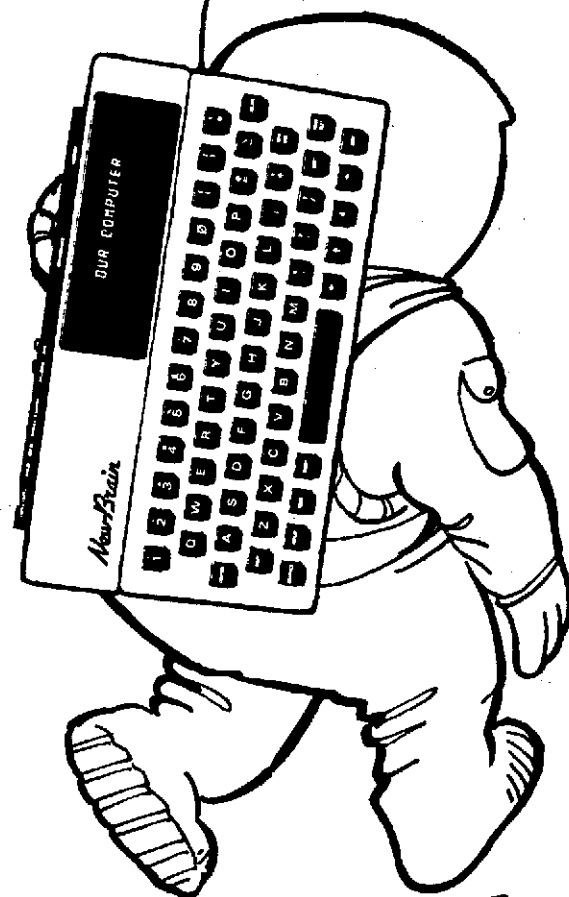




7 april 1990
 6 oktober 1990
newbraindag
 sbbo
 leiden

New Brain on-line



14

oktober 1989

*uitgave
 van de
 NewBrain-
 gebruikersgroep*

newbrain on-line is een uitgave van de nederlandse newbrain-gebruikersgroep. de bladen zijn licht gelijmd, zodat ze eenvoudig los te maken en in een ringband op te bergen zijn. de artikelen kunnen dan bijvoorbeeld op onderwerp geordend worden. 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 dan ook bij de inzender.

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

abbonementen kunnen alleen beëindigd worden door schriftelijke opzegging bij de gebruikersgroep.

eerder verschenen nummers kunt u bestellen door overmaking van f 10,- per nummer op girorekening 2505800 ten name van hcc newbrain-gebruikersgroep te amsterdam

kopij

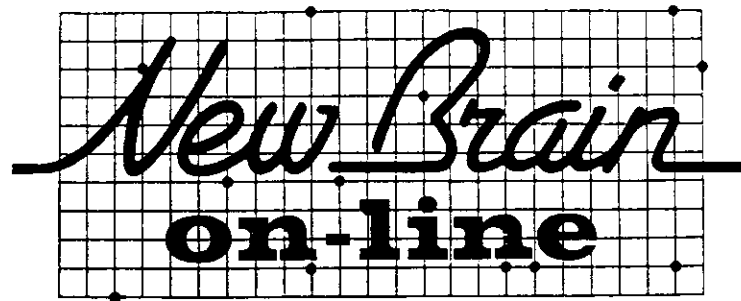
newbrain on-line 15 zal (bij voldoende kopij!) verschijnen op de newbrain-dag op 7 april 1990. sluitingsdatum voor de kopij is 1 maart 1990. maar stuur uw kopij zo spoedig mogelijk op: het werkt veel prettiger, als niet alles op het laatst gehaast gedaan moet worden. ons adres is:

newbrain-gebruikersgroep, postbus 4494, 1009 AL amsterdam

het is voor de redactie verreweg het makkelijkst de kopij aangeleverd te krijgen op cassette of diskette. dat bespaart de fouten bij het overtypen. vermeld bij een diskette wel het formaat (alle newbraininformaten en 360k ms-dos kunnen we aan). de tekst mag op alle gangbare manieren zijn opgeslagen: wordperfect-, text-, wordstar-, wp- en alle mogelijke ascii-bestanden zijn bruikbaar. de cassette of diskette krijgt u natuurlijk terug. 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

menne feit



TEN GELEIDE

dit najaarsnummer van newbrain on-line is zeer geschikt voor de lange winteravonden: geen verzameling kleine weetjes, maar een aantal wat langere artikelen, waar u wat aan hebt. een greep uit de inhoud:

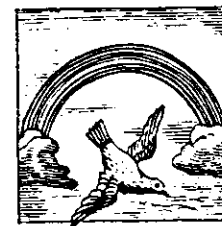
er is de aankondiging van twee delen nieuwe software, niet oninteressant. in een samenvattend artikel over het paged operating system heeft wim luijt de lezing uitgewerkt, die hij op een newbraindag gehouden heeft. het is een lang artikel, maar zeer de moeite waard.

co koning heeft een aantal stukken hardware onder handen genomen. dat heeft geresulteerd in twee gedegen artikelen, over de rombox en over versie 5 van de diskcontroller.

m s vreedenburg draagt wat informatie aan over enkele obscure onderdelen van de newbrain.

tot slot de obligate oproep voor kopij: er zijn nog steeds mensen actief met hun newbrain, en nog steeds geïnteresseerd in uw bevindingen en ontdekkingen: na alle jaren is nog lang niet alles algemeen bekend, dus met uw artikeltje doet u anderen plezier: over een nuttige besteding van lange winteravonden gesproken ...

de redactie



NewBrain- gebruikersgroep postbus 4494 1009 AL amsterdam

voorzitter: m s vreedenburg, (02159) 11068

secretaris-penningmeester: menno stevens, (020) 924137

postrekening 2505800 tnv hcc newbrain-gebruikersgroep, amsterdam

de newbrain-gebruikersgroep is een onderdeel van de



HOBBY COMPUTER CLUB

inschrijvingsnummer kvk leiden: V445230

LANDELIJKE NEWBRAIN DAGEN

coördinatie: m s vreedenburg, (02159) 11068

komende newbraindagen: zaterdag 7 april 1990 en zaterdag 6 oktober 1990

in de sbbo, lammenschanspark 1, 2321 JK leiden

NEWBRAIN ON-LINE

redactie: menno stevens, (020) 924137

sluitingsdatum voor de kopij voor on-line 15 is 1 maart 1990

zie de binnenkant van het omslag voor het inzenden van kopij

SOFTWAREBIBLIOTHEEK

bibliothecharis: bas boetekees, hans van hoek, w a van hoek, (020) 948430

uitlevering op cassette: guus von morgen, (02158) 3381

uitlevering op diskette: m s vreedenburg, (02159) 11068

bas boetekees, (03465) 76211

henk dekker, (020) 150298

pieter van dijk, (01751) 12367

maarten floor, (02963) 4374

jos hermans

hans van hoek (020) 940851

w a van hoek, (020) 948430

co koning, (02521) 12742

wim luijt, (01100) 28197

rob maris, (055) 424485

guus von morgen, (02158) 3381

jan wubben, (010) 4557698



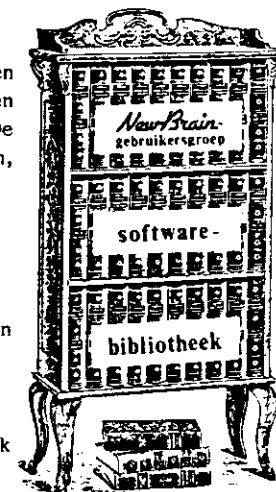
bibliotheek

De nieuwe schijf N233 staat voornamelijk in het teken van tekstverwerkingsprogramma's. Allereerst is er het basicprogramma TEKSTBEWERKING van de hand van Ger van Laere. Dit programma was al een poosje in reserve en nu is dan het ogenblik gekomen, dat het gepubliceerd wordt.

Het is een programma dat in eerste instantie geschikt is voor de 32k NewBrain; het kan echter ook op de 96k NewBrain gebruikt worden met verlies van de mogelijkheid om de rechter kantlijn in te stellen. Dit is op zich geen groot nadeel. Wil je hier wel gebruik van maken, RUN dan eerst UNPAGE.BAS.

Een goed overzicht van de gebruiksmogelijkheden krijgt men door via het programma de bestanden TEKSTUIT.001 tot en met .003 te bekijken. De diverse onderdelen die daarin ter sprake komen, laten we hieronder volgen.

1. Tekst opslaan in een aparte tekstlade
2. Lade ledigen (tekst eruit wegwerpen!)
3. Opgeslagen tekst ergens (tussen) plaatsen
4. Informatie over vrije regels en vrij geheugen
5. Hoofdletters / kleine letters
6. Signalering paginalengte
7. Tekst naar rechts / links via @
8. Rechter kantlijn instellen (alleen op 32k NewBrain)
9. Tabulatie-instelling c.q. linker-kantlijninstelling
10. Printen van tekst via de printer
11. Wegschrijven en opvragen van tekst van / naar cassette of disk
12. Uitvullen van teksten naar rechts



Alle toetsenbord-functies blijven werkzaam (gemakkelijk intypen).

Na het laden van het programma en het commando RUN verschijnt, met inachtnaam van het bovenvermelde, het menu. Dit wijst verder de weg door het programma.

Vervolgens presenteren we het tekstverwerkingsprogramma A4.BAS van de hand van M. S. Vreedenburg. De mogelijkheden die door dit programma

worden geboden, zijn zeer vele.

Een paar opmerkingen vooraf tot goed begrip van de behandelingswijze. Oorspronkelijk is het programma bedoeld voor gebruik met de matrixprinter IDS 480. Hierop zijn ook de toelichtingsbestanden A4HLP0.TXT tot en met A4HLP.TXT gebaseerd.

We hebben een aangepaste versie voor de matrixprinter Epson FX80 gemaakt. Laad eerst A4.BAS en daarna: MERGE "A4EPSON.OVL" en u hebt de versie voor de Epson. Deze kunt u SAVEn onder een naam die u welgevallig is; dan hoeft u een volgende maal de ingewikkelde en naar verhouding langdurige methode van het MERGEn niet meer toe te passen. De tien informatiepagina's A4EHLP0.TXT tot en met A4EHLP9.TXT vormen een geheel en zijn wat printer codes betreft aangepast.

De opzet is zeer systematisch en modulair. Vooral de gegeven schema's op de pagina's 6, 7 en 8 geven een inzicht in de werking van het programma en stellen ons in staat om zelf nodig gevonden veranderingen aan te brengen, respectievelijk printer codes te wijzigen naar behoefte.

Tot slot nog een opmerking: in de oorspronkelijke versie is aangenomen dat men over een NewBrain AD beschikt. In het line-display verschijnen dan verschillende mededelingen over bijvoorbeeld cursorpositie, lengte van de pagina enzovoort. Aangezien diverse leden (waaronder uw bibliothecaris) over een NewBrain A beschikken, wordt het programma door de Epson-overlay zo gewijzigd, dat de betreffende mededelingen op het scherm verschijnen. Na intypen van een willekeurige toets gaat het programma weer verder. Voor wijzigingen in het programma om de functie van het line-display te gebruiken, moeten de volgende veranderingen in A4EPSON.OVL worden aangebracht:

- ¶ in regel 120: vervang OPEN #30, 0, 30, "5" door OPEN #30, 3, "18"
- ¶ laat de volgende regels vervallen:
1060, 1090, 1120, 1300, 1320, 1330, 1350, 1360 en 1370

Dit dient te gebeuren, voordat u het programma als overlay gaat gebruiken.



Als volgende is er nog een verbeterde versie van de eerder op bestelnummer N504 verschenen grafische interpreter POGO.BAS opgenomen. De handleiding POGO.HAN is voldoende duidelijk (zie pagina 17) en de bijgeleverde programma's kunnen u de mogelijkheden van dit programma laten zien.

Als nieuwe schijf in de N300-serie komt N314 uit. Deze bevat een aantal programma's die nog uit het archief van onze vorige bibliothecaris Wim Luijt zijn opgedoken. Er zijn heel interessante programma's bij. Een voordeel is, dat van de meeste programma's de source code is opgegeven, waardoor men, als daar behoefte aan is, er zelf z'n licht kan opsteken en eventueel wijzigingen of toevoegingen bij kan maken. Niet alle programma's zijn zelfverklarend. Er zijn er ook bij waarvan de functies in reeds beschikbare programma's terug te vinden zijn. Wees voorzichtig met programma's die veranderingen in gecompileerde files mogelijk maken! Speciale aandacht verdienen de programma's RELOBJ.BAS en RELOC.BAS samen met LADER.BAS en TRUN.BAS. Dit zijn programma's om uw eigen programma's inbraakvrij te maken! Hiervoor zij verwezen naar het artikel van Wim Luijt op pagina 8 in deze On-line.

Op cassette zijn van bovenstaande programma's verkrijgbaar: TEKSTBEWERKING, POGO, DISEXP en RELOBJ en RELOC, elk met de bijbehorende bestanden.

W. A. van Hoek



het bestellen van software

neem het bestelformulier achterin deze on-line, en vul daarop in, welke software u wenst te ontvangen (niet alleen de opgesomde delen zijn leverbaar, maar ook alle cp/m-volumes van de 'programmatheek' van de dos-gebruikersgroep!). kruis het gewenste formaat aan: cassette of diskette van 200k, 400k ss, 400k ds of 800k voor de newbrain, of vul het proton-formaat in, stuur de bon samen met een girobetaalkaart of betaal- of eurocheque (vergeet niet het nummer van giro- of betaalpas in te vullen) op naar postbus 4494, 1009 AL amsterdam. het verschuldigde bedrag kan ook worden voldaan door overschrijving op girorekening 2505800 ten name van hcc newbrain-gebruikersgroep te amsterdam. ook in dat geval het bestelformulier opsturen. zonder bestelformulier gaat het vast mis

catalogus

NIEUW

bestelnummer: N233 (diskette)

TEKSTBEW.BAS (6k) tekstbewerker van ger van laere, voor newbrain 32k en 96k. met gebruiksaanwijzing TEKSTUIT.001, .002 en .003 (resp 6k, 4k en 4k), te lezen met het programma zelf

A4.BAS (8k) tekstverwerkingsprogramma van m s vreedenburg voor newbrain ad 96k. met uitgebreide toelichting, te lezen met het programma zelf:

A4HLP0.TXT (2k) overzicht, toelichting en handleiding

A4HLP1.TXT (4k) 1. instructieset van de newbrain

A4HLP2.TXT (4k) 2. toelichting op de SAVE- en LOAD-mogelijkheden

A4HLP3.TXT (3k) 3. instructieset via GRAPHICS-handelingen bij open pagina

A4HLP4.TXT (4k) 4. kopiëren van tekst tussen pagina's onderling

A4HLP5.TXT (4k) 5. printerinstellingen

A4HLP6.TXT (6k) 6. schema van het tekstprogramma

A4HLP7.TXT (4k) 7. schema van de onderstreeproutine

A4HLP8.TXT (4k) 8. schema van de 'hulpjes'

A4HLP9.TXT (2k) 9. lijst van variabelen

A4EPSON.OVL (2k) aanpassing van A4.BAS voor matrixprinter met epson-codes. na LOAD "A4.BAS" volgt MERGE "A4EPSON.MRG" en u kunt verder. A4EHL0.TXT tot en met A4EHL9.TXT is de hiervoor aangepaste versie van A4HLP*.TXT

POGO.BAS (8k) herziene versie van 'turtle graphics'-interpreter van N504. nu wel met de beschrijving POGO.HAN (12k). ESTATE, FLOWER, PATTERN, POLSPI en TEST zijn (verbeterde) pogo-voorbeeldprogramma's (elk 1k)

bestelnummer N314 (diskette)

ASM.COM (6k) z80-assemblerprogramma door nicolas hatzirvassanis.

OBJ.COM (2k) maakt van het gecreëerde .OBJ-bestand een .COM-file;

LAB.COM (2k) maakt van het gecreëerde .DAT-bestand een tekstbestand met de labels en hun waarden. toelichting in ASM.DOC (4k); source in ASM1.ASM (12k), ASM2.ASM (12k), OBJ.ASM (2k) en LAB.ASM (2k)

DIS.COM (6k) z80-disassemblerprogramma door nicolas hatzirvassanis. cp/m-versie van DISASSEM.BAS (van deel N206). met source in DIS1.ASM (10k) en DIS2.ASM (14k). toelichting in DIS.DOC (5k)

DISEXP.BAS (2k) met DISEXP.BIN (4k) is de aanpassing door wim luijt van DISASSEM.BAS voor de 96k newbrain. deze aanpassing heeft als voordeel, dat andere z80-newbrain programma's op hun originele (32k) plaats in het geheugen kunnen worden gedisassembleerd.

DISASSEM.DOC (16k) is de uitvoerige (engelse) toelichting bij DISASSEM LIST.COM (2k) programma om assemblerlistings in 2 of 3 kolommen (naar keuze) te maken. alleen te gebruiken voor listings! met source LIST.ASM (6k)

DISK.COM (4k) schijfmonitorprogramma: informatie over de schijfinhoud. dir (ook gewiste bestanden), bitmap, dec/hex-conversie, sectoren lezen. werkt alleen op drive A. source DISK.ASM (12k)

ERASE.COM (2k) programma om op een nette manier bestanden te verwijderen. met source ERASE.ASM (2k)

LINED.COM (6k) regeeditor door nicolas hatzirvassanis, waarbij sommige opdrachten van de newbraineditor gebruikt kunnen worden. met handleiding LINED.DOC (4k) en de source code in twee stukken: LINED1.ASM (12k) en LINED2.ASM (10k)

RELOBJ.BAS (4k) en RELOC.BAS (2k) beveiligingsprogramma's. hierbij horen de laadprogramma's TRUN.BAS (2k) en LADER.BAS (2k). bespreking door wim luijt in het artikel op pagina 8 (= RELO.TXT (2k)). gebruiksaanwijzing in RELO.DOC (10k)

TYP.COM (2k) programma om onder cp/m basic-programma's te listen en andere bestanden zonder CR/LF aan het einde van de regel (onder andere bovengenoemde .ASM-bestanden)

bestelnummer DIVERS-7 (cassette)

TEKSTBEW.BAS (6k) tekstbewerker van ger van laere, voor newbrain 32k en 96k. met gebruiksaanwijzing TEKSTUIT.001, .002 en .003 (resp 6k, 4k en 4k), te lezen met het programma zelf

POGO.BAS (8k) herziene versie van 'turtle graphics'-interpreter van N504. nu wel met de beschrijving POGO.HAN (12k). ESTATE, FLOWER, PATTERN, POLSPI en TEST zijn (verbeterde) pogo-voorbeeldprogramma's (elk 1k)

DISEXP.BAS (2k) met DISEXP.BIN (4k) en DISASSEM.DOC (16k): zie de toelichting bij schijf N314

RELOBJ.BAS (4k) en RELOC.BAS (2k) beveiligingsprogramma's. hierbij horen de laadprogramma's TRUN.BAS (2k) en LADER.BAS (2k). bespreking door wim luijt in het artikel op pagina 8 (= RELO.TXT (2k)). gebruiksaanwijzing in RELO.DOC (10k)

deel n314

TOELICHTING BIJ ENKELE PROGRAMMA'S DIE VERSCHENEN ZIJN
OP BESTELNUMMER N314 VAN DE SOFTWAREBIBLIOTHEEK



In mijn niet gecatalogiseerde schijven vond ik een aantal programma's, die ik mede-NewBraingebruikers niet wil onthouden. Het betreft een aantal NewBraingereedschappen afkomstig van NBUG en Grundy. Voor zover er documentatie aanwezig was, heb ik die bijgevoegd. Dit artikel is een aanvulling hierop, gebaseerd op eigen ervaringen met deze software.

