

frankeren
als briefkaart

NewBrain-
gebruikersgroep

postbus 4494

1009 AL amsterdam

frankeren
als briefkaart

NewBrain-
gebruikersgroep

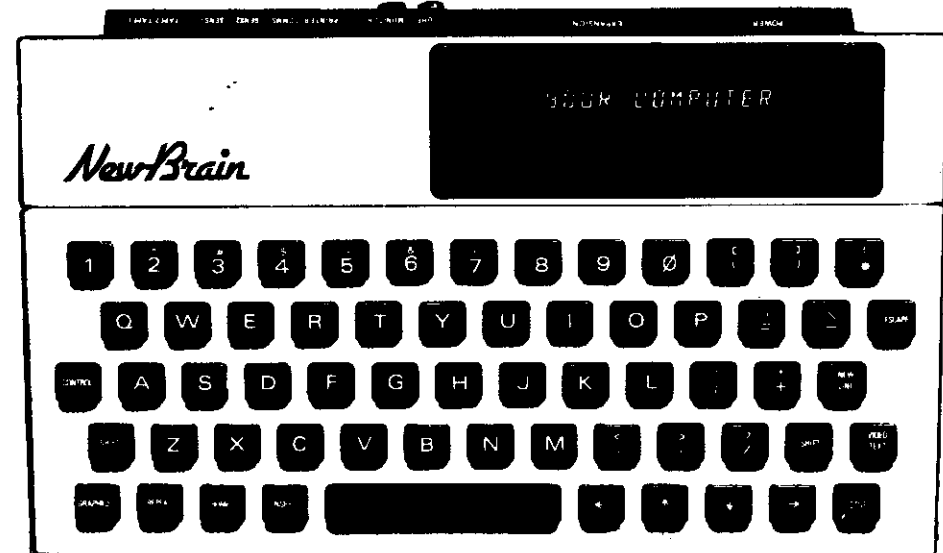
postbus 4494

1009 AL amsterdam

NewBrain
on-line

uitgave van de
NewBrain-gebruikersgroep
augustus 1985

5



newbrain on-line is een officiële uitgave van de nederlandse newbrain-gebruikersgroep. de bladen zijn licht gelijmd zodat ze eenvoudig los te maken en daarna in een ringband op te bergen zijn, de perforatie is afgestemd op de ringband met hcc-opdruk, die bij de hcc te koop is. geplaatste artikelen mogen alleen voor niet-commerciële doeleinden, onder bronvermelding, overgenomen worden.

het is voor de redactie nu eenmaal onmogelijk om alle ingezonden artikelen en programma's op originaliteit te controleren, de aansprakelijkheid voor de ingezonden stukken ligt derhalve dan ook bij de inzender.

voor informatie omtrent commerciële advertenties of advertenties van leden, kunt u contact opnemen met de redactie.

door de aanmeldingskaart achterop het omslag op te sturen kunt u zich verzekeren van de toezending van nieuw verschenen nummers van on-line

kopij

kopij opsturen naar het adres van de redactie:

newbrain-gebruikersgroep
postbus 4494
1009 AL amsterdam

het is de bedoeling, dat on-line 6 half januari 1986 verschijnt. sluitingsdatum is 18 december 1985. maar stuur uw kopij zo spoedig mogelijk op, dan hoeven we niet alle werk met de feestdagen te doen.

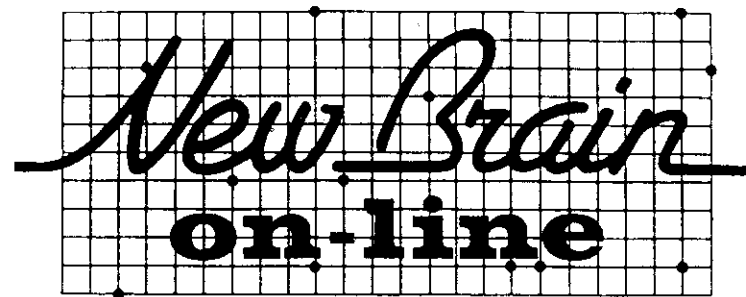
het is voor de redactie verreweg het makkelijkst de kopij aangeleverd te krijgen op cassette of diskette. dan kunnen we een uniforme lay-out maken en wordt het in elkaar zetten van de on-line wat minder werk. vermeld bij een diskette wel het formaat. de tekst mag op alle gangbare manieren zijn opgeslagen: text-, wordstar-, wp- en alle mogelijke ascii-bestanden zijn bruikbaar. de cassette of diskette wordt natuurlijk teruggezonden.

zeker als in een artikel een programma-listing van enige omvang voorkomt, is een cassette of diskette onontbeerlijk.

dit alles mag u natuurlijk niet beletten om kopij in te sturen. de redactie ontvangt liever kopij, waar ze wat meer werk mee heeft, dan helemaal geen kopij



memmo fecit



TEN GELEIDE

een nieuwe on-line met veel primeurs: er is een prijsvraag met interessante prijzen; in deze on-line beschreven of afgedrukte programma's zijn tesamen op een cassette in de softwarebibliotheek verkrijgbaar: dat bespaart u het overtypen met alle typfouten van dien.

daarnaast is er veel nieuws van het 'bibliotheekfront': de softwarebibliotheek is uitgebreid, verbeterd en gereorganiseerd. de programma's zijn nu verdeeld over systematische standaardpakketten. dat maakt de uitlevering veel efficiënter en sneller. u vindt in deze on-line uiteraard de nieuwe catalogus. meteen daarna is de catalogus van de onderwijsbibliotheek afgedrukt: een kleine 900 k aan software, die niet alleen op scholen van pas zal komen. en vergeet niet het grote nieuws, dat de cp/m-bibliotheek brengt: naast de uitbreiding van de newbrain-cp/m-software ook een fikse prijsverlaging.

verder is on-line vol met een verscheidenheid aan artikelen, zodat het voor alle categorieën newbrain-gebruikers een interessant nummer is geworden. dat is zowel aan de vaste leveranciers van kopij als aan de incidentele inzenders te danken. wij prijzen ons gelukkig het eerste artikel in een reeks van de kant van tradecom te kunnen publiceren: cor boer beschrijft, hoe je op een unexpanded newbrain een klokdevice implementeert. hopelijk volgt nog eens de beschrijving voor een expanded system.

u mist de ledenlijst. niet omdat er geen nieuwe leden zouden zijn (na de stormachtige groei van het begin blijft de toeloop boven verwachting; de penningmeester heeft zijn handen vol aan de ledenadministratie: er zijn nu 563 leden), maar omdat wij de beschikbare ruimte liever aan artikelen besteden. de ledenlijst zal eens per jaar volledig afgedrukt worden.

als u ons van genoeg kopij voorziet, verschijnt er bij elke landelijke bijeenkomst een nieuwe on-line (de inzendertermijn voor on-line 6 sluit 18 december 1985). de data van de landelijke bijeenkomsten in 1986 zijn: 18 januari, 7 juni en 18 oktober. maar eerst is er nog als ontmoetingspunt voor newbrain-gebruikers de stand van de gebruikersgroep op de hcc-dagen de redactie

NewBrain-gebruikersgroep

postbus 4494
1009 AL amsterdam

voorzitter	bert hovius	(038) 228252
secretaris	menno stevens	(020) 924137
secretaris buitenland	rob van albada	(020) 739671
penningmeester	jan de vries	(030) 511243
	postrekening 2700380	19.00 - 20.00 uur

de newbrain-gebruikersgroep is een onderdeel van de

HOBBY COMPUTER CLUB

inschrijvingsnummer kvk leiden: V445230



landelijke bijeenkomsten

coördinator	m s vreedenburg	(02159) 11068
-------------	-----------------	---------------

in 1986 te houden op de zaterdagen 18 januari, 7 juni en 18 oktober, telkens in 'de bron' in utrecht. voor elke bijeenkomst ontvangen de leden van de gebruikersgroep een uitnodiging met het programma

newbrain on-line

redactie	paul van oss	(020) 739022
	menno stevens	(020) 924137

sluitingsdatum voor de kopij voor on-line 6 is 18 december 1985. zie de binnenkant van het omslag voor het inzenden van kopij

softwarebibliotheek

bibliothecaris	jan tolkamp	(08894) 11786
----------------	-------------	---------------

zie bladzijde 7 voor de bestelprocedure en een overzicht van de programma's. achterin deze on-line is een bestelformulier afgedrukt

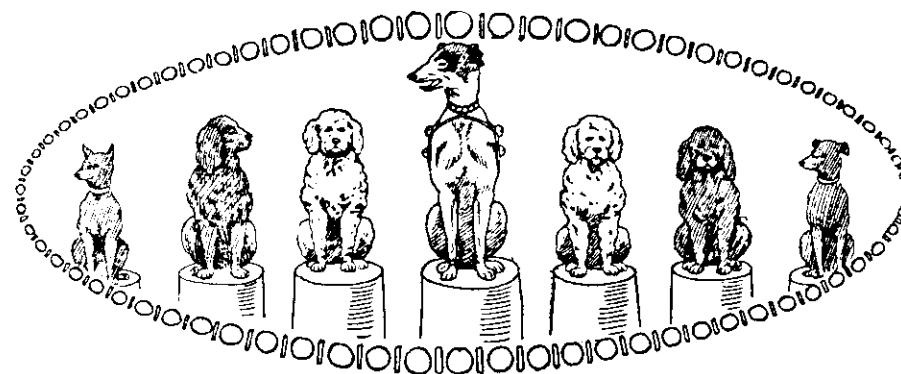
cp/m

bibliothecaris	jos hermans	(050) 142097
vraagbaak	henk blik	ma-za tot 17.00 uur
		di-avond tot 22.00 uur

zie pagina 25 voor de bestelprocedure en een overzicht van specifieke newbrain-cp/m-software. achter in deze on-line staat een bestelformulier

contactpersonen werkgroepen

onderwijs	pieter van dijk	(01751) 12367
basicode	pieter van dijk	(01751) 12367
hardware	rob maris	(055) 424485
	henk dekker	(076) 713390
	henk blik	(050) 142097
eprom-programmer	henk dekker	(076) 713390
pascal	rob van albada	(020) 739671



contactpersonen regionale bijeenkomsten

groningen	evert drijver	(050) 778415
deventer	r e van gelden	(05720) 3016
utrecht	m s vreedenburg	(02159) 11068
zeeland	wim luijt	(01100) 28197
tilburg	jos hermans	(013) 552530
born	jos hermans	(013) 552530



REGIONALE BIJEENKOMSTEN IN HET ZUIDEN

Op dit moment draaien de regionale bijeenkomsten in het 'Limburgse' land al ruim 2 (= twee!) jaar. Elke maand (met uitzondering van de zomermaanden juli en augustus) hebben deze bijeenkomsten plaatsgevonden. De regionale bijeenkomsten in Tilburg draaien nu zowat driekwart jaar. Ik mag zeggen, dat deze bijeenkomsten redelijk tot druk bezocht worden. Dat er behoefte is aan zulke bijeenkomsten, leid ik af uit de afstanden, die toch steeds worden afgelegd door de bezoekers. Deze bedragen in het uiterste geval wel bijna honderd kilometer. Het bezoekersaantal ligt voor beide plaatsen gemiddeld rond de 15 personen.

Wat wordt er nu zoal gedaan op zo'n bijeenkomst?

- uitwisselen van de laatste nieuwtjes
- uitwisselen van de eigen (!) software
- Z80-assemblercursus
- hardware-uitbreidingen voor de NewBrain (onder andere printerinterface, joystickaansluiting en spraaksynthesizer)
- aansluitproblemen oplossen (modem, printer, etcetera)
- problemen op 'taalgebied' (o.a. basic, pascal, machinetaal en zelfs cobol)
- en alles wat nog meer met computers te maken kan hebben

Zoals reeds vermeld vinden beide bijeenkomsten maandelijks plaats (met uitzondering van de maanden juli en augustus). Per september 1985 worden de bijeenkomsten uitgebreid. Nu niet meer alleen NewBrain, maar OOK ANDERE CP/M-MACHINES. Het wordt dus steeds meer de moeite waard om eens naar het zuiden 'af te zakken' (dus niet alleen met carnaval!).

Waar en wanneer worden deze bijeenkomsten gehouden?

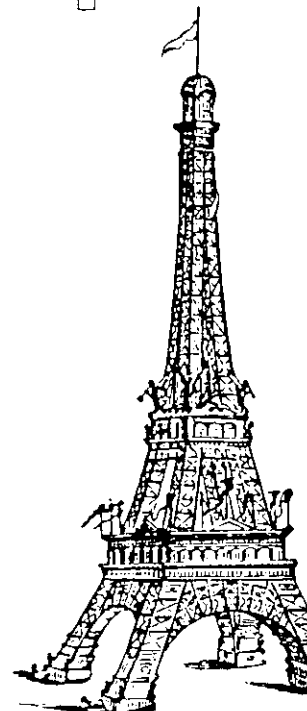
In Limburg: elke tweede of derde zondag van de maand, aanvang 10.00 uur, in Café Restaurant 'Marktzicht', Kerkstraat 33, Born. Voor het najaar van 1985 zijn bijeenkomsten gepland op 22 september, 20 oktober, 17 november en 15 december.

In Brabant: elke eerste dinsdag van de maand, aanvang 20.00 uur, in Wijkgebouw 'De Schans', De Schans 123, Tilburg-Noord. Geplande bijeenkomsten najaar 1985: 3 september, 1 oktober, 5 november en 3 december.

Mensen, die een regionale bijeenkomst willen bezoeken, kunnen bij twijfelgevallen in verband met feestdagen etc. contact opnemen met: Jos Hermans, Mozartlaan 523, 5011 SM Tilburg, telefoon (013) 552530 (maandag - vrijdag 18.00 - 20.00 uur). Tot ziens?

Jos Hermans

paris-micro



EEN RETOURTJE UTRECHT - PARIJS (VOOR EEN ENKELING EEN ENKELTJE)

Op zaterdag 1 juni is een delegatie van de NewBrain-gebruikersgroep naar Parijs getogen per auto om aldaar een clubbijeenkomst te bezoeken van de Franse gebruikersgroep. Wij zijn daarheen gegaan, vooral omdat de Franse gebruikersgroep (Paris-Micro), ondanks dat ze minder dan 150 leden telt, zeer actief is op met name het gebied van hardware.

We hadden gedurende de middag de gelegenheid om kennis te nemen van het een en ander bij Paris-Micro. De ochtend en avond was benodigd voor de reistijd. De reiskosten van circa f 200,- benzine, tol en parkeren zijn voor de helft vergoed uit de kas van de NewBrain-gg, terwijl het resterende bedrag opgebracht is door de vier mensen die naar Parijs zijn gegaan, te weten Rob van Albada, Henk Dekker, Wim Luijt en Rob Maris.

Helaas was de aankomst wat te vroeg, want de heren hadden juist pauze. Des morgens was er namelijk een ledenvergadering (in die zin van het woord, dat ze voor niet-leden ontoegankelijk zou zijn volgens de Franse wet!), met na de pauze een informeel gebeuren met onder andere demonstraties. We schreven al, dat de Fransen bijzonder actief zijn op het gebied van hardware, en dan ook in uitvoerende zin. Dit is hun voornamelijk mogelijk gemaakt, doordat ze relaties hebben waardoor ze goedkoop en zonder veel risico printen kunnen laten ontwerpen en maken. Laat ik eens een opsomming geven van wat ze zoal hebben:

- PC0. Een interfacekaart tussen de NewBrain en hun van Elektuur afgeleide busstructuur.

- PC1. Een EPROM-kaart voor 4 EPROMs in het 8000h-gebied, die softwarematig in- en uitschakelbaar zijn.
- PC7. Een 48 input/output-kaart (digitaal)
- PC14. Geluidsgenerator, stemsynthesizer (met MEA 8000 chip) en joysticks.
- PC18. RS232, Centronics, Mintel (modem) met 2 tot 8 k buffers.

Deze kaarten hebben wij in werkende staat gezien. Verder zijn in ontwikkeling een krachtige grafische kaart gebaseerd op de grafische processor EF 9367 en een disk controller voor de non-expanded NewBrain gebaseerd op de WD 2797 FDC chip (werkt heel leuk tot nu toe). Een indrukwekkend lijstje, dacht ik toch wel. Als bussysteem gebruiken zij zoals gezegd de Elektuurbus, met enige niet gebruikte aansluitingen toegepast voor specifieke NewBrain-signalen.

Een van de redenen van onze komst naar Parijs was, naast dat van nadere kennismaking (met tegeninformatie om op 24 augustus bij ons present te zijn), om te praten over de in ontwikkeling zijnde disk controller, met name interessant sinds Simac geen normaal geprijsde oplossing meer kan bieden. De hardwarewerkgroep was namelijk al een tijdje bezig met het voorbereiden van een goedkoop ontwerp - een dat bij voorkeur zonder expansion interface 64k op/m zou kunnen draaien. Echter, voordat tot publiciteit zou worden overgegaan, wilden we eerst zekerheid hebben over de realiseerbaarheid. Voordat het zover was, hoorden we op 16 februari, dat de Fransen met eenzelfde ontwikkeling al ver gevorderd waren. Reden om ons werk voorlopig te stoppen.

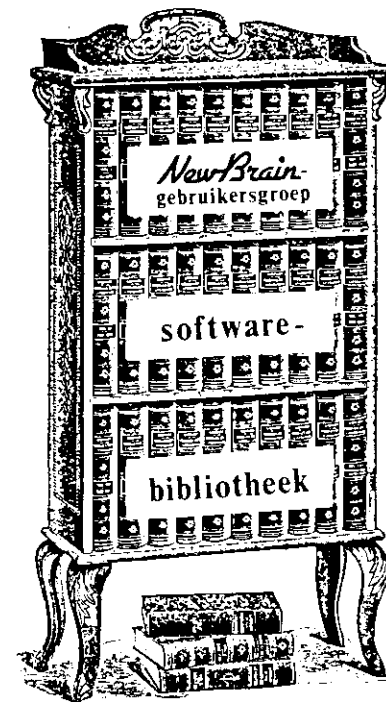
In Parijs konden we de disk controller-kaart zien, waarbij het in het stadium van ontwikkeling mogelijk was om sectoren te lezen en schrijven. Tevens is men bezig om BDOS geheel te herschrijven, zodat het efficiënter wordt. Er was echter een nadeel: bij diskoperaties valt het beeldscherm weg. Vanuit onze eigen ontwerpvoorbereiding hebben wij daarvoor een goedkope oplossing aangedragen, die hun ook zeer welkom was. Wij hopen, dat met deze bundeling van krachten een disk controller ontstaat, die naar tevredenheid zal werken tegen een kostprijs van minder dan f 400,-.

Voor het overige hebben we nuttige gesprekken gehad met diverse mensen. Een van hen kan goed converseren in het Engels, een bijzondere prestatie voor een Fransman. Tegen zes uur werd het helaas tijd om te vertrekken, behalve dan voor een van ons, die een dag Parijs te kort vond . . .

Rob Maris



vernieuwd



De softwarebibliotheek is gereorganiseerd. Een aantal programma's is verbeterd of voor diverse configuraties geschikt gemaakt; en de ontwikkelingen, die we in On-line 4 aangekondigden, zijn gerealiseerd. Tegelijk met deze On-line is een cassette verschenen met afgedrukte of besproken programma's. Ook de overige programma's worden voortaan standaard 'gebundeld' op een drager geleverd. Ze worden veelal gerubriceerd aangeboden in de vorm van cassettes of diskettes, al naar gelang uw voorkeur. De prijs voor deze software is een uniform bedrag van f 15,00 per cassette of per diskette; zijnde de kosten van de drager, van verzenden en een geringe vergoeding per programma.

Zo gaat het bestellen: vul uw keuze van de gewenste cassettes in op de bon achterin deze On-line. Kruis aan, of u cassettes of diskettes wenst (en het formaat van uw diskdrive: 200k, 400k ss, 400k ds of 800k). Stuur de bon samen met een girobetaalkaart of betaal- of eurocheque (vergeet niet het nummer van giro- of betaalpas in te vullen) op naar:

NewBrain-gebruikersgroep
Postbus 4494
1009 AL Amsterdam

Het verschuldigde bedrag kan ook voldaan worden door overmaking ten laste van uw post- of bankrekening naar:

postrekening 2700380
t.n.v. J. de Vries te Utrecht

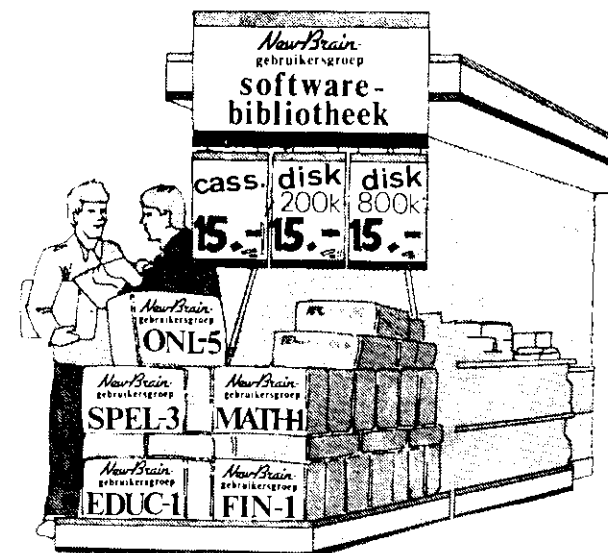
Indien u cheques bijsluit, en uw pasnummer op cheques en bestelformulier vermeldt, kan direct bij ontvangst van uw bestelling tot verzending worden overgegaan, als de gewenste cassettes / diskettes voorradig zijn. In andere gevallen kan dit met een vertraging van soms weken gepaard gaan.

De NewBrain-gebruikersgroep bemiddelt bij de verspreiding van 'public domain'-software, d.w.z. van programma's, waarop geen auteursrechten (meer) rusten.

De meeste programma's in de bibliotheek zijn zelfdocumenterend: ze bevatten een instructie en een overzicht van te gebruiken commando's. Sommige grotere programma's hebben een aparte documentatie nodig. Deze zijn of in de vorm van een apart programma (b.v. KASBOEK) of d.m.v. een beschrijving in On-line (b.v. KAARTENBAK, MICROCALC). In het eenvoudige geval wijst het zijn weg vanzelf.

De programma's zijn over het algemeen geschikt voor de UNEXPANDED systemen met cassette recorder. Gebruikers met diskdrives kunnen deze vaak ook direct gebruiken. In andere gevallen is aanpassing eenvoudig te verwezenlijken door device 12 te koppelen aan de input- en output-stream, i.p.v. device 1. Andersom geldt het zelfde: voor diskdrives geschreven programma's kunnen aangepast worden door device 12 in device 1 te veranderen, b.v.: OPEN OUT#1, 1, "SPEL.BAS" wordt voor drives OPEN OUT#1, 12, "SPEL.BAS"; en OPEN #1, 12, "ADRESSEN.DAT" wordt voor cassette OPEN #1, 1, "ADRESSEN.DAT".

Nog makkelijker wordt het door OPEN #1, PEEK (19), "ADRESSEN.DAT" te gebruiken. Dan gaat het altijd naar het aangesloten medium.



Voor EXPANDED systemen (96k) zijn de programma's UNEXPAND.BAS en UNPAGE.BAS aanwezig op de cassette HULP-1, om softwarematig het systeem om te zetten in een unexpanded systeem, resp. zonder en met diskdrives.

Diskettes die geleverd worden, zijn CP/M compatibel. Vanwege eventuele auteursrechten op CP/M moet u er zelf m.b.v. SYSGEN de systeemtracks opzetten.

Zoals is aangegeven zijn alle formaten nu verkrijgbaar: 200k, 400k ss, 400k ds en 800k. Vermeld duidelijk bij uw bestelling, welk formaat gewenst is; reeds uitgevoerde bestellingen en leveranties worden niet teruggenomen.

De NewBrain-gebruikersgroep streeft er naar, om goede programma's in de bibliotheek op te nemen. Getracht wordt, om zoveel mogelijk aan te geven waarop het programma draait zonder verdere aanpassing door de gebruiker in het programma zelf. Helaas is dit nog niet bij alle programma's het geval. Mocht u programma's tegenkomen met fouten, dan stellen wij het op prijs om dit van u te vernemen om voor verbetering zorg te kunnen dragen.

INZENDEN VAN EIGEN SOFTWARE

De gebruikersgroep heeft sinds zijn oprichting al diverse programma's op kunnen nemen in haar bibliotheek. Toch blijft er behoefte aan nieuwe programma's. Let echter bij het opsturen wel op de volgende puntjes:

- probeer zo origineel mogelijke programma's op te sturen
- vermeld in het programma, met behulp van REM-statements, duidelijk wat het programma doet, hoe het werkt, en wie het geschreven heeft door naam, adres etc. te vermelden en een eventuele bronvermelding
- probeer de programma's compatibel te houden voor zoveel mogelijk NewBrain-configuraties. Wees dus voorzichtig met PEEK- en POKE-opdrachten (zie ook het artikel op pagina 33 van deze On-line)
- stuur uw programma's, eventueel vergezeld van commentaar, dat in On-line geplaatst kan worden, tesamen op diskette of cassette naar:

NewBrain-gebruikersgroep
Postbus 4494
1009 AL Amsterdam

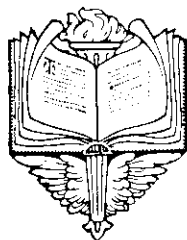
- na ontvangst van de programma's wordt, in overleg met de redactie van On-line, bekeken, of opname zal plaatsvinden. Is dit het geval, dan worden ze gehonoreerd met software uit de bibliotheek.
- om de gebruikersgroep de zekerheid te geven, dat de programma's van eigen makelij zijn, wordt tevens per ingezonden programma een getekende verklaring gevraagd. De programma's blijven eigendom van de maker, de gebruikersgroep is tevreden met de toestemming om ze uit te leveren.

Mocht blijken, dat er programma's in de bibliotheek voorkomen, waarbij de copyrights niet vrij zijn voor de gebruikersgroep, dan zal met onmiddellijke ingang de uitlevering van dat programma worden gestaakt.

Het ligt in de bedoeling, om nieuwe programma's, gecombineerd met praktische listings uit On-line tegelijk met die On-line uit te brengen, waardoor het principe van vaste rubrieken niet altijd gehandhaafd kan worden. Daarentegen zal elke volgende cassette geheel nieuwe software bevatten. Zie bv. nu de programma's van cassette 'ONLINE-5'.

Wij wensen u nog veel genoeglijke uurtjes met uw NewBrain.

