

frankeren als  
briefkaart

NEWBRAIN G.G.  
LEDENADMINISTRATIE  
TALMASTRAAT 20-2e

1073 JX AMSTERDAM

---

*New Brain*  
**on-line**

UITGAVE VAN DE  
NEWBRAIN  
GEBRUIKERS GROEP

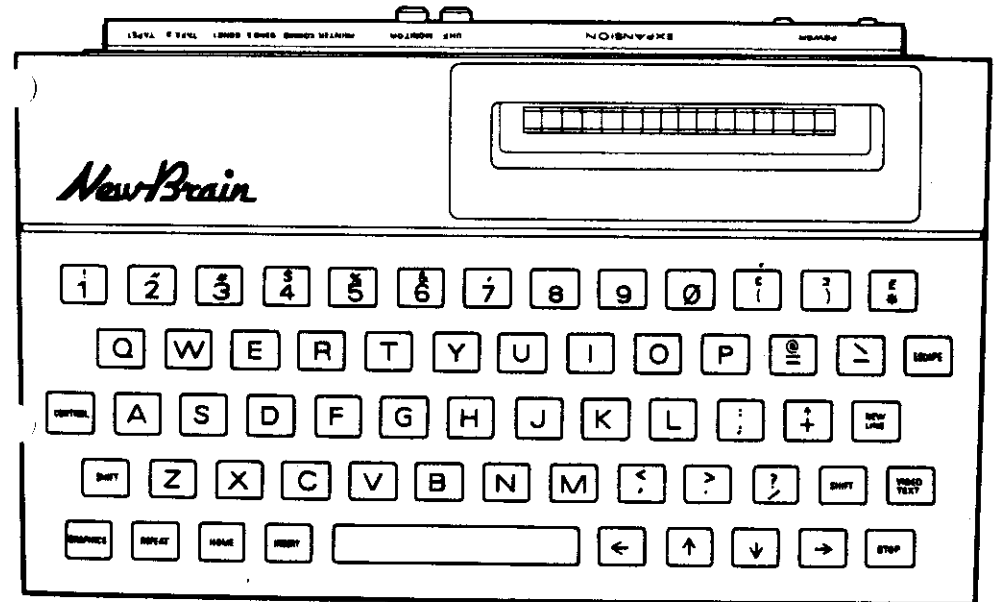
**84-2**

frankeren als  
briefkaart

NEWBRAIN G.G.  
LEDENADMINISTRATIE  
TALMASTRAAT 20-2e

1073 JX AMSTERDAM

---



**NEWBRAIN-ON-LINE** is een officiële uitgave van de **NewBrain Gebruikers Groep Nederland**. Zodra er voldoende, voor de NewBrain gebruiker, nuttige kopij voorhanden is volgt er een nieuwe uitgave. De bladen zijn licht gelijmd zodat ze eenvoudig los te maken en daarna in een klapper op te bergen zijn. De perforatie is afgestemd op de klapper met H.C.C. opdruk. Deze is tijdens de landelijke NewBrain gebruikers dagen of bij de H.C.C.te koop.

Er bestaat de mogelijkheid om zich te abonneren op regelmatige toezending. Zie het aanmeldingskaartje achter in de uitgave. De prijsstelling zal ongeveer liggen op Fl. 5.50 voor 50 pagina's, plus Fl. 2.50 voor administratie- en verzendkosten.

De redactie ziet graag kopij van de leden tegemoet. Het liefst getypt op A4 formaat; laat links en rechts een marge van 3.5 cm. en boven en onder een marge van 4.5 cm. blank.

Geplaatste artikelen mogen alleen voor niet-commerciele doeleinden onder bronvermelding overgenomen worden.

Voor informatie omtrent commerciële advertenties of advertenties van leden kunt U contact opnemen met de redactie.

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

**Hoofdredacteur:**

C.L. Truijens  
Hamburgerstraat 19  
3512 NP Utrecht  
030-311087

PRIJS PER EXEMPLAAR : **F 7.50**

Het bestuur van de **NewBrain Gebruikers groep**.

Voorzitter : J. de Vries  
Wolter Heukelslaan 17  
3581 SP Utrecht  
030-511243

Secretaris : R. van Albada  
Talmastraat 20-2e  
1073 JX Amsterdam  
020-739671

Penningmeester : B. Hovius  
Bitterstraat 58  
8011 XM Zwolle  
038-228252  
Postrek. 2090007

Redacteur : C.L. Truijens  
HCC-Nieuwsbrief Hamburgerstraat 19  
NewBrain On-line 3512 NP Utrecht  
030-311087

**Werkgroepen:**

Onderwijs	: P. van Dijk	01751-12367
CP/M	: P. Reinking	02510-44687/36493
Tekstverwerking	: R. van Albada	020-739671
Viditel	: P. Reinking	02510-44687/36493
	: B. Rijnbeek	02995-2264
Basicode	: B. Rijnbeek	02995-2264
Hardware	: R. Maris	055-424485
Data-Comm.	: T. Kamperman	03438-17084
	: J. den Oude	03410-21913
Pascal	: J. de Vries (a.i.)	030-511243
Assembler	: B. Rijnbeek	02995-2264
Inkoop	: A. Bothof	010-218390

**Software Bibliotheek:**

	: J. Hermans	Postbus 2286 5001 CG Tilburg
alleen CP/M	: P. Reinking	02510-44687/36493

**Informatie regionale bijeenkomsten:**

Utrecht	:030-511243	Limburg	:013-552530/04498-53507
Overige	:030-511243		

## Voorwoord

Het succes van NewBrain On-line 84-1 heeft velen geïnspireerd om hun ervaringen met de NewBrain op te schrijven. Een dik deeltje 84-2 dat nu voor U ligt is er het resultaat van. Je kunt merken dat we allemaal in vergelijking tot begin 1984 weer heel wat meer weten; de stukken in dit deel zijn dan ook wat moeilijker en bevatten meer informatie dan de stukken in het eerste deel.

Dit boekje is als volgt ingedeeld. Eerst komen er een aantal stukken en stukjes die betrekking hebben op de NewBrain in zijn basisuitvoering; vervolgens enige bijdragen die gaan over de Expansion Unit en/of diskgebruik. Daarna volgen enige handlei-leidingen van programma's uit de NewBrain software bibliotheek. U weet wel, die unieke service die voor U op maat samengestelde bandjes of diskettes aflevert. Van het programma EDITOR is een zeer uitgebreide instructie opgenomen; de redactie vond dit een schoolvoorbeeld van hoe documentatie er uit moet zien. Ten slotte is er een ledenlijst.

De ledenlijst is deze keer gesorteerd op postcode. We hebben het bij wijze van experiment een keer zo gedaan, om contacten tussen NewBraingebruikers in eenzelfde regio te bevorderen. Om zoeken op naam mogelijk te maken, is er na de ledenlijst nog een verkorte lijst op naam.

De catalogus van de softwarebibliotheek is hier niet opgenomen; die wordt los meegeleverd. We doen dit zo, om U de beschikking te geven over een catalogus die zo goed mogelijk "bij" is.

In een volgend nummer van On-line zullen we een register maken van tot dan toe geplaatste stukken, om het terugvinden van informatie over bepaalde onderwerpen te vergemakkelijken.

Ik wijs nog even op de bestelprocedure voor programma's uit de software bibliotheek. De catalogus is te verkrijgen door een aan jezelf geadresseerde enveloppe, gefrankeerd met f 1.10 te sturen aan Kees Truijens. Bestellingen zijn alleen mogelijk met het bestelformulier, dat bij de catalogus zit. Het beheer van de bibliotheek is van Jan Vooijs naar Jos Hermans gegaan (adressen op pag. 3). Voor bestellingen van CP/M programma's geldt een iets andere regeling. Zie hiervoor de HCC-Nieuwsbrief van juni 1984.

De map met stukken voor On-line is nu leeg. Dus graag bijdragen voor nr 84-3 aan de redactie. Programma's voor de bibliotheek zijn ook altijd welkom. Maar één noodkreet van de nieuwe bibliothecaris: documenteer a.u.b de programma's, met REM statements in het programma en met een bijgevoegde toelichting of gebruiksaanwijzing. We hebben verschillende erg knap uitzijnde programma's binnen gekregen, waarvan we niet kunnen achterhalen waarvoor ze precies bedoeld zijn.

Kees Truijens

## WERKGROEP TERMINAL EMULATIE

Theo Kamperman, Schlippersdreef 65, 3972 VB Driebergen, 03438- 17084

Bij de oprichting van de NBGG heb ik "terminal emulatie" op het kaartje ingevuld als speciale belangstelling. Ik werd dan ook meteen gebombardeed tot voorganger van een werkgroep met die naam. Op deze plaats wil ik graag duidelijk maken wat ik met die kreet bedoel.

Een terminal is een input/output station aan een mini/maxi of supercomputer. Daarmee kan meestal zowel interactieve als batch verwerking gedaan worden door programma's op de betreffende computer. Ieder operating systeem van grote computers heeft eigen afspraken wat betreft cursor besturing, speciale karakters e.d.. Er is wel een standaard voor (ANSI), maar daar houden weinigen zich aan. Doordat de eigen intelligentie van terminals steeds groter wordt, zijn er nu ook die, naar keuze, zich kunnen gedragen of zij de elgen terminal van verschillende grote computer systemen zijn. Die terminals emuleren dan een ander type terminal.

De werkgroep wil net zolets doen, namelijk een NewBrain zich laten gedragen als terminal van een ander computersysteem, zodat via bijvoorbeeld een telefoonlijn ingelogged kan worden op zo'n systeem. Software in de NewBrain zorgt er dan voor dat, bij interactief gebruik, de speciale karakters die de grote computer begrijpt, geconverteerd worden.

Een andere toepassing die belangstelling heeft is het opsturen van files van NewBrain naar een andere computer en andersom. Daarbij is het b.v. mogelijk foutcorrectie in te bouwen, want dat is bij gebruik van telefoonlijnen zeer wenselijk.

Voor beide toepassingen zijn, bij gebruik van telefoonlijnen, modems nodig. En die zijn duur. Ik ben bezig er zelf een te maken, voorlopig 300 baud, op basis van de EXAR 14412 chip. Die is goedkoop, en daarmee moet het mogelijk zijn voor zo'n f 175,- een compleet modem te maken. Maar de documentatie van EXAR zit boordevol fouten, zodat het nog even kan duren voordat het werkt.

De echte interactieve toepassing eist nog meer hardware, n.l. een gebufferde input/output port aan de NewBrain. Zonder die is geen onafhankelijke input/output mogelijk en zitten we b.v. "zonder" scherm als we op input wachten. Kuma in Engeland levert dit wel, maar dat kost vierhonderd gulden! Liever zou ik zien dat het nederlandse CTM ontwerp voor een redelijke prijs hier geproduceerd kon worden.

Het probleem voor de werkgroep is dus zowel hard- als software. Eerst zal het eerste en dan pas het laatste probleem opgelost moeten worden. Voor b.v. een IBM-PC is het allemaal aanwezig. Dan moet het toch zeker voor een NewBrain opgelost kunnen worden? Is er iemand in de buurt van Utrecht die wil helpen? Wat ervaring op dit terrein is wel nodig.

Inkoop hardware randapparatuur.

De werkgroep 'inkoop' houdt zich bezig met de inkoop van hardware randapparatuur voor Neubrain gebruikers.

Door gezamenlijke inkoop van b.v. printers, disc-drives, monitors, diskettes, cassettebandjes etc., kan vaak een aanzienlijk kortingspercentage gerealiseerd worden (15 tot 30%).

De nu volgende vragenlijst is bedoeld om een inzicht te verkrijgen, waar op dit moment de meeste belangstelling voor is.

De werkgroep kan op deze manier een goed overzicht krijgen van hardware apparatuur die leden nog wensen aan te schaffen.

Een kort voorbeeld kan de werkwijze van de werkgroep inkoop verduidelijken.

De vragenlijst wijst uit dat er belangstelling voor printers is.

De werkgroep onderzoekt welke printers aantrekkelijk zijn gezien prijs/kwaliteit verhouding en kortingspercentage, waarbij rekening gehouden wordt met door de leden op te geven wensen zoals prijsindicatie, bijzonderheden etc.

Daarna doet de werkgroep een voorstel aan die leden die tot aanschaf van een printer willen overgaan met exacte opgave van prijs, kortingspercentage en garantiebepalingen, (nog steeds vrijblijvend!).

Daarna kan inkoop en distributie plaats vinden.

Leden die precies weten welke printer zij willen aanschaffen, kunnen dit op de vragenlijst vermelden.

Ook in het geval waarin het gaat om aanschaf van slechts 1 printer, kan een behoorlijk kortingspercentage gerealiseerd worden (dit is wel afhankelijk van de importeur).

Stuur de vragenlijst in!

Het verplicht u tot niets en het bespaart u zeker geld, indien u van plan bent om nu of in de nabije toekomst randapparatuur aan te schaffen.

----- VRAGENLIJST -----

Naam: .....
Adres: .....
Postcode: .....
Plaats: .....
Telefoon: .....