GROEP 1 PROGRAMMA-ONTWIKKELING

Een aantal programma's voor programma-ontwikkeling, hoofdzakelijk voor cp/m, maar ook voor het operating system van de 32k NewBrain. De meeste programma's zijn zonder aanpassingen alleen geschikt voor een 32k NewBrain. Zie ook de opmerkingen over omzetting naar 96k NewBrain. Inclusief source codes in het formaat van device 12 (.ASM). Onder andere:

- 1 cp/m-regeeditor waarbij de opdrachten van de NewBraineditor gebruikt kunnen worden
- 1 assembler; niet getest, voor bijzonderheden zie de handleiding ASM.DOC
- 1 cp/m-versie (DIS) van de disassembler voor NewBrain (DISASSEM; uitgegeven op deel N206 van de softwarebibliotheek); lijkt veel op NEWMON. DISASSEM herkent de combinaties RST xx-DB uit het operating system van de NewBrain. Is uitgebreider dan Z80-DIS (deel N206), maar minder veelzijdig dan DSM (deel N312). Net als NEWMON bevat het een relocate-opdracht. Mijn indruk is echter, dat deze opdracht niet perfect werkt. Opletten dus! Is door mij aangepast voor de 96k NewBrain als DISEXP.BAS met DISEXP.BIN. Deze aanpassing heeft als voordeel, dat andere z80-NewBrainprogramma's op hun originele (32k plaats) in het geheugen kunnen worden gedisassembleerd.

GROEP 2 BESCHERMING EN SNELLADEN

Een aantal programma's afkomstig van Grundy en waarschijnlijk bedoeld om de mogelijkheden van de NewBrain uit te breiden, maar die door het fiasco van Grundy geen bekendheid hebben gekregen. Beter laat dan nooit. De bedoelde programma's zijn:

- RELOC
- RELOBJ met het bijbehorende TRUN

De benodigde documentatie (in het Engels) is ontleend aan het blad van NBUG. De source codes zijn helaas niet gevonden. Om de methode van deze programma's te ontrafelen heb ik ze met DISEXP gedisassembleerd. Deze programma's hebben volgens de documentatie van NBUG drie doelen:

- 1 'bescherming' van basicprogramma's door ze inbraakvrij te maken
- 2 mogelijkheid om basicprogramma's snel te laden en eventueel sneller te starten
- 3 aanpassing van basicprogramma's om ze in de Grundy's rombox te kunnen gebruiken

Om de werking van de programma's te begrijpen is een beknopte uitleg nodig over hoe een basicprogramma het ram-geheugen van een 32k NewBrain gebruikt. Een uitvoerige uitleg is te vinden in het Software Technical Manual, in Technical Note nr 4, of het boek van Braga.

Na het laden van een basic staan de regelnummers in de Line Number Table (LNT) en de opdrachten (source) in de Source Code Area (SCA). De keywords zijn vervangen door 'tokens' van één byte. Wanneer een basicprogramma op cassette of schijf is opgeslagen zijn de keywords op dezelfde wijze door symbolen vervangen. De wijze van opslag van de regelnummers op cassette of schijf is echter anders dan die in de LNT. Ze staan decimaal op cassette of schijf (op dezelfde wijze als ze worden ingevoerd), met een nummer voor iedere regel. In de LNT heeft ieder statement (elke opdracht afgesloten door een dubbele punt of een geregeleinde) een afzonderlijke ingang van zes bytes: twee voor het regelnummer, twee voor de pointer naar de bijbehorende plaats in de SCA en twee voor die in de Object Code Area (zie onder). Na het laden van een programma zijn alleen de LNT en de SCA 'gevuld'. Verder zijn de 'Fixed Locations'



van basic aangepast. Wordt nu een basicprogramma geRUNd, dan wordt ieder statement 'gecompileerd' op het moment, dat dit voor de uitvoering van het programma nodig is. Dit 'compileren' levert een aantal pointers en parameters op, die in de Object Code Area (OCA) worden geplaatst. Verder worden de overige tabellen geïnitieerd. Wordt een bepaald statement in een programma nooit uitgevoerd (bijvoorbeeld omdat nooit aan een gestelde voorwaarde wordt voldaan), dan wordt dat statement ook nooit gecompileerd.

RELOC bewaart eerst een korte z80-routine en vervolgens van een basicprogramma een afbeelding van de LNT en de SCA op cassette of schijf. TRUN reserveert voldoende geheugen, zet het .RLC-bestand in dit gereserveerde gedeelte en start de z80-routine. Deze zet de inhoud van de LNT en SCA weer op de juiste plaats in het geheugen en start het programma. Als het programma eindigt, verwijdert de routine het programma uit het geheugen. Als nu in het programma een ON BREAK GOTO de STOP-toets deactiveert, dan is het programma onLISTbaar geworden. Basicprogramma's die op deze wijze bewerkt worden, krijgen .RLC als toevoeging. Voor het laden heeft TRUN een gereserveerd gedeelte ter grootte van het .RLC-bestand nodig. De grootte hiervan moet voor ieder programma worden aangepast. Helaas gaat dit ten koste van het vrije geheugen, dat voor het programma beschikbaar is. Nu blijft het programma wel werken, als dit gedeelte geannuleerd (gedeRESERVEerd) wordt. Dit is te bereiken door als eerste opdracht in het basicprogramma op te nemen:

`I RESERVE I : RESERVE 2115 + TOP`

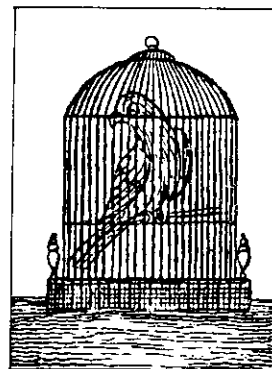
De TOP wordt dan weer op 32768 gezet. Waarschijnlijk zijn er ook mogelijkheden om de z80-routines in basicprogramma's in een geRESERVEerd stuk geheugen achter te laten, zodat deze routines niet opnieuw gePOKEt hoeven te worden. Ook lijkt het mij mogelijk om het laden met TRUN nog sneller te maken. TRUN vraagt namelijk welk bestand (= programma) er geladen moet worden. Deze opdracht kan achterwege blijven, als voor ieder afzonderlijk programma een aangepaste versie van TRUN bewaard wordt.

Met de rombox van Grundy hoeft het geRELOCte programma niet op cassette te worden opgeslagen, want dan zorgt de rom voor de permanente opslag van het programma. De rombox kan niet in combinatie met de diskcontrollor worden gebruikt, zodat opslag op schijf dan niet mogelijk is. Bij het starten van de NewBrain worden eerst de ram-test en het opzetten van de zeropage uitgevoerd. Vervolgens wordt niet naar basic gespron-

gen, maar automatisch het geRELOCte programma geladen en gestart. Het geRELOCte programma (in rom) hoeft niet naar een gereserveerd stuk ram te worden overgebracht, zodat het gehele ram voor het programma beschikbaar blijft.

RELOBJ

Het ontrafelen van RELOBJ leverde minder informatie op dan het ontrafelen van RELOC. Het is dus mogelijk, dat wat ik hierover vermeld, niet (geheel) juist is. Het bewerken met RELOBJ levert nauwelijks snelheids-



winst op bij de uitvoering van het basicprogramma (zie onder), zodat ik minder diep op de werking ben ingegaan. RELOBJ maakt programma's met de toevoeging .RLX aan de naam. RELOBJ zet eerst weer een z80-routine op cassette of schijf. Vervolgens 'compileert' het een te bewerken programma, dat wil zeggen het vult niet alleen de LNT en SCA in, maar alle door basic gebruikte geheugenstructuren, zoals de Object Code Area (OCA), Symbol Table (ST), enzovoort. De grootte van deze structuren mag tijdens de uitvoering van het basicprogramma niet veranderen, zodat alle array's in het program-

ma geDIMensioneerd moeten worden. Daarna zet het de inhoud van deze structuren, met uitzondering van de SCA, op de cassette of schijf. De inhoud van de SCA wordt 'leeg' bewaard. Hierdoor is het onmogelijk om na het laden met TRUN het programma (of gedeelten ervan) opnieuw te 'compileren'. De inhoud van de OCA mag in de RUN-fase niet vernietigd worden. Basicopdrachten die dit doen, mogen dus niet in het programma voorkomen. Dit zijn: CLEAR, MERGE, DELETE, NEW, STOP, END, REPORT, LOAD en RUN. RESERVE mag ook niet gebruikt worden, omdat dit de adressen van de basicstructuren verandert, zodat ze niet meer in overeenstemming zijn met de toestand tijdens RELOBJ. Hierdoor is een stuk geheugen ter grootte van het door RELOBJ geRESERVEerde geheugen (1767 bytes) plus het aantal bytes gelijk aan de lengte van het .RLX-bestand niet meer beschikbaar is voor het basicprogramma. END mag voorkomen in het programma, maar heeft als nadeel, dat het programma opnieuw moet worden geladen om het opnieuw te RUNnen. Komt END niet voor, dan moet het programma met NEW beëindigd worden, omdat het anders niet mogelijk is naar basic terug te gaan en andere programma's te RUNnen.

VERGELIJKING VAN RELOC MET RELOBJ

Beide programma's geven bescherming tegen 'inbraak' in een basicprogramma. De bescherming van RELOC is maar betrekkelijk, die van RELOBJ 100 % wat betreft het ontrafelen van de oorspronkelijke programmalisting. Bescherming tegen kopiëren biedt geen van beide programma's. Het laden en starten van basicprogramma's gaat in bewerkte vorm veel sneller. De winst die RELOBJ ten opzichte van bewerking met RELOC oplevert, is betrekkelijk. Bij het RUNnen van een bewerkt programma wordt alleen de 'compilatie'-fase overgeslagen. Het is niet zo, dat een basicprogramma in een snel programma in z80-code wordt omgezet. De geringe snelheidswinst gaat ten koste van het vrije geheugen, dat voor het basicprogramma beschikbaar is. Met een .RLC-programma loop je het risico, dat basic, ten gevolge van gebrek aan vrij geheugen, de object code gaat vervangen tijdens het RUNnen van het basicprogramma. Is die object code later in een programma weer nodig, dan kan die niet opnieuw gecompileerd worden, omdat de source code ontbreekt. Bovendien is het .RLX-bestand veel groter dan een vergelijkbaar .RLC-bestand waardoor het laden weer langer duurt. In de praktijk beperkt de toepassing van RELOBJ zich tot basicprogramma's van maximaal 3 à 4 kilobytes.

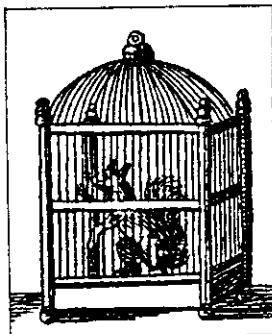
Bij het compileren controleert RELOBJ, of er genoeg geheugen vrij is, zodat een programma altijd kan RUNnen, zonder dat er gebrek aan vrij geheugen ontstaat. Aangezien strings onbeperkt van lengte kunnen worden, kan deze toestand (onvoldoende geheugen) volgens mij wel optreden. Het gebruik van RELOBJ raad ik daarom af.

In plaats van het oorspronkelijke TRUN heb ik LADER.BAS geschreven om een RLC-programma te laden. De gebruiker moet dit zelf aanpassen aan het gewenste programma.

GEBRUIKSAANWIJZING RELOC

Gebruik alleen een 32k NewBrain. Bij mij werkt het ook op een 96k, wanneer die met UNEXPAND.BAS tot 32k verkleind is.

- LOAD en RUN RELOC
- volg de aanwijzingen op het scherm
- RESERVE 2↑15 + TOP
- OPEN #1, "filenaam.RLC"
- FOR i = 1 TO 65000 : GET #1, x : NEXT i



- als ERROR 131 of ERROR 133 op het scherm komt: PRINT i
- onthoud de waarde van i
- LOAD LADER.BAS
- LIST en wijzig het programma zoals aangegeven in de REMs
- verwijder de REMs
- SAVE onder een eigen naam

LADER.BAS

```
10 GOSUB100
20 RESERVE 1 : RESERVE 2 ↑ 15 + TOP : RESTORE
30 a = i + 2 : REM vervang i door de lengte van het .RLC-
  bestand
40 a$ = "filenaam.RLC" : REM vervang filenaam door de gekozen
  filenaam; bij gebruik van cassette mag de filenaam worden
  voorafgegaan door *20: of *40:
50 FOR i = 0 TO 23 : READ b : POKE 121 + i, b : NEXT i :
  REM installeer de z80-routine op een vrij gedeelte van de
  systempage
60 RESERVE a
70 OPEN #1, a$
80 CALL 121, a
90 GOSUB 100 : END
100 FOR i = 1 TO 255 : CLOSE #i : NEXT i : RET
110 DATA 86, 43, 94, 135, 131, 39, 135, 129, 17, 1, 0, 231, 60,
  175, 231, 52, 201
```

Helaas heeft het mij door onvoorziene omstandigheden aan tijd ontbroken om alle genoemde programma's uitvoerig te testen. Toch vind ik de programma's zo interessant, dat ik ze in deze min of meer onbewerkte vorm aan de gebruikersgroep wil doorgeven. Als er belangstelling is voor deze programma's in verband met de toepassing van basicprogramma's in romboxen, dan zou ik dit gaarne vernemen. Mogelijk, dat ik er in een volgende On-line op terugkom.

Wim Luijt





GRUNDY BUSINESS SYSTEMS LIMITED
(IN LIQUIDATION)

Final report on the conduct of the liquidation of the above named company.

1. GRUNDY (TEDDINGTON) GROUP LIMITED

You will recall that, at the first meeting of creditors which was held on 8th September 1983, the questions raised mainly related to the involvement of the Grundy (Teddington) Group Limited. After substantial investigations, I instructed solicitors to instigate proceedings against 'Group'. A trial date for the hearing of an alleged fraudulent preference claim against 'Group' was set down for the last week in February 1988. Shortly prior to the proposed date of the hearing, the defendants solicitors proposed an out of Court settlement. After much negotiation, settlement was reached and the sum of £ 75,000 was subsequently passed over to me by the solicitors acting for 'Group'. In addition, 'Group' have dropped all of their claims in the liquidation, including preferential and unsecured, which reduced the total creditors by a substantial sum amounting to approximately £ 487,000.

2. SALE OF ASSETS

One of my primary objectives in this liquidation had been to ensure the continued existence of the Newbrain project, both for the purpose of increasing the value of stocks and decreasing the volume of claims against the company. On 26th October 1983, after lengthy negotiations, a three party contract was signed between Thorn EMI Datatech Limited, Tradecom

BV and myself, as Liquidator. The intellectual assets of the company were assigned to Tradecom BV for a consideration which was payable on deferred terms. All stocks held by me were also sold to Tradecom BV. All assets and stocks which were sold have now been paid for.

3. OTHER LEGAL PROCEEDINGS

Legal proceedings against a company which had been a subsidiary of 'Group' had been considered. However, as this company has since been sold and is currently operating under new management, the members of the liquidation committee and myself agreed that pursual of the matter would not benefit creditors.

Legal proceedings against the factoring company were also considered. In view of the materiality of the amounts involved in this case, the liquidation committee and I agreed that the matter should be closed.

4. PREFERENTIAL CREDITORS

Preferential creditors have now been paid in full.

5. UNSECURED CREDITORS

You will recall that a first interim dividend of 8p in the £ was paid on all unsecured claims received prior to September 1984. Since that date, a number of additional claims have been received which have been approved and the appropriate dividend paid accordingly.

A secondary and final dividend of 7.864p in the £ had today been declared and the appropriate dividend cheque in respect of your claim in the liquidation is enclosed.

The primary reason for the delay in making a second and final distribution to unsecured creditors is due to the late submission of the assessment from HM Inspector of Taxes in respect of the Final Corporation Tax Computation. This matter has now been concluded and the assessment has been paid in full.

6. LIQUIDATORS REMUNERATION

The liquidation committee agreed that, in view of the length of the proceedings and the matters dealt with and considering the benefit achie-



ved by the liquidator for the unsecured creditors, the costs and expenses of the liquidator should be met in full.

You will note from the enclosed Receipts and Payments Account that an amount of £ 100,000 plus VAT was paid to me, over the entire period of the liquidation.

7. FINAL MEETING OF CREDITORS

The final meeting of creditors will be held on Thursday 27th April 1989 at 11:30 am, as per the enclosed notice of meetings.



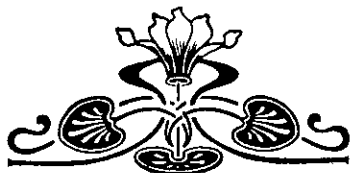
GRUNDY BUSINESS SYSTEMS LIMITED (IN LIQUIDATION)

NOTICE IS HEREBY GIVEN, pursuant to Section 300 of the Companies Act 1948 that Final Meetings of members and of Creditors of the above named company will be held at 128 Queen Victoria Street, London EC4 on Thursday, 27th April 1989 at 11 a.m. and 11:30 a.m. respectively for the purpose of having an account laid before them, and to receive the Report of the Liquidator showing how the winding-up of the company has been conducted and it's property disposed of, and of hearing any explanation that may be given by the Liquidator. Any Member or Creditor entitled to attend and vote at the above mentioned meeting is entitled to appoint a proxy to attend and vote instead of him, and such proxy need not also be a Member or Creditor.

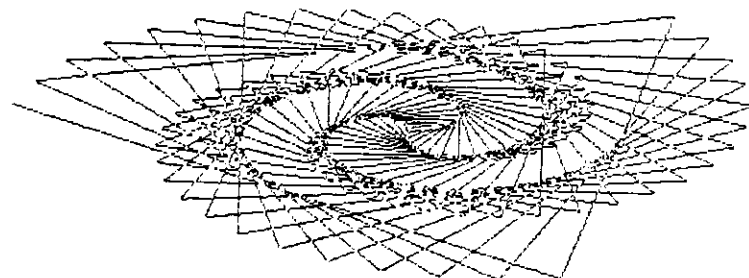
Dated this 23rd March 1989

T R Harris
Liquidator

(Brief gestuurd aan de crediteuren door de curator in het faillissement van Grundy. Overgenomen uit 'NewBrain Support Group' nr 3, juli 1989)



pogo



Pogo is geschreven als een oefening om aan niveau A informatiekunde leerlingen aan The Downs School in Dartford (Kent, Groot-Brittannië) te tonen, welke moeilijkheden je tegenkomt bij het schrijven van geïnterpreteerde talen. De oefening is tamelijk ingewikkeld geworden, en in de definitieve versie is Pogo (pico-logo) een aardige implementatie van de 'turtle graphics'-taal, die door Seymour Papert beroemd gemaakt is.