Jan Tolkamp



bestelnummer: EDUC-1

- vierdelige MUZIEKCURSUS met een demoprogramma
- tweedelige TYPECURSUS

bestelnummer: EDUC-2

- vierdelige cursus NEWBRAIN-BASIC
- driedelige 'GRAFISCHE' CURSUS: leer omgaan met het graphics device van de newbrain

bestelnummer: HULP-1

- PROTO voor het ontwikkelen van basicprogramma's (hernummeren, opzoeken, wijzigen). zie voor uitleg on-line 3, pag. 61
- MONITOR voor het ontwikkelen van machinetaalprogramma's (inhoud van registers en Z80-adressen)
- CONVERSI: omzetten van getallen van en naar hexadecimaal, decimaal en binair
- BASICODE2-leesprogramma
- BASICODE2-subroutines
- BASICODE2-schrijfprogramma (on-line 2, pag. 50 vv., on-line 3, pag. 70, en on-line 4, pag. 61 vv.)
- UNEXPANDED maakt van een 96k-machine een 32k-machine
- UNPAGE maakt van een 96k met drive(s) een 32k met drive(s)
- COPY D/C: kopiëren van disk naar cassette
- COPY C/C: kopiëren van cassette 1 naar cassette 2 in diverse formaten
- TAPE-UTILITY, hulpmiddel bij het kopiëren

bestelnummer: GRAFI-1

- HISTOGRAMMEN maakt kolomendiagrammen en kleurt ze in
- 3D tekent driedimensionale afbeeldingen
- LICHTKRANT geeft ingevoerde tekst vergroot weer
- PLOTGEN zet je tekening om in plotopdrachten
- GLOBE tekent aardbol vanuit de opgegeven coördinaten
- EUROPA tekent alle landen en hoofdsteden van europa
- MARGARET tekent een gestileerde bloem
- RANDCITY tekent een stadsgezicht in perspectief
- BIRD, meetkundige tekening van een sierlijke vogel
- FIETSKE tekent de bekende tweewieler

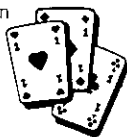


bestelnummer: FIN-1

- BOEKHOUDPROGRAMMA: per run 50 rekeningen en 150 mutaties
- AUTOKOSTEN berekent exact de jaarlijkse kosten (scherm of print)
- FINFORM, tiental berekeningen voor annuïteit, afschrijving etc
- DEBITEUREN-CREDITEURENBOEK, met save-, laad- en printerroutines
- KASBOEK 1 en 2: 15 rubrieken, 50 boekingen per maand (geen btw). zie voor nadere beschrijving on-line 2, pag. 60
- BENZINE.BAS houdt per kenteken het verbruik en kosten bij. voor cassette-, diskette-, expanded- en unexpanded-systemen, met scherm- en printerroutines

bestelnummer: SPEL-1

- SPEURPUZZEL; los hiermee de wekelijkse radiogids puzzel op
- KANON: putje schieten
- METEOR, een behendigheidsspel; een vluggertje
- BIORITME: zie in een grafiek je beste (en slechtste) dagen
- BLACK-JACK, eenentwintigen tegen de computer
- SLANG: eet de pillen voor punten, maar pas op je staart
- PATIENCE, een geduldig kaartspelletje
- PSYCHIATER, een rechtlijnige neef van ELIZA
- KERNSIMULATIE in een kerncentrale (welke factor heeft welke invloed; ook grafisch)
- 4 OP EEN RIJ en VIEROPRIJ: probeer je tegenstander / de computer af te troeven



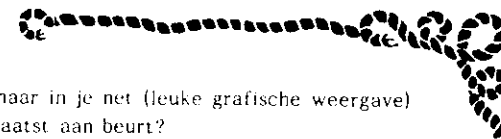
bestelnummer: SPEL-2

- BOWLING, met baan en automatische wedstrijdformulieren
- HOUTHAKKER: spaar de bomen op zoek naar goudklompen
- RUPS; anders dan SLANG van cassette SPEL-1
- STARTREK 1 en 2: de 'enterprise' in gevecht met de klingeons
- ANTHILL: verzamel de miere-eieren, maar pas op voor de mieren
- FLIPPEREN, een gokkastspel
- GALGJE: vind het woord, anders volgt de strop
- OTHELLO, een bordspel met denkwerk
- ABC: zet de letters goed; een schuifspelletje



bestelnummer SPEL-3

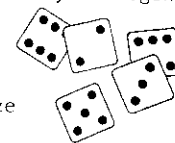
- NEWWORM: stuur de gladde aal naar zijn prooi
- ROULETTE, het casino thuis
- MAANLANDER: je space-shuttle dreigt op aarde neer te storten
- MEMORY: draai de kaarten om (nieuwe grafische versie)



- ZEEMEERMIN: vang haar in je net (leuke grafische weergave)
- TOT 100: wie is het laatst aan beurt?
- BLACKBOX: vind atomen die in een veld verborgen liggen: uit de richtingverandering is de positie te achterhalen
- BEURSPLEIN: probeer je kapitaal met effecten te vergroten
- GOKKAST: een vierwielige en eenarmige bandiet

bestelnummer: SPEL-4

- JACKPOT, een gokkast om arm (of rijk) van te worden
- BOMBER: probeer de obstakels te vernietigen, voordat je er tegen te pletter vliegt
- BRAINMAN, een snelle happer in machinetaal
- MASTERMIND: ontdek de verborgen code
- YATHZEE: gooi de 5 dobbelstenen, en maak je keuze
- BACK-GAMMON, een bordspel
- HANOI: 6- en 7-ringsspel, inclusief een demonstratie, verleg de schijven in zo weinig mogelijk beurten
- AWARI, denkspel waar go-spelers vast heel goed in zullen zijn



bestelnummer: SPEL-5

- DOOLHOF: probeer de uitgang te vinden
- REVERSE: zet 9 cijfers door omkeren in de juiste volgorde
- TIC TAC TOE, ofwel boter-kaas-en-eieren
- CONTEST maakt gebruik van de ingebouwde klok, houdt tussentijden, ronden en eindtijden bij
- SCHAATSWEDSTRIJDEN helpt je tijdens wedstrijden de classificering te bepalen
- BALSPEL: een moeilijk te vangen bal wordt grafisch beloond
- STERRENSLAG: probeer de sterren te vernietigen (machinetaal)
- KALENDER berekent precies het aantal dagen, weeknummer en -dag, leeftijd etc. met verschillende jaartellingen

bestelnummer: MATH-1

- FACULTEIT berekent exact 250! en misschien wel meer
- BOOLEAN FUNCTIONS
- N-BODY PROBLEM
- CURVE-FITTING
- RISOLIN: oplossing van stelsels van lineaire vergelijkingen met behulp van gauss-eliminatie

bestelnummer: VIDITEL-1

voorlopig alleen voor EXPANDED newbrain

- VIDIINFO.BAS legt de werking uit
- VIDITEL.BAS het eigenlijke programma
- VIDIEX.COM machinetaal sub-programma
- VIDIHLP.BAS om reeds opgeslagen beelden te lezen en eventueel te wijzigen
- HCC.VDI, VSG1.VDI en NBGG.VDI zijn voorbeelden



bestelnummer: DIVERS-1

- KAARTENBAK 1 en 2
- EDITOR, voor de handleiding zie on-line 2, pag. 65
- TXY-I alle gebruikelijke functies van een tekstverwerker op 5 royale pagina's, aan te passen voor vrijwel elke type printer, hulpmenu bij stoptoets (na edit)
- SPRITES: het maken van 'sprites' op de newbrain
- SPRITES-DEMO: als model in een spin weergegeven, zie het artikel in on-line 4, pag. 47
- LEVEN, een model van de explosieve vermenigvuldiging met een sub-programma in machinetaal: LEVEN-BIN

bestelnummer: ONLINE-5

de programma's uit on-line 5 op cassette:

- STROMING geeft de vloeistofstroomlijn weer rond ellipsen, een programma van a. van de velde, beschrijving en afbeelding op pag. 38
- MICROCAL, een spreadsheet-programma door john van der ploeg, SCHILDOP.CAL, DEUR.DAT en OPTIE zijn voorbeeldbestanden, de handleiding is te vinden op de pagina's 39-44
- DIVERSE HULPROUTINES, een soort SUBMIT in basic; van patrick vercammen: OM, O-1 t/m O-7, de werking wordt uitgelegd op pag. 45-48
- SCREENDUMP voor star dp-510 door h. balemans, zie het artikel en de listing op pag. 61-64
- DOORTJE STOOT DANST door de toevoeging door arno roelofs, dit is het volledige programma, op pag. 50-52 staan alleen de aanvullingen op het artikel in hcn 63
- MOLENS: 4 roulerende schermen laten de molens draaien, van arno roelofs, zie de pagina's 49 en 53 voor het principe en de listing
- KLOK: breid je unexpanded newbrain uit met een extra device, de machinetaalroutine uit het artikel van cor boer op pag. 71-80



NewBrain-
gebruikersgroep

onderwijsbibliotheek

de actiefste werkgroep binnen de newbrain-gebruikersgroep is de werkgroep onderwijs, een belangrijk resultaat van haar werkzaamheden is de uitgebreide bibliotheek van onderwijssoftware, waaraan door vele scholen en docenten bijdragen geleverd zijn, zoals u al in hcnnieuwsbrief 71 hebt kunnen lezen, wordt daar nu wat ruimere bekendheid aan gegeven.

de schijven met software hebben de praktische namen schijf 1, schijf 2, schijf 3, schijf 4 en schijf 5, ze worden geleverd in het 200k-formaat, de programma's staan ook op cassettes, dan te bestellen als cassette 1, cassette 2 enz.

voor alle inlichtingen over het bestellen van de schijven, over de software, die erop staat, en alle overige activiteiten van de werkgroep onderwijs kunt u terecht bij de werkgroep onderwijs, in de persoon van

pieter van dijk, telefoon (01751) 12367

hierna volgt een korte beschrijving van de onderwijssoftware op de diverse schijven.

de redactie

catalogus schijf 1

1. CIJFERS1: het bijhouden van een cijferadministratie: lijst laden, save, afdrukken, printen; nieuwe lijst; kolom definiëren, vullen, wissen.
2. GRAFIEK1: het tekenen van grafieken van functies, 8 functies voor-geprogrammeerd.
3. 3DGRAF2: het tekenen van grafieken van functies met 2 variabelen in stereometrische projectie. De grondverzameling is $\{(x, y) \in \mathbb{R} \times \mathbb{R} \mid 0 < x < a \wedge 0 < y < a\}$. De waarde van a kan de gebruiker zelf bepalen. 9 functies

voorgeprogrammeerd.

4. KROMMEN1: het tekenen van grafieken. De opties zijn:

$x = \cos(t + \pi / k)$, $y = \sin(2 * t)$

$x = t * \log(t)$, $y = 2 + \log(t)$

$x = \log(\text{abs}(t)) - 1 / t$, $y = 4 * t$

$x = 2 * t / (t * t - 4)$, $y = \text{sqr}(t) - 3$

5. INTEGR1: het tekenen van integraalkrommen m.b.v. Kutta's Simpson-regel. 9 mogelijkheden.

6. TAFELS1: de leerling geeft op, welke tafel hij wil oefenen. De computer geeft willekeurig opgaven.

7. DICTEE1: de computer zet woorden op het scherm. De leerling moet ze natypen, nadat ze van het scherm zijn verdwenen.

8. HOOFDST1: meerkeuzevragen om de hoofdsteden van Europese landen te overhoren.

9. DICTEE2: de leerling moet een serie woorden natypen. Als alle woorden geweest zijn, komen de woorden nog eens, maar dan heel kort, op het scherm. De leerling moet de woorden weer natypen. Bij een fout wordt het woord nogmaals, iets langer, op het scherm getoond.

10. PROCENT1: procentsommen oefenen.

11. LOONBER1: loonberekening. Ingevoerd moeten worden: bruto loon, percentage pensioenpremie, aow, belastinggroep, percentage sociale lasten, contributie en onkostenvergoeding.

12. DEBCRED1: debiteuren- / crediteurenadministratie. Facturen en creditfacturen maken en inboeken; betalingen verwerken; lijsten maken.

13. RUTHERF1: proef van Rutherford. De opties zijn:

1. de baan van een alfa-deeltje, dat op een kern geschoten wordt, wordt getekend, waarbij de kern, de energie en beginpositie van het alfa-deeltje gekozen worden.

2. idem, maar nu wordt de beginpositie van het alfa-deeltje door de computer bepaald en herhaaldelijk getekend.

3. een hoekverdeling van de alfa-deeltjes bij verstrooiing aan de kern.

4. een listing van de baanberekening.

14. GOLVEN1: weerkaatsing van een lopende transversale golf. Door de leerling kan gekozen worden uit: vast of vrij uiteinde; hele golf of alleen golfberg; alleen staande golf of volledig program; inkleuren of niet; langzaam of snel; aantal halve golven.

15. DOORSNEE: doorsnee van vlak en kubus. Eerst wordt een kubus getekend. Daarna kunnen coördinaten worden opgegeven van lijnen, die vervolgens getekend worden.

16. TAPECOPY: kopieert een programma van tape naar tape.

17. VERMVL1: 10 vermenigvuldigopgaven. De getallen worden onder elkaar gezet.

18. INVULDT1: 10 zinnestelsels waar d, dd, dt, t of tt moet worden ingevuld.

19. REKEN1: 20 hoofdrekenopgaven: optellen, aftrekken, vermenigvuldigen, delen, combinatie optellen-aftrekken, combinatie vermenigvuldigen-delen, combinatie van alle mogelijkheden.

20. REKEN2: 4 x 10 rekenopgaven in de vorm: $76 + \dots = 81$, $65 : \dots = 13$.

21. MULTCH1: algemeen meerkeuzevragenprogramma om opgaven in te voeren, te saven naar cassette, te laden van cassette en te laten beantwoorden.

22. CIJFERS2: cijferverwerking per klas. Namen van leerlingen en cijfers per klas invoeren, controleren en veranderen; cijfers en gemiddelde raadplegen; alfabetisch sorteren; leerlingenlijst (klas + naam).

23. GRAFIEK2: grafieken van lineaire vergelijkingen. Er kan gekozen worden uit $y = ax$ en $y = ax + b$.

24. GRAFIEK3: grafieken van goniometrische functies.

25. TEKEN1: hi-res tekenprogramma met elementen van LOGO. Opdrachten kunnen van tape geladen worden.

26. SNELLEES: van een woord wordt iedere letter even getoond. Daarna moet de leerling het woord natypen. De tijdsduur waarin de letters zichtbaar zijn, wordt steeds korter. Om het moeilijker te maken kan het woord ook achterstevoren getoond worden.

27. PARAB1: het tekenen van parabolen. 6 voorgeprogrammeerde functies.

28. MEMORY1: op het scherm komen 20 vakjes waarachter 2 x 10 woorden verborgen zitten. De leerling moet raden welke 2 vakjes bij elkaar horen.

29. BEROEP1: de leerling moet de lettergrepen van een aantal beroepen bij elkaar zoeken.

30. WRDBLD1: woorden verschijnen gedurende korte tijd op het scherm en moeten daarna uit het hoofd worden nagetypt.

schijf 2

31. REKEN3: optel- en aftreksonnetjes. De sommen worden makkelijker resp. moeilijker afhankelijk van de resultaten.

32. VOLUIT: getallen tussen 1 en 999 kunnen worden ingetypt, waarna de computer ze voluit schrijft.

33, 34. KALEND1, KALEND2: deze programma's, die een jaarkalender uitprinten (tussen 1900 en 1999), zijn vrijwel identiek. Programma 33 drukt 12 losse maandblaadjes, programma 34 drukt alle maanden direkt onder elkaar.

35. SORT1: van 10 getallen wordt het grootste opgezocht.

36. SORT2: van 10 getallen wordt het kleinste opgezocht.

37. SORT3: 10 getallen worden gesorteerd.

38. SORT4: 10 getallen worden gesorteerd, waarbij het sorteerproces op het scherm te volgen is.

39. SORT5: 10 door de leerling in te voeren namen worden gesorteerd, waarbij het sorteerproces op het scherm te volgen is.

De programma's 35 t/m 39 vormen samen een inleiding op het sorteren en daarop volgende lessen over bestanden.

40. SPAARB1: alle leerlingen uit een klas kunnen een bankrekening openen. Zij kunnen daarna geld naar elkaar overmaken: rekeningen openen en opheffen, cheques en overboekingsformulieren drukken, boeken, afrekeningen drukken, totaaloverzicht.

41. SPOORW1: een lijst van 140 plaatsnamen wordt geprint.

42. SPOOR2: nabootsing van de computer die de loketbediende van de NS gebruikt om treinbiljetten te drukken. De leerlingen moeten van de 140 plaatsen uit programma 41 de afstanden berekenen (m.b.v. een atlas) en in de computer invoeren, de tarieven invoeren en kunnen dan biljetten afdrukken.

43. TEKEN2: met behulp van de cursortoetsen kan een tekening op het scherm worden gemaakt.

44. VOEDING1: op het scherm verschijnen achtereenvolgens 10 tabellen met voedingsmiddelen. De leerling kiest daaruit wat hij gedurende een dag wil eten. Aan het eind vertelt de computer de hoeveelheid kcal, kJ, eiwitten, vetten en koolhydraten en tevens beoordeelt hij het gekozen.

45. CIJFERS3: berekenen van cijfergemiddelden.

46. REKEN4: staartdelingen met rest.

47. ZKWOORD1: in een vierkant van letters zit een woord verstopt.

48. BOTERKA1: programma om 25 vragen en antwoorden in te voeren t.b.v. programma 49.

49. BOTERKA2: het spel "boter, kaas en eieren". Om een kruisje of rondje te kunnen zetten moet eerst een vraag goed beantwoord worden.

50. CHEMIE0: geeft informatie over de programma's 51 t/m 56.

51. CHEMIE1: oefenen van molberekeningen in 5 fasen: kloppend maken van de vergelijking; berekenen van de molverhouding; aantal mol gegeven stof berekenen; aantal mol gevraagde stof berekenen; omrekenen van mol naar gram.

52. CHEMIE2: demonstratieprogramma over de vorm van het ionrooster en het kristalrooster en de consequenties daarvan voor de geleidbaarheid en de vervormbaarheid. Zonder tekst.

53. CHEMIE3: hetzelfde programma als 52, echter opgezet als eeuwigdurend demonstratieprogramma (voor een open avond).

54. CHEMIE4: demonstratieprogramma om de begrippen 'het evenwicht ligt rechts', 'links' of 'verschuift' toe te lichten.

55. CHEMIE5: grafische weergave van wat er gebeurt met snelheden en concentraties bij omkeerbare reacties.

56. CHEMIE6: het bepalen van vriespuntsdalingen en kookpuntsverhogingen. De beschikbare stoffen zijn water, fenol, glucose en alcohol.

57. ENGELS1: overhoring van de Engelse onregelmatige werkwoorden. Gevraagd worden de tegenwoordige tijd, verleden tijd en voltooid deelwoord.

58. ENGELS2: oefenen in het maken van vragende en ontkennende zinnen in het Engels. Gekozen kan worden uit: zinnen met een hulpwerkwoord; zinnen zonder hulpwerkwoord (tegenwoordige tijd of verleden tijd); gemengde zinnen.

59. VALUTA1: het oefenen in omrekenen van vreemd geld in Nederlands geld.

60. VERZEKE1: opgaven in het oefenen met verzekeringen. Gekozen kan worden uit: berekenen van premie; berekenen van uitkering in geval van schade; autoverzekering.

61. REGSIM: procesregelaar-simulatie. Opgegeven kunnen worden de PB van de regelaar (1 - 500 %) en de regelaktie (P of PI).

schijf 3

63. STATIST1: statistiekprogramma. grafiek, spreiding, trend line analyse.

64. CONVERS1: conversieprogramma voor getalstelsels, van en naar decimaal, hexadecimaal en binair.

65. GLOBE1: hi-res-programma dat een globe tekent vanuit elke verschillende hoek. Opgegeven kunnen worden de as-stand, lengte- en breedtegraad en de grootte van het raster (in graden)

66. RISOLIN: oplossing van lineaire systemen met behulp van Gauss-eliminatie.
67. GRAFIEK4: grafieken van goniometrische functies. 10 stuks voorgeprogrammeerd.
68. GRAFIEK5: het tekenen van grafieken van functies.
69. GRAFIEK6: oefenprogramma voor het berekenen en tekenen van parabolen. 7 functies voorgeprogrammeerd.
70. GRAFIEK7: drie driedimensionale tekeningen: kubus, kromme op een cilinder en kromme op een ellipsvormig lichaam.
71. GRAFIEK8: grafieken van functies.
72. LOG1: eenvoudige en moeilijke logaritme-opgaven. Voorbeelden:
a) gemakkelijk: $6\log(1/6) = \dots$
b) moeilijk: $7\log(713) = \dots$
73. VERMVL1: vermenigvuldigen van tweetermen, bv. $(x - 3) * (x + 2) = \dots$
74. LENZEN1: lenzen en hun beeldvorming. Beeldconstructie bij lenzen, lensformules en lineaire vergroting. Een hi-res reken- en tekenprogramma.
75. SCHAKEL1: berekent de vervangingsweerstand bij serie- of parallel-schakeling van weerstanden.
76. SCHAKEL2: aan de hand van tekeningen worden vragen gesteld over schakelingen. Voorbeeld: $I = 1 \text{ A}$, $P = 4 \text{ W}$. Wat is de spanning van de batterij?
77. ENGELS3: Engelse grammatica-oefeningen. Voorbeeld: Somebody (see) . . . a UFO yesterday. Maak een keuze uit: 1. was seeing, 2. saw, 3. has seen.
78. ENGELS4: Engelse invuloefening. Bij ieder goed antwoord wordt de leraar een stukje verder opgehangen.
79. ENGELS5: overhoren van de Engelse onregelmatige werkwoorden.
80. ECON0: docentenversie grafiek Keynes. Het model ziet er zo uit (we laten de overheid en het buitenland buiten beschouwing): $Y = C + I$, $C = cY + C_0$, $I = I_0$, (Y = nationaal inkomen, c = marginale consumptiequote, c_0 = autonome consumptiequote, $I = I_0$ = autonome investeringen).
81. ECON1: macro-economisch model van Keynes, leerlingenversie. Het programma lost 10 keer het model op. Bij het begin zijn alle autonome waarden 111. Na de eerste oplossing kan 1 variabele veranderd worden.
82. CHEMIE7: uitleg over het kloppend maken van reactievergelijkingen tussen de metalen Mg, Al, Pb, Fe, Ca en Na en de niet-metalen F, Cl, O en S.
83. CHEMIE8: berekenen van de massaverhoudingen aan de hand van

reacties tussen zwavel en ijzer, koolstof en zuurstof, stikstof en zuurstof.

84. CHEMIE9: omzetten van naam naar formule en vice versa.

schijf 4

85. CHEMIE10: uitleg en eenvoudige opgaven over pH-berekening. Rekenmachine en BINAS-boekje zijn bij dit programma nodig.
86. CHEMIE11: dezelfde opzet als programma 86. Moeilijker opgaven.
87. BEHANG: de leerlingen ontwerpen een eenvoudig figuurtje. Door herhaling van dit ontwerp ontstaat een ritmisch patroon, dat de computer op het scherm afdruckt.
88. ABC: binnen een schuifraampje moeten de leerlingen letters gaan schuiven, totdat alle letters in volgorde staan. Naar keuze 16 of 25 vakjes.
89. GALGJE: de ene leerling voert een woord in, de andere leerling moet dat woord raden. Bij iedere foute letter wordt de galg een stukje verder getekend.
90. JAPLOG: een variant op "De toren van Hanoi". De leerlingen moeten cijfers verplaatsen, totdat alle cijfers weer in volgorde staan. Verschillende moeilijkheidsgraden.
91. MASTERMI: mastermind. De computer neemt een willekeurig getal van 4 verschillende cijfers. De leerling moet het getal raden.
92. MEMORY: 26 kaartjes kunnen 2 aan 2 worden omgedraaid. Als ze hetzelfde plaatje bevatten, mag de speler ze 'houden'. Aanleren van het alfabet en geheugentraining (het programma maakt gebruik van een gewijzigde karakterset).
93. SLEUTELS: de leerling krijgt een sleutelbos en moet daar achter-eenvolgens een aantal kamers mee openen. Als dit binnen het vastgestelde aantal pogingen lukt, wacht een geweldige prijs!
94. TOT100: variant van het Nimm-spel. De computer en de leerling gaan om de beurt een getal bijtellen, te beginnen bij nul. Wie het eerst bij 100 komt, heeft gewonnen.
95. VERGEL1: oplossen van eerstegraads vergelijkingen met x , in opeenvolgende stappen. Voorbeeld: $3 * x + 4 = 16 - x$.
96. GRAFIEK8: het tekenen van grafieken van functies. Modulair geprogrammeerd.
97. GRAFIEK9: het tekenen van krommen m.b.v. parametervoorstelling. Modulair geprogrammeerd.

98. KEYBOARD: maakt de leerlingen vertrouwd met het toetsenbord van de NewBrain, in het bijzonder met de speciale toetsen zoals STOP, CONTROL, HOME enz.

99. SCHAKEL3: opgaven over de wet van Ohm. Berekenen van het elektrisch vermogen en het rendement (eenvoudig). Met extra training van het gebruik van de juiste dimensies. Na 3 fouten volgt de uitwerking van de opgave met aansluitend eenzelfde soort opgave.

100. SCHAKEL4: als programma 99. Bij alle opgaven staan de gegevens in het schema aangegeven. De schema's zijn volgens het enkelleidingsysteem (motor-voertuigentechniek).

101. SCHAKEL5: berekeningen met de oude eenheid Ohm/m/mm². Vergelijken leidingweerstand bij toe- of afname van de doorsnede of de lengte. Spanningsverlies in de leidingen.

102. SCHAKEL6: extra opgaven met serie- en parallelschakeling met uitsluitend gelijke weerstanden. Nu echter zonder schema's.

103. CIJFERS4: na invoering van de meerkeuze-antwoorden volgen naar wens de leerlingresultaten of de foutenanalyse op scherm of printer (IDS). Zeer geschikt als meerkeuzevragen met open vragen worden gecombineerd. Beide soorten vragen kunnen t.o.v. elkaar 'gewogen' worden bij de bepaling van het eindcijfer.

104. LES1: eerste les met de computer. De computer berekent de geboortedatum van de leerling en zijn bioritme. Een getal raden (hoger ... lager) en een paar sommetjes.

105. REDDING: er bevindt zich een drenkeling in zee. De zee is 30 x 30 mijl. Red de drenkeling door zijn positie te bepalen. Gebruik van passer en rekenmachine.

106. ENGELS6: oefeningen in het gebruik van some en any. Eerst komen 20 vragen om het niveau te bepalen (A, B of C), daarna volgen 20 oefeningen op het bepaalde niveau.

schijf 5

107. DOBBEL1: aan de hand van een dobbelsteen worden de begrippen 'gemiddelde', 'modus' en 'mediaan' aanschouwelijk gemaakt.

108. REKEN5: optellen en aftrekken van negatieve getallen. De computer geeft uitleg met behulp van de getallenlijn. De leerling moet zelf de opgaven uitrekenen.

109, 110, 111. BENOEM1, BENOEM2, BENOEM3: benoemen in een zin van: werkwoord, zelfst. naamwoord, bijv. naamwoord, pers. voornaamwoord, lidwoord, bijwoord, aanw. voornaamwoord, bez. voornaamwoord en voorzetsel.

112. ONTL1: benoemen in een zin van: onderwerp, pers. voornaamwoord en gezegde, meew. voorwerp, lijd. voorwerp en bijw. bepaling.

113, 114. ONTL2, ONTL3: benoemen in een zin van: onderwerp, werkw. gezegde, meew. voorwerp, lijd. voorwerp, bijw. bepaling, bijv. bepaling, naamw. deel van het naamw. gezegde en werkw. deel van het naamw. gezegde.

115. WERKW1: overhoren van een aantal werkwoorden. Voorbeeld: wat is de verleden tijd (enkelvoud / meervoud) van zweren? Wat is het voltooid deelwoord?

116. BREUKEN1: uitgebreid programma dat helpt bij het maken van sommen over breuken. De leerling moet eerst zelf de som maken, daarna laat de computer de uitwerking zien. Keuze uit: optellen, aftrekken, vermenigvuldigen, delen en willekeurig. Leuk geprogrammeerd, met vergrote cijfers en tekens.

117. MASTERM2: het raden van een getal van 4 cijfers.

118. ENQUETE: de leerlingen stellen een enquête op. Na invoering van de vragen drukt de printer vragenformulieren en antwoordbladen. Het is de bedoeling, dat de leerlingen de enquête ook daadwerkelijk uitvoeren. Nadat de gegeven antwoorden zijn ingevoerd, produceert de computer een totaaloverzicht.

119. WOORDR1: de ene leerling typt een woord in. De computer zet de letters van dat woord in een willekeurige volgorde en de andere leerling moet proberen het woord te raden.

120. WOORDR2: variant op programma 119. In DATA-regels staan 50 beroepen. De computer gooit de letters van het beroep door elkaar, waarna de leerling moet proberen het beroep te raden. Opdat meer leerlingen tegelijk met het programma kunnen werken, komen de beroepen in een willekeurige volgorde. Uiteraard kunnen in de DATA-regels ook andere woorden worden geplaatst.

121. MULTCH2: de leerlingen stellen zelf een meerkeuzetoets op van 15 vragen met elk 4 antwoorden. Vragen en antwoorden worden in DATA-regels opgeslagen. Als de toets klaar is, kunnen andere leerlingen de toets afleggen. De resultaten worden eveneens in DATA-regels gezet.

122. BITTEST: gewijzigde versie van de BIT-test van het 2^e Teachip-bandje. Omdat de oorspronkelijke versie in Basicode geschreven is, werkte het programma erg traag. De gewijzigde versie werkt snel.

123. BIT/DATA: DATA voor programma 122.

124. KEGELSN: tekenen van de snijkromme, die ontstaat bij het snijden van een kegel met een plat vlak. De algemene vergelijking is: $ax^2 + by^2 + cx + dy + e = 0$. De leerling voert de waarden in voor a, b, c, d en e, waarna de grafiek getekend wordt. De computer vertelt ook, welk type doorsnede ontstaat, zoals lege grafiek, 1 punt, horizontale lijn, dalparabool enz.

125. CIJFERS5: hulp bij het berekenen van schoolonderzoekcijfers. De computer vraagt eerst naar de verdeling over de diverse onderdelen en naar een ev. bonus. Daarna kunnen de resultaten per onderdeel worden ingevoerd en geeft de computer het eindcijfer, afgerond tot op 1 decimaal.

126. SCHAKEL7: oefenen met de gevolgen van de inwendige weerstanden bij het leveren van een stroom en bij het opladen van een accu. Ook de gevolgen van de tegen-EMK (tegen-bronspanning) bij het leveren van stroom aan een elektromotor komen aan de orde. Berekeningen over de accu als stroombron; berekeningen betreffende het opladen van een accu; berekeningen over accu + elektromotor.

127. GALTON1: demonstratie van het bord van Galton. Het aantal balletjes en het aantal regels kan de leerling opgeven.

128. WISK1: een raamprogramma, bedoeld voor alle mogelijke oefeningen, waarbij een vorm herleid moet worden. Voorbeelden: $(x - 3) * (x + 2) = \dots$; $-4 + 6 - 7 * 8 = \dots$; $\cos(2 * x) = \dots$

129. CHEMIE12: meerkeuzetoets voor klas 3 havo/vwo over roosters. De toets bestaat uit 4 onderdelen: metalen, niet-metalen, ionbinding en gemengde vragen.

130. CHEMIE13: 10 titratie-opgaven van zuur/base-reacties en redox-reacties. Voor het met succes oplossen van de vraagstukken dient de leerling de principes van het titreren te begrijpen.

131. CHEMIE14: 10 opgaven voor het maken van chemische berekeningen, van eenvoudig tot moeilijk. Voorbeelden:

opgave 1: hoeveel gram H_2SO_4 is nodig om 20,5 g Fe volledig om te zetten? Hoeveel liter H_2 (1 atm en 273 K) ontstaat hierbij?

opgave 9: bereken hoeveel ml verdund salpeterzuur (s.m. 1,12 g/ml, 30 massa %) er nodig is voor de neutralisatie van 25 ml kalkwater (s.m. 1,05 g/l, 10 massa % $Ca(OH)_2$). Bereken ook hoeveel gram $Ca(NO_3)_2$ zou ontstaan na indampen.



NewBrain-
gebruikersgroep

cp/m-bibliotheek

Bestellen door een bestelformulier (achter in deze On-line) in te vullen en dit samen met een girobetaalkaart of betaal- of eurocheque (vergeet niet het nummer van de giropas in te vullen) op te sturen aan:

NewBrain-gebruikersgroep
Postbus 4494
1009 AL Amsterdam

Het verschuldigde bedrag kan ook voldaan worden door overschrijving op:

postrekening 2700380
t.n.v. J. de Vries, te Utrecht

PER 24 AUGUSTUS 1985 NIEUWE PRIJZEN

Prijs per volume: 400k SS, 400k DS of 800k: f 15,00 (1 diskette)
200k f 25,00 (2 diskettes)
Per 10 volumes in één bestelling een volume gratis.