Ik overweeg nu of in de nabije toekomst aan te schaffen:

Table with columns: apparatuur, ja/nee, specificatie/bijzonderheden, beschikbaar bedrag. Rows include PRINTERS (dot matrix, daisy wheel, inkjet, kleurenprinter, overige), PLOTTERS (zwart/wit, kleur, overige), DISC-DRIVES (5.25 inch, 3 inch, overige), DISC-CONTROLLER, EXP. INTERFACE, CASSETTE-REC. (analoog, digitaal, overige), MONITORS (zwart/wit, kleur (RGB), overige).

Heeft u een expansion-interface? (van belang als u een printer wilt gaan aanschaffen) ja / nee

Ook inkoop van inktlinten voor printers, papier (kettingformulier en normaal), cassettebandjes, diskettes, kabels etc. is mogelijk. Aantal en type opgeven.

Overige: .....

Vragenlijst opsturen naar: Werkgroep "inkoop"
A.H. Bothof
Inkarnaatklaver 11
3069 DN, Rotterdam

Handwritten number: 218390

NOGMAALS: HOE SNEL IS LANGZAAM ?

De interactie tussen de Z-80 en de video chip loopt via locatie 31. De inhoud van deze geheugenplaats is van groot belang voor de snelheid waarmee de Newbrain zijn opdrachten uitvoert. Zodra PEEK(51) nul is wordt een gedeelte van de tijd die het operating system voor de interne huishouding uittrekt, gebruikt om het beeldscherm van de juiste informatie te voorzien. U ziet dan wat er op het scherm (#) en ook andere videostreams aanwezig is. Een mogelijkheid om de snelheid van de Newbrain op te voeren is het leegmaken van het scherm door middel van PUT#p,31. Maar eigenlijk is dit sjouwen met een groter emmertje. Het lege scherm wordt toch nog gescreend of er karakters op staan die verfrist moeten worden.

Op dat moment moet PEEK(51) bekeken worden. Een even waarde, ook nul, laat de 4Mc-klok voor de COP-processor werken. Iedere oneven waarde, b.v. 1, laat de COP niet toe zijn werk op het scherm te verrichten. Het scherm wordt niet gechecked, en het O.S. gaat snel verder met andere taken, o.a. programmaonderdelen die uitgevoerd moeten worden.

De verschillen in exectietijd met een leeg scherm sec en een leeg scherm d.m.v. POKE 51,1 komen duidelijk in de executietijden van de benchtests zoals die in Op Line 84-1 staan tot uitdrukking. Een lijstje:

Benchtest	PUT 31	POKE 51,1
1	1.55	1.58
2	4.95	4.66
3	15.88	15.92
4	15.34	14.94
5	17.26	16.3
6	28.16	26.9
7	43.40	41.40
8	61.98	58.72
	Totaal 23.8	22.55

Beide tijden zijn dus met een leeg scherm opgetekend ! De verschillen met een redelijk gevuld beeldscherm zijn nog veel groter.

Zodra er echter iets misgaat op het moment dat peek (51) gelijk aan 1 is, zal de rapportering ERROR x AT y wel verschijnen, maar onzichtbaar zijn. Een aparte trap in de vorm van

```
1 ON ERROR GOTO 20000
20000 POKF 51,0*REPORT
```

lost dit probleem op.

Anton vd Repe

SNELLER DAN SNEL

In On-line 84-1 geeft Jan de Vries door hem gemeten tijden voor acht populaire benchmark programma's. Elders in dit nummer laat Anton van de Repe zien dat het nog een pietse sneller kan door het display tijdens het rekenen uit te schakelen

Wij te het allemaal nog sneller, dan zul je je toevlucht moeten nemen tot een taal als FORTH (hoe staat het met de FORTH-werkgroep ??) of tot een gecompileerde taal. Voor de standaard NewBrain is er bij mijn weten maar een compiler beschikbaar en dat is HiSoft Pascal (HP4T) Ik heb de benchmarks in Pascal vertaald en ze met HP4T getest. Ik geef niet alle listings, slechts nr 1, 7 en 8. De nummers 6 tm 2 zijn uit nr 7 te verkrijgen door successievelijk programmaregels weg te laten

In tegenstelling tot NewBrain BASIC (helaas !) kent Pascal het data-type "integer". Rekenen met integers is veelal aanmerkelijk sneller dan met reals. Om van de ene kant de vergelijking zo eerlijk mogelijk te maken, maar van de andere kant ook het onderste uit de kan te halen, zijn de tests zowel met reals als met integers uitgevoerd. Bedenk hierbij overigens wel dat BASIC floating-point variabelen werken met een mantisse van vijf bytes lang en HP4T reals met een mantisse van drie bytes. Dat is uiteraard zowel van invloed op de nauwkeurigheid als op de snelheid waarmee gerekend wordt

Test nr 1 kan niet met reals worden uitgevoerd (in een Pascal DO loop mag alleen een "scalar data type" worden gebruikt, test nr 8 kan niet met integers worden uitgevoerd ranscendentale functies).

Onderstaande tabel bevat voor NewBrain BASIC de tijden zoals die door Jan de Vries zijn opgemeten. Behalve voor test 8 was het in FASCAL nodig om de testloop 10000 keer te laten lopen i.p.v. 1000 maal, om de tijdschatting nauwkeurig genoeg te laten zijn. In de kolom "factor" is aangegeven hoeveel maal sneller HP4T is in vergelijking tot BASIC.

Overigens is het met het oog op de snelheid zeer belangrijk dat de juiste compiler opties gekozen worden (regel 5 van de listings). De listings laten zien dat besturingstekens voor het scherm en peek en poke adressen in HP4T vaak net zo kunnen worden gebruikt als in BASIC

De tests zijn gedaan met een leeg beeldscherm, maar het display stond wel aan. Benchmark 8 heb ik achteraf nog gewijzigd n.a.v. het stukje van Anton van de Repe. (zie regels 120 en 135) Dit leverde nog eens een tijdwinst op van ruim 5 %.

Tot slot: Voor een compilertaal werkt HP4T gemakkelijk en snel. Editor, compiler, run time library, source en objectprogramma staan alle tegelijk in het geheugen, waardoor het geheel zich aan de gebruiker zowat als een interpreter voordoet. Ik heb nog geen ervaring hoe e.e.a. werkt bij wat langere programma's. Misschien iets voor een volgende On-line?

Bench- mark	Basic	HP4T		factor	
		integer	real	integer	real
1	1.7	0.024	-	69	-
2	5.0	0.038	0.47	150	11
3	16.9	0.53	1.77	32	10
4	15.8	0.42	1.94	38	8
5	17.3	0.46	1.97	38	9
6	28.2	0.60	2.12	47	13
7	43.4	0.79	2.38	55	18
8	62.0	-	8.18	-	6

```

5 {#0-,C-,S-,A-,I-}
10 PROGRAM benchmark8;
20 VAR a,b,c,k:REAL;
50 t1,t2:CHAR;
60 tijd:REAL;
100 BEGIN
105 WRITE(CHR(31));
110 k:=0;
120 POKE(106,CHR(0));POKE(107,CHR(0));POKE(51,CHR(1));
130 WHILE k<1000 DO
140 BEGIN
150 k:=k+1;
160 a:=SQR(k);
170 b:=LN(k);
180 c:=SIN(k);
220 END;(WHILE)
230 t1:=PEEK(106,CHAR);t2:=PEEK(107,CHAR);
235 POKE(51,CHR(0));
240 tijd:=(256*ORD(t1)+ORD(t2))/50;
250 WRITELN(CHR(16),tijd,CHR(13),CHR(13),CHR(16));
260 END.

```

```

5 {#0-,C-,S-,A-,I-}
10 PROGRAM benchmark7r;
20 VAR a,k:REAL;
30 l:INTEGER;
40 m:ARRAY[1..5] OF REAL;
50 t1,t2:CHAR;
60 tijd:REAL;
70 PROCEDURE dummy;
80 BEGIN
90 END;(dummy)
100 BEGIN
105 WRITE(CHR(31));
110 k:=0;
120 POKE(106,CHR(0));POKE(107,CHR(0));
130 WHILE k<10000 DO
140 BEGIN
150 k:=k+1;
160 a:=k/2*3+5-5;
170 dummy;
180 FOR l:=1 TO 5 DO
190 BEGIN
200 m[l]:=a;
210 END(FOR)
220 END;(WHILE)
230 t1:=PEEK(106,CHAR);t2:=PEEK(107,CHAR);
240 tijd:=(256*ORD(t1)+ORD(t2))/50;
250 WRITELN(CHR(16),tijd,CHR(13),CHR(13),CHR(16));
260 END.

```

```

5 {#0-,C-,S-,A-,I-}
10 PROGRAM benchmark1;
20 VAR k:INTEGER;
50 t1,t2:CHAR;
60 tijd:REAL;
100 BEGIN
105 WRITE(CHR(31));
120 POKE(106,CHR(0));POKE(107,CHR(0));
130 FOR k:=1 TO 10000 DO
140 BEGIN
220 END;(FOR)
230 t1:=PEEK(106,CHAR);t2:=PEEK(107,CHAR);
240 tijd:=(256*ORD(t1)+ORD(t2))/50;
250 WRITELN(CHR(16),tijd,CHR(13),CHR(13),CHR(16));
260 END.

```

## Hoe krijg je een tekening op papier?

Als je met je NewBrain prachtige plaatjes op je beeldscherm hebt getoverd, wil je die plaatjes ook wel eens overbrengen op papier. Met mijn matrix printer, een IDS microprism, waarvan je de printkop naar believen naar iedere positie op het papier kunt sturen en waarvan je bovendien de printnaalden apart kunt besturen, moet zoiets kunnen. Dacht ik.

En het kan ook. Maar ik moest eerst twee problemen uit de weg ruimen: ten eerste blijkt mijn NewBrain van een ouder type te zijn waarin de functies PEN(7), PEN(8) en PEN(9) niet zijn ingebouwd. Dank zij het stukje van B. Rijnbeek (NewBrain On-line 84-1, pag. 16) weet ik dat ik een NewBrain heb met operating system 1.4. De functies PEN(7) en PEN(8) geven bij nieuwere NewBrains het beginadres en het eindadres +1 van het stuk geheugen waar het high resolution gedeelte van de scherm informatie is opgeslagen. In het Handboek noch in het Software Technical Manual is te vinden hoe je op een andere manier aan deze adressen kunt komen. Terwijl je ze beslist nodig hebt om een 'screendump' te kunnen maken.

Het tweede probleem was het volgende: als je precies weet welk gedeelte van het geheugen je naar de printer moet sturen, dan blijkt dat je in mijn geval (met die IDS microprism) niet simpelweg alle bytes uit dat stuk geheugen kunt oversturen, maar dat je die bytes eerst in bits moet ontleden en dan nieuwe bytes moet vormen die geschikt zijn om naar de printer te sturen. Dit probleem is op te lossen door met een Basic programma de bytes te ontleden en opnieuw samen te stellen, maar het duurt dan vele minuten voor er een plaatje op papier staat. Daarom ben ik me maar eens gaan verdiepen in het programmeren in machinecode. Op de volgende bladzijde staat een Basic programma waarin de binnenste lus is vervangen door een call naar een stukje machinecode en dit programma maakt een screendump binnen een acceptabele tijd. Later heb ik een programma gekregen, gemaakt door Rob Maris en dat klaart de klus in een minimum van tijd.

Maar dat tweede probleem kon ik pas aanpakken nadat ik het eerste had opgelost. Hieronder wil ik weergeven hoe ik dat heb gedaan en hoe die oplossing eruit ziet.

Ik heb eerst eens gebeld met de firma Tradecom, maar daar wist men ook niet hoe je aan het begin- en eindadres van het betreffende stuk geheugen kon komen. Toen heb ik het zelf maar geprobeerd en eigenlijk viel het wel mee. In de eerste plaats weet je dat een graphics stream (van type 11) zijn geheugen moet delen met een stream van het type 0 (scherm) of 4 (scherm en regeldisplay). Het gezochte stuk geheugen ligt dus tussen het begin- en eindadres van het eigen geheugen van de bijbehorende stream en deze twee adressen zijn terug te vinden via de stream table. Dit is een tabel in het geheugen waarin van iedere geopende stream onder meer het beginadres van het eigen geheugen is opgeslagen. Het bijbehorende eindadres +1 is gelijk aan het beginadres van de volgende stream of aan het adres STRTAB. (Zie ook het stukje van Jan Rekers in NewBrain On-line 84-1, pagina 6) Als je het geheugen tussen deze twee grenzen (dus tussen het beginadres en het eindadres van het eigen geheugen van de bijbehorende stream van het type 0 of 4) bekijkt met een "print peek(i)"-programmaatje, dan zie je tot een bepaald punt bytes met ASCII-waarde 32 en daarna met waarde 0.

Vele bytes verder gaat het patroon van ASCII-waarden 0 weer over in waarden 32. Het aantal bytes dat tussen deze overgangen in ligt blijkt aardig in de buurt te komen van het aantal bytes dat het graphics gedeelte van het scherm in beslag zou moeten nemen. Het zijn er in feite 40 meer bij een stream die is geopend met parameter "s" (40 tekens per regel). Dit bracht mij op de volgende gedachte: zoek in het geheugen naar twee opeenvolgende bytes die samen het geheugenadres vormen waarop de waarden 32 overgaan in 0 en naar twee bytes die het adres bevatten waarop de nullen overgaan in 32. Dit zoeken kan weer met een eenvoudig "peek-programmaatje" gebeuren. Op deze wijze kwam ik er achter dat in de twee bytes die zich op de posities 44 en 45 van het eigen geheugen van de graphics stream bevinden het adres staat van de overgang van nullen naar 32. In de twee bytes daar vlak voor (posities 42 en 43) blijkt een adres te staan dat precies zoveel plaatsen eerder in het geheugen aanwijst als oversenkomt met het aantal bytes dat nodig is voor het high resolution graphics deel van het scherm.