De filosofie van deze taal is die van de procedure ('mind sized bites' zoals Papert ze noemt, brokken die te overzien zijn), die opgebouwd is uit bestaande woorden van de taal en zo een heel nieuwe wereld vormt, die aan de taal toegevoegd wordt (een 'threaded interpreter').

Pogo heeft ruim twintig opdrachten, waarmee het bouwen kan beginnen. De taal werkt op twee manieren: direct en door de uitvoering van een programma. Bij de uitvoering van een programma is recursie toegestaan als onderdeel van het bouwproces.

Pogo is niet zonder beperkingen. Ten eerste is het tamelijk langzaam, omdat het dubbel geïnterpreteerd wordt.

Het programma is een basic-bestand, dat op de normale manier geladen en uitgevoerd wordt. Na de titelpagina*) staat het programma klaar voor

*) In de oorspronkelijke versie komt de titelpagina op het scherm met de keuze, of schijf- of cassette-opslag gebruikt wordt. Die pagina waarschuwt ook de gebruiker het hoofdletterslot te gebruiken alvorens verder te gaan. Nu regelt het programma die dingen zelf.

directe opdrachten, met een grafisch en een tekstscherf in beeld. De pijl geeft de richting van de pen. De opdrachtregel geeft aan, waar de tekst zal verschijnen, die op het toetsenbord ingetypt wordt. Wat er dan gebeurt, is afhankelijk van wat er ingetypt wordt. Als het opgevat kan worden als een directe opdracht, dan zal die direct uitgevoerd worden (nou ja, bijna direct).

DIRECTE OPDRACHTEN

Probeer, om te beginnen, een paar van de volgende bewegingsopdrachten: FORWARD (voorwaarts), BACK (achterwaarts), LEFT (links) en RIGHT (rechts). Tussen de opdracht en het getal moet een spatie staan. Probeer bijvoorbeeld eens het volgende in te typen.

```
LEFT 120 <newline>
FORWARD 300 <newiine>
RIGHT 45 <newline>
BACK 200 <newline>
```

Er zijn ook andere directe opdrachten beschikbaar; probeer eens:

```
CLEAR <newline>
```

Dit wist het grafische scherm en zet de pen terug in zijn beginpositie (de newline-toets moet na elke opdracht aangeslagen worden, maar dat zullen we voortaan niet meer aangeven). De opdracht PENUP tilt de pen op, zodat het mogelijk wordt om hem te verplaatsen, zonder dat er getekend wordt. Dit is ongedaan te maken met PENDOWN. Probeer bijvoorbeeld eens het volgende in te typen.

```
FORWARD 300
RIGHT 90
FORWARD 300
RIGHT 90
PENUP
FORWARD 100
PENDOWN
FORWARD 200
RIGHT 90
FORWARD 30
```



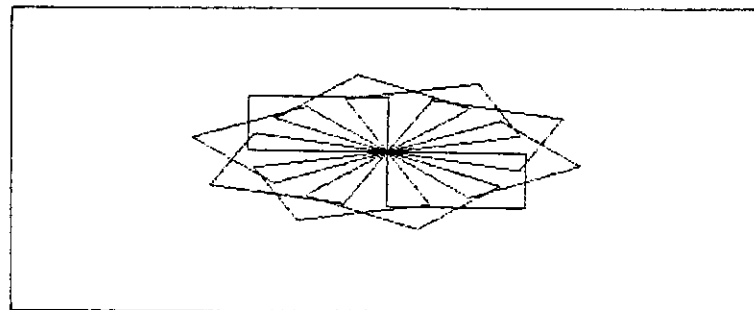
HOE LEER JE DE COMPUTER IETS?

Met Pogo is het mogelijk de computer te programmeren. Dat wordt bereikt door de computer te laten onthouden, hoe hij iets moet doen. Dit gebeurt met de opdracht TO. Probeer het volgende eens om er een idee van te krijgen, hoe zoiets gaat.

```
TO KADER (het scherm verandert in een tekstscherf)
REPEAT 4
FORWARD 200
LEFT 90
AGAIN
STOP (het grafische scherm komt terug)
```

De opdracht KADER is nu toegevoegd aan het repertoire. Type KADER om te zien, hoe die uitgevoerd wordt. Het KADER zou in een andere procedure ingebouwd kunnen worden op de volgende manier:

```
TO MOOI
REPEAT 10
RIGHT 36
*KADER (het sterretje is belangrijk: het vertelt de computer,
dat het een gedefinieerde procedure is)
AGAIN
STOP
```



De procedure MOOI zou ingebouwd kunnen worden in een volgende procedure, enzovoort, totdat er tien procedures op deze manier in elkaar gezet zijn.

Probeer als oefening eens de computer te leren, hoe hij een vijfhoek moet tekenen.

HOE LEER JE DE COMPUTER GETALLEN TE GEBRUIKEN?

Pogo laat het gebruik van numerieke variabelen toe. Dit zijn zogenaamde globale variabelen: ze kunnen door elke procedure, overal in het programma, gebruikt en gewijzigd worden. Het volgende voorbeeld is een illustratie van de manier waarop een variabele gebruikt wordt.

```
TO SPIRAAL
LENGTE := 100
REPEAT 10
  RIGHT 50
  FORWARD LENGTE
LENGTE := LENGTE + 100
AGAIN
STOP
```

De := is het 'wordt'-teken, in vele talen gebruikt. De zesde regel in bovenstaand programma gebruikt het wordt-teken om 100 bij de vorige waarde van de variabele LENGTE op te tellen, en dat als nieuwe waarde aan LENGTE toe te kennen.

Een beperking van := in Pogo is, dat alleen de operatoren +, -, / en * gebruikt kunnen worden.

Er zijn nog een paar andere mogelijkheden van Pogo, die het beste aan de hand van een voorbeeld duidelijk gemaakt kunnen worden. Sommige voorbeelden worden meegeleverd op de cassette of diskette, waar Pogo op staat. De namen van deze procedures zijn: FLOWER, ESTATE, PATTERN en POLSPI. Om ze te kunnen gebruiken, moeten ze in het computergeheugen geladen worden. Wanneer u LOAD intypt, wordt u gevraagd een bestandsnaam op te geven. Type die in en sla <newline> aan. Op dezelfde manier wordt SAVE gebruikt om alle op dat moment gedefinieerde procedures op te slaan op cassette of diskette.

BEPERKINGEN

Programma's kunnen tot 10 diep in elkaar gepiaatst worden. In een REPEAT-opdracht kan niet zomaar een tweede REPEAT gebruikt worden. Dat kan alleen door een afzonderlijke procedure (met daarin de tweede REPEAT-opdracht) te gebruiken.

Als er teveel programma's in elkaar geplaatst zijn, dan komt bij de

uitvoering van het programma deze foutmelding op het scherm: *** error
*** nesting too deep (in de oorspronkelijke versie: ** SORRY! ** I
CAN'T CARRY ON WITH THIS **).

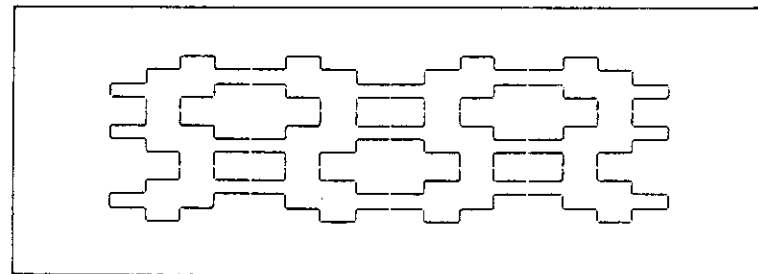
Het programma raakt in de war en geeft onvoorspelbare resultaten, als voor procedure- of variabelenamen de namen van de ingebouwde opdrachten genomen worden, bijvoorbeeld LEFT als variabelenaam.

TWEE EXTRA OPDRACHTEN

Om het werken met willekeurige getallen mogelijk te maken, zijn in een late fase van de ontwikkeling de opdrachten RANDOM en POSITION ingevoerd. Voorbeeld van de syntaxis:

```
XPOS := RANDOM 100
POSITION XPOS YPOS GRADEN
```

Eerst wordt een random getal van 1 tot 100 aan de variabele XPOS toegekend. De tweede opdracht zet de pijl op de coördinaten XPOS en YPOS, waarbij de pijl wijst in de richting die door de variabele GRADEN gegeven wordt (uiteraard is elke geldige variabele toegestaan bij de opdracht POSITION, en nu ook absolute getallen).



LIJST VAN OPDRACHTEN

Hier is een overzicht van de opdrachten, die door Pogo gebruikt worden, met hun betekenis.

FORWARD verplaatst de pen voorwaarts, in dezelfde richting als de pijl.
Net als bij BACK moet er een getal of een variabelenaam bij staan, bijvoorbeeld FORWARD 240 of FORWARD AFSTAND

BACK verplaatst de pen achterwaarts, in de tegengestelde richting van de pijl. Er moet een getal bij staan, of de naam van een variabele, bijvoorbeeld BACK 300 of BACK BREEDTE

LEFT verandert de richting, waarin de pijl wijst. Het moet altijd gevolgd worden door een getal of een variabele. De pijl draait linksom, dus tegen de richting van de klok in. Voorbeeld: LEFT 45 of LEFT HOEK

RIGHT verandert de richting, waarin de pijl wijst. Het moet altijd gevolgd worden door een getal of een variabele. De pijl draait naar rechts, dus met de klok mee. Voorbeelden: RIGHT 45 of RIGHT DRAAI

PENUP betekent, dat 'de pen van het papier gehaald wordt': er wordt vanaf dat moment geen lijn getrokken, wanneer de pijl verplaatst wordt

PENDOWN betekent, dat 'de pen op het papier gezet wordt': bij het verplaatsen van de pijl blijft er een lijn achter op het scherm

POSITION plaatst zonder een lijn te trekken de pen op de opgegeven plaats en in de opgegeven richting. POSITION wordt gevolgd door drie parameters (getallen of variabelen): eerst de horizontale coördinaat, dan de verticale coördinaat en tenslotte de richting waarin de pijl moet wijzen

FILL kleurt het vak waarin de pen staat, in. Als het vak getekend is op de langzame manier (SLOW), gaat het vrijwel zeker mis: bij het weghalen van de pijl ontstaat er een gaatje in de lijn, waardoor de figuur uitloopt

HOME verplaatst de pen naar het midden van het scherm, zonder het scherm te veranderen

CLEAR veegt het scherm schoon en zet de pen terug in het midden

FAST schakelt de pijl uit, en opdrachten worden niet afgedrukt bij de uitvoering van een procedure

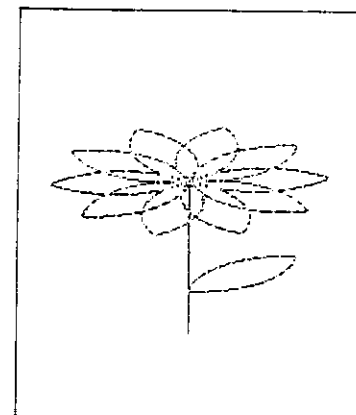
SLOW doet het omgekeerde van FAST

TO maakt het mogelijk een nieuwe procedure te definiëren. Het

wordt gevolgd door de naam van de procedure, bijvoorbeeld TO SPIRAAL

STOP is de afsluiting van het definiëren van een nieuwe procedure (begonnen met TO)

:= (uitgesproken als 'wordt') verandert de waarde van een variabele of initialiseert een nieuwe variabele. Bijvoorbeeld ZIJDE := 30, HOEK := 360 / 10, LENGTE := LENGTE - 10, HOOGTE := HOOGTE * 35



RANDOM geeft willekeurige waarde aan een variabele

IF maakt een keuze van de handeling mogelijk. Een variabele kan met elke andere vergeleken worden, of met een getal, waarbij de volgende operatoren toegelaten zijn:

= isgelijk
< kleiner dan
> groter dan

voorbeelden van het gebruik:

```
IF PUNTEN < 10 THEN FORWARD ZIJDE
IF ZIJDE = 0 THEN *BOX
IF FRED > WIM THEN WIM := WIM * 24
```

REPEAT wordt gebruikt om een verzameling opdrachten te herhalen. Het moet gevolgd worden door een getal of een variabele (namelijk het gewenste aantal herhalingen). De opdracht AGAIN wordt gebruikt om het einde aan te geven van de lijst van opdrachten, die herhaald moeten worden

AGAIN wordt alleen gebruikt na de opdracht REPEAT. Alle opdrachten tussen REPEAT en AGAIN zullen herhaald worden zo vaak als achter REPEAT aangegeven staat

- NEW wist alle bestaande procedures uit het geheugen
- LIST toont alle procedures die op dat moment gedefinieerd zijn
- LOAD laadt een verzameling procedures van cassette of diskette
- SAVE slaat de procedures, die in het geheugen staan, op cassette of diskette op
- BASIC is de enige manier om Pogo te verlaten. Het grafische scherm wordt dan gesloten. Terugkeer naar Pogo met behoud van variabelen en gedefinieerde procedures is mogelijk met de opdracht GOTO 18

IDEËN

Net als Logo is deze taal bedoeld om creatief te zijn. Kinderen gaat het beter af om de voordelen van zo'n taal uit te buiten. Wat vindt u van het volgende, om er een beetje in te komen:

1. een aanpassing van het programma POLSPI om vijfhoeken in plaats van driehoeken te tekenen;
2. een tuin vol bloemen, met het programma FLOWER als uitgangspunt;
3. een mozaïek van zeshoeken over het hele scherm.

Simon Murphy
(vertaling en bewerking: menno stevens)

PROGRAMMA VERKRIJGBAAR OP CASSETTE EN DISKETTE ZIE PAGINA 6

bij de bewerking van het programma zijn enige wijzigingen en verbeteringen toegepast:

- † foutmelding is nauwkeuriger en uitgebreider
- † absolute getallen en variabelen kunnen vaker elkaars plaats innemen
- † compactere code (van 10k naar 6½k); geen ongebruikte opdrachten
- † 'random' is nu op elke plaats te gebruiken waar een getal of een variabele moet staan

gebufferd

HET GEBUFFERDE TOETSENBORD
OP DE 32K NEWBRAIN MET DISKCONTROLLER

onder cp/m heeft de newbrain de beschikking over een gebufferd toetsenbord en een ander schermdevice (het adm3a-scherm). op de 96k-machines kan ook onder basic daarmee gewerkt worden: dan zijn het de devices 19 en 18. de diskcontroller verschaft ze aan de 32k-machine voor gebruik onder cp/m. dan zeg je, dat je er ook onder basic bij zou moeten kunnen. wim luijt heeft het daar niet bij gelaten en is op zoek gegaan naar een methode om het gebufferde toetsenbord onder basic aan de praat te krijgen. door tijdgebrek is deze onderneming in de ontwikkeling blijven steken. omdat de theoretische beschouwingen op zich wel interessant zijn, volgt hier het verslag van zijn bevindingen. wellicht voltooit iemand anders het project

Zoals al eerder is vermeld, bevat de diskcontroller in de rom de device driver voor de devices 18 en 19 (het gebufferde toetsenbord en het ADM3a-scherm). Deze drivers gebruiken echter zcalls die alleen onder het paged operating system aanwezig zijn. Pogingen om deze devices op een 32k NewBrain met diskcontroller te gebruiken zijn dus gedoemd te mislukken. 32k-cp/m gebruikt echter wel een gebufferd toetsenbord, zodat er toch iets mogelijk moet zijn.

METHODE

De structuur en het installeren van een device driver is al meerdere keren beschreven (onder andere in On-line 8) en kunnen we hier dus overslaan. Als invoerroutine gebruiken we CONIN uit cp/m. Deze kijkt, of er een of meer tekens in de toetsenbordbuffer staan, en zo ja, dan plaatst ze het eerste teken in register A. Het device wacht, totdat er een teken beschikbaar is, en is zo te vergelijken met device 5. De buffer wordt gevuld door middel van een gewijzigde interruptroutine. De cp/m-interrupt-routine is helaas niet te gebruiken, omdat deze is gebaseerd op een vast scherm, wat niet in de streamstructuur van de NewBrain past en ook de

cassettepoorten onbruikbaar maakt. Alleen het gedeelte van de interruptroutine dat naar de toetsenbordinterrupts leidt, moet worden gewijzigd. Deze wijziging kost 80 bytes. Verder is er een stuk geheugen nodig om als eigen geheugenruimte (onder andere buffer) te fungeren. Als we hier van de streamstructuur afwijken, kunnen we hetzelfde stuk geheugen nemen als cp/m hiervoor gebruikt. Dit heeft twee voordelen:

- (gedeelten van) de bestaande routines in rom kunnen worden gebruikt;
- de ram in de diskcontroller wordt gebruikt, zodat de buffer en de driver niet ten laste van de 32k ram komen.

BESCHRIJVING VAN DE DEVICE DRIVER

Definitie: gebufferd toetsenbord op de 32k NewBrain met diskcontroller; 128 byte buffer; gebruikt alleen reservegeheugen; maakt de devices 0, 4, 5 en 6 onbruikbaar (0 en 4 alleen voor uitvoer te gebruiken); gewijzigde interruptroutine.

OPENIN en OPENOUT: het device kent alleen invoer (INPUT). Er is dus maar een OPEN-routine. OPEN moet de nieuwe interrupt activeren en de eigen geheugenruimte installeren. Voor beide kunnen we een stuk uit de BOOT van 32k-cp/m gebruiken, als we enige adressen aanpassen. INPUT bestaat uit een jump naar CONIN, terwijl de OUTPUT niet gedefinieerd hoeft te worden. CLOSE bestaat uit het herstellen van de oorspronkelijke interrupt.

De normale interruptroutine voor het toetsenbord zet in de COPBUFF (3Dh) de laatste toetsaanslag. Deze waarde wordt gebruikt door de systeemroutines (zcalls) GETKEY en IMMKEY, die op hun beurt weer worden gebruikt door de devices 0, 4, 5 en 6. Dit adres wordt na het veranderen van de interruptroutine niet meer ververs. Gebruik van het nieuwe device sluit dus het gebruik uit van deze devices en van de zcalls GETKEY en IMMKEY voor invoer vanaf het toetsenbord. De invoer via deze routines blijft dezelfde, onbepaalde waarde geven. Hetzelfde geldt ook voor de zcall BRKTST. De cp/m-interrupt scant niet of de STOP-toets is ingedrukt, dus deze toets werkt dan ook niet meer. De buffer bestaat uit 128 bytes en bevindt zich geheel in de DC-ram. In deze ram bevinden zich nog 80 ongebruikte adressen, wat de mogelijkheid biedt om de gewijzigde interruptroutine te plaatsen (9F7Dh - 9FCCh).