Voor een OVERZICHT van de te bestellen volumes, meer dan 300, verwijst ik naar Softwarebus (uitgave van de CP/M-gebruikersgroep).

Over de LEVERTIJD van de bestelde volumes het volgende. Indien de volumes reeds in het bezit zijn van de NewBrain-gebruikersgroep, ontvangt u ze ongeveer een week na ontvangst van het verschuldigde bedrag.

Indien een of meerdere van de bestelde volumes nog niet in het bezit van de NewBrain-gebruikersgroep zijn, zullen ze eerst door ons bij de CP/M-gebruikersgroep besteld moeten worden. Daar ik in dit geval geheel afhankelijk ben van 'derden', kan de levertijd wel eens oplopen tot een maand of nog langer.

Het snelst krijgt men de volumes door verzending van het bestelformulier en girobetaalkaart (of betaal- of eurocheque) SAMEN naar bovenvermelde postbus.

Jos Hermans

cp/m-bibliotheek catalogus

van de volumes met specifieke NewBrain-CP/M-software

nbgg-1

- vreemde formaten
- INV, inverse karakters onder CP/m-highlight
- TOETS, gebruik van de NewBrain-screeditor onder CP/M
- PTR, schakelt de automatische linefeed na carriage return uit onder CP/M
- FAST, dit programma maakt de expanded NewBrain twee keer (!) zo snel met de beeldschermuitvoer onder CP/M

INV, TOETS en PTR zijn in On-line 4 beschreven;
in deze On-line kan de lezer een artikel over FAST vinden

nbgg-2 en 3

- geïnstalleerde ZCPR2-files (onder andere zelfstartende ZCPR2)

nbgg-4

- MODEM 7, aangepast voor de NewBrain (incl. SET- en DOC-file)
- PIOS en SYSPAGE, programma's om onder CP/M devices te sluiten en opnieuw te definiëren en te openen, etc. (incl. PASCAL-file)
- PROMPT, maak onder CP/M gebruik van het line display van NewBrain model AD (incl. ASM-file)
- DISKEDIT, programma om direct op disk te schrijven, lezen, veranderen; sectoren, tracks etc. (incl. PASCAL-file)
- GRDEMO en GRDEMO2, maak onder CP/M gebruik van high resolution (incl. PASCAL-files)
- SDMP, screendump onder CP/M

prijsvraag

programma's die de specifieke mogelijkheden van de newbrain gebruiken, kunnen alleen gemaakt worden, door mensen die vertrouwd zijn met de newbrain; en dat zijn de gebruikers. om nu de ontwikkeling van dergelijke programma's te bevorderen en om de leden er (via de softwarebibliotheek) de beschikking over te geven, heeft de gebruikersgroep een prijsvraag uitgeschreven. de opgave is niet al te makkelijk, maar de prijzen zijn ook niet kinderachtig:

eerste prijs: HONDERD GULDEN

tweede prijs: VIJFTIG GULDEN



een deskundige jury onder leiding van jan de vries zal de inzendingen beoordelen. de uitslag zal in on-line 6 bekend gemaakt worden. voor deze eerste prijsvraag is als onderwerp gekozen:

een 'databank'-programma

dat wil zeggen, een programma, dat in staat is gegevens op te slaan en op verschillende manieren te bewerken: toevoegen, wijzigen, verwijderen; het kan ook lijsten en gedeelten van lijsten gesorteerd aan de gebruiker presenteren, als antwoord op eenvoudige vragen: b.v. geef naam en telefoonnummer van de records met postcode > 1000 AA en postcode < 1100 AA (bij een adresbestand waarbij de records bestaan uit de velden: naam, adres, postcode, woonplaats en telefoonnummer).

aan de ingezonden programma's worden de volgende eisen gesteld: ze moeten voldoen aan de gegeven omschrijving van het onderwerp, geschreven zijn in newbrain-basic (machinetaalroutines zijn toegestaan), en uitgevoerd kunnen worden op een unexpanded newbrain met cassetterecorder, maar ook op een machine met expansion. aparte documentatiefiles en opsplitsing in deelprogramma's zijn toegestaan. gebruikersvriendelijkheid zal een van de beoordelingscriteria zijn. er mogen uiteraard alleen originele ontwerpen ingestuurd worden.

de programma's blijven eigendom van de maker; door inzending wordt slechts toestemming gegeven aan de gebruikersgroep om ze via de softwarebibliotheek te verspreiden.

de inzendtermijn sluit op 10 DECEMBER 1985. stuur uw programma('s) op cassette of diskette naar:

newbrain-gebruikersgroep, postbus 4494, 1009 AL amsterdam



INGENIEURSBUREAU J. L. SCHAGEN

**BEKEND OM
SERVICE EN PRIJZEN**

TEAC diskdrives:

FD-55A	1x40 tr.	5,25"	f	485,--
FD-55B	2x40 tr.	"	f	575,--
FD-55E	1x80 tr.	"	f	530,--
FD-55F	2x80 tr.	"	f	625,--

MITSUBISHI diskdrives:

M4851	2x40 tr.	5,25"	f	540,--
M4853	2x80 tr.	"	f	595,--
Mf-351	1x80 tr.	3,5"	f	530,--
Mf-352	2x80 tr.	3,5"	f	595,--

TEAC 3,5" diskdrives zijn in prijs gelijk aan de 5,25" types.

MATRIX printers:

STAR	SG-10	n.l.q.	f	1150,--
	SD-10	n.l.q.	f	1650,--
	SR-10	n.l.q.	f	2495,--
EPSON	RX-80-T		f	1245,--
	RX-80-F/T		f	1499,--
	FX-80-F/T		f	1950,--
BROTHER	M-1009		f	775,--
NAKAJIMA	NP-2200		f	1595,--

DAISYWHEEL printers:

BROTHER	CE-25	f	1190,--
	CE-51 Super	f	1625,--
	CE-60 Super	f	1825,--
	HR-15	f	1695,--

ACCESSOIRES:

Parallelkabel	f	75,--
Diskdrivekabel 1 dr.	f	65,--
Diskdrivekabel 2 dr.	f	75,--
Kast v. 2 drives	f	59,50

DISKETTES: Fuji

MD1D 5,25" ssdd 40tr.	f	67,50
MD2D ,, dsdd 40tr.	f	89,--
MD2DD ,, dsdd 80tr.	f	119,--
MF1DD 3,5" ssdd	f	165,--
MF2DD 3,5" dsdd	f	235,--

MONITOREN:

TAXAN:	RGB VISION II, 12", 510 dots x 262 lijnen		f	1350,--
	RGB VISION III, 12", 640 dots x 262 lijnen		f	1599,--
Prijzen van TAXAN	tijdelijk zo laag !!!!!!!!!!!!!!!			

TAXAN:	RGB VISION IV, 12", 790 dots x 410 lijnen		f	2675,--
--------	---	-------	---	---------

ALLE PRIJZEN INKLUSIEF BTW !!!!!!!!!!!

BEL ONS OOK VOOR DE PRIJZEN VAN ONZE ANDERE PRODUKTEN.

VISSERINGSTRAAT t/o nr. 45 — TEL. 861738

1051 KJ AMSTERDAM



HET WERKEN MET DISK VANUIT BASIC

De commando's 'era', 'dir' en 'ren' van CP/M zijn ook vanuit BASIC uit te voeren, namelijk via device 15. Dit gaat als volgt.

'era': OPEN #15, 15
PRINT #15, 0, "<filename>"
CLOSE #15

'ren': OPEN #15, 15
PRINT #15, 1, "<oldfile>", "<newfile>"
CLOSE #15

Om een directory op te vragen verwijst ik naar het 'Handbook for disk users', dat bij de controller wordt geleverd. Een programma staat op pagina 52.

H. Balemans



GESPIKKELDE SPINNEN?

bij de produktie van on-line 4 is in het artikel SPRITES van John van der ploeg een programmaregel in het ongerede geraakt. wie regel 670 (op bladzijde 52) zo overgenomen heeft, als hij afgedrukt staat, heeft een gespikkeld blok in plaats van een spin op het scherm gekregen. het is wel duidelijk, dat daar de waarden horen te staan, die ook naast de tekening

van de spin op bladzijde 49 vermeld zijn, voor de volledigheid hier de juiste regel 670:

```
670 DATA 0, 32, 4, 0, 48, 160, 5, 12, 200, 225, 135, 19, 8, 162, 69, 16,
100, 228, 39, 38, 146, 105, 150, 73, 17, 210, 75, 136, 9, 228, 39, 144, 201,
124, 62, 147, 37, 125, 190, 164, 37, 30, 120, 164, 37, 28, 56, 164, 165, 127,
254, 165, 165, 124, 62, 165, 165, 224, 7, 165, 165, 0, 0, 165
```

de redactie



NEWBRAIN TEXT EDITOR + STAR GEMINI 10 X

Om de NewBrain text editor van de heren Haring en Pinto, versie 3 release 7, op de Star Gemini 10 X te doen werken, heb ik in de lijnen 11515, 11517 en 11518 telkens PUT #9, 27, 80 moeten vervangen door: PUT #9, 27, 66. Hierna werkt de printroutine perfect.

A. Van de Velde

HOOFDLETTERSLOT VOOR "TXY2"

Het tekstverwerkingsprogramma "TXY2" kent niet de mogelijkheid om met CONTROL + 1 in de hoofdlettermode te komen. Om een zin in hoofdletters te typen zit er niets anders op dan om met de ene hand de SHIFT-toets ingedrukt te houden en met de andere hand de toetsen aan te slaan. Vooral als je veel in hoofdletters moet typen, kan dit frustrerend werken.

Er zijn slechts enkele kleine wijzigingen in het programma nodig om wel direct hoofdletters in te kunnen typen:

oude tekst:

```
111 IF a > 127 GOTO 3000
112 IF a = 27 GOSUB 3500 : GOTO 320
113 IF a = 95 THEN a = 223
```

...

```
330 PUT a : IF a < 3 OR a > 29 THEN ed = 1
```

wijzigen in:

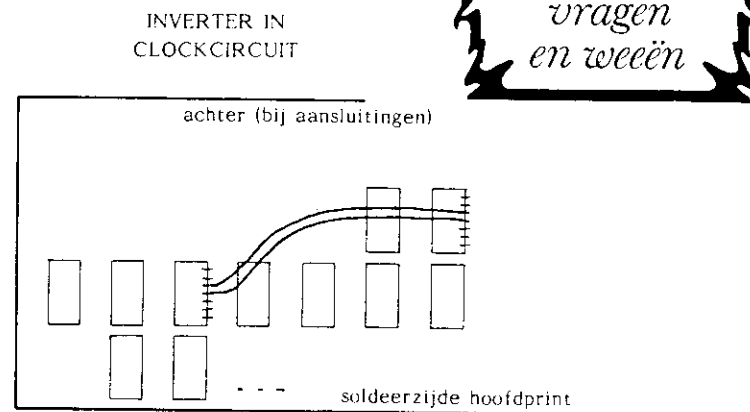
```
111 IF a = 95 THEN hl = ABS (hl - 32) : GOTO 110
112 IF a > 127 GOTO 3000
113 IF a = 27 GOSUB 3500 : GOTO 320
114 IF a = 95 THEN a = 223
```

...

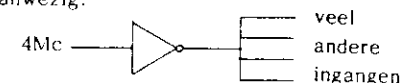
```
330 IF a > 96 IF a < 123 THEN a = a - hl
335 PUT a : IF a < 3 OR a > 29 THEN ed = 1
```

Door de SHIFT-toets in te drukken samen met de VIDEOTEXT-toets wissel je van hoofd- naar kleine letters en andersom.

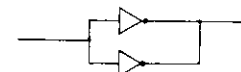
Pieter W. M. van Dijk



Bij het clockcircuit is 1 inverter aanwezig:



Dit is te weinig. Een parallel te schakelen inverter zorgt ervoor, dat de NewBrain significant minder gevoelig wordt voor storingen, vooral als er uitbreidingen bij zijn.



Rob Maris

THE NEWBRAIN FILES

a completely new book all about the NewBrain & its graphics & the 25 device drivers & all 80 operating system Z-calls with their parameters and how to get them & the most from the BASIC & how the paged memory works & CP/M's BIOS & why VIDEOTEXT & all the other things they never told you before plus lots of routines & examples & a text-handler & a database & smart graphics & details of key memory locations & what's there & how to read it & change it

The NewBrain Files

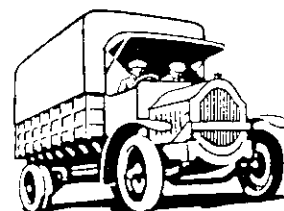
Hfl 40,00 or £sterling 10.00

Outside Europe, please add £2.00

Payment, by sterling bank draft or Eurocheque
or by local banknotes, to:

Vitagraph Ltd,
26 College Road,
Bromley,
BR1 3PE,
Kent, England

trans- portabel



HOF.
KUN JE PROGRAMMA'S ZO MAKEN,
DAT ZE OP ALLE NEWBRAIN-
CONFIGURATIES WERKEN?

de trotse maker zal zijn programma, waar hij zelf zoveel plezier van heeft en dat bij hem altijd zo goed werkt, even bij een ander demonstreren. "moet je opletten . . .", nauwelijks is het opgestart, of het komt terug met een foutmelding.

eindelijk is dan de tweede diskdrive gekocht. meteen proberen: LOAD "b:progr.bas". RUN. dat gaat goed. maar zodra het programma zelf een bestand op schijf wil aanspreken, komt er ERROR 153. het programma weet nog niet, dat voortaan de letter van de schijf bij de bestandsnaam hoort.

elke configuratie van de newbrain heeft haar eigen mogelijkheden, maar stelt ook haar eigen eisen aan een programma. en wie, zoals ik gedaan heb, zijn machine stap voor stap uitbreidt, mag telkens zijn programma's doorlopen om ze te 'updaten'; het is een mooie naam voor een vervelende klus. daarom zal ik in dit artikelje een aantal punten noemen, waarop de diverse configuraties van elkaar verschillen. de programmeur, die daar rekening mee houdt, zal zijn programma's ongewijzigd (of met een simpele wijziging in het begin van het programma) op een andere configuratie laten draaien.

voor de programma's, die aan de softwarebibliotheek ter beschikking gesteld worden, is het van groot belang, dat iedereen ze op zijn configuratie kan draaien, zonder dat er hard- of softwarematig componenten afgekoppeld worden, of eerst nog aangeschaft moeten worden (dit geldt uiteraard niet voor programma's, die juist bestaan bij de gratie van het gebruik van bepaalde devices)

HOE ZIET DE COMPUTER, WAT AANWEZIG IS?

de computer kan niet zien, van welke componenten de snoertjes aangesloten zijn. maar met een paar PEEKs komt hij een heel eind. adres 19 bevat het 'default backup storage device number': het nummer van de device driver, die de computer zal uitkiezen voor opslag van gegevens, als het programma of de gebruiker het niet expliciet vermeldt, bij voorbeeld in de opdracht LOAD. de disk controller verandert dat nummer van 1 in 12: dus als PEEK (19) als resultaat 12 heeft, is de disk controller aangesloten. net zo verandert de expansion adres 51 door het topbit 1 te maken: 0 wordt dan 128. dus als PEEK (51) > 127, dan is het paged operating system in werking

PEEK EN POKE

aansluiting van een expansion heeft trieste gevolgen voor mensen, die aan PEEKs en POKEs verslaafd zijn: de inrichting van de system page wordt nogal ingrijpend veranderd; stream table, device table, de plaats van het scherm, alles is anders.

de adressen 119 en hoger worden verschillend gebruikt, evenals 2 en 3, 6 en 7, 20 - 23, 30 en 31, 38 en 39, 44 - 47, 86 - 101, en de bits 5 - 7 van adres 14. van adres 51 wordt in het non-paged system het bit met de laagste waarde gebruikt om de uitlezing van het scherm te onderbreken. het paged system maakt, zoals gezegd, gebruik van het hoogste bit. in plaats van POKE 51, 1 kan beter gebruikt worden POKE 51, PEEK (51) OR 1; en in plaats van POKE 51, 0 komt POKE 51, PEEK (51) AND 128. dan blijft steeds de waarde van het topbit gelijk, terwijl het laagste bit wel veranderd wordt

CALL

zoals na de mededeling over PEEK en POKE te verwachten was: CALLS naar het operating system kunnen niet zonder ingewikkelde voorbereiding gedaan worden in een expanded system, dit betekent onder andere, dat de gemakkelijke klok, waar even met 'CALL 62399, t' op gekeken wordt, niet meer beschikbaar is. als je weet, dat die routine gewoon de inhoud van de adressen 105 t/m 107 bekijkt, dan is er niets aan de hand: dat kunnen we zelf ook (cf. on-line 1, pag. 26):

$$t = 65536 * \text{PEEK}(105) + 256 * \text{PEEK}(106) + \text{PEEK}(107)$$

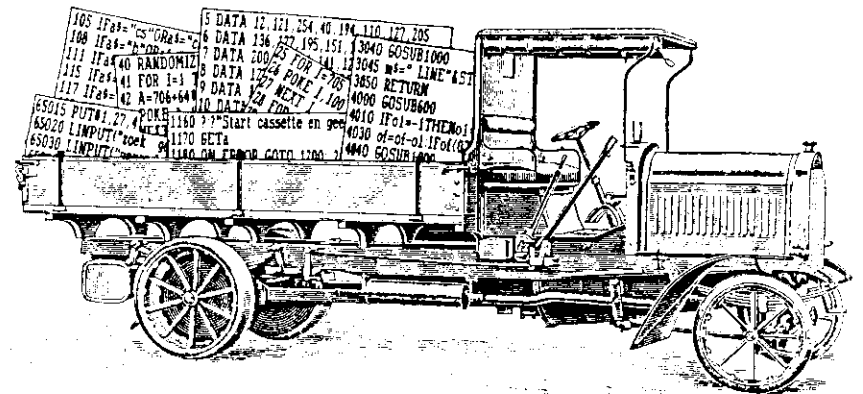
en de CALL 62383 doen we handmatig door op de drie adressen 0 te POKEn.

bij de expanded newbrain wordt op precies dezelfde manier ruimte voor machinetaalroutines gereserveerd, als bij een unexpanded. het gebeurt alleen op een ander adres, want TOP ligt veel hoger. zolang als de aanroep gebruik maakt van de functie TOP (CALL TOP [+ . . .]), en er geen absolute adressen gebruikt worden in de machinetaalroutine, gaat alles goed.

het vrijmaken van gereserveerde geheugenruimte geschiedt vrijwel identiek, alleen neemt 2^{14} de plaats in van 2^{15} , dus een algemene formule is:

```
RESERVE TOP + 2 ↑ (15 + (PEEK (51) > 127))
```

denk er wel aan, dat de newbrain met expansion niet de twee vrije bytes heeft, die hij kaal wel bewaart (voor alle zekerheid?) tussen het gebied van basic en het ontoegankelijke erboven: $\text{RESERVE TOP} + 2 \uparrow 14 + 2$ geeft geen goed resultaat



DEVICES

de kale newbrain kent de devices 0 t/m 11, de disk controller draagt 12 t/m 15 bij, en de expansion voegt er de nummers 16 en hoger aan toe.

de newbrain zonder disk controller kan dus alleen sequentiële bestanden gebruiken, daarvoor gebruikt de de disk controller device 12, schrijven en lezen gaan net zo, het grote verschil is de parameterstring bij OPEN, de devices 1 en 2 vragen om [buffergrootte plus] een bestandsnaam, waarvoor weinig beperkingen gelden; de disk controller wil [drive-aanduiding plus] bestandsnaam in cp/m-formaat: 1 tot 8 letters, eventueel gevolgd door een punt en 0 tot 3 letters, als letter zijn toegestaan: ASCII 33 t/m 126, met uitzondering van < > . , ; = * ? [en], er wordt geen verschil gemaakt tussen hoofd- en kleine letters, drive-aanduiding is a: voor de eerste drive,

b: voor de tweede, enz.

het is misschien wel de moeite waard om voor elke bestandsnaam een string-variabele als drive-aanduiding in het programma op te nemen (bij voorbeeld: OPEN OUT# 1, PEEK (19), di\$ + "[bestandsnaam]"), ook als dat nog niet nodig is, omdat op cassette of een enkele diskdrive gewerkt wordt. als di\$ een lege string is, heeft het programma er geen last van, en wanneer er dan een tweede drive gebruikt wordt, hoeft er alleen maar een regeltje in het begin gemaakt te worden, dat di\$ een waarde geeft, in plaats van dat het hele programma doorgelopen moet worden om te zien, of er ergens een bestand op schijf geopend wordt.

bij opslag op cassette is het gebruikelijk om de gebruiker, door middel van een mededeling op het scherm, uit te nodigen een cassettebandje op de juiste plaats klaar te zetten en daarna een toets aan te slaan, als dit in een subroutine gebeurt, kan het programma het gemakkelijk overslaan, als met schijf gewerkt wordt: IF PEEK (19) < 12 GOSUB [uitnodiging]

wie de beschikking heeft over een expansion interface, zal veel liever de printer- en comms-uitgangen aan de EIM gebruiken (dus de devices 16 en 17 in plaats van 8 en 9), omdat dan het scherm niet knippert en de linefeed na de carriage return uitschakelbaar is.

ook hier kan het programma zelf uitzoeken, welk device gebruikt moet worden: OPEN OUT# 1, 8 - 8 * (PEEK (51) > 127)

ook ten aanzien van het toetsenbord weet de expansion een verbetering te brengen: device 19, het toetsenbord met buffer. je hoeft niet te wachten totdat het programma zo ver is, dat het je toetsaanslag kan verwerken. praktisch bij tekstverwerking, menu-gestuurde programma's en zo. unexpanded systemen kennen het alleen onder cp/m; basic heeft de expansion ervoor nodig. device 19 is van het type 'without immediate return': als er nog geen toets is aangeslagen op het moment, dat het programma om een toetsaanslag vraagt, blijft het programma wachten, totdat het wel gebeurt (net als bij device 5). maar in de volgende programmaregel, die typisch is voor device 6, kan stream kb net zo goed verbonden zijn met device 19:

```
1000 GET #kb, g : IF g = 0 THEN 1000
```

er schuilt echter wel een addertje onder het gras: als er naast device 19 ook andere gebruikt worden voor invoer via het toetsenbord (5 of 6, maar ook 0 via (L)INPUT), komen die toetsaanslagen ook in de buffer van device

19 terecht, dat is iets, waarmee de programmeur terdege rekening moet houden, maar het is geen enkel probleem een programma zo te schrijven, dat alle toetsenbordinvoer via hetzelfde device loopt

BASIC

met de disk controller komen de nieuwe basicopdrachten SAVF, die programma's zo opslaat, dat ze daarna sneller geladen worden (VERIFY is dan niet meer mogelijk, maar op schijf ook niet nodig), en CPM, die de overgang van basic naar cp/m bewerkstelligt. voor dat laatste gebruikt de expansion EXIT. nu mag een machine, die zo'n opdracht niet kent, hem ook niet ter interpretatie aangeboden krijgen. wil het programma zichzelf (in een gewijzigde versie bij voorbeeld) opslaan en daarbij kiezen uit SAVE en SAVF, dan zal een regel als

```
IF PEEK (19) = 12 THEN SAVF "[bestandsnaam]"
```

onherroepelijk tot de foutmelding ERROR 55 leiden op een unexpanded systeem, ook al moet de opdracht niet uitgevoerd worden. beter is:

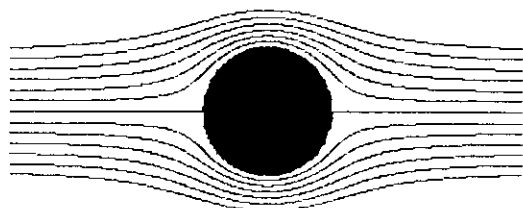
```
1620 IF PEEK (19) < 12 GOSUB [uitnodiging] : SAVE "[bestandsnaam]" :  
GOTO 1640  
1630 savf "[bestandsnaam]"  
1640 . . .
```

behalve uitbreidingen brengt de expansion ook een beperking mee: de parameterstring bij een OPEN-opdracht mag niet langer zijn dan 256 bytes. dit geldt ook voor de PLOT-opdracht. die zal dan in twee opdrachten gesplitst moeten worden

menno stevens



stroming



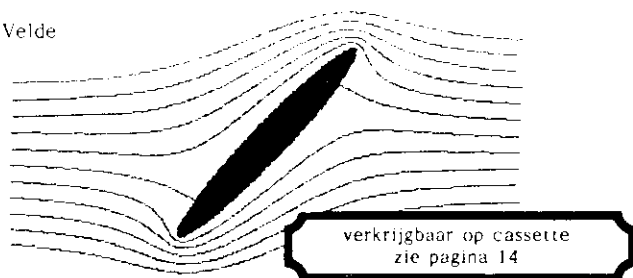
Stroming is een programma, dat het verloop tekent van stroomlijnen rond elliptische cilinders. De zeer goede hi-res van de NewBrain komt hier zeer tot uiting. De elliptische cilinders kunnen elke gewenste vorm hebben, van een dunne plaat tot een cirkelvormige cilinder. Ze kunnen ook elke gewenste hoek met de stroming maken.

Het belang van het programma is zuiver didactisch op het gebied van de hydrodynamica en de potentiaaltheorie. Het is niets anders dan de berekening van een functie van een complexe variabele die, zoals bekend, een conforme afbeelding oplevert. De gebruikte transformatie is:

$$F(w) = \frac{1}{2} \left(w + \sqrt{w^2 - (a + b)^2} + \frac{a^2 - b^2}{e^{-2it} \left(w + \sqrt{w^2 - (a + b)^2} \right)} \right)$$

W en F(w) zijn de complexe variabelen, die de punten bepalen van de ongestoorde parallelle stroomlijnen en de punten van de gestoorde stroomlijnen. Hierin zijn a en b respectievelijk de halve grote en de halve kleine as van de ellipse. Het getal e is de basis der natuurlijke logaritmen en i is de imaginaire grootheid $\sqrt{-1}$. Tenslotte is t de hoek door de grote as van de ellipse gemaakt met de ongestoorde stroming.

A. Van de Velde



microcalc

Het programma Microcalc is een soort spreadsheet-programma, geschreven voor de NewBrain, en werkt zowel op de 32k- als op de 96k-versie. Het zoekt zelf uit, of er een diskdrive aanwezig is; anders wordt cassettepoort 1 gebruikt. Bij de 32k-versie gebruikt het printerdevice 8 en bij 96k een device 16. Baudrate etc. worden bij het opstarten van het programma ingevoerd.

Na het opstarten komt men op het spreadsheet. Vanuit deze toestand kan men alle Microcalc-routines aanroepen. Men kan door het spreadsheet heen en weer wandelen met de cursortoetsen. Hierbij wordt het bestand iedere keer doorgerekend. Als dit niet nodig is, gebruik dan de toetsen graphics-op en -neer. Dat is een stuk sneller.

OPTIES (W, I, L, S)

U komt in deze routine met de toets O. U kunt dan kiezen uit W: wijzigen, I: inlezen, L: laden, S: save.

De opties wijzigen en invoeren. Hier kan men het programma instellen voor het doel, waarvoor men het wil gebruiken. Eerst wordt er om een scherm balk gevraagd. Deze scherm balk is de bovenste regel van elk scherm. Het is dus handig om hier bijvoorbeeld uw naam en de datum neer te zetten, of het doel van de berekening. Daarna moeten de namen van de tekstkolommen ingevoerd worden. Er zijn maximaal 8 kolommen, maar als u er minder wilt gebruiken moet u als laatste kolomnaam 'stop' opgeven. Dit geldt ook voor de numerieke kolommen, die daarna ingevoerd moeten worden. Er is nog iets speciaals met de kolomnamen. Er kunnen namelijk onzichtbare kolommen gemaakt worden. Stel u bent penningmeester van een vereniging en u wilt de contributie-administratie via dit programma doen. U heeft dan namen, adressen enz. nodig, maar die passen niet allemaal op het spreadsheet. Dan maakt u alleen de kolom 'naam' zichtbaar op het spreadsheet en de rest van de tekstkolommen onzichtbaar. Die kolommen zijn nu niet verloren. Bij het adresseren van b.v. aanmaningen kunt u ze nog gewoon gebruiken. U maakt een kolom onzichtbaar door als laatste letter van de betreffende kolomnaam een graphics-n (n van 'niet afdrukken') in te voeren.

Als alle kolommen (tekst en getallen) een naam hebben, komt het invoeren van de formules. Elke numerieke kolom wordt voorgesteld als een k()-

variabele. Kolom 1 is dus k(1), kolom 2 k(2) enz. Als de inhoud van kolom 3 nu gelijk moet zijn aan de inhoud van kolom 1 + de helft van kolom 2 voert u bij de vraag k(3) = in: k(3) = k(1) + (k(2) / 2). Als b.v. kolom 1 geen bewerking hoeft te ondergaan en dus een gegevenskolom is, voert u in: k(1) = k(1). Na het invoeren van alle formules komt het instellen van de som-routine. Bijvoorbeeld: u bent betonstortter en u hebt een kolom voor kubieke meters beton. Dat is kolom 4. Bij som zou u dan een berekening kunnen programmeren, die volumes uitrekent. Dat zijn 3 waarden (volume = l x b x h). Dat moet ingevoerd worden. Daarna op welke kolom dit moet werken. In dit geval kolom 4. De in te voeren formule is dan OP TE TELLEN = si(1) * si(2) * si(3). Nu zijn alle opties ingevoerd en keert het programma terug naar het spreadsheet.

Opties laden en save. U kunt de naam van het bestand opgeven. De rest spreekt voor zich. Na het laden van de opties komt u in de wijzigroutine. Daar kunt u met NEWLINEs gewoon doorheen wandelen.