Zo had ik de oplossing gevonden: bij verschillende situaties - streams met andere parameters en in andere volgordes geopend - bleek steeds weer dat ik op deze manier het beginadres en het eindadres +1 kon vinden van het high resolution graphics gedeelte van het scherm.

Tot slot volgt hier een programma waarvan in het eerste deel (regels 10000 tot en met 10080) de variabelen p7 en p8 de waarden krijgen die overeen komen met de functies PEN(7) en PEN(8) bij de nieuwere NewBrains. Het tweede deel van het programma (regels 10100 tot en met 10330) zorgt er voor dat de bytes op de juiste wijze naar de IDS-microprism printer gestuurd worden. Er wordt verondersteld dat de printer is aangesloten op de COMMS uitgang en dat er met OPEN#9,9 reeds een stream naar de printer is geopend. Ik ben er tot nu toe niet in geslaagd de waarde van PEN(9) te vinden. Wie kan mij daar mee helpen?

```

10000 DEF FN(ad)=PEEK(i) + 256*PEEK(i+1)
10010 a1=FNad(86):a2=FNad(100): REM a1=STRTAB; a2=STRTOP
10020 i=a1+1:REM zoek in de stream table naar device driver 11
10030 IF PEEK(i)=11 GOTO 10060: REM device driver 11 gevonden
10040 i=i+6: IF i<a2 GOTO 10030
10050 PRINT "er is geen graphics stream geopend": STOP
10060 a=FNad(i+3): REM a=beginadres eigen geheugen van graphics stream
10070 p7=FNad(a+42): p8=FNad(a+44)
10080 PRINT "beginadres is " p7: PRINT "eindadres + 1 is " p8

10090 PRINT "staat de printer aan?"
10100 PUT#9,13,3: RESERVE 41: RESTORE 10000
10110 FOR i=TOP TO TOP+40
10120 READ a: POKE i,a
10130 NEXT i
10140 POKE TOP+1,7
10150 INPUT("hoeveel tekens per regel? ")nb: POKE TOP+7,nb
10160 n7=7*nb
10170 FOR i=p7 TO p8-1 STEP n7
10180 IF i+n7=p8 POKE TOP+1,(p8-i)/nb
10190 FOR ad=i TO i+nb-1
10200 a=INT(ad/256):POKE TOP+12,a: POKE TOP+11,ad-256*a

```



```

10210 CB=70
10220 FOR j=0 TO 7
10230   POKE TOP+14,CB:CB=CB+8
10240   CALL TOP
10250   NEXT j
10260 NEXT ad
10270 PUT#9,3,14
10280 NEXT i
10290 PUT#9,3,2
10300 DATA 6,0,62,8,144,79,30,0,175,87,33,0,0,203,0,40,2,203,199,15,25
10310 DATA 16,246,65,15,16,253,30,9,71,231,48,62,3,184,32,2,231,48,175
10320 DATA 201
10330 END

```

Deventer, 4 maart 1984

Sytse Knipstra

### CHRROM

CHRROM ,de tabel van de karakterpatronen die de New Brain gebruikt om karakters in een hirescreen te plotten.

- Startadres v/d tabel:peek(119)+256\*peek(120)
- Voor elk karakter geeft de tabel 8 bytes.  
De 1e byte is te vinden op startadres tabel + ASC(kar)  
Voor de 2e,3e etc.steeds 256 bijtellen.
- De 1e - of de 1e en de 2e byte van een niet - graphics karakterpatroon kunnen oneven zijn (bit 0 set).Dan is het karakter een descender (zoals de g en j) en moeten deze bytes Even gemaakt en onder de rest v/h patroon geplaatst worden.
- Voor de graphics (0 t/m 31 en 128 t/m 159) geeft de tabel ook 8 bytes.Er zijn er eigenlijk 10 nodig,maar omdat de laatste 3 altijd gelijk zijn hebben we er aan 8 voldoende.
- Deze patronen zijn goed te gebruiken voor het maken van grote karakters (slechtzienden,reclame,lichtkrant).
- Onderstaand programma maakt het e.e.a wat duidelijker.

P.Visman

```

10 REM                                     demo bij CHRROM
20 REM
100 kt=PEEK(119)+256*PEEK(120)
110 PUT31:INPUT("karakter ")k$
120 ?"karakter ";ASC(k$)
130 sk=kt+ASC(k$)
140 IFASC(k$)AND96?"gewoon karakter":GOTO190
150 ?"graphic":?"2 ontbrekende bytes = 8 ste"
160 FORi=0TO7
170   k(i)=PEEK(sk+i*256)
180 NEXTi:GOTO230
190 FORi=0TO7
200   k(i)=PEEK(sk+i*256)
210   IFk(i)AND1?i+1;"e moet zijn";k(i)-1;"op rij";9+i
220 NEXTi
230 ?":FORi=0TO7:?k(i):NEXTi
240 ??:INPUT("ander kar? (j/n)")a$:IFa$="j"ORa$="J"GOTO110
250 END

```

## De Karaktergenerator in de NewBrain

Onderstaande beschrijft hoe op efficiënte wijze de karaktergenerator van de NewBrain aangesproken kan worden. Het frappante is nu, dat de eigenlijke karaktergenerator helemaal niet gelezen kan worden, omdat simpelweg de hardware dat niet toelaat. Toch wordt er voor de zeropage van de NewBrain een pointer opgegeven die het begin van de karaktergenerator aanwijst. Dit blijkt dan echter een kopie te zijn van een gedeelte van de karaktergenerator. De karaktergenerator zit in een 4K ROM. In de A000-BFFF ROM zit een kopie van exact 2K. Dat is die van karakterset 2 (de meest uitgebreide van 2,5K met o.a. de griekse tekens). Men heeft dit in de grafische ROM in "packed format" opgeslagen. In de echte karaktergenerator ROM kan dat niet want je kunt hardware niet zomaar ingewikkelde klussen laten doen als het "unpacken" van informatie. Daar kost deze generator dus inderdaad 2,5K. Dan blijft er dus nog 1,5K over voor de andere sets. Welnu set 1 komt vanuit set 2 tevoorschijn. Set 4 is weer een op zichzelf staande set, waarvan de helft van het gebied boven code 127 domweg de inverse is van de karakters 32 t/m 95. Dus er zijn  $128+64=192$  verschillende karakters. Dit is  $192*8=1536$  bytes. Dus exact 1,5K. Set 3 is weer afgeleid, en wel van set 4. Dus het totaal is inderdaad 4K.

Als we nu voor snelle toepassingen (dus zonder tussenkomst van een grafisch scherm) karaktermatrices willen uitlezen, zullen we dus rechtstreeks de kopie van de generator moeten uitlezen. De hamvraag daarbij is, hoe is de kopie "gepacked". Het antwoord is niet zo moeilijk te vinden. Ieder karakter kan beschreven worden met 8 bytes informatie (dat moet ook wel, als we binnen 2k willen blijven). De 8 bytes per karakter liggen als volgt in het geheugen. De eerste byte ligt op het adres (in het adres zo u wilt)  $B800 + \text{codenummer van het karakter}$ . Dus bijv. voor een A is dat dus B841 hex. Iedere volgende infobyte ligt 256 plaatsen verder (hier dus B941, BA41, BB41, BC41, BD41, BE41 en BF41). In de onderstaande routine (helaas in machinetaal vanwege de snelheid en de eenvoudigere implementatie) worden deze bytes verzameld in een blok vanaf adres KEYBUF. Dit gebeurt m.b.v. de 5 instructies na TRANSFR. Even voor TRANSFR wordt getest of het karakter voldoet aan de naamgeving "FIGS", het is een FIGS (figures) als het karakter de waarden 0-32 of 128-160 heeft. Regel is hier dat het achtste byte (op  $\text{KEYBUF}+7$ ) gekopieerd moet worden naar het 9de en 10de byte. Hiermee is zo'n matrix dan weer volledig. Dat was dus eenvoudig en bovendien makkelijk na te gaan in het NewBrain handboek. Nu blijkt ook, waarom sommige tekens vrij hoog in de  $8*10$  matrix liggen (bijv. de 4 pijltjes vanaf code 155). Direct na de TRANSFR operatie staat een JR C,FIGS instructie. Bij FIGS gebeurt het genoemde kopiëren.

Het andere geval geldt als alle cijfers, letters e.d. gekozen worden. Het is hier zo, dat de meest rechtse kolom van ieder karakter geen informatie bevat. Verder is het zo, dat nooit meer dan 8 rijen informatie bevatten. Of het zijn de onderste 2 die leeg zijn, of het zijn de bovenste

2. Als volgt is dit nu gepacked: Altijd worden de bovenste 8 rijen in het ROM gevonden. Stel nu dat de onderste 2 rijen informatie behoren te bevatten (bijv. enkele kleine letters en tekens), dan zijn de bovenste 2 rijen leeg (eigenschap van kleine letters). In het ROM zijn nu op de plaatsen van de bovenste rijen de onderste rijen neergezet, met als toevoeging, dat bit 0 geset is (in de meest rechtse kolom). De decodering ervan in deze routine gebeurt na NOFIGS. Het begrip ervan laat ik aan den lezer over. Ik vermeld slechts de rare DEFB instructie. Hier is de byte FE gedefinieerd. Dit is de opcode voor CP n. Als de JR naar SPEC1 plaatsvindt, zal de waarde van A 0 zijn na dat label. Als het programma die JR niet doet, komt het langs die FE byte en ziet de XOR A (A=0) instructie als zijn operand, zodat er na het label niets aan A is veranderd. Dit bespaart 1 byte.

Het programma is te gebruiken als interface tussen een programma dat een karaktermatrix verwacht en het ROM, dat de karakters in "packed form" opgeslagen heeft. Het karakter moet in C meegegeven worden, in DE wordt het begin van de buffer meegegeven (bijv. net na TOP). Deze beginwaarde is aan het eind terug te vinden in HL.

Het programma kan op een gewone manier ingeladen worden boven TOP, of ergens anders, het programma is reloceerbaar. Dus als module goed te gebruiken.

Eigenaren van de expansion interface zullen er rekening mee moeten houden dat de karaktergenerator in SLOT 1 moet zitten voordat de routine aangesproken wordt.

Het pagnummer van de page waar de karaktergenerator staat (we praten hier nog steeds over de "packed" kopie) staat op adres AE en AF.

Rob Maris

KEYBUF EQU 2000H

;CHR NR. SPECIFIED IN C

```

11 00 20      ( LD  DE,KEYBUF  ;=MATRIXFIELD )
2A 77 00      LD  HL,(77H)  ;START OF CHRGEN
06 00         LD  B,0
09           ADD  HL,BC
79           LD  A,C
E6 7F        AND  7FH
FE 20        CP  32
06 08        LD  B,8
DS          PUSH DE
7E          TRANSFR LD  A,(HL)
12          LD  (DE),A
13          INC  DE
24          INC  H
10 FA        DJNZ TRANSFR
E1          POP  HL
38 14        JR  C,FIGS      ;C FROM FIGTEST
E5          PUSH HL
06 02        LD  B,2
7E          NOFIGS LD  A,(HL)
1F          RRA
30 05        JR  NC,SPEC1
CB 27        SLA  A
36 00        LD  (HL),0
FE          DEFB OFEH      ;CP IMMED.
AF          SPEC1 XOR  A
12          LD  (DE),A
23          INC  HL
13          INC  DE
10 F1        DJNZ NOFIGS
AF          XOR  A
E1          POP  HL
12          FIGS  LD  (DE),A
13          INC  DE
12          LD  (DE),A
C9          ( RET )

```

;CHARACTER MATRIXFIELD FROM (HL) ON

KARAKTERS: Overvloed of chaos?

Op pagina 142 van de handleiding staat dat de meeste karakters vanaf het toetsenbord kunnen worden ingevoerd. Dit is onjuist !! Alle 256 karakters kunnen via het toetsenbord worden ingevoerd. Bovendien kan dit meestal zonder moeilijke trucs. Veel karakters kunnen zelfs op meer dan één manier worden ingevoerd. Een en ander is beschreven in de NewBrain Technical Note 13 Issue 1. Welk karakter door een bepaalde toets gegenereerd wordt hangt af van de Control Mode. Hoewel er regels zijn die het verband tussen toets, control mode en ingevoerd karakter bepalen, zijn er teveel uitzonderingen om deze regels om op een eenvoudige wijze een en ander duidelijk te maken. Ik zal me daarom beperken tot de tabellarische informatie uit bovengenoemde technical note en enkele praktische aanwijzingen. In onderstaande tabel heb ik geen karakters gegeven omdat deze afhankelijk zijn van de gebruikte karakter-set (zie pg.142 van de handleiding) en mijn printer en typemachine ze ook niet kent.

