD C470 JMP  E70        VAN STUURTEKEN
08DF FC  MOV  A,R4        PRINTBAAR TEKEN
08E0 72E6 JB3  E6        MAKEN
08E2 9400 CALL C000
08E4 04E8 JMP  8E8
08E6 34D8 CALL 9D8
08E8 FF  MOV  A,R7
08E9 A8  MOV  R0,A
08EA FA  MOV  A,R2
08EB A0  MOV  @R0,A
08EC 1F  INC  R7
08ED FC  MOV  A,R4
08EE 53FB ANL  A,#FB      (251)
08F0 AC  MOV  R4,A
08F1 B821 MOV  R0,#21     (33)
08F3 B004 MOV  @R0,#04    (4)
08F5 BB01 MOV  R3,#01     (1)
08F7 B818 MOV  R0,#18     (24)
08F9 B001 MOV  @R0,#01    (1)
08FB BA92 MOV  R2,#92     (146)
08FD E5  SEL  MBO
08FE 04B5 JMP  OB5
```

```

0900      3E 41 5D 55 5E 00
          ***
          *   *
          * ***
          * * *
          * ***
          *
          *****

0906      7E 09 09 09 7E 00
          ***
          *   *
          *   *
          *****
          *   *
          *   *
          *   *

090C      41 7F 49 49 36 00
          *****
          *   *
          *   *
          ***
          *   *
          *   *
          *****

0912      3E 41 41 41 22 00
          ***
          *   *
          *
          *
          *
          *   *
          ***

0918      41 7F 41 41 3E 00
          *****
          *   *
          *   *
          *   *
          *   *
          *   *
          *****

091E      7F 49 49 49 41 00
          *****
          *
          *
          *****
          *
          *
          *****

```

```

0924      7F 09 09 09 01 00
          *****
          *
          *
          ****
          *
          *
          *

092A      3E 41 41 49 3A 00
          ***
          *      *
          *
          *      **
          *      *
          *      *
          ***

0930      7F 08 08 08 7F 00
          *      *
          *      *
          *      *
          *****
          *      *
          *      *
          *      *

0936      00 41 7F 41 00 00
          ***
          *
          *
          *
          *
          *
          ***

093C      20 40 41 3F 01 00
          ***
          *
          *
          *
          *
          *      *
          **

0942      7F 08 14 22 41 00
          *      *
          *      *
          *      *
          **
          *      *
          *      *
          *      *

```

```

0948      7F 40 40 40 40 00
          *
          *
          *
          *
          *
          *
          *****

094E      7F 02 0C 02 7F 00
          *   *
          ** **
          * * *
          * * *
          *   *
          *   *
          *   *

0954      7F 04 08 10 7F 00
          *   *
          *   *
          **  *
          * * *
          *  **
          *   *
          *   *

095A      3E 41 41 41 3E 00
          ***
          *   *
          *   *
          *   *
          *   *
          *   *
          ***

0960      7F 09 09 09 06 00
          ****
          *   *
          *   *
          ****
          *
          *
          *

0966      3E 41 51 21 5E 00
          ***
          *   *
          *   *
          *   *
          * * *
          *   *
          **  *

```

```

096C      7F 09 19 29 46 00
          ****
          *   *
          *   *
          ****
          * *
          * *
          *   *
          *   *

0972      26 49 49 49 32 00
          ***
          *   *
          *
          ***
          *
          *   *
          ***

0978      01 01 7F 01 01 00
          *****
          *
          *
          *
          *
          *
          *

097E      3F 40 40 40 3F 00
          *   *
          *   *
          *   *
          *   *
          *   *
          *   *
          ***

0984      07 18 60 18 07 00
          *   *
          *   *
          *   *
          * *
          * *
          * *
          *
          *

098A      7F 20 18 20 7F 00
          *   *
          *   *
          *   *
          * * *
          * * *
          ** **
          *   *

```

0990	63 14 08 14 63 00
	* * *
	* * *
	* *
	*
	* *
	* *
	* *
	*
0996	07 08 78 08 07 00
	* *
	* *
	* *

	*
	*
	*
099C	61 51 49 45 43 00

	*
	*
	*
	*

09A2	00 7F 41 41 00 00

	*
	*
	*
	*
	*

09A8	48 7E 49 49 42 00
	**
	* *
	*

	*
	*

09AE	00 41 41 7F 00 00

	*
	*
	*
	*

09B4 00 04 02 7F 02 04

*

* * *
*
*
*
*

09BA 08 1C 2A 08 08 08

*
*
*

*
*

09C0 A3 MOV P A,@A MATRIXKOLOM
09C1 A1 MOV @R1,A HALEN
09C2 83 RET

09C3 FC MOV A,R4 TEKEN
09C4 52D2 JB2 D2
09C6 B821 MOV R0,#21 (33) VOOR
09C8 F0 MOV A,@R0
09C9 CF DEC R7 REVERSE
09CA 52D2 JB2 D2
09CC 1F INC R7 IN
09CD FF MOV A,R7
09CE A8 MOV R0,A BUFFER
09CF B012 MOV @R0,#12 (18)
09D1 1F INC R7 INVVOEGEN
09D2 83 RET

09D3 FA MOV A,R2 VAN STUURCODE
09D4 0340 ADD A,#40 (64) PRINTBAAR
09D6 AA MOV R2,A
09D7 83 RET MAKEN

***** TEKEN CONVERTEREN

09D8 B814 MOV R0,#14 (20)
09DA F0 MOV A,@R0
09DB 37 CPL A
09DC 32E0 JB1 E0
09DE 8409 JMP 409
09E0 9409 CALL 409
09E2 FA MOV A,R2 TEKEN
09E3 0325 ADD A,#25 (37) >219?
09E5 F6FE JC FE JA -->
09E7 031B ADD A,#1B (27) =192?
09E9 C6FE JZ FE JA -->
09EB E6F1 JNC F1 <192? JA -->
09ED 0340 ADD A,#40 (64)
09EF 24FD JMP 1FD
09F1 0320 ADD A,#20 (32) >160?
09F3 F6FE JC FE JA -->
09F5 FA MOV A,R2
09F6 03A5 ADD A,#A5 (165) >91?
09F8 F6FE JC FE JA -->
09FA FA MOV A,R2
09FB 0380 ADD A,#80 (128)
09FD AA MOV R2,A
09FE C44B JMP 64B

0A00 00 00 00 00 00 00

0A06 7F 7F 7F 00 00 00

0A0C 78 78 78 78 78 78

0A12 01 01 01 01 01 01

0A18 40 40 40 40 40 40

0A1E 7F 00 00 00 00 00
*
*
*
*
*
*
*

0A24 55 2A 55 2A 55 2A
 * * *
 * * *
 * * *
 * * *
 * * *
 * * *
 * * *

0A2A 00 00 00 00 00 7F
 *
 *
 *
 *
 *
 *
 *

0A30 50 28 50 28 50 28
 * * *
 * * *
 * * *
 * * *

0A36 3F 1F 0F 07 03 01

 **
 *

0A3C 00 00 00 00 00 7F
 *
 *
 *
 *
 *
 *
 *

0A42 00 00 00 7F 08 08
 *
 *
 *

 *
 *
 *

0A48 00 00 00 78 78 78

0A4E 00 00 00 0F 08 08

*
*
*

0A54 08 08 08 78 00 00

*
*
*

0A5A 60 60 60 60 60 60

0A60 00 00 00 78 08 08

*
*
*

0A66 08 08 08 0F 08 08

*
*
*

0A6C 08 08 08 78 08 08

*

*

*

0A72 08 08 08 7F 00 00

*

*

*

*

*

*

0A78 7F 00 00 00 00 00

*

*

*

*

*

*

*

0A7E 7F 7F 00 00 00 00

**

**

**

**

**

**

**

0A84 00 00 00 00 7F 7F

**

**

**

**

**

**

**

0A8A 01 01 01 01 01 01

0A90 03 03 03 03 03 03

0A96 70 70 70 70 70 70

0A9C 40 40 40 40 40 7F
*
*
*
*
*
*

0AA2 78 78 78 00 00 00

0AA8 00 00 00 07 07 07

0AAE 08 08 08 0F 00 00
*
*
*

0AB4 07 07 07 00 00 00

0ABA 07 07 07 78 78 78

OAC0 A3 MOVP A,@A MATRIKKOLOM

OAC1 A1 MOV @R1,A HALEN

OAC2 83 RET

OAC3 00 00 00 00 00 00 00 00 00 00 00 00 00 00

OADO 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

OAE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

OAF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

0B00 08 08 08 08 08 08

0B06 1C 4E 7F 4E 1C 00

*

* * *

*

0B0C 00 00 7F 00 00 00

*

*

*

*

*

*

*

0B12 04 04 04 04 04 04

0B18 02 02 02 02 02 02

0B1E 01 01 01 01 01 01

0B24 20 20 20 20 20 20

0B2A 00 7F 00 00 00 00

*
*
*
*
*
*
*

0B30 00 00 00 00 7F 00

*
*
*
*
*
*
*

0B36 08 08 10 60 00 00

**

*
*
*

0B3C 00 00 00 03 04 08

*
*
*
*

0B42 08 08 04 03 00 00

*
*
*
**

```

0B48      7F 40 40 40 40 40
          *
          *
          *
          *
          *
          *
          *****

0B4E      02 04 08 10 20 40
          *
          *
          *
          *
          *
          *

0B54      40 20 10 08 04 02
          *
          *
          *
          *
          *
          *
          *

0B5A      7F 01 01 01 01 01
          *****
          *
          *
          *
          *
          *
          *

0B60      01 01 01 01 01 7F
          *****
          *
          *
          *
          *
          *
          *

0B66      3C 7E 7E 7E 3C 00
          ***
          *****
          *****
          *****
          *****
          *****
          ***

```

0B6C 20 20 20 20 20 20

0B72 1C 3E 7C 3E 1C 00

* *

*
*

0B78 00 7F 00 00 00 00

*
*
*
*
*
*
*

0B7E 00 00 00 60 10 08

*
*
*
*

0B84 42 24 18 18 24 42

* *
* *
**
**
* *
* *

0B8A 3C 42 42 42 3C 00

* *
* *
* *
* *

0B90 1C 0A 7F 0A 1C 00

*

* * *

* * *
*
*

0B96 00 00 00 00 7F 00

*
*
*
*
*
*
*

0B9C 18 3C 7E 3C 18 00

*

*

0BA2 08 08 08 7F 08 08

*
*
*

*
*
*

0BA8 55 2A 55 00 00 00

* *
*
* *
*
* *
*
* *

0BAE 00 00 00 7F 00 00

*
*
*
*
*
*
*

0BB4 08 7C 04 7C 04 00

** *
* *
* *
* *

0BBA 01 03 07 0F 1F 3F

**
*

0BC0 00 00 00 7F 7F 7F

0BC6 10 10 10 10 10 10

0BCC 78 20 10 08 04 00

*
* *
* *
**
*

0BD2 12 49 24 12 49 24

* *
* *
* *
* *
* *
* *
* *

```
OBD8  A3      MOVP  A,@A          MATRIKKOLOM
OBD9  A1      MOV   @R1,A        HALEN
OBDA  83      RET
```

```
OBDB  00 00 00 00 00
OBEO  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
OBFO  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
OC00  B814  MOV   R0,#14  (20)
OC02  F0     MOV   A,@R0
OC03  1207  JBO   07
OC05  8409  JMP   C09
OC07  24E0  JMP   9E0
```

***** TEKEN CONVERTEREN

```
OC09  FA     MOV   A,R2      TEKEN
OC0A  0360  ADD   A,#60      (96)  <160?
OC0C  E620  JNC   20         JA -->
OC0E  03C0  ADD   A,#C0      (192) <224?
OC10  E634  JNC   34         JA -->
OC12  03E1  ADD   A,#E1      (225) =255?
OC14  C61D  JZ    1D         JA -->
OC16  0309  ADD   A,#09      (9)   =246?
OC18  9630  JNZ   30         NEE -->
OC1A  BAE0  MOV   R2,#E0     (224)
OC1C  83     RET
OC1D  BADE  MOV   R2,#DE     (222)
OC1F  83     RET
OC20  0340  ADD   A,#40      (64)  <96?
OC22  E634  JNC   34         JA -->
OC24  03FA  ADD   A,#FA      (250) =102?
OC26  962B  JNZ   2B         NEE -->
OC28  BAE1  MOV   R2,#E1     (225)
OC2A  83     RET
OC2B  FA     MOV   A,R2
OC2C  0360  ADD   A,#60      (96)
OC2E  8433  JMP   C33
OC30  FA     MOV   A,R2
OC31  03C0  ADD   A,#C0      (192)
OC33  AA     MOV   R2,A
OC34  83     RET
```

```
OC35  00 00 00 00 00 00 00 00 00 00 00
OC40  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
OC50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
OC60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
OC70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

OC80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
OC90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
OCA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
OCB0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
OCC0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
OCD0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
OCE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
OCF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

0D00 08 08 08 08 08 08

0D06 20 54 54 3C 40 00

*

* *
** *

0D0C 7F 48 44 44 38 00

*
*
* **
** *
* *
* *

0D12 38 44 44 44 44 00

*
*
*

0D18 38 44 44 48 7F 00

*
*
** *
* **
* *
* *

0D1E 38 54 54 54 08 00

* *

*

```

0D24      08 7E 09 01 02 00
          **
          *  *
          *
          ***
          *
          *
          *

```

```

0D2A      0C 52 52 52 3C 00
          ***
          *  *
          *  *
          ****
          *
          ***

```

```

0D30      7F 08 04 04 78 00
          *
          *
          *  **
          **  *
          *  *
          *  *
          *  *

```

```

0D36      00 44 7D 40 00 00
          *
          **
          *
          *
          *
          ***

```

```

0D3C      00 20 40 3D 00 00
          *
          *
          *
          *  *
          *

```

```

0D42      7F 10 28 44 00 00
          *
          *
          *  *
          *  *
          **
          *  *
          *  *

```

0D48 00 01 7F 40 00 00
**
*
*
*
*
*
**

0D4E 7C 04 38 04 78 00

** *
* * *
* * *
* * *
* *

0D54 04 78 04 04 78 00

* **
* *
* *
* *
* *

0D5A 38 44 44 44 38 00

* *
* *
* *

0D60 7E 12 12 0C 00 00

* *
* *

*
*

0D66 0C 12 12 7E 00 00

* *
* *

*
*

0D6C 7C 08 04 04 08 00

* **
** *
*
*
*

0D72 48 54 54 54 20 00

*

*

0D78 04 3F 44 44 20 00

*
*

*
*
* *
**

0D7E 3C 40 40 3C 40 00

* *
* *
* *
* *
** *

0D84 0C 30 40 30 0C 00

* *
* *
* *
* *
*

0D8A 3C 40 30 40 3C 00

* *
* *
* * *
* * *
* *

0D90 44 28 10 28 44 00

* *
* *
*
* *
* *

0D96 4C 50 50 3C 00 00

* *
* *

*

0D9C 44 64 54 4C 44 00

*
*
*

0DA2 08 08 08 7F 08 08

*
*
*

*
*
*

0DA8 55 2A 55 00 00 00

* *
*
* *
*
* *
*
* *

0DAE 00 00 00 7F 00 00

*
*
*
*
*
*
*

```

ODB4      73 73 0C 0C 73 73
          **  **
          **  **
           **
           **
          **  **
          **  **
          **  **

```

```

ODBA      24 49 12 24 49 12
          *  *
           *  *
          *  *
          *  *
           *  *
          *  *
          *  *

```

```
ODCO  A3    MOVP  A,@A          MATRIXKOLOM
ODC1  A1     MOV   @R1,A        HALEN
ODC2  83     RET
```

```
ODC3  00 00 00 00 00 00 00 00 00 00 00 00 00 00
ODD0  00 00 00 00 00 00 00 00 00 00 00 00 00 00
ODE0  00 00 00 00 00 00 00 00 00 00 00 00 00 00
ODF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

***** LINE FEED, STAPGROOTTE
IN GEHEUGEN 16

```
OE00  B910  MOV   R1,#10  (16)
OE02  F1     MOV   A,@R1
OE03  A9     MOV   R1,A
OE04  99FD  ANL   P1,#FD  (253) LINE FEED MOTOR AAN
OE06  D437  CALL  E37      DELAY
OE08  D424  CALL  E24      MOTOR 1 STAP VERDER
OE0A  E90E  DJNZ  R1,OE
OE0C  C41D  JMP   E1D
OE0E  D43B  CALL  E3B      DELAY
OE10  C419  JMP   E19
OE12  D43F  CALL  E3F      DELAY
OE14  E918  DJNZ  R1,18
OE16  D43F  CALL  E3F      DELAY
OE18  19     INC   R1
OE19  D424  CALL  E24      MOTOR 1 STAP VERDER
OE1B  E912  DJNZ  R1,12
OE1D  D437  CALL  E37      DELAY
OE1F  8902  ORL   P1,02   (2)  LINE FEED MOTOR UIT
OE21  E5     SEL   MB0
OE22  A4EE  JMP   5EE
```

***** LINE FEED MOTOR 1 STAP VERDER

```
OE24  0A     IN    A,P2
OE25  122C  JBO   2C
OE27  3234  JB1   34
OE29  8A02  ORL   P2,#02  (2)
OE2B  83     RET
OE2C  3231  JB1   31
OE2E  9AFE  ANL   P2,#FE  (254)
OE30  83     RET
OE31  9AFD  ANL   P2,#FD  (253)
OE33  83     RET
OE34  8A01  ORL   P2,#01  (1)
OE36  83     RET
```

***** DELAY

```
OE37  2330  MOV   A,#30   (48)
OE39  C441  JMP   E41
```

```

OE3B 2351 MOV A,#51 (81)
OE43 C441 JMP E41
OE3F 32A7 MOV A,#A7 (167)
OE41 62 MOV T,A
OE42 55 STRT T
OE43 1645 JTF 45
OE45 1649 JTF 49
OE47 C445 JMP E45
OE49 65 STOP TCNT
OE4A 83 RET
***** KARAKTERS CONVERTEREN
OE4B FA MOV A,R2
OE4C 031F ADD A,#1F (31) KARAKTER
OE4E 9653 JNZ 53 =225?
OE50 BA46 MOV R2,#46 (70) NEE -->
OE52 83 RET
OE53 0301 ADD A,#01 (1) =224?
OE55 9658 JNZ 58 NEE -->
OE57 83 RET
OE58 0320 ADD A,#20 (32) =192?
OE5A C657 JZ 57 JA -->
OE5C E662 JNC 62 <192? JA -->
OE5E 0360 ADD A,#60 (96)
OE60 AA MOV R2,A
OE61 83 RET
OE62 0306 ADD A,#06 (6) =186?
OE64 9669 JNZ 69 NEE -->
OE66 BAE2 MOV R2,#E2 (226)
OE68 83 RET
OE69 0311 ADD A,#11 (17) =169?
OE6B 9657 JNZ 57 NEE -->
OE6D BAE3 MOV R2,#E3 (227)
OE6F 83 RET
*****
OE70 03C0 ADD A,#C0 (192)
OE72 C676 JZ 76
OE74 04DF JNP 8DF
OE76 04E8 JMP 8E8
***** ADRES CONTROLLEREN
OE78 FO MOV A,@RO AL GEADRESSEERD?
OE79 9295 JB4 95 JA --> SECUNDAIR ADRES
OE7B 0A IN A,P2 HEEFT DE PRINTER ADRES 4?
OE7C D282 JB6 82 JA -->
OE7E 23DB MOV A,#DB (219)
OE80 C484 JMP E84
OE82 23DC MOV A,#DC (220)
OE84 6A ADD A,R2 HET JUISTE LISTENER-ADRES?
OE85 C68A JZ 8A JA -->
OE87 E5 SEL MBO ANDERS

```