EDIT

U komt in deze routine met de toets E. Als u in de edit mode zit, moet u even antwoord geven op de vragen. Bij de numerieke kolommen, waar een antwoord in moet komen te staan, hoeft u natuurlijk geen antwoord te geven. Dat rekent het programma via de bij opties ingevoerde formules zelf wel uit.

TEKST

U komt in de tekstverwerker met de toets T. U zit dan in de normale NewBrain-teksteditor. Er zijn echter drie editortoetsen, die wat anders dan gebruikelijk doen, namelijk:

shift-home	werkt helemaal niet meer.
control-home	schuift de spreadsheet-totalenbalk naar deze positie en verlaat de tekstverwerker
escape	verlaat de tekstverwerker

PRINTERUITDRAAI

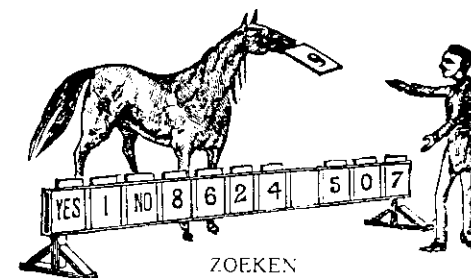
Met een druk op de toets P wordt er een letterlijke kopie van het spreadsheet gemaakt op de printer. Ook de via de tekstverwerker toegevoegde tekst wordt meegenomen. Na het printen komt u weer terug in het spreadsheet. De regel met de Microcalc-commandotokens (2e van boven) wordt bij het printen overgeslagen.

SOM

U komt in de som-routine met de S-toets. U moet hier de nodige getallen invoeren. Die getallen worden dan volgens het som-formuletje, dat bij opties ingevoerd was, berekend. Als een heel berekeningetje af is, begint u vanzelf aan het volgende. Als u de som-routine wilt verlaten, moet u zorgen, dat de uitkomst van de som-berekening nul is. Dan komt u vanzelf weer op het spreadsheet.

INLEZEN EN WEGSCHRIJVEN

U kunt de data uit het Microcalc-bestand inlezen en wegschrijven van en naar het randgeheugen d.m.v. de toetsen I en W. U kunt hierbij de naam van het bestand opgeven. Bij gebruik van disk moet altijd een naam opgegeven worden. Als, bij gebruik van tape, bij het inlezen geen naam wordt opgegeven, wordt de eerste de beste file ingelezen. Denk eraan, dat de tekst van de Microcalc-tekstverwerker niet wordt meegesaved.



U hebt de mogelijkheid om in het bestand te zoeken op elk trefwoord of deel daarvan en op elke kolom met de Microcalc-routine ZOEKEN. U roept deze routine aan met de toets Z. Dan moet u het type kolom (tekst of numeriek) opgeven, en in welke kolom van het door u gekozen type u wilt zoeken. Dan moet u invoeren, in welk deel van het bestand u wilt zoeken.

< trefwoord: alles kleiner of eerder dan het trefwoord + het trefwoord zelf
 = trefwoord: alles, wat met het trefwoord begint + het trefwoord zelf
 > trefwoord: alles groter of later dan het trefwoord + het trefwoord zelf

U moet dus een '<', een '=' of een '>' intoetsen, gevolgd door het trefwoord. Met alleen de eerste paar letters vindt het programma het ook wel.

Met tekstkolommen wordt gekeken naar het alfabet, waarbij geldt, dat hoofdletters voor kleine letters komen. Als het programma de post

gevonden heeft, wordt die op het scherm afgedrukt, en kunt u hem ook nog op de printer af laten drukken.

SPECIALS

De specials zijn te bereiken met de toetsen 1 en 2. Deze routines zijn nog niet aanwezig en zult u dus zelf nog moeten schrijven in BASIC. Hier zou u dus een routine kunnen maken voor diskdrive, of voor het rechtstreeks uitprinten van aanmaningen. In het volgende hoofdstukje kunt u zien, waar en hoe de gegevens zijn opgeslagen.

OPBOUW EN DATA-ORGANISATIE

Het programma is opgebouwd in subroutines:

1-999 initialisatie en voorpagina
 1000 haalt bytes op #5 in variabele gk
 1002 commandoscan op gk
 1120 tekst
 1130 edit
 1200 display cursor, zonodig verplaats totaalbalk
 1210 opties
 1220 opties invoeren of wijzigen
 1362 invoeren van formules
 1400 werk scherm bij
 1500 gebruik formules op postnummer pn
 1600 som
 1700 invoeren somopties
 2000 printer screendump
 3000 inlezen
 3100 wegschrijven
 4000 opties laden / saven
 5000 special 1 (zelf te schrijven subroutine)
 7000 special 2 (zelf te schrijven subroutine)
 9000 zoeken
 9100 druk post af na zoeken

Variabelen (het getal tussen haakjes is de maximale waarde):

tt(8) totalen
 si(16) som waarden
 kn\$(8) kolomnaam tekst
 nk\$(8) kolomnaam numeriek
 ki\$(8,40) inhoud tekstkolom (kolomnummer, postnummer)

ki\$(8,40) inhoud numerieke kolom (kolomnummer, postnummer)
 k\$(8) formules
 k(8) tussenwaarden bij gebruik van formules
 ak aantal tekstkolommen
 ka aantal numerieke kolommen
 qq\$ schermbalk
 pn lopend postnummer
 pn\$ postnummer wel of niet gebruikt (J/N)
 fs\$ formule voor somroutine
 bd\$ baudrate of parameter van de printerdevice.
 pr nummer printerdevice (8 of 16)
 dv devicekeuze randgeheugen (1 of 12: PEEK (19))
 md\$ bevat " tape " bij dv = 1, en " disk " bij dv = 12
 ym y-coördinaat totaalbalk
 gk ASCII-waarde lettercode commando
 ot op te tellen tussenwaarde bij som

en enkele hulpvariabelen, die na terugkeer uit een subroutine waardeloos zijn geworden.

Toetsen voor aanroepen van de subroutines:

E edit
 I inlezen
 O opties
 Z zoeken
 W wegschrijven
 P printer
 T tekst
 S som
 1 basic-special 1 op regel 5000
 2 basic-special 2 op regel 7000

pijltje omhoog ga regel omhoog in spreadsheet
 pijltje omlaag ga regel omlaag in spreadsheet
 graphics-pijltje zie boven maar dan snel en zonder het hele bestand door te rekenen

verkrijgbaar op cassette
 zie pagina 14

GEBRUIK VAN MICROCALC

Als u tijdens het gebruik van Microcalc uit het programma raakt (er verschijnt STOPPED AT ... op het scherm), type dan CONT en geef NEWLINE. In de meeste gevallen zal dit wel helpen. Als de opmaak van het spreadsheet scherm niet meer in orde is, bijvoorbeeld door gebruik van de

tekstverwerker of door een foutmelding, roep dan opties aan. Tik dan een W (van wijzigen). Tijdens het wijzigen gewoon overal NEWLINE geven. Als u dan op het spreadsheet terugkeert, is het scherm weer in orde gebracht. Als de totalenbalk te ver gezakt is, kunt u die 'ophijzen' d.m.v. control-home uit de tekstverwerker.

Om de opties handig in te stellen zijn hier ook een paar tips aan de hand van een voorbeeld (kasboek van een vereniging).

De situatie is :

tekstkolom 1: naam
 tekstkolom 2: adres graphics-n (onzichtbaar op spreadsheet)
 tekstkolom 3: postcode graphics-n (onzichtbaar op spreadsheet)
 enz.
 num. kolom 1: bet.wk. (betaald aantal weken)
 num. kolom 2: con.wk. graphics-n (onzichtbaar, contributie per week)
 num. kolom 3: schuld
 num. kolom 4: totaal
 formule 1: $k(1) = k(1)$
 formule 2: $k(2) = 1.75 * (-1 * (len(ki\$ (1, pn)) > 1))$
 formule 3: $k(3) = 20 - k(1)$
 formule 4: $k(4) = k(1) + k(3)$

De contributie is f 1,75 per week. Er moet alleen contributie betaald worden, als iemand bestaat. Met $-1 * (len(ki\$ (1, pn)) > 1)$ wordt dit uitgezocht (boolse vergelijking). De lengte van de naam moet dus langer zijn dan 1 om bij $k(2)$ 1.75 te krijgen. Als de naam 1 teken of korter is, dan $k(2) = 0$.

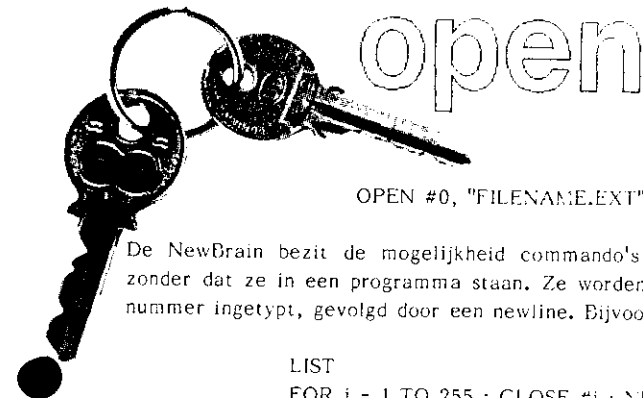
De variabele pn is het postnummer. Het postnummer wordt hier dus wel gebruikt.

Aantal getallen bij som = 1. Werkend op kolom 1 (weken betaald). De formule is: OP TE TELLEN = $si(1) / 1.75$

Nu kun je dus, als er iemand betaalt, gewoon som aanroepen, het bedrag, dat hij betaalt, intikken en weer terugkeren. Er moet door 1.75 gedeeld worden, omdat kolom 1 het aantal betaalde weken is.



John van der Ploeg
 Goudsbloemstraat 31
 1782 AT Den Helder



OPEN #0, "FILENAME.EXT"

De NewBrain bezit de mogelijkheid commando's direct uit te voeren, zonder dat ze in een programma staan. Ze worden daartoe zonder regelnummer ingetypt, gevolgd door een newline. Bijvoorbeeld:

```
LIST
FOR i = 1 TO 255 : CLOSE #i : NEXT i
LOAD "filename.ext"
```

Deze commandoregels mogen zeer lang zijn, en kunnen dus uit hele programma's bestaan (one-liner).

De NewBrain kan deze commandoregels niet alleen via het scherm ontvangen, maar ook via de standaard data-device [PEEK (19)]. Op de diskette (of cassette) staat dan een file, die meteen vanaf het eerste karakter de commandoregel bevat, gevolgd door een newline-teken. Typt men 'OPEN #0, "filename.ext"', dan wordt deze commandoregel ingelezen en meteen uitgevoerd.

Er zijn echter enkele verschillen met de normale manier van commandoregel uitvoeren:

1. Stream 0 wordt geblokkeerd, en moet weer heropend worden (b.v. 'OPEN #0, 0').
2. Als bij het afbreken van de commandoregel (laatste statement, END, NEW, LOAD, STOP of foutmelding) stream 0 nog niet heropend is, voert de NewBrain de volgende commandoregel uit, die achter deze commandoregel in de file staat. Dit gebeurt net zolang, totdat stream 0 heropend is. Als dit na de laatste commandoregel nog niet gebeurd is, klappt het systeem dicht (enige oplossing is: NewBrain uit- en aanzetten).
3. Als stream 0 heropend is, gelden dezelfde regels als normaal op het scherm.
4. Voordat stream 0 heropend is, mag stream nul niet gebruikt worden (dichtklappen!) (b.v. 'PRINT AS'), maar moet er een kopie gebruikt worden (b.v. 'OPEN #1, 0, 1 : PRINT #1, AS'). Dit geldt ook voor het

openen van de standaard data-device (b.v. 'OPEN #100, PEEK (19), 1, "...").

5. Een commandoregel mag ook een regelnummer bevatten, wordt dan niet uitgevoerd, maar als programmaregel geMERGED bij het programma in het computergeheugen. computergeheugen.
6. Bij verwisseling van diskette (of cassette) kunnen alleen de regels die in de device-buffer (grootte: 128-meervoud) aanwezig zijn, uitgevoerd worden.

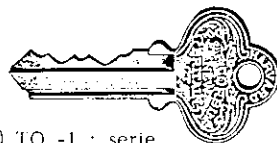
Omdat de NewBrain geen ELSE-statement kent, kan er alleen een lange commandoregel gemaakt worden door gebruik van FOR - NEXT.

Bijvoorbeeld:

```
10 ...
20 IF voorwaarde THEN serie statements a : GOTO 40
30 serie statements b
40 ...
```

wordt:

```
... : FOR i = (voorwaarde) TO -1 : serie
statements a : NEXT i : FOR i = 0 TO
(voorwaarde) : serie statements b : NEXT i :
...
```



Er volgen nog enkele voorbeelden van direct oproepbare commandoregels. Met 'OPEN #0, "OM"' kunnen deze files bekeken en veranderd worden, en met 'OPEN #0, "O-x"' kunnen ze gedraaid worden.

O-1

```
FS = "filename.ext" : CLOSE #1 : OPEN #1, 0, 1 : CLOSE #12 : OPEN
IN#12, PEEK (19), 1, FS : REM In deze regel worden 2 kopieën geopend.
? #1 : ? #1, "HET PROGRAMMA " + FS + " WORDT NU" : ? #1, "OP-
GESTART." : ? #1 : ? #1, "EEN OGENBLIKJE ALSTUBLIEFT." : REM
Deze regel drukt een mededeling af op het scherm.
LOAD #12 : REM Dit REM-statement komt de computer nooit tegen.
CLOSE #12 : CLOSE #1 : OPEN #0, 0 : RUN : REM door nu deze
commandofile te draaien, wordt het programma FS zowel gelOAD als
geRUNd.
REM Ook dit REM-statement komt de computer nooit tegen, zowel door
RUN als door OPEN #0, 0 (zie vorige regel).
```

O-2

```
10 Y = X * X : RET
OPEN #0, 0 : FOR X = 0 TO 10 : GOSUB 10 : ? "X="; X; TAB(10); "X*X=";
Y : NEXT X : DELETE 10
REM Het is mogelijk om subroutines te gebruiken. Deze subroutines kunnen
met deze commandofile gemaakt worden, of er kunnen al bestaande
subroutines worden aangeroepen.
```

O-3

```
FOR DI = 1 TO 255 : CLOSE #DI : NEXT DI : OPEN #0, 0, "1" : OPEN #15,
15 : ? #15, 3, "A:*.*)" : FOR DI = 1 TO 1 : LINPUT #15, DAS : ? LEFT$
(DAS + " ", 16); : PUT -13 * (INT (DI / 5) = (DI / 5)) : ? #15, 4
: DI = -(DAS = "") : NEXT DI : ? #15, 6 : CLOSE #15 : ?
REM Bovenstaande regel geeft alle files op de diskette in drive A, zonder
dat er een programmaatje aan te pas komt, dat het programma in de
computer kan aantasten.
```

O-4

```
CLOSE #1 : OPEN #1, 0, 1 : ? #1, "Type een getal " : INPUT #1, A : A =
INT (A)
IF A / 2 = INT (A / 2) ? #1, A; "is een even getal."
IF A / 2 <> INT (A / 2) ? #1, A; "is een oneven getal."
FOR I = 1 TO 1000 : NEXT I : CLOSE #1 : OPEN #0, 0
REM Dit programmaatje bevat wel IF-statements, maar kan daardoor maar
een keer worden doorlopen, omdat een FOR-NEXT-loop niet meer dan een
regel kan omvatten.
REM De wachtloop in de laatste regel is noodzakelijk, omdat de tekst op
een kopiescherm staat en zal verdwijnen.
```

O-5

```
OPEN #0, 0 : FOR I = 1 TO 1 : INPUT ("Type een getal ") A : A = INT (A) :
FOR J = (A / 2 = INT (A / 2)) TO -1 : ? A; "is een even getal." : NEXT J :
FOR J = (A / 2 <> INT (A / 2)) TO -1 : ? A; "is een oneven getal." : NEXT J
: I = -(A = 0) : NEXT
REM Dit programmaatje gaat net zolang door tot het getal nul is ingetypt.
```

O-6

```
OPEN #0, 0
Deze regel hoeft niet eens met een REM-statement te beginnen, omdat
stream 0 al heropend is in de vorige regel.
```

```

OPEN #0, 0, "I25" : ? "*** PROGRAMMA KLAD ***" : ? : ? "ALS KLAAR
MET AANTEKENINGEN MAKEN: GRAPHICS/E" : ? : LINPUT ("WELK
KLADPAPIERTJE?") KFS : KFS = LEFTS (KFS, INSTR (KFS + ".", ".") - 1)
+ ".KLD" : PUT 31 : CLOSE #15 : OPEN #15, 15 : ? #15, 3, "A:" + KFS :
LINPUT #15, KAS : ? #15, 6 : CLOSE #15 : FOR KA = (KAS <> "") TO -1 :
CLOSE #12 : OPEN IN#12, 12, KFS : LINPUT #12, KAS : ? KAS : PUT 12 :
CLOSE #12 : NEXT KA : CLOSE #5 : OPEN #5, 5 : PUT 6 : FOR KA = 1
TO 1 : GET #5, KB : PUT -KB * (KB <> 133) : KA = -(KB = 133) : NEXT KA :
CLOSE #5 : KAS = "" : PUT 12, 3 : FOR KA = 1 TO 25 : LINPUT ("" ) KB$ :
KAS = KAS + KB$ + CHR$ (10) + CHR$ (28) : NEXT KA : PUT 31, 7 : ?
"NAAM VAN KLADPAPIERTJE? " : KFS : PUT 28, 9, 9, 9, 26 : LINPUT
( "" ) KFS : KFS = LEFTS (KFS, INSTR (KFS + ".", ".") - 1) + ".KLD" : OPEN
OUT#12, 12, KFS : ? #12, KAS : CLOSE #12 : PUT 31

```



OM

```

FOR i = 1 TO 255 : CLOSE #i : NEXT i : FOR j = 1 TO 100 : OPEN #0, 0,
"II04" : POKE 43, 145 : OPEN #15, 15 : ? #15, 3, "A:*,*" : FOR i = 1 TO
100 : LINPUT #15, a$ : ? LEFTS (a$ + " ", 16) : PUT -13 *
(INT (i / 5) = (i / 5)) : ? #15, 4 : i = i - 100 * (a$ = "") : NEXT i : ? #15, 6 :
CLOSE #15 : ? : ? "welke file veranderen (leeg = terug naar opnieuw
directory weergeven)? O-" + CHR$ (8) + CHR$ (8) : LINPUT ( "" ) f$ : f$ =
LEFTS (f$, INSTR (f$ + " ", " ") - 1) : j = j - 100 * (f$ <> "") : FOR k = (j >
100) TO -1 : OPEN #15, 15 : ? #15, 3, "A:" + f$ : LINPUT #15, a$ : CLOSE
#15 : j = j + 100 * (a$ = "") : NEXT k : NEXT j : PUT 31, 13 : OPEN IN#13,
13, f$ : FOR i = 1 TO 100000 : GET #13, q : PUT -27 * (q <> 13), -q * (q <>
26 AND q <> 0) : i = i - 100000 * (q = 26 OR q = 0) : NEXT i : CLOSE #13 :
OPEN #5, 5 : PUT 0, 12 : ? "type het programma in (graphics(m) = klaar) : "
: PUT 6 : FOR i = 1 TO 100000 : GET #5, q : PUT -q * (q <> 141) : i = i -
100000 * (q = 141) : NEXT i : b$ = "" : PUT 12, 10, 7 : FOR i = 2 TO 100 :
PUT 5 : LINPUT ( "" ) a$ : b$ = b$ + a$ + LEFTS (CHR$ (13), LEN (a$)) :
NEXT i : CLOSE #5 : PUT 31 : ? "filename (kort) (leeg = niets saven) : " +
f$ : PUT 22, 38, 1 : LINPUT ( "" ) f$ : f$ = LEFTS (f$, INSTR (f$ + " ", " ") -
1) : FOR i = 1 TO -(f$ <> "") : OPEN OUT#13, 13, f$ : ? #13, b$ : CLOSE
#13 : NEXT i : OPEN #0, 0, "s255" : POKE 43, 144 : ? b$ : END

```

verkrijgbaar op cassette
zie pagina 14

Patrick Vercammen

animatie



LEER DOORTJE STOOT DANSEN

ofwel: bewegende beelden

De fraaie plaat van de heer Koolstra in de HCCN 63 was voor Pa en mij aanleiding om te demonstreren, hoe je op de NewBrain hi-res beelden kunt laten bewegen. De truc is, dat je meerdere schermen achter elkaar kunt openen, stuk voor stuk voltekenen en daarna beurtelings oproepen, zodat een tekenfilmidee ontstaat.

De hierna volgende list is een aanvulling op die van de HCCN 63 en tekent Doortje in een iets andere stand.

Om een en ander nader te verklaren eerst een toelichting.

HET OPENEN VAN SCHERMEN

Je kunt theoretisch zoveel schermen openen als het geheugen toelaat. Elk nieuw scherm heeft een eigen getal nodig. Dit komt op de derde plaats achter OPEN te staan (bij het eerste scherm, meestal 0, hoeft dit niet). Bij voorbeeld: OPEN #0, 4, "100" : OPEN #1, 4, 1, "100" : enz.

Om op deze schermen naar willekeur te kunnen werken, hoef je alleen maar #p te zetten (p is het eerste getal van het scherm), bij voorbeeld: PRINT #1, "Hallo".

Om nu een plotscherf te verkrijgen, over deze tekstscherf heen, kun je eerst gewoon OPEN #2, 11, "50" doen. Dit komt dan op scherm 0. Voor elk willekeurig ander plotscherf moet je eerst het derde getal verhogen en dan vertellen in welk scherm het komt: bij voorbeeld OPEN #3, 11, 1, "#1n50". Hierbij staat n voor normaal. Als daar in de plaats w staat, wordt het scherm met 20 % versmald (wat geheugenruimte bespaart). Het getal, dat achter de 11 staat, geldt ook alleen voor OPEN #x, 11 schermen; het mag dus hetzelfde zijn als elke willekeurige andere OPEN #x,y, als y maar geen 11 is, zodat je op een OPEN #x, 11 scherm een keer eenzelfde derde getal hebt (je krijgt dan een foutmelding).

Het gedeelte . . . "#1n50" dient om het plotscherf op dat geopende scherm te krijgen (dit getal moet dan bij OPEN #x, 4 gelijk zijn aan x, dus niet gewoon aan het zoveelste scherm).

Op regel 13 wordt het eerste plotscherf op scherm 0 en op regel 14 het tweede plotscherf op scherm 3 geopend. In regel 14 staat PUT #0, 12 om het eerste scherm zichtbaar te maken. In regel 50 gebeurt voor plotscherf 3 hetzelfde met PUT #3, 12. Anders wordt Doortje getekend zonder dat je het ziet.

In regel 80 worden de schermen 0 en 3, uiteraard met een wachtloop, opgeroepen.

Voor het programma DOORTJE kunnen we met een standaard NewBrain maar twee schermen openen i.v.m. de geheugenruimte. Bij een wat kleinere tekening kun je drie of meer schermen krijgen, wat gedemonstreerd wordt in het navolgend afgedrukte programma MOLFENS, waarbij je een vloeiender beeld krijgt door de vier verschillende standen van de molen.

Arno Roelofs

LISTING VAN HET PROGRAMMA "DOORTJE"



```
2 REM *** DOORTJE STOOT HCC 63 ***
5 FORx=1TO255 CLOSE#x NEXTx
12 RESTORE
13 OPEN#0,4,"s150":OPEN#2,11,"#0n220"
14 OPEN#3,4,1,"s150":OPEN#4,11,1,"#3n220" PUT#0,12
15 plotrange(1920,1260):x2=10:y2=200
18 x=0:y=0:RESTORE 122:11=872:a=2:GOSUB30:GOTO37
30 FORi=1TO11
31 READa$,x1,y1
32 IFa$="m"THEN x=x1+x2:y=y1+y2:plot#a,pla(x,y):GOTO36
33 IFa$="j"THENx=x+x1:y=y+y1:plot#a,mve(x,y):GOTO36
34 x=x+x1:y=y+y1:plot#a,pla(x,y)
36 NEXTi:RETURN
37 plotrange (640,420):IF x2=150THEN END
38 x2=150:y2=-570:RESTORE122:GOSUB30
50 x=0:y=0:11=184:RESTORE a=4:PUT#3,12:plot#4,range(1920,1260):x2=10:y
2=180
```

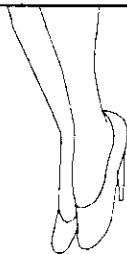
```
60 GOSUB30
70 x=0:y=0:11=710:RESTORE230:GOSUB30
75 plot#4,range (640,420):x2=150:y2=-570:RESTORE122:11=872:GOSUB30
80 PUT#0,12:FORi=1TO100:NEXTi:PUT#3,12:FORi=1TO100:NEXTi:GOTO80
100 REM data voor tweede plaatje
101 REM linker schoen
102 DATA m,310,-84,j,4,-40,j,6,0,j,0,70,j,-2,10,j,-6,10,j,-8,10,j,-2
0,10,j,8,-10,j,3,-10,j,0,-10,j,-1,-10,j,-3,-20,j,-6,-20,j,-10,-20,j,-
6,-10,j,-14,-10,j,-4,-10,j,-4,-10,j,-3,-10,j,-3,-10,j,0,-10,j,2,-5,j,6
,-5,j,6,0,j,5,3,j,5,5,j,10,10,j,8,10,j,4,10,j,4,10,j,3,10,j,3,10,j,2,1
0,j,3,10,j,3,10,j,3,10,j,10,30
103 DATA m,288,-30,j,-10,30
104 DATA m,257,-135,j,-1,10,j,1,83
105 REM rechter schoen
106 DATA m,260,84,j,21,-40,j,4,4,j,-28,65,j,-2,6,j,-2,5,j,-10,5,j,-10
,5,j,-20,0,j,10,-10,j,10,-20,j,5,-10,j,5,-20,j,3,-20,j,-1,-20,j,-3,-10
,j,-5,-10,j,-6,-2,j,7,-28,j,7,-20,j,7,-8,j,10,2,j,4,5,j,6,10,j,2,10,j,1
,10,j,0,10,j,-1,10,j,-1,10,j,-2,10,j,-4,10,j,-2,10,j,-1,10,j,-2,10,j,
-2,10,j,-1,30
107 REM voorkant linker been
108 DATA m,233,10,j,-10,10,j,-6,10,j,-4,10,j,-3,10,j,-4,10,j,-10,20,
j,-10,20,j,-20,30,j,-20,30,j,-90,150,j,-8,16,j,1,5,j,4,6,j,30,20,j,30,
21,j,30,23,j,30,22,j,30,17,j,30,14,j,20,5
109 DATA r,-60,-25,j,-13,20
110 REM rechter dij
111 DATA m,111,384,j,-1,16,j,2,20
112 DATA j,1,25,j,3,25,j,5,25,j,8,28,j,8,15,j,7,10,j,8,10,j,10,10,j,1
1,10
113 DATA j,15,10,j,7,5,j,4,5,j,2,5,j,1,15
114 REM binnenkant rechter been
115 DATA m,228,144,j,-5,20,j,-3,20,j,-3,20,j,-2,20,j,-1,20,j,0,20,j,1
,40,j,1,38
116 REM achterkant linkerbeen
117 DATA m,218,123,j,-10,15,j,0,5,j,-2,10,j,-1,10,j,-2,10,j,-2,10,j,
-4,10,j,-3,10,j,-4,10,j,-4,10,j,-4,10,j,-3,10,j,-4,10,j,-5,10,j,-4,10,
j,-5,10,j,-8,10,j,-10,10,j,-10,10,j,-12,8,j,10,2,j,10,1,j,10,2,j,10,2,
j,30,5,j,30,7,j,30,10,j,20,6,j,10,5,j,10,5,j,10,6,j,10,6,j,10,6,j,5,5,
j,5,5,j,4,5,j,3,5,j,2,5,j,1,5,j,1,10,j,1,10,j,2,10,j,0,10,j,-2,15
118 DATA j,-3,15,j,-6,20,j,-2,10,j,-1,10,j,0,10,j,2,10
119 DATA j,8,15,j,17,25,j,16,25,j,5,10,j,4,10
122 DATA m,235,0,j,9,-20,j,5,-10,j,1,-10,j,0,-20,j,-1,-20,j,0,-45
123 DATA j,1,-5,j,5,-10,j,5,-3,j,5,2,j,5,4,j,3,7,r,-24,5,j,-3,-15,j,-4
,-20
```

```

124 DATA j,-1,-5,j,0,-5,j,1,-5,j,3,-5,j,5,-2,j,2,0,j,5,2,j,5,5,j,8,15,
j,5,15
125 DATA m,261,0,j,5,-20,j,1,-20,j,0,-10,j,-2,-20,j,0,-10,j,1,-15,j,
3,-18,j,2,-7
126 DATA j,3,-5,j,5,0,j,5,2,j,5,3,j,7,5,j,6,10,j,5,15,j,3,20,j,2,10,j,
6,20
127 DATA j,0,10,j,-4,14,j,-5,8,j,-9,8,j,10,-2,j,10,-3,j,10,-7,j,4,-8,
j,1,-5
128 DATA j,0,-10,j,-1,-10,j,-8,-25,j,-2,-10,j,-2,-20,j,-3,-25,j,-2,-1
5,j,-6,-15
129 DATA j,-12,-15,j,-10,-6,j,-5,0,j,-5,1,j,-3,5,j,-2,10,j,0,10,j,-2,
20,j,-2,10
130 DATA r,53,45,j,3,-25,j,1,-15,j,3,0,j,2,3,j,0,65
139 DATA m,235,0,j,-50,100,j,-7,25,j,-5,25,j,-3,25,j,-1,25,j,1,25
140 DATA j,0,10,j,-1,10,j,-4,10,j,-5,10,j,-37,75,j,-7,15,j,-5,15,j,-5
,35,j,0,10
150 DATA j,1,25,j,3,25,j,5,25,j,8,25,j,8,15,j,7,10,j,6,10,j,10,10,j,1
1,10
151 DATA j,15,10,j,7,5,j,4,5,j,2,5,j,1,15
160 DATA m,261,0,j,-6,25,j,-7,25,j,-20,50,j,-19,50,j,-15,49,r,4,-5,j,
-22,31
170 DATA j,-5,10,j,-2,10,j,0,10,j,1,10,j,2,10,j,30,165,j,15,10,j,25,1
0,j,20,6
171 DATA r,-60,-26,j,-5,15,j,-3,15,j,-1,15
180 DATA m,310,-20,j,-9,20,j,-5,20,j,-6,30,j,0,20,j,4,60,j,0,10,j,-1
,10,j,-3,15
190 DATA j,-10,25,j,-14,25,j,-17,25,j,-8,10,j,-10,12
200 DATA j,15,18,j,15,20,j,34,50,j,15,25,j,10,25,j,9,25,j,8,25,j,1,15
,j,0,10
210 DATA j,-3,15,j,-6,20,j,-2,10,j,-1,10,j,0,10,j,2,10
220 DATA j,8,15,j,17,25,j,16,25,j,5,10,j,4,10
230 DATA m,278,670,j,0,-10,j,3,-10,j,5,-10,j,8,-10,j,8,-7,j,10,-5,j,
10,-3
235 REM regel 240 t/m1060 als HCCN 63, bladzijde 20 vv