Gebruikte afkortingen:

S/	: shift	gelijktijdig op shift en de gewenste	
G/	: graphics shift	,, , graphics	,, toets
C/	: control shift	,, , control	,,
E:	: escape	achtereenvolgens op escape	,,
A:	: attributes on	,, , S/↑	,,
		of ,, , C/N	,,
		vergeet geen attributes off (S/E of	
		C/O) om weer "normaal" in te voeren!	
1/	: mode 1: control 2,3,6,7,8, of 9	cl cursor left	
2/	: mode 2: control 4,5,6, of 7	cu cursor up	
3/	: mode 3: control 8, of 9	cd cursor down	
sb	spatiebalk	nl newline	cr cursor right
ins	insert	esc escape	vt video text

1<sup>e</sup> toets(en)

2 <sup>e</sup> toets	E:C/	E:	--	S/	--	G/	2/G/	3/G/	1/S/	96:G/
=	ø		32 sb	64		128	160	192	224	192:G/
A	1	S/ins	33 S/1	65	97	129	161	193	225	193:G/
B	2	S/cd	34 S/2	66	98	130	162	194	226	194:G/
C	3	C/nl	35 S/3	67	99	131	163	195	227	195:G/
D	4		36 S/4	68	100	132	164	196	228	196:G/
E	5		37 S/5	69	101	133	165	197	229	197:G/
F	6		38 S/6	70	102	134	166	198	230	198:G/
G	7		39 S/7	71	103	135	167	199	231	199:G/
H	8	cl	40 S/8	72	104	136	168	200	232	200:G/
I	9	C/esc	41 S/9	73	105	137	169	201	233	201:G/
J	10	cd	42 *	74	106	138	170	202	234	202:G/
K	11	cu	43 +	75	107	139	171	203	235	203:G/
L	12	home	44 ,	76	108	140	172	204	236	204:G/
M	13	S/nl, nl	45 -	77	109	141	173	205	237	205:G/
N	14	S/cu	46 .	78	110	142	174	206	238	206:G/
O	15	S/esc	47 /	79	111	143	175	207	239	207:G/
P	16	G/esc	48 ø	80	112	144	176	208	240	208:G/
Q	17	G/nl	49 1	81	113	145	177	209	241	209:G/
R	18	ins	50 2	82	114	146	178	210	242	210:G/
S	19	G/cu	51 3	83	115	147	179	211	243	211:G/
T	20	G/home	52 4	84	116	148	180	212	244	212:G/
U	21	G/ins	53 5	85	117	149	181	213	245	213:G/
V	22	C/ins	54 6	86	118	150	182	214	246	214:G/
W	23		55 7	87	119	151	183	215	247	215:G/
X	24	cl	56 8	88	120	152	184	216	248	216:G/
Y	25	S/cr	57 9	89	121	153	185	217	249	217:G/
Z	26	cr	58 S/;	90	122	154	186	218	250	218:G/
(	27	esc	59 ;	91		155	187	219	251	219:G/
-	28	C/cl	60 S/,	92		156	188	220	252	220:G/
)	29	C/cr	61 =	93		157	189	221	253	221:G/
+	30	C/home	62 S/.	94		158	190	222	254	222:G/
Attributes off				Attributes on						
31	63	95	127	159	191	223	255			
E:S/home	S/l	S/vt	C/,	S/home	S//	S/vt	C/,			

Toelichting bij de tabel.

1<sup>e</sup>toets: mode (1,2,3), Shift (S,C,G) en Escape toetsen. S,C en G dienen gelijktijdig te worden aangeslagen met de 2<sup>e</sup> toets; Escape moet voor iedere aanslag om een grafisch karakter 0-31 te krijgen. Wordt dit niet gedaan dan worden de aangeslagen toetsen als editor opdrachten beschouwd (zie appendix 3 van de handleiding). Meerdere mogelijkheden staan in kolom E:. Ook deze toetsen moeten worden voorafgegaan door Escape voor een grafisch karakter.

2<sup>e</sup>toets: aanslag van de 2<sup>e</sup> toets (in combinatie met de 1<sup>e</sup>) voert het karakter met het vermelde ASCII-nummer in. De invoer van de karakters 32-62 gebeurt niet door de combinatie 1<sup>e</sup> en 2<sup>e</sup> toets, maar direkt zoals ze op de toetsen staan.

Ook de karakters 96, en 123-126 vallen buiten de combinatie. 192-195 kunnen ook op een andere wijze worden ingevoerd.

Praktische aanwijzingen.

Attributes on (Shift/ of C/N) telt <sup>128</sup> Bij de ASCII-nummers 0-127 op en trekt 128 van de ASCII-nummers 128-255 af. Attribute off = Shift/Escape of C/O.

Video text geeft 2 karakters: \_ en ü (met Att/off resp. Att/on). Bij het intypen van teksten in verschillende talen (zoals boektitels en tijdschriftartikelen) vind ik persoonlijk de gemakkelijkste wijze om de "vreemde" letters in te typen het gebruik van de ATT/on. Met een klein spiekbriefje voor het vinden van de 2<sup>e</sup> toets werkt dit het doeltreffendst.

Worden veel griekse letters gebruikt dan is Control 4 handig. Kleine en hoofdletters zijn "normaal" (zonder of met Shift) en Graphics geeft het griekse alfabet.

Heeft men veel tekst in een vreemde taal dan is een softwarematige wijziging van het toetsenbord handig. Zelf heb ik de volgende subroutine voor de Franse taal in een tekstverwerker (TEXY 2) aangebracht. Het principe is om een aantal karakters die gemakkelijk zijn aan te slaan en zeer zelden in een tekst voorkomen een andere code te geven.

```

9   REM ## 5 is device 5
10  GET##5,z:IFz > 124IFz < 159GOSUB50000
50000 ON INSTR("keytrap",CHR$(z))+1 GOSUB 50010,50020,
      50030,50040,50050,50060,50070,50080,50090:RET
50010 RET
50020 z=204:RET:REM à
50030 z=205:RET:REM a
50040 z=208:RET:REM é
50050 z=209:RET:REM è
50060 z=210:RET:REM e
50070 z=217:RET:REM ó
50080 z=221:RET:REM ù
50090 z=225:RET:REM ç
60000 REM "keytrap"=CHR$(125)+CHR$(126)+CHR$(155)+
      CHR$(157)+CHR$(128)+CHR$(156)+CHR$(149)+CHR$(158);
      deze STRING kan je met behulp van de tabel nu zelf
      direkt intypen i.p.v. "keytrap" in regel 50000.

```

Als geheugensteuntje heb ik op mijn NewBrain boven de desbetreffende toetsen kleine etiketjes geplakt.

Zonder de pretentie te hebben om volledig te zijn hoop ik, dat bij jullie met deze aanwijzingen de overvloed aan karakters niet verandert in een chaos van karakters.

Wim Luijt

P.S. Wanneer je nog een andere manier weet om een bepaald karakter in te typen, laat het me dan niet weten, maar stuur het in voor de volgende NewBrain On-Line.

## GROTE TEKENEN

Dit is een programma voor de NewBrain om, na intoetsing van een willekeurig teken, dit op een high resolution scherm in een grote matrix te plotten. Het werkt als volgt.

regel	werking
5 - 30	High resolution gebied opzetten.
40 - 90	Opbouw grote matrix.
100 - 110	Input en plot teken.
120 - 170	Tast teken horizontaal en verticaal af. Is de kleur gelijk aan 1, FILL dan een corresponderende cel in de grote matrix. PEN(b) geeft de kleur van het punt waar op dat moment de pen op staat.
180 - 200	Nog een keer ?

```

5 FOR i=1 TO 255:CLOSE#i:NEXT i
10 OPEN#0,0,"200":OPEN#2,11,"w120"
20 plot rng(300,110),centre(0,0)
30 plot wipe
40 FOR i=1 TO 11
50 plot pla(100,i*10-9),mve(180,i*10-9)
60 NEXT i
70 FOR i=1 TO 9
80 plot pla(i*10+90,100),mve(i*10+90,1)
90 NEXT i
100 ?CHR$(31):"WELK TEKEN WIL JE ":LINPUT a$
110 plot pla(11,6),a$
120 FOR v=1 TO 10:REM verticaal
130 FOR h=1 TO 8:REM horizontaal
140 plot pla(h+10,v+5)
150 IF pen(6)=1 THEN plot pla(h*10+92,v*10-4),fill
160 NEXT h
170 NEXT v
180 INPUT("WIL JE NOG EENS ? (JA/NEE)")b$
190 IF LEFT$(b$,1)="j" OR LEFT$(b$,1)="J" THEN 30
200 END
210 REM Copyright (c) KJ & PP Hermans

```

KJ & PP Hermans  
Euidwijk 2  
2771 CC Boskoop

## GRAFISCH

Het minimum aantal te openen karakterregels (d) voor hires.

## - BEREKENING

# zichtbare karakterregels = 25  
 # bytes / karakterregel = bk (64 of 128, incl EXCESS)  
 # grafische regels = h  
 # bytes / grafische regel = bg (32, 40, 64 of 80)  
 # graf.regels / kar.regel = 10 (op het scherm)

totaal aantal bytes =  $(25-h/10)*bk+h*bg$   
 totaal aantal hiervoor  
 te openen karakterregels =  $((25-h/10)*bk+h*bg)/bk$   
 $d = 25-h/10+h*bg/bk$

## 4 gevallen :

bk=64 bg=32  $d = 25-h/10+h*32/64 = 25+0.4*h$   
 bk=64 bg=40  $d = 25-h/10+h*40/64 = 25+0.525*h$   
 bk=128 bg=64  $d = 25-h/10+h*64/128 = 25+0.4*h$   
 bk=128 bg=80  $d = 25-h/10+h*80/128 = 25+0.525*h$

Er blijven dus 2 gevallen over !

Indien de geheugenruimte voor de "grafische bytes" direct zou aansluiten aan die voor de "karakterbytes", er geen endbytes zouden zijn, de FILL functie geen extra geheugen zou vereisen en we niet het aantal regels met een geheel getal zouden hoeven opgeven, dan zouden we nu klaar zijn.

Er zijn dus nog een paar aanpassingen nodig en wel:

- 1 extra regel om de gaps en endbytes te dekken (4-12 bytes),
- 4 extra voor de FILL functie en
- 1/2 regel extra voor de afronding.

Hiermee komen we dan op de volgende formuletjes:

- voor een N(arrow) display  $d = INT(30.5+0.4*h)$
- voor een W(ide) display  $d = INT(30.5+0.525*h)$

Met behulp van deze formuletjes is een lijstje te maken. Leuker is nog ze toe te passen in een programma, b.v. om automatisch een grafisch scherm van de juiste hoogte te laten openen bij het dumpen van hires van cassette naar scherm.

Onderstaand een lijstje van min aantal te openen regels bij een bepaalde hireshoogte.

W(ide)		N(arrow)	
320 of 640 bits		256 of 512 bits	
h	d	h	d
10	35	10	34
20	41	20	38
30	46	30	42
40	51	40	46
50	56	50	50
60	62	60	54
70	67	70	58
80	72	80	62
90	77	90	66
100	83	100	70
110	88	110	74
120	93	120	78
130	98	130	82
140	104	140	86
150	109	150	90
160	114	160	94
170	119	170	98
180	125	180	102
190	126	190	106
200	135	200	110
210	140	210	114
220	146	220	118
		230	122

P. Visman

De NEWBRAIN en RS 232

Op een unexpanded Newbrain is er geen mogelijkheid het frame waarin bytes in device 8 en 9 worden verzonden en ontvangen, te veranderen.

Deze seriële poorten kennen bytes alleen in het volgende formaat: 1 startbit + 8 databits + 2 stopbits.

Het startbit is 0, de stopbits zijn beide 1.

Een uit- of in te voeren byte wordt dus als volgt in behandeling genomen:

0-X-X-X-X-X-X-X-1-1

Standaard RS232/V24 is dit echter niet en wel om de reden dat RS232 ook andere frames kent.

Bij onzekerheid of de communicatie tussen twee apparaten wel goed verlopen is kan een extra meegezonden bit ervoor zorgen dat het ontvangende apparaat op goede gronden de juistheid van de zojuist ontvangen byte kan bepalen. Dit zogenoemde pariteitsbit wordt dan nl. meegezonden om het aantal nullen of het aantal enen in de DATAbits tot een even getal (0,2,4,6 of 8 enen/nullen) te maken. In andere woorden: er wordt gechecked op "even parity" of op "uneven parity". In ieder geval staat dit bit altijd op de plaats van de eerste stopbit die de Newbrain uitzendt, en die dus altijd een logische 1 is.

Het gevolg is dat b.v. een printer die de pariteit wil checken bij 1200 Baud, zoals mijn Datasab 3572, de helft van de ontvangen karakters niet wil afdrukken omdat de pariteitscheck negatief uitvalt.

Voor mensen die met ditzelfde (op het eerste gezicht) nogal raadselachtige probleem te kampen hebben gehad volgen hier drie oplossingen.

Ten eerste kan de devicedrivertable die op de adressen `peek(88)+256*peek(89)` t.m. dit adres + 24 staat naar een RESERVED gedeelte in RAM worden verplaatst. Daarna dient er een aangepaste versie van de drivers 8 en 9 geschreven te worden (VERGELIJK het model in de Software Technical Manual: "Patching a new Device Driver"), waarin pariteit gechecked wordt, en waarin het ROM-adresgedeelte dat o.a. de startbit en de twee stopbits verzorgt in herschreven versie naar een eveneens RESERVED RAM-gedeelte moet worden verplaatst. Eenvoudig is deze oplossing niet.