```

OE88 E4C8 JMP 7C8          TERUG
***** PRINTER IS GEADRESSEERD
OE8A 2310 MOV A,#10      (16)
OE8C 40   ORL A,@R0      LISTENER FLAGS
OE8D A0   MOV @R0,A
OE8E A5   CLR F1          INSTELLEN
OE8F B5   CPL F1
OE90 C4AE JMP EAE
*****
OE92 E5   SEL MBO
OE93 E4BA JMP 7BA
***** SECUNDAIR ADRES CONTROLEREN
OE95 B292 JB5 92          AL GEBEURD? JA -->
OE97 2320 MOV A,#20      (32)
OE99 40   ORL A,@R0
OE9A A0   MOV @R0,A
OE9B FA   MOV A,R2
OE9C 0396 ADD A,#96      (150)
OE9E C6BE JZ BE
OEAO 0302 ADD A,#02      (2)
OEAA C6AE JZ AE
OEAA 17   DEC A
OEAA C6B4 JZ B4
OEAA 17   DEC A
OEAA C6BE JZ BE
OEAA 0306 ADD A,#06      (6)
OEAA 96BE JNZ BE
OEAA 23F7 MOV A,#F7      (247)
OEBO 5C   ANL A,R4
OEB1 AC   MOV R4,A
OEB2 C4B8 JMP EB8
OEB4 2308 MOV A,#08      (8)
OEB6 4C   ORL A,R4
OEB7 AC   MOV R4,A
OEB8 237C MOV A,#7C      (124)
OEBA 50   ANL A,@R0
OEBB A0   MOV @R0,A
OEBB C4C2 JMP EC2
OEBE 2380 MOV A,#80      (128)
OECO 40   ORL A,@R0
OEC1 A0   MOV @R0,A
OEC2 E5   SEL MBO
OEC3 E4BA JMP 7BA
***** E I N D E *****

```

Vanaf hier tot aan het einde van het geheugen (\$0FFF) staat alleen nog \$00.

8 DE PLOTTER COMMODORE 1520

8.1 Werkwijze en gebruiksmogelijkheden van de plotter

U moet eerst iets meer van de algemene werkwijze van de plotter afweten om het apparaat zo optimaal mogelijk te kunnen benutten en zijn functies te begrijpen. Daarom een overzicht van de belangrijkste technische gegevens:

- * 4 pennen (zwart, blauw, groen, rood)
- * 4 verschillende tekengroottes
- * 10, 20, 40 of 80 karakters per plotterregel
- * 13 letters per seconde bij de kleinste lettersoort
- * 96 verschillende karakters
- * 50 tekenstappen per centimeter
- * 480 tekenstapen in de x-richting
- * circa 999 tekenstappen in de y-richting
- * 264 tekenstappen per seconde
- * 5 centimeter tekenafstand per seconde (x- of y-richting)
- * 7 centimeter tekenafstand per seconde (bij een hoek van 45 graden)
- * 11.4 centimeter breed papier (4.5 inch)

Deze gegevens moeten voorlopig voldoende zijn om u inzicht in de mogelijkheden van de plotter te geven. De resolutie, het oplossend vermogen, valt gelijk op: binnen een vierkant van 480 x 480 stappen kunnen 230.400 punten onafhankelijk van elkaar aangestuurd worden. Ook de tekensnelheid van 5 centimeter per seconde laat niets te wensen over. Sommigen zal het wat vreemd voorkomen dat in x- of y-richting stappen van 5 centimeter per seconde gelden en bij een hoek van 45 graden 7 centimeter per seconde. De verklaring hiervoor luidt als volgt: we tekenen eerst gedurende 1 seconde in de x-richting en daarna 1 seconde lang in de y-richting. Resultaat: twee lijnen van 5 centimeter die loodrecht op elkaar staan onder een hoek van 90 graden. Hier geldt de stelling van Pythagoras. We hebben namelijk een rechte hoek en twee zijden en moeten alleen nog de derde zijde berekenen. We gebruiken de volgende vergelijking:

$$x^2 + y^2 = z^2$$

of in getallen:

$$5^2 + 5^2 = z^2$$

De onbekende zijde z is dan de wortel uit 50, dus ongeveer 7 centimeter. Door de combinatie van beide tekenrichtingen blijft de tekentijd van een seconde gehandhaafd, maar wordt toch een

groter aantal centimeters afgelegd. Deze ietwat uitvoerige analyse is nodig om de programmering vanaf het begin te begrijpen. U wilt tenslotte zelf op creatieve wijze met het apparaat om kunnen gaan en niet alleen maar bestaande routines overtypen. We gaan weer een stapje verder en halen de beschermkap van de plotter af om de mechanische functie-onderdelen nauwkeuriger te bekijken. Allereerst zal de trommel met de ballpoints u opvallen. Ook de andere onderdelen bekijken we even, omdat nauwkeurigheid en bedrijfszekerheid afhangen van hun precisie. Links en rechts van de rubberrol ziet u een gecompliceerde set tandwielen. Dit geheel zorgt voor het transport in het algemeen, want de trommel kan alleen maar in de x-richting bewegen, dus moet het papier voor de y-richting zorgen. Op de rubberrol bevindt zich een tractorfeed mechanisme (waarvan u de fijne punten haast niet kunt zien), dat zeer nauwkeurig werkt. En dan is er nog het groene blokje in de rechterhoek van het aandrijfmechanisme. Dit is een elektromagneet. Is de magneet aan dan drukt de veer aan de linkerkant van de metalen strip die overdwars in het apparaat zit de strip tegen een pen zodat die het papier raakt. Is de magneet uit dan verhindert de veer dat de strip met een pen in contact komt. We laten het technische gedeelte verder voor wat het is, maar geven nog een advies: ga er niet te ruw mee om. De plotter is weliswaar robuust gebouwd en kan zware belasting verdragen, maar de fijne kabeltjes en tandwielletjes kunnen niet veel verduren. Houdt u daarom rekening met de volgende zaken bij het hanteren van uw plotter:

- De rubberrol slechts in noodgevallen met de hand draaien, bijvoorbeeld om papier te vervangen.
- De trommel met de pennetjes nooit met de hand draaien.
- Mechanische onderdelen zoals het metalen stripje van de elektromagneet of de tandwielen nooit onnodig met de hand bewegen. Anders komt de nauwkeurigheid in gevaar.
- Niet printen zonder papier.
- De stroomtoevoer mag tijdens het printen niet onderbroken worden.

Aan het einde van deze technische inleiding nog iets over het onderhoud en de reiniging van de plotter:

- Behoed het apparaat voor teveel stof, want dat gaat vastzitten aan de gesmeerde onderdelen van het aandrijfmechanisme. Dit leidt tot storingen.
- Als de rol per ongeluk met de ballpoints wordt beschreven, moet u de beschermkap openen en de rol met een vochtige doek afvegen.
- Mocht het eens voorkomen dat het papier vastzit, verwijder dan de kap en verhelp het euvel voorzichtig met de hand. Druk vooral niet op de paper-feed toets!

De plotter kan haast alle denkbare karweitjes opknappen, van mathematische figuren tot statistieken. Maar het is aan u om die mogelijkheden er uit te halen. Bij alle volgende beschrijvingen proberen we aan te geven welke mogelijkheden er allemaal zijn en hoe ze benut kunnen worden. De voorbeelden dienen voornamelijk als inspiratiebron. De plotter doet natuurlijk alleen het werk dat u hem opdraagt en daarvoor zijn programma's nodig. Om deze routines te kunnen schrijven, krijgt u nu een uitvoerige uitleg van de instructieset. In het computergeheugen van de Commodore 64 hoeft geen plaats te worden gereserveerd om de plotter te sturen. Het apparaat heeft zelf een geheugen en een bedrijfssysteem met een eigen instructiesyntax. De communicatie verloopt net als bij andere apparaten (zoals de diskdrive of de datasette) via de instructies OPEN, PRINT# of CLOSE. Deze instructies worden met adressen verbonden en via secundaire adressen nog verder gespecificeerd.

OPEN

Deze instructie maakt samen met een serie adressen (waarop we dadelijk nader ingaan) de plotter ontvankelijk voor bepaalde functies. Deze corresponderen met de adressen. Pas na de OPEN-instructie kan het apparaat instructies verwerken. Dat geldt in de directe modus en ook als de aansturing via een programma gebeurt. De adressen bestaan doorgaans uit drie getallen, gescheiden door een komma:

OPEN NR, DN, IN

De afkortingen hebben de volgende betekenis: NR is het nummer waaronder later PRINT#- of CLOSE-instructies worden aangesproken. U kunt nummers kiezen van 1-255 maar dan moet u goed in de gaten houden wat er zojuist gezegd is, omdat anders de foutmelding FILE NOT OPEN op het scherm verschijnt. In de loop van het verhaal komen we nog op deze instructie terug en zullen we het gebruik aan de hand van voorbeelden toelichten. Het DN

(device-nummer) levert ook geen problemen op: het is door de constructie een vast gegeven en kan niet zonder meer worden veranderd. Het device-nummer van de plotter 1520 is 6. Deze 6 zal bij alle voorbeelden opduiken als tweede nummer en behoeft geen verdere toelichting. Interessanter wordt het bij IN, de instructie of het secundair adres. Die variabele maakt uw bedoelingen kenbaar aan de processor in de 1520. Dit getal is bepalend voor de geplande uitvoering. Hiermee kiest u de functies van het apparaat en voert u ze later uit. U kunt de volgende instructies of secundaire adressen gebruiken:

- in=0 normaal printwerk voor schrijven of listen
- in=1 tekenroutines uitvoeren
- in=2 keuze van pen/kleur
- in=3 keuze van tekengrootte
- in=4 tekens 90 graden draaien
- in=5 keuze van lijnsoort
- in=6 keuze van hoofdletter-/kleine letter-/grafische modus
- in=7 plotter terug in basispositie, plottergeheugen wissen

OPEN 34,6,2 opent filennummer 34 met betrekking tot de plotter 1520 (6). Het gaat om de keuze van een pen.

OPEN 1,6,1 opent filennummer 1 met betrekking tot de plotter 1520 (6). Een tekenroutine moet worden uitgevoerd.

OPEN 99,9,9 is niet zinvol. Het device-nummer is niet 6 en secundair adres 9 bestaat niet. Op het scherm verschijnt de foutmelding DEVICE NOT PRESENT.

We gaan hieronder nader in op alle secundaire adressen omdat de handleiding van de plotter sommige instructies erg summier behandelt. Beginners kunnen dan verder alleen maar gissen.

PRINT#

Deze instructie zorgt ervoor dat de gegevens op papier zichtbaar worden gemaakt en kiest de gewenste ballpoint of zet de plotter terug. De instructie luidt:

PRINT# NR, ...

Denk aan wat we hiervoor hebben uitgelegd: NR is het nummer waarmee u de file hebt geopend. Dit nummer kan tussen 1 en 255 liggen. Als u met de PRINT#-instructie naar een geopende file wilt verwijzen moet het NR van de OPEN-instructie gelijk zijn aan het NR van de PRINT#-instructie. Het komt in het kort hier op neer: als u een file met OPEN 1,6,1 opent, is de eerste 1 het referentienummer. Als instructies met betrekking tot de geopende

file uitgevoerd moeten worden, moet de PRINT#-instructie als volgt luiden:

PRINT# 1, ...

Met de rest hoeft u nu nog geen rekening te houden. Dat komt aan de orde bij de secundaire adressen. Nog een opmerking over de structuur van het woord PRINT#, die u overigens ook in het handboek van de Commodore 64 kunt terugvinden. Als u het woord PRINT# wilt afkorten, wordt dat vaak geprobeerd met ?#. Dat levert SYNTAX ERROR op. De juiste afkorting is P SHIFT R.

CLOSE

Deze instructie sluit een file die met OPEN werd geopend. Een PRINT#-instructie heeft op dat nummer dus geen zin meer. De vorm van de CLOSE-instructie luidt:

CLOSE NR

Een file die u met OPEN 2,6,2 hebt geopend, moet u dus met CLOSE 2 sluiten. Probeer u daarna maar eens PRINT#2,2 in te voeren. De plotter kan deze instructie niet meer ontvangen waardoor de foutmelding FILE NOT OPEN op het scherm verschijnt.

CMD

De CMD-instructie is ook op de plotter 1520 van toepassing. Omdat de 1520 zeer geschikt is om kleinere programma's te listen, is deze instructie van wezenlijk belang. Hoe kunt u de aanwijzing nu op de juiste wijze gebruiken om bijvoorbeeld een programma te listen? Hiervoor zijn twee mogelijkheden. U kunt bijvoorbeeld - als het programma is geladen - de instructies direct zonder programmaregels invoeren. De volgorde van de stappen is dan:

- a Het programma moet in het geheugen van de Commodore 64 voorhanden zijn.
- b U voert OPEN 4,6 in, want de CMD-instructie werkt alleen in combinatie met OPEN.
- c Daarna voert u CMD 4 in om alle programma-onderdelen rechtstreeks naar het met OPEN geopende apparaat te zenden.
- d Nu wordt de instructie verwacht en kunt u bijvoorbeeld LIST invoeren. Het programma dat in de computer zit, wordt door de plotter gelist.
- e De CMD-instructie is ook nog werkzaam na het listen en daarom moet het kanaal weer vrijgemaakt worden. Daartoe

voert u achtereenvolgens PRINT# NR en CLOSE NR in.

Kort samengevat geldt voor de directe invoer:

```
OPEN NR,6  
CMD NR  
LIST  
PRINT# NR  
CLOSE NR
```

De CMD-instructie hangt niet af van de rechtstreekse invoer, hoewel die voor het listen meer geschikt lijkt, omdat de invoer die tot het LISTproces leidt niet op de listing verschijnt. U kunt de CMD natuurlijk ook gebruiken tijdens de afloop van een programma-instructie. Hiervoor moet u het volgende programma maar eens bestuderen:

```
10 OPEN4,6  
20 CMD4,"PLOTTER 1520"  
30 LIST
```

In regel 10 wordt de plotter geopend om te schrijven. Regel 20 werkt volgens het principe CMD 4, instructie. De instructie is in dit voorbeeld gewoon een tekst die tussen aanhalingstekens moet staan. Als u dit programma laat draaien, is de file na het listen nog geopend en moet u dit bestand met PRINT# 4 en CLOSE 4 in de directe modus sluiten. Breid het programma uit met

```
40 PRINT# 4  
50 CLOSE 4
```

en laat het draaien. Na afloop voert u CMD 4, "PLOTTER 1520" in. Hoewel er een file wordt aangesproken die in regel 50 al werd gesloten, zult u nu toch geen foutmelding krijgen. Dit is niet terug te voeren op een fout in uw programma of in de plotter, maar op het bedrijfsstelsel van de Commodore 64. Daar wordt het programma namelijk afgebroken na een LIST-instructie. Daarom zijn de regels 40 en 50 nutteloos en moet het bestand in de directe modus worden gesloten.

8.2 De secundaire adressen van de Commodore 1520

Dit gegeven was bij de OPEN-instructie het laatste cijfer in onze serie adressen. Het kan tussen 0 en 7 liggen en vervult dan bepaalde functies, die met PRINT# NR kunnen worden opgeroepen. De betekenis van de cijfers 0-7 staan in de vorige paragraaf. We nemen nu alle adressen door en lichten ze toe met kleine programma's.

IN = 1

Bij dit eerste cijfer zullen we iets langer stilstaan dan in de handleiding van de 1520 het geval is. Dit cijfer reserveert de instructieset voor tekenroutines. We moeten tot een goed begrip van de relative, absolute en overige tekenfuncties komen, om de tekenkunst van het apparaat zo effectief mogelijk te kunnen gebruiken. Eerst moeten we het secundaire adres bekendmaken. We kiezen OPEN 1,6,1. Nu kunnen we de instructieset van de plotter aanspreken. Dat gebeurt met een PRINT#-instructie. Er zijn drie mogelijkheden:

```
PRINT# 1, "aanwijzing"; x-gegevens; y-gegevens  
PRINT# 1, "aanwijzing", x-gegevens, y-gegevens  
PRINT# 1, "aanwijzing" x-gegevens, y-gegevens
```

Aanwijzing, x-gegevens en y-gegevens kunnen gescheiden worden door een puntkomma, een komma of een spatie (space). Er zijn in totaal zes aanwijzingen die de plotter begrijpt.

1 H brengt de pennen in de uitgangspositie, dus naar de linkerkant van de rol. Deze plaats wordt het absolute nulpunt genoemd. Dit is de absolute beginpositie $x=0:y=0$. Op dit absolute nulpunt hebben ook de instructies D en M betrekking.

2 M staat voor beweging van de pennen gerelateerd aan het absolute nulpunt aan de linkerkant van de plotter. Het uitgangspunt van de instructie blijft dus steeds $x=0:y=0$! De stift is van het papier gehaald en de pennen bewegen zich naar de aangegeven coördinaten ($x=0-480:y=+-999$), maar er wordt nog niet getekend!

3 D is in gebruik identiek met M, maar nu wordt de elektromagneet geactiveerd en drukt de metalen strip de gekozen ballpoint tegen het papier. De plotter tekent volgens de opgegeven coördinaten. Ook de D-instructie heeft altijd betrekking op het absolute nulpunt, dus op de positie (0/0) aan de linkerkant van de rol.

De invoer gaan we straks aan de hand van voorbeelden toelichten maar eerst willen we u de drie overige tekeninstructies van de 1520 geven:

4 I Naast het absolute nulpunt (0/0) bestaat er nog een ander nulpunt. Dit relatieve nulpunt kunt u naar wens kiezen en wijzigen. U gaat bijvoorbeeld eerst met M naar een bepaalde positie en daar voert u PRINT# 1, "I" in. De positie waarop u staat is dan het actuele relatieve nulpunt. Als u de pennen op een andere positie zet en weer PRINT#, "I" invoert, wordt het relatieve nulpunt door het nieuwe vervangen. Net als bij H hebben op de instructie I ook twee stuuraanwijzingen betrekking.

5 R komt overeen met de M-instructie: de pen is van het papier gehaald en de houder gaat naar de aangegeven x/y-positie. Voor de R-instructie geldt een ander referentiepunt, namelijk het relatieve nulpunt. Als u het relatieve nulpunt bijvoorbeeld op 100/-100 zet en dan PRINT# 1,"R" ;1;1 invoert, gaat de plotter bij zijn stappen uit van 100/-180 en niet van bijvoorbeeld 0/0! Om hetzelfde effect te bereiken vanuit het absolute nulpunt zou u met de M-instructie moeten werken en als coördinaten 101/-99 moeten invoeren.

6 J is de tekeninstructie gerelateerd aan het relatieve nulpunt en komt in het coördinatensysteem overeen met D. De stift wordt gezet en er wordt een lijn getrokken naar de gewenste positie.

Na deze uitleg gaan we over naar de toepassing van deze instructies. We proberen een driehoek te construeren om aan de hand daarvan de verschillen te demonstreren tussen de absolute en de relatieve manier van tekenen. Kijkt u maar eens naar de volgende routine:

```
10 OPEN1,6,1
20 PRINT#1,"M";100;0
30 PRINT#1,"D";300;0
40 PRINT#1,"D";200;100
50 PRINT#1,"D";100;0
60 PRINT#1,"H"
70 PRINT#1,"M";0;-100
80 CLOSE1
```

Als u het programma hebt overgenomen, kunt u het runnen. U krijgt dan de driehoek te zien. Het is duidelijk dat het hierbij om de absolute tekenmanier gaat, daar alleen de instructies H, M en D een rol spelen. In regel 20 beweegt de kop van het schrijfinstrument naar de beginpositie 100/0. De regels 30-50 tekenen de driehoek, waarbij alle posities vanuit 0/0 berekend

moeten worden. De regels 60 en 70 schuiven het papier nog iets verder uit het apparaat, zodat de tekening zichtbaar wordt. Het wordt hier al duidelijk dat u vooral bij gecompliceerde tekeningen niet met absolute nulpunten moet werken omdat het referentiepunt niet ideaal is. Bij deze eenvoudige driehoek zou u al gemakkelijker kunnen werken met een mengvorm van relatieve en absolute instructies zoals in het volgende programma.

```
10 OPEN1,6,1
20 PRINT#1,"R";100;0
30 PRINT#1,"I"
40 PRINT#1,"J";200;0
50 PRINT#1,"J";100;100
60 PRINT#1,"J";0;0
70 PRINT#1,"R";-100;-100
80 CLOSE1
```

Als u het programma runt, zult u aan het eindresultaat geen verandering kunnen ontdekken. Wel is de weg naar het doel iets anders. De fixering van het startpunt op 100/0 in regel 20 is identiek gebleven. Daarna wordt er niet direct en dus absoluut getekend, maar positie 100/0 wordt door PRINT# 1, "I" verklaard tot het relatieve nulpunt. De regels 40-60 tekenen met de instructie J de driehoek. De berekeningen zijn eenvoudiger geworden, omdat u direct naar de linkerhoek van de driehoek kunt verwijzen. Mengvormen van de instructies zijn in ieder geval aan te bevelen als u niet te veel werk in de berekeningen wilt steken. Als we nu de driehoek willen tekenen door alleen gebruik te maken van relatieve instructies, dan ziet het er zo uit:

```
10 OPEN1,6,1
20 PRINT#1,"M";100;0
30 PRINT#1,"I"
40 PRINT#1,"J";200;0
50 PRINT#1,"J";100;100
60 PRINT#1,"J";0;0
70 PRINT#1,"R";-100;-100
80 CLOSE1
```

In regel 20 werd M vervangen door R. Dat was mogelijk omdat het relatieve en het absolute nulpunt voor het tekenbegin identiek zijn en er daarom geen grote veranderingen in het programma nodig waren. U moet echter steeds de beste instructiecombinaties kiezen. Dat wordt nog wel duidelijk wanneer meer gecompliceerde routines aan de orde komen of de afbeelding van wiskundige functies. U moet ervoor zorgen dat u voldoende over het

secundaire adres weet om zelf tekeningen te kunnen ontwerpen. Sla deze uitleg dus niet over maar voer de gegevens in en analyseer het resultaat.

IN = 2

Met dit secundaire adres is het mogelijk in een programma de pennen te draaien en zo van kleur te wisselen. Dit adres vervangt de COLOR CHANGE-toets. U moet eerst het bestand openen met OPEN 2,6,2 of OPEN NR,6,2. Daarbij moet u rekening houden met wat bij OPEN is uitgelegd (NR tussen 1 en 255). Met PRINT# 2, kleur kunt u een van de vier pennen uitkiezen. Het gaat om de volgende getallen:

zwarte stift = 0,4,8,12,16.....($0+n*4<255$)
blauwe stift = 1,5,9,13,17.....($1+n*4<255$)
groene stift = 2,6,10,14,18.....($2+n*4<255$)
rode stift = 3,7,11,15,19.....($3+n*4<255$)

Als u voor de kleur een cijfer > 255 kiest, wordt automatisch de kleur zwart ingesteld. Het is u vast al opgevallen dat de plotter met zwart begint. Bij de zelftest is de volgorde overigens blauw, groen, rood, zwart. Als de beschermkap geopend is, ziet u naast een van de pennetjes een flinterdun streepje metaal. Daarmee controleert de 1520 de stand van de pennen. De kleur die u hebt gekozen, kunt u aan de 1520 meedelen als getal tussen aanhalingstekens, als gewoon getal of als variabele. Het volgende voorbeeld geeft de combinatie van de kleuren en hun volgorde aan. U kunt dit lijstje natuurlijk ook uitprinten en als geheugensteuntje gebruiken:

```
100 REM===1520 INSTELLEN=====
110 OPEN4,6
120 OPEN2,6,2
130 REM===NAMEN INLEZEN=====
140 FORF=0TO3
150 READF$(F)
160 NEXT
170 REM===KLEUR KIEZEN/PRINTEN=====
180 FORF=0TO3
190 PRINT#2,F
200 PRINT#4,F;F$(F)
210 NEXT
220 REM===DATA=====
230 DATA ZWART,BLAUW,GROEN,ROOD
240 REM===EINDE=====
250 CLOSE2
260 CLOSE4
```

In het eerste gedeelte van het programma worden de namen van de vier kleuren in een veld ondergebracht. Het tweede deel (vanaf regel 180) verbindt een naam met een kleur en plot het resultaat uit. Het tweede programma voor dit secundaire adres is een rekenhulp voor de kleurcijfers, die tussen 0 en 255 mogen liggen. Het programma vraagt naar een cijfer en levert de kleur die erbij hoort. Daarna een routine die het proces omkeert en alle mogelijke getallen voor een bepaalde kleur levert.