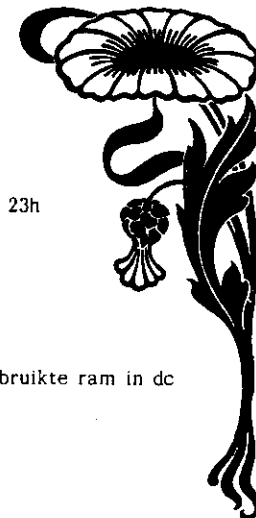
Op dezelfde wijze is ook device 19 te installeren, namelijk door de CONOUT-routine uit cp/m te nemen. Er is hier echter een probleem. Het

videogeheugen heeft onder basic geen vaste plaats. De interruptroutine zoekt uit, waar dit begint, om het de hardware mogelijk te maken dit videogeheugen op de monitor af te beelden. Onder cp/m echter heeft het videogeheugen een vaste plaats (7380h - 7FFFh). De cp/m-interruptroutine hoeft dus niet uit te zoeken, waar dit begint. Willen we CONOUT gebruiken, dan moeten we het videogeheugen op dezelfde plaats zetten. Een en ander lijkt mij onmogelijk. Wel lijkt het mij mogelijk om het scherm en het toetsenbord tot 1 device te combineren.

Helaas ontbreekt mij de tijd om een en ander uit te zoeken. Als er andere gebruikers zijn, die dit zouden willen doen, dan zal ik ze gaarne advies geven.

Wim Luijt

0000	=	org 0	; 79 + 23h
8f76	=	conin equ 8f76h	
8e7e	=	ckbint equ 8e7eh	
905f	=	cboot equ 905fh	
908e	=	eboot equ 908eh	
e198	=	int equ 0e198h	
9f7d	=	nint equ 9f7dh	; ongebruikte ram in dc
9ed9	=	kbuf equ 9ed9h	
0000	04	db 4	
0001	01	db \$ - kbbop	
0002	02	db \$ - kbbop	
0003	fd	db \$ - kbbout	
0004	00	db 0	
0005	fc	db \$ - kbbcls	
0006	c3768f	kbbout jp conin	
0009	f3	kbbcls di	
000a	2198e1	ld hl, int	
000d	223800	ld (56), hl	
0010	fb	ei	
0011	c9	ret	
0000	=	org 0	
0000	2198e1	kbbop ld hl, int	; oude interrupt
0003	117d9f	ld de, nint	; t/m bit copintbar, a
0006	011200	ld bc, 0e1adh - 0e19bh	
0009	edb0	ldir	



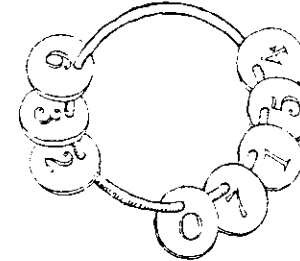
```

000b 3ec2      ld a, 0c2h      ; jp nz naar oude int
000d 12        ld (de), a      ; bit clkintbar, a
000e 13        inc de
000f 3eaf      ld a, 0afh      ; = elaf
0011 12        ld (de), a
0012 13        inc de
0013 3ee1      ld a, 0e1h
0015 12        ld (de), a
0016 13        inc de
0017 2125e2    ld hl, 0e225h    ; copint t/m sub b op e23b
001a 011700    ld bc, 0e23ch - 0e225h
001d edb0      ldir
001f 212a00    ld hl, weg      ; rest gewijzigd
0022 010600    ld bc, wegl - weg
0025 edb0      ldir
0027 c3d99e    jp kbuf
002a da7e8e    weg jp c, ckbint
002d c3ee2     jp 0e23eh      ; oude int sub b op e23e
0030 =        wegl equ $
9ed9 =        org kbuf      ; plaats de routine in de buffer
9ed9 f3        di
9eda 216090    ld hl, cboot + 1
9edd 011700    ld bc, 9076h - cboot
9ee0 11f69e    ld de, eind      ; maak er een vervolg van
9ee3 edb0      ldir
9ee5 23        inc hl          ; laat de sp ongewijzigd
9ee6 23        inc hl
9ee7 23        inc hl
9ee8 011800    ld bc, 9091h - 9079h
9eeb edb0      ldir
9eed 3ec9      ld a, 0c9h
9eef 12        ld (de), a
9ef0 217d9f    ld hl, nint
9ef3 22049f    ld (eind + 906dh - cboot), hl
9ef6 =        eind equ $

```

cboot	905f	int	e198	nint	9f7d
ckbint	8e7e	kbbcls	0009	weg	002a
conin	8f76	kbbop	0000	wegl	0030
eboot	908e	kbbout	0006		
eind	9ef6	kbuf	9ed9		

pos



PAGED MEMORY OPERATING SYSTEM

Verleden jaar heb ik op de NewBraindag een lezing gehouden over de geheugenindeling van de NewBrain, zowel van de 32k als de 96k. Onlangs kwam ik weer het schema tegen, dat ik toen verstrekt heb. Dit artikel is om deze kennis vast te leggen en te integreren met wat ik eerder heb geschreven over het paged memory operating system. De beste bron voor gedetailleerde informatie over het pos, voor zover die niet in de bijgeleverde documentatie staat, is te vinden in de source code van het pos. Deze is nooit uitgegeven, maar is bij mij op schijf verkrijgbaar. Ook kan met onderzoek van de system page een aantal details worden gevonden (zie On-line 8, pag. 65, en On-line 10, pag. 55).

Voor de duidelijkheid even een korte samenvatting. De z80 kan 64k geheugen adresseren. Dit wordt ook wel de z80-map genoemd. Iedere geheugenplaats heeft een uniek adres in het bereik van 0000h tot en met FFFFh. Deze 64k kunnen we verdelen in stukken van 8k, ook wel sloten genoemd. Een slot heeft als beginadres een veelvoud van 2000h. Ze beginnen dus op 0000h, 2000h, 4000h enzovoort.

In de documentatie wordt dikwijls de term 'gebruikersprogramma' (user program) gebruikt. Dit is, gezien vanuit het operating system, het programma, dat de gebruiker gaat gebruiken voor zijn (toepassings)programma's. Het gebruikersprogramma gebruikt voor zijn in- en uitvoer het operating system via de zogenaamde zcalls en kan ook gebruik maken van andere routines in de rom. Op het moment zijn beschikbaar: basic, cp/m, z80-machinecode en comal (in de rombox). Het gebruikersprogramma moet dus niet verward worden met een programma, zoals Taxy2, Pascal (beide onder basic) of PIP (onder cp/m).

In de 32k is 56k van de totale z80-adresruimte gevuld: viermaal 8k met ram en driemaal 8k met rom. De adressen van de afzonderlijke (hardware-) rom's en ram's liggen vast. De rom's kunnen daarom benoemd worden aan de hand van de plaats, die ze in de geheugenruimte innemen. De CD-rom heeft als adressen C000h tot en met DFFFh. Een andere naam voor deze rom is de basic-rom, omdat de code voor de basic in deze rom staat. De EF-rom (of OS-rom) bevat het operating system en de mathematische routines, terwijl de AB-rom (of graphics-rom) de grafische routines voor device 11 bevat.

Er is echter een complicatie. De AB-rom bevat ook een gedeelte van het operating system. De EF-rom is (vermoedelijk) gemaakt voor de 'voorloper' van de NewBrain, die hardwarematig iets anders in elkaar zit (het Software Technical Manual behoort ook bij deze voorloper). Ook zitten er enkele fouten in de routines van deze rom. In plaats van de EF-rom te corrigeren in latere produktieversies heeft men ervoor gekozen om aanvullingen en verbeteringen in de AB-rom aan te brengen. Dit geldt voor de OS-versies lager dan 2.0. In versie 2.0 is de rom-indeling sterk gewijzigd. Hier volgt een overzicht wijzigingen ten opzichte van de voorloper:

versie 1.4

- † wijziging OPEN-routine van device driver 0 en 3
- † toevoeging system calls voor BLOCKIN en BLOCKOUT (RST20h-routine 3Ch en 3Dh)
- † opname van de routine FINDSTREAM als RST20h-routine 3Bh

versie 1.9

- † een kleine wijziging voor het starten van expansions
- † wijzigingen in de grafische routines

versie 1.91

- † een geheel nieuwe mathematische routine voor SQRT

versie 2.0

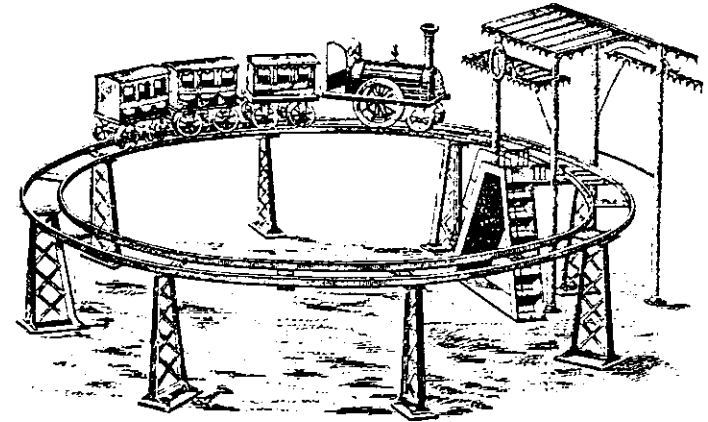
- † de rom's van deze versie zijn opnieuw ingedeeld (voor details zie de source listing van de NewBrain, softwarebibliotheek deel N216 - N218)

Wil men de NewBrain zonder AB-rom (bijvoorbeeld met behulp van de rombox) laten werken, dan is dit mogelijk met het verlies van het grafische pakket. Wel moet de EF-rom worden verbeterd.

Zoals uit het overzicht blijkt, is de ruimte van 8000h tot en met 9FFFh vrij. Deze 8k kan nu op verschillende manieren benut worden, waarbij de verschillende mogelijkheden elkaar uitsluiten:

- 1 rom voor de diskcontrollerbesturing en device drivers;
- 2 rom met programma's of uitbreiding van het operating system;
- 3 extra ram (zie On-line 6, pag. 75) of met een gewijzigde rombox (zie deze On-line, pag. 43).

Bij mogelijkheid 1 is er een stukje ram nodig met een adres groter dan 8000h. Dit bevindt zich van 9C00h tot 9FFFh.



DISKCONTROLLER

De diskcontroller bevat 1k ram en tweemaal 8k rom. De eerste rom en de ram behoren tot de adresruimte van de NewBrain. De andere rom bevat het besturingsprogramma voor de z80, die de 765 floppydiskcontroller bestuurt. De z80, de 765, de tweede rom en de ram zijn op te vatten als een afzonderlijke computer, die de 1k ram gemeenschappelijk heeft met het operating system van de NewBrain, en die als taak heeft de in- en uitvoer naar de floppydrive te verzorgen.

96K NEWBRAIN

In de 96k is twaalfmaal 8k ram en zevenmaal 8k rom aanwezig (totaal 152k). Toch blijft de door z80 te adresseren geheugenruimte 64k. Om toch

het hele aanwezige geheugen te kunnen gebruiken wordt 'paging' toegepast. Hiervoor is de z80-adresruimte in acht gelijke stukken van 8k verdeeld, sloten genaamd. Deze sloten zijn genummerd van S0 tot en met S7. De afzonderlijke ram- en rom-chips (zowel in de NewBrain, als in de disk-controller en in de EIM) hebben een nummer gekregen, het zogenaamde paginanummer. Dit paginanummer wordt bepaald door de hardwarechakelingen in de NewBrain, de diskcontroller en de EIM. In een rom zit geen etiket met het paginanummer!

In tabel 1 staat, welke nummers de afzonderlijke pagina's hebben en in welk onderdeel ze zitten. Om nu ram of rom te kunnen gebruiken moet een rom of ram eerst in een bepaald slot gezet worden. Verder kan een routine in rom in het algemeen alleen correct werken, indien zij op dezelfde adressen staat als ze geschreven (geassembleerd) is. Pagina's met routines als device drivers moeten dus in een bepaald slot staan, willen ze correct werken. Gelukkig regelt het pos alles voor de gebruiker, zodat die zelf het verwisselen van de pagina's niet hoeft te programmeren. De routines daarvoor zijn in z80 geschreven, zodat alleen de z80-programmeur

tabel 1. rom- en ram-pagina's			
nr	plaats	type en inhoud	
60h	EIM	ram	
61h	EIM	ram	
62h	EIM	ram	
63h	EIM	ram	
64h	EIM	ram	
65h	EIM	ram	
66h	EIM	ram	
67h	EIM	ram	
68h	NB	ram	
6Bh	NB	ram	
70h	DC	rom devices 12 - 15, 32k-cp/m, 96k-cp/m, devices 18 en 19	
78h	NB	rom device 11, aanvullingen 32k operating system	
79h	NB	rom basic	
7Ah	NB	rom 32k os: start, overige devices, zcalls, mathpack	
7Bh	EIM	rom 96k pos: start + aanvullingen op bovenstaande 32k-rom's voor werking onder pos, zoals zcalls, basic, plot; menu	
7Ch	EIM	rom devices 1, 2, 5 - 10, 20 - 23, gewijzigde zcalls	
7Dh	EIM	rom devices 0, 3, 4, 11, 33, zcalls	

ermee kan 'stoeien'. In basic is paging ook beperkt mogelijk met

POKE Sx, pg

waarin Sx het slotnummer en pg het paginanummer is. Iets verfijnder is de CALL-routine PAGECALL.BAS die de basicprogrammeur dezelfde mogelijkheden als de z80-programmeur geeft.

START BIJ DE 32K

Om het gebruik van het geheugen enigszins te kunnen begrijpen, zal ik samenvatten, wat er na het aanzetten van de NewBrain gebeurt.

- ¶ De startprocedure begint op E000h.
- ¶ Er wordt een ram-test uitgevoerd om eventuele defecten te vinden en om te zien hoeveel ram aanwezig is (32k of 40k).
- ¶ De zeropage wordt gevuld.
- ¶ Er wordt gekeken, of er op adres A000h rom aanwezig is, en zo ja dan wordt daarnaartoe gesprongen.
- ¶ De startprocedure wordt voortgezet (zeropage gewijzigd).
- ¶ Er wordt gekeken, of er op 8000h rom aanwezig is, en zo ja dan wordt hiernaartoe gesprongen.
- ¶ Stel dat de diskcontroller aanwezig is. Dan wordt weer de zeropage aangepast.

Het operating system is nu geladen en er wordt gesprongen naar het gebruikersprogramma, in beide gevallen basic. De bekende mededeling komt op het scherm. Wordt tijdens het starten met een diskcontroller op de STOP-toets gedrukt, dan wordt er niet naar het gebruikersprogramma gesprongen, maar begint het laden van 32k-cp/m. Dit is géén afzonderlijk gebruikersprogramma, want voor de in- en uitvoer maakt het géén gebruik van de streamstructuur van het operating system van de NewBrain. Het is echter wel op te vatten als een z80-programma, dat op de 32k NewBrain werkt. Ook als cp/m vanuit basic met de opdracht CPM wordt gestart, wordt het z80-programma geladen en wordt het NewBrain operating system overschreven. Zie verder onder 32k-cp/m.

START BIJ DE 96K

Na het aanzetten start de NewBrain op dezelfde wijze als de 32k. Als de routine in A000h wordt uitgevoerd, dan worden de afzonderlijke pagina's in

de gewenste sloten gezet en springt het programma naar slot 4 (8001h), waar zich rom-pagina 7Bh bevindt. Het programma van deze pagina wordt uitgevoerd. Hoe dit door de hardware geregeld wordt, gaat boven mijn pet, maar voor het vervolg doet dit niet ter zake. Achtereenvolgens wordt

- 1 paging aangezet;
- 1 getest hoeveel videopagina's er zijn;
- 1 de zeropage (ram-pagina 60h) in slot 0 gezet en ingericht, beginnend vanaf adres 0000h. Hierbij worden alle variabelen en tabellen opgesteld en eventueel gevuld. Ook wordt vanaf 1702h een aantal paging routines ingevuld.
- 1 Vervolgens worden alle rom's in de EIM en de diskcontrollor afgewerkt, waarbij met behulp van een index wordt bepaald, of een routine op de desbetreffende pagina:
 - 1 direct moet worden uitgevoerd, of
 - 2 een zcall-routine is, of
 - 3 een device driver is, of
 - 4 de rest van de pagina en een aantal volgende pagina's moet worden overgeslagen;
 - 5 een gebruikersprogramma is. Eventuele zeropage-parameters en -routines worden ingevuld.
- 1 Het pos is nu geladen. Op dit punt komen steeds alle gebruikersprogramma's terug.
- 1 De routine EXIT wordt aangeroepen, waarbij alle streams worden gesloten en overbodige parameters en parameterstructuren worden schoongemaakt.
- 1 Een menuprogramma wordt gestart en er komt eindelijk iets op het scherm. Alle in rom beschikbare gebruikersprogramma's komen op het scherm en de gebruiker kan kiezen. Als gebruikersprogramma zijn beschikbaar basic en cp/m. Het pos blijft actief.

Welk geheugen is er nu beschikbaar? Met uitzondering van pagina 60h zijn alle ram-pagina's beschikbaar voor het pos en de gebruikersprogramma's. Pagina 60h is de pos-zeropage. In tegenstelling tot de 32k-zeropage bevat deze niet alleen parameters, maar ook systeemtabellen, zoals de stream- en devicetable. Ook staat er een aantal pos-routines voor het pagen, die bij het starten vanuit rom in ram gekopieerd werden. Aan het einde is van 1907h tot 1FFFh een stuk vrij, waarin eigen z80-routines geplaatst kunnen worden.

De pagina's 61h - 6Bh kunnen worden gebruikt voor eigen geheugenruimte, buffers en gebruikersprogramma's. Pagina's 60h - 67h (EIM-ram) zijn

hardwarematig niet geschikt als videogeheugen, zodat de eigen geheugenruimte van de videodevices zich altijd op de pagina's 68h - 6Bh (NewBrain-ram) moet bevinden.

Heeft het gebruikersprogramma routines van het pos nodig, dan worden eerst de paginanummers van het gebruikersprogramma opgeslagen, waarna de pagina's met de benodigde routines aangeroepen worden en in het slot gezet worden, waarin ze werkzaam zijn. Na afloop van een pos-routine worden de pagina's van het gebruikersprogramma weer teruggezet. Op de zeropage bevindt zich een aantal systeemroutines, die dit verwisselen van de pagina's regelen. Voor iedere routine ligt vast, welk slot gebruikt wordt. Zo worden zcalls uitgevoerd vanuit slot 7, en de device drivers komen in slot 4, ongeacht op welke pagina de routine staat.

In tegenstelling tot de 32k gebruikt pos geen ram behorend tot de z80-adressen van het gebruikersprogramma maar ram behorend tot andere pagina's. Onder basic is bijvoorbeeld 2000h - BFFFh, 40k geheugen, beschikbaar voor programma, variabelen en de eigen variabelen van basic. De rom-pagina's 78h - 7Ah uit de NewBrain en 70h uit de diskcontrollor worden ook onder het pos gebruikt. Voor degene die alleen onder basic werkt, zal bovenstaande informatie weinig praktische waarde hebben.

HET 96K GEHEUGEN ONDER BASIC

Weer kijken we, wat er gebeurt als basic gekozen is. Pagina 60h bevindt zich in slot 0 als zeropage, blijft daar en is de schakel met pos.