```

verkrijgbaar op cassette
zie pagina 14

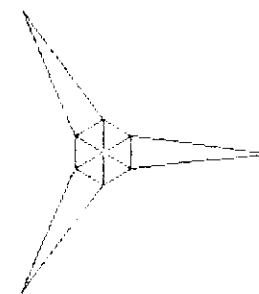
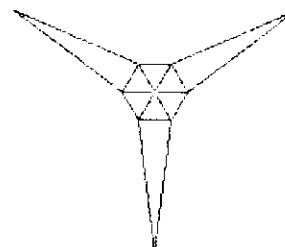


LISTING VAN HET PROGRAMMA "MOLENS"

```

10 OPEN#0,4,"70" CLOSE#1 OPEN#1,11,"#0n100"
20 CLOSE#2 OPEN#2,4,2,"70" CLOSE#3 OPEN#3,11,3,"#2n100"
30 CLOSE#4 OPEN#4,4,4,"70" CLOSE#5 OPEN#5,11,5,"#4n100"
40 CLOSE#6 OPEN#6,4,6,"70" CLOSE#7 OPEN#7,11,7,"#6n100"
50 DIM x(100),y(10),i(10)
70 x=50:y=50:b=0:w=0 ww=360 ws=60:q1=1
80 FOR i=1 TO 7 STEP 2:q=0:plot#i,rng(256,100),cen(0,0),col(1),deg:PI*T#i-1,
12:IF b(<)0 THEN b=b+ws/2
90 FOR j=b+w TO ww+b STEP ws

```



```

100 plot#i,pla(x,y),trn(j),mby(10):x(q)=pen#i,0:y(q)=pen#i,1:q=q+1
110 NEXT j:x(9)=x(0):y(9)=y(0):q=0
120 FOR j=b+30+w TO ww+b STEP ws*2
130 plot#i,pla(x,y),col(2),trn(j),mby(50),col(1):xx=pen#i,0:yy=pen#i,1:
plot#i,pla(xx,yy),mve(x(q),y(q)),pla(xx,yy)
135 IF q=0 AND q1=-1 THEN plot#i,mve(x(9),y(9)):q=q+2 GOTO 140
136 plot#i,mve(x(q+q1),y(q+q1)):q=q+2
140 NEXT j:plot#i,pla(x(0),y(0))
150 FOR j=1 TO 5:plot#i,mve(x(j),y(j)):NEXT j:plot#i,mve(x(0),y(0))
160 b=b+ws:NEXT i
165 IF a=0 THEN a=1 b=360:x=50*4:w=360 ww=0 ws=-60:q1=-1:GOTO 80
170 FOR i=1 TO 1000:FOR j=0 TO 6 STEP 2 PUT#j,12 FOR h=1 TO 60 NEXT h:NE
X Tj:NEXT j
200 END

```

verkrijgbaar op cassette
zie pagina 14

plot circle

Het plotten van cirkels en cirkelbogen op het high-resolution scherm is mogelijk met de plotarc-opdracht. Dit is een zeer moeilijke opdracht en het resultaat is dikwijls anders dan je verwacht. Het plotten van een cirkel met een gegeven middelpunt en straal is met het operating system 2.0 of met device 33 op de Expanded NewBrain aanzienlijk eenvoudiger met de plotcircle-opdracht. Voor de bezitters van een 32k NewBrain met operating system 1.91 of lager heb ik een pseudo-plotcircle-opdracht in BASIC gemaakt, die je in een programma kan opnemen.

De volgende regels moet je in je programma opnemen:

```
20 DEFFNr=2*PI*r:DEFFNp=2*PI:DEFFNy=y-r
50 plotpla(x,FNy),trn(0),arc(FNr,FNp):REM cirkel(x,y,r)
```

De regelnummers zijn willekeurig en kan je naar keuze wijzigen. Wanneer je nu in je programma aan de variabelen x en y de waarden toekent van de coördinaten van het gewenste middelpunt van de cirkel en aan de variabele r de waarde voor de gewenste straal, dan zal de plot-opdracht in regel 50 de gewenste cirkel tekenen.

Het volgende demonstratieprogramma maakt een en ander duidelijk.

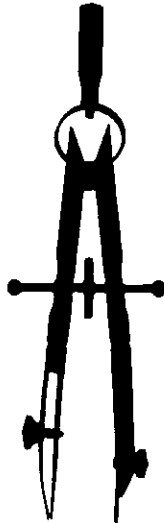
```
10 OPEN#0,0,"200":CLOSE#1:OPEN#1,11
15 plotrng(202,102),cen(101,51),pla(0,0),axe(10,10)
20 DEFFNr=2*PI*r:DEFFNp=2*PI:DEFFNy=y-r
30 INPUT("straal r ")r
40 INPUT("middelpunt x,y ")x,y
50 plotpla(x,FNy),trn(0),arc(FNr,FNp)
60 END
```

Gebruik je niet radialen (de default) om hoeken in uit te drukken, maar graden dan moet je regel 20 als volgt wijzigen:

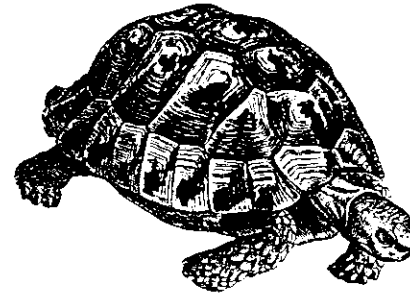
```
20 DEFFNr=2*PI*r:DEFFNp=360:DEFFNy=y-r
```

Geen echte plotcircle, maar het resultaat is te voorspellen.

Wim Luijt



langzaam



HOE SNEL IS LANGZAAM
EN HOE GROOT IS 0?

In Personal Computer World van juli 1985 wordt de snelheid en nauwkeurigheid van de BASIC van een aantal populaire home computers vergeleken. De NewBrain is hier niet bij. Daarom heb ik de beschreven test-"programma's" uitgevoerd en vergeleken met de geteste computers. Bovendien heb ik van verschillende Operating Systems in de NewBrain de nauwkeurigheid onderzocht.

De geteste computers zijn:

type	proc.	MHz	
Spectrum 48k	Z80A	3,6	
Sharp MZ-80K	Z80	2	
Sharp MZ-700	Z80A	3,6	
Olivetti M20	Z8001	4 ?	16 bits!
BBC B	6502	1	
Commodore 64	6510	1	
PDP 11/34	?	?	16 bits!
Apple II	Z80A	2	
Spectravideo SV328	Z80A	3,6	

De snelheid is getest met het volgende BASIC-programma:

```
10 DIM X(20), Y(20), A(20, 20), B(20, 20)
20 K = 5 : Y(15) = 7 : B(5, 10) = 3 : CALL 62383 : REM zet klok op 0
30 FOR I = 1 TO 1000
40 REM opdracht, zie onder
```

```

50 NEXT I
60 CALL 62399, Z : PRINT Z / 50 : END : REM Z / 50 = tijd in sec
200 RETURN

```

Om de invloed van de compileertijd uit teschakelen werd dit programma herhaald, waarbij regel 30 gewijzigd werd in:

```

30 FOR I = 1 TO 2000

```

Het verschil in tijd, gecorrigeerd voor de tijd voor het uitvoeren van de FOR-NEXT-loop, is als maat genomen voor het uitvoeren van de afzonderlijke opdrachten. De compileertijd bedroeg, afhankelijk van de opdracht, 16 tot 42 msec en is te verwaarlozen. De gebruikte opdrachten zijn in tabel 1 vermeld.

De test is uitgevoerd op een NewBrain met Operating System 1.9 gekoppeld aan een Expansion interface, nadat het programma UNEXPAND.BAS geRUNd was, zodat het als een 32k NewBrain met O.S. 1.9 is te beschouwen. Op een 96k NewBrain kon het programma niet geRUNd worden, omdat de klok niet werkt (wie weet hoe dit moet?).

De nauwkeurigheid is getest met het volgende BASIC-programma:

```

10 DEF FNY = formule
20 M = 0 : PI = 3,14
30 FOR X = laag TO hoog STEP interval
40 Y = ABS (FNY)
50 IF Y > M THEN M = Y : J = X
60 NEXT X
70 PRINT "Error is"; M; "for X ="; J
80 END

```

De gebruikte formules en de waarden voor laag, hoog en interval staan in tabel 2. De uitkomst van de formules moet 0 zijn, zodat de gevonden waarde een maat is voor de onnauwkeurigheid (hoe te groter de waarde des te onnauwkeuriger is de routine). De nauwkeurigheid is getest met een 32k NewBrain met O.S. 1.9 en 1.91 en met een 96k NewBrain.

Om de computers te kunnen rangschikken op hun 'gemiddelde' nauwkeurigheid is er per formule een rangnummer aan iedere computer toegekend, waarbij de computer met de geringste afwijking een 1 en die met de grootste afwijking een 9 kreeg. Als meerdere computers dezelfde afwijking hebben voor een formule, dan is het gemiddelde rangnummer toegekend. De score in tabel 2 vormt de som van de rangnummers per computer.

TABEL 1 snelheden [msec]	Sharp MZ-80K	Sharp MZ-700	NewBrain	Apple II met Microsoft BASIC	Spectrum 48k	Spectravideo SV328	BBC B 64	Commodore MZ0	Olivetti MZ0	PDP 11/34
computer	1	2	3	4	5	6	7	8	9	10
FOR-NEXT	1.4	1.7	2.1	2.5	5.8	2.5	0.7	1.9	1.0	1.1
X=5	2.7	1.0	0.8	2.0	2.3	1.9	0.7	1.5	0.7	0.3
X=K	1.1	1.7	1.2	2.0	3.0	1.3	0.6	1.3	0.6	0.2
X=Y(15)	4.9	2.8	2.3	3.2	4.6	2.1	1.1	3.9	1.2	0.7
X(7)=K	4.8	2.5	2.3	2.5	5.1	1.8	0.9	3.9	1.1	0.7
X(7)=5	6.0	2.1	1.9	2.6	3.5	2.2	1.0	4.2	1.2	0.9
X(7)=Y(15)	8.6	3.9	3.4	3.7	6.0	2.4	1.3	16.5	1.7	1.4
A(1,12)=B(5,10)	13.7	6.1	4.8	6.9	9.4	4.7	2.6	14.1	2.7	3.0
A(1,12)=K	7.2	3.7	3.0	4.3	6.0	2.7	1.6	7.8	1.6	1.6
A(1,12)=5	8.6	3.2	2.6	4.3	5.4	3.1	1.7	8.3	1.7	1.8
gemiddeld	6.4	3.0	2.5	3.5	5.0	2.5	1.4	5.7	1.4	1.2
X=5+6	5.3	1.7	1.8	2.5	3.1	2.3	0.9	3.2	1.1	0.8
X=5-4	5.5	1.8	1.7	2.7	3.3	2.3	1.0	3.6	1.1	0.9
X=5*4	5.9	2.0	2.9	3.8	3.5	2.5	1.1	4.0	1.1	0.9
X=5/4	6.6	2.6	5.4	6.5	5.4	6.7	2.6	4.9	1.5	0.7
gemiddeld	5.8	2.0	2.9	3.9	3.8	3.4	1.4	3.9	1.2	0.8
X=SQR(5)	12.2	32.8	19.1	49.0	118.0	164.0	10.2	55.1	2.0	1.7
X=5^2	45.7	32.0	4.7	49.0	113.3	2.7	4.0	53.7	1.8	1.4
X=ABS(5)	3.3	1.6	1.0	2.7	3.3	2.2	0.9	2.5	1.0	0.5
X=LOGe(5)	24.6	17.7	29.9	24.5	69.4	128.5	17.6	24.3	2.9	0.9
X=EXP(5)	22.4	17.1	22.8	23.9	44.1	105.5	12.4	27.4	1.0	0.9
X=SIN(5)	22.2	20.1	22.8	21.5	49.2	89.5	30.3	30.0	2.6	1.0
X=COS(5)	21.6	19.5	22.3	22.5	50.2	94.5	19.2	28.9	2.7	0.9
X=TAN(5)	42.8	39.7	49.8	45.5	92.2	197.5	45.0	53.6	4.5	2.6
X=ATN(.5)	20.2	19.5	41.1	27.5	66.2	119.5	22.7	43.5	2.7	1.5
gemiddeld	23.9	22.2	23.8	29.6	67.3	100.4	18.0	35.4	2.4	1.3
A\$="A"	51.8	1.2	4.4	1.9	3.6	1.5	0.5	1.1	0.7	0.4
X=VAL("1")	4.7	2.7	3.7	4.5	7.8	2.7	1.0	2.0	1.8	1.2
A\$=CHR\$(64)	54.2	2.1	4.3	3.0	5.8	2.0	0.8	4.1	1.1	0.9
X=ASC("1")	2.2	1.7	1.9	3.5	2.9	2.3	0.8	2.0	1.1	0.9
A\$=STR\$(1)	59.2	5.2	21.6	4.7	13.6	3.9	6.6	11.1	1.5	2.2
A\$=INKEY\$	3.5	1.6	5.1	2.4	3.6	1.2	1.2	1.4	0.8	0.6
gemiddeld	29.3	2.4	6.9	3.3	6.2	2.3	1.8	3.6	1.2	1.0

VERVOLG TABEL 1: snelheden [msec]										
computer	1	2	3	4	5	6	7	8	9	10
CLS *)	32.3	67.1	20.1	17.0	64.2	18.5	4.2	43.3	31.4	?
GOSUB200	3.0	0.6	1.2	1.2	2.4	0.7	0.4	1.4	0.6	0.1
GOTO50	1.7	0.2	-0.	0.7	1.4	0.5	0.2	0.6	0.2	0.1
IFK=14THENX=1	3.0	1.7	1.8	3.3	3.0	2.8	1.0	3.7	1.4	0.7
PLOT3,4	5.8	1.9	?	2.0	2.4	2.0	1.0	6.5	2.5	?
X=RND	4.7	2.3	3.9	4.5	15.6	2.1	1.5	6.3	0.7	0.5
X=PEEK(32000)	1.9	1.8	1.3	2.7	3.3	2.5	1.2	6.9	?	?
POKE32000,0	3.6	1.9	0.9	2.0	3.0	?	1.0	6.5	?	?
gemiddeld	3.4	1.5	1.8	2.3	4.4	1.5	0.9	4.6	1.1	0.4
totaal gem.	13.8	7.5	8.4	10.0	20.5	27.6	5.7	12.5	1.5	1.0

*) niet in gemiddelde

TABEL 2:					
computer			1	2	
formule	laag	hoog step			
1 TAN(ATN(X))-X	0,1	0,01	1.5E-08	2.8E-09	
2 TAN(ATN(X))-X	1,50	0.5	1.2E-06	4.9E-07	
3 TAN(ATN(X))-X	50,100	10	4.5E-04	1.7E-04	
4 ATN(TAN(X))-X	-PI/2,PI/2	0.01	0	2.1E-09	
5 SIN(X)*SIN(X)+COS(X)*COS(X)-1	0,100	1	5.0E-08	2.4E-08	
6 ABS(SIN(X))-SQR(1-COS(X)*COS(X))	0,100	1	4.0E-07	3.1E-08	
7 ABS(COS(X))-SQR(1-SIN(X)*SIN(X))	0,100	1	1.2E-06	3.1E-08	
8 X-X/255*255	0,100	1	0	0	
9 X-(SQR(X)+2)	0,100	1	0	7.2E-07	
10 X-SQR(X+2)	0,100	1	0	4.8E-07	
11 X-SQR(X)*SQR(X)	0,100	1	0	5.1E-07	
12 X-SQR(X*X)	0,100	1	0	5.1E-07	
13 X*X-X+2	0,100	1	0	4.6E-05	
14 SQR(X)-.5*(SQR(X)+X/SQR(X))	1,100	1	0	4.6E-05	
15 X-LOGE(EXP(X))	1,100	0.1	0	3.7E-09	
16 X-EXP(LOGE(x))	1,100	0.1	0	1.9E-08	
score			60	94	
computer 3 = NewBrain 96k;					

TABEL 3: nauwkeurigheid 32k NewBrain met O.S. 1.9 [-]			
1 1.6E-10	5 2.8E-10	9 3.8E-08	13 0
2 3.2E-07	6 9.7E-06	10 1.9E-08	14 2.1E-09
3 2.8E-07	7 3.2E-08	11 3.8E-08	15 4.7E-09
4 2.3E-10	8 9.1E-13	12 1.9E-08	16 4.6E-08

De negatieve waarde voor de opdracht GOTO 50 in tabel 2 komt waarschijnlijk, doordat de FOR-NEXT-lus sneller wordt afgehandeld. Bij o.a. de opdracht AS = "A" duurde de afhandeling tijdens de lange loop (2000 x) meer dan 2 x zo lang als tijdens de korte loop (1000 x), wat waarschijnlijk veroorzaakt wordt door het opvullen van het geheugen met string-variabelen, een bekende bug van de NewBrain.

Uit de vergelijking van tabel 1 met tabel 2 blijkt, dat de meeste geteste

nauwkeurigheid [-]							
computer	3	4	5	6	7	8	9
1	1.6E-10	1.2E-07	1.2E-09	2.0E-10	1.6E-09	4.7E-10	1.8E-07
2	3.2E-07	5.0E+01	8.0E-07	1.0E-09	6.6E-07	7.5E-07	3.7E-04
3	2.8E-07	1.0E+03	3.9E-04	7.8E-08	3.2E-06	2.4E-04	1.5E+00
4	2.3E-10	7.2E-07	7.0E-10	9.6E-11	7.0E-10	4.7E-10	1.2E-07
5	2.8E-10	5.6E-06	9.3E-10	3.0E-14	4.7E-10	2.2E-08	1.2E-07
6	9.7E-06	7.1E-06	2.3E-08	1.4E-12	1.5E-08	2.2E-08	8.9E-07
7	3.2E-08	1.7E-05	2.2E-08	1.4E-12	3.5E-08	3.3E-08	3.7E-06
8	9.1E-13	0	3.0E-08	2.0E-12	0	0	7.6E-06
9	2.1E-09	6.1E-05	1.8E-07	4.0E-12	6.0E-08	2.1E-07	7.6E-06
10	0	3.1E-05	1.8E-07	0	0	7.3E-08	0
11	2.1E-09	3.8E-05	1.2E-07	4.0E-12	6.0E-08	5.0E-08	7.6E-06
12	0	4.6E-05	1.5E-07	0	0	3.4E-08	0
13	0	4.9E-03	2.7E-05	0	0	6.4E-06	0
14	2.3E-10	2.9E-06	1.1E-08	0	0	3.7E-09	9.5E-07
15	4.7E-09	2.9E-06	3.7E-09	5.0E-13	3.7E-09	4.8E-10	6.0E-08
16	4.6E-08	4.8E-07	7.4E-09	2.0E-12	7.5E-09	5.1E-09	9.5E-07
score	62	133	87.5	36	61.5	73	114
overige cijfers zie tabel 1							

computers een compromis vormen tussen snelheid en nauwkeurigheid; een snelle computer werkt onnauwkeurig en omgekeerd. Kijken we alleen naar de wiskundige opdrachten, dan geldt dit ook, maar is het minder duidelijk.

De frekwentie van de verschillende computers met een Z80A (zoals onze NewBrain) blijkt geen invloed te hebben op de relatieve snelheid van de BASIC. De NewBrain blijkt een redelijk snelle computer te zijn met een (voor mij) acceptabele nauwkeurigheid. Alleen de BBC B scoort hoger wat betreft snelheid terwijl de nauwkeurigheid gelijk is.

Hoewel niet getest moet vermeld worden, dat sommige van de geteste BASICs ook een 'double precision'-versie kennen, welke de nauwkeurigheid van een aantal bewerkingen kan verbeteren. Ook Microsoft-BASIC kent deze optie, zodat in voorkomende gevallen deze double precision ook voor een 96k NewBrain onder CP/M beschikbaar is.

Bij vergelijking van tabel 2 en 3 blijkt, dat de nieuwe routine voor worteltrekken in O.S. 1.91 en hoger en in de Expansie nauwkeuriger zijn dan die in vroegere O.S. Bovendien blijkt de routine een fractie sneller te zijn.

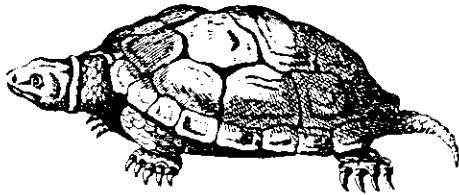
De suggestie van Rob Maris in On-line 4, dat de routines in het maths-pack van de NewBrain traag zouden zijn, blijkt niet uit de gevonden testresultaten. Bovendien zullen snellere routines, zoals hij voorstelt, waarschijnlijk tot onnauwkeuriger resultaten leiden. Dat wil niet zeggen, dat een aantal van deze routines niet voor verbetering vatbaar is. Een duidelijk voorbeeld hiervan is de formule $X - X / 255 * 255$. Omkering van de vermenigvuldiging en de deling $(X - X * 255 / 255)$ geeft wel 0 als resultaat. Of het resultaat 0 bij de andere computers veroorzaakt wordt door een betere afhandelingsroutine of door het beter oversteken van de (ezels)brug Meneer Van Dalen Wacht Op Antwoord, weet ik niet, maar bij de NewBrain is het laatste kennelijk zwak.

Moraal: de NewBrain scoort niet slecht in vergelijking tot andere computers. De snelheid en de nauwkeurigheid kun je in een programma verhogen door bijv.:

$x = a * .5$ in plaats van $x = a / 2$
 $y = x * 255 / 255$ in plaats van $y = x / 255 * 255$

Doe er je voordeel mee.

Wim Luijt



screendump



In dit stukje zal ik proberen uit te leggen, waar de gegevens van het grafische scherm te vinden zijn, en een listing geven van een screendump-routine in machinetaal. Deze routine is geschreven voor een Star DP-510 printer, maar kan waarschijnlijk gemakkelijk aan andere printers worden aangepast.

Eerst iets over het grafische scherm van de NewBrain. In de nummers 84-2 en 85-4 van On-line worden door S. Knijpstra en P. Kramer de problemen bij het dumpen van het scherm naar de printer besproken. Het eerste probleem is het vinden van de gegevens van het scherm in het geheugen. Dit gaat (bij unexpanded machines) als volgt.

Alle geopende streams zijn opgeslagen in de stream table. Iedere stream neemt in deze tabel 6 bytes in beslag, die de volgende informatie bevatten:

- 1 byte: streamnummer
- 1 byte: devicenummer
- 1 byte: poortnummer
- 1 byte: wordt niet gebruikt
- 2 bytes: beginadres van het eigen geheugen

De bytes staan in deze volgorde in de tabel. Het beginadres van de stream table is te vinden op de adressen 86 en 87, dus $STRTAB = PEEK(86) + 256 * PEEK(87)$. STRTAB heeft nu de waarde van het adres waar het streamnummer van de eerste stream staat (meestal is dit stream 0, omdat deze als eerste geopend wordt). We vinden zo:

```
STRNR = PEEK (STRTAB)
DEVNR  = PEEK (STRTAB + 1)
PORTNR = PEEK (STRTAB + 2)
OWNMEM = PEEK (STRTAB + 4) + 256 * PEEK (STRTAB + 5)
```

OWNMEM bevat nu het beginadres van het eigen geheugen van stream STRNR. Voor het grafische scherm (device 11) is dit eigen geheugen 220

bytes lang en het bevat alle informatie over de parameters van het scherm: range, centre etc., de PEN()-functies en breedte, hoogte, begin en einde van het scherm:

OWNMEM + 25: breedte van het scherm in bytes
OWNMEM + 26: breedte in puntjes (pixels) (2 bytes)
OWNMEM + 28: hoogte in pixels
OWNMEM + 42: beginadres van het scherm (2 bytes)
OWNMEM + 44: eindadres + 1 (2 bytes)

Dan is er nog een ander probleem. In het geheugen vormen 8 puntjes naast elkaar een byte. De printer werkt echter met 8 puntjes boven elkaar (minst significante bit beneden). Een eenvoudige oplossing is, om het scherm op zijn kant af te drukken (P. Kramer, On-line 85-4). Om het scherm zonder knip- en plakwerk horizontaal te krijgen, moeten de bytes dus omgezet worden (S. Knijpstra, On-line 84-2). Het onderstaande programma doet dit voor elk willekeurig scherm. De routine staat op de regels 40000 - 40070 en moet bij het hoofdprogramma geladen worden d.m.v. MERGE.

De routine gebruikt de variabelen 'zz', 'xx' en 'ca'. zz en xx worden alleen bij het laden gebruikt, ca bevat het adres waar de machinetaalroutine begint. Het is belangrijk, dat ca de juiste waarde heeft, omdat anders naar de verkeerde plaats in het geheugen wordt gesprongen, met als resultaat, dat de computer zeer waarschijnlijk op 'tilt' gaat. Verder is ook belangrijk, dat er een grafisch scherm aanwezig is (de routine controleert dit niet en de computer zal op 'tilt' gaan, wanneer dit niet het geval is). Er moet bovendien een stream 9 geopend zijn voor de uitvoer. Deze kan naar de printer, maar ook naar bij voorbeeld cassette of disk! De aanroep van de routine geschiedt met GOSUB 40000. Na afloop is de printer in de 'default'-toestand, het scherm blijft onaangetast.

De routine zelf tenslotte wordt ingekort door de DATA-regels te verwijderen. De machinetaalroutine blijft staan, totdat TOP wordt verhoogd (kan alleen met RESERVE). Let er wel op dat bij een nieuwe RUN ca weer de juiste waarde heeft. Wat er van het basic-gedeelte overblijft, zijn de statements voor de initialisatie en, na afloop, de 'reset' van de printer, en de CALL naar het machinetaalgedeelte. Overigens kan de machinetaalroutine op een willekeurige plaats in het geheugen worden gezet: de waarde van TOP voor het laden is niet belangrijk.

Wat moet er nu gebeuren om de routine aan te passen voor een andere printer?

Voor de Star DP-510 zijn er vier bytes nodig om over te gaan op de zogenaamde 'bit-image mode' (kolommetjes van acht puntjes onder elkaar): 27, 75, n1, n2 (480 kolommetjes per regel) of 27, 76, n1, n2 (960

kolommetjes per regel), waarbij $n = n1 + 256 * n2$ het aantal af te drukken kolommetjes is. Samen met een NEWLINE (ASCII-code 13) na iedere afgedrukte regel zijn dat 5 bytes per regel (in de listing zie regel 40035). Deze bytes worden meteen achter de routine in het geheugen geplaatst (regel 40042). Voor deze bytes moet ruimte gereserveerd worden. Het RESERVE-statement in regel 40010 moet zijn: RESERVE 202 + p (p is het aantal voor de besturing van de printer noodzakelijke bytes, p = 5 voor de Star DP-510). Het begin van de door de routine gebruikte buffer is TOP + 194 + p (regel 40044). De FOR-NEXT-loop in regel 40020 moet zijn: FOR zz = 1 TO (193 + p). De initialisatie van de printer vindt plaats in regel 40050 ('reset' en instellen regelafstand). Na afloop van de routine moet er een NEWLINE voor de laatst afgedrukte regel naar de printer worden gestuurd, waarna een 'reset' volgt. Het overgaan op de bit-image mode gebeurt in machinetaal, omdat dat voor iedere regel opnieuw moet gebeuren.

Nu moeten nog de nullen in regel 40042 vervangen worden. De eerste nul is voor de bit-image mode, de ander nullen moeten vervangen worden door n1 en n2 (zie boven). Dit gebeurt door het stukje machinetaal in regel 40033, in assembler:

```
LD    A, (IY + 27)
LD    (IX - 1), A
ADD    74
LD    (IX - 3), A
LD    A, (IY + 26)
LD    (IX - 2), A
```



In woorden staat er: vervang de laatste nul door PEEK (OWNMEM + 27), de eerste door PEEK (OWNMEM + 27) + 74 en de middelste door PEEK (OWNMEM + 26). Op de adressen OWMEM + 26 en OWMEM + 27 staat de breedte van het scherm in pixels en dat is precies het aantal kolommetjes per regel (n1, n2). Deze bytes worden dan ook rechtstreeks ingevuld. Door bij n2 74 op te tellen (ADD 74) krijgen we voor de Star-printer de juiste bit-image mode: 480/regel (code 75) bij 256 of 320 pixels en 960/regel (code 76) bij 512 of 640 pixels. Voor printers met opeenvolgende codes (zoals hier 75 en 76) hoeft alleen het getal 74 veranderd te worden, in andere gevallen zullen er enkele machinetaalinstructies bij moeten komen.