```

100 REM===DATA INLEZEN=====
110 FORF=0TO3
120 READF$(F)
130 NEXT
140 REM===INVOER/BEREKENING=====
150 INPUT" sPARAMETER VOOR DE KLEUR (0-3)
";K$
160 K1=VAL(K$):K=K1:IF K<0 OR K>3 THEN G
OTO150
210 REM===UITVOER=====
220 PRINT
230 PRINTK1"= "F$(K)
240 REM===OPNIEUW BEGINNEN=====
250 FORN=1TO1000
260 NEXT
270 RESTORE
280 GOTO100
290 REM===DATA=====
300 DATA ZWART,BLAUW,GROEN,ROOD

```

```

100 REM===INVOER=====
110 PRINT" sWELKE KLEUR?"
120 PRINT
130 PRINT"Z    B    G    R"
140 PRINT
150 GETA$:WAIT203,63
160 REM===KLEUR BEPALEN=====
170 IFA$="Z"THENK=0:GOTO230
180 IFA$="B"THENK=1:GOTO230
190 IFA$="G"THENK=2:GOTO230
200 IFA$="R"THENK=3:GOTO230
210 GOTO150
220 REM===BEREKENING=====
230 FORF=0TO63
240 G=K+F*4
250 PRINTG,,:
260 NEXT
270 REM===OPNIEUW BEGINNEN=====

```

```
280 PRINT
290 PRINT"<TOETS>"
300 WAIT203,63
310 GOTO100
```

IN = 3

Dit secundaire adres is van groot belang voor de bediening van de printer met ingebouwde karakterset. Met dit adres kunt u een keuze maken uit de vier tekengroottes en het aantal karakters per regel bepalen. Open het kanaal met OPEN NR,6,3. Met PRINT# 3, getal kunt u daarna de stapgrootte kiezen. Voor het getal bij de PRINT-instructie geldt hetzelfde als voor de getalskeuze bij secundair adres 2. De getallen moeten tussen 0 en 255 liggen. Zinvol zijn ook in dit geval vooral 0, 1, 2 en 3. Het getal mag rechtsstreeks, tussen aanhalingstekens of als variabele worden ingevoerd. De volgende tabel helpt u bij de keuze van de tekengrootte:

PRINT# 3,0 het kleinste schriftbeeld (80 karakters
per regel)

```
PRINT# 3,1  het normale schriftbeeld (40 karakters
            per regel)
```

```
PRINT# 3,2  het middelgrote schriftbeeld (20 karakters
            per regel)
```

```
PRINT# 3,3  het grootste schriftbeeld (10 karakters
             per regel)
```

Misschien bent u in het handboek het begrip defaultwaarde tegengekomen bij de kleurinstelling of bij de keuze van de tekengrootte. Deze aanduiding vermeldt alleen hoe de plotter is ingesteld op het moment dat u hem aanzet. Het bedrijfssysteem zet hem op normaal schrift met zwart, dat is alles. Indien u verschillende schriftgroottes wilt zien, moet u dit programma eens invoeren:

```
100 REM===1520 INSTELLEN=====
110 OPEN4,6
120 OPEN3,6,3
130 PRINT#3,0
140 REM===GROOTTE VAN TEKEN=====
150 FORZ=0TO3
160 PRINT#3,Z
170 REM===PRINTEN=====
180 PRINT#4,"PLOTTER"
190 Y=X+2
200 IFX<YTHENX=X+1:PRINT#4:GOTO200
230 NEXTZ
240 REM===EINDE=====
250 PRINT#3,0
260 CLOSE3
270 CLOSE4
```

Na het starten van het programma verschijnt op het papier de volgende afbeelding:

PLOTTER

PLOTTER

PLOTTER

PLOTTER

Nu de keuze van de tekens wel duidelijk zal zijn, nog een stukje theorie voor de technisch geïnteresseerden. Hoe worden letters en symbolen getekend? Als de Commodore 1520 een Y tekent, begint hij van onder af aan. Hij tekent naar rechtsboven. Het apparaat zou oneconomisch te werk gaan als het nu via dezelfde lijn terug zou

gaan om vervolgens de lijn naar linksboven te tekenen. De plotter is zo slim om rechtstreeks naar de linkerbovenhoek te gaan, waarbij de pen even van het papier wordt gelicht. Als de pen op de juiste positie staat, wordt hij weer op het papier geplaatst om de Y te voltooien. Andere karakters, die er wel ingewikkeld uitzien maar een duidelijke structuur hebben, worden geplotted zonder dat de stift van het papier genomen wordt. Tot deze karakters behoort bijvoorbeeld de apestaart (@), die van binnen naar buiten wordt getekend. Bij weer andere karakters, die met 1 strek op het papier kunnen worden gezet, worden sommige stukken twee keer getekend, als dat de kortste weg is. Een voorbeeld daarvan is het matje (#). Bij zo'n variatie in bewegingen is de snelheid van 13 karakters per seconde een wonder te noemen. Men heeft bij de constructie van de plotter een compromis weten te sluiten tussen snelheid, nauwkeurigheid en netheid. Een hogere snelheid zou een vertekening van het schriftbeeld tot gevolg hebben. Ook is er rekening gehouden met de ruimte tussen de karakters. Zelfs bij het kleinste formaat (80 karakters per regel) is dat zo geregeld dat de karakters nog duidelijk gescheiden zijn. Het totale beeld wordt niet vaag, ook niet als u langere tijd met het kleinere schriftbeeld werkt.

IN = 4

De Commodore 1520 bezit enkele nuttige bijzondere functies, waarvan er een in secundair adres 4 is opgenomen. Deze functies moeten ervoor zorgen dat sommige programma's gemakkelijker lopen. Via secundair adres 4 kunt u alle karakters van de Commodore 1520 90 graden gedraaid op papier zetten. Dat is handig voor tekst bij de Y-as van een assenstelsel.

De invoer ziet er als volgt uit:

```
OPEN nr,6,4  
PRINT# 4,getal
```

Het getal mag, zoals gewoonlijk, tussen 0 en 255 liggen. Bij voorkeur kiest u (voor de overzichtelijkheid) de cijfers 0 en 1. De geprogrammeerde standaardwaarde is na het aanzetten van het apparaat 0. Deze 0 staat dus voor het normale schriftbeeld. Als u nu na het openen van het bestand PRINT# 4,1 invoert, worden de ASCII-karakters 90 graden naar rechts gedraaid. Net als bij de kleurkeuze of bij de keuze van het schriftformaat kan ook hier een getal direct of in de vorm van een variabele aan het secundaire adres worden doorgegeven. De draaiing van de ASCII-karakters is overigens niet afhankelijk van de default- of standaardwaarde van het schrift. De karakters worden bij alle schriftgroottes gedraaid en ook in de gekozen schriftgrootte gerealiseerd. De

mogelijkheden van secundair adres 4 worden in de volgende voorbeelden kort besproken. Ze zijn bedoeld om u tot eigen experimenten aan te zetten:

```
100 REM===1520 INSTELLEN=====
110 OPEN5,6
120 OPEN4,6,4
130 PRINT#4,0
140 REM===PRINTEN=====
150 PRINT#5,"PLOTTER 1520"
160 PRINT#4,1
170 PRINT#5,"PLOTTER 1520"
180 REM===EINDE=====
190 PRINT#4,0
200 CLOSE4
210 CLOSE5
```

Dit is de eenvoudigste methode om karakters op de X-as te draaien, maar het is tevens de minst nette manier. Ten eerste liggen de karakters veel te dicht naast elkaar en ten tweede moet u van onderen naar boven lezen om PLOTTER 1520 te kunnen ontcijferen, nadat u het papier 90 graden heeft gedraaid. In de laatste routine stond niets nieuws; we staan er niet langer bij stil, maar gaan het proces omdraaien. Dat wil zeggen dat we een programma willen maken dat niet alleen de karakters (PLOTTER 1520) op de X-as draait, maar de tekst ook van boven naar beneden leesbaar maakt. De meest eenvoudige, maar om te vervaardigen meest moeilijke methode is het laatste programma te gebruiken en de tekst achterstevoren in te voeren. In plaats van PLOTTER 1520 voeren we dan in 0251 RETTOLP. Maar dit levert de volgende problemen op:

- grote tijdsinvestering bij het vervaardigen
- veel kans op fouten
- karakters staan te dicht naast elkaar

Het volgende programma biedt een oplossing voor die problemen:

```
100 REM===1520 INSTELLEN=====
110 OPEN5,6
120 OPEN4,6,4
130 OPEN1,6,1
140 PRINT#4,1
150 REM===TEKEN BEREKENEN=====
160 X$="PLOTTER 1520"
170 FORX=LEN(X$)TO1STEP-1
```

```

180 Y$=MID$(X$,X,1)
190 REM===TEKEN PRINTEN=====
200 PRINT#1,"M",20*(LEN(X$)-X);0
210 PRINT#5,Y$;:
220 NEXT
230 REM===EINDE=====
240 PRINT#4,0
250 PRINT#1,"M";0;0
260 CLOSE1
270 CLOSE4
280 CLOSE5

```

Voor een beter begrip geven we een analyse van de belangrijke regels 150-220. Allereerst wordt de tekst in de variabele X\$ gezet (regel 160). In regel 170 begint een FOR-NEXT-lus, die alle posities van de karakters van voren naar achteren afhandelt. Om ook de positie van het laatste karakter te bereiken, moet u met LEN(x\$) werken, zodat niet steeds de tekstlengte geteld hoeft te worden. Voor regel 170 zouden we ook

FORX=12 to 1 STEP -1

mogen schrijven, als we de letters van tevoren geteld hebben. In Y\$ (regel 180) leest de MID\$-instructie telkens een karakter uit X\$ en wel van achteren naar voren, omdat regel 170 terugtelt. Regel 200 brengt de pen naar de juiste positie met de M-instructie. De positie op de X-as kan gemakkelijk worden berekend. U ziet daar de term $20 - (\text{LEN}(X\$) - X)$. $\text{LEN}(X\$)$ is het aantal karakters in X\$. Omdat we van links naar rechts schrijven, moet de afstand tot het absolute nulpunt 0/0 dus ook groter worden. In dit voorbeeld gebeurt dat door X van $\text{LEN}(X\$)$ af te trekken en met een constante (hier 20) te vermenigvuldigen. Omdat X kleiner wordt (STEP -1 in regel 170) wordt het verschil kleiner en de factor daardoor groter. De pennen lopen steeds verder van de rechterkant van de rol vandaan. In regel 210 wordt de letter dan op papier gezet. U mag de puntkomma niet vergeten, omdat het schrift anders diagonaal zou verschijnen. Met de NEXT-instructie wordt de lus in regel 170 opnieuw gestart. Omdat dezelfde routine in het handboek van de Commodore 1520 niet verder wordt uitgelegd, meenden wij de verschillende stappen in het programma nog eens goed te moeten bespreken. De draaiing van de karakters langs de X-as is nu wel goed besproken, dus gaan we naar de Y-as. Als u de laatste routine had ingetypt, wis deze dan nog niet; u kunt er veel regels van overnemen.

```

100 REM===1520 INSTELLEN=====
110 OPEN5,6
120 OPEN4,6,4
130 OPEN1,6,1
140 PRINT#4,1
150 REM===TEKEN BEREKENEN=====
160 X$="PLOTTER 1520"
170 FORX=1TOLEN(X$)
180 Y$=MID$(X$,X,1)
190 REM===TEKEN PRINTEN=====
200 PRINT#5,Y$
210 NEXT
220 REM===EINDE=====
230 PRINT#4,0
240 PRINT#5
250 CLOSE1
260 CLOSE4
270 CLOSE5

```

In de regels 150 tot 210 wordt de tekst PLOTTER 1520 van voor naar achter gelezen en geprint.

IN = 5

Een van de nuttigste functies uit het bedrijfssysteem van de Commodore 1520 is de linieerfunctie. Deze wordt geactiveerd door secundair adres 5 en heeft een capaciteit van 16 lijnsoorten. Deze variëren van ononderbroken tot stippellijn. Net zoals bij de andere adressen gebruikelijk was, opent u het bestand met OPEN nr,6,5. Daarna volgt PRINT# 5, getal. Het getal mag tussen 0 en 255 liggen. Ook hier mag u het getal direct, als variabele of als cijfer tussen aanhalingstekens invoeren. Zinvol zijn vooral de nummers 0-15. De waarde bij het inschakelen van de Commodore 1520 is 0, de standaardlijn. Dat is een ononderbroken lijn. Als u de 0 vervangt door een getal tussen 0 en 15, wordt het gewenste karakterset geprint. Bekijk u eerst eens alle realiseerbare lijnen. Daarna volgt een programma waarmee u een type kunt uitkiezen en printen.

```

0 _____
1 _____
2 _____
3 -----
4 -----
5 -----
6 -----
7 -----
8 -----
9 -----
10 -----
11 -----
12 -----
13 -----
14 -----
15 -----

```

```

100 REM===1520 INSTELLEN=====
110 OPEN4,6
120 OPEN5,6,5
130 OPEN1,6,1
140 PRINT#5,0
150 REM===INVOER=====
160 INPUT"sWELK LIJNNUMMER";A$
170 A=VAL(A$)
180 IFA<0ORA>15THEN160
190 PRINT
200 INPUT"LENGTE IN CM (MAX. 8 CM)";B$
210 B=VAL(B$)
220 IFB<0ORB>8THENPRINT"qq":GOTO200
230 REM===TEKENING=====
240 L=50*(B+1)
250 PRINT#4,A;
260 PRINT#5,A;
270 PRINT#1,"D";L;0
280 PRINT#5,0
290 PRINT#4
300 REM===EINDE=====
310 CLOSE1
320 CLOSE5
330 CLOSE4

```

De regels 100-140 stellen de waarde van de standaardlijn in en openen de nodige files. Bij de INPUT-invoer in regel 160 duikt er een nieuw teken op. Het betreft een reverse afgebeeld hartje, dat gewoonlijk verschijnt als u SHIFT/HOME indrukt. De Commodore 1520 kan geen reverse karakters produceren (we komen hier later op

terug). Daarom onderstreept hij de betreffende letters. De onderstreepte s staat dus voor SHIFT/HOME. De hele set van bijzondere karakters wordt na de secundaire adressen besproken. Nadat u in het programma de invoer hebt verzorgd, begint het deel van het programma dat de lijn print (regel 240-290). Daartoe moet het aantal centimeters eerst in plotter-lengte-eenheden veranderd worden. Dit gebeurt in regel 240. In regel 250 wordt het lijnnummer voor de lijn geprint. Regel 260 geeft de juiste waarde voor secundair adres 5. De lijn zelf ontstaat door de D-instructie; hij wordt getekend tot aan de x-coördinaat L. Meet u de gekozen lijnen maar eens na. De fouten zijn minimaal en de stapgrootte van 0.2 millimeter wordt dus zeer nauwkeurig nageleefd. De linieerfunctie biedt ongekende mogelijkheden; ook meer ingewikkelde figuren kunnen met deze functie worden gemaakt.

```

100 REM====1520 INSTELLEN=====
110 OPEN1,6,1
120 OPENS,6,5
130 PRINT#5,5
140 REM===TEKENEN=====
150 PRINT#1,"M";100;0
160 PRINT#1,"I"
170 PRINT#1,"J";200;0
180 PRINT#1,"J";100;100
190 PRINT#1,"J";0;0
200 PRINT#1,"R";-100;-100
210 REM===EINDE=====
220 PRINT#5,0
230 CLOSE1
240 CLOSE5

```

Als voorbeeld nemen we de reeds eerder besproken driehoek, aan de hand waarvan we de relatieve en absolute tekeninstructies hebben gedemonstreerd. De tekenroutine is wat betreft positie en lengte van de zijden van de driehoek identiek. Het verschil is dat regel 120 en 130 secundair adres 5 aanspreken en dat PRINT# 5,5 de lijnsoort kiest. De plotter tekent vervolgens de driehoek als drie stippellijnen. Deze manier van tekenen is vooral makkelijk als u in een statistisch programma veel rechte lijnen (meetwaarden) moet verwerken en de vier kleuren niet meer voldoende zijn om ze te onderscheiden. Als het moet, kunt u 64 curven of rechte lijnen tekenen die nog goed van elkaar zijn te onderscheiden. Per kleur staan tenslotte 16 lijnsoorten ter beschikking.

IN = 6

In het gebruik van hoofdletters en kleine letters is de plotter wat eigenzinnig, ook als de Commodore 64 op het kleine lettertype is ingesteld. Dit is noodzakelijk omdat de Commodore 1520 niet over een grafische karakterset beschikt. Dat kunt u zien als u in de hoofdlettermodus van de computer bijvoorbeeld het woord PLOTTER 1520 print. Met SHIFT verschijnen er dan geen grafische tekens maar kleine letters en zonder SHIFT hoofdletters. Dat laatste is normaal. In de kleine lettermodus (PRINTCHR\$(14)) verandert het vreemde gedrag van de plotter niet. Op het beeldscherm komen kleine letters te staan en met SHIFT hoofdletters, maar de plotter doet precies hetzelfde als in de hoofdlettermodus. In de kleine lettermodus reageert de plotter dus tegengesteld aan de computer. De Commodore 1520 registreert de omschakeling niet en daarom kan zijn bedrijfsysteem er niet mee uit de voeten. Via het secundaire adres 6 kunt u uw bedoelingen aan het apparaat kenbaar maken. Er zijn twee mogelijkheden:

1) de hoofdlettermodus met kleine letters via SHIFT = 0

2) de kleine lettermodus met hoofdletters via SHIFT = 1

Open de file met OPEN nr,6,6 en toets daarna PRINT# 6,0 of PRINT# 6,1 in. De laatste optie komt overeen met de kleine lettermodus, waarin u met SHIFT hoofdletters krijgt. Voor de plotter maakt het niets uit of de computer in de hoofdletter- of de kleine lettermodus staat; het geSHIFTE karakter verschijnt als letter op het papier, of het op het beeldscherm nu als grafisch symbool of als letter te zien is (afhankelijk van de modus). De geprogrammeerde standaardwaarde komt overeen met PRINT# 6,0. Dat wil zeggen dat geSHIFTE letters als kleine letters worden geprint. Om het onderscheid tussen de twee toestanden te verduidelijken moet u de karakterset met het volgende programma listen. U kunt het beste de Commodore 64 op de kleine lettermodus zetten voor u de routine intypt. Bij grafische symbolen is het moeilijk te zien welke letter er achter schuilgaat. De listing zelf is ook in de kleine lettermodus geschreven.

```
100 REM===1520 INSTELLEN=====
110 OPEN4,6
120 OPEN6,6,6
130 PRINT#6,0
140 REM===TEKST=====
150 PRINT#4,"6,0 = HOOFDLETTERMODUS"
160 GOSUB270
170 PRINT#6,1
180 GOSUB270
```

```

190 PRINT#4,"6,1 = KLEINE LETTERMODUS"
200 GOSUB270
210 PRINT#6,0
220 GOSUB270
230 REM===EINDE=====
240 CLOSE6
250 CLOSE4:END
260 REM===PLOTTE=====
270 FORX=1TO35
280 PRINT#4,CHR$(90-X);:
290 NEXT
300 PRINT#4
310 RETURN

```

Na het starten van het programma krijgt u deze print:

```

6,0 = HOOFDLETTERMODUS
YXWUUTSRQPONMLKJIHG FEDCBA@?>=<;:987
yxwuutsrqponmlkji hg fedcba-?>=<;:987
6,1 = kleine lettermodus
yxwuutsrqponmlkji hg fedcba-?>=<;:987
YXWUUTSRQPONMLKJIHG FEDCBA@?>=<;:987

```

Het resultaat van het programma bewijst ons verhaal. In de normale modus tekent de plotter hoofdletters en met SHIFT de kleine letters. In de kleine lettermodus doet hij precies het tegenovergestelde. We hebben secundair adres 6 later nodig bij het typemachineprogramma in de directe modus.

IN = 7

We zijn aangeland bij het laatste secundaire adres, nummer 7. Dit kunnen we zeer kort behandelen, omdat we het bijna nooit nodig hebben. U geeft eerst OPEN 7,6,7 en daarna PRINT# 7. Direct daarna voert de Commodore 1520 de zelftest uit. Dit doet hij ook automatisch bij het inschakelen. Als secundair adres 7 wordt aangesproken, staat het apparaat weer in de begintoestand. Dat betekent:

- zelftest
- alle secundaire adressen op default
- gegevens in werkgeheugen worden gewist

Na deze uitgebreide verhandeling kan ook een nieuweling de programmering ter hand nemen en routines ontwerpen die speciaal op zijn of haar wensen zijn toegesneden. Natuurlijk is het vanwege de vele mogelijkheden van de Commodore 1520 onmogelijk van alle soorten programma's een voorbeeld te geven. Wel hebben we getracht zoveel mogelijk thema's aan te snijden om u tot eigen experimenten aan te zetten. Sommige geometrische figuren of routines werden uitvoerig geanalyseerd om u inzicht te geven in de programmering.

Tot op heden hebben we secundair adres 0 buiten beschouwing gelaten. Adres 0 zorgt voor het printen van de ASCII-tekens. Dit moet in samenhang met de karakterset worden besproken om de verschillen met het beeldscherm of de printeruitvoer te tonen. Laten we eerst eens de normale modus van de Commodore 1520 bekijken. De volgende tabellen laten een gedeelte van de karakterset zien, op normale wijze en met SHIFT.

\$ (, 0 4 8 < @ D H L P T X \$ (, 0 4 8 < @ D H L P T X \$

%) - 1 5 9 = A E I M Q U Y J %) - 1 5 9 = A E I M Q U Y J

& * . 2 6 : > B F J N R U Z + & * . 2 6 : > B F J N R U Z +

' + / 3 7 ; ? C G K O S W [+ ' + / 3 7 ; ? C G K O S W [+

U ziet dat de karakterset met de beeldschermuitvoer correspondeert van CHR\$(33) tot CHR\$(95). Bij de geSHIFte letters, die op het beeldscherm als grafische symbolen verschijnen vanaf CHR\$(161), wordt in de normale plottermodus een klein lettertype uitgevoerd. Als u met PRINT# 6,1 naar de kleine lettermodus overschakelt, geeft de plotterlisting precies het omgekeerde. Hier geldt vanaf CHR\$(33) het tweede gedeelte van de tabel. Tot zover treden er geen grote verschillen op in de vorm van de uitvoer. Maar hoe staat het met stuurtekens die op het beeldscherm of op de printer verschijnen als reverse karakters? De Commodore 64 heeft heel veel van dat soort symbolen, van de cursorbesturing tot aan de kleurkeuze voor de karakters. We weten dat de Commodore 1520 vanwege de tijdsduur geen reverse karakters kan realiseren; die dingen precies uittekenen zou veel te lang duren. Maar hoe doen we dat nou als we deze bijzondere tekens willen gebruiken om de omslachtige CHR\$(x)-waarden te omzeilen? Daar is het volgende op bedacht. De reverse karakters worden als de letters geprint die als basisfunctie op de betreffende toets staan (bij cursor omlaag de Q). Dan wordt dit teken onderstreept, zodat u het teken als reverse teken kunt identificeren bij het lezen van de listing. Om te wennen aan deze plotterschrijfwijze hebben we een tabel gemaakt die het verschil aangeeft tussen een printer en de plotter 1520 ten aanzien van de uitvoer van reverse karakters. Achter elk symbool staan de toetsen waarmee u de bijzondere tekens kunt oproepen. De kleine lettermodus wordt ook behandeld, omdat de tekens daar op een andere manier worden opgeroepen.

bijzondere tekens	BIJZONDERE TEKENS	BIJZONDERE TEKENS
= functietoets	= FUNCTIETOETS	= FUNCTIETOETS
= functietoets	= FUNCTIETOETS	= FUNCTIETOETS
= functietoets	= FUNCTIETOETS	= FUNCTIETOETS
= functietoets	= FUNCTIETOETS	= FUNCTIETOETS
= functietoets	= FUNCTIETOETS	= FUNCTIETOETS
= functietoets	= FUNCTIETOETS	= FUNCTIETOETS
= functietoets	= FUNCTIETOETS	= FUNCTIETOETS
= functietoets	= FUNCTIETOETS	= FUNCTIETOETS
= delay	= DELAY	= DELAY
= home	= HOME	= HOME
= clear	= CLEAR	= CLEAR
= cursor omlaag	= CURSOR OMLAAG	= CURSOR OMLAAG
= cursor omhoog	= CURSOR OMHOOG	= CURSOR OMHOOG
= cursor rechts	= CURSOR RECHTS	= CURSOR RECHTS
= cursor links	= CURSOR LINKS	= CURSOR LINKS
= control/blk	= CONTROL/BLK	= CONTROL/BLK
= control/wht	= CONTROL/WHT	= CONTROL/WHT
= control/red	= CONTROL/RED	= CONTROL/RED
= control/cyn	= CONTROL/CYN	= CONTROL/CYN

= control/pur	= CONTROL/PUR	= CONTROL/PUR
= control/grn	= CONTROL/GRN	= CONTROL/GRN
= control/blu	= CONTROL/BLU	= CONTROL/BLU
= control/yel	= CONTROL/YEL	= CONTROL/YEL
= control/rvs on	= CONTROL/RVS ON	= CONTROL/RVS ON
= control/rvs off	= CONTROL/RVS OFF	= CONTROL/RVS OFF
= c=/blk	= C=/BLK	= C=/BLK
= c=/wht	= C=/WHT	= C=/WHT
= c=/red	= C=/RED	= C=/RED
= c=/cyn	= C=/CYN	= C=/CYN
= c=/pur	= C=/PUR	= C=/PUR
= c=/grn	= C=/GRN	= C=/GRN
= c=/blu	= C=/BLU	= C=/BLU
= c=/yel	= C=/YEL	= C=/YEL

Houdt u enkele symbolen achter de hand: in de volgende programma's heeft u er een aantal nodig.

8.3 Allerlei soorten figuren

Na de theorie over de plotter 1520 gaan we het geleerde in praktijk brengen. We houden ons eerst bezig met de constructie van eenvoudige geometrische figuren, beginnend met rechthoek, cirkel en ellips. Hierbij gaan we wiskundig verantwoord te werk. Er worden stralen getekend en de eenheden zijn meestal centimeters. Bij de programma's geven we steeds een korte uitleg en na de listing hebben we de afbeelding afgedrukt, zodat u fouten gemakkelijk kunt opsporen.

8.3.1 Vierkant