Ten tweede kan de printer optionele doorverbindingen, "JUMPERS", hebben die ruimte overlaten voor het format dat de Newbrain zich permitteert. Dan zult u de printer anders moeten instellen. Bent u niet zeker, vraag dan advies !...

Tenslotte de eenvoudigste oplossing, al kost die wel wat meer tijd dan een simpel `PRINT#8,"Hier is uw printer"`.

Hiertoe moet de Newbrain iedere weggezonden byte laten vergezellen van een "schaduwbyte".

Als het bitpatroon van de normale verzonden byte b.v. 0011000 is, dan zal de printer dit karakter niet afdrukken. De printer ontvangt nl. 0-0011000-1-1 en checkt m.b.v. de voorlaatste 1 of het karakter goed is ontvangen. De pariteit klopt niet, en dus zal hij niets uitvoeren. Als de Newbrain nu de ASCII-waarde + of - 128 (=2 tot de macht 7 = het eerste databit) met deze byte meezendt, dan zal de printer op dit karakter moeten reageren omdat de pariteit dan wel klopt: B.v. 0-0011000-1-1 + 0-1011000-1-1

of 0-1010101-1-1 + 0-0010101-1-1.

In principe moet de printer dus een van de twee aldus verzonden bytes accepteren.

Daarmee zijn de voornaamste problemen opgelost, want via een User Defined Function kan een dergelijke schaduwbyte vrij snel bepaald worden:

- gesteld dat de printer via device 9 gestuurd wordt, dus met apart in een GOSUB-statement gedefinieerde CR en LF

GESTELD: SR\$ naar printer, met een maximale regellengte RM dan:

```
OPEN#32,9,"1200":rem of andere baudrate
DEF FNP0$=LEFT$(P8$,RM)
DEF FNP1$=MID$(SR$,J3,1):
DEF FN P2$=CHR$(ASC(FNP1$)+128*SGN(128-ASC(FNP1$)))
P8$="":P9$=""
FOR J3=1 TO LEN(SR$):P8$=P8$+FNP1$+FNP2$:NEXT J3
****1: IF LEN(P8$)>RM THEN P9$=RIGHT$(P8$,RM+1):P8$=FNP0$
PRINT#32,P8$:P8$="" : GOSUB cr + lf
IF P9$>"" THEN P8$=P9$:GOTO ****1
einde print SR$
****2: voorbeeld GOSUB cr+lf:PUT#32,13,141,10,138:return
```

De Newbrain trekt wel een seconde uit om een string van 80 karakters op deze wijze te converteren, maar het gaat in ieder geval. De convertering zal wel sneller gaan wanneer ASCII-waarden die groter zijn dan 128 nooit zullen voorkomen. Ik heb nogal eens te maken met character set 4, waarin de onderkast net een hogere ASCII-waarde heeft.

Wie dit terzijde kan laten kan DEF FN P2\$ definiëren als `DEF FN P2$=FN P1$+CHR$(ASC(FNP1$)+128)` en FN P1\$ in de FOR...NEXT J3-loop weglaten.

FNP1\$ in FN P2\$ is economischer dan de methode in het voorbeeld.

Aansturing van de printeruitgang kan tot problemen leiden omdat een dubbele regellengte moet worden opgegeven. Dit kan d.m.v. een reeds meermalen gepubliceerde routine gebeuren. Het hoofdbezwaar is echter, dat Carriage Return, oftewel CHR\$(13) NIET automatisch wordt uitgevoerd. Oneven pariteit..... toch weer een subroutine.

Prettiger werkt het zelf niet denkende device 9. Het is dan mogelijk LF en CR via aparte GOSUBS uit te voeren.

```
gosub Carriage Return alleen = 6001
gosub Linefeed + Delay       = 6000
```

```
6000 PUT#32,10:FOR WT=1 TO 50:NEXT WT
6001 PUT#32,141
6010 RETURN
```

Pure, door de tekst gestuurde linefeeds kunnen d.m.v. GRAPHICS/j gegenereerd worden wanneer een LF gelijk staat aan CHR\$(10).

Aan de stand van de printerkop: wordt dan niets veranderd.

Ook een CR kan op deze wijze gegenereerd worden: GRAPHICS/m in de tekst zendt CHR\$(141) naar de printer. Onderstrepingen kunnen dus simpel in de tekst worden opgenomen.

B.v. STREEP @@@ Graphics/m + \_\_\_\_\_ YYYZZZ

wordt:

B.v. STREEP\_@@@YYYZZZ

Voor horizontale en verticale TAB en formfeed geldt hetzelfde. Een paar tekentjes in de uit te printen tekst kunnen de printer op deze manier dus laten doen wat je wilt.

Anton van de Repe

### SHOESTRING opslag van data.

De maximale string-lengte wordt bij de NewBrain bepaald door het aantal vrije geheugenplaatsen d.w.z. de lengte van de string is onbeperkt, mits er maar voldoende geheugen vrij is. In databases kan hierdoor een groot bestand zeer efficiënt worden opgeslagen.

Als voorbeeld is een stuk uit een ledenlijst gebruikt. Kenmerken van het bestand zijn: (20 kaarten)

- per lid (kaart) 4 velden

- 1) naam
- 2) adres
- 3) woonplaats
- 4) telefoonnummer

- totaal aantal tekens: 885

- maximaal aantal tekens per veld:

- 1) 27
- 2) 24
- 3) 13
- 4) 11

- 8 velden zijn leeg (geen telefoon).

Wanneer de 20 kaarten met programma 1 worden opgeslagen dan gebruikt het array a\$(20,4) bij dimensionering 646 bytes en na invoer van de 20 kaarten 1531 bytes.

Een zeer efficiënte wijze van opslag is door de afzonderlijke velden achter elkaar, gescheiden door een teken, in één string te plaatsen. Er zijn dan slechts 981 bytes nodig. Als teken tussen de velden wordt ASCII 13 (Newline) gebruikt, zie programma 2. Het voordeel van ASCII 13 als veldscheider is, dat PRINT a\$ een keurig geformatteerde lijst op de printer of het scherm geeft.



Een nadeel is dat a\$ niet in één keer van de tape is in te lezen (wat het array in programma 1 ook niet is). De teruglezing in programma 2 is erg traag. Programma 3 is wel sneller, maar vergt meer zorgvuldigheid bij het invoeren van de gegevens. Lege velden moeten dan ook met Newline worden ingevoerd.

Met deze wijze van opslag kon in het voorbeeld (een steekproef uit de ledenlijst van de NBBG) bijna 40 % geheugenruimte worden bespaard, m.a.w. er kunnen meer dan 50 % meer data in dezelfde ruimte worden opgeslagen. Bij de opslag van data met veel lege velden kan deze besparing nog groter zijn. Vergeet niet om deze lege velden in te voeren met alleen op Newline te drukken, anders gebeuren er rare dingen.

Wim Luijt

```
PROGRAMMA 1
10 DIM A$(20,4): FOR I = 1 TO 20: FOR J = 1 TO 4:
LINPUT A$(I,J): NEXT J: NEXT I: STOP: REM invoer data
100 CLOSE#1: OPENOUT#1,1: FOR I = 1 TO 20: FOR J = 1
TO 4: PRINT#1,A$(I,J): NEXT J: NEXT I: PUT#1,4:
CLOSE#1: STOP: REM schrijven op tape
200 CLOSE#1: OPENIN#1,1: FOR I = 1 TO 20: FOR J = 1
TO 4: LINPUT#1,A$(I,J): NEXT J: NEXT I: STOP: REM lezen
data
999 END
```

```
PROGRAMMA 2
10 A$=""
20 LINPUTB$: IF B$ <> "*" THEN A$ = A$ + B$: GOTO 20
30 STOP
100 CLOSE#1: OPENOUT#1,1: PRINT#1,A$: PUT#1,4: CLOSE#1:
STOP
200 CLOSE#1: OPENIN#1,1: A$ = ""
210 GET#1,B: IF B <> 4 THEN A$ = A$ + CHR$(B): GOTO 210
220 STOP
```

```
PROGRAMMA 3
10 A$ = "": T = 0
20 LINPUTB$: IF B$ <> "*" THEN A$ = A$ + B$: T = T + 1:
GOTO 20
30 STOP
100 CLOSE#1: OPENOUT#1,1: PRINT#1,T: PRINT#1,A$:
PUT#1,4: CLOSE#1: STOP
200 A$ = "": CLOSE#1: OPENIN#1,1: LINPUT#1,T: FOR I = 1
TO T: LINPUT#1,B$: A$ = A$ + B$: CLOSE#1: STOP
```

Simulatie van de INPUT (of LINPUT) statement.

De gebruiker (m/v) van een programma kan na een INPUT opdracht een langer antwoord intypen dan de programmeur bedoeld had. Daardoor kan de hele opmaak van het scherm verknoeid worden. Vaak is dat te verhelpen door de INPUT opdracht te vervangen door een voldoende aantal GET opdrachten. Maar de lengte van het antwoord moet vast staan. Bovendien heeft de gebruiker niet, zoals bij de INPUT opdracht, de mogelijkheid zijn antwoord te controleren en te verbeteren alvorens dit door het programma wordt verwerkt.

Door de subroutine in onderstaand programma heeft de gebruiker meer speelruimte: hij kan in zekere mate de lengte van zijn antwoord bepalen en heeft een correctie mogelijkheid met behulp van de ← toets (SHIFT hoeft niet te worden gebruikt). Maar hij kan niet verder typen dan de regel lang is!

```
1 FOR i=1 TO 255:CLOSE#i:NEXT i:GOTO 1000
2
3 REM (L)INPUT simulatie
4
100 PUT 21:GET x:GET y:lm=40-x:'waar is cursor? hoeveel letters
maximaal?
110 in$="":lt=0 :'nog geen letter ingevoerd
120 PUT 6:GET 5,c :'cursor aan:'pak letter'
130 IF c=13 THEN 260 :'NEW LINE, dus antwoord klaar
140 IF c=8 IF lt > 0 THEN lt=lt-1:c=24:GOTO 180
141 '← toets; al letter ingevoerd? dan letterteller met 1
verminderen; ← omzetten in SHIFT←
150 IF c < 32 THEN 120 :'geen geldig karakter
160 IF c > 122 THEN 120 :'idem
170 lt=lt+1 :'letterteller verhogen
180 PUT c :'druk letter (of SHIFT←) af
190 in$=in$+CHR$(c) :'plak de letter aan het antwoord
200 IF lt<=lm THEN 120 :'nog niet maximale aantal tekens
210 PUT 7,22,x,y:FOR i=1 TO lm:PUT 25:NEXT i
211 :'haal het antwoord weg
220 PUT 22,20,y+1:'Antwoord te lang!";
230 FOR i=1 TO 800:NEXT i :'wacht even
240 PUT 30,22,x,y :'wis regel en zet cursor goed
250 GOTO 110 :'opnieuw laten beginnen
260 PUT 7 :'cursor weg
270 RET
9998
9999 'hoofdprogramma
1000 OPEN #0,0:OPEN #5,5
1020 ? "Je antwoord is: ";
1040 GOSUB 100
1060 ?:"Je antwoord was: ";in$
1080 CLOSE #5
9990 END
```

Henk Hiddink  
Ereprijs 53  
1273 XJ Huizen

STANDAARD INPUTROUTINE IN BASIC.

Het nut van deze routine is dat men op een standaard manier om een invoer kan vragen, met een aantal vaste foutmeldingen als er iets onjuist ingevoerd wordt. Doordat er behalve een antwoord ook zogenaamde speciale functies kunnen worden ingevoerd, kan men een programma wezenlijk gebruikersvriendelijker maken.

In het voorbeeldprogramma dat aan de inputroutine voorafgaat staan een aantal suggesties van speciale functies. Als de routine afgewerkt is kan afhankelijk van de waarde van Z8 de loop van het programma beïnvloed worden. De waarde van de input die op dat moment nog in Z\* of in Z kan dan gecopieerd worden naar de variabele waarin hij bewaard moet worden.

Ik gebruik deze speciale functies voor de volgende doeleinden:

- ↑ terug naar de vorige input.
- CAN breek de keuze waarin je zit af.
- END beëindig het programma normaal. (dus zonder STOP)
- SPR ga naar slot van de huidige routine.
- TOE geef een toelichting op de gestelde vraag.