- 1 pagina 7Bh wordt in slot 7 gezet en er wordt naar het basic entry point gesprongen;
- 1 de zeropage wordt aangepast;
- 1 pagina 7Bh (de CD-rom van de 32k) wordt in slot 6 gezet;
- 1 er wordt gekeken of er vijf ram-pagina's beschikbaar zijn, en deze worden in de volgorde 61h - 67h in slot 1 tot en met slot 5 gezet;
- 1 er wordt naar de CD-rom gesprongen op dezelfde wijze als in de 32k.

De verschillen tussen 32k- en 96k-basic in verband met basicprogramma's voor de basicgebruiker zijn al uitgebreid beschreven in On-line 5, pag. 33.

CP/M

Voordat we gaan zien, wat er gebeurt met de geheugenindeling onder cp/m, even een kleine zijsprong. Iedereen die enigszins thuis is in cp/m,

weet hoe de z80-adresruimte door cp/m gebruikt wordt: 0 - FFh = (cp/m-)zeropage, 100h tot een niet-vaststaand adres = tpa, gevolgd door ccp, bdos en bios. Verder kent cp/m als logische in- en uitvoerdevices (in Nederlandse cp/m-boeken ook wel apparaten genoemd):

- 1 console CON: = combinatie toetsenbord en scherm
- 2 ponsbanduitvoer PUN: = seriële uitvoer
- 3 ponsbandinvoer RDR: = seriële invoer
- 4 printer LST: = printer

Ook moet cp/m naar disk kunnen in- en uitvoeren.

Tussen de in- en uitvoer naar disk en de overige in- en uitvoer is er in de NewBrain een zeer essentieel verschil.

In- en uitvoer vindt in de NewBrain plaats via streams en devices. Echter, de in- en uitvoer naar disk vindt min of meer plaats zoals in het 'standaard bios' van Digital Research is beschreven. Onder cp/m gaat de in- en uitvoer naar disk via een aanpassing van dit 'standaard bios'. Wordt onder basic van disk gebruik gemaakt, dan gebeurt dit door middel van de devices 12 - 15. Deze devices bestaan uit een normaal NewBrain-device. Dit device bevat bijvoorbeeld het file control block en een sector-buffer. Voor de eigenlijke in- en uitvoer worden echter disk-in- en -uitvoeroutines uit cp/m (bdos en bios) gebruikt. Hiervoor bevat de diskcontroller-rom een gestripte versie van het normale cp/m-bios. De ccp en het bdos worden echter geheel vanaf de schijf geladen, wat nodeloos veel geheugen kost op de 32k. Gelukkig worden ze maar een keer geladen, ongeacht hoeveel streams er naar deze devices zijn geopend.

32K-CP/M

Zoals boven al is uiteengezet, is het operating system niet actief en wordt alle in- en uitvoer door zelfstandige routines in het bios uitgevoerd. Dit bios bevindt zich in de diskcontroller-rom. Cp/m-zeropage (0000h), tpa (0100h), ccp (5A00h) en bdos (6200h) staan aaneengesloten in de ram. Het videogeheugen moet echter ook in ram staan en mag niet in de zeropage, tpa, ccp of bdos of tussen deze onderdelen staan. Dit is bij de 32k opgelost door achter het bdos direct de sprongtabel van het bios (7000h) te zetten. Hierna volgen de onderdelen van het bios die in ram moeten staan (703Ah), waaronder het videogeheugen (7380h). Alleen de vaste routines van het bios staan in rom (8F03h). Onderdelen van het bios in ram, die ook zichtbaar moeten zijn voor de z80 in de diskcontroller, moeten op een adres groter dan 8000h staan. Hiervoor wordt de ram van

9C00h tot 9FFFh gebruikt. Voor een volledige indeling zie tabel 2. Als gevolg hiervan is er enige ongebruikte ram onder 32k-cp/m: 723Ah - 737Fh en 9F7Dh - 9FCDh. Ook kan onder cp/m rechtstreeks naar het scherm worden geschreven, omdat dat zich op een vaste plaats bevindt. De z80 krijgt ineens memory-mapped I/O.

96K-CP/M

Op de 96k is de totale 64k als ram voor deze onderdelen beschikbaar. Hoewel cp/m een operating system wordt genoemd, gebruikt de 96k als operating system nog steeds het pos en is cp/m slechts een gebruikersprogramma onder dit operating system. Het pos creëert slechts een toestand in de 96k, die door de gebruiker kan worden benut om programma's te RUNnen, die aan bepaalde regels voldoen. Ook hier zal ik beschrijven, wat er gebeurt na de keuze cp/m vanaf het openingsmenu.

Bij het starten van het gebruikersprogramma cp/m gebeurt het volgende.

- 1 Pagina 70h met het cp/m-gebruikersprogramma wordt in slot 1 gezet en het programma vervolgt in slot 1.

tabel 2. geheugenindeling 32k-basic en 32k-cp/m					
32k-basic			32k-cp/m		
0000	ram 6b		0000	ram 6b	0000 ram
2000	ram 6a		2000	ram 6a	0100 tpa
4000	ram 69		4000	ram 69	tpa + rst
6000	ram 68		6000	ram 68	5a00 ccp
8000			8000	ram 68	6200 bdos
	rom/ram:			rom/ram	7000 bios
	rombox			70/dram	(jptab)
a000	rom 7a		a000	rom 7a	703a dpbase
c000	rom 79		c000	rom 79	723a - 737f vrij
e000	rom 78		e000	rom 78	7380 tvram
ffff			ffff		8000 rom (bios)
					9c00 ram
					9f7d - 9fcd vrij
					a000 rom
					ffff

tabel 3. geheugenindeling 96k-basic en 96k-cp/m

	96k-basic	ram eim	rom eim
s0 8b 0000	ram 60	67	7c
s1 8d 2000	ram 61	66	7b
s2 8f 4000	ram 62	ram nb	rom dc
s3 91 6000	ram 63	6b	70
s4 93 8000	ram 64	6a	rom nb
s5 95 a000	ram 65	69	7a
s6 97 c000	rom 7d	68	79
s7 99 e000	rom 78	dram?	
ffff			

	96k-cp/m	ram eim	rom eim	
s0 8b 0000	ram 61	60 sys0	7d	0000 cpm0
s1 8d 2000	ram 68	ram nb	7c	0100 tpa
s2 8f 4000	ram 67	6b	7b	
s3 91 6000	ram 66	6a	rom nb	
s4 93 8000	ram 65	69	7a	
s5 95 a000	ram 64	rom dc	79	de00 ccp
s6 97 c000	ram 63	70	78	e600 bdos
s7 99 e000	ram 62	dram		f400 bios
ffff				f99e - facd (vrij)
				fd4e - ff4d (vrij)
				ff4e - ffcc vrij
				ffff

vrije ruimte: sys0 32k: 79 - ff; sys0 96k: 1907 - 1fff

- † Pos kijkt, of er acht ram-pagina's beschikbaar zijn, en zet de bijbehorende in slot 2 tot en met 7.
- † Het bios wordt vanuit de rom-pagina vanaf F400h in slot 7 gezet.
- † Het bios voor de 32k en 96k zijn hetzelfde, alleen de adressen zijn verschillend. Deze adressen moeten voor de 96k worden aangepast (RELOCATE), wat gebeurt aan de hand van een bit-map in de rom.
- † Enkele parameters worden ingevuld.
- † Vanaf drive 0 wordt de eerste sector gelezen (we horen wat). In deze sector staat de informatie, die wordt samengesteld met het programma CONFIGUR.COM.
- † Aan de hand van deze informatie wordt voor iedere gedefinieerde disk drive het Disk Parameter Block ingevuld.
- † Nu worden voor de bovengenoemde in- en uitvoer de benodigde streams geopend:

#1 = CON:(in) = device 19
 #2 = CON:(uit) = device 18
 #3 = LST: = device 16 of 21 (of 8 of 9)
 #4 = PUN: + RDR: = device 17
 #5 = onbenoemd = device 3

- † De copyrightmededeling wordt op het scherm gezet (we zien weer wat).
- † De CPM0-pagina wordt in slot 0 gezet en de routine COLD BOOT wordt gestart.
- † Deze zet de ram-pagina in slot 2, zodat de hele z80-map nu met ram is gevuld, en vult het adres voor de cp/m-interruptroutine in.
- † Tenslotte wordt de routine WARM BOOT gestart, die bdos en ccp van schijf leest en REALLOCATEt, waarna de besturing aan de ccp wordt overgedragen en de prompt A> op het beeldscherm verschijnt. De gebruiker mag nu met cp/m-opdrachten beginnen.

Welke situatie treffen we nu aan? Het pos is actief, maar neemt geen plaats in de z80-map in. Alle in- en uitvoer (met uitzondering van disks) vindt plaats via de geopende streams. De zeropage is aanwezig en er zijn vijf streams geopend, ieder met zijn eigen geheugenruimte en eventuele buffer. Er zijn in de z80-map acht ram-pagina's voor de gebruiker zichtbaar zonder paging, namelijk de cp/m-pagina's 61h en 68h - 62h (in de volgorde slot 0 - slot 7).

Als we uitgaan van ongebufferde devices voor stream 2 en 3, een default buffer van 256 bytes voor stream 1 en een scherm van 25 regels, dan zijn elders als pos-pagina's aanwezig, en zonder paging onzichtbaar voor de gebruiker:

- † ram-pagina 60h met de zeropage
- † ram-pagina 69h voor de buffer van stream 1
- † ram-pagina 6Ah voor de eigen geheugenruimte van de gebruikte streams (region 8 van het MMS)
- † de rom-pagina's voor de in- en uitvoer naar de streams. Zelfs de rom-pagina met basic kan geactiveerd worden onder cp/m met BA-SIC.COM (deel N311 in de softwarebibliotheek)

De bovengenoemde devices voor de verschillende streams zijn die zoals Grundy ze gekozen heeft. De gebruiker mag zelf andere devices kiezen, mits ze maar geschikt zijn voor het desbetreffende cp/m-device. Nostalgie: open stream 1 op device 8 en stream 2 op device 5, en je hebt de teletype-terminal terug; werkelijk het werkt!

Indien nodig kunnen alle rom-pagina's zichtbaar gemaakt worden, met uitzondering van pagina 70h (diskcontroller). Waarom dit niet mogelijk is, blijft voor mij onduidelijk. Vermoedelijk heeft dit met de hardware te maken.

Als we de ram-pagina's nader bekijken, dan vinden we een aantal ongebruikte stukken geheugen, zowel in de cp/m-pagina's als in de pos-pagina's. Deze lege ruimten kunnen worden gebruikt om wijzigingen van de bios-routines aan te brengen. Een goed voorbeeld hiervan is FAST.COM van Henk Blik, dat uitsluitend geheugen gebruikt dat onvoorwaardelijk vrij is. De combinatie van softkeys en FAST van Gordon Crosse is minder fraai, omdat het ook geheugen gebruikt, dat slechts vrij is, als er een bepaalde configuratie gebruikt wordt. Altijd ongebruikt in het standaard-bios zijn op de cp/m-pagina's de adressen FF4Eh - FFCCh en in de pos-pagina's op de zeropage 1907h - 1FFFh.

Z80-ROUTINES IN DE 96K

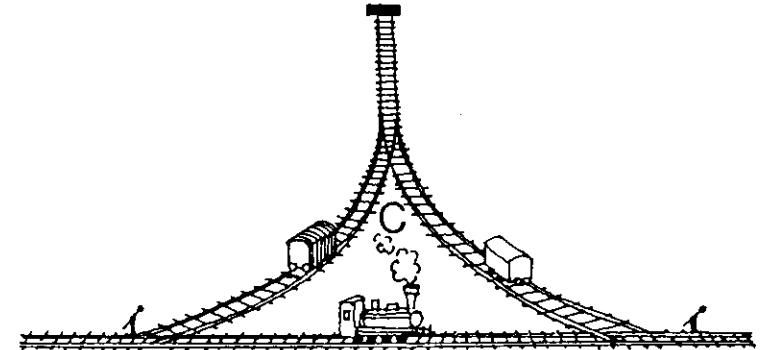
Zoals al eerder door mij is opgemerkt, is er in theorie geen verschil tussen de 32k en 96k in de toepassing van een z80-routine in een RESERVEde gedeelte van het geheugen. De routine moet dan op dezelfde adressen staan als in de 32k en moet absoluut geladen worden. Dat wil zeggen, dat er een stuk geheugen moet worden geRESERVEerd met RESERVE TOP - startadres, en dat de routine gePOKEt worden vanaf TOP of vanaf het startadres. RESERVE xxxx, waarin xxxx de waarde is van TOP - startadres in een 32k is niet toegestaan.

Bijvoorbeeld: RESERVE 2048 geeft in de 32k een TOP van 30720 (bij de meeste 32k's echter 30718). Bij de 96k wordt de TOP: 46912. RESERVE TOP - 30720 geeft bij alle NewBrains een TOP van 30720.

Zoals ik al eerder heb opgemerkt, wordt het gebruik van slot 4 voor z80-routines afgeraden. Ondanks naspeuringen ben ik er nog steeds niet achter, waarom z80-routines niet werken in slot 4.

HET GEBRUIK VAN PAGINA'S EN SLOTEN DOOR HET POS

- † De zeropage (ram-pagina 60h) blijft onder het gebruikersprogramma basic altijd aanwezig in slot 0. Zelf geschreven z80-routines voor het switchen van pagina's dienen bij voorkeur op de zeropage te staan.
- † Een zcall gebruikt altijd slot 7. De in- en uitvoerroutines gebruiken ook slot 4, wanneer ze de device drivers aanroepen.
- † Een device driver is werkzaam in slot 4. Een zelf geschreven device driver moet dus in code voor slot 4 worden geschreven. Het is ook mogelijk om een device driver in een ander slot te schrijven (zie bijvoorbeeld KLOK, On-line 8), maar dit kan moeilijkheden geven.
- † Vanuit slot 4 kan een device driver ook de pagina's in andere sloten veranderen. Let op, dat de originele pagina's weer worden teruggezet.
- † Wanneer de videopagina's worden aangesproken, dan worden ze, afhankelijk van het aantal, vanaf slot 4 tot en met 7 gezet. Om dit mogelijk te maken, verplaatsen de device drivers voor device 0, 3 of 4 zich van slot 4 naar 3.
- † GETOBJECT zet het eigen geheugen van de stream in slot 1 en 2.
- † De buffer van gebufferde devices staat in slot 6.



HET GEBRUIK VAN ZCALLS ONDER 32K CP/M

Cp/m heeft een eigen indeling van de zeropage. Details staan in ieder boek over cp/m. Het bios van de 32k NewBrain gebruikt ook nog enkele lokaties in de zeropage.

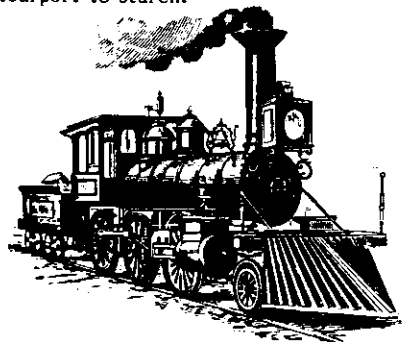
In verband met het gebuifderde toetsenbord heeft 32k-cp/m een eigen interrupt. Deze maakt geen gebruik van de zeropage-variabelen van de normale interrupt, maar plaatst een aantal nieuwe variabelen in de disk-controller-ram. In de interrupt is niet voorzien in het gebruik van de devices 1 en 2, noch in de functie van de STOP-toets.

Het 'anker' van de streams (STRTOP) wordt door cp/m overschreven. Het gevolg is, dat de streamstructuur van het ios niet meer bruikbaar is. Dit maakt het vrijwel onmogelijk om andere devices dan de gebruikelijke (CON, LST, RDR en PUN) toe te passen. Wil men ze toch toepassen, dan kunnen ze niet met zcalls worden aangeroepen, omdat deze van de streams gebruik maken.

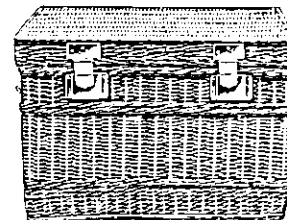
Onbruikbaar zijn dus alle zcalls, die een stream nodig hebben en die gebruik maken van de normale interrupt. Dit zijn alle zcalls boven 2Fh met uitzondering van KLOOK. De overige zcalls (mathpack-routines) kunnen wel gebruikt worden, mits het volgende in acht wordt genomen: het geheugen van 100h - 19Fh is het werkgebied van deze routines. Deze ruimte moet door het cp/m-programma worden vrijgehouden. Dit kan door een programma te beginnen met JP 01A0h en de overige code te assembleren vanaf dit adres.

Ik hoop, dat ik met dit artikel opnieuw wat heb kunnen toevoegen aan de kennis van het operating system van de NewBrain, speciaal dat van de 96k. Naast de door Grundy en Tradecom verstrekte documentatie is de source code onmisbaar om alle details van de NewBrain te kunnen begrijpen. Voor geïnteresseerde gebruikers heb ik de source code in geassembleerde vorm van de diskcontroller en de expansion beschikbaar (.PRN-files) voorzien van de actieve adressen. Ze staan in geSQUeesde vorm op een 800k-diskette. Te bekomen door mij een geformatteerde diskette voorzien van deugdelijke verpakking en retourport te sturen.

Wim Luijt



rombox



Voor de NewBrain zijn via de gebruikersgroep enige tijd originele GBS-romboxjes versie 3 verkocht tegen sterk gereduceerde prijs. De boxjes waren bedoeld voor een residente tekstverwerker met de naam Letterwriter en zijn zodanig bedraad, dat de NewBrain na een koude start opstart in de (ep)rom in socket 4. Van de betreffende tekstverwerker is niet bekend, of deze ooit in Nederland in de handel is gebracht, en de betreffende rom ontbreekt in alle verkochte exemplaren.

De lege rombox is echter vrijwel indentiek aan versie 1, welke beschreven wordt in de Rombox Instructions in de service manual. Een uitzondering is echter de aanwezigheid van IC 409, dat zorgt voor een extra buffering van de databus en zodanig ingericht is, dat er wel data vanuit de rombox kunnen worden gelezen, maar er geen kunnen worden geschreven. Voor het gebruik als rombox is dit geen bezwaar, maar voor andere toepassingen kan dit een beperking betekenen, zoals we later zullen zien.

Waarvoor kan de rombox in de huidige staat worden toegepast?

- Upgrades: voor het wijzigen van de geïnstalleerde rom op de processorprint zonder de bestaande roms uit te solderen. Zo kan het operating system vervangen worden door een latere versie. Ook zou bijvoorbeeld de graphics-rom (EF) of de IO-rom (AB) vervangen kunnen worden door een meer recente uitgave. Door het ontbreken van nieuwe uitgaven komt deze toepassing voor de meeste gebruikers niet in aanmerking, of men moet zelf modificaties in de systeemsoftware willen en kunnen aanbrengen.
- Extra talen: de CD-rom, waarin de basic interpreter zich bevindt, kan worden vervangen door een met een andere programmeertaal. MPE

Forth bijvoorbeeld, welke 2 romslots in beslag neemt, komt dan in slot 89 en CD en vervangt basic.