```

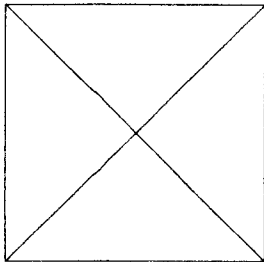
100 REM===VIERKANT=====
110 OPEN1,6,1
120 OPEN2,6,2
130 PRINT#2,1

```

```

140 REM===TEKENEN=====
150 PRINT#1,"M";100;-50
160 PRINT#1,"I"
170 S=200
180 PRINT#1,"J";S;0
190 PRINT#1,"J";S;S
200 PRINT#1,"J";0;S
210 PRINT#1,"J";0;0
220 REM===DIAGONALEN=====
230 PRINT#2,3
240 PRINT#1,"J";S;S
250 PRINT#1,"R";S;0
260 PRINT#1,"J";0;S
270 REM===EINDE=====
280 PRINT#1,"R";-100;-100
310 CLOSE1
320 CLOSE2

```



U ziet het aan de listing: een vierkant construeren is zeer eenvoudig, temeer daar alle zijden even lang zijn. In regel 170 wordt de lengte van de zijden (S) vastgelegd. Van regel 180 tot 210 wordt het eigenlijke vierkant getekend. Daarna volgen beide diagonalen, die ook dezelfde lengte hebben.

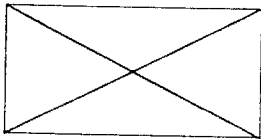
8.3.2 Rechthoek en kubus

Eerst geven we u de listing van een rechthoek met diagonalen:

```

100 REM===RECHTHOEK=====
110 OPEN1,6,1
120 OPEN2,6,2
130 PRINT#2,1
140 REM===TEKENEN=====
150 PRINT#1,"M";100;-50
160 PRINT#1,"I"
170 S=200;H=100
180 PRINT#1,"J";S;0
190 PRINT#1,"J";S;H
200 PRINT#1,"J";0;H
210 PRINT#1,"J";0;0
220 REM===DIAGONALEN=====
230 PRINT#2,3
240 PRINT#1,"J";S;H
250 PRINT#1,"R";S;0
260 PRINT#1,"J";0;H
270 REM===EINDE=====
280 PRINT#1,"R";-100;-100
310 CLOSE1
320 CLOSE2

```



Als u het programma van de rechthoek vergelijkt met dat van het vierkant, kunt u vaststellen dat er nauwelijks verschillen zijn. Alleen werd in regel 170 een tweede variabele ingevoerd ten behoeve van een andere lengte van de zijde. De tekenroutine werd daaraan aangepast. Deze voorbeelden zijn zo eenvoudig dat nadere analyse onnodig is. We proberen nu een ruimtelijke figuur te maken. Om te beginnen nemen we een eenvoudige figuur, zoals de kubus. Voordat we met het tekenen van meer ingewikkelde figuren beginnen, moeten we de manier van programmeren veranderen om het typewerk zo kort mogelijk te houden. Daarbij moeten we zoeken naar een compromis tussen een overzichtelijke programmastructuur en geringe tijdsinvestering. Dat een dergelijk compromis bestaat, laten we zien in het voorbeeld van de kubus. Het programma wordt geoptimaliseerd, maar verliest niets aan duidelijkheid. Maar eerst laten we zien hoe u het probleem van de kubus beter niet kunt aanpakken, als u tijd wilt besparen.

```

100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 OPEN2,6,2
130 OPENS,6,5:PRINT#5,0
140 REM===TEKENEN=====
150 X=100:Y=-50:S=200
160 PRINT#1,"M";X;Y
170 PRINT#1,"I"
180 PRINT#1,"J";S;0
190 PRINT#1,"J";S;S
200 PRINT#1,"J";0;S
210 PRINT#1,"J";0;0
220 X=70:Y=70
230 PRINT#1,"R";S;0
240 PRINT#1,"I";
250 PRINT#1,"J";X;Y
260 PRINT#1,"J";X;Y+S
270 PRINT#1,"J";0;S
280 PRINT#1,"R";-S;S
290 PRINT#1,"J";-S+X;Y+S
300 PRINT#1,"J";X;Y+S
310 REM===STIPPELEN=====
320 PRINT#5,7
330 PRINT#1,"R";X;Y
340 PRINT#1,"J";-S+X;Y
350 PRINT#1,"J";-S;0
360 PRINT#1,"R";-S+X;Y+S
370 PRINT#1,"J";-S+X;Y
380 REM===EINDE=====
390 PRINT#1,"M";0;-100
400 PRINT#5,0
410 CLOSE1
420 CLOSE2
430 CLOSE5

```

In de regels 150 en 220 worden de zijden van de figuur in de variabelen X, Y en S gezet. Daarna wordt de kubus getekend (met de nodige stippellijnen) met PRINT#1, "instructie", coördinaten. De coördinaten zijn een minpunt, want u moet de posities eerst uit het hoofd berekenen om de routine te kunnen maken. En ook het hele gedoe met de PRINT#-instructies is vervelend, evenals de lange weg die de pen aflegt. Dit programma voldoet dus absoluut niet en we korten het nu met twee stappen in, zonder dat de overzichtelijkheid verloren gaat. Een eerste stap in de goede richting is de PRINT#-instructies terugbrengen tot een acceptabel aantal. We zorgen er ook voor dat de pen de kortste weg kiest. Daarvoor moet het relatieve nulpunt met I verplaatst worden naar de linker bovenhoek van de voorkant (100/200). Dan kan het

buitenste deel van de figuur in een keer getekend worden en hoeft de pen geen omwegen meer te maken. De coördinaten worden met betrekking tot het relatieve nulpunt direct (niet als variabelen) in DATA-regels samengevat en vervolgens met READ ingelezen. Ook de instructies van de Commodore 1520 kunnen in DATA-regels worden ondergebracht (M,D,J,H). Als we deze maatregelen op het vorige programma toepassen, krijgen we de volgende listing:

```

100 REM===1520 INSTELLEN=====
110 OPENS,6,5
120 OPEN1,6,1
130 PRINT#5,0
140 PRINT#1,"M";100;200
150 PRINT#1,"I"
160 REM===DATA LEZEN/TEKENEN=====
170 FORX=0TO7
180 READY,Z
190 PRINT#1,"J";Y;Z
200 NEXT
210 REM===DATA LEZEN/TEKENEN=====
220 FORX=0TO7
230 READX$,Y,Z
240 IFX=2THENPRINT#5,7
250 PRINT#1,X$,Y,Z
260 NEXT
270 REM===EINDE=====
280 PRINT#5,0
290 CLOSE1
300 CLOSE5
310 REM===DATA (GEGEVENS)=====
320 DATA0,-200,200,-200,270,-130
330 DATA270,70,70,70,0,0,200,0
340 DATA270,70
350 REM===DATA (TEKENEN&STIPPELEN)===
360 DATAR,200,0,J,200,-200,R,270,-130
370 DATAJ,70,-130,J,0,-200,R,70,-130
380 DATAJ,70,70,M,0,-100

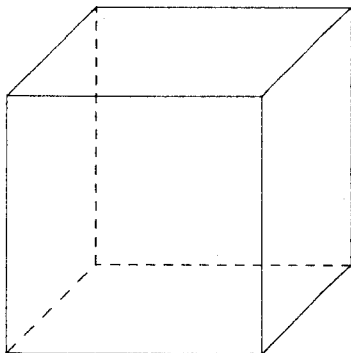
```

Als u het programma heeft geRUND, verschijnt er weer een kubus op papier. Het programma is wel korter dan het vorige, ondanks een veel uitvoeriger REM-documentatie. De teekentijd is korter omdat de weg is veranderd. Van regel 170-260 ziet u nu nog twee FOR-NEXT lussen, die bij het inlezen van de gegevens onderscheid maken tussen stippellijnen en ononderbroken lijnen. U krijgt de kortste vorm door beide leeslussen te koppelen en na het bereiken van een bepaalde waarde over te schakelen op liniëring (regel 190). De DATA-regels bevatten de totale kubus.

```

100 REM===1520 INSTELLEN=====
110 OPEN5,6,5
120 OPEN1,6,1
130 PRINT#5,0
140 PRINT#1,"M";100;200
150 PRINT#1,"I"
160 REM===DATEN INLEZEN TEKENEN=====
170 FORX=0TO15
180 READX$,Y,Z
190 IFX=10THENPRINT#5,7
200 PRINT#1,X$,Y,Z
210 NEXT
220 REM===EINDE=====
230 PRINT#5,0
240 CLOSE1
250 CLOSE5
260 REM===ALLE DATA=====
270 DATAJ,,,-200,J,200,-200,J,270,-130
280 DATAJ,270,70,J,70,70,J,,,J,200,0
290 DATAJ,270,70,R,200,,J,200,-200
300 DATAR,270,-130,J,70,-130,J,,,200
310 DATA R,70,-130,J,70,70,M,,,-100

```



Vergelijkt u deze listing eens met de eerste versie. U ziet dat economisch programmeren de snelheid van de Commodore 1520 positief beïnvloedt. De laatste listing blijft heel duidelijk omdat de coördinaten direct in de DATA-regels staan. Samenvattend gelden de volgende adviezen:

- nauwkeuring planning van de figuur
- de kortste weg voor de pen kiezen
- PRINT#-instructies vermijden ten gunste van DATA-regels

- leesbaarheid bevorderen (REM-regels en een bepaalde structuur)

Als u deze punten in de gaten houdt bij het ontwikkelen van tekenprogramma's zult u meer plezier beleven aan de programmering en kan de plotter zijn mogelijkheden volledig benutten. De volgende programma's gebruiken uitsluitend de beide laatste programmeermogelijkheden. Dat wil zeggen: of er wordt een onderverdeling in twee leeslussen gemaakt, of er vindt een samenvoeging plaats tot een DATA-blok.

8.3.3 Parallellogram

Nog 1 keer de uitvoerige programmering. Uitleg is overbodig.

```

100 REM===PARALLELLOGRAM=====
110 OPEN1,6,1
120 OPENS,6,5
130 PRINT#5,0
140 REM===BEGINPOSITIE=====
150 PRINT#1,"M";100;0
160 PRINT#1,"I"
170 REM===PARALLELLOGRAM TEKENEN=====
180 FORX=0TO3
190 READY,Z
200 PRINT#1,"J";Y;Z
210 NEXT
220 REM===HOOGTES TEKENEN=====
230 PRINT#5,3
240 FORX=0TO4
250 READX$,Y,Z
260 PRINT#1,X$,Y,Z
270 NEXT
280 REM===EINDE=====
290 PRINT#5,0
300 PRINT#1:CLOSE1
310 PRINT#5:CLOSE5
320 REM===DATA VAN PARALLELLOGRAM=====
330 DATA 200,0,250,100,50,100,0,0
340 REM===DATA VOOR HOOGTES=====
350 DATA R,50,0,J,50,100,R,200,100
360 DATA J,200,0,M,0,-100

```

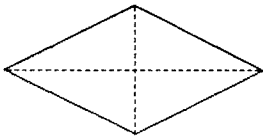


Ook een ruit is makkelijk te construeren:

```

100 REM====RUIT=====
110 OPEN1,6,1
120 OPENS5,6,5
130 PRINT#5,0
140 REM====BEGINPOSITIE==N=====
150 PRINT#1,"M";100;0
160 PRINT#1,"I"
170 REM====DATA LEZEN/TEKENEN=====
180 FORX=0TO7
190 READX$,Y,Z
200 IFX=4THENPRINT#5,3
210 PRINT#1,X$,Y,Z
220 NEXT
230 REM====EINDE=====
240 PRINT#5,0
250 CLOSE1
260 CLOSE5
270 REM====ALLE DATA=====
280 DATAJ,100,-50,J,200,,J,100,50,J,,
290 DATAJ,200,,R,100,50,J,100,-50
300 DATAM,,,-150

```



In de ruit zijn de diagonalen stippellijnen. De listing blijft overzichtelijk als we alle DATA's in een blok onderbrengen. U moet er bij het overtypen van de programma's wel om denken dat nullen weggelaten zijn en dat er dus meerdere komma's op elkaar kunnen volgen. De rest van het programma is in deze vorm al vaker voorgekomen en moet voor iedereen te volgen zijn.

8.3.4 Driehoeken 1

Voordat we verder gaan met de ruimtelijke figuren, behandelen we de constructie van driehoeken. We bekijken drie soorten driehoeken die in de geometrie veelvuldig aan de orde komen:

- a) gelijkzijdige driehoeken
- b) gelijkbenige driehoeken
- c) rechthoekige driehoeken

Laten we beginnen met de gelijkzijdige driehoeken. Zoals de naam al zegt, moeten alle drie de zijden van zo'n driehoek even lang zijn. Elke hoek is dan 60 graden. Om deze kennis in een tekening om te zetten, moet u eerst even nadenken over de praktische aanpak van het probleem. De basis kunt u zonder meer met de plotter tekenen, omdat de ligging ervan vrij gekozen kan worden. Het probleem is het volgende: we kennen weliswaar de lengte van alle zijden, maar het positieve referentiepunt in de positieve Y-richting ontbreekt. Met een kleine berekening kunt u de mogelijkheden omzeilen. Bekend is de lengte van alle drie de zijden. Als u een hoogtelijn tekent op de basiszijde, wordt de driehoek opgedeeld in twee gelijke, rechthoekige driehoeken. Nu kan de hoogte worden berekend. En met het eindpunt van de hoogtelijn heeft u tegelijkertijd het referentiepunt voor de beide ontbrekende zijden van de driehoek. Om de hoogte te berekenen zijn er twee mogelijkheden. Een methode is de stelling van Pythagoras, maar als u toch een computer hebt, kunt u ook gebruik maken van de sinusfunctie. Nu moeten we deze theorie in een algemeen geldende formule gieten, om bij invoer van de gewenste lengten meteen de hoogten te kunnen berekenen. En dan maar tekenen. De formule die zo dadelijk volgt, berust op de stelling van Pythagoras:

lengte van de zijden = 1
hoogte = h

algemeen geldt:

$$\begin{aligned}h^2 + (1/2)^2 &= 1^2 \\h^2 &= 1^2 - (1/2)^2 \\h^2 &= 3/4 * 1^2\end{aligned}$$

$$h = \text{SQR}(3/4 * 1^2)$$

Dit is de algemene formule voor de hoogte van de gelijkzijdige driehoek. We nemen als concreet voorbeeld een lengte van 4

centimeter. Voor h resulteert dat in $h = \text{SQR}(12)$ ofwel 3,464. Gebruik ter controle gerust een stukje papier. Teken daarop een basiszijde van 4 centimeter en de hoogte als middelloodlijn van 3,464 centimeter. De resterende lijnen van beide eindpunten van de basiszijde naar het tweede eindpunt van de hoogtelijn hebben een lengte van precies 4 centimeter. Nu blijkt dat de formule klopt, kunt u de verkregen kennis toepassen en een goed functionerend programma schrijven. Al is dit voorbeeld nog zo simpel, u ziet hoe belangrijk een goede voorbereiding ook bij kleine routines is.

```

100 REM===DRIEHOEK=====
110 OPENS,6,5
120 OPEN1,6,1
130 PRINT#1,"M";200;-50
140 PRINT#1,"I"
150 REM===INVOER=====
160 PRINT"§L = 1 - 8 CM"
170 INPUT"LENGTE VAN ZIJDE";A$
180 L=VAL(A$)
190 IFL<10RL>8THEN160
200 L=L*50
210 REM===BEREKENING/TEKENING=====
220 H=SQR(3/4*L^2)
230 PRINT#1,"J";-L/2;0
240 PRINT#1,"J";0;H
250 PRINT#1,"J";L/2;0
260 PRINT#1,"J";0;0
270 REM===HOOGTE TEKENEN=====
280 PRINT#5,5
290 PRINT#1,"J";0;H
300 PRINT#5,0
310 REM===EINDE=====
320 PRINT#1,"M";0;-100
330 CLOSE1
340 CLOSE5

```

In regel 220 staat de algemene formule van daarnet. Voer maar een paar waarden in en meet de zijden na. De verkregen gegevens komen overeen met de berekeningen en worden in centimeters getekend. Vandaar ook regel 200 met $L = L * 50$ (1 centimeter = 50 stappen van de pen). Omdat eerst aan de hand van een tekening de algemene hoogteformule is afgeleid, kunt u op basis daarvan een universeel inzetbaar programma construeren en de lengten van de zijden vrijer kiezen. De hoogte kunt u ook vaststellen door te experimenteren. En wanneer de tekening nauwkeurig genoeg is, kunt u die ook met een passer bepalen. Maar dit soort methodes hebben niet het gewenste effect als de gebruiker de keuze van de zijden open wil

laten. Het is daarom altijd beter wat meer voorbereidingstijd in een concept voor de plotteroutines te steken en er algemene formules in te verwerken. Een gelijkbenige driehoek ontwerpen zal nu niet veel problemen meer geven. De hoogteformule kan ongewijzigd overgenomen worden. Alleen de berekeningen moeten iets worden uitgebreid, evenals de invoerroutine, omdat twee lengtegegevens variabel moeten zijn:

- 1) lengte van de basis
- 2) lengte van de twee zijden (gelijkbenig)

Als deze gegevens zijn ingevoerd, kan het middelpunt van de basis worden berekend. De hoogte kan dan als middelloodlijn worden berekend en de zijden kunnen worden getekend. De structuur van het programma ziet er zo uit:

algoritme

- 1) plotter instellen
- 2) basis en lengte van de benen bepalen
- 3) hoogte berekenen
- 4) driehoek tekenen
- 5) hoogte tekenen (middels een stippellijn)
- 6) einde

Als we dit algoritme (handelingsschema) omzetten in een programma, ziet het er als volgt uit:

```

100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 OPENS,6,5
130 PRINT#1,"M";200;-50
140 PRINT#1,"I"
150 REM===BASIS/BENEN=====
160 PRINT"MAX. 8 CM VOOR BASIS"
170 PRINT"MAX.15 CM VOOR BENEN"
180 INPUT"QQBASIS";A$
190 INPUT"QQBENEN ";B$
200 A=VAL(A$):A=ABS(A)
210 B=VAL(B$):B=ABS(B)
220 IFA>8ORB>15ORA<1THEN160
230 REM===HOOGTE BEREKENEN=====
240 A=A*50
250 B=B*50
260 H=SQR(B^2-(A/2)^2)
270 REM===DRIEHOEK TEKENEN=====
280 PRINT#1,"J";-A/2;0
290 PRINT#1,"J";0;H
300 PRINT#1,"J";A/2;0

```

```

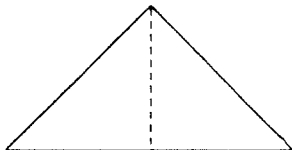
310 PRINT#1,"J";0;0
320 REM===HOOGTE TEKENEN=====
330 PRINT#5,5
340 PRINT#1,"J";0;H
350 REM===EINDE=====
360 PRINT#5,0
370 PRINT#1,"M";0;-100
380 CLOSE1
390 CLOSE5

```

Bij de analyse van het programma merken we het volgende op: het is u vermoedelijk al opgevallen dat bij de vraag naar de INPUT steeds met stringvariabelen wordt gewerkt, terwijl er alleen maar gegevens over de lengte en dus getallen ingelezen worden. Dat gebeurt om te voorkomen dat invoer van een letter in plaats van een getal tot REDO FROM START leidt. In eerste instantie worden alle gegevens geaccepteerd en in regel 220 worden ze geschild. Terwijl het programma naar het begin van de leesroutine (regel 160) terugspringt bij een 0 voor de basiszijde, worden beenlengten van 0 centimeter toegelaten. Op het papier ziet u dan alleen de basiszijde afgebeeld. En hoe zit het dan met de formule voor de hoogte? Net werd nog beweerd dat die ongewijzigd bleef en nu zien we $H = \text{SQR}(B^2 - (A/2)^2)$ in plaats van $H = \text{SQR}(3/4 * L^2)$. Deze spreken elkaar niet tegen. De eerste berekening werd - net als de tweede - uitgevoerd met de stelling van Pythagoras. Met dien verstande dat het eerste geval bijzonder is, omdat alle zijden van de driehoek even lang zijn. Misschien kunt u zich nog herinneren dat bij de afleiding van de formule in eerste instantie ook sprake was van $H = \text{SQR}(L^2 - (L/2)^2)$, voordat die omgevormd werd tot $H = \text{SQR}(3/4 * L^2)$. De rest van de routine bestaat uit de bekende instructies voor de Commodore 1520. Nu blijft alleen nog de rechthoekige driehoek over, waarbij we ons in eerste instantie beperken tot de rechte hoek op de basiszijde. Deze driehoeken zijn eenvoudig te construeren. Toch kunnen we hier nog geen programma geven voor iedere willekeurige driehoek met een rechte hoek. Daarvoor ontbreekt nog bepaalde kennis om het programma voldoende te kunnen begrijpen. Als we de cirkel hebben besproken, kunnen we een manier ontwikkelen om rechthoekige driehoeken in alle soorten en maten te ontwerpen. Nu gaan we over tot de drie manieren waarop we een rechthoekige driehoek kunnen tekenen:

1) U kunt aan het linker einde van de basislijn (punt A) van een driehoek een verticale lijn tekenen en het andere eind van die lijn (punt C) met het rechter einde van de basislijn (punt B) verbinden.

2) Deze methode kan ook uitgaan van punt B. De verticale lijn eindigt dan in C en A kan met C worden verbonden. Deze twee methoden hebben het nadeel dat de basiszijde niet in de X-richting ligt, maar overeenkomt met lijn BC of AC. De basiszijde ligt steeds tegenover de rechte hoek.



3) Om de basiszijde op zijn plaats te laten, kunnen we nog een andere methode toepassen.

De basis is hier (zoals het hoort) de lijn AB. Op punt C bedraagt de hoek precies 90 graden. Dat bereikt u door de hoeken op de punten A en B 45 graden te maken; de som van de hoeken is altijd 180 graden. Deze tekening is alleen mogelijk als de driehoek gelijkbenig is en de hoogte de helft is van de basiszijde. Met een cirkel wordt de constructie wat meer variabel gemaakt, omdat dan ook hoeken opgegeven kunnen worden. Daarover straks meer. Voor de eerste twee soorten driehoeken worden geen programma's aangeboden, daar ze geen formules bevatten en makkelijk te schrijven zijn. Het volgende programma geldt voor voorbeeld 3. De rechte hoek wordt getekend in C. U hoeft alleen maar de lengte van de basiszijde in te voeren.