Voor degenen die enigszins thuis zijn in het machinetaalprogrammeren enkele aanwijzingen: het aantal bytes in regel 40033 kan willekeurig uitgebreid worden, maar houdt wel het RESERVE-statement etc. (zie boven) in de gaten. Verder is IY = OWMEM en IX is het adres, waar de buffer van de routine begint; de bytes voor de printer worden voor de buffer opgeslagen.

Heeft uw printer een ander systeem, waardoor u niet weet, hoe u de routine moet aanpassen, dan kunt u mij een (nauwkeurige!) beschrijving sturen. Ik zal dan proberen de routine voor u aan te passen. Zijn er veel reacties m.b.t. dezelfde printer, dan zal ik de (eventuele) oplossing naar On-line sturen.

```
40000 REM var: zz,xx,ca By H.C.H. Bailemans
40005 DEFFNhb(x)=x-256*FNhb(x):DEFFNhb(x)=INT(x/256)
40010 RESERVE207:DELETE40010
40020 CLOSE#255:OPEN#255,0,1,"s1":PRINT#255,"Loading...":RESTORE40
030.FORzz=1TO198:READxx:POKETOP+3+zz,xx:NEXTzz
40030 DATA221,229,253,229,221,33,0,0,42,86,0,1,6,0,35,62,11,190,40,
3,9,24,250,35
40031 DATA35,35,94,35,86,213,253,225,253,78,28,6,3,203,57,203,29,16,
250,12,6,5
40032 DATA203,61,16,252,62,0,169,32,3,46,8,13,253,94,42,253,86,43,22
1,117,8
40033 DATA253,126,27,221,119,255,198,74,221,119,253,253,126,26,221,1
19,254
40034 DATA62,0,185,40,99,213,213,197,30,9,1
40035 DATA 5
40036 DATA0,221,229,225,175,237,66
40037 DATA231,61,193,209,253,70,25,197,213,6,8,221,229,225,54,0,35,1
6,251
40038 DATA221,70,8,221,229,225,14,1,26,161,40,1,55,203,22,35,203,33
,48,244,197
40039 DATA6,0,253,78,25,235,9,235,193,16,227,221,229,225,30,9,1,8,0,
231,61,209
40040 DATA19,193,16,197,80,221,70,8,253,94,25,225,25,16,253,235,221,
54,8,8,13
40041 DATA24,152,253,225,221,225,201
40042 DATA 13,27,0,0,0
40044 POKETOP+7,FNhb(TOP+199):POKETOP+8,FNhb(TOP+199)
40045 CLOSE#255:PUT0:DELETE40005-40045
40050 ca=TOP+1:PUT#9,27,64,27,65,8
40070 CALLca:PUT#9,13,27,64:RET
```

H. Bailemans
Beatrixstraat 10
6267 BG Cadier en Keer

verkrijgbaar op cassette
zie pagina 14

O.S.-routines

OPERATING SYSTEM-ROUTINES

De NewBrain kent in zijn operating system een aantal routines welke in een machinecodeprogramma met RST 32 of RST 40 kunnen worden aangeroepen. In het Handbook, Appendix 8, worden er 33 genoemd en de Software Technical Manual beschrijft er ook een aantal zonder de calling code te vermelden. In de Technical Note nr 10 zijn er 67 beschreven, waarvan er een aantal waarschijnlijk alleen aanwezig is in de ROM van de Expansion.

Het doel van dit artikel is om een overzicht te geven van deze routines als aanvulling op Appendix 8, zodat je ze kunt toepassen, wanneer je in machine code programmeert. Bovendien beperkt het zich tot de routines, die aanwezig zijn in de 32k NewBrain met Operating System 1.4, 1.9 of 1.91. De extra routines in de Expansion en mogelijk in O.S. 2.0 zullen in een volgend artikel worden behandeld. Voor volledige informatie wordt verwezen naar de Technical Note en de Software Technical Manual.

De routines worden in 2 groepen onderscheiden:

- 1 mathematische routines
- 2 Operating System- en in- en outputroutines.

1a. M0-routines

Deze routines zetten het resultaat direct in de FACC (floating point accumulator op 100h - 105h) en behoeven behoeven geen parameter.

1b. M1-routines

Deze routines vereisen een parameter, welke zich bij aanroep van de routine in de FACC moet bevinden en waarvan het resultaat in de FACC komt te staan.

1c. M2-routines

Vereisen 2 parameters, waarvan de eerste in de FACC en het laagste byte van het adres van de andere parameter zich in DE moet bevinden. Het resultaat staat weer in de FACC.

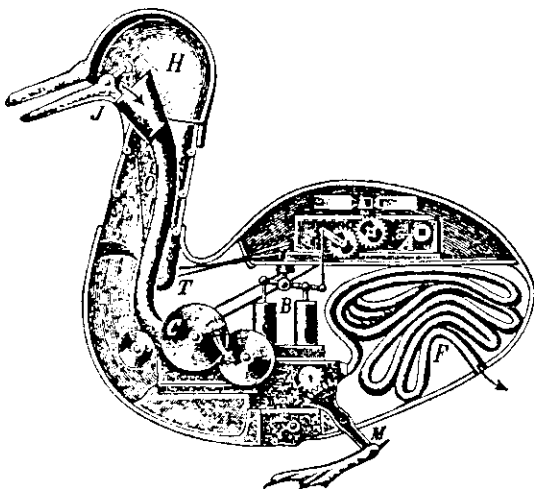
1d. MS-routines

Vereisen speciale aanroep (zie tabel).

MATHS-routine (23h)

Met de speciale routine MATHS kunnen de M0 - M2 routines worden aangeroepen. Bij aanroep moet in register A de Maths code (= nr in bijgevoegde lijst) staan en moet HL naar een stack area wijzen. Dit is vooral handig wanneer machine code routines vanuit BASIC met een CALL met parameters worden aangeroepen.

Bij de behandeling van een M0-routine wordt HL met 6 verminderd en wijst naar het resultaat; bij de aanroep van een M1-routine moet HL naar de parameter wijzen en wijst na afloop naar het resultaat wat zich op hetzelfde adres bevindt; bij een M2-routine moet HL bij aanroep naar de tweede parameter wijzen en moet de eerste parameter zich bevinden op adres HL + 6 en hoger, waar ook het resultaat na afloop te vinden is. In alle gevallen wijst HL na afloop naar het resultaat.



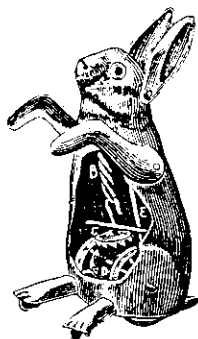
Calling code

hex	dec	naam	type/nr	resultaat
00	0	ZRND	M0:00	random getal
01	1	ZPI	M0:02	constante pi
02	2	ZSTPOS	M0:04	PFEK (28), printkoppositie #0
03	3	ZFPONE	M0:06	constante 1
04	4	ZFPMONE	M0:08	constante -1
05	5	ZFPZERO	M0:0A	constante 0
06	6	ZNOP	M1:80[F]	geen actie
07	7	ZNEG	M1:82	wissel teken (* -1) indien <> 0
08	8	ZNOT	M1:84	logische NOT *)

09	9	ZABS	M1:86	absolute waarde
0A	10	ZATAN	M1:88	arc tangens
0B	11	ZCOS	M1:8A	cosinus
0C	12	ZEXP	M1:8C	exponent (e tot de macht)
0D	13	ZINT	M1:8E	grootste integer <=
0E	14	ZLOG	M1:90	natuurlijke logaritme
0F	15	ZSIGN	M1:92	teken: +1, 0 of -1
10	16	ZSIN	M1:94	sinus
11	17	ZSQRT	M1:96	vierkantswortel
12	18	ZTAN	M1:98	tangens
13	19	ZASIN	M1:9A	arc sinus
14	20	ZACOS	M1:9C	arc cosinus
15	21	ZPEEK	M1:9E	waarde van het byte op het door HL aan- gegeven adres
16	22	ZADD	M2:40	som
17	23	ZSUB	M2:42	verschil (parameter1 - parameter2)
18	24	ZMULT	M2:44	produkt
19	25	ZDIV	M2:46	quotient (par.1 / par.2)
1A	26	ZRAISE	M2:48	macht (par.1 ↑ par.2)
1B	27	ZAND	M2:4A	logische AND *)
1C	28	ZOR	M2:4C	logische OR *)
1D	29	ZNUMEQ	M2:4E	TRUE als parameters gelijk, anders FALSE
1E	30	ZNUMGT	M2:50	TRUE als par.1 > par.2, anders FALSE
1F	31	ZNUMLT	M2:52	TRUE als par.1 < par.2, anders FALSE
20	32	ZNUMNE	M2:54	TRUE als par.1 <> par.2, anders FALSE
21	33	ZNUMLE	M2:56	TRUE als par.1 <= par.2, anders FALSE
22	34	ZNUMGE	M2:58	TRUE als par.1 >= par.2, anders FALSE
23	35	ZMATHS	MS	laadt de FACC met de door HL geadres- seerde waarde, zet DE en roept de M0, M1 of M2 routine met het codenummer in A aan. Slaat het resultaat op. Indien fout dan CY = 1 en A = 2 (ERRNO)
24	36	ZINITRAND	MS[F]	random getal generator wordt op een standaardwaarde gezet
25	37	ZRANDOM	MS[F]	random getalgenerator wordt op de waar- de gezet van CLOCK (= 4 bytes op zeropage 52 - 55h)

*) De logische handelingen NOT, AND en OR zetten eerst de parameters om in 16 bits, twee-complement getallen tussen -32768 en +32767. Als dit niet mogelijk is, wordt er een fout gemeld. Na de bitsgewijze behandeling staat het resultaat in de FACC.

26	38	ZCOMP	MS	zie Handbook
27	39	ZFIX	MS	zie Handbook
28	40	ZFLT	MS	zie Handbook
29	41	ZROUND	MS	zie Handbook
2A	42	ZINP	MS	zie Handbook BUG **)
2B	43	ZLDF	MS	zie Handbook
2C	44	ZOUT	MS	zie Handbook
2D	45	ZSTF	MS	zie Handbook
2E	46	ZRELBOO	MS	controleert Carry en Zero flag en geeft TRUE of FALSE. Aanroep met bit x van B: bit 0 = 1 TRUE indien NC en NZ bit 1 = 1 TRUE indien C en NZ bit 2 = 1 TRUE indien Z



2. Operating System-, in- en outputroutines

Als er een fout wordt gemaakt, wordt de Carry flag gezet en bevat het A-register het foutnummer (ERRNO). Sommige routines keren altijd terug met Carry clear; bij andere (bijv. TTCAPS) duidt de Carry vlag niet op een fout.

hex	dec	naam	bewaarde registers	details
2F	47	ZTTCAPS	BCDEHL	byte in A (indien "a" t/m "z") wordt omgezet in "A" t/m "Z"

***) BUG. Volgens een mededeling van de Engelse gebruikersgroep wijst HL na afloop NIET naar de FACC (alleen O.S. 2.0 en expansion). N.B. Na afloop van een M0-, M1- of M2-routine wijst HL naar de FACC.

30	48	ZOUTPUT	BCDEHL	zie Handbook
31	49	ZINPUT	BCDEHL	zie Handbook
32	50	ZOPENIN	DE	Opent stream voor input Aanroep: A = driver (device), D = port, E = stream, BC = lengte parameterstring, HL = startadres parameterstring Terugkeer: BC, HL = idem parameterstring
33	51	ZOPENOUT	DE	open stream voor output, als OPENIN
34	52	ZCLOSE	DEHL	close stream. Aanroep: E = stream
35	53	ZSTKTST	BCDE	controleer of resterende stackruimte groot genoeg is. Aanroep: BC = aantal bytes Terugkeer: CY gezet als er onvoldoende ruimte op de stack is om BC bytes toe te voegen
36	54	ZBRKTST	BCDEHL	zie Handbook
37	55	ZMKBUFF	BCDE	maakt een eigen-geheugengebied in de IOS-ruimte van BC bytes Aanroep: BC = aantal gewenste bytes, E = stream Terugkeer: als CY = 0 wijst HL naar het begin van het eigen-geheugengebied
38	56	ZGETKEY	geen	haalt toetsenbord-matrixcoördinaat van de COP. Z80 kan zich uitschakelen tijdens GETKEY Terugkeer: A bevat toetsenbord-matrixcoördinaat
39	57	ZIMMKEY	BCDE	haalt toetsenbord-matrixcoördinaat van de COP, indien aanwezig. Terugkeer: CY = 0: A bevat coördinaat CY = 1: geen aanslag gedaan
3A	58	ZKLOOK	DE	zet toetsenbord-matrixcoördinaat om in ASCII of grafisch teken Aanroep: A = coördinaat Terugkeer: A = teken N.B. zero page KBMODE (= PEEK (43)) wordt gebruikt om de actieve attributes te bepalen
3B	59	ZFSTRM	ABCDE	vindt een passend stream nummer Aanroep: E = stream nummer Terugkeer: CY = 0: HL wijst naar stream table entry met offset 0

3C 60 ZBLKIN DE voert een blok bytes van een stream in het geheugen in.
 Aanroep: BC = aantal bytes, HL = eerste geheugenplaats, E = stream nummer
 Terugkeer: CY = 0: BC = 0, HL verhoogd met BC
 CY = 1: BC = aantal bytes niet ingevoerd, HL verhoogd met aantal ingevoerde bytes

3D 61 ZBLKOUT DE voert een blok bytes uit het geheugen naar een stream.
 Als BLKIN

De routines 62 - 67 (3E - 43h) die in de Technical Note nr 10 worden vermeld, heb ik niet kunnen vinden in de ROM van O.S. 1.4, 1.9 en 1.91.

De adressen van de routines kun je vinden in de ZOSTABLES, die je weer kunt vinden, wanneer je de adressen A0CC en verder disassembleert, zoals ook in het boek van John Braga is gedaan.

TOEPASSING

Als toepassing wil ik 2 routines geven uit het boek van Roger Hutter, Cursus Z-80 assembleertaal. De beschreven routine COUT, resp. CIN (pag. 24 en pag. 30), zet de inhoud van het A-register op het scherm, resp. zet de toetsaanslag als ASCII-waarde in register A. De in het boek beschreven routines werken niet op de NewBrain. Met behulp van de RST 20H-routines kunnen nu zeer eenvoudige vervangende routines worden gemaakt:

COUT: LD E,0 ;output byte in A naar stream 0
 RST 20H
 DEFB 30H ;OUTPUT
 RET

CIN: XOR A ;maak A = 0 en CY = 0
 RST 20H
 DEFB 39H ;IMMKEY
 JR C,CIN ;geen toets? dan terug naar CIN
 PUSH BC ;bewaars inhoud BC
 RST 20H
 DEFB 3AH ;KLOOK zet keyboard-matrixcoördinaat om in ASCII
 POP BC ;A bevat de ingetypte ASCII-waarde
 RET



Wim Luijt

drivers

DEVICE DRIVERS EN DE KLOK OP DE NEWBRAIN.

Een van de eerste dingen, die opvalt, als we een NewBrain aanschaffen, is het intensieve gebruik van zogenaamde device drivers. Dit is vooral voor mensen die aan het gebruik van een andere computer gewend zijn, een nogal vreemd begrip, en je vraagt je dan in het begin soms ook af, waarom dit nou zo nodig moet. Kreten als stream, device en poort worden nu eenmaal op veel computers niet gebruikt. Als het echter na enige oefening begint te dagen, ontdek je pas hoe ongelooflijk handig deze drivers zijn. Het blijkt nu plotseling mogelijk om meer videoschermen, grafische schermen, toetsdevices etc. tegelijkertijd en met diverse afmetingen en buffergroottes te openen. Je kan je ene tekstscherf als 'cassetterecorder' gebruiken, en data daarvandaan MERGEN met bestaande basic-programma's enz.

Maar hoe werkt nu zo'n device driver?

De standaard NewBrain is uitgerust met een twaalfstal device drivers, die aan iedereen wel bekend zullen zijn. De NewBrain zelf moet van het gebruik van de drivers een boekhouding bijhouden, anders weet hij zelf ook niet meer, wat nu wel of niet gebruikt wordt. Met dit doel houdt hij een zogenaamde stream table bij, die we kunnen vinden als we adres 86 & 87 PEEKen (PEEK (86) + PEEK (87) * 256). Het einde van deze tabel vinden we op adres 100 & 101. De volgorde in deze tabel is dezelfde, als we gebruiken bij het openen van een stream, eerst het streamnummer, dan het devicenummer, en als laatste het poortnummer. Verder wordt in deze tabel bijgehouden, waar het zogenaamde 'own memory' van een stream staat. In dit eigen geheugen staan dan de diverse parameters, die we hebben opgegeven bij het openen van een stream (baudrate, afmeting van de buffer, aantal regels, etc.). Als we na het openen en sluiten van streams telkens in deze stream table kijken, dan zien we én dat de stream table verschoven is, én dat het eigen-geheugenadres vaak ergens anders staat. Als we na het aanzetten van een NewBrain kijken, waar de stream table zit, dan vinden we:

PEEK (86) + PEEK (87) * 256 = 2306 (begin stream table)

Als we nu typen:

OPEN #0, 0, "150" : OPEN #11, 11, "100"

en we gaan vervolgens de stream table opzoeken, dan vinden we:

PEEK (86) + PEEK (87) * 256 = 10590

Nu gaan we eens in de stream table kijken. Daar staat achtereenvolgens (ingedeeld in groepjes van 6):

```
000 000 000 000 104 002 (str 0, dev 0, port 0, own mem = 616)
011 011 000 000 130 040 (str 11, dev 11, port 0, own mem = 10370)
```

Het vierde getal wordt niet gebruikt, en de laatste twee getallen wijzen naar het eigen geheugen van een stream (1e + 2e * 256). Willen we nu b.v. weten van stream 11, waar PEN 7, 8 en 9 staan, dan kijken we in het eigen geheugen, en vinden daar:

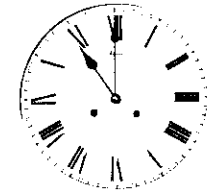
```
eigen geheugen + 42 = PEN(7) (PEEK (10412) + PEEK (10413) * 256)
eigen geheugen + 44 = PEN(8) (PEEK (10414) + PEEK (10415) * 256)
eigen geheugen + 26 = PEN(9) (PEEK (10396) + PEEK (10397) * 256)
```

Zover over de stream table. De NewBrain moet natuurlijk ook nog weten, waar de diverse device drivers zich ergens bevinden (meestal in ROM). Om dit te vinden kunnen we kijken op locatie 88 & 89 (ja, op dezelfde manier PEEKen). Bij het aanzetten van de NewBrain vinden we dan adres 41048 (ergens in de ROM dus). Dit adres kan trouwens verschillen met de diverse ROM-versies van de NewBrain. Gaan we nu kijken, wat op dit adres staat, dan vinden we 12 (PEEK (41048)). Dit getal kennen we al, er zijn namelijk 12 device drivers op de NewBrain. De volgende 24 adressen geven achtereenvolgens de locaties, waar de device drivers zich bevinden. Dus device driver 0 vinden we op: PEEK (41049) + PEEK (41050) * 256 = 41510.

De device driver zelf start ook weer met een tabelletje. Als we gaan kijken hoe deze driver 0 is opgebouwd, vinden we:

```
41510 = 5 is het aantal entries in deze tabel - 1
41511 = 21 hoeveel bytes verder staat het OPENIN-programma
41512 = 20 hoeveel bytes verder staat het OPENOUT-programma
41513 = 4 hoeveel bytes verder staat het INPUT van device
41514 = 6 hoeveel bytes verder staat het OUTPUT naar device
41515 = 8 hoeveel bytes verder staat het CLOSE device
41516 = 22 hoeveel bytes verder staat het move memory
(deze laatste wordt trouwens alleen gebruikt bij de videodevices)
```

De device drivers worden allemaal serieel benadord, dat wil zeggen: de communicatie tussen de processor en een driver gaat byte voor byte. Dit heeft natuurlijk een nadeel, het komt de snelheid niet ten goede (dat is ook vaak te merken bij de NewBrain). Het heeft echter ook een groot voordeel, het is namelijk erg makkelijk de NewBrain te interfaceren met nog niet bestaande devices. Dit kunnen dan hardware-uitbreidingen zijn, maar het is mogelijk een stukje software te besturen als een randapparaat (net zoals b.v. het graphics device 11). Dit gaan we nu in het volgende stukje programma ook doen. Allereerst moeten we de device table die normaal in ROM zit kopiëren naar een stukje RAM, zodat we wijzigingen kunnen aanbrengen. De ROM-locatie van de device driver vinden we op zero page adres 88.



```
1:
2:          ORG 32400
3:
4: DEV:     EQU 10
5: STREG:   EQU 14H      ;poort waar de interrupt status
6:                          ;te lezen is
7: CLOCKI:  EQU 5        ;bit 5 van STREG duidt op klok int.
8:
9: ;#####
10: ;
11: ;      kopieer de device table naar een nieuwe locatie in RAM
12: ;
13: ;#####
14:
15: COPY:    LD  HL,(0088)  ;haal adres device table in HL
16:          LD  A,(HL)     ;A = aantal entries in tabel
17:          ADD A,A        ;A = A * 2
18:          INC  A          ;A = totale lengte devtab
19:          LD  C,A
20:          LD  B,0        ;lengte nu in BC
21:          LD  DE,NEWDEV  ;DE wijst naar nieuwe locatie tabel
22:          LD  (0088),DE  ;bewaars nieuwe device table pointer
23:          LDIR
24:          LD  HL,DEV*2+NEWDEV+1 ;HL wijst naar nieuwe entry
25:          LD  DE,TDEV     ;DE wijst naar nieuwe tijd device
26:          LD  (HL),E
27:          INC  HL
```

```

28:      LD      (HL),D
29:      RET                ;terug naar basic
30:
31: ;#####
32: ;
33: ;   device 10 (vroeger de dummy device is nu het klok device)
34: ;
35: ;#####
36:

```

Nu gaan we de definitie van de device driver maken (aantal bytes offset naar de diverse routines).

```

37: TDEV:  DEFB 4           ;vijf entries
38:      DEFB OPEN-$       ;openin      (zet de klok aan)
39:      DEFB OPEN-$       ;openout     (= openin)
40:      DEFB INP-$        ;input van dev (haal de tijd)
41:      DEFB OUTP-$       ;output naar dev (zet de tijd en opties)
42:      DEFB CLOSE-$      ;close       (zet de klok af)
43:

```

En vervolgens kunnen we de diverse routines voor openin, openout, input, output en close definiëren.

input van device

```

44: INP:  LD      HL,STATUS ;kijk of er wel input gepleegd wordt
45:      BIT      1,(HL)
46:      JR       Z,ERROR
47:
48: STUUR: LD      HL,(POINT) ;pak de te versturen tijdbyte in A
49:      LD      A,(HL)
50:      CP      32           ;als 32 dan klaar
51:      JR       Z,DONE
52:      INC     HL           ;update de pointer
53:      LD      (POINT),HL
54:      RET
55:
56: DONE:  LD      HL,STATUS
57:      RES     1,(HL)      ;reset het 'stuur-bit'
58:      LD      A,13        ;stuur carriage return (om INPUT
59:      RET                ;of LINPUT af te sluiten)
60:

```

openen van het device

```

61: OPEN: LD      HL,(57)   ;verander de interrupt handler pointer
62:      LD      (ADRES),HL ;bij een interrupt wordt nu naar het
63:      LD      HL,BEGIN    ;begin van het klokprogramma
64:      LD      (57),HL     ;gesprongen
65:      RET
66:

```

sluiten van het device

```

67: CLOSE: LD      HL,(ADRES) ;verander de interrupt handler pointer
68:      LD      (57),HL      ;weer in de originele waarde
69:      RET
70:

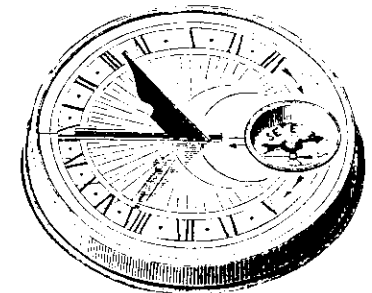
```

output naar het device

```

71: ;#####
72: ;
73: ;   de routine outp heeft een aantal opties:
74: ;   (de waarde van de optie wordt in het A-register meegestuurd)
75: ;
76: ;   T (tijd)      hierna volgt de tijd in uumss
77: ;   S (stuur)     stuur de tijd bij volgende input
78: ;
79: ;   D (display)   display de tijd continu in het scherm
80: ;   U (uit)       zet display af
81: ;
82: ;#####
83:
84: OUTP:  CP      'T'
85:      JR       Z,SET
86:      CP      'S'
87:      JP      Z,INITST
88:      CP      'D'
89:      JP      Z,DISON
90:      CP      'U'
91:      JP      Z,DISOFF
92:      LD      HL,STATUS
93:      BIT      2,(HL)
94:      JR       NZ,KOMT
95: ERROR: SCF      ;als geen geldig commando, dan error 55
96:      LD      A,55
97:      RET

```



98:

en nu nog wat ondersteunende routines

```

99: SET:   LD   HL,STATUS ;set bit 2 status aan. Dit betekent,
100:       SET  2,(HL)    ;dat de tijd geset wordt
101:       LD   HL,CLKBUF ;stuur de tijd als uumms <newline>
102:       LD   (POINT),HL

103:       RET

104:

105: KOMT:  CP   13        ;carriage return sluit af
106:       JR   Z,CALC     ;dan naar de conversie routine
107:       LD   HL,(POINT)  ;pak pointer
108:       SUB  48          ;maak van ascii-waarde nu 0..9 binair
109:       LD   (HL),A      ;bewaar in clkbuf
110:       INC  HL
111:       LD   (POINT),HL  ;bewaar pointer
112:       RET

113:

114: CALC:  LD   HL,CLKBUF ;wijs naar buffer
115:       LD   B,3         ;3 cijfers converteren
116:       LD   DE,UUR      ;pointer voor verwerkte cijfers
117: CALCLP: LD   A,(HL)    ;eerste digit
118:       SLA  A           ;roteer naar hoogste 4 bits
119:       SLA  A
120:       SLA  A
121:       SLA  A
122:       INC  HL          ;wijs de volgende aan
123:       OR   (HL)        ;bij A in ORren
124:       LD   (DE),A      ;bewaar in de tijdtabel
125:       DEC  DE          ;update pointers
126:       INC  HL
127:       DJNZ CALCLP      ;herhaal driemaal
128:       RET

129:

130:

131: INITST: LD   HL,STATUS
132:       SET  1,(HL)
133:       LD   HL,CLKBUF ;zet de pointer op goede adres
134:       LD   (POINT),HL
135:       CALL MAKBUF      ;maak de tijd vertoonbaar in buffer
136:       RET

```

```

137:

138: DISON: LD   HL,STATUS ;zet display aan
139:       SET  0,(HL)
140:       RET

141:

142: DISOFF: LD   HL,STATUS ;display uit
143:       RES  0,(HL)
144:       RET

145:

146: BEGIN: PUSH AF        ;nieuwe interrupt server
147: START: IN   A,(STREG) ;kijk of een klok interrupt
148:       BIT  CLOCKI,A    ;      (50 keer per seconde)
149:       JP   NZ,QUIT     ;anders naar normale interrupthandler
150:       LD   A,(COUNT)  ;verhoog de teller
151:       INC  A
152:       LD   (COUNT),A
153:       CP   50          ;kijk of 50 interrupts geweest zijn
154:       JP   NZ,QUIT
155:       XOR  A           ;zet de teller weer op 0
156:       LD   (COUNT),A
157:       PUSH BC          ;bewaar nu ook de overige registers
158:       PUSH DE
159:       PUSH HL
160:       LD   HL,MAXTAB ;HL wijst naar de maximale tijd
161:       LD   DE,SEC      ;DE naar de lopende tijd
162:       LD   B,3         ;B = aantal
163:       SCF
164: TIKTAK: LD   A,(DE)    ;pak de tijd nu
165:       ADC  A,0
166:       DAA
167:       LD   (DE),A
168:       SUB  (HL)        ;kijk of waarde groter dan in maxtab
169:       JR   C,TEVEEL
170:       LD   (DE),A      ;bewaar de nieuwe tijd
171: TEVEEL: CCF
172:       INC  HL          ;wijs volgende in maxtab
173:       INC  DE          ;wijs volgende in tijdtabel
174:       DJNZ TIKTAK      ;en herhaal
175:

176:       LD   A,(STATUS) ;kijk wat er moet gebeuren
177:       BIT  0,A         ;display op scherm?
178:       JR   Z,EXIT
179:       CALL DISP        ;en toon op scherm

```



```

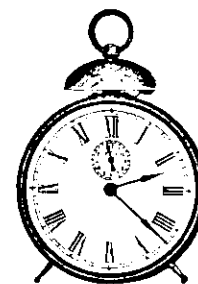
180: EXIT:   POP  HL           ;herstel alle originele registers
181:         POP  DE
182:         POP  BC
183: QUIT:   POP  AF           ;herstel register A
184:         DEFB 0C3H         ;hex-code voor jump
185: ADRES:   DEFW 0000         ;hier staat straks het adres van de
186:                                     ;originele interrupt serve routine
187:
188: MAKBUF:LD  HL,CLKBUF ;zet de tijd in de stuurbuffer
189:         JR   MAKBU
190:
191: DISP:    LD   HL,(005CH) ;zoek startadres van console stream
192:         LD   E,(HL)      ;vind juiste video-adres
193:         LD   D,0
194:         ADD  HL,DE
195:         LD   E,5
196:         ADD  HL,DE
197: MAKBU:LD  DE,UUR          ;stuur de tijd naar het scherm
198:         LD   B,3          ;3 items
199: CONVER:LD  A,(DE)         ;maak er iets vertoonbaars van
200:         SRL  A            ;eerste digit
201:         SRL  A
202:         SRL  A
203:         SRL  A
204:         ADD  A,'0'        ;maak er een cijfer van
205:         LD   (HL),A
206:         INC  HL
207:         LD   A,(DE)       ;tweede digit
208:         OR   0F0H         ;maskeer hoogste 4 bits
209:         XOR  0F0H
210:         ADD  A,'0'        ;+ 48
211:         LD   (HL),A
212:         INC  HL
213:         LD   (HL),':'     ;puntjes tussen uu:mm:ss
214:         INC  HL           ;wijs volgende bufferpositie
215:         DEC  DE           ;wijs volgende tabelitem
216:         DJNZ CONVER       ;totdat alle 3 gedaan
217:         DEC  HL
218:         LD   (HL),32      ;maak van laatste ':' een spatie
219:         RET
220:
221: COUNT:   DEFB 0          ;interruptteller
222: STATUS:  DEFB 0          ;status van het klokdevice

```

```

223: POINT:  DEFW 0000
224:
225: MAXTAB:DEFB 60H          ;maximaal te vertonen seconden
226:         DEFB 60H          ;maximaal te vertonen minuten
227:         DEFB 24H          ;maximum te vertonen uren (maak 12
228:                                     ;voor 12 uurs klok)
229: SEC:     DEFB 0           ;opslag voor seconden
230: MIN:     DEFB 0           ;opslag voor minuten
231: UUR:     DEFB 0           ;opslag voor uren
232:
233: NEWDEV:DEFS 33           ;plaats voor nieuwe device table
234:
235: CLKBUF:DEFS 10           ;hier staat de vertoonbare tijd
236:
237:         END
238:

```



Als we nu dit alles hebben ingetikt (of zie de DATA-regels onderaan) en geassembleerd, dan kunnen we het in de top van het NewBrain-geheugen POKEN (wel even RESERVEEn s.v.p.). Met een CALL TOP wordt dan de nieuwe device driver geïnstalleerd.