Uiteraard hoeft men al deze functies niet te gebruiken. Ik houd mij aanbevelen voor wijzigingen en/of verbeteringen, bv. deze routine in assembler.

```

8000 REM INPUTROUTINE
8010 Z1=VAL(LEFT$(Z1$,2)):Z2=VAL(MID$(Z1$,3,2)):Z3=VAL(MID$(Z1$,5,1)):Z4=VAL(MID$(Z1$,6,1)):Z5=VAL(MID$(Z1$,7,2)):Z6=VAL(MID$(Z1$,9,2)):Z7=Z6
8020 PUT 22,Z1,Z2:FOR Z9=1TOZ7:PUT 25:NEXT Z9
8030 IF Z4=1 THEN ?Z4:PUT 22,Z1,Z2
8040 LINPUT("") Z1$:ZF=0:Z=0:Z8=0:IF NUM(Z1$) LET Z=VAL(Z1$)
8050 LET Z7=LEN(Z1$):IF Z1$=""?THEN 8140
8060 IF Z7>0 LET Z8=INT((INSTR(Z2$,Z1$)+2)/3)
8070 IF Z8<>0 AND MID$(Z3$,Z8,1)=""1" THEN LET ZF=1
8080 IF Z8=1 AND ZF=0 THEN Z1$=Z:PUT 22,Z1,Z2:ZF=Z8
8090 IF Z8<>0 GOTO 8130
8100 IF Z7<Z5 LET ZF=2
8110 IF Z7>Z6 LET ZF=3
8120 IF NOT NUM(Z1$) AND Z3=1 AND Z7>0 LET ZF=4
8130 IF ZF=0 LET Z$=Z1$:RETURN
8140 PUT 22,1,25,30:?"MELDING : ";
8150 IF ZF=1 ?"INPUT NIET TOEGESTAAN.";
8160 IF ZF=2 ?"TE WEINIG TEKENS INGEGEVEN.";
8170 IFZF=3?"TE VEEL TEKENS INGEGEVEN.";
8180 IF ZF=4 ?"INPUT IS NUMERIEK.";
8190 IF ZF<>0 FORZ9=1TO1000:NEXTZ9:PUT30:GOTO 8020
8200 FOR Z9=1 TO INT(LEN(Z2$)/3)
8210 IF MID$(Z3$,Z9,1)=""0" THEN ? MID$(Z2$,Z9*3-2,3);" ";
8220 NEXT Z9:FORZ9=1TO1000:NEXTZ9:PUT30:GOTO 8020
8300 REM L.G.E. LAPROI
8310 REM ZUIDERZEESTRATWEG 133
8320 REM 8096 ZG OLDEBROEK
8330 REM TEL. 05253-1286

```

Listing "INPUT.BAS"

```

10 PUT 22,1,8:?"GEEF EEN GETAL VAN DRIE OF MEER CIJFERS."
20 Z2$="" CANENDSPRTOE":Z3$=""11011"
30 Z$=""2345"
40 Z1$=""4508110308"
50 GOSUB 8000
60 ?Z$,Z,Z8,ZF
70 END
7000 REM*****
7010 REM* OVERZICHT VARIABELEN *
7020 REM* VAN DE INPUTROUTINE *
7030 REM* *
7040 REM* Z : INPUT NUMERIEK *
7050 REM* *
7052 REM* Z1: POSITIE *
7054 REM* *
7056 REM* Z2: REGELNUMMER *
7058 REM* *
7060 REM* Z3: 0=ALFA NUMERIEK *
7070 REM* 1=NUMERIEK *
7080 REM* *
7090 REM* Z4: PROMPTING *
7100 REM* 0=NEE 1=JA *
7110 REM* *
7120 REM* Z5: MINIMAAL AANTAL POSITIES *
7130 REM* *
7140 REM* Z6: MAXIMAAL AANTAL POSITIES *
7150 REM* *
7160 REM* Z7: LENGTE VAN DE INPUT *
7170 REM* *
7180 REM* Z8: NUMMER SPEC FUNCTIE *
7190 REM* *
7200 REM* Z9: TELLER ALGEMEEN *
7210 REM* *
7220 REM* Z$: PROMPT/INPUT *
7230 REM* *
7240 REM* Z1$: POS 1-2 POSITIE *
7260 REM* POS 3-4 REGELNUMMER *
7270 REM* POS 5 ALFA/NUMERIEK *
7280 REM* POS 6 PROMPTING *
7290 REM* POS 7-8 MIN AANT TEKENS *
7300 REM* POS 9-10 MAX AANT TEKENS*
7310 REM* *
7320 REM* Z2$:SPECIALE FUNCTIES *
7330 REM* *
7340 REM* Z3$:TOEGESTANE FUNCTIES *
7350 REM* *
7360 REM* ZF :NUMMER VAN DE FOUT *
7370 REM* *
7380 REM* Z1$:HULPVARIABLE INPUT *
7390 REM*****
7400 REM

```

Kleine tip.

```
----THE QUICK BROWN FOX JUMPS OVER THE L
LAZY DOG----
```

```
Vind je dit ook zo lelijk op je scherm ?
1234567890123456789012345678901234567890
123456789012345678901234567890123456789
01234 enz
```

En dit ook ? Heb je niet liever dit:

```
----THE QUICK BROWN FOX JUMPS OVER THE
LAZY DOG----
```

```
en dit:
1234567890123456789012345678901234567890
1234567890123456789012345678901234567890
1234 enz ?
```

```
Programmer dan niet
PRINT "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG"
maar:
PRINT "THE QUICK BROWN FOX JUMPS OVER THE","LAZY DOG"

respectievelijk:
PRINT "1234567890123456789012345678901234567890",
"1234567890123456789012345678901234567890","1234 enz".
```

Dit werkt alleen op stream Ø !!!!!

Wim Luijt

# New Brain



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

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

### Enkele technische gegevens die het weten waard zijn:

De NEW BRAIN micro-computer:

- is uitgerust met een zgn. ANSI BASIC compiler: een standaard programmeertaal;
- heeft een volledig geïntegreerd schrijfmachine-toetsenbord van normale afmetingen;
- hij is standaard uitgerust met een 32k RAM geheugen dat echter met inplugbare modules is op te voeren tot twee megabytes;
- hij beschikt over aansluitmogelijkheden voor twee cassetterecorders, TV, monitor, printer en modem;

- biedt de mogelijkheid om andere software-pakketten te laden vanaf cassette of onder CP/M vanaf schijf;
- beschikt voorts over 80 tekens maal 30 regels video;
- eveneens biedt hij de mogelijkheid voor grafische toepassingen (zgn. high resolution graphics);
- alle interfaces en uitgangen zijn standaard;
- bescheiden afmetingen: 28 x 15 x 5 cm;
- is ook te leveren met nikkel-cadmium batterijen voor automatische beveiliging tegen kortdurende stroomuitval!

Nadere Inlichtingen worden u desgewenst graag verstrekt door:

Prijzen excl. BTW



Tradecom International B.V.  
Hondsdijk 3,  
Postbus 60,  
2396 ZH Koudekerk a.d. Rijn  
Tel. 01714-43 00  
Telex 39466

## VERVOLG NEWBRAIN TIPS

## NEWBRAIN BASIC

In On-line No.1 (p. 30) beschreef ik hoe IF-statements soms vervangen kunnen worden door Boolean uitdrukkingen. Ter herinnering nog even een voorbeeld:

```
200 IF A=10 THEN X=X+A
```

kan worden vervangen door:

```
200 X = X + A*(A=10)^2
```

Dit soort opdrachten is elegant, maar de benodigde tijdsduur is iets langer dan die van een IF-statement. Waar het op snelheid aankomt gebruikte men dus IF (of machinetaal).

## HET TOETSENBORD

Helaas blijkt het noodzakelijk iets over het toetsenbord te zeggen. In tegenstelling tot de bewering van F.Kramer in het HCC Bulletin kan men alle tekens uit de karakterset direct via het toetsenbord op het scherm krijgen, ook de u umlaut (ü).

Een paar voorbeelden:

```
~ (tilde) d.m.v. GRAPHICS/?
! chr$(124) d.m.v. CONTROL/.
_ chr$(95) d.m.v. SHIFT/VIDEO TEXT
ü chr$(223) d.m.v. ATTRIBUTE ON (CONTROL 0) en SHIFT/VIDEO TEXT
` accent grave d.m.v. GRAPHICS/:
```

## WERKEN MET SLECHTS EEN DISKDRIVE

Copieren met DDT.COM (On-line 1, p.82 ff.) kan alleen als het te copieren bestand of programma niet groter is dan 56k (wat maximaal in het geheugen - TPA -past). Voor grotere bestanden is men toch weer aangewezen op SDCOPY.COM.

## DEVICES 12-33 (EXPANSION INTERFACE)

## Device 12 en 13:

Bij device 13 wordt een Newline (control code 13) ALTIJD gevolgd door een Carriage Return (control code 10). Let erop dat bij het inlezen van een Device 13 bestand de Carriage Return niet gezien wordt! De combinatie 13+10 wordt ingelezen als alleen 13, dus één byte i.p.v. twee.

Stel de file "TEST" bestaat uit de letter A + newline + CR.

```
10 OPEN#13,13,"TEST":FOR I=1 TO 4:GET#13,A:IF A=26 END
```

```
20 ?A;";":NEXT I
```

```
RUN
```

```
65 , 13
```

```
10 OPEN#12,12,"TEST":FOR I=1 TO 1000:GET#12,A:IF A=26 END
```

```
20 ?A;";":NEXT I
```

```
RUN
```

```
65 , 13 , 10
```

Nog een waarschuwing:

Er ontbreekt een foutmelding voor het onjuist gebruik van de devices 12-14.

Voor printen naar deze devices moet men de stream openen met OPENDOUT#X, niet met OPEN#X.

Als de file nog niet bestaat, geeft de NewBrain een foutmelding, maar als de file wel bestaat, niet. Printopdrachten worden echter niet uitgevoerd als OUT is weggelaten.

Dit is fout:

```
OPEN#13,13,"BANDUNG":PRINT#13,"ABCDE":CLOSE#13
```

Juist is natuurlijk OPENDOUT#13 enz. Het ontbreken van een foutmelding is misleidend.

## Device 14

Zoals al gezegd in On-line No. 1 (p.64 ff.) is het wel degelijk mogelijk om Newline karakters te gebruiken. In de praktijk vind ik het het handigst om Device 14 en 13 onderling uitwisselbaar te maken.

Printen naar device 14 doe ik als volgt:

```
?#14,a$:put#14,10
```

Met deze opdracht wordt het formaat precies gelijk aan dat van device 13, namelijk elke string wordt gevolgd door een Newline en een Carriage Return karakter.

In een bestand met records van gelijke lengte L geeft men als aantal weg te schrijven bytes op L+2 (lengte van de string + NL + CR), maar (let op!) als aantal in te lezen bytes per string slechts L+1. (Zie verder On-line No.1)

Dit is handiger dan de methode die ik in On-line 1 had beschreven. Door weglating van de opdracht PUT#14,10 wordt het formaat van de file gelijk aan dat van Device 12.

In mijn database-programma "MAX" heb ik uitwisselbaarheid met Device 13 vooropgesteld, wat verschillende voordelen heeft (o.m. kan het hele bestand met het CP/M type commando op het scherm gebracht worden).

## Devices 16 en 17

Het blijkt dat sommige printers nogal specifieke eisen stellen aan de opgegeven parameters. Bovendien kan men niet ongestraft elke gewenste buffergrootte opgeven.

Hier volgen mijn ervaringen met de STAR DP 510 printer.

In de parameterstring moet altijd staan: S2 (2 stopbits).

Voor listings moet er staat L80 (of een andere waarde) maar nooit L0.

Er moet altijd een buffer zijn (\*1 tot \*12)

De maximale grootte van de buffer is \*12 (anders foutmelding "Buffer already in use" bij herhaald openen van dit device).

Bij het opzetten van een configuratie met een andere baudrate dan de standaardwaarde moet Device 17 dezelfde baudrate hebben, anders werkt het niet (dan wordt namelijk een baudrate van 9600

aangehouden).

Bij het configureren van de streams moet device 17 de parameter #0 meekrijgen.

Voorbeelden:

GOED:

```
1) OPEN#16,16,"S2L0T9600*12":OPEN#17,17,"*0"
2) OPEN#16,16,"S2L0T1200*12":OPEN#17,17,"T1200*0"
```

FOUT is:

```
3) OPEN#16,16,"S2L0T3600*12":OPEN#17,17,"*0" (Baudrate blijft 9600)
4) OPEN#16,16,"S2L0T3600":OPEN#17,17,"T3600" (Device 17 heeft default buffer van 256 bytes)
```

Voor tekstverwerking kan met het beste de parameter L0 gebruiken. De Control Code 9 (TAB) wordt bij andere waarden (L80, L40 enz.) als een serie spaties naar de printer gestuurd. Bij gebruik van L0 voorkom je fouten die daardoor zouden kunnen ontstaan.

Device 33 (en device 11)

Het is mogelijk om eerst te plotten naar device 12 (disk) en daarna met GET van de schijf in te lezen en naar een grafisch scherm te sturen. Hele grafische schermen kunnen aldus op disk opgeslagen worden.

```
100 OPEN#12,12,"GRAFIEK"
110 PLOT#12,background,fill,move enz. enz.
120 CLOSE#12
130 OPEN#12,12,"GRAFIEK"
140 OPEN#11,11,"WI00"
150 FOR I=1 TO 10000
160 GET#12,A
170 PUT#11,A
180 NEXT I
1000 END
```

In feite is het ietsje ingewikkelder: Je moet het aantal bytes dat naar #12 weggeschreven wordt precies bijhouden (zoek zelf maar uit hoeveel bytes elke plotopdracht lang is) om datzelfde aantal daarna weer in te lezen. De eerste byte van elke plot is een Control Code 16 karakter (Graphics Escape). Veel succes!

EEN BUG

Als men na gebruik in BASIC van één van de devices 16,17,19, 20 of 21 met buffer naar CP/M wil gaan, kan het zijn dat het toetsenbord niet meer reageert en/of het scherm weg is. Zoals beschreven in on-line 1 helpt het vaak om eerst een stream naar device 15 te openen. Nog beter is, om eerst de gebufferde stream te sluiten en daarna opnieuw te openen zonder buffer (met #0).