```

100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 OPENS,6,5
130 PRINT#5,0
140 PRINT#1,"M";240;-50
150 PRINT#1,"I"
160 REM===INVOER=====
170 PRINT#5,BASIS MAX. 8 CM"
180 INPUT"QQBASIS ";A$
190 A=VAL(A$):A=ABS(A)
200 IFA>8THEN170
210 B=A*25
220 REM===DRIEHOEK TEKENEN=====
230 PRINT#1,"J";-B;0
240 PRINT#1,"J";0;B
250 PRINT#1,"J";B;0
260 PRINT#1,"J";0;0

```

```

270 REM===HOOGTE TEKENEN=====
280 PRINT#5,5
290 PRINT#1,"J";0;B
300 REM===EINDE=====
310 PRINT#5,0
320 PRINT#1,"M";0;-100
330 CLOSE1
340 CLOSE5

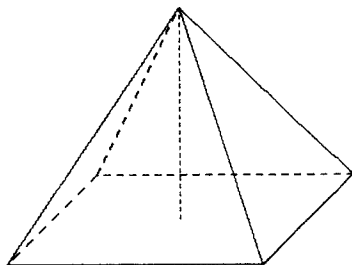
```

In regel 170 tot 210 gaat het om de lengte van de basiszijde in centimeters. Er volgt een controle op te grote of te kleine waarden en regel 210 zet de eenheid om in plotterformaat. Dan wordt de gebruikelijke tekenmethode toegepast. De driehoek wordt getekend en de hoogte gestippeld. De bovenste hoek is nu precies 90 graden. Voordat we ons aan cirkels en ellipsen wagen, behandelen we nog een ruimtelijke figuur: de pyramide. Deze bestaat uit een vierkant grondvlak. Op de zijden van het vierkant staan driehoeken die naar binnen nijgen zodat hun toppen de top van de pyramide vormen. Het volgende programma geeft de pyramide in perspectief weer:

```

100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 OPEN5,6,5
130 PRINT#1,"M";100;0
140 PRINT#1,"I"
150 REM===DATA LEZEN=====
160 FORX=0TO11
170 READX$,Y,Z
180 IFX=6THENPRINT#5,5
190 IFX=10THENPRINT#5,3
200 PRINT#1,X$,Y,Z
210 NEXT
220 REM===EINDE=====
230 PRINT#5,0
240 CLOSE1
250 CLOSE5
260 END
270 REM===ALLE DATA=====
280 DATAJ,200,0,J,270,70,J,135,200
290 DATAJ,200,0,R,135,200,J,0,0
300 DATAJ,70,70,J,270,70,R,70,70
310 DATAJ,135,200,J,135,35,M,0,-100

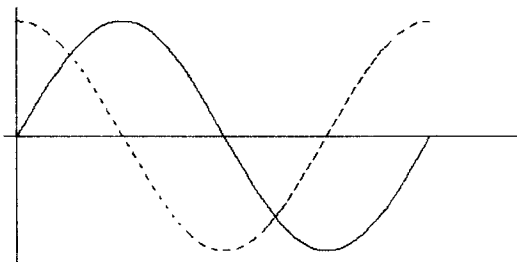
```



Omdat bij een ruimtelijke tekening de pen veel beweegt, is het zinvol DATA-regels te gebruiken. De regels 280 en 290 zijn verantwoordelijk voor het zichtbare gedeelte (ononderbroken lijnen). De regels 300 en 310 geven de onzichtbare hoogtelijn en de achtergrond weer als stippellijn. Ook bij de leesregels 160-200 is het typewerk tot een minimum beperkt, maar toch blijft de programmastructuur overzichtelijk. De regels 180 en 190 kiezen de gewenste lijnsoort.

8.3.5 Curven

We combineren beide tekenrichtingen zodat er een curve ontstaat. Hiervoor hebben we niet langer voldoende aan de functies en formules die we tot nu toe hebben leren kennen. Sinus en cosinus zijn onontbeerlijk geworden. Om uw geheugen wat op te frissen laten we eerst een tekening met beide functies zien. De ononderbroken lijn is de sinus en de stippellijn de cosinus.



Met de juiste combinatie van sinus en cosinus gaat er een nieuwe wereld van mogelijkheden voor u open, waartoe ook cirkels en ellipsen behoren. We beginnen met een cirkel en daarna geven we u een listing waarin de principes voor het maken van een cirkel staan.

```
100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 OPEN5,6,5
130 PRINT#1,"M";240;0
140 PRINT#1,"I"
150 REM===CIRKEL TEKENEN=====
160 FORX=-1.6TO4.8STEP.04
170 Y=150*COS(X)
180 Z=150*SIN(X)
190 IFX=-1.6THENPRINT#1,"R";Y;Z
200 PRINT#1,"J";Y;Z
210 NEXT
220 REM===STRAAL=====
230 PRINT#5,5
240 PRINT#1,"J";0;0
250 REM===EINDE=====
260 PRINT#5,0
270 PRINT#1,"M";0;-200
280 CLOSE1
290 CLOSE5
```

We beginnen onze uitleg met de regels 160 tot 210, omdat we aannemen dat het openen van de Commodore 1520 geen problemen meer oplevert.

160 het bereik van de getallen voor de sinus- en cosinusfunctie en de stapgrootte. In ons voorbeeld vindt u ... STEP .04, dus 160 stappen. De stapgrootte bepaalt de oplossing van de karakters. Al te grote waarden kunt u beter niet gebruiken, omdat dat het totaalbeeld niet ten goede komt. En bij te kleine getallen duurt het allemaal erg lang. Omdat we bij -1.6 zijn begonnen, is het beginpunt van de tekening met een vierde van de cirkel opgeschoven. Als u FOR X=0 TO 6.4 invoert, krijgt u natuurlijk hetzelfde resultaat, alleen ligt het beginpunt dan anders.

170 bevat variabele Y, de cosinus van de X-waarde. De berekende cosinus wordt met een constante (in het voorbeeld 150) vermenigvuldigd. Deze constante bepaalt de straal (radius) voor de X-richting. Een deel van de cirkel wordt met deze functie gemaakt.

- 180 bevat variabele Z, het sinus-aandeel van de cirkel. De berekende sinus-waarde wordt vermenigvuldigd met een constante factor, die de straal voor de Y-richting bepaalt. Omdat we uiteindelijk een cirkel willen zien, moet de straal voor beide richtingen gelijk zijn.
- 190 Het middelpunt van de cirkel is 240/0 (regel 130). Voordat de plotter de cirkel tekent, moet de pen de straal boven het papier afleggen. Dat gebeurt meteen nadat de eerste waarde (-1,6) in de functies is gesubstitueerd. Als u in regel 160 bij 0 begint, staat dus in regel 190 ook een 0.
- 200 tekent het gedeelte van de curve tot aan het berekende punt. In regel 210 begint het hele proces opnieuw.

In de handleiding van de VC-1520 is wel een soortgelijke routine opgenomen, maar zonder uitleg. Met deze analyse wordt alles een stuk makkelijker. Het volgende programma verduidelijkt de cirkelfunctie:

```

100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 OPEN5,6,5
130 PRINT#1,"M";240;0
140 PRINT#1,"I"
150 REM===INUOER=====
160 PRINT"sSTRAAL MAX 4 CM"
170 PRINT"QOPLOSSING .01 - .1"
180 INPUT"QSTRAAL";A$
190 INPUT"QOPLOSSING";B$
200 A=VAL(A$):A=ABS(A)
210 B=VAL(B$):B=ABS(B)
220 IFA>4THEN160
230 IFB>.1ORB<.01THEN160
240 A=A*50
250 REM===CIRKEL TEKENEN=====
260 FORX=-1.6TO4.8STEPB
270 Y=A*COS(X)
280 Z=A*SIN(X)
290 IFX=-1.6THENPRINT#1,"R";Y;Z
300 PRINT#1,"J";Y;Z
310 NEXT
320 REM===STRAAL=====
330 PRINT#5,5
340 PRINT#1,"J";0;0
350 REM===EINDE=====
360 PRINT#5,0
370 PRINT#1,"M";0;-200

```

```

380 CLOSE1
390 CLOSE5

```

De straal en de resolutie bepalen de tekensnelheid. Er zijn twee stralen, 1 parallel aan de X-as en 1 parallel aan de Y-as. We kunnen dus ook een ellips maken, een langgerekte cirkel met stralen van verschillende lengte. U kunt de laatste listing door toevoeging van een paar regels omwerken voor ellipsen. Als u een cirkel wilt, kan dat ook, als de stralen maar dezelfde lengte hebben.

```

100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 OPENS,6,5
130 PRINT#1,"M";240;0
140 PRINT#1,"I"
150 REM===INVOER=====
160 PRINT"STRAAL MAX 4 CM ←"
170 PRINT"QSTRAAL MAX 8 CM ↑"
180 PRINT"QOPLOSSING .01 - .1"
190 INPUT"QSTRAAL ←";A$
200 INPUT"QSTRAAL ↑";A1$
210 INPUT"QOPLOSSING";B$
220 A=VAL(A$):A=ABS(A)
230 REM==CONTROLE OP FOUTE INVOER=====
240 A1=VAL(A1$):A1=ABS(A1)
250 B=VAL(B$):B=ABS(B)
260 IFA>4THEN160
270 IFA1>8THEN160
280 IFB>.1ORB<.01THEN160
290 A=A*50
300 A1=A1*50
310 REM===ELLIPS TEKENEN=====
320 FORX=-1.6TO4.8STEPB
330 Y=A*COS(X)
340 Z=A1*SIN(X)
350 IFX=-1.6THENPRINT#1,"R";Y;Z
360 PRINT#1,"J";Y;Z
370 NEXT
380 REM===STRAAL=====
390 PRINT#5,5
400 PRINT#1,"R";0;0
410 PRINT#1,"J";A;0
420 PRINT#1,"R";0;0
430 PRINT#1,"J";0;A1
440 REM===EINDE=====
450 PRINT#5,0
460 PRINT#1,"M";0;-200

```

470 CLOSE1
480 CLOSE5

Als er bij dit programma problemen opdoemen, raden we u aan de theorie over de constructie van een cirkel nog eens door te lezen. De volgende ruimtelijke figuren die we aan u voorstellen, zijn de cylinder en de kegel. Hierbij combineren we weer rechte lijnen en curves. Bij de constructie van een cylinder hebben we twee ellipsen nodig. Deze verbindt u met twee lijnen tot een totale figuur. Het startpunt voor de onderste ellips moet worden verlegd, want een helft wordt met ononderbroken lijn getekend en de andere helft met stippellijn. Dit om voor- en achtergrond te scheiden. Om dit te bereiken, heeft de FOR-NEXT-lus in het programma andere waarden gekregen. Het bereik van -3.2 tot 0 wordt nu gebruikt en zo verschijnt precies de onderste helft van de ellips op het papier. U leest het positieve bereik van 0 tot 3.2 in, nadat u met PRINT# 5,X de lijnsoort heeft veranderd.

```
100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 OPEN5,6,5
130 PRINT#1,"M";240;0
140 PRINT#1,"I"
150 REM===ELLIPSEN TEKENEN=====
160 FORX=-3.2TO3.2STEP.04
170 Y=120*COS(X)
180 Z=30*SIN(X)
190 IFX=-3.2THENPRINT#1,"R",Y,Z
200 IFBN=1THENV=V+1
210 IFV=80THENPRINT#5,5
220 PRINT#1,"J";Y;Z
230 NEXT
240 IFBN=1THEN320
250 REM===ELLIPS 2 VOORBEREIDEN=====
260 PRINT#1,"J";-120;-300
270 PRINT#1,"R";0;-300
280 PRINT#1,"I"
290 BN=1
300 GOTO160
310 REM===ZIJDE/HOOGTE=====
320 PRINT#1,"R";120;0
330 PRINT#5,0
340 PRINT#1,"J";120;300
350 PRINT#1,"R";0;300
360 PRINT#5,5
370 PRINT#1,"J";0;0
380 REM===EINDE=====
390 PRINT#5,0
```

```

400 PRINT#1,"R";-240;-100
410 CLOSE1
420 CLOSE5

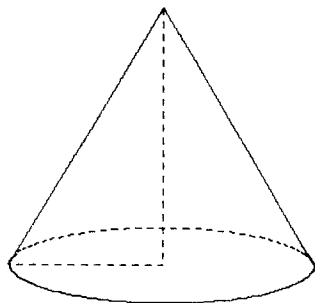
```

Het programma begint de cylinder met de bovenste ellips te tekenen vanaf regel 160. De factor voor de straal in de X-richting is 120, voor de Y-richting 30. Als u wijzigingen in het programma wilt aanbrengen, moet u daar rekening mee houden omdat deze waarden ook de plaats van de zijden bepalen. Van regel 260 tot 300 wordt ellips nummer 2 voorbereid. De eerste zijkant wordt op papier gezet en er wordt een nieuw relatief nulpunt gedefinieerd. In 300 wordt teruggesprongen naar de tekenroutine voor de ellipsen, waarbij regel 210 ditmaal de stippellijn kiest zodra de onderste helft van de ellips is gepasseerd. Dan ontbreken nog de tweede zijde en de gestippelde hoogte. Daar houden regel 320 tot 370 zich mee bezig. Bij een kogel ligt de zaak iets eenvoudiger, omdat dan slechts 1 ellips nodig is om een ruimtelijke afbeelding te creëren.

```

100 REM====1520 INSTELLEN=====
110 OPEN1,6,1
120 OPEN5,6,5
130 PRINT#1,"M";240;0
140 PRINT#1,"I"
150 REM====ELLIPSEN TEKENEN=====
160 FORX=-3.2TO3.2STEP.04
170 Y=120*COS(X)
180 Z=30*SIN(X)
190 IFX=-3.2THENPRINT#1,"R",Y,Z
200 U=U+1
210 IFU=80THENPRINT#5,5
220 PRINT#1,"J";Y;Z
230 NEXT
240 REM====KEGEL TEKENEN=====
250 PRINT#1,"J";0;0
260 PRINT#1,"J";0;200
270 PRINT#5,0
280 PRINT#1,"J";-120;0
290 PRINT#1,"R";120;0
300 PRINT#1,"J";0;200
310 REM====EINDE=====
320 PRINT#1,"M";0;-100
330 CLOSE1
340 CLOSE5

```



Het ellipsgedeelte (regel 160-230) komt grotendeels overeen met de routine van de cylinder. De rest hebben we bij de driehoeken behandeld.

8.3.6 Driehoeken 2

Ter afsluiting van onze uitleg over geometrische figuren bespreken we nu de rechthoekige driehoek (met een rechte hoek tegenover de basiszijde) met verschillende beenlengten. We doen dat nu pas omdat we de cirkel nodig hebben. Het vraagstuk van de driehoek met ongelijke benen wordt met de cirkel van Thales opgelost. De belangrijkste kenmerken van de cirkel van Thales zijn: als men om het middelpunt van een lijn met de lengte 1 een halve cirkel trekt met een straal $1/2$, ontstaat er een rechthoekige driehoek als men het beginpunt van de lijn (punt A) verbindt met een willekeurig punt op de halve cirkel en vanuit dat punt een lijn trekt naar het eindpunt van de lijn. Omdat deze theorie een beetje ingewikkeld klinkt, ontwikkelen we een programma waarmee u elke rechthoekige driehoek kunt tekenen waarvan de hoek tegenover de basiszijde AB ligt. De halve cirkel is ook in dit programma opgenomen.

algoritme

- 1) oorsprong definiëren
- 2) beginnen met de halve cirkel
- 3) driehoek tekenen (beginnen bij de oorsprong)
- 4) cirkel afmaken
- 5) basiszijde tekenen

Dit levert het volgende programma op:

```
100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 OPENS,6,5
130 PRINT#1,"M";240;0
140 PRINT#1,"I":E=50
150 REM===INGESCHREVEN CIRKEL=====
160 FORX=0TO3.2STEP.04
170 Y=120*COS(X)
180 Z=120*SIN(X)
190 IFX=0THENPRINT#1,"R",Y,Z
200 V=V+1
210 IFV=ETHENPRINT#1,"J";120;0
220 IFV=ETHENPRINT#1,"R";Y-.04;Z-.04
230 IFV=ETHENPRINT#1,"J";-120;0
240 PRINT#1,"J";Y;Z
250 NEXT
260 REM===BASIS=====
270 PRINT#1,"R";120;0
280 PRINT#1,"J";-120;0
290 REM===EINDE=====
300 PRINT#1,"M";0;-100
310 CLOSE1
320 CLOSE5
330 CLOSE1
340 CLOSE5
```

In regel 140 wordt met E=50 een punt gedefinieerd waarop het tekenen van de halve cirkel wordt onderbroken en de driehoek moet worden getekend. Dit getal lijkt op het eerste gezicht nietszeggend, maar wordt duidelijk wanneer u de FOR-NEXT-lus voor de halve cirkel bekijkt. Daarin wordt de cirkel in 80 afzonderlijke stappen verdeeld ($3,2/0,04=80$). Regel 200 verhoogt de variabele V bij elke tekenstap met 1, zodat de driehoek bij V=E=50 getekend kan worden. Als het punt van oorsprong bereikt is (regels 210-230), worden de hoekpunten van de basiszijde AB aangestuurd. In C, het punt van oorsprong, ontstaat automatisch een rechte hoek. De basiszijde levert geen problemen meer op en de tekening is compleet. Probeer u maar eens de waarde voor E tussen 1 en 89 te variëren; u zult zien dat het resultaat steeds een rechthoekige driehoek is. De geometrische figuren die we tot nu toe besproken hebben, dienen als voorbeelden om de programmastructuur en de oplossing van vraagstukken toe te lichten. Als u vindt dat er een figuur in deze serie ontbreekt,

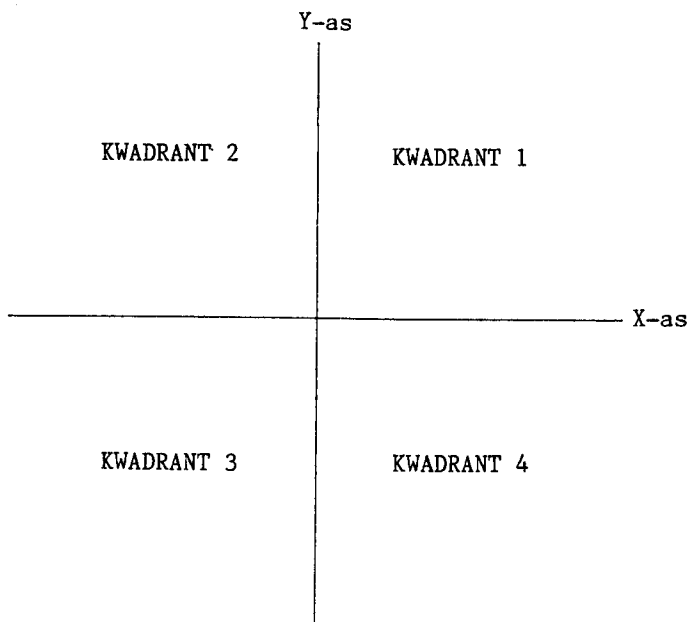
probeert u dan die figuur met een eigen programma te construeren. Om u wat ideeën te geven:

- a) prisma
- b) afgeknotte pyramide of kegel
- c) kogel (ruimtelijk)
- d) ellipsoïde

Deze figuren kunt u met de opgedane kennis zonder meer maken. Het vergt enig denkwerk, maar plotter-technisch gesproken is het best te doen.

8.4 Functies plotten

Aan het begrip functiegrafiek verbindt men gewoonlijk de sinus, de cosinus en de x-functies. Omdat vooral x-functies zeer vaak voorkomen (het curvenprobleem), staan in dit hoofdstuk de x-grafieken centraal. Interessant is hier de ruimte van X tot X , omdat wiskundige vraagstellingen vaak in dit bereik liggen. Ook de bekende parabool X ligt in dit gebied. We zullen in het kort iets over de tekenwijze van de functies zeggen, om misverstanden bij de beschrijvingen te voorkomen. De grafiek wordt (zoals bij functies gebruikelijk is) in een coördinatenstelsel getekend. Dit coördinatenstelsel bevat een negatieve en een positieve X -respectievelijk Y -as. Het coördinatenstelsel wordt vaak in kwadranten verdeeld, in totaal 4 stuks. De kwadranten worden tegen de wijzers van de klok in genummers, te beginnen met het kwadrant boven het positieve gedeelte van de X -as.



Bij de verdere uitleg gaan we uit van dit systeem. We stellen nu een paar varianten voor, zodat u voor elke tekening de passende versie kunt kiezen. U hoeft niet altijd het hele stelsel te tekenen. Als u bijvoorbeeld een parabool tekent, hoeft u de ruimte onder de X-as niet te gebruiken zolang het om een normale parabool gaat of een parabool die naar de positieve Y-as is verschoven. Een sinus- of een cosinuscurve beweegt zich haast altijd in de richting van de positieve X-as en aan beide zijden van de Y-as. Het negatieve deel van de X-as kunt u dus weglaten. De functies waarbij het positieve of negatieve gedeelte van de X-as gelijk wordt gesteld aan het positieve of negatieve gedeelte van de Y-as leveren een rechte lijn op. Deze doorsnijdt de oorsprong onder een hoek van 45 graden ten opzichte van de assen. Bij $Y=X$ ligt de rechte lijn in het eerste kwadrant en loopt na het middelpunt doorsneden te hebben door naar het derde kwadrant. Bij $Y=-X$ loopt de rechte lijn van het tweede naar het vierde kwadrant. Dit moet door een programma evengoed verwerkt kunnen worden als het feit, dat de rechte lijn op de Y-as verschoven kan worden en om de oorsprong kan draaien. Als we de genoemde eisen samenvoegen, krijgen we dit programma:

```

100 REM===1520 INSTELLEN=====
110 OPEN4,6
120 OPEN1,6,1
130 OPEN2,6,2
140 REM===COORDINATENSYSTEEM=====
150 FORA=1TO6
160 READX$,Y,Z
170 PRINT#1,X$;Y;Z
180 NEXT
190 DATAM,0,0,D,480,0,M,240,240
200 DATAD,240,-240,M,240,0,I,0,0
210 REM===DEFINITIE VAN DE FUNCTIE===
220 DEFFNA(X)=X
230 REM===TEKENEN=====
240 FORA=200TO-200STEP-1
250 IFA=200THENPRINT#1,"R";A;FNA(A)
260 PRINT#1,"J";A;FNA(A)
270 NEXT
280 PRINT#1,"M";0;0
290 REM===EINDE=====
300 CLOSE1
310 CLOSE2
320 CLOSE4

```

Een analyse van deze kleine routine is wel op zijn plaats, omdat ook x^n -functies daarop gebaseerd zijn.

150-180 De gegevens voor het coördinatenkruis worden ingelezen en getekend.

190-200 Gegevens voor het coördinatenstelsel.

220 De Commodore 64 kent de mogelijkheid functies te definiëren (DEF), om de programmeur werk te besparen. In het verdere verloop van het programma kunt u met PRINT FNA (GETAL) enzovoort naar de tevoren gedefinieerde functie verwijzen. Het gewenste getal wordt ingevoerd en u heeft meteen het benodigde resultaat, zonder elke keer de hoofdfunctie te hoeven invoeren. Deze routine maakt ook gebruik van die techniek, omdat de overzichtelijkheid op die manier bewaard blijft. De computerinstructie DEFFNA (X) = X komt overeen met de vergelijking $Y=X$. Daarmee is de rechte lijn door de oorsprong gedefinieerd. Als u de later te construeren rechte lijn op de Y-as wilt verschuiven, moet u er voor een verschuiving naar boven een bepaalde waarde bij optellen. Bij een verschuiving naar beneden moet u er een bepaalde waarde van aftrekken (50 = 1 cm). Op die manier kunt u bijvoorbeeld heel

gemakkelijk parallelle rechte lijnen tekenen.

240 Nu moeten we voor elke functie een individueel bereik zien te vinden, waarbinnen getallen in de functie kunnen worden ingevoerd. Bij de Commodore 1520 is de maat voor alle dingen de lengte van de X-as, namelijk 480 tekenstappen. Voor een rechte lijn door de oorsprong is een bereik van -200 tot 200 geschikt omdat de X-waarde met de Y-waarde overeenstemt. De X-waarden schommelen dus ook tussen 200 en -200, wat binnen het toegestane bereik ligt. Stapgrootte 1 levert een keurig plaatje op.

260 Hier worden de waarden getekend. De Y-waarde wordt berekend door de X-waarde in de functie te substitueren ($Y=X$).

Ook kunnen we voldoen aan de laatste eis, namelijk een variabele stijging. Daarvoor moeten we alleen een paar kleine wijzigingen aanbrengen. In regel 240 worden 200 en -200 vervangen door 100 en -100. Voor regel 250 geldt dan $FORA=100....$ De A in de regels 250/260 wordt nu vermenigvuldigd met de factor 2. Voor 260 krijgt u dus:

260 PRINT#1, "J"; 2xA; FNA (A)

Voor 250 geldt iets dergelijks. Als u dit programma start, is de stijging van de rechte lijn zichtbaar kleiner geworden. Met het programma kunt u voldoen aan alle eisen van een eerstegraadsfunctie $y=mx+b$. De parameter m geeft de stijging aan (richtingscoëfficiënt) en kan naar wens worden gevarieerd; b is de afstand van het nulpunt (de oorsprong 0/0) tot aan het snijpunt van de rechte lijn met de Y-as. Ook de eis om de eerstegraadsfunctie op de Y-as te kunnen verschuiven, is in het programma verwerkt. Het programma voor de eerstegraadsfunctie dient in de basisvorm als uitgangspunt voor het plotten van x^n bij $n>1$. Als u het programma voor de eerstegraadsfunctie hebt ingetypt, laat het dan staan; het komt straks nog van pas. Het programma kan namelijk worden omgewerkt voor functiegrafieken in het bereik van x^2 tot x^5 . Aan de oplossing mag niet worden getornd. We willen de kwaliteit behouden. De verandering moet dus op een andere plek in het programma plaatsvinden. De meest gunstige oplossing is deze: de met FNA (A) berekende waarde wordt met een constante vermenigvuldigd, waardoor de verhoogde exponent van de functie zodanig wordt verminderd dat de openingen van de beginpunten van de tekeningen bijna identiek zijn. Om tijd te besparen, wijden we ons aan een theoretische beschouwing: we vergelijken de functies van x tot x^4 met elkaar. Stel, u hebt de functie $Y=X$. Als we voor de X in gedachten 100 invullen, blijft het resultaat 100. Dit feit gebruiken we als uitgangspunt voor de

verdere beschouwingen. Als we in de zojuist genoemde functie waarden substitueren met als maximum 100, tekent de plotter alleen tot 100 als een stap gelijk aan 1 is. De tekening ligt bij een juiste keuze van de oorsprong (bijvoorbeeld in het midden van het papier) in het toegestane bereik in de X-richting (0-480). Als de exponent met 1 wordt verhoogd, krijgt u de functie $y=x^2$. Als u nu weer waarden tot een maximum van 100 invult, ligt het resultaat voor Y al bij $100^2=10.000$. Het resultaat is bij een verhoging van de exponent met 1 dus 100 keer groter. De functies $y=x^3$ en $y=x^4$ bevestigen dit. Rekent u maar na. Om de proportionele factor ook voor reële exponenten gemakkelijker te kunnen bepalen, geven we straks een algemene formule. We gaan eerst verder met de constante factor. De waarde die in de functie wordt gesubstitueerd, mag niet groter dan 100 zijn. Maar het berekende getal voor Y stijgt bij elke verhoging van de exponent van X. Daarom moet het met FNA (A) berekende getal zodanig met een factor worden vermenigvuldigd, dat de uitkomst voor Y altijd gelijk is - welke exponent X ook heeft. Bij de parabool x^2 wordt de uitkomst voor Y steeds met .01 vermenigvuldigd. De maximale waarde is dan weer 100. Reken maar na: $100 \times 100 \times 0.01 = 100$. Voor x^3 ligt de proportionele factor bij 0.0001. Om fouten te voorkomen bij de zoveel nullen, volgt hier een berekeningsformule voor de constante C. Bij $x^n=Y$ kiest u voor n een willekeurig getal. Naast de natuurlijke getallen kunnen zelfs reële getallen worden gebruikt. Als u een exponent hebt gekozen, geldt de volgende formule voor de constante waarmee FNA (A) is vermenigvuldigd:

$$c=1/100^{(n-1)}$$

Een voorbeeld. U wilt de functie x^6 in een grafiek afbeelden. De exponent n is 6. Dit getal wordt in bovenstaande formule ingevuld:

$$c=1/100^{(6-1)}$$

en de uitkomst is 10^{-10} . En daarmee hebt u de waarde die aan de functie $y=x^6$ verbonden moet worden om in het bereik tot 100 voor de waarde X een tekening te krijgen. In de volgende drie routines wordt deze theorie in praktijk gebracht. Het gaat om de afbeelding van de functies x^2 , x^3 en x^4 . Bekijk u de listings maar eens om de constructie te begrijpen. Na de programma's volgt de uitleg.

```
100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 REM===COORDINATENSYSTEEM=====
130 FCRA=1T06
140 READX$,Y,Z
```

```

150 PRINT#1,X$;Y;Z
160 NEXT
170 DATAM,0,0,D,480,0,M,240,240
180 DATAD,240,-20,M,240,0,I,0,0
190 REM===DEFINITIE VAN DE FUNCTIE=====
200 DEFFNA(X)=X↑2
210 C=.02
220 REM===TEKENEN=====
230 FORA=100TO-100STEP-1
240 IFA=100THENPRINT#1,"R";A;C*FNA(A)
250 PRINT#1,"J";A;C*FNA(A)
260 NEXT
270 PRINT#1,"M";0;0
280 REM===EINDE=====
290 CLOSE1

```

```

100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 REM===COORDINATENSYSTEEM=====
130 FORA=1TO6
140 READX$,Y,Z
150 PRINT#1,X$;Y;Z
160 NEXT
170 DATAM,0,0,D,480,0,M,240,240
180 DATAD,240,-240,M,240,0,I,0,0
190 REM===DEFINITIE VAN DE FUNCTIE=====
200 DEFFNA(X)=X↑3
210 C=.0002
220 REM===TEKENEN=====
230 FORA=100TO-100STEP-1
240 IFA=100THENPRINT#1,"R";A;C*FNA(A)
250 PRINT#1,"J";A;C*FNA(A)
260 NEXT
270 PRINT#1,"M";0;0
280 REM===EINDE=====
290 CLOSE1

```

```

100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 REM===COORDINATENSYSTEEM=====
130 FORA=1TO6
140 READX$,Y,Z
150 PRINT#1,X$;Y;Z
160 NEXT
170 DATAM,0,0,D,480,0,M,240,240

```

```

180 DATAD,240,-20,M,240,0,I,0,0
190 REM===DEFINITIE VAN DE FUNCTIE===
200 DEFFNA(X)=X↑4
210 C=.000002
220 REM===TEKENEN=====
230 FORA=100TO-100STEP-1
240 IFA=100THENPRINT#1,"R";A;C*FNA(A)
250 PRINT#1,"J";A;C*FNA(A)
260 NEXT
270 PRINT#1,"M";0;0
280 REM===EINDE=====
290 CLOSE1

```

Als u deze drie programma's met elkaar hebt vergeleken, is het u vast opgevallen dat de listings vrijwel identiek zijn. De verschillen zitten in drie regels, te weten regel 180, 200 en 210.

regel 180 In deze DATA-regel zijn delen van de tekeninstructies voor het coördinatenstelsel opgenomen. Bij de functie x^2 hebben we alleen het bovenste gedeelte van het assenstelsel nodig, omdat de parabool x^2 niet langs de Y-as is verschoven. Er staat:

```
180 DATAD, 240, -20,...
```

Bij de afbeelding van x^3 moet u de kwadranten onder en boven de X-as gebruiken, vanwege de oneven exponent. Grafiek x^4 ligt daarentegen weer in kwadrant 1 en 2. In de listing van x^3 begint regel 180 daarom zo:

```
180 DATAD, 240, -240,...
```

Het negatieve getal geeft in de DATA-regel de lengte van de negatieve Y-as aan.

regel 200 Deze regel definieert de functie die bewerkt moet worden. In regel 210 gaat het om de daarbij behorende proportionele factor. Deze factor wordt met bovenstaande formule berekend en daarna met 2 vermenigvuldigd, omdat het waardenbereik nog maar tussen 100 en -100 ligt.

functie	factor
x^2	0.02
x^3	0.0002
x^4	0.000002

Het programma kan dus zeer snel aangepast worden aan andere functies. Om de voor C ontwikkelde formule in het programma toe te passen, introduceren we een routine die in het bereik van natuurlijke getallen No van 0-11 werkt. Deze routine substitueert de door u gekozen exponent in de formule en vervaardigt ook de

grafiek van de functie. Eerst wordt een assenstelsel getekend. Vervolgens wordt naar de exponent gevraagd, C berekend en de grafiek kan op papier gezet worden.

```

100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 REM===COORDINATENSYSTEEM=====
130 FORA=1TO6
140 READX$,Y,Z
150 PRINT#1,X$;Y;Z
160 NEXT
170 DATAM,0,0,D,480,0,M,240,240
180 DATAD,240,-240,M,240,0,I,0,0
190 REM===DEFINITIE VAN DE FUNCTIE===
200 INPUT"§WELKE EXPONENT (N E N)";N$
210 N=VAL(N$):N=ABS(N)
220 IFN>11THEN200
230 FORP=NT00STEP-1
240 IFP=0THEN270
250 NEXT
260 GOTO200
270 DEFFNA(X)=X↑N
280 C=2/100↑(N-1)
290 REM===TEKENEN=====
300 FORA=100TO-100STEP-1
310 IFA=100THENPRINT#1,"R";A;C*FNA(A)
320 PRINT#1,"J";A;C*FNA(A)
330 NEXT
340 PRINT#1,"M";0;0
350 REM===EINDE=====
360 CLOSE1

```

De invoerroutine ligt in dit programma van regel 200 tot 250. Regel 220 test de invoer op overflow, zodat die foutmelding wordt voorkomen. De regels 230 tot 250 beperken het bereik van de getallen tot No, dat is het bereik van de natuurlijke getallen inclusief de nul. Als de invoer correct is, wordt met x^n de functie gedefinieerd in regel 270. De formule voor de constante wordt in regel 280 toegepast, alleen is de factor gelijk ingevoerd omdat het waardenbereik van de in te vullen getallen (regel 300) zich tussen 100 en -100 bevindt. Als u van de 2 in regel 280 een 1 maakt, wordt de functie te kort en te breed. De tekenroutine is hetzelfde als bij voorgaande programma's. Alleen de regelnummers zijn veranderd. Voordat we met de afbeelding van de sinus- en cosinuscurve starten, brengen we het curvenprobleem nog even onder uw aandacht. Er bestaan namelijk twee methodes om het tekenwerk wat gemakkelijker te maken. Hierbij moet u wel uitgaan van kleine stapgroottes, want de geringe breedte van het papier laat niet toe

dat u met centimeters werkt. We kunnen u hier niet het hele programma voor het curvenvraagstuk tonen, omdat dit programma 61 blokken omvat en de listing dus erg lang zou worden. We laten het gedeelte van de ingewikkelde berekeningen gewoon zitten en beperken ons tot de tekening. We moeten rekening houden met de volgende zaken. Bij het curvenvraagstuk is het gebruikelijk de nulpunten, de minimumwaarden en de maximumwaarden uit de basisfunctie af te leiden. Bij het voorbeeldprogramma bepalen we ons tot gehele getallen en gaan we door tot $y=x^5$. We geven in het kort de regel voor de afleiding nog eens weer: de variabele met een bepaalde exponent (bijvoorbeeld x^4) wordt als volgt afgeleid: de variabele wordt met de exponent vermenigvuldigd en de oude exponent wordt met 1 verminderd ($4*x^3$). Algemeen geldt: van x^n wordt afgeleid $n*x^{(n-1)}$. Voor de tweede afleiding gelden soortgelijke regels. U kunt de hele werkwijze ook omdraaien. Als u bijvoorbeeld een afgeleide functie hebt, kunt u de basisfunctie bepalen. Een voorbeeld: de eerste afgeleide van een functie is bekend: $3*x^2$. U kunt nu ook de tweede afgeleide bepalen, en u hebt de tekeningen van de eerste en tweede afgeleide. Maar het belangrijkste, de basisfunctie, ontbreekt. De basisfunctie van een afgeleide functie krijgt u door de exponent te verhogen met 1 en de uitkomst te delen door de nieuwe exponent. Dit moet u zo vaak uitvoeren als de functie werd afgeleid. Verhoging van de exponent met 1 resulteert in $3*x^{(2+1)}$. Het resultaat delen door de nieuwe exponent levert als uitkomst $(3*x^{(2+1)})/(2+1)$ op. Samengevat krijgt u x^3 ; dit is dus de basisfunctie van de gegeven afleiding $3*x^2$. U kunt dit controleren door x^3 af te leiden.

Als u alleen maar de derde afgeleide kent, moet u drie keer terugrekenen. Nu volgt het programma.

```

100 REM===INVOER=====
110 INPUT "WELKE FUNCTIE?";X$
120 REM===FUNCTIE INLEZEN=====
130 FORX=1TOLEN(X$)
140 Y$=MID$(X$,X,1)
150 Y=VAL(Y$)
160 IFY$="" THENNEXT
170 IFY$="X" THENE=E+1
180 IFY$="↑" THENNEXT
190 IFE=1ANDY>0 THENA=Y
200 IFE<1ANDY>0 THENB=Y
210 IFY$="-" THEN110
220 NEXT
230 REM===AFLEIDEN=====
240 PRINT "Q1.1AFLEIDING";A*B;"*X↑";A-1
250 C=A*B*(A-1)
260 PRINT "Q2.1AFLEIDING";C;"*X↑";A-2

```

Dit korte programma is als voorbeeld bedoeld. Het werkt alleen met kleine functies als $9 \times x^7$. De eerste en de tweede afgeleide worden dan uitgevoerd. Deze werkwijze is naar wens uit te breiden, zodat u met een aangepast programma ook meer omvangrijke functies kunt bewerken. Een dergelijk programma is te specialistisch en te lang om hier te tonen.

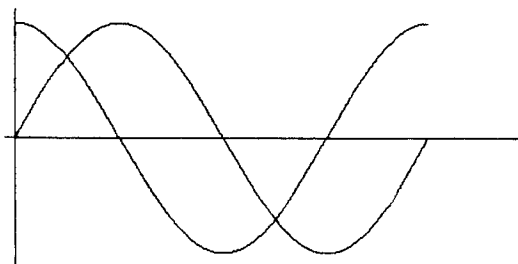
Als laatste wiskundige functies willen we de belangrijke trigonometrische functies sinus en cosinus bespreken. Omdat deze functies al aanwezig zijn in de Commodore 64 kunnen we heel snel een programma ontwikkelen.

```

100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 OPEN2,6,2
130 PRINT#2,0
140 REM===COORDINATENSYSTEEM=====
150 PRINT#1,"D";400;0
160 PRINT#1,"M";10;100
170 PRINT#1,"D";10;-100
180 PRINT#1,"M";10;0
190 PRINT#1,"I"
200 REM===FUNCTIE=====
210 DEFFNA(X)=SIN(X)
220 REM===PUNT BEREKENEN/PLOTTEN=====
230 FORX=0TO320
240 PRINT#1,"J";X;FNA(X/51)*90
250 NEXT
260 REM===EINDE=====
270 PRINT#1,"R";0;0
280 CLOSE1
290 CLOSE2

```

In regel 210 wordt de betreffende functie (hier sinus) gedefinieerd. Zonder speciale maatregelen te treffen, kunnen we ook de cosinus invullen. Met een vergrotingsfactor (regel 240) kan de curve naar boven en in de breedte worden uitgebreid. Als u sinus en cosinus na elkaar invult en dan het programma laat lopen, ontstaat de volgende afbeelding:



De sinus start in de oorsprong. Bij de cosinus heeft een faseverschuiving van 90 graden plaatsgevonden. De top van de cosinus ligt bij 0 en daarna daalt de curve weer. Als u het ingesloten vlak van beide functies wilt inkleuren of arceren, kunt u de volgende routine gebruiken:

```

100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 OPEN2,6,2
130 PRINT#2,0
140 REM===COORDINATENSYSTEEM=====
150 PRINT#1,"D";400;0
160 PRINT#1,"M";10;100
170 PRINT#1,"D";10;-100
180 PRINT#1,"M";10;0
190 PRINT#1,"I"
200 REM===FUNCTIE=====
210 DEFFNA(X)=SIN(X)
220 DEFFNB(X)=COS(X)
230 REM===PUNT BEREKENEN/PLOTTEN=====
240 FORX=0TO320
250 PRINT#1,"J";X;FNA(X/51)*90
260 PRINT#1,"J";X;FNB(X/51)*90
270 NEXT
280 REM===EINDE=====
290 PRINT#1,"R";0;0
300 CLOSE1
310 CLOSE2

```

In regel 210 en 220 worden sinus en cosinus gedefinieerd. Vervolgens wordt afwisselend een punt op de sinus- en op de cosinuslijn aangesproken. De punten worden met elkaar verbonden. Deze routine kleurt het vlak helemaal in. Als u het vlak

gearceerd wilt hebben, hoeft u alleen in regel 240 een STEP in te voeren (bijvoorbeeld FORX = 0 TO STEP 4).

8.5 Bijzondere werkzaamheden met de Commodore 1520

Voordat we de statistische functies aan u voorstellen, volgt hier een korte inleiding over een paar bijzondere werkzaamheden, die u ook met een plotter kunt verrichten. De print-using is namelijk niet alleen voorbehouden aan een printer of beeldscherm; ook de Commodore 1520 kan met een klein programma bijvoorbeeld getallen of tekst netjes in een kolom plotten. Het voorbeeldprogramma schrijft telkens twee getallen in een rij en wel zo dat hun decimale punten onder elkaar komen te staan. Daarna begint een nieuwe regel.