Dit nieuwe device laat buiten het maken van een device driver ook nog zien, hoe we gebruik kunnen maken van de clock interrupt van de NewBrain. Het nieuwe device 10 heeft de volgende opties:

- 1 bij OPEN #10, 10 wordt de klok gestart
- 2 bij CLOSE #10 wordt hij weer gestopt
- 3 met PUT #10, "D" komt de tijd continue boven in het scherm staan
- 4 met PUT #10, "U" wordt dit display weer uitgezet
- 5 met PUT #10, "S" kunnen we de tijd ophalen met een INPUT #10 (dus PUT #10, "S" : INPUT #10, t\$: PRINT t\$: REM drukt de tijd af)
- 6 met PRINT #10, "Tuummss" (uu = uren, mm = minuten, ss = seconden) kunnen we de klok gelijk zetten

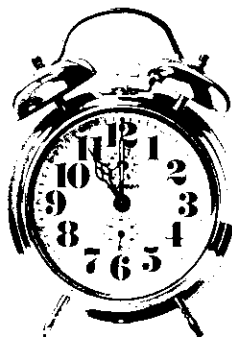
Om het een beetje 'kort' te houden wordt in het programma niet gecontroleerd, of de juiste data wel naar het device gestuurd worden. Het is verder ook nog mogelijk om het uit te breiden met b.v. een alarm, dat een break genereert, die weer af te vangen is met ON BREAK GOTO, zodat we in een programma tijdfanhankelijke subroutines kunnen starten. Hierop gaan we de volgende keer door.

POKE vanaf adres 32400 de volgende DATA. Dan CALL TOP.

```
042,088,000,126,135,060,079,006,000,017,208,127,237,083,088,000,
237,176,033,229,127,017,172,126,115,035,114,201,004,033,032,003,
050,042,033,199,127,203,078,040,067,042,200,127,126,254,032,040,
005,035,034,200,127,201,033,199,127,203,142,062,013,201,042,057,
000,034,144,127,033,079,127,034,057,000,201,042,144,127,034,057,
000,201,254,084,040,026,254,083,202,052,127,254,068,202,067,127,
254,085,202,073,127,033,199,127,203,086,032,016,055,062,055,201,
033,199,127,203,214,033,241,127,034,200,127,201,254,013,040,011,
042,200,127,214,048,119,035,034,200,127,201,033,241,127,006,003,
017,207,127,126,203,039,203,039,203,039,203,039,035,182,018,027,
035,016,240,201,033,199,127,203,206,033,241,127,034,200,127,205,
146,127,201,033,199,127,203,198,201,033,199,127,203,134,201,245,
219,020,203,111,194,142,127,058,198,127,060,050,198,127,254,050,
194,142,127,175,050,198,127,197,213,229,033,202,127,017,205,127,
006,003,055,026,206,000,039,018,150,056,001,018,063,035,019,016,
242,058,199,127,203,071,040,003,205,151,127,225,209,193,241,195,
000,000,033,241,127,024,010,042,092,000,094,022,000,025,030,005,
025,017,207,127,006,003,026,203,063,203,063,203,063,203,063,198,
048,119,035,026,246,240,238,240,198,048,119,035,054,058,035,027,
016,228,043,054,032,201,000,000,000,000,096,096,036,000,000,000,
```

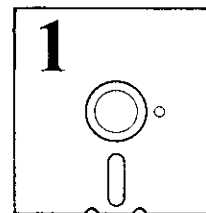
Cor Boer

verkrijgbaar op cassette
zie pagina 14



vreemde formaten

DRIE MANIEREN OM EEN 200K-DISKETTE TE LEZEN
MET TWEE 800K-DRIVES



Met twee 800k-drives is het mogelijk om 200k-diskettes te lezen en te beschrijven. U dient dan als volgt te handelen:

- start uw systeem op onder CP/M
- duw een kopie van uw systeemdiskette in de B-drive
- ga naar de B-drive door het intikken van: B: <NEWLINE>
- type nu het volgende in: RESTCON CS80408 <NEWLINE>

De NewBrain meldt zich dan als volgt:

NewBrain restore configuration program Version 1.0
Copyright (C) 1983 Grundy Business Systems

Als alles goed verloopt, dan volgt de mededeling:

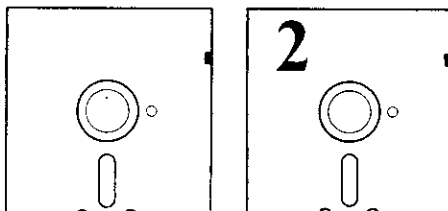
Configuration restoration successful

Remember that to activate the new configuration you must
EXIT and reenter CP/M

Als dit gedaan is, verlaat u CP/M door het intoetsen van EXIT <NEWLINE>. Als u nu weer opstart en in de B-drive uw 200k-diskette duwt, ziet u, dat het 'karwei' geklaard is, en dat u alle programma's moeiteloos kunt kopiëren naar uw eigen formaat. Dit kan het eenvoudigste gebeuren met het PIP-commando. Men dient dan als volgt te handelen:

```
PIP B:=A:*.*[V]
```

Alle programma's worden dan van de B-drive (200k-formaat) naar de A-drive (eigen formaat) gekopieerd.

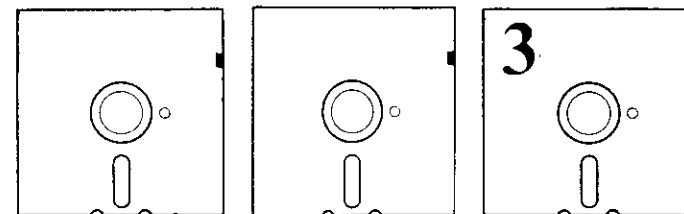


De meest eenvoudige manier om diskettes van een ander formaat te lezen is het bestellen van volume NBGG-1 via de CP/M-bibliotheek van de NewBrain-gebruikersgroep. U kunt deze diskette verkrijgen op elk gewenst formaat (200k, 400k SS, 400k DS, 800k). Op deze diskette staan onder andere een aantal conversieprogramma's, bij voorbeeld 200K.COM, 400KSS.COM, 400KDS.COM en 800K.COM, en ook nog een aantal andere formaten, onder andere MORROW, OSBORNE, SUPERBRAIN, VT180).

Met behulp van deze programma's is het een 'fluitje van een cent' om de B-drive op een ander formaat in te stellen. Typ bij voorbeeld:

```
200K B: <NEWLINE>
```

en uw B-drive gedraagt zich als een 200k-drive. Nu kunt u moeiteloos kopiëren, bij voorbeeld met PIP.



Er is ook nog een derde manier om 200k-diskettes te lezen en te beschrijven. Dit gaat met het programma CONFIGUR.COM.

De antwoorden, die bij dit programma horen, staan in onderstaand overzicht. Aan de linkerkant staan de vragen, die beantwoord moeten worden, en aan de rechterkant staan de antwoorden.

	1e keer	2e keer
NewBrain CP/M 2.2 Configuration program Version 2.0		
Copyright (C) 1983 Grundy Business Systems		
How many disc drives do you wish to configure	2	
What is the physical drive number (Usually n)	0	1
What gap length is used (usually 42)	42	42
What gap length on formatting	40	40
What is the special parameter byte of the drive (usually 0)	0	2
What is the seek rate for the disc in milliseconds	4	20
How big is the sector size (512 or 1024)	512	512
How many sectors are there per track	10	10
What is the skew factor (usually 0)	0	0
How many sides does the drive have (1 or 2)	2	1
Are the tracks numbered . . .	2	
How many tracks per side are there on the disc	80	40
How many tracks are reserved (Usually 2)	2	2
What is the blocksize in Kbytes (1, 2, 4, 8 or 16)	4	2
How many entries are there in the directory (multiple of n)	128	64
Is disc removable (1=yes, 2=no)	1	1

Het configureren van de devices.

Deze informatie is alleen nodig bij een 64k CP/M-systeem. Stream 0 verwijst naar het toetsenbord, stream 1 verwijst naar het scherm, stream 2 verwijst naar de printer, stream 3 verwijst naar de reader / punch en stream 4 is voor het video display. U dient als volgt te antwoorden (null betekent: alleen NEWLINE):

Stream 0 - 19, 0, null
Stream 1 - 18, 0, null
Stream 2 - 16, 0, 10*0
Stream 3 - 17, 0, n*0
Stream 4 - 3, 0, 16

Tenslotte wordt er gevraagd, naar welke drive u de nieuwe configuratie wilt schrijven (drive A = 0, drive B = 1 etc.). De informatie wordt nu op de gewenste schijf geschreven en het programma is beëindigd.

N.B. Als antwoorden zijn een aantal default-waarden gegeven. Deze moet men naar eigen situatie aanpassen; bij voorbeeld seek rate (6 ms), baud-rate, printerbuffer, parallelle printeraansluiting etc.

Tilburg, juni 1985
Jos Hermans



wp(demo?)

Een aantal mensen zullen het programma WPDEMO.COM wel kennen. Het wordt immers standaard meegeleverd op een 400k of een 800k CP/M-systeemdiskette. Na een tijdje met deze 'demo' gespeeld te hebben, moet men toch bekennen, dat dit een leuke en eenvoudige tekstverwerker zou KUNNEN zijn. Zoals het programma standaard op diskette staat, heeft het een groot gebrek. De tekstfiles, aangemaakt met deze tekstverwerker, kan men niet 'saven'. Dit is natuurlijk nogal irritant, als je de tekst wilt bewaren en hier later eventueel iets aan wilt wijzigen. Het blijkt echter mogelijk om deze demo met een minimaal aantal wijzigingen om te bouwen tot een 'heuse' tekstverwerker.

Hieronder volgt de beschrijving:

A>DDT WPDEMO.COM <NEWLINE>

DDT VERS 2.2
NEXT PC
3800 0100
-s113 <NEWLINE>
0113 FF 00 <NEWLINE>
0114 02 . <NEWLINE>
-s13c <NEWLINE>
013C 46 50 <NEWLINE>
013D 02 . <NEWLINE>
-^C (CTRL + C)

A>SAVE 55 WP.COM <NEWLINE>

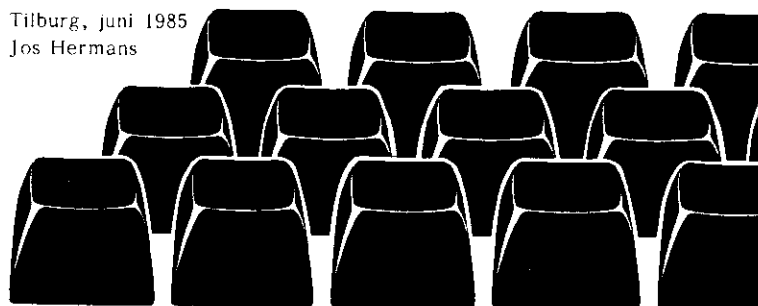
Door het veranderen van het getal FFh (= 255 decimaal), in 0 (nul), op adres 113 is het mogelijk om aangemaakte tekstfiles te 'saven'. De regellengte staat standaard op 70 tekens per regel. Deze werd hierboven, op adres 13C, gewijzigd van 70 (= 46h) naar 80 (= 50h). Deze kan men natuurlijk naar eigen keuze instellen.

Wil men deze DEMO draaien op een ander CP/M-systeem, dan is het handig, als men de cursorbesturingstoetsen kan wijzigen. Dit kan op de volgende adressen:

12c cursor rechts
130 cursor links
134 cursor omhoog
138 cursor omlaag

Veel plezier met deze 'demo'(?).

Tilburg, juni 1985
Jos Hermans



adventure

Hoewel ik niet zo'n spelletjesfreak ben, is er toch een computerspel, waar ik weg van ben, en dat is 'Adventure'. De oorsprong van dit spel ligt in de eind jaren 50, en het is geschreven in de toen gloednieuwe eerste hogere programmeertaal FORTRAN. Deze adventure is verkrijgbaar bij de CP/M bibliotheek, volume 55 (runtime code) en volume 56 (fortran source code, alleen nodig, als je het spel wilt veranderen). Het programma is later uitgebreid, deze nieuwe versie staat op volume 57. Dit is de leukste versie en ook zonder fouten.

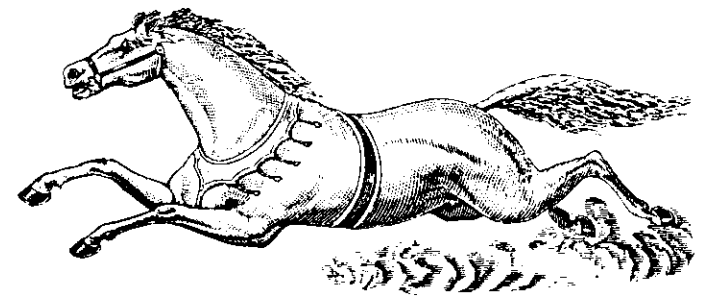
De bedoeling van het spel is, dat je door aanwijzingen te geven aan de zogenaamde Grotgeest een grot (ergens in het bos) binnen gaat, daar allerlei schatten uithaalt en deze in het gebouw brengt, waar je voor staat, als je begint. Na iedere aanwijzing die jij geeft, geeft de Grotgeest een beschrijving van de omgeving waar je bent, of van de gruwelijke dood die je bent gestorven (gelukkig kun je een aantal malen worden gereïncarneerd). De Grotgeest spreekt en verstaat alleen Engels, en dan slechts bepaalde woorden. Alle aanwijzingen die je geeft, bestaan uit een of twee woorden, waarbij de Grotgeest alleen kijkt naar de eerste zes letters van ieder woord. Bepaalde woorden kun je afkorten, b.v. U voor 'up' of NE voor 'northeast'. Als je begint, liggen er in het gebouw een aantal nuttige artikelen. In de grot bevinden zich een slang (die laat zich trouwens verjagen door een klein vogeltje - dit is de enige aanwijzing die ik geef), een beer, dwergjes die met bijlen of messen gooien (naar jou), een troll die tol eist, en nog wat van die leuke dingen. Verder zijn er een aantal leuke doothoven (ik heb er al vier ontdekt) en zijn er wat ravijnen waar je over moet. Als iemand dit spel al tot een goed einde heeft gebracht, bel mij dan even, want ik kom nog niet tot 400 punten (van de 550). Bovendien zit ik vaak zonder licht; hoe kom ik trouwens in die stad die ik zie vanaf een balkon.

Zo word je steeds voor nieuwe problemen geplaatst, die je moet oplossen om verder te komen. Hierbij wil ik wel het advies geven om een soort kaart te maken, waarop je bijhoudt, in welke ruimtes je bent geweest, wat daar aanwezig is of was, hoe je er komt, en waar de uitgangen naar toeleiden. Als je er in een dag niet uitkomt (grapje), kun je de opdracht SAVE geven, dan bewaart de Grotgeest de huidige stand van het spel. Als je dan de volgende dag verder wilt gaan, geef je als eerste opdracht RESTORE en kun je verder gaan waar je gebleven bent. Al met al een spel voor regenachtige dagen.

Henk Blink

fast

Voordat ik een disk controller bij mijn NewBrain kreeg, had ik al een expansion interface. Toen ik de diskcontroller kocht, heb ik die ook gewoon aangesloten en ben ik CP/M gaan gebruiken. Dit ging allemaal erg goed en het ging ook sneller als op b.v. de Kaypro II. Ik was dan ook tevreden, totdat ik eens op een Kaypro IV ging werken. Het viel toen meteen op, dat deze computer veel sneller was met het schrijven naar het scherm. De NewBrain is daar langzaam mee, maar daar doe je niets aan. Dit dacht ik, totdat ik hoorde, dat de NewBrain onder CP/M, maar zonder expansion interface, bijna twee keer zo snel was met het scherm. De oorzaak van de traagheid moest dus liggen bij het Paged Operating System. Ik ben toen met behulp van DASM (een erg goede disassembler uit de CP/M-bibliotheek, schijf nr. 1091) gaan onderzoeken, hoe het een en ander in zijn werk ging. Hier volgen de resultaten, voor zover ik ben gekomen of mij duidelijk is geworden. Als een CP/M-programma een karakter wil uitvoeren naar het scherm, de printer of RS232 poort, gebeurt dit via het BDOS (die weer gebruikt maakt van BIOS-routines) of direct via het BIOS. Bij het aanroepen van het BIOS moet het uit te voeren karakter zich in het C-register bevinden. Er gebeurt nu het volgende:



```

ld    e,str#    ;dit is het streamnummer dat bij het device
                ;hoort, scherm = stream 1
                ;    printer = stream 2
                ;    reader (rs232 in) = 3
ld    a,c       ;het uit tevoeren karakter komt in a
call  cpmzcall  ;deze routine wordt verderop behandeld
defb  zoutput    ;dit is het nummer dat hoort bij de zcall
                ;zoals we die onder basic gebruiken
ret         ;terug naar BDOS of programma

```

De routine CPMZCALL doet het volgende. Eerst worden de registerparen gewisseld: exx, ex af,af'. Daarna wordt het registerpaar AF op de stack gezet, daarna wordt het byte, dat direct na de call staat, gekopieerd naar een plaats verderop in de routine CPMZCALL; deze plaats heet ZCODE. Dan worden de registerparen HL, DE en BC op de stack geplaatst. Vervolgens wordt de huidige stackpointer bewaard en krijgt de stackpointer een nieuwe waarde. Dan wordt page0 omgewisseld: de gewone CP/M-page0 met daarin het programma en de jumps naar warm boot en BDOS wordt vervangen door de syp0, de page0 die daar ook aanwezig is, als we met NewBrain-basic werken. Daarna volgt:

```

exx      ;wissel de registerparen weer om
ex  af,af'
rst  32   ;doe een zcall
zcode: defb 0 ;hier staat nu het byte zoutput

```

Dit is iedereen, die in basic wel eens met machinecode heeft gewerkt, vertrouwd: er wordt gewoon een standaardroutine aangeroepen (zie technical notes 10). In het A-register moet zich het uit te voeren karakter bevinden en in register E het streamnummer.

Na terugkeer van de RST 32 wordt syp0 weer vervangen door de CP/M-page0, de stackpointer krijgt weer zijn oude waarde, en de registerinhouden die op de stack zijn geplaatst, worden weer teruggehaald.

Wat gebeurt er in de routine ZCALL?

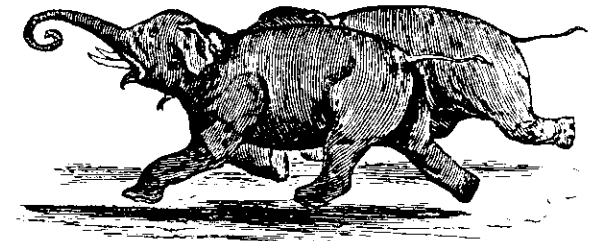
Op 32 staat de opdracht JP 17ACh. Op 17ACh staat een routine die het byte dat na RST 32 staat (ZCODE) ophaalt en met behulp van dat byte uit de tabel ZPTABLE het paginanummer haalt, waar de desbetreffende routine staat. Deze pagina wordt, nadat het huidige paginanummer voor slot 7 is bewaard, in slot 7 gezet. Daarna wordt uit de tabel ZPTABLE het adres van de routine gehaald en wordt er naar toegesprongen. Als we terugkeren van de aangeroepen routine naar ZCALL, wordt de oude pagina 7

weer teruggeplaatst in slot 7.

De routine (OUTPUT), waar we nu zijn aangeland, doet het volgende:

- Huidige paginanummers voor slot 1, 2 en 4 worden bewaard.
- M.b.v. de STREAMTABLE wordt bepaald, welk device is toegekend aan de desbetreffende stream. De bijbehorende buffer (het object) wordt in slot 2 en 3 geplaatst. In het HL-register komt het adres van de buffer.
- Het HL-registerpaar komt op de stack.
- De pagina met de device driver, in dit geval de simple screen editor (SSEIO) wordt in slot 4 geplaatst. In HL komt het adres van de uitvoerroutine van de driver.
- Het adres van de routine wordt in IX gezet, het bufferadres komt weer in HL.
- De uitvoerroutine van de SSEIO wordt aangeroepen.

Na terugkeer vanuit de SSE worden in alle slots weer de pagina's gezet die daar werden aangetroffen door OUTPUT; daarna wordt teruggekeerd naar ZCALL.



Wat doet SSEIO?

Hier volgen de hoofdzaken voor de uitvoer routine:

- Het adres van de buffer wordt in IX gezet.
- De paginanummers van slot 3 t/m 4 worden op de stack gezet.
- De pagina met de SSEIO driver wordt gedupliceerd in slot 3.
- Er wordt naar slot 3 gesprongen.
- M.b.v. informatie uit het object worden de ram-pagina's die het video-geheugen vormen in slot 4 t/m 7 geplaatst, en in HL komt het beginadres van het scherm.

Nu kan er eindelijk echt iets gebeuren op het beeldscherm, en de EDIT-routine wordt dan ook aangeroepen. Deze routine voert de functies uit die zijn beschreven in NewBrain CP/M system description issue 3.0 par 4. Als we terug komen van de edit-routine worden alle slots weer in de oorspronkelijke toestand gebracht.

Sinds de ZCALL heeft het programma negen keer een pagina in een slot verwisseld, en steeds de oude paginanummers bewaard. Tevens moeten we deze slots weer terugbrengen in de oorspronkelijke staat. Dit kost tijd. Waarom is dit gebeurd? Eigenlijk hebben we aan een videopagina voor het scherm genoeg; het scherm beslaat onder CP/M slechts 3200 bytes. Ook ligt het beginadres van het scherm vast. En waarom moeten er voor het object 2 pagina's (= 16 K) worden gewisseld, als het object slechts een tiental bytes bevat? Waarom kunnen we eigenlijk niet direct de goede pagina kiezen voor de device driver en er direct naar toe springen?

De reden van dit alles is, dat we een gedeelte van het operating system gebruiken, dat ook gebruikt wordt door de basic. In basic kunnen de schermlechte, de plaats van het object, het streamnummer en het beginadres van het videoscherm variëren. De oorzaak van deze variatie ligt in de volgorde, waarin de streams worden geopend, en bij de parameters, die hierbij worden gebruikt. Deze flexibiliteit moet worden betaald met extra zoekwerk in tabellen door de routines die zorgen, dat we het juiste deel van het systeem vinden. Onder CP/M ligt alles voor het scherm vast, tenzij men het systeem op een afwijkende wijze gaat configureren (m.b.v. CONFIGUR). Uitgaande van die wetenschap heb ik met wat kunst- en vliegwerk alle relevante gegevens bepaald. Dit zijn de paginanummers van de gebruikte video-ram en het beginadres van de video. Het bleek, dat er maar een pagina videoram gebruikt werd. Deze moet zich in slot 4 bevinden, het beginadres is 8004h.

Toen ik dit eenmaal wist, heb ik het uitvoergedeelte van de SSEIO aangepast: er hoeft maar een paginanummer te worden bewaard, voor slot 4; het beginadres is bekend; er is een nieuw object gedeclareerd en het geheel wordt verplaatst naar slot 0. Tevens heb ik het een en ander aangepast in de EDIT-routine. Deze was namelijk zo geschreven, dat hij zowel in slot 3 als 4 kon werken (in slot 3 voor een expanded system en in "slot" 4 voor een unexpanded system). Nu hoeft hij alleen te werken in slot 0. Dit maakt de calls in deze routine gemakkelijker omdat ze naar absolute adressen gaan. Ook zijn er een aantal editfuncties aan toegevoegd, ClearToEndOfScreen (13h), InverseOn (0Eh) en InverseOff (0Fh).

Deze routine is in sysp0 geplaatst en de adrespointer, die wijst naar het eind van de systeemvariabelen in sysp0, is opgehoogd, zodat hij naar het byte achter de nieuwe routine wijst.

Het BIOS heb ik aangepast. Een CONOUT-call roept de routine CPMZCALL niet meer aan, maar bewaart nu zelf de alternatieve registers en de stackpointer. Ook wordt de stackpointer gewijzigd en wordt sysp0 in slot 0 gebracht. Dan wordt de nieuwe SSEIO-uitvoeroutine aangeroepen, die dan slot 4 verwisselt en vervolgens de nieuwe EDIT-routine aanroept. Na terugkeer vanuit EDIT wordt de CP/M-pagina 4 weer in slot 4 geplaatst en wordt er teruggekeerd naar de BIOS, waar CP/M-pagina 0 weer in slot 0

wordt geplaatst. We zien, dat er nu maar 4 keer een pagina is verwisseld, en dat er geen gegevens uit tabellen hoeven te worden gehaald. Dit gaat aanzienlijk sneller (ongeveer 40 %).

Nadat dit was gerealiseerd, heb ik de conin-routine, de const-routine en de keyboard-interrupt-routine bekeken en ben tot de ontdekking gekomen, dat dit op ongeveer dezelfde wijze werd afgehandeld als de conout-routine, dus veel pagina wisselen en veel zoekwerk in tabellen. Dit heb ik op ongeveer dezelfde wijze aangepast, en het geheel in sysp0 geplaatst. Dit was zelfs nog iets eenvoudiger, daar we niet te maken hadden met vaste geheugenplaatsen, zoals we hadden voor het scherm. Tevens heb ik het programma TOETS (zie On-line 4) in de nieuwe keybord-routine verwerkt. Deze nieuwe routines heb ik ook in sysp0 geplaatst. Het gevolg was, dat de totale tijdswinst bij conin en conout ongeveer 50 % bedraagt. Dit merken we vooral bij programma's die het scherm of het toetsenbord veel aanspreken (WS, SC): het werken wordt stukken plezieriger.

Deze nieuwe routines vormen nu samen het programma FAST. FAST zorgt er ook voor, dat de routines op de juiste plaatsen in het geheugen komen. Je hoeft FAST tijdens een CP/M-sessie slechts een keer te gebruiken en je zult er tijdens het werken onder CP/M niets anders van merken dan dat alles wat sneller gaat. Een klein probleempje voor de mensen die veel van CP/M naar basic gaan: als we FAST hebben gebruikt onder CP/M, werkt de gewone toetsenbordroutine niet meer. Als we EXIT gaan, komen we in het NewBrain paged system main menu. Dit menu gebruikt de gewone toetsenbordroutine, die niet ondersteund wordt, zodat je niets meer kunt invoeren. Oplossing: de computer uit en weer aan zetten.

Het programma had natuurlijk kunnen worden uitgebreid voor de printer en rs232 io, maar omdat deze devices zelf zo traag zijn zet het geen zoden aan de dijk.

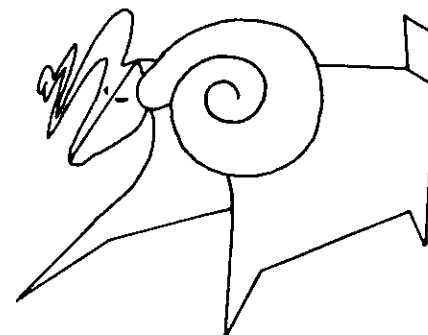
FAST is verkrijgbaar bij de NewBrain cp/m-bibliotheek en draait vanzelfsprekend alleen op een expanded system. Oja, je mag de tekst, die erin staat best veranderen, maar ik zou het op prijs stellen, dat de copyright melding aanwezig blijft.