- c. Auto run programs: dit zijn applicatieprogramma's in (ep)rom en ze kunnen zowel uit z80-machincode, als getokende basic bestaan. De tekstverwerker Letterwriter zou een voorbeeld kunnen zijn van een applicatieprogramma in z80-machinecode. Basicprogramma's kunnen niet zonder meer in eeprom worden gezet. Voor meer informatie hierover kunt u zich wenden tot ondergetekende.

Voor alle romslots op de box geldt, dat ze in elk 8k-blok in het geheugen kunnen worden geplaatst door het aanbrengen van de juiste verbindingen. In page 0 dient echter altijd ram-geheugen aanwezig te zijn, omdat de systeemvariabelen en de stack zich hier bevinden. De juiste verbindingen vindt u in de bijgevoegde tabel, welke is overgenomen uit de Rombox Instructions.

RAM OP DE ROMBOX

Een zinvolle toepassing van de rombox is om het standaard ram-geheugen van de unexpanded NewBrain uit te breiden tot 40k. Het geheugenbereik 8000h tot 9FFFh is in de standaardversie ongebruikt gelaten, en het is aansluitend op de bestaande ram van 0000h tot 7FFFh uitstekend geschikt voor dit doel. Technisch is het echter niet zonder meer mogelijk om een (ep)rom te vervangen door een (statische) ram om de volgende redenen:

1. De databuffer (IC 409) kan slechts data doorgeven in een richting, namelijk van de rombox naar de NewBrain.
2. Pen 27 (NotWrite in geval van ram) is niet verbonden met de z80 maar ligt aan de +5 V.

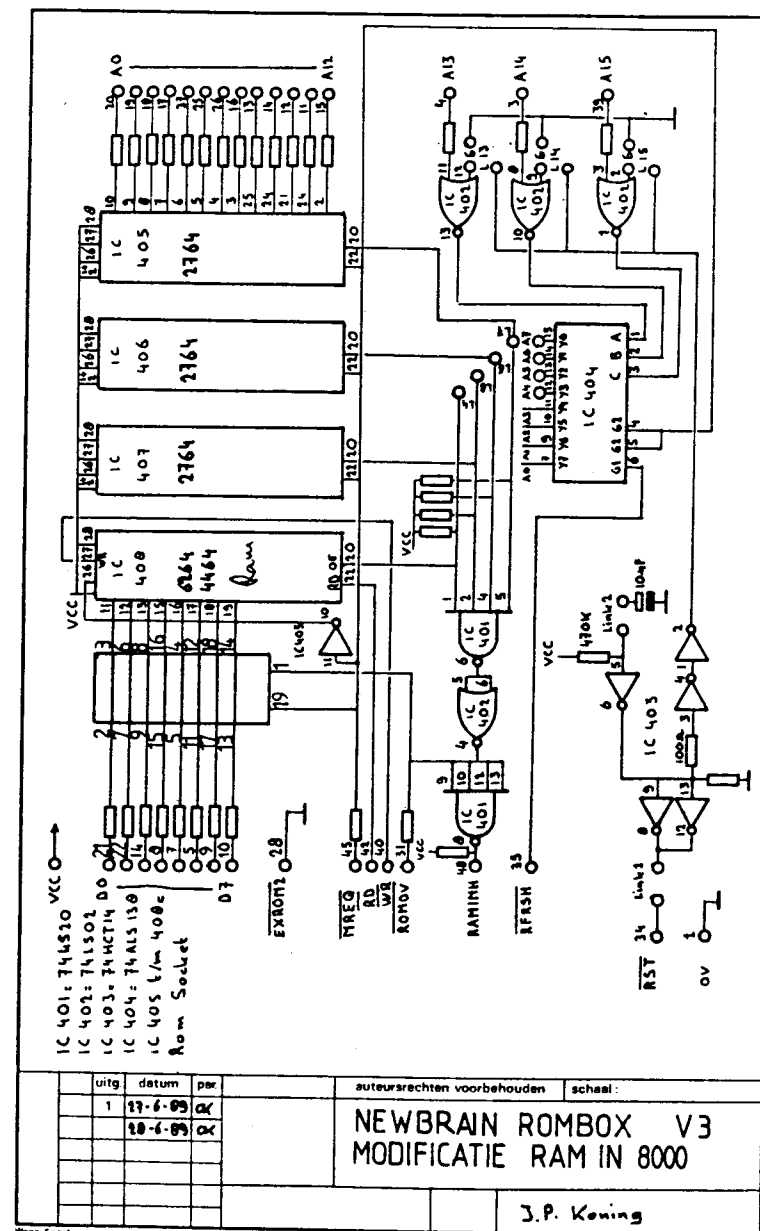
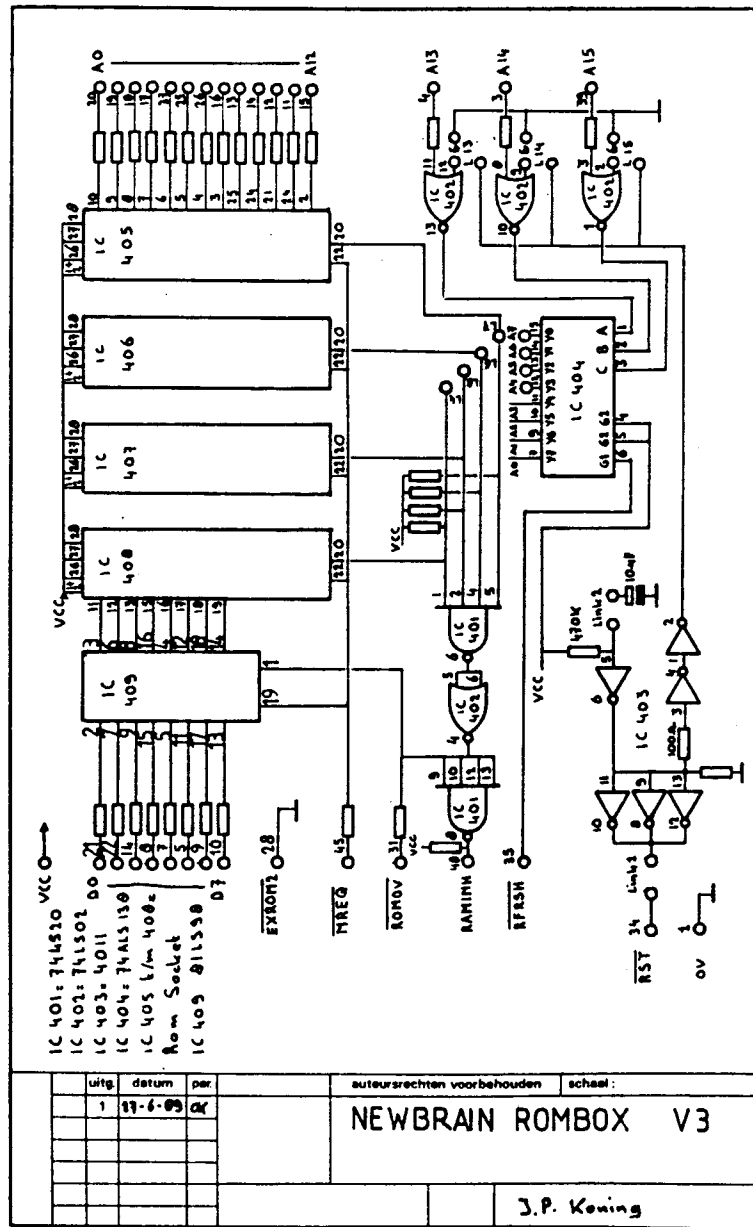
Om nu toch een ram te kunnen aansluiten is een aantal modificaties van het printplaatje noodzakelijk. Voor een juist begrip wil ik de aangebrachte modificaties aan de hand van het bijgevoegde schema trachten uit te leggen.

De databuffer IC 409 is niet bidirectioneel en moet worden verwijderd. De datalijnen worden nu rechtsreeks doorverbonden door middel van draadbrugjes op de printplaat. Hierdoor is tweeweg dataverkeer mogelijk en de oplossing geeft, zeker wanneer CMOS-eproms en -ram wordt gebruikt, nauwelijks extra belasting op de datalijnen. In versie I van de GBS-rombox was dit IC niet aanwezig en het is mogelijk dat hiervan ook exemplaren bij de gebruikers in omloop zijn, even nakijken dus!

Een tweede belangrijke functie van IC 409 is het doorschakelen van de datalijnen bij een NotMREQ, ofwel het blokkeren van dataoverdracht, wanneer geen geheugenoperatie wordt gevraagd door de z80-processor. Deze functie kan worden overgenomen door de adresselector IC 404. Dit IC (een 74ALS138) heeft namelijk nog twee ongebruikte selectingangen op de pennen 4 en 5 die aan de 0 V liggen. Door deze twee ingangen los te maken van pen 8 en te verbinden met NotMREQ (pen 45 op de expansieconnector) bereiken we, dat de rombox alleen tijdens een geheugenoperatie wordt geselecteerd. Om nu ook daadwerkelijk naar de ram te kunnen schrijven dient het NotWR-sigitaal van de processor via pen 40 op de expansionconnector te worden verbonden met pen 27 van de socket. Bij een 2764 eeprom heeft deze pen een functie bij het programmeren, maar hier op de print is hij verbonden met de +5 V. De verbindingen tussen de pennen 26, 27 en 28 dienen dan ook op de print te worden verwijderd, en de vrijkomende pen 27 moet worden verbonden met pen 40 op de expansieconnector. De nu eveneens vrijgekomen pen 26 heeft bij een CMOS-ram de functie van enable-ingang, en kan weer verbonden worden met +5 V. Beter is het echter om deze pen te verbinden met een MREQ-sigitaal. Omdat een dergelijk sigitaal op de print alleen als NotMREQ aanwezig is, dient dit eerst te worden geïnverteerd. Dat kan door hiervoor een poort te 'lenen' van IC 403. Dit CMOS IC heeft drie poorten parallel aangesloten om voldoende drivecapaciteit te krijgen. Door nu dit IC te vervangen door een pincompatibele 74HCT14 is dit parallelschakelen niet meer nodig en kan poort 5 (pen 10 en 11) worden geïsoleerd en gebruikt voor het inverteren van NotMREQ. Tevens wordt hiermee het signaalniveau van NotRST verbeterd. Na deze wijziging is het mogelijk geworden data naar de rombox te schrijven. Om nu ook de data weer terug te kunnen lezen is een gewijzigde timing van het output-controlsigitaal op pen 22 gewenst. Deze pen wordt nu gestuurd door NotMREQ, die bij elke geheugenoperatie laag wordt, maar alleen laag mag worden gedurende een leescyclus. Daarom dient deze pen los te worden gemaakt van NotMREQ en te worden verbonden met NotRD op pen 42 van de EC.



Tot zover de functionele uitleg van de modificatie. Wanneer we ervan uitgaan, dat in IC 408 een 8k statische ram komt en de overige sockets alleen geschikt blijven voor eeprom, komt dit in de praktijk hierop neer:



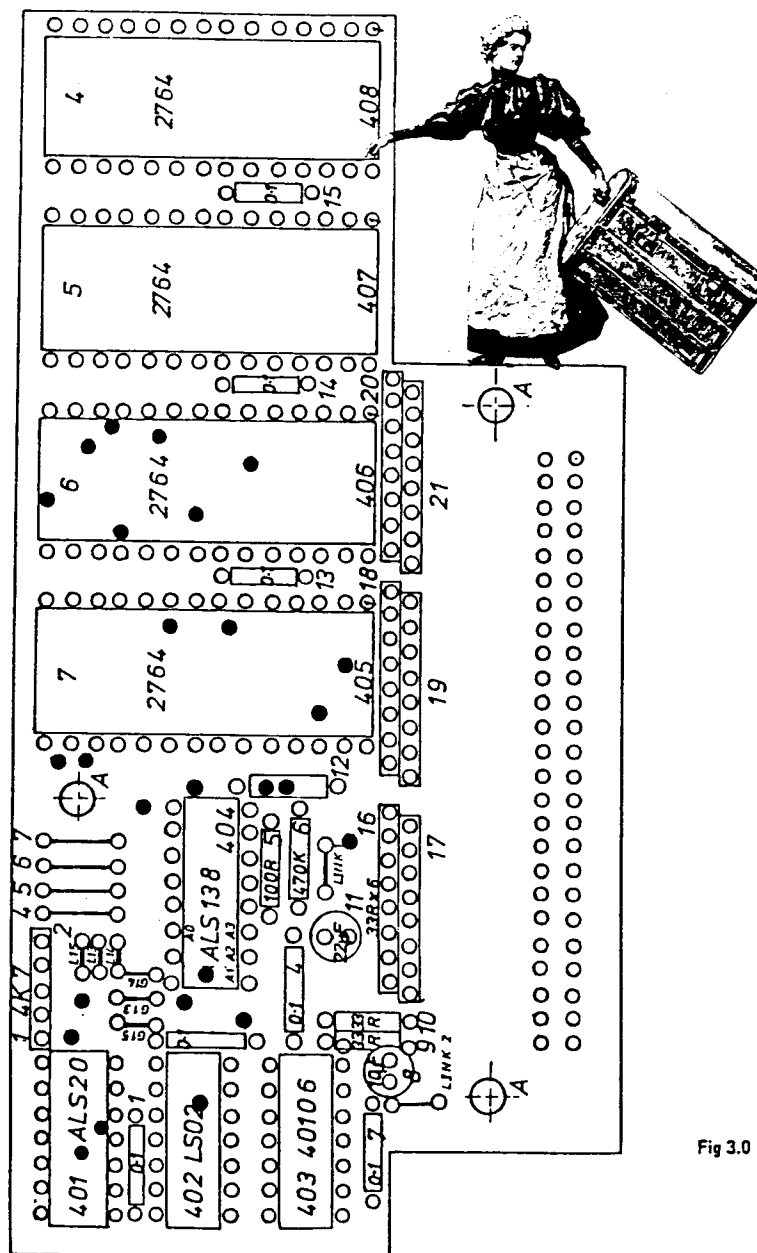


Fig 3.0

ROM BOX INSTRUCTIONS

FOUR WAY STANDARD CONNECTIONS. (Fig 3.0)

This unit is capable of placing four 2764 EPROMs anywhere within the 64K address range of the Z80. Chip 404 (74LS138) divides the top three address lines into individual chip select lines each of 8K bytes width. Links L4 through to L7 connect to the chip select on the four sockets. With straight linking on the board

EPROM socket 7 will be selected in the address range E000 HEX to FFFF HEX.

EPROM socket 6 will be selected in the address range C000 HEX to OFFF HEX.

EPROM socket 5 will be selected in the address range A000 HEX to BFFF HEX.

EPROM socket 4 will be selected in the address range 8000 HEX to 9FFF HEX.

To achieve any other 8K byte address range it will be necessary to either strap across the links or (if address below 8000H is wanted) link directly to the LS138 chip. (See Fig 1.0)

Another feature of this board is the ability to 'power up' into an EPROM on this board. Normally the machine 'powers up' at E000H internally, but by linking L1 and L2 the power up condition is lengthened and moved to a device in the ROM BOX. Links L13 to L15 and G13 to G15 determine which device is selected in power up. Strapping L13 to L15 will make the machine power up at EF. For other strapping option see Fig 2.0

S	E/F	C/D	A/B	8/9	0/7	4/5	2/3	0/1
O	7 L7-A7	L7-A6	L7-A5	L7-A4	L7-A3	L7-A2	L7-A1	Not Recommended
C	6 L6-A7	L6-A6	L6-A5	L6-A4	L6-A3	L6-A2	L6-A1	
K	5 L5-A7	L5-A6	L5-A5	L5-A4	L5-A3	L5-A2	L5-A1	
E	4 L4-A7	L4-A6	L4-A5	L4-A4	L4-A3	L4-A2	L4-A1	
T								

Fig 1.0

* A3-A1 are pins on chip 404

	ADDR		STRAP		
E/F	L1	L2	L13	L14	L15
C/D	L1	L2	G13	L14	L15
A/B	L1	L2	L13	G14	L15
8/9	L1	L2	G13	G14	L15
6/7	L1	L2	L13	L14	G15
4/5	L1	L2	G13	L14	G15
2/3	L1	L2	L13	G14	G15

Fig 2.0.

voeding

1. Verwijder socket 408 voorzichtig van de print. Dit kan met desoldeerlitze of met een goede tinzuiger.
2. Verwijder de verbindingen tussen pen 26, 27 en 28 op de componentzijde van de print met een fijn vijltje, mesje of iets dergelijks.
3. Soldeer de socket weer op zijn plaats.
4. Maak een verbinding tussen pen 28 van IC 408 en pen 1 van IC 407.
5. Isoleer pen 22 van IC 408 van NotMREQ.
6. Isoleer pen 4 en 5 van IC 404 van pen 8 van IC 404.
7. Maak een verbinding tussen pen 4 en 5 van IC 404 en pen 45 van de expansionconnector.
8. Maak een verbinding tussen pen 22 van IC 408 en pen 42 van de expansionconnector.
9. Maak een verbinding tussen pen 27 van IC 408 en pen 40 van de expansionconnector.
10. Verwijder IC 403.
11. Isoleer pen 10 en 11 van IC 403.
12. Soldeer 74HCT14 op de plaats van IC 403.
13. Maak een verbinding tussen pen 26 van IC 408 en pen 10 van IC 403.
14. Verwijder IC 409.
15. Maak een verbinding tussen pen 11 van IC 403 en pen 19 van IC 409.
16. Maak doorverbindingen tussen de volgende pennen: 2 en 3, 4 en 5, 6 en 7, 8 en 9, 11 en 12, 13 en 14, 15 en 16, 17 en 18.
17. Plaats een 8k statische CMOS-ram in de socket van 408. Dit kan een D4464 (NEC), 16264 (Intel) of compatibel type zijn.

Wanneer we de NewBrain na de uitgevoerde modificatie weer aanzetten, duurt het iets langer, voordat de bekende melding NewBrain basic ready op het scherm verschijnt; er valt namelijk meer geheugen te testen en dat kost wat meer tijd. Om te controleren, of we nu inderdaad meer ram tot onze beschikking hebben, typen we PRINT PEEK (7). We krijgen nu de TOP van de ram en, als het goed is, krijgen we nu de waarde 160 retour, dit is $160 \times 256 = 40960$ bytes.