Een voorbeeld:

```
50 OPEN#16,16,"S2L0*8"
60 GOSUB 2000:REM PRINTROUTINE
70 CLOSE#16
80 OPEN#16,16,"*0":REM OPEN STREAM NU ZONDER BUFFER (#0)
90 OPEN#15,15:?*15,6:REM RESET CP/M
```

100 EXIT:REM GA NAAR CP/M TERWIJL #15 OPEN IS.

Helaas werkt ook deze truc niet altijd.

PASCAL

De firma Grey Matter in London adverteert in Personal Computer World met JRT Pascal. Volgens Grey Matter is JRT weer "back in business".

Een andere Pascalcompiler is Turbo Pascal, die in de V.S. voor \$ 50.00 aangeboden wordt (bespreking en advertentie in Microsystems). Deze compiler zou veel sneller zijn dan de concurrenten.

STORINGEN

De NewBrain met Expansion Interface Module is erg gevoelig voor storingen van het lichtnet. Bij het aanslaan van een koelkast kan de computer op tilt gaan. Er zijn echter storingsonderdrukkers in de handel (bestemd voor HiFi apparatuur) die deze ellende voorkomen. Ik heb veel plezier van de 750 W Mains Interference Suppressor van QED (o.m. te koop bij Art Fidelity in Amsterdam). Bovendien gaf mijn netvoeding (de zgn. Footbreaker) te veel spanning af, nml. 6.5 Volt ipv. 5 Volt. De spanningsstabilisator in de computer zet het teveel om in warmte, waardoor bij een dergelijk groot verschil het apparaat nogal heet wordt. Als dat probleem U parten speelt kunt U het verhelpen door een uitgangsweerstandje in de Footbreaker te vervangen door een met de juiste waarde (even meten en de Wet van Ohm hanteren).

Een ander probleem doet zich voor met de disk controller. Soms is het plotseling niet meer mogelijk om naar CP/M te gaan of om een programma van schijf in te laden. De computer meldt dan "No response from Disk Controller". Vermoedelijk zijn gecorrodeerde contacten de oorzaak - bij mij blijkt het probleem tenminste te verhelpen door het bespuiten van de contacten met kanaalkiezerspray (Tuner 600 van Kontakt Chemie). Deze spray laat geen residu achter en is veilig. Nooit gewone contactspray gebruiken!!

Natuurlijk is dat slechts een noodoplossing - dergelijke gevoelige contacten behoren verguld te zijn. Om de kosten hoeft Tradecom dat niet te laten, want de benodigde hoeveelheid goud is minimaal.

Rob van Albada

## EEN SUBROUTINE VOOR RANDOMFILES.

In het Handbook for Disk Users staat dat je alleen naar een random file kunt schrijven met PRINT #n,d\$, .die punt-komma is ervoor dat er geen newline naar de file wordt geschreven. Dit is lastig als je de informatie weer wilt lezen, je moet dan met een GET werken. Als je de punt-komma vergeet moet je er om denken dat je behalve de informatie ook een newline mee stuurt. De newline die nu ook op de schijf staat maakt het mogelijk dat de informatie te lezen is met een (L)INPUT #n, je leest dan de gegevens + de newline

De subroutine die ik er voor heb geschreven spaart een hoop rekenwerk uit. De te lezen of te schrijven string staat in Y\$. De parameters staan in YP\$. Door deze subroutine te gebruiken wordt het programma overzichtelijker omdat alle I/O in een subroutine staat. Het is mogelijk deze routine te gebruiken voor meerdere files tegelijk

Ik hoop dat dit artikel en de subroutine zullen bijdragen tot een groter begrip van de random files in het NewBrain CP/M systeem en dat het de interesse opwekt om de routine te vervolmaken

L.G.E. Laspri

```

9000 REM ----- LEES/SCHRIJFRoutine RANDOMFILES 0-----
9010 REM FILE MOET AL GEOPEND ZIJN
9020 Y0=VAL(LEFT$(YP$,1)):Y1=VAL(MID$(YP$,2,4)):
    Y2=VAL(MID$(YP$,6,4)):Y3=VAL(MID$(YP$,10,4)):
    Y4=VAL(MID$(YP$,14,4)):Y5=VAL(MID$(YP$,18,2))
9030 REM PLAATS IN DE FILE = RECORDGROTE * RECNO + PLAATS
    IN REC
9040 REM Y6 = Y1 + Y2
9050 Y6=INT(Y4*Y1+Y2)/256)
9060 Y7=(Y4*Y1+Y2)-256*Y6
9070 Y8=INT(Y3/256)
9080 Y9=Y3-256*Y8
9090 PUT #YS,Y6,Y7,Y8,Y9
9100 IF Y0=1 THEN PRINT#YS,Y$
9110 IF Y0=2 THEN LINPUT#YS,Y$
9120 RET
9130 REM *****
9140 REM ** Y0 INPUT / OUTPUT **
9150 REM ** **
9160 REM ** Y1 RECORDNUMMER **
9170 REM ** **
9180 REM ** Y2 POSITIE IN HET RECORD **
9190 REM ** **
9200 REM ** Y3 LENGTE VAN HET VELD **
9210 REM ** **
9220 REM ** Y4 RECORDGROTE **
9230 REM ** **
9240 REM ** Y5 KARAKTERNUMMER **
9250 REM ** **
9260 REM ** Y6 PLAATS OP SCHIJF **
9270 REM ** **
9280 REM ** Y7 IDEM **
9290 REM ** **
9300 REM ** Y8 LENGTE VAN DE STRING **
9310 REM ** **
9320 REM ** Y9 PLAATS OP SCHIJF **
9330 REM ** **
9340 REM ** Y$ VELD **
9350 REM ** **
9360 REM ** YP$ PARAMETERSTRING **
9370 REM ** **
9380 REM *****

```

## Technische aspecten van de Expansion Interface. Deel 1

De belangrijkste verandering die de expansion interface teweeg brengt is het paged operating system dat dan in werking treedt. Voor de eindgebruiker (BASIC bijv.) van de NewBrain is er niet zoveel veranderd, behalve dan dat er meer geheugencapaciteit inszit en uteraard het feit dat er een onafhankelijke printer en communication device is, enz. Het paged operating system echter, is wel degelijk een merkbare verandering voor mensen die in het scherm willen poken of met machinetaal bezig zijn. Dit verhaal is bedoeld om die mensen te helpen. Behandeld zullen worden o.a. paginering, de bijbehorende systeemadministratie, het Memory Management System (MMS), schermen en hoe die in het Z80 geheugen te stoppen en een aantal belangrijke systeemvariabelen of -parameters.

Dit verhaal is het resultaat van het spelen met de expansion interface en het lezen van het begeleidende NewBrain paged memory operating system description. Er zij opgemerkt dat het verhaal niet compleet is, er valt nog meer te vertellen over het paged operating system, maar dat gaat nogal diep. Misschien iets om later op terug te komen.

### Paginering.

Het systeem is zeer flexibel opgebouwd, net zoals de hele NewBrain zelf trouwens, want je kunt elke RAM page van 8K in ieder gewenst adresgebied plaatsen. Bijv. pagennr. 100 (welke een gedefinieerd fysisch stukje RAM voorstelt) kun je in het gebied 0000-1FFF plaatsen, maar ook in A000-BFFF. Hoe wordt nu een page in het gewenste RAM gebiedje gehaald?

Hiertoe is de hardware voorzien van een aantal registers, die bereikt kunnen worden via de Z80 I/O poort nr. 2. Op deze plaats moeten we nu goed zien hoe en wat er gebeurt. Er moeten nl. 12 bits weggestuurd worden, terwijl de Z80 I/O poorten slechts 8 bit breed zijn. Gelukkig bezit de Z80 een snuffje waarvan nuttig gebruik gemaakt wordt. Als we nl. input of output plegen met het C register als I/O poort nr. (de IN (C),r en OUT (C),r instructies) dan wordt de inhoud van het E register op de adreslijnen A8 t/m/ A15 gezet. We kunnen dus totaal 16 bits naar de buitenwereld sturen. Dit werkt niet bij de gewone IN (n),A of OUT (n),A instructies. Dus deze instructies mogen we niet gebruiken bij onze page operaties.

Nu iets over de betekenis van de pageregisters. Het pagenummer wordt aangegeven door de waarde op de databus, m.u.v. E7 welke niet gebruikt wordt. In plaats hiervan gebruikt men A11. Het pagenummer wordt cijfermatig gevormd als volgt (meest naar minst significant)

A11, A10, A9, A8, D7, D6, D5, D4, D3, D2, D1, D0.

A10 t/m D7 worden niet gebruikt en worden beschouwd 0 te zijn. Op deze manier komen we aan pagenummers zoals die gebruikt worden in NewBrain documentatie. De getallen 0 t/m 127 stellen RAM pages van 8K voor, totaal dus 1 Mbyte. Als A11 een "1" is dan zijn het de getallen 2048 t/m 2175. Dit zijn ROM pages. ROM's zullen in de meeste gevallen Z80 programma's bevatten. Aangezien Z80 code niet reloceerbaar is, moet een ROM page perse in een bepaald vantevoren aangegeven slot zitten. Hiertoe wordt in het geval van ROM pages ook vast aangegeven in welk slot hij hoort. Dit gebeurt d.m.v. de meest significante 3 bits, t.w. A15, A14 en A13. Een en ander zullen we illustreren aan de hand van voorbeelden.

\* Op de locaties B0 t/m B5 vinden de de pagenummers van de AB, CD en EF ROM (zie system page description op pagina 28 e.v. van voornoemde beschrijving). De meest significante 3 bits van het AB ROM pagenummer zullen het patroon 101 geven omdat dit overeenkomt met slot 5 (A000-BFFF). De andere bits beschrijven het pagenummer zodanig, dat bij aantrengen van de page de hardware de page ook inderdaad kan "vinden". Dus om de ROM in het Z80 geheugen te krijgen volstaat het om de twee bytes die het pagenummer vormen via de OUT (C),r instructie weg te sturen.

\* Op de locaties A1/A2 vinden we het pagenummer van de eerste videopage. Hier gaat het om een RAM page. De 3 meest significante bits zijn hier 0, evenals bit 11, welke tenslotte aangeeft of het om een ROM, dan wel een RAM page (nummer) gaat. Hier moet men dus zelf nog aangeven in welk slot de page moet komen door de 3 meest significante bits te zetten op de gewenste waarde voordat het gehele 16 bit woord (pagenummer) weggestuurd wordt naar de hardware (dus via poort 2 met de OUT instructie).

### Systeemadministratie.

Nadat dit duidelijker is geworden gaan we het wat gecompliceerder en geavanceerder maken. Stel dat u in een subroutine een bepaalde page nodig hebt, terwijl u bij terugkeer uit de subroutine weer dezelfde pages weer terug wilt hebben als voordat u de subroutine betrad. Het mooiste kan dit gedaan worden op dezelfde manier als het gebeurt met de subroutine zelf, nl. het onthouden van het terugkeeradres op de stack. In dit geval dus het onthouden van het pagenummer in het slot waarin u een andere page wilt brengen om het dan vervolgens weer van de stack terug te halen (aan het eind van de subroutine) zodat de page weer in het slot gebracht kan worden. Om dit systeem mogelijk te maken is er een schaduwadministratie nodig van de pagenummers in ieder slot, dus een kopie van de inhoud van de hardware page registers.

Deze kopieën bevinden zich voor slot 0 t/m slot 7 in de locaties 8B t/m 9A (16 bytes). Een uitgewerkt voorbeeld zal benut worden om de werking van de schaduwadministratie, het save en terughalen van een pagenummer etc. te illustreren.

```

SUBROUT LD BC,(0080H) ;zet AB ROM pagenr. in BC
LD HL,(0095H) ;zet slot 5 pagenr. in HL
PUSH HL ;save dit pagenummer
LD (0095H),BC ;nieuw pagenummer in schaduwadministratie
LD L,C
LD C,2 ;poort 2
OUT (C),L ;output BL op adres- en databus

```

nu rest van de subroutine, al dan niet met nieuwe CALL's erin

```

TERUG POP HL ;haal oud pagenummer terug
LD B,H
LD C,2 ;poort 2
OUT (C),L ;output BL, dus oude page weer terug
RET

```

Hierbij wil ik het voor wat betreft de paginering het erbij laten. Tot slot alleen 2 opmerkingen:

- Bovengenoemd voorbeeld wordt natuurlijk iets ingewikkelder als een nieuwe RAM page ingebracht moet worden, want dan moeten voor het outputten en in de schaduwadministratie stoppen eerst nog een aantal bits in het B register van de Z80 de juiste slot informatie bevatten (met OR instructie bijv.).

- Bit 12 van de pagenummer informatie is in het geheel nog niet ter sprake geweest. Dit bit moet 0 zijn. Als het 1 is wordt er nl. geen nieuwe page ingebracht, maar dan wordt een pageregister geladen van de zgn. alternate slot set. Deze alternate slot set wordt (in zijn geheel, dus 8 slots) ingebracht als adreslijn A16 actief wordt. A16 wordt bepaald door bit 2 van poort 255. Deze poort is het load pageing status register. De betekenis van de bits is als volgt:

D0: 1 = enable pageing elektronica

D1: niet gebruikt

D2: 1 = A16 wordt 1, alternate slot set ingeschakeld

D3: 0 = selecteer multiprocessing mode.