Henk Blik





ram-disk



DEVICE 20: SERIAL MEMORY DEVICE DRIVER

de gebrekkige documentatie van de newbrain is berucht, en de toelichting bij device 20 is wel bijzonder pover. gelukkig is er in verschillende afleveringen van 'open #stream', het blaadje van de gelijknamige britse gebruikersgroep, een en ander over gepubliceerd. toen ben ik ook ermee aan de slag gegaan: device 20 blijkt een heel echte 'ram disk' te zijn: er worden in het geheugen bestanden gemaakt, die aangesproken worden met een bestandsnaam, als ware het op schijf, compleet met directory. het is net zo gemakkelijk om gegevens of (gedeelten van) programma's naar device 20 weg te schrijven en terug te lezen, als naar diskette

SERIËLE BESTANDEN

de bestanden, die device 20 in het geheugen maakt, komen overeen met de bestanden, die door device 12 op schijf (of door device 1 op cassette) gemaakt worden: het zijn seriële bestanden. dat betekent, dat ze altijd alleen maar van voor naar achter afgewerkt worden. als er een stream naar het bestand geopend wordt, wijst de computer met een pointer naar het begin van het bestand. voor elk byte, dat gelezen of geschreven wordt, schuift de pointer een byte op. de pointer kan weer aan het begin gezei worden door de stream te sluiten en opnieuw te openen. niet het hele bestand hoeft gelezen te worden: zodra er genoeg gelezen of geschreven is, kan de stream gesloten worden.

device 20 is iets slimmer dan device 12, een stream die device 12 gebruikt, wordt geopend óf voor invoer óf voor uitvoer, device 20 nu kan door elkaar schrijven en lezen, en gebruikt bij beide opdrachten dezelfde pointer, laat maar eens de volgende opdrachten uitvoeren:

```
OPEN #20, 20, "voorbeeld" : PRINT #20, "achterdochtig" : CLOSE #20
OPEN #20, 20, "voorbeeld" : GET #20, a, a, a, a, a : PRINT #20, "nta";
CLOSE #20
OPEN #20, 20, "voorbeeld" : LINPUT #20, a$ : CLOSE #20 : PRINT a$
```

PARAMETERSTRING BIJ OPEN

het openen van een stream met device 20 gebeurt op de gebruikelijke manier met streamnummer, devicenummer, poortnummer (indien niet 0) en parameterstring, een volledige parameterstring bestaat uit bestandsgrootte, vlag en bestandsnaam

bestandsgrootte: een asterisk gevolgd door een getal, dat het aantal eenheden van 256 bytes aangeeft, die gereserveerd moeten worden voor het bestand, dit gebeurt alleen, als het bestand gecreëerd wordt: van de volgende streams, die naar het bestand geopend worden, is de mededeling over de buffergrootte overbodig en wordt ze genegeerd, indien niet vermeld, dan wordt de defaultwaarde van 1 k (dus: *4) gehanteerd

vlag, die aangeeft, met welke bedoeling het bestand geopend wordt ('environment option'): een dollarteken gevolgd door een of meer van de volgende letters (in kapitaal of onderkast):

- d er wordt onder de naam, die in de parameterstring opgegeven wordt, een inhoudsopgave gemaakt van alle seriële bestanden in het geheugen
- i de stream, die geopend wordt, kan alleen gebruikt worden voor invoer uit het seriële bestand, via deze stream is het bestand 'read-only', bij een PRINT- of PUT-opdracht wordt er niets geschreven (de pointer wordt dus niet verplaatst), maar er volgt ook geen foutmelding *)
- o via deze stream is alleen uitvoer mogelijk, bij de opdracht GET #st, x krijgt x de waarde 12, LINPUT blijft wachten, totdat op de STOP-toets gedrukt wordt, de pointer wordt niet opgeschoven

*) het nut van de optie \$i heb ik pas ingezien, toen ik de constructie, die patrick vercammen in zijn artikel op pag. 45 van deze on-line beschrijft, ging toepassen op device 20, de NEWLINE, die de computer (bij het uitvoeren van losse opdrachten) na sommige opdrachten op het scherm zet, schreef hij ook naar het seriële opdrachtenbestand, dat ik gemaakt had, dat werd zo helemaal verknoeid, met de optie \$i wordt dat verhinderd

- p als de stream bij opening de optie '\$p' meekrijgt, dan wordt het bestand gewist, als de stream gesloten wordt EN er op dat ogenblik geen andere stream naar het bestand geopend is

er mag meer dan een optie vermeld worden, wel elk met eigen dollarteken: '\$d\$p' en niet '\$dp'. bij combinatie van '\$i' en '\$o' telt alleen de laatste in de rij; '\$d' en '\$p' zijn van een andere orde en gelden altijd, spaties bij bestandsgrootte of vlag worden genegeerd

de bestandsnaam kent weinig beperkingen, spaties aan het begin worden weggelaten, die na het eerste teken, dat geen spatie is, tellen mee in de naam, er is verschil tussen hoofd- en kleine letters, na het zestiende teken wordt de naam afgekapt; kortere namen worden met spaties aangevuld

als bestandsgrootte of vlag vermeld zijn, wordt de bestandsnaam door een dubbele punt ervan gescheiden, een voorbeeld: OPEN #3, 20, 2, "\$d\$p *2 : dit is de inhoud"

de parameterstring lijkt dus het meest op die bij cassettebestanden, met toevoeging van de vlag en beperking van de bestandsnaam tot 16 letters

INHOUDSOPGAVE

als bij het openen van een stream de vlag '\$d' staat, maakt de computer op dat moment een bestand, dat een volledige 'directory' is van alle seriële bestanden, inclusief het net geopende bestand zelf, het inhoudsbestand is dan een min of meer gewoon tekstbestand geworden, dat je net zo vaak kunt lezen, als je wilt, veranderingen in het aantal bestaande bestanden, of in het aantal geopende streams worden daar niet meer in opgenomen, wil je een actueel overzicht hebben, dan moet er opnieuw een inhoudsopgave gemaakt worden door met de '\$d'-vlag te zwaaien **), zo kunnen er oude en nieuwe inhoudsopgaven naast elkaar bestaan, een bestaand inhoudsbestand kan alleen bijgewerkt worden, als er op het moment, dat er een stream met de '\$d'-vlag en dezelfde bestandsnaam geopend wordt, geen andere stream naar dat bestand open is

elk bestand krijgt in de inhoudsopgave één regel, die er als volgt uitziet: zestien plaatsen voor de bestandsnaam; een dubbele punt; een S met daarna op vier plaatsen de grootte van het bestand; een dubbele punt en een R gevolgd door twee plaatsen voor het aantal streams, dat naar het bestand geopend is, de getallen achter S en R zijn hexadecimale getallen in ascii-

**) de beschrijving, die mcphie in open #stream 3 geeft, wijkt essentieel af van het gedrag, dat ik aangetroffen heb en hier beschrijf

weergave (0 tot en met F), na het laatste bestand volgt een regel, waarop slechts ascii 4 (control-d) staat.

een bestand dat geopend is met de opdracht 'OPEN #20, 20, "\$o*28 : voorbeeldprogramma" staat als volgt in de inhoudsopgave:

voorbeeldprogram:SIC00:R01

de inhoudsopgave is op te vragen met het volgende programma, waarbij het bestand, dat het laatst geopend is, het eerst vermeld wordt

```
200 CLOSE #20 : OPEN #20, 20, "$d : inhoudsopgave"
210 INPUT #20, a$ :IF a$ <> CHR$(4) PRINT a$ : GOTO 210
220 CLOSE #20 : END
```

WISSEN

ook als alle streams naar een bestand gesloten zijn, blijft het (uiteraard) bestaan, om een bestand te wissen en de gebruikte geheugenruimte weer vrij te maken moet er een speciale opdracht gegeven worden: de stream, die het laatst nog open zal zijn, moet bij opening de '\$p'-vlag dragen (p van purge = 'verwijderen'), als de teller van het aantal streams naar het bestand van R01 daalt naar R00, wordt gekeken of de stream, die juist gesloten wordt, geopend is met '\$p', zo ja, dan wordt het bestand verwijderd, anders blijft het gewoon bestaan, niet altijd zal aan het begin bekend zijn, welke stream als laatste gesloten zal worden, een bestand kan dus ook verwijderd worden door alle streams ernaar te sluiten; er een te openen met de '\$p'-vlag en die dan weer te sluiten.

omdat de bestanden zich bevinden buiten de geheugenruimte, die aan basic is toegewezen, kunnen de opdrachten NEW, CLEAR, LOAD of RUN ze niet beschadigen, zelfs met EXIT rast je ze niet aan, als je dan naar cp/m gaat en weer terugkomt in basic, kun je ze zo weer aanspreken, maar let op: de 32 k geheugenruimte, waarover de seriële bestanden zich kunnen uitstrekken, heeft een zekere overlap met de videoruimte, dat betekent, dat bij cp/m de schermuitlezing wegvalt, als nogal wat ruimte door device 20 ingenomen is, cp/m werkt wel gewoon, je krijgt alleen niet te zien, wat er gebeurt, er moeten dus seriële bestanden uit het geheugen verwijderd worden, hieronder staat een voorbeeldprogramma, dat alle bestanden van device 20 verwijdert, iets dat ook nodig is, voordat een programma als UNPAGE gedraaid wordt.

```
100 REM verwijdert alle bestanden van device 20
110 FOR i = 1 TO 255 : CLOSE #i : NEXT i
```

```
120 OPEN # 1, 20, "$d$p"
130 INPUT #1, a$ : IF a$ = CHR$(4) CLOSE #1 : END
140 OPEN # 2, 20, 1, "$p:" + LEFT$(a$, 16) : CLOSE #2 : GOTO 130
```

VELE POINTERS

niet alleen zijn er vele bestanden tegelijk te openen en aan te spreken, via verschillende streams, die elk een aparte poort gebruiken, ook zijn er vele streams naar hetzelfde bestand mogelijk, het aantal hiervan is niet beperkt tot negen, zoals g mcphee in open #stream 3 schrijft.

elke stream heeft zijn eigen pointer, die alleen verzet wordt, als er via die stream gelezen of geschreven wordt, waarschijnlijk heeft de gebruiker er meer moeite mee dan de computer, om uit elkaar te houden, wat welke stream in het bestand verandert

FOUTMELDINGEN

ERROR 193. onvoldoende geheugenruimte voor nieuw bestand
ERROR 197. er wordt gepoogd voorbij het einde van een bestand te lezen; of er wordt een niet-bestaande bestandsnaam gevraagd, bij voorbeeld LOAD
ERROR 198. fout bij opgegeven vlag; of er zijn nog streams open naar een bij te werken inhoudsbestand
ERROR 110. syntactische fout in parameterstring: begint met \$ of *, zonder dat vlag of bestandsgrootte volgt

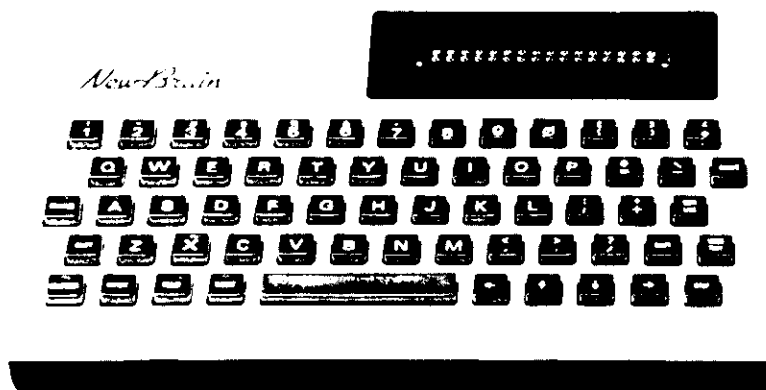
TOT SLOT

het is wel mooi om dit allemaal te weten, maar wat kun je ermee? alles, althans alles, wat je anders tijdelijk op schijf zet, stop je nu in het geheugen; gegevens die door verschillende programma's of programma-onderdelen na elkaar gebruikt worden, het scheelt schijfgeschrijf, zelf gebruik ik device 20 bij een menugestuurd geheel van grote programma-onderdelen, die worden niet vervangen door middel van het tijdrovende DELETE en MERGE, maar na afloop van een onderdeel gaat de besturing over naar een serieel opdrachtenbestand, dat het scherm met het menu in beeld brengt, laat kiezen, het gewenste onderdeel laadt en dat start met RUN

menno stevens



NewBrain



\$25,-

v. Egmond

PROFESSIONELE MICRO-COMPUTER IN KLASSIKAAL NETWERK, NU BEREIKBAAR MET EEN KLEIN BUDGET!

Deze micro-computer met Z 80A processor is al leverbaar voor **f 1190,-** = Een geavanceerde netwerk module (f 2500,-) integreert 16 New Brain's tot een volledige informatica klas met alle onderwijs faciliteiten.

Enkele technische gegevens die het weten waard zijn:

De NEW BRAIN micro-computer:

☐ is uitgerust met een zgn. ANSI BASIC compiler: een standaard programmeertaal;

☐ heeft een volledig geïntegreerd schrijfmachine-toetsenbord van normale afmetingen;

☐ hij is standaard uitgerust met een 32k RAM geheugen dat echter met inplugbare modules is op te voeren tot twee megabytes;

☐ hij beschikt over aansluitmogelijkheden voor twee cassetterecorders, TV, monitor, printer en modem;

☐ biedt de mogelijkheid om andere software-pakketten te laden vanaf cassette of onder CP/M vanaf schijf;

☐ beschikt voorts over 80 tekens maal 30 regels video;

☐ eveneens biedt hij de mogelijkheid voor grafische toepassingen (zgn. high resolution graphics);

☐ alle interfaces en uitgangen zijn standaard;

☐ bescheiden afmetingen: 28 x 15 x 5 cm;

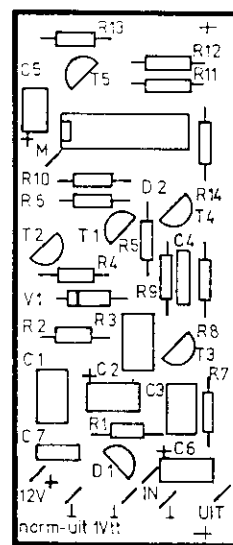
☐ is ook te leveren met nikkel-cadmium batterijen voor automatische beveiliging tegen kortdurende stroomuitval!

Nadere inlichtingen worden u desgewenst graag verstrekt door:

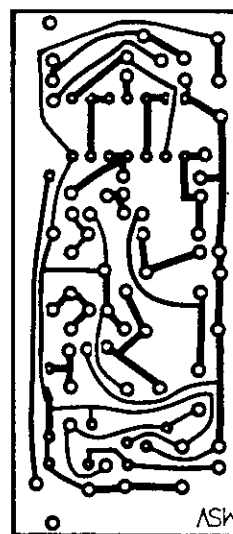
Prijzen excl. BTW



Tradecom International B.V.
Hondsdijsk 3,
Postbus 60,
2396 ZH Koudekerk a.d. Rijn
Tel. 0174-43 00
Telex 39466



ONDERDELEN-
OPSTELLING



PRINTAANZICHT

video- signaal



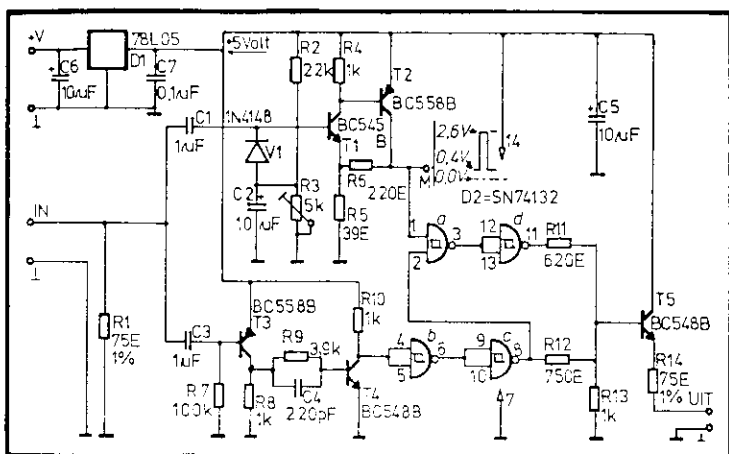
VIDEO-UITGANG NORMALISEREN

Het videosignaal dat de NewBrain afgeeft, is niet volgens de norm van 1 Volt top/top. Dit zal in het algemeen geen probleem opleveren. Met de contrastregelaar van de monitor op maximum, gaat het meestal wel. Het wordt echter een probleem als de uitgang van de NewBrain verbonden moet worden met de ingang van een apparaat, dat voor normvideosignaal is bedoeld.

Waar gaat het hier om? Onderstaande tabel geeft uitleg.

Waarden over 75 Ohm	NewBrain	Norm
helderheidsdeel	0,30 V _{tt}	0,70 V _{tt}
synchronisatie-impuls	0,50	0,30
totaal	0,80 V _{tt} *	1,00 V _{tt} *

Met behulp van het schema, dat bovenaan de volgende pagina staat afgedrukt, is het video-sig-naal weer op "norm" te brengen. De opzet



SCHEMA

van de schakeling berust op de eigenschap van de Schmitt-Trigger (D 2) om een logische "1" af te geven voor een signaal dat boven de drempel van 1,7 Volt komt, en een "0" voor een signaal dat onder 0,4 Volt komt.

In het schema vormen de transistoren T 1 en T 2 de ingangsversterker, die het helderheidssignaal zesmaal versterkt. Met R 3 moet het werkpunt ingesteld worden. T 3 en T 4 vormen een synchronisatie-scheider. D.w.z.: de schakeling onderdrukt het helderheidssignaal en laat de synchronisatie-impuls door. Beschikbaar aan de uitgang van poort D 2-c.

Met de weerstanden R 11, R 12 en R 13 wordt een afdeling gemaakt in de juiste verhouding. T 5 stuurt de uitgang aan.

Met R 3 in de middenstand zal de schakeling direct moeten werken. Met een scoop op punt "M" gemeten kan de aangegeven waarde worden ingesteld. Is geen scoop beschikbaar, dan is het zoeken van een stand van R 3 te vinden door uitmiding tussen veel signaal (CTRL/W-c) en weinig (CTRL/W-b).

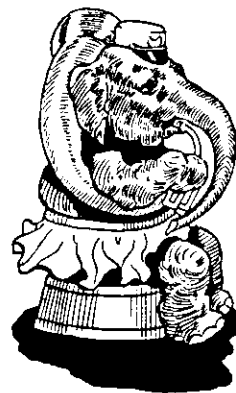
De schakeling wordt gevoed met D 1. Ingangsspanning mag liggen tussen 8 en 16 Volt. Stroomafname 30 tot 40 mA.

Bussum, M. S. Vreedenburg



geluid

DE COMPONENTEN VAN HET SOUND DEVICE



In de NewBrain Expansion interface bevinden zich op het printboard nog een aantal lege plaatsen, waar nog een aantal componenten kunnen worden geplaatst. Dit zijn de componenten die nodig zijn voor het sound device, het parallel input/output device en de analoge i/o devices. De software voor deze devices is wel aanwezig. Als je dus de componenten monteert, kun je ze gebruiken. Waarom Grundy zelf deze componenten niet heeft ingebouwd, is wat onduidelijk; redenen kunnen zijn, dat de voeding te klein was, dat de EIM te heet zou worden, of dat de componenten niet te krijgen waren.

Hier volgt in ieder geval de lijst van benodigde componenten.

ref.nr.	type	waarde	opmerkingen
53, 54	weerstandbankje	270R	{ weerstand bankje met 4 individuele weerstanden
57	weerstandbankje	4k7	
59, 65, 125, 126	weerstand	10k	eventueel 4k7
60, 142	(instel)pot	500k	
61	(instel)pot	250k	
64, 122, 131, 135	weerstand	4k7	
66, 119, 127, 128	weerstand	5k0	
102	weerstand	37k	
111	weerstand	25k	
113	weerstand	100k	
115, 116	weerstand	2k5	
117, 124	DIODE	1N914	
118	condensator	47pF	
120	condensator	0.01uF	
129, 143, 144	condensator	10nF	
130	condensator	1nF	
133, 134	ZENERDIODE	5v1 400mW	
137, 139	weerstand	220R	

IC reference numbers

ref.nr.	functie	typenr.	opmerkingen
424	Oct. Bi. Dir. Buff.	74LS245	User Bus
427 *	8 ch. A-D	ADC 0809	Analoge I/P's
452, 463	Quad OP Amp.	LM 324 C	Buffer Amp's
454	Dual F/F.	4013	Sound Dividers
457	Analogue Mltplx.	4053	Sound
460	Oct. latch	74C374	An. Out. (equiv. 40374)
461	D-A	DAC 0808	(equiv. MC1408-8N)

Component 149 is een potmeter voor de volumeregeling. Het beste is om een losse potmeter te gebruiken, die m.b.v. afgeschermd kabeltjes wordt aangesloten.

Weerstand 111 is op de meeste EIM's 100k, dit moet een weerstand van 25k worden. Een van de printbanen naar deze weerstanden is doorgesneden en er is van daar een draad getrokken naar pen 24 van de Z80 CTC. Deze draad moet worden weggehaald, en de doorgesneden printbaan moet worden gerepareerd.

Voor eventuele verbeteringen aan de voeding van de EIM verwijs ik naar het artikel van Rob Maris over geluid in On-line 4.

Het kan zijn, dat bepaalde componenten in uw EIM al aanwezig zijn. Controleer dit dus even, voordat u tot de aanschaf overgaat. Bepaalde componenten zijn soms moeilijk of niet te krijgen. Tradecom heeft ooit eens toegezegd om setjes met deze componenten samen te stellen voor leden van de NB-gg, maar dat was al meer als een jaar geleden en ze zijn er nog steeds niet. Had u anders verwacht van Tradecom?

Voordat we deze componenten kunnen aanbrengen, zullen we eerst de gaatjes in het printboard invrij moeten maken met behulp van een tinzuigpompje. Stook het board niet al te lang heet, de printbaantjes willen erg gemakkelijk loslaten. Het werken aan het printboard is eigenlijk alleen maar te doen voor mensen met ervaring, waag u er dus niet te gauw aan. Als u er aan begint, vervalt uw garantie en ik garandeer ook niets. Het werkt trouwens wel, getuige het apparaat van Philip Crookes, dat te zien en te horen was op de vorige NB-gg-dag in Utrecht.

Henk Blik



netvoeding

PROBLEMEN BIJ HET GEBRUIK VAN ANDERE VOEDINGEN

Zoals de heer Vreedenburg in On-line 4 (pag. 69) al vermeldde, blijkt het normaal niet mogelijk een geschakelde voeding of een andere voeding te gebruiken, waarbij de uitgangsspanningen +5, +12 en -12 volt bedragen. Het lezen van band wordt namelijk onmogelijk.

De oplossing van de heer Vreedenburg heb ik aangebracht. Maar ik wilde het toch beter zien te krijgen, daar het nog steeds niet mogelijk was van band te lezen, wanneer de +12 V gezakt zou zijn tot bijvoorbeeld 11,5 of 11 V. Aangezien ik het systeem uit accu wil kunnen voeden, ofwel daar waar geen netspanning aanwezig is, of als noodstroom bij storing in de netspanning, stelde ik als eis, dat het systeem tot 11 V betrouwbaar zou werken.

Nu, dat doet het ook, behalve de tape-input, hetgeen in den beginne vrij moeilijk te vinden was. De VOA voedt alleen de twee opamps, namelijk die in de voedingsstabilisatie en die voor de tape-interface. Als men nu de +12 V langzaam lager draait en de spanning op de opamp van de tape-interface bekijkt, ziet men, dat deze plotseling 1,5 tot 2 V zakt, terwijl de spanning zeker 5 V moet zijn in verband met de poort, die erachter hangt.

Remedie: die opamp uit 5 V voeden. Zo simpel is dat nou!

Handelwijze: knip voorzichtig met een dunne kniptang pen 8 van IC 414 los van de print en buig die pen iets naar buiten. Soldeer daaraan 1,5 tot 2 cm draad en verbind dat met pen 16 van IC 413 (+5 V), en het systeem werkt naar volle tevredenheid. Degenen, die de modificatie van de heer Vreedenburg hebben ingebouwd, kunnen die laten zitten, maar die is nu niet meer noodzakelijk. Het hele systeem (inclusief Teac-diskdrives) werkt zonder meer betrouwbaar op 11 V, zodat accuvoeding mogelijk is.

Nog wel twee omvormertjes voor 5 V (in verband met het rendement) en -12 V. De +12 V voor de drives kan met een speciale stabilisatieschakeling (in verband met verlies over de transistor) uit de accu gevoed worden. De -12 V mag zelfs wel -2,5 V bedragen (in mijn geval), maar een maximum van -8 V lijkt mij wel verstandiger, mede voor de betrouwbaarheid van de RS232-poorten.

Ik ben reeds bezig een dergelijk accuvoedingssysteem te ontwikkelen. In On-line 6 hopelijk meer.

Succes!

Evert Drijver



inhoud on-line 5

- 2 adressenlijst
- 4 regionale bijeenkomsten in het zuiden /jos hermans/
- 5 een retourtje utrecht-parijs /rob maris/
- 7 softwarebibliotheek vernieuwd /jan tolkamp/
- 11 catalogus softwarebibliotheek
- 15 catalogus onderwijsbibliotheek
- 25 cp/m-bibliotheek /jos hermans/
- 27 prijsvraag
- 29 vraagjes en weetjes - vragen en weeten
het werken met disk vanuit basic /h balemans/
gespikkelde spinnen?
newbrain text editor + star gemini 10x /a van de velde/
hoofdletterslot voor texy2 /pieter w m van dijk/
inverter in clockcircuit /rob maris/
- 33 hoe kun je programma's zo maken, dat ze op alle newbrain-con-
figuraties werken? /menno stevens/
- 38 stroming /a van de velde/
- 39 microcalc /john van der ploeg/
- 45 open #0, "filenaam.ext" /patrick vercammen/
- 49 leer doortje stoot dansen /arno roelofs/
- 54 plot circle /wim luijt/
- 55 hoe snel is langzaam en hoe groot is 0? /wim luijt/
- 61 screendump /h balemans/
- 65 operating system-routines /wim luijt/
- 71 device drivers en de klok op de newbrain /cor boer/
- 81 een 200k-diskette lezen met twee 800k-drives /jos hermans/
- 84 wp(demo?) /jos hermans/
- 86 adventure /henk blik/
- 87 fast /henk blik/
- 93 device 20: serial memory device driver /menno stevens/
- 99 video-uitgang normaliseren /m s vreedenburg/
- 101 de componenten van het sound device /henk blik/
- 103 problemen bij het gebruik van andere voedingen /evert drijver/

NewBrain-
gebruikersgroep
postbus 4494
1009 AL amsterdam

softwarebibliotheek

BESTELFORMULIER

naam _____

adres _____

postcode en woonplaats _____

telefoon _____

hec-lidmaatschapsnummer _____

bestelt bij deze de aan ommezijde aangegeven software

het totaalbedrag ad f _____

☐ is overgemaakt d d _____

op postrekening 2700380 tnv j de vries te utrecht

afkomstig van bank-/postrek nr _____

tnv _____ te _____

☐ wordt voldaan met bijgesloten girobetaalkaart,
betaal- of eurocheque

nummer giropas of betaalpas _____

handtekening

NewBrain-
gebruikersgroep
postbus 4494
1009 AL amsterdam

cp/m-bibliotheek

BESTELFORMULIER

naam _____

adres _____

postcode en woonplaats _____

telefoon _____

hec-lidmaatschapsnummer _____

bestelt bij deze de aan ommezijde aangegeven volumes

het totaalbedrag ad f _____

☐ is overgemaakt d d _____

op postrekening 2700380 tnv j de vries te utrecht

afkomstig van bank-/postrek nr _____

tnv _____ te _____

☐ wordt voldaan met bijgesloten girobetaalkaart,
betaal- of eurocheque

nummer giropas of betaalpas _____

handtekening

naam _____

straat _____

postcode en woonplaats _____

telefoon _____

☐ is al lid van de HCC, lidmaatschapsnummer _____ en geeft zich op als lid van de NewBrain-gebruikersgroep

☐ geeft zich op als lid van de HCC en de NewBrain-gebruikersgroep

☐ geeft zich op voor regelmatige toezending van NEWBRAIN ON-LINE (prijsstelling is voorlopig circa f 5,50 per 50 pagina's)

datum _____ handtekening _____

AANMELDINGSKAART

naam _____

straat _____

postcode en woonplaats _____

telefoon _____

☐ is al lid van de HCC, lidmaatschapsnummer _____ en geeft zich op als lid van de NewBrain-gebruikersgroep

☐ geeft zich op als lid van de HCC en de NewBrain-gebruikersgroep

☐ geeft zich op voor regelmatige toezending van NEWBRAIN ON-LINE (prijsstelling is voorlopig circa f 5,50 per 50 pagina's)

datum _____ handtekening _____

AANMELDINGSKAART

NewBrain-gebruikersgroep
postbus 4494
1009 AL Amsterdam

CPA/m-bibliotheek

formaat aankruisen:

☐ diskette 200k
☐ diskette 400k ss
☐ diskette 400k ds
☐ diskette 800k

NBGG-1	_____	_____	_____
NBGG-2	_____	_____	_____
NBGG-3	_____	_____	_____
NBGG-4	_____	_____	_____
53	_____	_____	_____
56	_____	_____	_____
57	_____	_____	_____
1091	_____	_____	_____

prijs per volume: 400k ss, 400k ds of 800 k: f 15,00 (1 diskette) 200k f 25,00 (2 diskettes)

per 10 volumes in één bestelling een volume gratis

prijs per volume _____

besteld aantal _____

NewBrain-gebruikersgroep
postbus 4494
1009 AL Amsterdam

software-bibliotheek

formaat aankruisen:

☐ cassette
☐ diskette 200k
☐ diskette 400k ss
☐ diskette 400k ds
☐ diskette 800k

EDUC-1	☐	SPEL-1	☐	MATH-1	☐
EDUC-2	☐	SPEL-2	☐	VIDEOL-1	☐
HILP-1	☐	SPEL-3	☐	DIVERS-1	☐
GRAF-1	☐	SPEL-4	☐	ONLINE-5	☐
LIN-1	☐	SPEL-5	☐		☐

prijs per cassette / diskette f 15,00

besteld aantal _____

totaalbedrag _____