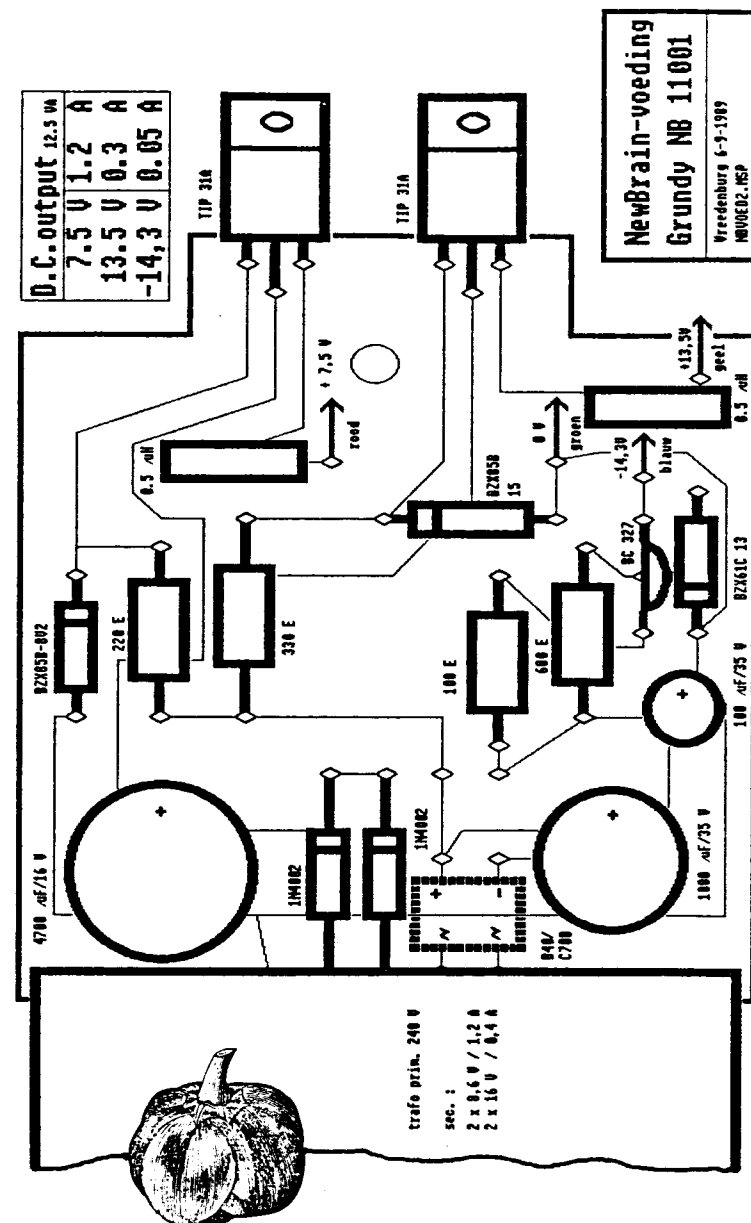
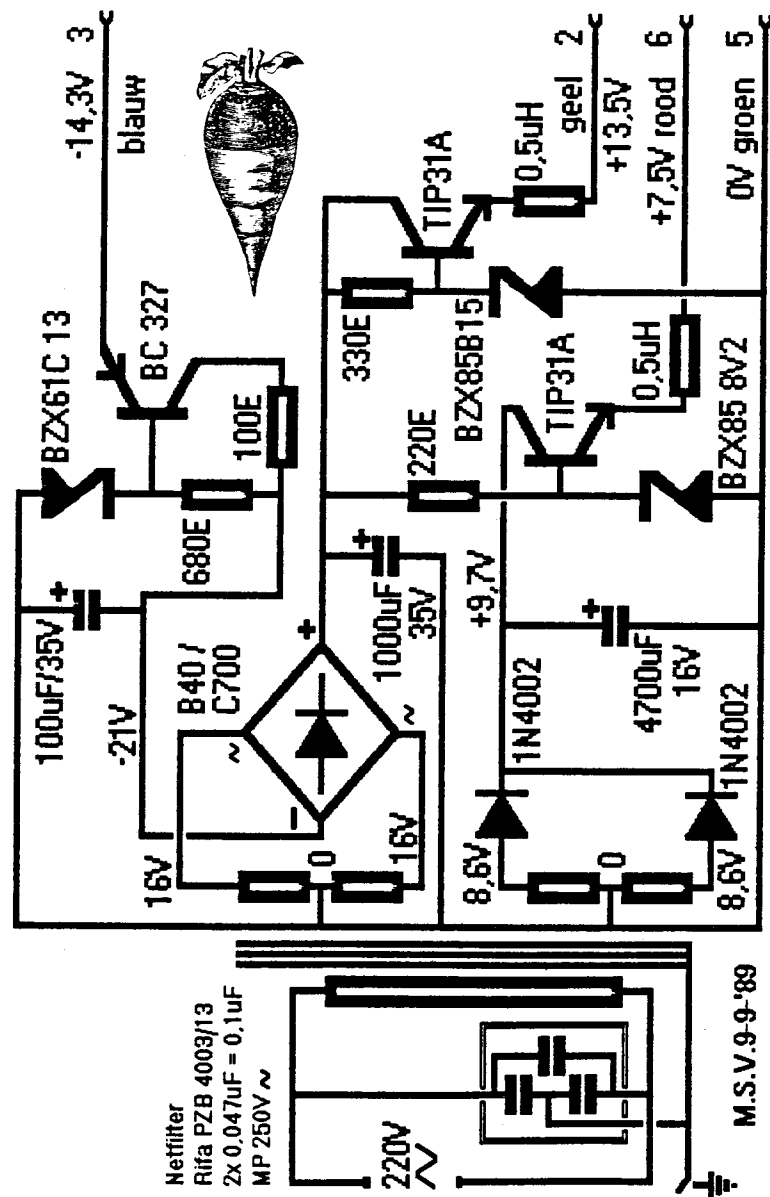
J. P. Koning

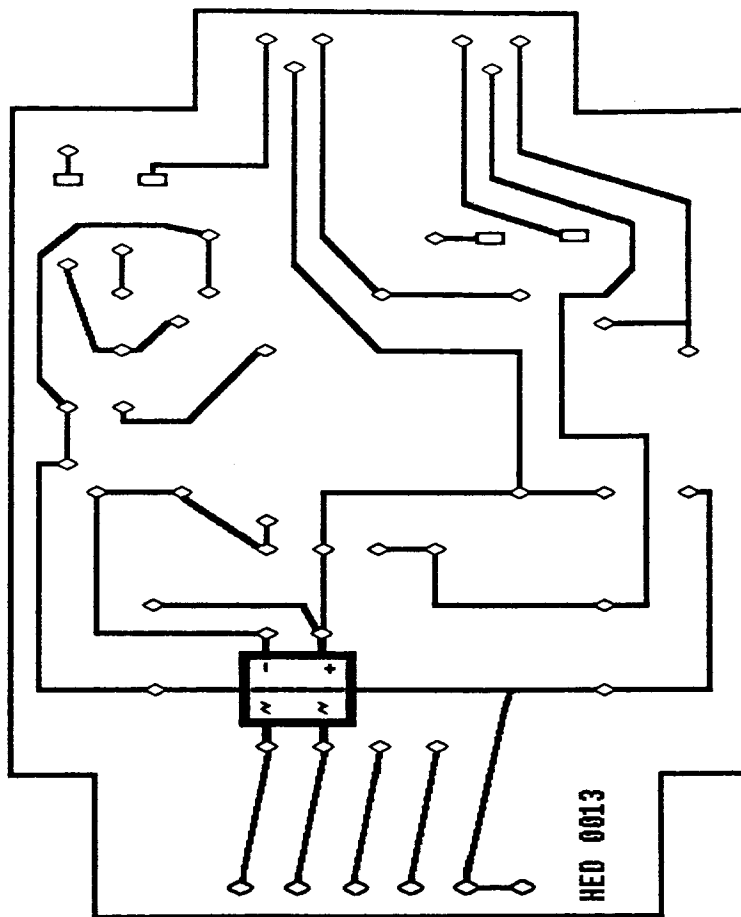


laat de newbrain in het algemeen slecht gedocumenteerd zijn, van de voeding die grundy erbij leverde (typenummer NB 11001), is helemaal niets bekend

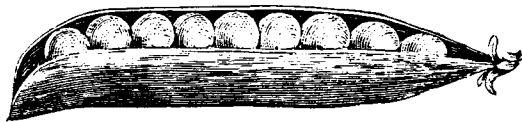
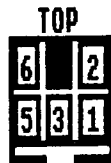
m s vreedenburg heeft zo'n bruin blokje opengemaakt en is met zijn tekenpakket gaan stoeien, aldus het nuttige met het aangename verenigend. als resultaat daarvan treft u op de volgende pagina's aan: het schema van de voeding en het aanzicht van beide zijden van het printplaatje dat erin zit. zelfs van het stekertje waarmee de voeding aan de newbrain aangesloten wordt, heeft hij een tekening gemaakt

Schema NewBrain-voeding





steker voor voeding NewBrain
aanzicht op de bussen



controller



DISKCONTROLLER VERSIE 5

Tijdens de vorige NewBraindag zijn door de gebruikersgroep ongemonteerde printplaten van de NewBrain diskcontroller versie 5 verkocht met de aanduiding NB12145/B/5. Van de betreffende controller zijn door de firma Tradecom geen schema's vrijgegeven en ook de betreffende software in de vorm van twee eproms ontbrak. Een grondige controle van de print aan de hand van het schema van versie 3 bracht een aantal belangrijke wijzigingen aan het licht, die zijn onder te verdelen in drie categorieën:

1. correctieve herstellingen van een aantal noodlijnen (draadverbindingen op de print als gevolg van ontwerpfout).
2. principiële verbeteringen. Zo is in de nieuwe versie voorzien in een handmatige resetschakeling en is een aantal decoderingssignalen op een andere manier uitgepoort.
3. kwalitatieve verbeteringen van een aantal signalen. Het 4MHz kloksignaal stond bekend als een zwakke schakel en is door extra buffering verbeterd. Ook het MREQ- en RD-signaal zijn nu gebufferd ten opzichte van de expansiebus.

Met deze feiten voor ogen kunnen we ons af vragen, of het zinvol is de controller zelf op te gaan bouwen. De kosten voor de onderdelen zullen volgens een globale schatting zo'n f 125,- bedragen, wanneer men voor het controller-IC en de eproms gebruikte exemplaren gebruikt, en kunnen nog iets worden gedrukt, wanneer ook voor de andere onderdelen kan worden geput uit een dumpvoorraad. De opbouwtijd moet voor iemand met goede soldeerervaring gesteld worden op 20 uur, wanneer vanaf het schema wordt gebouwd, en 10 uur wanneer naar een voorbeeld wordt gewerkt. Voor het monteren, afregelen en testen moet nog eens minimaal 10 uur worden uitgetrokken.

Aan de andere kant is het niet irreëel te verwachten, dat binnenkort kant-en-klare gebruikte controllers via de gebruikersgroep of particulieren voor eenzelfde of lagere prijs zullen worden aangeboden.

Zelfbouw kan een plezierige en leerzame bezigheid zijn, waarbij men de nodige ervaring kan opdoen, zowel op elektronisch gebied als met de werking van diskunits. Als resultaat heb je dan een kwalitatief goede controller voor een redelijke prijs.

Als eerste kennismaking met elektronica is het bouwen van de controller absoluut ongeschikt en zelfs voor ervaren knutselaars, die niet over een zee van tijd en veel geduld beschikken, een project dat niet is aan te bevelen.

Zelf heb ik nu een werkend exemplaar van deze controller, en na vier weken proefdraaien ziet het er naar uit, dat alle bugs eruit zijn en het systeem goed functioneert. Voor de IC's in de 7400-serie heb ik waar mogelijk gebruik gemaakt van de HC- of HCT-uitvoering. Het voordeel hiervan is een lager stroomgebruik en een hogere storingsongevoeligheid. De ontkoppelcondensatoren dienen echter wel wat ruimer te worden gekozen in verband met de grotere terugwerking op de voedingslijnen van deze typen IC's.

Een goede waarde is bijvoorbeeld 220 nF in plaats van 100 nF. Verder heb ik de voedingsschakeling van IC 449 wat aangepast. In versie 3 werd deze verzorgd door R123, D122 en TR113 maar in versie 5 wordt de spanning rechtstreeks vanaf zehnerdiode D122 (die in dit geval een waarde moet hebben van 5,2 volt) afgenomen. Daar deze methode een minder stabiele instelling tot gevolg heeft, heb ik een spanningsregelaar 78L05 toegepast in plaats van D122. Dit IC kan zonder wijziging van de print worden ingepast.

De schema's van versie 3 heb ik zo goed mogelijk aangepast aan de gevonden wijzigingen en tevens heb ik de volledige onderdelenlijst met

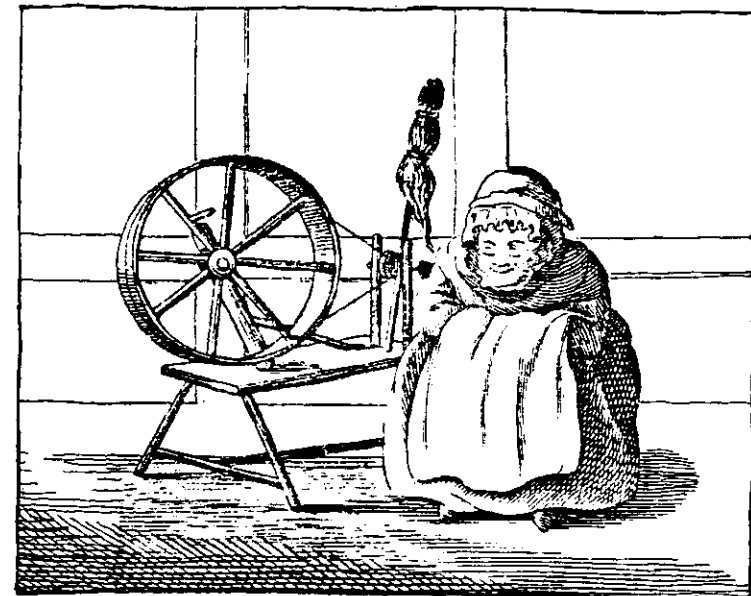
alle gebruikte componenten opgesteld (zie onder). Het schema is op verzoek bij mij te verkrijgen, en waar mogelijk zal ik kandidaat-bouwers met advies bij staan.

J. P. Koning

type	waarde	fabrikaat	leverancier	prijs	opmerking	bestelnnummer
IC401	LM 358	national	elincom	0,66	voedingsregeling	2120
IC402	74HCT244	philips	elincom	0,92	stuursignaalbuffer	2023
IC403	74LS260	ti	texim	1,10	pagingdecoder	2020
IC404	74HC244	philips	elincom	0,92	adresbusbuffer	2025
IC405	74HCT21	philips	elincom	0,39	pagingdecoder	2023
IC406	74HCT02	philips	elincom	0,39	reset en clock	2023
IC407	74HCT14	philips	elincom	0,40	drivebuffer	2023
IC408	7438	philips	elincom	0,67	floppydiskdriver	2010
IC409	7438	philips	elincom	0,67	floppydiskdriver	2010
IC410	SY2114AL		sloop		statische ram	
IC411	SY2114AL		sloop		statische ram	
IC412	74HC244	philips	elincom	0,92	adresbusbuffer	2025
IC413	AM2764	amd	sloop		geprogrammeerd v2	
IC414	74HCT245	philips	elincom	1,05	databusbuffer nb	2023
IC415	74HCT245	philips	elincom	1,05	databusbuffer fdc	2023
IC416	Z80	zillog	sloop		cpu 4 MHz	
IC417	AM2764	amd	sloop		geprogrammeerd v1	
IC418	D765C	nec	sloop		diskcontroller	
IC419	74HC32	philips	elincom	0,39	diverse functies	2025
IC420	7438	philips	elincom	0,67	floppydiskdriver	2010
IC421	74HC161	philips	elincom	0,79	leesclockdeler	2025
IC422	74HC11	philips	elincom	0,39	diverse functies	2025
IC423	74HC153	philips	elincom	0,74	schrijfclocksturing	2025
IC424	74HC174	philips	elincom	0,72	schrijfclocksturing	2025
IC425	74HC08	philips	elincom	0,47	diverse functies	2025
IC426	74HC32	philips	elincom	0,39	diverse functies	2025
IC427	74HC133	philips	elincom	0,67	poortdecoder	2025
IC428	74HC174	philips	elincom	0,72	poort 255	2025
IC429	74HCT244	philips	elincom	0,92	diverse functies	2023
IC430	74HC174	philips	elincom	0,72	765 enable register	2025
IC431	74HC138	philips	elincom	0,63	enable register dec.	2025

IC432	74HCT32	philips	elincom	0,39	diverse functies	2023
IC433	74HC125	philips	elincom	0,55	besturingsbusdriver	2025
IC434	26S02	fairchild	sloop		dual monostabiele	
IC435	74HC112	philips	elincom	0,39	data separator	2025
IC436	uA741T	rca	elincom	0,55	data separatie	2120
IC437	74HCT04	philips	elincom	0,40	diverse functies	2023
IC438	74HCT20	philips	elincom	0,39	pagingdecoder	2023
IC440	74HCT11	philips	elincom	0,39	diverse functies	2023
IC441	74HCT11	philips	elincom	0,39	diverse functies	2023
IC442	74HCT04	philips	elincom	0,40	diverse functies	2023
IC443	74HCT08	philips	elincom	0,39	memorydecoder	2023
IC444	74HCT04	philips	elincom	0,40	diverse functies	2023
IC445	74HCT74	philips	elincom	0,55	busrequest timing	2023
IC446	74HC00	philips	elincom	0,39	diverse functies	2025
IC447	74HCT74	philips	elincom	0,47	diverse functies	2023
IC448	74HC112	philips	elincom	0,39	leesclockdeler	2025
IC449	MC4024	motorola	sloop		vco equivalent 74ls627	

TR001	ZTX212	ferranti	sloop			
C 002	100 NF	siemens	elincom	0,29	MS100NB4	4180
R 003	4700 ohm	philips	elincom	0,06	MR 25	4060
C 004	100 NF	siemens	elincom	0,29	MS100NB4	4180
R 005	560 ohm	philips	elincom	0,06	MR 25	4060
R 006	180 kohm	philips	elincom	0,06	MR 25	4060
R 007	47 kohm	philips	elincom	0,06	MR 25	4060
TR008	ZTX107	ferranti	sloop			
D 009	1N4002		elincom	0,10		1050
R 010	4700 ohm	philips	elincom	0,06	MR 25	4060
C 011	680 PF	philips	elincom	0,20	PK 1	4205
C 012	22 uF	philips	elincom	0,20	E22R63	4220
C 013	100 NF	siemens	elincom	0,27	MS100NB4	4180
R 014	39 kohm	philips	elincom	0,06	MR 25	4060
D 015	ZN404	ferranti	sloop		referentiediode	
D 016	1N4002	div	elincom	0,10		1050
R 017	4700 ohm	philips	elincom	0,06	MR 25	4060
C 018	100 NF	siemens	elincom	0,29	MS100NB4	4180
C 019	680 PF	philips	elincom	0,20	PK 1	4205
C 020	100 NF	siemens	elincom	0,29	MS100NB4	4180
TR021	ZTX650	ferranti	sloop			
C 022	47 uF	philips	elincom	0,32	E47R50	4220
R 023	56 ohm	philips	elincom	0,06	MR 25	4060
R 024	41K2	philips	elincom	0,06	MR 25	4060
R 025	1200 ohm	philips	elincom	0,06	MR 25	4060