```
100 REM===1520 INSTELLEN=====
110 OPEN4,6
120 PRINT#4
130 REM==LENGTE VAN HET GETAL BEPALEN==
140 GG=4
150 DG=3
160 A=0
170 REM===INVOER/VOORBEREIDING=====
180 INPUT"GETAL";NR
190 A=A+1
200 NR$=STR$(NR)
210 NN=LEN(NR$):W=0
220 FORX=1 TONN
230 IF MID$(NR$,X,1)=". " THEN 250
240 W=W+1:NEXT
250 REM===PRINTEN VAN HET GETAL=====
260 PRINT#4,TAB(GG+1-W);NR;:
270 PRINT#4,TAB((DG+GG)-(GG-W+NN)+1);:
280 IFA<2 THEN 180
290 PRINT#4
300 A=0
310 GOTO 180
```

140 In de variabele GG komen de posities te staan van de gehele getallen (plaatsen voor de komma).

- 150 De variabele DG bevat het aantal posities van de decimale getallen. Houdt u zich bij de invoer aan de waarden die in regel 140/150 zijn gedefinieerd.
- 200-240 bereiden het plotten voor door de positie van de decimale punt te localiseren.
- 260 Plot het getal. Met regel 270 wordt de rest van het bereik tot aan het volgende getal opgevuld met spaties. Daarna worden de cijfers van het eerste getal geordend. Het referentiepunt is de plaats van de decimale punt.

U kunt het programma alleen beëindigen met RUN/STOP-RESTORE omdat na het plotproces een nieuw getal wordt gevraagd. U moet daarna de paperfeed-toets gebruiken om de pennen in de uitgangspositie te zetten. De print-using biedt een hoop variatiemogelijkheden. U kunt bijvoorbeeld nog een string opnemen, om een voorraadlijst inclusief prijzen te maken. Dit krijgt u gedaan door een bepaalde tekstlengte te definiëren, bijvoorbeeld 25 plaatsen. Dit getal dient als richtlijn voor de volgende uitvoer in dezelfde regel. Tot aan deze markering wordt de tekst met spaties opgevuld als er in de tekst minder dan 25 plaatsen staan. De invoer zou er dan zo uit kunnen zien:

```
PRINT#4,"TEXT";(gedef. tekstlengte=werkelijke tekstlengte)
```

De print-using speelt bij de 1520 geen grote rol omdat die zich eerder voor tekendoeleinden leent. Daar komen we dadelijk op terug, maar eerst bespreken we de mogelijkheid om de Commodore 1520 als schrijfmachine te gebruiken in de directe modus via het toetsenbord van de Commodore 64. Deze mogelijk is vooral nuttig als u zich niet vooraf wilt vastpinnen op een tekst. Dit is het geval bij opschriften, tekeningen enzovoort. Hiervoor is maar een klein programma nodig:

```
100 REM===1520 INSTELLEN=====
110 OPEN4,6
120 REM===TOETS INLEZEN=====
130 PRINT"s"
140 POKE198,0
150 WAIT198,1
160 GETX$:PRINTX$;;
170 IFX$="e"THEN220
180 REM===TEKEN PRINTEN=====
190 PRINT#4,X$;;
200 GOTO140
210 REM===EINDE=====
220 CLOSE4
```

De invoer loopt in dit geval via de toetsenbuffer. In \$00C6 (198) staat het getal van het karakter dat moet worden uitgevoerd.

- 140 wist de inhoud van de buffer door een nul in de geheugenplaats te schrijven.
- 150 houdt het programma tegen tot er op geheugenplaats 198 precies 1 teken staat. WAIT 198,3 zou betekenen dat er pas na invoering van het derde teken verder wordt gewerkt. Het teken wordt met GET gelezen en in regel
- 190 uitgevoerd. Het symbool in regel
- 170 is de afkorting voor functietoets 1. Als u deze toets indrukt wordt het programma beëindigd.

Bij het maken van statistieken met de Commodore 1520 is de directe modus nodig om een staafdiagram van tekst te voorzien. Nu komen we bij een programma waarmee u op zeer creatieve wijze met de plotter kunt werken, de JOYDRAWER. Deze lelijke term wil alleen maar zeggen, dat u met de adressen 56320 of 56321 (afhankelijk van de port) de richting bepaalt waarin de joystick beweegt.

```
100 REM====JOYSTICK UITLEZEN=====
110 X=PEEK(56320)
120 IF(XAND1)=0THENGOTO *REGELNUMMER*
130 IF(XAND2)=0THENGOTO *REGELNUMMER*
140 IF(XAND4)=0THENGOTO *REGELNUMMER*
150 IF(XAND8)=0THENGOTO *REGELNUMMER*
160 IF(XAND16)=0THENGOTO*REGELNUMMER*
170 GOTO110
```

Door een richting te kiezen stuurt u het programma naar een bepaalde regel om het instructies te laten uitvoeren. Het uitlezen van de joystick brengt een probleem met zich mee dat bij voorbaat al het tekenplezier kan bederven. Zoals u weet is de joystick opgebouwd uit membraamschakelaars, een voor de drukknop en de andere voor de verschillende richtingen. Deze schakelaars worden met plastic stiften bediend, die op hun beurt weer met de stuurknuppel in verbinding staan. Als u die in een diagonale richting duwt, worden er twee schakelaars in een keer bediend. Maar in bovengenoemd programma hebt u daar niets aan. Er kunnen zo nooit twee schakelaars tegelijk ingelezen worden, omdat de computer zich niet in tweeën kan delen om twee GOTO-instructies

tegelijkertijd uit te voeren. Dus wordt het signaal van de schakelaar die het eerst contact maakt, het eerst afgehandeld. Dat betekent dat u geen diagonale lijn kunt trekken. In plaats daarvan verschijnt een lijn die meer op een trap lijkt. We moeten daarom onze toevlucht nemen tot een andere methode en de boel zonder GOTO regelen. We moeten er ook om denken dat het leesprogramma en de uitvoering zo met elkaar gekoppeld worden dat er een grote lus ontstaat. Die lus moet per seconde zo vaak doorlopen worden dat het net lijkt alsof twee schakelaars tegelijk ingelezen worden, zodat er een diagonale lijn verschijnt. Een mogelijke oplossing biedt dit programma:

```

100 REM====1520 INSTELLEN=====
110 SG=3:C=2
120 OPEN4,6
130 OPEN1,6,1
140 OPEN2,6,2:PRINT#2,C
150 REM====JOYSTICK UITLEZEN=====
160 J=PEEK(56320)
170 IF(JAND1)=0 THEN Y=Y+SG
180 IF(JAND2)=0 THEN Y=Y-SG
190 IF(JAND4)=0 THEN X=X-SG
200 IF(JAND8)=0 THEN X=X+SG
210 IF(JAND16)=0 AND KK=0 THEN KK=1:GOTO240
220 IF(JAND16)=0 AND KK=1 THEN KK=0
230 REM====OVERLOOP TESTEN=====
240 IF X<0 THEN X=0:GOTO160
250 IF Y<-480 THEN Y=-480:GOTO160
260 IF X>480 THEN X=480:GOTO160
270 IF Y>480 THEN Y=480:GOTO160
280 REM====TEKENEN=====
290 IF KK=1 THEN K$="D"
300 IF KK=0 THEN K$="M"
310 PRINT#1,K$;X;Y
320 GOTO160

```

110 De variabelen SG (stapgrootte) en C (kleur) bevatten de oplossing (dus de tekensnelheid) en de kleur van de pen. Stapgrootte 1 komt overeen met de hoogste oplossing bij het tekenproces.

160-200 De vier basisrichtingen van de joystick worden ingelezen en daarbij worden de variabelen X en Y vergroot of verkleind, afhankelijk van de bewegingsrichting. Ter oriëntatie zijn de variabelen zo gekozen dat ze overeenkomen met de aanduiding van de assen: als u de stuurknuppel naar voren duwt, wordt Y vergroot (positieve Y-as).

- 210 De drukknop wordt ingelezen en variabele KK van 0 veranderd in 1. Dit is nodig om de pen op het papier te zetten en van het papier te halen. Als KK 1 is, wordt 220 overgeslagen. Er volgt dan een directe sprong naar de tekenroutine.
- 220 Als de drukknop een tweede keer wordt bediend, wordt regel 210 overgeslagen, omdat aan de eis $KK=0$ niet is voldaan. Stond de pen nog op het papier ($KK=1$) dan haalt de volgende beweging van de stuurknuppel hem eraf.
- 240-270 Met de overflow-controle wordt voorkomen dat de variabelen X en Y verder vergroot worden als de pen aan de rand komt.
- 290-300 De teken-/bewegingsinstructie (D/M) wordt in de variabele K\$ gedefinieerd afhankelijk van de toestand van KK.
- 310 De tekeninstructie wordt samen met de actuele waarden X en Y uitgevoerd, waardoor de routine opnieuw doorlopen kan worden.

Als u het programma hebt overgetypt, kunt u het starten en zullen de voordelen heel snel duidelijk worden. De tekensnelheid is heel redelijk en de diagonalen komen traploos op papier. Nog een paar woorden over de bediening: als u drukknop indrukt, gebeurt er niets met de stiften. Pas als u de stuurknuppel beweegt, wordt er getekend. Dit heeft het volgende voordeel: tijdens de tekenbeweging kan tegelijk de drukknop worden ingedrukt, waardoor het tekenen van stippellijnen mogelijk wordt gemaakt (al naar gelang de stapgrootte). De stift kan daardoor steeds worden gezet of van het papier worden gehaald. De joystick moet u overigens op port 2 aansluiten. Lezers die geen joystick hebben, kunnen de toetsen gebruiken. Daartoe verandert u in regel 160 adres 56320 in 56321. Hiermee schakelt u over naar port 1, die u met stuursignalen ook via het toetsenbord rechtstreeks kunt bereiken.

TOETS	FUNCTIE
2	rechts
CTRL	links
1	boven
<	onder
SPACE	drukknop

De routine kan worden uitgebreid door de functietoetsen erbij te betrekken.

```

100 REM===DATA INLEZEN=====
110 PRINT"s"
120 FORF=0TO3
130 READB$(F)
140 NEXT
150 REM===1520 INSTELLEN=====
160 SG=3:C=0
170 OPEN4,6
180 OPEN1,6,1
190 OPEN2,6,2
200 PRINT#2,C
210 REM===JOYSTICK UITLEZEN=====
220 GETA$:J=PEEK(56320)
230 IF(JAND1)=0THENY=Y+SG
240 IF(JAND2)=0THENY=Y-SG
250 IF(JAND4)=0THENX=X-SG
260 IF(JAND8)=0THENX=X+SG
270 IF(JAND16)=0ANDKK=0THENKK=1:GOTO350
280 IF(JAND16)=0ANDKK=1THENKK=0
290 IFA$=CHR$(133)THENCLOSE1:CLOSE2:END
300 IFA$=CHR$(134)THENSG=SG+1
310 IFA$=CHR$(138)ANDSG>1THENSG=SG-1
320 IFA$=CHR$(135)THENC=C+1
330 IFC=4THENC=0
340 REM===OVERLOOP TESTEN=====
350 IFX<0THENX=0:GOTO200
360 IFY<-480THENY=-480:GOTO200
370 IFX>480THENX=480:GOTO200
380 IFY>480THENY=480:GOTO200
390 REM===TEKENEN=====
400 IFKK=1THENK$="D"
410 IFKK=0THENK$="M"
420 PRINT#1,K$;X;Y
430 PRINT"STEP";SG,B$(C)"
440 PRINT"gg"
450 IFSG<10THENPOKE1070,32
460 GOTO200
470 DATAZWART,BLAUW,GROEN,ROOD

```

TOETS

FUNCTIE

F1	einde van het programma
F2	verhoogt de stapgrootte met 1
F4	verlaagt de stapgrootte met 1
F5	draait de pennen een kwartslag

De kleuren in het programma kunnen worden verwisseld en de oplossing is variabel geworden.

8.6 Statistiek met de VC-1520

Ook al werkt u normaal niet met statistieken, soms kunt u er niet buiten als u een aantal getallen grafisch moet weergeven. Dat kunnen bijvoorbeeld meetwaarden of andere gegevens zijn, die in procenten worden uitgedrukt. We tonen u dadelijk twee vaak toegepaste mogelijkheden van statistische weergave met de plotter 1520. Dat zijn staafdiagrammen en schijfdiagrammen. Beide programma's verwerken de waarden in procenten en geven deze ook in procenten weer. We beginnen met het staafdiagram:

```
100 REM===1520 INSTELLEN=====
110 OPEN4,6
120 OPEN1,6,1
130 OPEN2,6,2
140 REM===INVOER=====
150 INPUT"§HOEVEEL GEGEVENS (MAX.7)";A$
160 A=VAL(A$)
170 IFLEN(A$)>10RA<10RA>7THEN150
180 FORS=1TOA
190 PRINT
200 PRINTS;" . GEGEVEN(IN %)"::INPUTB(S)
210 IFB(S)>100THEN150
220 NEXT
230 REM===COORDINATEN=====
240 PRINT#1,"D";450;0
250 PRINT#4,"X"
260 PRINT#1,"M";20;-20
270 PRINT#1,"D";20;520
280 PRINT#4,"Y"
290 PRINT#1,"M";20;-480
300 PRINT#1,"I"
310 REM===MAATSTAF AANLEGGEN=====
320 FORE=0TO500STEP50
330 PRINT#1,"R";0;E
340 PRINT#1,"J";-10;E
350 NEXT
360 PRINT#1,"H"
370 REM===KOLOMMEN=====
380 FORS=1TOA
390 PRINT#1,"R";50;0:PRINT#1,"I"
400 Y(S)=B(S)*5
410 PRINT#1,"J";0;Y(S)
420 PRINT#1,"J";40;Y(S)
430 PRINT#1,"J";40;0
440 NEXT
```

```

450 PRINT#1,"R";-A*50;0
460 PRINT#1,"I"
470 PRINT#1,"R";50;-20
480 PRINT#1,"I"
490 REM===VAN TEKST VOORZIEN=====
500 POKE198,0:WAIT198,1
510 GETX$:IFX$<>"e"THENPRINT#4,X$;;
520 IFX$<>"e"THEN500
530 REM===EINDE=====
540 CLOSE1
550 CLOSE2
560 CLOSE4

```

Met dit programma kunt u maximaal zeven kolommen tekenen, die een maximale lengte van 10 centimeter (=100 procent) mogen hebben. Op de Y-as wordt elke centimeter gemarkeerd. Een centimeter is dan 10 procent. De tekst is zeer variabel gehouden omdat na beëindiging van het tekenproces op de directe modus (de schrijfmachine) wordt overgeschakeld. Met de toetsen kunt u in dat geval getallen, merktekens en dergelijke onder de kolommen van het staafdiagram schrijven. Hier volgt een analyse van het programma:

- 150 vraagt naar het aantal kolommen dat u wilt afbeelden.
- 170 controleert of de invoer correct was.
- 180-220 doorloopt een invoerroutine zo vaak als er kolommen zijn aangegeven. Let erop dat alleen numerieke invoer plaatsvindt, het programma controleert dat niet. De ingevoerde waarden worden in een 1-dimensionaal veld opgeslagen. In regel 210 wordt getest of de getallen kleiner dan of gelijk aan 100 procent zijn. Als de invoer zonder fouten heeft plaatsgevonden, wordt met de regels
- 240-300 een coördinatenstelsel met tekst bij de assen geconstrueerd. Dit programma-onderdeel is niet zo moeilijk.
- 320-350 geeft de onderverdeling van de Y-as in stappen van 1 centimeter. Omdat 100 procent overeen moet komen met 1 centimeter is in regel 320 gekozen voor STEP 50. Een tussenstap bedraagt 0.02 millimeter en $50 * 0.02$ is 1 centimeter. Dit wordt in totaal 10 keer uitgevoerd. De totale afstand wordt vanaf nu als 100 procent beschouwd. In de regels

380-440 kunnen nu de eigenlijke kolommen van het staafdiagram worden getekend.

Het volgende onderdeel moeten we iets nauwkeuriger bekijken omdat het belangrijk is voor een eventuele verandering van de breedte van de kolommen enzovoort.

380 De tekenlus begint met de eerste waarde en eindigt met A, het aantal kolommen.

390 Hier wordt bij elke doorloop een nieuw relatief nulpunt gedefinieerd, nadat de pennen een stuk verder naar rechts zijn geschoven. Daardoor kan de tekenroutine gelijk blijven en hoeft alleen het begin van de kolommen te worden verplaatst.

400 De percentages in B(S) worden hier omgerekend in lengte-eenheden: 500 tekenstappen = 10 centimeter = 100 procent. Met een eenvoudige formule kunt u de factor bepalen waarmee de invoer van de percentages vermenigvuldigd moet worden om tot de juiste lengte-eenheid te komen.

Nu geldt:

500 stappen = 100 %

X stappen = B(S) %

Alleen X is nog onbekend, want B(S) is al ingevoerd

Formule:

500 stappen = 100 % = X stappen B(S) %

Daaruit volgt:

B(S) % - 500 stappen, 100 % = X stappen

en daarmee krijgt u de formule in regel 400

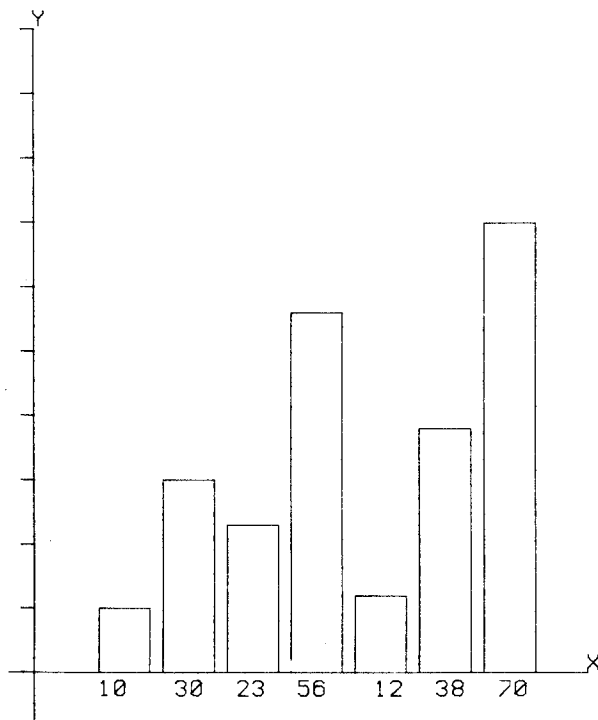
410-440 De kolommen worden op papier gezet. De breedte (40) is vooraf bepaald en de hoogte staat per kolom in Y(S).

450-480 De plotter wordt voorbereid op de directe modus voor de tekst. De pen wordt onder de eerste kolom geplaatst, het nieuwe relatieve nulpunt.

500-520 Deze routine verwacht een toets, die vervolgens wordt geplotted (schrijfmachine). Als u klaar bent met de tekst, kunt u met functietoets 1 het programma beëindigen.

We herhalen de uitleg bij regel 500 nog eens kort. Daar wordt de toetsenbuffer, of liever gezegd het aantal tekens in de toetsenbuffer op nul gezet. Dit gebeurt met een POKE-instructie. Bij 198 is het aantal tekens in het geheugen opgeslagen. WAIT 198,1 wacht tot er precies 1 teken in de buffer staat. Dan wordt

het met GET in een variabele gehaald en geplot. De routine valt op door een precieze oplossing en een redelijke snelheid. Als u meer kolommen nodig heeft, kunt u de listing veranderen door in regel 390 het getal 50 te verhogen of te verlagen. Meteen wordt dan ook de breedte van de staven in regel 420/430 gewijzigd. Hieronder het resultaat van een proefuitdraai.



Ook het schijfdiagram is zeer geschikt voor de weergave van procenten. Dit is een zeer overzichtelijke manier omdat de cirkel gelijk is aan 100 procent. Met een beetje inspanning is het mogelijk dit soort werkzaamheden door de Commodore 1520 te laten verrichten. U hoeft dan zelf alleen nog maar de percentages in te voeren. Om het programma een beetje overzichtelijk te houden, worden eerst alleen de stralen getekend, aan het eind volgt dan de cirkel. Zo kunnen onnauwkeurigheden, die bij de sinus en de cosinus veelal optreden, vermeden worden. Met ons programma kunt u maximaal 30 percentages in een keer verwerken. Aan het begin vraagt het programma hoeveel tekens u wilt invoeren. Dan wordt de tekening vervaardigd. Dit proces kost wat tijd (bij 10 gegevens

ongeveer 1 minuut) maar het resultaat is betrouwbaar.