C 026	47 uF	philips	elincom	0,38	E47R50	4220
R 027	100 ohm	philips	elincom	0,06	MR 25	4060
C 028	100 NF	siemens	elincom	0,29	MS100NB4	4180
C 029	100 NF	siemens	elincom	0,29	MS100NB4	4180
R 030	4700 ohm	philips	elincom	0,06	MR 25	4060
TR031	2N5193	div	sloop			
R 032	3300 ohm	philips	elincom	0,06	MR 25	4060
R 033	4x100 ohm	bourns	elincom	0,37	R082S101	4075
R 034	4x100 ohm	bourns	elincom	0,37	R082S101	4075
R 035	4x100 ohm	bourns	elincom	0,37	R082S101	4075
R 036	4x100 ohm	bourns	elincom	0,37	R082S101	4075
R 037	8x4700 ohm	bourns	elincom	0,48	R091C472	4075
R 038	4x100 ohm	bourns	elincom	0,37	R082S101	4075
R 039	4x100 ohm	bourns	elincom	0,37	R082S101	4075
R 040	2200 ohm	philips	elincom	0,06	MR 25	4060
R 041	8x4700 ohm	bourns	elincom	0,48	R091C472	4075
R 042	4x100 ohm	bourns	elincom	0,37	R082S101	4075
R 043	4x100 ohm	bourns	elincom	0,37	R082S101	4075
R 044	4x100 ohm	bourns	elincom	0,37	R082S101	4075
R 045	4x150 ohm	bourns	elincom	0,45	R061C151 (5 weerst.)	4075
R 046					vervallen	

R 047			vervallen		
R 048			vervallen		
R 049			vervallen		
C 050 220 NF	siemens	elincom 0,33	MS220NB4	4180	
R 051 220 NF	siemens	elincom 0,33	MS220NB4	4180	
R 052 220 NF	siemens	elincom 0,33	MS220NB4	4180	
R 053 220 NF	siemens	elincom 0,33	MS220NB4	4180	
C 054 220 NF	siemens	elincom 0,33	MS220NB4	4180	
C 055 220 NF	siemens	elincom 0,33	MS220NB4	4180	
C 056 220 NF	siemens	elincom 0,33	MS220NB4	4180	
C 057 220 NF	siemens	elincom 0,33	MS220NB4	4180	
C 058 220 NF	siemens	elincom 0,33	MS220NB4	4180	
C 059 220 NF	siemens	elincom 0,33	MS220NB4	4180	
C 060 220 NF	siemens	elincom 0,33	MS220NB4	4180	
L 062 4,7 uH	siemens	elincom 0,50	SP4U7	6090	
C 063 220 NF	siemens	elincom 0,33	MS220NB4	4180	
C 064 22 uF	philips	elincom 0,20	E22R63	4220	
C 065 220 NF	siemens	elincom 0,33	MS220NB4	4180	
R 066 1200 ohm	philips	elincom 0,06	MR 25	4060	
C 067 100 NF	siemens	elincom 0,27	MS100NB4	4180	
C 068 100 NF	siemens	elincom 0,27	MS100NB4	4180	
TR069 ZTX212	ferranti	sloop			
D 070 BZV85 5Z6	philips	elincom 0,35	zehner 5,6 volt	1070	
R 071 10 kohm	bourns	elincom 1,33	P3006P10K multituurn	4140	
R 072 33 ohm	philips	elincom 0,06	MR 25	4060	
R 073 4x100 ohm	bourns	elincom 0,37	R082S101	4075	
074			afwezig		
R 075 22 ohm	philips	elincom 0,06	MR 25	4060	
TR076 2N3906	div	sloop			
C 077 33 pF	philips	elincom 0,10	KP1-33P	4205	
078					
R 079 1200 ohm	philips	elincom 0,06	MR 25	4060	
R 080 220 ohm	philips	elincom 0,06	MR 25	4060	
R 081 3300 ohm	philips	elincom 0,06	nr 25	4060	
082					
R 083 33 ohm	philips	elincom 0,06	MR 25	4060	
R 084 10 kohm	bourns	elincom 1,33	P3006P10K multituurn	4140	
R 085 15 kohm	philips	elincom 0,06	MR 25	4060	
C 086 150 pF	philips	elincom 0,10	KP2-330P	4205	
R 087			vervallen		
R 088			vervallen		
R 089			vervallen		
R 090			vervallen		

L 091 4,7 uH	siemens	elincom 0,50	SP4U7	6090	
C 092 22 pF	philips	elincom 0,11	KP2-22P	4205	
R 093 5100 ohm	philips	elincom 0,06	MR 25	4060	
R 094 2200 ohm	philips	elincom 0,06	MR 25	4060	
C 095 100 NF	siemens	elincom 0,27	MS100NB4	4180	
C 096 220 PF	philips	elincom 0,13	KP2-220P	4205	
R 097 2200 ohm	philips	elincom 0,06	MR 25	4060	
R 098 3300 ohm	philips	elincom 0,06	MR 25	4060	
C 099 100 NF	siemens	elincom 0,27	MS100NB4	4180	
C 100 100 NF	siemens	elincom 0,27	MS100NB4	4180	
R 101 10 kohm	philips	elincom 0,06	MR 25	4060	
C 102 4700 pF	philips	elincom 0,15	PK204N7	4205	
C 103			vervallen		
R 104 1000 ohm	philips	elincom 0,06	MR 25	4060	
R 105 1000 ohm	philips	elincom 0,06	MR 25	4060	



C 106 1500 pF	philips	elincom 0,11	KP2-1N5	4205	
R 107 3300 ohm	philips	elincom 0,06	MR 25	4060	
R 108 10 kohm	philips	elincom 0,06	MR 25	4060	
C 109 47 NF	siemens	elincom 0,36	MS47NB4	4180	
C 110 1500 pF	philips	elincom 0,11	KP2-1N5	4205	
C 111			vervallen		
R 112 1000 ohm	philips	elincom 0,06	MR 25	4060	

D 114 BZV85 921	philips	elincom	0,37		1070
C 115 22 uF	philips	elincom	0,20	E22R63	4220
R 116 1000 ohm	philips	elincom	0,06	MR 25	4060
D 117 BAX 16	div	elincom	0,32	verv voor 1N914	1060
D 118 BAX 16	div	elincom	0,32	verv voor 1N914	1060
R 119 1000 ohm	philips	elincom	0,06	MR 25	4060
TR120 ZTX212	ferranti	sloop			
R 121 680 ohm	philips	elincom	0,06	MR 25	4060
IC122 78L05	div	elincom	0,72	wijziging schema	2100
R 122				vervallen wijziging	
R 124 10 kohm	philips	elincom	0,06	MR 25	4060
R 125 10 kohm	philips	elincom	0,06	MR 25	4060
C 126 47 uF	philips	elincom	0,38	E47R50	4220
R 132 499 ohm	philips	elincom	0,06	MR 25	4060
R 133 100 ohm	philips	elincom	0,06	MR 25	4060
R 127 200 ohm	philips	elincom	0,06	MR 25	4060
R 128 499 ohm	philips	elincom	0,06	MR 25	4060
S 129 schakelaar	onbekend			doorverbinding	
R 130 20 kohm	philips	elincom	0,06	MR 25	4060
R 131 20 kohm	philips	elincom	0,06	MR 25	4060
C 134 100 NF	siemens	elincom	0,27	MS100NB4	4180
C 135 100 NF	siemens	elincom	0,27	MS100NB4	4180
R 136 100 ohm	philips	elincom	0,06	MR 25	4060
C 137 220 pF	philips	elincom	0,13	KP2-220P	4205
C 138 220 NF	siemens	elincom	0,36	MS220NB4	4180

specificatie

TECHNISCHE GEGEVENS VAN
ZTX107, ZTX212, ZTX650 EN ZN404

in de newbrain is een aantal componenten verwerkt van britse fabrikanten, die aan deze kant van het kanaal nauwelijks bekend, en wier produkten dus in lang niet alle handboeken te vinden zijn. m s vreedenburg heeft toch de technische gegevens ervan achterhaald om ze op deze manier aan de solderende newbraingebruikers ter beschikking te stellen.

in deze aflevering van newbrain on-line komen aan de beurt van ferranti de transistoren ZTX107, ZTX212 en ZTX650, evenals de ZN404, op het eerste gezicht een gewone zenerdiode, maar in werkelijkheid een heel fabriekje



ZTX107: NPN general purpose
ZTX212: PNP general purpose
ZTX650: NPN medium power

Type	V_{CE0} V	V_{CE0} V	Max Cont. I_C mA	Max $V_{CE(sat)}$ at			h_{FE} at			Min f_T at		P_{tot} at $T_{amb} = 25^\circ C$ mW
				V	I_C mA	I_B mA	Min	Max	I_C mA	MHz	I_C mA	
ZTX107	60	50	100	0.1	10	1	125	500	2	350	10	300
ZTX212	60	50	200	0.25	10	0.5	60	400	2	200	10	500
ZTX650	60	45	2000	0.3	1000	100	100	300	500	100	100	1000

TE KOOP AANGEBODEN

newbrain ad
diskcontroller met 2 drives
expansion
monitor
matrixprinter

w van houten
telefoon (070) 453510

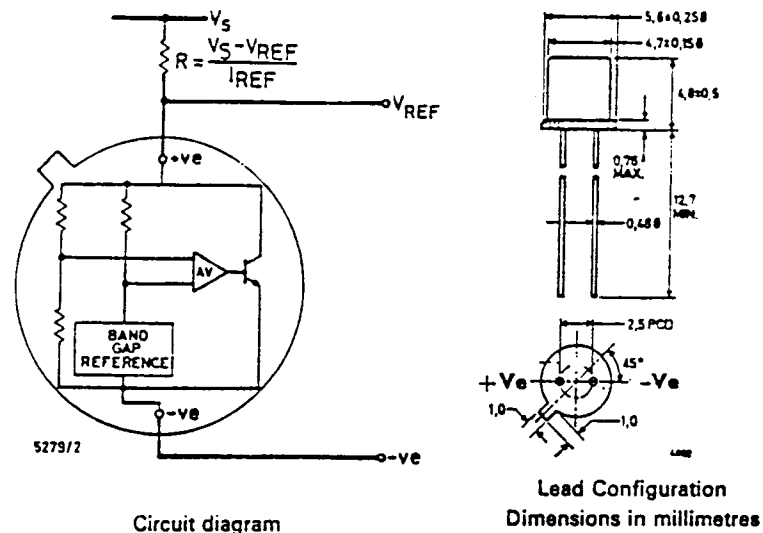
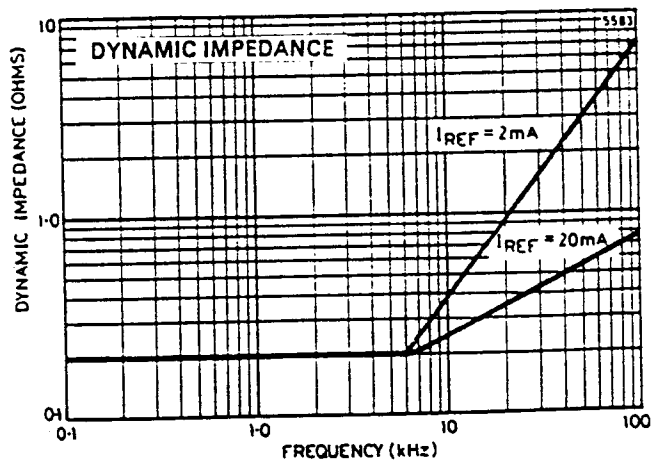
ZN404

The ZN404 is a monolithic integrated circuit providing a precise stable regulator source of 2.45 volts in a two lead package without the need for an external shaping capacitor.

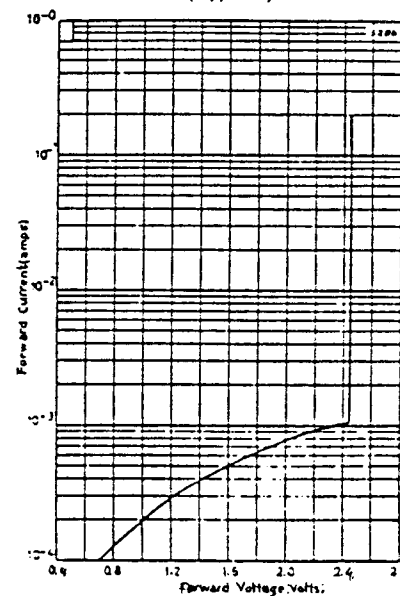
Maximum ratings: dissipation: 300 mW
operating temperature range: 0 to +70 °C
storage temperature range: -55 to +150 °C

ELECTRICAL CHARACTERISTICS (at 25°C ambient temperature unless otherwise stated).

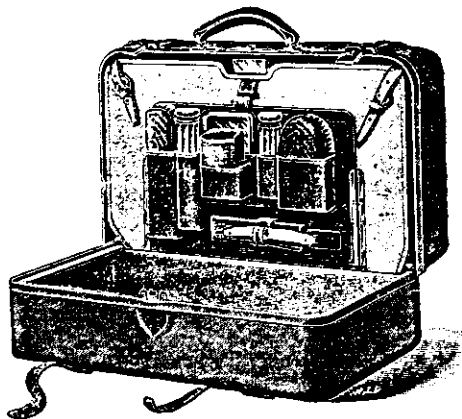
Parameter	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output Voltage	V_{REF}	2.38	2.45	2.52	V	Measured at 2 mA
Slope Resistance	R_{REF}	—	0.2	0.4	Ω	
Reference Current	I_{REF}	2	—	120	mA	
Maximum Change in V_{REF}	ΔV_{REF}	—	6	25	mV	0 to +70°C
RMS Noise Voltage 1 Hz – 10 kHz		—	10	—	μV	
V_{REF} Drift at 70°C		—	± 10	—	ppm/1000 hours	



FORWARD CHARACTERISTIC
(Typical)



inhoud on-line 14



- 2 lijst van medewerkers
- 3 nieuws betreffende de softwarebibliotheek /w a van hoek/
- 6 catalogus van de nieuwe software
- 8 toelichting bij enkele programma's die verschenen zijn op bestelnummer N314 van de softwarebibliotheek /wim luijt/
- 14 vraagjes en weetjes, vragen en weeën
grundy business systems limited (in liquidation)
- 17 pogo /simon murphy/
- 25 het gebufferde toetsenbord op de 32k newbrain met diskcontroller
/wim luijt/
- 29 paged memory operating system /wim luijt/
- 43 rombox /j p koning/
- 51 schema van de voeding /m s vreedenburg/
- 55 diskcontroller versie 5 /j p koning/
- 63 technische gegevens van ztx107, ztx212, ztx650 en zn404
- 67 bestelbiljet softwarebibliotheek

66

inhoud

newbrain on-line 14

NewBrain-
gebruikersgroep
postbus 4494
1009 AL amsterdam

softwarebibliotheek BESTELFORMULIER

naam _____

adres _____

postcode en woonplaats _____

telefoon _____

hcc-lidmaatschapsnummer _____

bestelt bij deze de aan ommezijde aangegeven software

formaat aankruisen: ☐ cassette ☐ diskette 200k ☐ diskette 400k ss ☐ diskette 400k ds ☐ diskette 800k ☐ _____	_____ bestelnummers à f 10,00 = f _____ _____ bestelnummers à f 15,00 = f _____ _____ bestelnummers à f 5,00 = f _____ verzendskosten f 2,50 f _____ T O T A A L f _____
---	--

bovenstaand totaalbedrag

☐ is overgemaakt d d _____

op postrek 2505800 tnv hcc newbrain-gebruikersgroep amsterdam

afkomstig van bank-/postrek nr _____

tnv _____ te _____

☐ wordt voldaan met bijgesloten girobetaalkaart,
betaal- of eurocheque

nummer giropas of betaalpas _____

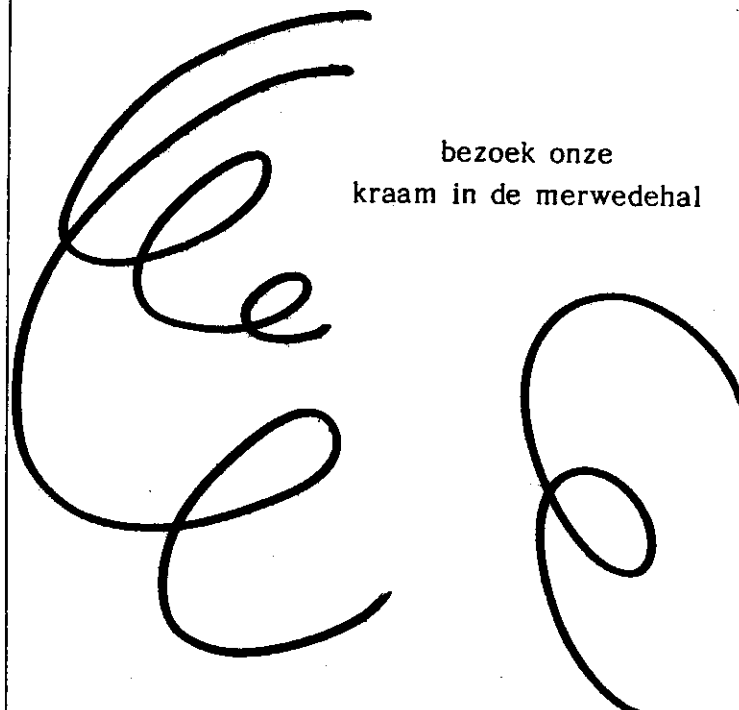
handtekening

cassettesoftware	*) = ook verkrijg- baar op cassette; de andere van de serie N... en de serie SCHIJF- alleen op diskette	___ N309 ___ N310 ___ N311 ___ N312 ___ N313 ___ N314 ___ N401 ___ N402	volumes uit de programmatheek van de dos- gebruikersgroep
___ ASM-1	___ N201	___ N501 *)	___
___ BWK-1	___ N202	___ N502 *)	___
___ DIVERS-1	___ N203	___ N503 *)	___
___ DIVERS-2	___ N204	___ N504 *)	___
___ DIVERS-3	___ N205	___ N505 *)	___
___ DIVERS-4	___ N206	___ N506	___
___ DIVERS-5	___ N207	___ N507	___
___ DIVERS-6	___ N208	___ N508	___
___ DIVERS-7	___ N209	___ N509 *)	___
___ EDUC-1	___ N210	___ N510 *)	___
___ EDUC-3	___ N211	___ N511 *)	___
___ EDUC-4	___ N212	___ N512	___
___ FIN-1	___ N213	___ N513	___
___ GRAFI-1	___ N214		___
___ GRAFI-2	___ N215		___
___ HULP-1	___ N216		___
___ HULP-2	___ N217		___
___ HULP-3	___ N218		___
___ MATH-1	___ N219		___
___ MATH-2	___ N220		___
___ MATH-3	___ N221		___
___ MATH-4	___ N222		___
___ MATH-5	___ N223		___
___ NBFILES	___ N224		___
___ ONLINE-5	___ N225		___
___ ONLINE-6	___ N226		___
___ ONLINE-7	___ N227		___
___ ONLINE-8	___ N228		___
___ ONLINE-9	___ N229		___
___ ONLINE-10	___ N230		___
___ SPEL-1	___ N231		___
___ SPEL-2	___ N232		___
___ SPEL-3	___ N233		___
___ SPEL-4	___ N307		___
___ SPEL-5	___ N308		___
___ SPEL-6			___
___ SPEL-7			___
___ SPEL-8			___
___ UGV-1			___

hcc-dagen 1989

24 en 25 november jaarbeurs utrecht

bezoek onze
kraam in de merwedehal



NewBrain-
gebruikersgroep