```
100 REM===1520 INSTELLEN=====
110 OPEN1,6,1
120 OPEN4,6
130 PRINT#1,"M";240;0
140 PRINT#1,"I"
150 REM===INVOER=====
160 DIMU(30)
170 INPUT"$$$HOEVEEL DELEN <=30!";M$
180 M=VAL(M$):M=ABS(M)
190 FORN=1TOM
200 PRINT"DEEL";N::
210 INPUTU(N)
220 NEXT
230 REM===TEST OP OVERLOOP=====
240 FORN=1TOM
250 P=P+U(N)
260 IFP>100THENCLR:GOTO170
270 PRINT"SPROEFSOM";TAB(20)P;" "
280 NEXT:N=0
290 REM===ONDERVERDELING=====
300 N=N+1
310 K=128*U(N)/100
320 FORL=0TO6.4STEP.05:T=T+1
330 X=200*SIN(L)
340 Y=200*COS(L)
350 IFT-K<1ANDT-K>=0THEN400
360 IFL=0THENPRINT#1,"J";X;Y
370 PRINT#1,"R";X;Y
380 NEXT:GOTO470
390 REM===VOLGENDE WAARDE=====
400 T=0
410 PRINT#1,"J";0;0
420 PRINT#1,"R";X;Y
430 N=N+1
440 K=128*U(N)/100
450 GOTO370
460 REM===CIRKEL TEKENEN=====
470 FORL=0TO6.4STEP.05
480 X=200*SIN(L)
490 Y=200*COS(L)
500 IFL=0THENPRINT#1,"R";X;Y
510 PRINT#1,"J";X;Y
520 NEXT
530 REM===VAN TEKST VOORZIEN=====
540 PRINT#1,"R";0;200
550 POKE198,0:WAIT198,1
560 GETA$:IFA$="e"THEN600
```

```

570 PRINT#4,A$;:
580 GOTO550
590 REM====EINDE=====
600 CLOSE1
610 CLOSE4
620 POKE214,12
630 PRINT

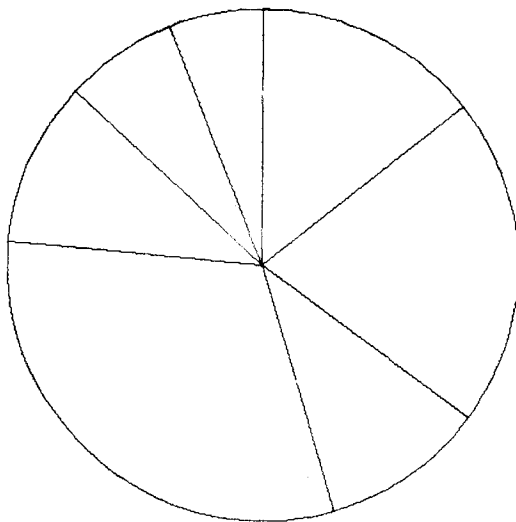
```

Bij zo'n gecompliceerd programma hoort een uitgebreide uitleg, zodat u de variatiemogelijkheden beter begrijpt.

- 160 De percentages die de gebruiker invoert, worden in een 1-dimensionaal veld opgeslagen. Maar omdat dit veld meer dan de normaal toegestane 11 waarden kan bevatten, moet het gedimensioneerd worden. Gebeurde dat niet dan zou de foutmelding BAD SUBSCRIPT ERROR verschijnen. In ons programma is het door DIMV(30) mogelijk in totaal 31 gegevens in te voeren.
- 170-220 De invoer van de gebruiker wordt ingelezen en in het zojuist gedimensioneerde veld opgeslagen. Overigens staat er in regel 170 geen fout als er in de begeleidende tekst gesproken wordt over ...<=30, terwijl er 31 elementen zijn toegestaan. Het ene element is niets anders dan een zekerheidsmarge.
- 240-280 De veldinhoud wordt opgeteld en er volgt een overflow controle. Daarbij zijn alle ingevoerde waarden toegestaan zolang de som kleiner of gelijk aan 100 procent is. Als de som van de ingevoerde gegevens kleiner dan 100 procent is, is de cirkel niet gesloten.
- 310-350 De waarde K wordt bepaald. Bij die waarde wordt de cirkel verlaten en de straal getekend. Deze formule kunt u alleen begrijpen in samenhang met de volgende regel. Daar wordt namelijk de tekening van de cirkel voorbereid met FORL=0 TO 6.4 STEP .05. De tekening bestaat na 6.4/.05 uit 128 stappen. Dit getal verschijnt weer in regel 310, waar het percentage wordt veranderd in plotterformaat. Zo krijgt u na de formule voor K precies de positie op de cirkel die overeenkomt met het ingevoerde percentage. Tegelijk wordt met regel 320 bij elke stap de tussensom (T) met 1 verhoogd. Dat wordt herhaald tot K ongeveer gelijk is aan de tussensom. Regel 350 controleert dit. De regels 330 tot 350 horen bij de cirkelroutine. De constante (hier 200) bepaalt de straal.

- 360 bereidt de sprong naar de subroutine voor, wanneer K en Z bij benadering aan elkaar gelijk zijn. De subroutine is te vinden vanaf regel 400.
- 400-450 De volgende waarde wordt gelezen, de constante wordt opnieuw berekend en de tussensom ongedaan gemaakt. Daarna springt het programma terug naar de tekenroutine.
- 470-520 Deze regels hoeven we niet meer te beschrijven. De cirkel wordt om de stralen getekend. Hoe de begeleidende tekst in
- 540-580 tot stand komt is ook bekend. Als de tekst klaar is, kunt u met functietoets 1 het programma beëindigen.

De statistiekfuncties hebben we theoretisch zo breed uitgemeten omdat de plotter ook op dit gebied heel handig is. Op basis van deze uitleg moet u nu in staat zijn zelf een programma te ontwikkelen dat aan uw wensen voldoet. Hieronder een voorbeeld van een schijfdiagram:



VOORBEELD VAN EEN ONBESCHREVEN
SCHIJFDIAGRAM

8.7 Hulpprogramma voor de VC-1520

Het volgende programma in BASIC brengt nogal wat typewerk met zich mee, maar het is de moeite waard. Het programma ondersteunt de secundaire adressen van de Commodore 1520 en biedt zelfs voorgeprepareerde routines, zoals de cirkel of de lijn. Bovendien kan de gebruiker het naar wens uitbreiden. De snelheid is naar BASIC-maatstaven zeer redelijk. Het programma is zo geconstrueerd, dat het Engelse en Nederlandse instructies accepteert. We gunnen u alvast een kijkje in de instructiekeuken.

CIRCLE,Mx,My,Rx,Ry,L tekent een cirkel, waarin het middelpunt, de stralen in X- en Y-richting en de lijnsoorten vrij te kiezen zijn.

LINE,X1,X2,Y1,Y2,L tekent een lijn van het ene punt naar het andere, waarbij de lijnsoort vrij gekozen kan worden.

PAPER UP/DOWN,N verschuift het papier in n stappen naar boven of naar beneden.

TURN-TEXT; TURN/TEXT draaien elke invoer 90 graden. De eerste variant doet dat langs de X-as, de tweede langs de Y-as.

Vanzelfsprekend kan ook de kleur, de grootte van de karakters, enzovoort veranderd worden. Hierop komen we terug bij de beschrijving van de instructieset. Bekijkt u eerst eens de

```
10 OPEN1,6,1
20 OPEN2,6,2
30 OPEN3,6,3
40 OPEN4,6,4
50 OPEN5,6,5
60 OPEN6,6,6
70 OPEN7,6,7
80 OPEN8,6:PRINT"§":DIMEN$(100):DIMW1(10
0):RESTORE
90 GOSUB9000:POKE53280,0:POKE53281,2:PRI
NT"E"
100 REM====INVOER====
110 E$="←":PO=56320
120 PRINT"< "":
130 GETA$:IFA$=""THEN130
140 IFA$="Q"ORA$="q"ORA$="1"ORA$="△"ORA$
=CHR$(34)ORA$="§"ORA$="§"THEN130
```

```

150 IFA$=CHR$(13)THENPRINT:PRINT"q<":Z=0
:S=S+1:GOTO240
160 IFA$=CHR$(20)THENZ=Z-1:IFZ<0THENZ=0:
GOTO130
170 IFA$=CHR$(20)THEN190
180 Z=Z+1:IFZ>70THENA$=CHR$(13):GOTO150
190 PRINTTAB(3)"▲"A$+E$;;
200 IFA$="*"THEN330
210 T$=T$+A$
220 GOTO130
230 IFA$=CHR$(20)THENTT$=LEFT$(TT$,LEN(T
T$)-2)
240 Y=LEN(T$):E=E+1
250 FORX=1TOY
260 S$=MID$(T$,X,1)
270 IFS$=" "ORS$="↑"ORS$="←"THENNEXT
280 EN$(E)=EN$(E)+S$
290 IFS$<>" "ANDS$<>"↑"ANDS$<>"←"THENNEX
T
300 T$="":A$="":S$=""
310 GOTO120
320 REM===VERWERKEN=====
330 E=1
350 REM===INSTRUCTIE LEZEN=====
360 FORX=1TOLEN(EN$(E))
370 BF$=MID$(EN$(E),X,1)
380 IFBF$<>"",THENB1$=B1$+BF$:NEXT
390 REM===GETALLEN LEZEN=====
400 FORX=LEN(B1$)+2TOLEN(EN$(E))
410 Z1$=MID$(EN$(E),X,1)
420 IFZ1$<>"",THENN1$=N1$+Z1$:NEXT
430 REM===GETALLEN LEZEN=====
440 FORX=LEN(N1$)+2+LEN(B1$)+1TOLEN(EN$(
E))
450 Z2$=MID$(EN$(E),X,1)
460 IFZ2$<>"",THENN2$=N2$+Z2$:NEXT
470 REM===GETALLEN LEZEN=====
480 FORX=LEN(N1$)+1+LEN(B1$)+1+LEN(N2$)+
2TOLEN(EN$(E))
490 Z3$=MID$(EN$(E),X,1)
500 IFZ3$<>"",THENN3$=N3$+Z3$:NEXT
510 REM===GETALLEN LEZEN=====
520 FORX=LEN(N1$)+1+LEN(B1$)+1+LEN(N2$)+
1+LEN(N3$)+2TOLEN(EN$(E))
530 Z4$=MID$(EN$(E),X,1)
540 IFZ4$<>"",THENN4$=N4$+Z4$:NEXT
550 REM===GETALLEN LEZEN=====
570 FORX=LEN(N1$)+1+LEN(B1$)+1+LEN(N2$)+
1+LEN(N3$)+1+LEN(N4$)+2TOLEN(EN$(E))
580 Z5$=MID$(EN$(E),X,1)
590 IFZ5$<>" "THENN5$=N5$+Z5$:NEXT

```

```

600 REM===STRINGS IN GETALLEN=====
610 N1=VAL(N1$):N1=ABS(N1)
620 N2=VAL(N2$):N2=ABS(N2)
630 N3=VAL(N3$):N3=ABS(N3)
640 N4=VAL(N4$):N4=ABS(N4)
650 N5=VAL(N5$):N5=ABS(N5)
660 REM===UITVOEREN=====
670 REM===DE INGEVOERDE INSTRUCTIES===
680 IFB1$<>"COLOR"ANDB1$<>"KLEUR"THENGOT
0710
690 IFN1>255THENPRINT#1,0:GOTO710
700 PRINT#2,N1
710 IFB1$<>"PEN"THEN740
720 IFN1>255THENN1=0
730 PRINT#2,N1:PRINT#1,"M";480,0
740 IFB1$<>"LINE"ANDB1$<>"LIJN"THEN780
750 IFN1>480ORN2>999ORN3>480ORN4>999ORN5
>15THEN780
760 PRINT#5,N5
770 PRINT#1,"M";N1;N2:PRINT#1,"D";N3;N4
780 IFB1$<>"SIGN"ANDB1$<>"TEKEN"THENGOTO
810
790 IFN1>255THENN1=1
800 PRINT#3,N1
810 IFB1$<>"TEST"THEN830
820 PRINT#8,"T E S T"
830 IFB1$<>"PAPERUP"ANDB1$<>"PAPIER NAAR
BOVEN"THEN870
840 IFN1>999THENN1=998
850 PRINT#1,"M";0;N1
860 PRINT#8
870 IFB1$<>"PAPERDOWN"ANDB1$<>"PAPIER NA
AR BENEDEN"THEN910
880 IFN1>999THENN1=998
890 PRINT#1,"M";0;-N1
900 PRINT#8
910 IFB1$<>"HOME"ANDB1$<>"BEGIN"THEN930
920 PRINT#1,"H"
930 IFB1$<>"CIRCLE"ANDB1$<>"CIRKEL"THENG
OTO1010
940 IFN1+N3>480ORN2+N4>999ORN5>15ORN1-N3
<0THENPRINT"WAARDEN TE HOOG"
950 PRINT#1,"M";N1;N2:PRINT#1,"I":PRINT#
5,N5
960 FORX=-3.2TO3.2STEP.04
970 Y=N3*COS(X):Z=N4*SIN(X)
980 IFX=-3.2THENPRINT#1,"R";Y;Z
990 PRINT#1,"J";Y;Z:NEXT
1000 PRINT#1,"M";0;0:PRINT#1,"I"
1010 IFB1$<>"BASIC"THEN1030
1020 SYS64738

```

```

1030 IFLEFT$(B1$,5)<>"TURN-"ANDLEFT$(B1$,5)<>"DRAAI-"THEN1090
1040 FORX=LEN(B1$)TO6STEP-1:PRINT#4,1
1050 Y$=MID$(B1$,X,1):
1060 PRINT#1,"M",N1*(LEN(B1$)-X)
1070 PRINT#8,Y$;:NEXT:PRINT#4,0
1080 PRINT#8
1090 IFLEFT$(B1$,5)<>"TURN/"ANDLEFT$(B1$,5)<>"DRAAI/"THEN1130
1100 PRINT#4,1:FORX=6TOLEN(B1$)
1110 Y$=MID$(B1$,X,1)
1120 PRINT#8,Y$:NEXT:PRINT#8:PRINT#4,0
1130 IFB1$<>"SCREEN"ANDB1$<>"SCHERM"THEN
1160
1140 IFN1>150RN2>15THENN1=0:N2=2
1150 POKE53280,N1:POKE53281,N2
1160 IFB1$<>"0-SET"THEN1190
1170 PRINT#2,0:PRINT#3,1:PRINT#4,0:PRINT
#5,0:PRINT#6,0
1180 POKE53280,0:POKE53281,2:PRINT"E"
1190 IFB1$<>"DIRECT"THEN1250
1200 IFN1>255ORN2>255THEN1230
1210 PRINT#6,N1:PRINT#3,N2
1230 POKE198,0:WAIT198,1
1240 GETA$:IFA$<>"_ "THENPRINTA$;:PRINT#8
,A$;:GOTO1230
1250 IFB1$<>"PORT"THEN1290
1260 IFN1>20RN1<1THENPO=56320
1270 IFN1=1THENPO=56321
1280 IFN1=2THENPO=56320
1290 IFB1$<>"JOY"THEN1680
1320 PRINT:RESTORE:FORF=0TO3
1330 READB$(F)
1340 NEXT
1350 REM===1520 INSTELLEN=====
1360 SW=3:C=0
1400 PRINT#2,C
1410 REM===JOYSTICK UITLEZEN=====
1420 GETA$:J=PEEK(PO)
1430 IF(JAND1)=0THENY=Y+SW
1440 IF(JAND2)=0THENY=Y-SW
1450 IF(JAND4)=0THENX=X-SW
1460 IF(JAND8)=0THENX=X+SW
1470 IF(JAND16)=0ANDKK=0THENKK=1:GOTO155
0
1480 IF(JAND16)=0ANDKK=1THENKK=0
1490 IFA$=CHR$(133)THEN8000
1500 IFA$=CHR$(134)THENSW=SW+1
1510 IFA$=CHR$(138)ANDSW>1THENSW=SW-1
1520 IFA$=CHR$(135)THENC=C+1
1530 IFC=4THENC=0

```

```

1540 REM===OVERLOOP TESTEN=====
1550 IFX<0THENX=0:GOTO1400
1560 IFY<-480THENY=-480:GOTO1400
1570 IFX>480THENX=480:GOTO1400
1580 IFY>480THENY=480:GOTO1400
1590 REM===TEKENEN=====
1600 IFKK=1THENK$="D"
1610 IFKK=0THENK$="M"
1620 PRINT#1,K$;X;Y
1630 PRINT"STEP";SW,B$(C)"      "
1640 PRINT"gg"
1650 IFSW<10THENPOKE1070,32
1660 GOTO1400
1670 DATAZWART,BLAUW,GROEN,ROOD
1680 IFB1$<>"PAUSE"ANDB1$<>"PAUZE"THEN80
00
1690 WAIT203,63
8000 REM===TERUGZETTEN=====
8010 N1$="":N2$="":N3$="":N4$="":N5$=""
8020 X$="":Y$="":N1=0:N2=0:N3=0:N4=0:N5=
0
8030 E=E+1:IFB1$<>" "THENB1$="":GOTO350
8040 CLR:RUN
9020 REM===STATUSREGEL=====
9010 PRINT"SE   *** PLOTTER 1520 HULPPRO
GRAMMA ***"
9020 RETURN

```

In regel 10-650 ziet u de invoer en de voorbereidingsroutine van het programma. Daarna volgt tot regel 1690 de instructieset, inclusief de uitvoering van die instructies. De regels 8000 tot 8040 starten het programma opnieuw. En in 9000 e.v. treft u de statusregel aan. Al deze deelprogramma's worden straks nog uitvoerig besproken, omdat er zeer veel stringbewerkingen in voorkomen. Het programma is erop ingericht dat de gebruiker eigen ideeën kan inbrengen. U mag zelf instructies definiëren. Hiervoor is een vrije ruimte opengelaten, de regels 1700-7990. Ter verduidelijking van de programmeerwijze zijn al enkele instructies ingebouwd om het programma te testen. Na een instructie kunt u ten hoogste 5 getallen invoeren. Ze moeten door een komma van elkaar worden gescheiden (bijvoorbeeld CIRCLE,100,100,90,80,0). In het programma staan de volgende instructies:

```

COLOR,x   x=0 zwart
          x=1 blauw
          x=2 groen
          x=3 rood

```

PEN,x stuurt de penhouder naar de wisselpositie aan de rechterkant van de rol. Voor x gelden dezelfde gegevens als voor COLOR.

LINE,x1,y1,x2,y2,L tekent een lijn van de begincoördinaten x1,y1 naar de eindcoördinaten x2,y2 met lijnsoort L. U kunt kiezen uit 0-15.

SIGN,x stelt de tekengrootte in. Zinvol zijn waarden tussen 0 (kleinste schrift) en 3 (grootste schrift).

TEST schrijft het woord "test" op papier.

PAPER UP,x schuift het papier met x stappen naar boven (dit is 0,2 mm per stap).

PAPER DOWN,x schuift het papier met x stappen naar beneden.

HOME brengt de penhouder naar de beginpositie terug en heeft betrekking op het absolute of op het relatieve nulpunt.

CIRCLE,Mx,My,Rx,Ry,L tekent cirkels of ellipsen. Daarbij is:

Mx = middelpuntscoördinaat op de X-as

My = middelpuntscoördinaat op de Y-as

Rx = straal in X-richting in tekenstappen

Ry = straal in Y-richting in tekenstappen

L = lijnsoort (0-15)

BASIC voert een systeemreset uit. Met deze instructie kunt u het programma beëindigen en wissen.

TURN-TEXT,x draait elke invoer op de X-as 90 graden. De afstand tussen de karakters bedraagt x stappen.

TURN/TEXT,x draait elke invoer op de Y-as 90 graden. De afstand tussen de karakters bedraagt x stappen.

SCREEN,r,b wist de beeldschermkleur en verandert deze met de kleurwaarden 0-15 voor r en b: r staat voor rand en b voor beeldig plaatje. De kleuren corresponderen met de volgende getallen:

0 zwart	8 oranje
1 wit	9 bruin
2 rood	10 lichtrood
3 cyaan	11 grijs1
4 violet	12 grijs2
5 groen	13 lichtgroen
6 blauw	14 lichtblauw
7 geel	15 grijs3

O-SET zet alle waarden terug op de defaultwaarde.

DIRECT,x,y maakt het mogelijk de Commodore 1520 te gebruiken als schrijfmachine:

x=0 hoofdletters ; met SHIFT kleine letters

x=1 kleine letters; met SHIFT hoofdletters

y kan tussen 0 en 3 liggen en bepaalt de grootte van de karakters.

PORT,x kiest de joystickport, voor x=1 of 2. Door het programma is PORT 2 ingesteld.

JOY schakelt over op de joystick, waarbij de functietoetsen de volgende betekenis hebben:

F1 verder in het programma

F3 stapgrootte verhogen met 1

F4 stapgrootte verlagen met 1

F5 wisseling van kleur

PAUZE stopt het programma tot er een toets wordt gebruikt.

Tot zover de instructieset, die u naar wens kunt uitbreiden. U kunt een instructie intypen. Vervolgens drukt u op de RETURN-toets. Als de instructie moet worden uitgevoerd, moet u op x drukken. Daarna verwacht het programma nieuwe invoer. Als u niet op x drukt, wacht het programma op een volgende instructie, die achter de vorige wordt gezet. Op die manier kunt u programma's schrijven van wel 100 instructies. Over het programma zelf valt ook nog wel wat te zeggen. De regels 10-90 bereiden het programma voor. Files worden geopend en het veld voor programma-opslag wordt gedefinieerd. De sprong naar 9000 levert de statusregel, vervolgens wordt het beeldscherm opgemaakt. De invoerroutine vanaf regel 100 is een stringbewerking. De karakters worden een voor een gecontroleerd door GET. Regel 140 blokkeert de cursortoetsen en beeldschermoperaties zoals CLR. In regel 210 wordt in de variabele T\$ alles opgeslagen wat er aan gegevens is ingevoerd. Per instructie zijn maximaal 70 tekens toegestaan, daarna vindt een AUTO-RETURN plaats. Regel 200 controleert of er opdracht is gegeven de zaak in behandeling te nemen. Anders wordt er automatisch teruggesprongen om de volgende instructie in ontvangst te nemen. Is die opdracht wel gegeven dan springt het programma naar 330. Bij elke RETURN volgt vanuit regel 150 een sprong naar regel 240 e.v., waar de instructie in een 1-dimensionaal veld (EN\$(E)) wordt opgeslagen. Daarbij wordt de instructie uit T\$ niet simpelweg overgedragen, maar worden er al voorbereidingen voor de afhandeling getroffen. Regel 270 filtert

namelijk spaties, de pijlcursor en het teken voor machtsverheffen (pijl omhoog) om de instructie in te krimpen. Als u in plaats van CIRCLE... bijvoorbeeld C IR C LE... invoert, worden alle spaties eruit gehaald. Onze ingevoerde gegevens bevinden zich als string (mengvorm van getallen en instructie) in het veld EN\$(E). Als * wordt ingedrukt, nadat alle invoer is beëindigd, springt het programma naar de leesroutine in regel 330. Die routine scheidt de instructie en de vijf getallen van elkaar en slaat ze op. De komma's zijn dus nodig om begin en einde van de instructie aan te geven. In regel 350-380 wordt eerst de instructie gelezen aan het begin van de string. De gekozen variabelen worden in variabele B1\$ opgenomen, totdat regel 380 een komma constateert. Dan wordt het eerste getal gelezen, dat in eerste instantie in de variabele N1\$ terechtkomt. De overige getallen komen in N2\$-N5\$ te staan. Regel 600-650 geeft de numerieke variabelen weer. De strings worden gewijzigd en de getallen komen in N1-N5 terecht. Nu is de invoerroutine pas beëindigd. De eigen instructies kunnen worden afgehandeld, wat bij ons vanaf regel 660 gebeurt. Als u het programma wilt uitbreiden, moet u met de genoemde variabelen werken. Als bijvoorbeeld een instructie als

CUBE, lengte v. zijden, beginpunt, kleur, draaiing, eind

gedefinieerd moet worden, zijn de getallen in de volgende variabelen te vinden:

CUBE	= B1\$
lengte van de zijden	= N1
beginpunt	= N2
kleur	= N3
vergroting	= N4
eindpositie	= N5

Nu kunnen de variabelen verwerkt worden. Bij een fout in de invoer wordt de instructie overgeslagen en de volgende instructie afgehandeld. Regel 8000 e.v. zet het programma en haast alle variabelen terug. De volgende instructie wordt uit het veld gelezen. Het programma is zo geconstrueerd dat het ook Nederlandse instructies accepteert. Bij sommige instructies als BASIC, PEN, DIRECT, JOY is geen vertaling nodig. De volgende equivalenten waren ons inziens zo niet nodig dan toch gewenst. Het is in ieder geval leuk als een apparaat je eigen taal verstaat.

COLOR	-	KLEUR
LINE	-	LIJN
SIGN	-	TEKEN
PAPER UP	-	PAPIER NAAR BOVEN
PAPER DOWN	-	PAPIER NAAR BENEDEN

HOME	-	BEGIN
CIRCLE	-	CIRKEL
TURN	-	DRAAI
SCREEN	-	SCHERM
PAUSE	-	PAUZE

We wensen u veel succes met de programmering van zelf ontwikkelde instructies. Volgens ons bent u nu we aan het eind van dit boek zijn gekomen zo ver gevorderd dat u zelf wel toepassingen voor dit laatste programma kunt verzinnen. Rest ons nog de hoop uit te spreken dat alle uitleg over de plotter 1520 u een stapje dichterbij dit apparaat heeft gebracht. De hier gedemonstreerde programma's en de uitleg daarvan waren niet bedoeld voor mensen die alles maar klakkeloos overtypen, maar eerder voor de geïnteresseerden, die ze als basis voor hun eigen creativiteit willen gebruiken.