D4: 1 = isoleer deze NewBrain (gebruikt in multiprocessing mode)

De alternate slot set wordt gebruikt door interrupts van het systeem. Het is dan ook niet aan te bevelen om er zelf gebruik van te maken.

Een ander belangrijk punt dat hoort bij systeemadministratie is het gebruik van zero page locaties als opslag voor belangrijke systeemvariabelen. Daar dit een beschouwing op zich waard is, verwijzen

we hiervoor naar deel 2 onder de kop systeemvariabelen en -tabellen. 1 ding is echter alvast interessant om te vermelden.

We kennen bij de NewBrain de structuur van een enkele entry voor operating system routines en F.P. berekeningen, waarbij d.m.v. een byte wordt aangegeven welke routine je wilt hebben. Deze set van routines is bij het expanded system veel uitgebreider en beslaat in totaal 24 Kbyte (inclusief de bestaande 8 K). Het leuke is, dat in de NewBrain wordt bijgehouden in welke ROM een bepaalde entry zich bevindt, waardoor het systeem bij aanroep van een bepaalde routine de juiste ROM eerst in slot 7 stopt en vervolgens de routine afwerkt. De routine die dit bestuurt zit uiteraard niet in slot 7, want dan zou hij zichzelf mollen, maar zit in RAM in slot 0, de van systempage. In dit slot 0 zit ontzettend veel informatie, daarover ook meer in deel 2.

### Het Memory Management System.

MMS staat beschreven op blz. 18 e.v. van het paged system description. Gezien het grote nut dat MMS kan hebben voor systeempruivers onder de Newbrain gebruikers, geef ik hier een beknopte beschrijving in eigen bewoordingen. Tevens zal hierbij het belang van slot 0 tevoorschijn komen als onmisbaar bestand.

De naam OBJECT is een essentieel begrip bij het MMS. Dit is zeer goed te vergelijken met het begrip FILE dat we van disk kennen. Dus een sequentiële datastructuur hebbende een naam, een fysisch ligingsgebied, en een lengte. Bij MMS is de naam van een OBJECT een getal tussen 1 en 65535 (2 bytes), de ligging kan overal in de beschikbare RAM zijn en de lengte is tussen 0 en 65520 bytes (maximum).

Wij zouden als toepassing bijv. kunnen denken aan het kopiëren van alle benodigde files van een bestand bijv. in OBJECTS op simpele en snelle wijze (Disk I/O 6 Kbyte per seconde), zodat alle mutaties en zoekacties op efficiënte en disk besparende wijze plaats kunnen vinden. Na het werken ermee kunnen op simpele wijze de gebruikte OBJECTS weer naar disk beschreven worden.

Hoe lezen we een OBJECT vooropgesteld dat er een is?

Dit gebeurt door de routine GETOB aan te roepen (CALL 007EH). In HL moet de OBJECTnaam (nummer) meegegeven worden. De routine keert terug met het volgende resultaat. Slot 1 en 2 gaan verloren, hier komen de pages waarbinnen zich de eerste 8 Kbyte van het gewenste OBJECT bevindt. De routine keert terug met in HL het beginadres (in slot 1) van het OBJECT. Waarom keert dit ons nu 2 slots, terwijl 8 K van het OBJECT nu beschikbaar zijn? Dat komt doordat een OBJECT niet noodzakelijkerwijs aan het begin van een page hoeft te zitten. In 1 page kunnen bijv. 2 OBJECTS zitten. Hierdoor wordt geen ruimte in het geheugen onbenut gelaten. Die twee verloren gegane pages garanderen ons dat altijd



minstens 8 K van het OBJECT leesbaar zijn. Als we na inspectie van het OBJECT de oorspronkelijke pagenummers weer terug willen hebben in slot 1 en 2, dan hadden we de pagenummers uit de schaduwadministratie moeten saven op de stack, om ze dan later weer terug te zetten (zie onder systeemadministratie).

Wat nu als het OBJECT heel lang is. Bijv. 40 Kbyte. Hoe bereiken we dan een OBJECTdeel verderop? Dit kan door een Random Access mechanisme, net zoals we dat van device 14 (Random Access filing op disk) kennen. Hier gaat het als volgt: Men roepe de routine IGETOB aan (CALL 0081h) onder dezelfde condities als GETOB, echter met een extra parameter, en wel een offset in het OBJECT gerekend vanaf het begin van het OBJECT in DE. Bijv. als we het 38000 ste byte in het OBJECT willen hebben dan geven we DE die waarde mee. Bij terugkeer zijn die pages in slot 1 en 2 gebracht waarvan geldt dat het gewenste 38000 ste byte in slot 1 staat. Het wordt aangewezen door HL, m.a.w. HL wijst naar het begin van het gewenste deel van het OBJECT en de 8 K daarna zijn dan ook leesbaar.

Als het goed is zijn nu de principes van het gebruik van MMS naar voren gebracht. Voor het daadwerkelijk gebruik ervan is het aan te raden blz. 18 en 19 van het paged system description over MMS te lezen. Daar staat ook in hoe je een OBJECT creeert (NEWOB) en hoe je er een weghaalt (DELOB). Al deze routines bevinden zich in slot 0 omdat het routines zijn die de zgn. slot allocation veranderen. Dergelijke routines mogen uiteraard niet in de slots voorkomen waar andere pages inkomen want dan krijgt de processor ter plekke van het programcounteradres andere koek voorgeschoteld die hij vast niet lust. Dit is meteen een nadeel van het paged operating system. Je moet enorm oppassen met wat je doet. Dit zal overduidelijk blijken bij de behandeling van het video gebeuren onder het paged operating system.

Rob Maris

Deel 2 handelt over video, systeemvariabelen en een oordeel over de expansion interface en verschijnt in de volgende NewBrain on line

## MAX database

Met een database programma kunt U gegevens op een floppy disk opslaan: adressen, recepten, woordenlijsten, statistieken enz. Een gegevensbestand bestaat uit een aantal velden, in een adressen bestand bijvoorbeeld Naam, Adres, Woonplaats, Telefoon, Beroep, Bijzonderheden.

MAX is geschikt voor de opslag van zowel strings (alfabetische gegevens zoals naam, adres en dergelijke) als getallen.

String bestanden worden op alfabetische volgorde bewaard - U kunt het sorteerveld uiteraard zelf kiezen.

Numerieke gegevens worden op grootte gesorteerd opgeslagen.

MAX is geschreven door Rob van Albada voor de 96k NewBrain met CP/M en wordt gebruikt voor de NBGG ledenadministratie. Dit doet MAX voor U:

-Maakt bestanden tot 8 Megabyte maximaal;			
-Sorteert sneller dan elk ander NewBrain BASIC programma:			
	MAX GEMINI (Cassette)	dBaseII	
275 records	15	80	3 min.
sorteren			
taal	BASIC BASIC		Machinetaal
prijs fl.	100	200	1500
theor.capaciteit	8Mb	24k	8Mb
pract.capaciteit	200k	24k	???

(De praktische capaciteit, d.w.z. de maximale omvang van een bestand, is kleiner dan de theoretische waarde van 8 Megabyte omdat bij zeer grote bestanden de sorteertijd te lang wordt.)

-Kan selectief printen, bijvoorbeeld uitsluitend alle Jansens die NIET in Elst wonen, of alle vrouwen die langer zijn dan 175 cm.

-Kan labels printen d.m.v. speciaal overlayprogramma MAXLAB. Een naam ingevoerd als: Heslinga,Dr. Sicco wordt afgedrukt als Dr. Sicco Heslinga. De gewone printoverlay gebruikt een eenvoudig afbreeksysteem.

-MAX is aangepast aan de STAR printer (EPSON codes) maar kan door de gebruiker gemakkelijk aan andere printers aangepast worden. Desgewenst kunt U adressen krijgen van andere MAX bezitters met dezelfde printer. D.m.v. het hulpprogramma MAX-PATCH kunt U zelfs alle non standard characters van Uw printer gebruiken (bijv. de internationale karaktersets).

-Maximaal 80 velden mogelijk.

-MAX houdt automatisch eenvoudige statistiek bij van de numerieke velden: som, gemiddelde, deviantie, standaard afwijking.

-Opzoeken van gegevens: (a) op sorteerveld d.m.v. Binary Search; (b) op nummer; (c) op willekeurig veld (sequentieel).

-Te koop bij de schrijver op de NewBrain dag in De Bron (26 mei).

## BASICODE

(bron: NOS Basicode)

Sinds 1978 zendt de NOS in het radioprogramma HOBBIYSCOOP computerprogramma's uit. In het begin voor 4 verschillende merken computers: Apple, Exidy Sorcerer, PET/IBM en TRS-80. Dit gebeurde in de eigen taal van die computers, zodat ieder merk eens per maand aan de beurt kwam.

Al snel bleek dat ook bezitters van andere merken computers programma's via de radio wilden opnemen. Men is toen gaan zoeken naar een manier om programma's zo uit te zenden dat ze op verschillende merken computers konden draaien.

Hierbij doen zich twee problemen voor. Op de eerste plaats heeft ieder merk computer een andere methode voor de data-overdracht. Dat wil zeggen, de toontjes die de computer op de band zet zijn voor iedere merk verschillend. Een NewBrain computer begrijpt niets van een bandje dat op een Apple is gemaakt en omgekeerd. Het tweede probleem is de taal. Hoewel wordt uitgegaan van Basic blijkt die taal op de verschillende merken zoveel verschillen te vertonen dat een programma dat op een bepaald merk computer is geschreven niet kan draaien op een ander merk.

Onze eigen NewBrain geeft daar fraaie voorbeelden van. Een IBM zal niet begrijpen wat wij bedoelen met PUT 31 of PRINT #8. Evenzo zal een TRS-80 geen raad weten met OPEN #5,5 of PUT 22,1,0.

Daar komen nog problemen bij als maximale regelengte, maximale stringlengte, aantal regels en kolommen per scherm, het wel of niet plaatsen van een spatie voor en achter een getal, het omzetten van een getal in een string of het inschakelen van een printer.

Na vele discussies over hoe deze problemen op te lossen kwam Klaas Robers met een briljant idee. Later is hier, in samenwerking met vele computerfanaten en gebruikersgroepen, BASICODE uit ontstaan.

Inmiddels is Basicode al weer verder ontwikkeld en nog universele geworden. De huidige versie heet dan ook Basicode-2.

## HOE WERKT BASICODE?

Elk teken van de programmalisting wordt omgezet in ASCII - code, bestaande uit een reeks enen en nullen. Deze enen en nullen worden dan op een vaste toonhoogte op de band gezet. Iedere '0' wordt weggeschreven als 1 volle periode van 1200 Hz, iedere '1' wordt weggeschreven als 2 volle perioden van 2400 Hz.

Om die tonen te kunnen begrijpen is een speciaal vertaalprogramma nodig, soms met een stukje hardware-uitbreiding (een klein extra printplaatje). De NewBrain heeft zo'n printje niet nodig. Het tweede probleem is opgelost door in Basicode alleen die commando's te gebruiken die elke computer kent. Alle woorden die wel gebruikt moeten worden maar voor elke computer verschillend zijn, (zoals PUT 31 voor het scherm wissen) staan in aparte subroutines die vanuit het Basicodeprogramma worden aangeroepen. Afgesproken is dat die subroutines op de regels 100 tot 1000 staan. Deze programmaregels horen dus niet bij het Basicodepro-

gramma en worden door Hobbyscoop ook niet uitgezonden. Je voegt ze later, via het MERGE-kommando, aan het Basicodeprogramma toe.

## BASICODE OP DE NEWBRAIN

## A. HET LEESPROGRAMMA.

De fa Tradecom heeft 2 programma's ontwikkeld waarmee het mogelijk is om zowel Basicodeprogramma's te lezen als te schrijven. De meeste NewBraingebruikers zullen alleen programma's van de radio willen opnemen om ze op hun NewBrain te laten draaien. Het is een fascinerende bezigheid om computerprogramma's uit de ether te plukken en ze later op je eigen computer te zien werken. Voor al diegenen is alleen het leesprogramma van belang. Omdat een gebruiksaanwijzing tot nu toe ontbreekt volgt hier een stap voor stap handleiding die, mits nauwkeurig gevolgd, moet leiden tot een goed werkend Basicodeprogramma.

1. U moet beschikken over het 'Basicodeleesprogramma', zoals dat is uitgezonden door de NOS. In dit programma is tevens het sub-routinegedeelte opgenomen. Dit gedeelte moet eerst op een aparte cassette worden opgenomen.

Laad het leesprogramma en verwijder regel 1. Verwijder daarna alle programmaregels vanaf regel 600. Het programma dat u nu nog overhoudt zet u onder de naam SUBROUTINES op de nieuwe cassette.

2. Laad het leesprogramma opnieuw. In de laatste regels ziet u in REM-statements een soort aansluitschema. Hieruit blijkt dat de NewBrain via de COMMONS-poort met de cassetterecorder moet worden verbonden, in geval van zwak signaal zelfs met behulp van een extra weerstandje. Gelukkig heeft Rob Maris ontdekt dat het lezen ook gewoon via de TAPE-1 ingang gaat. Daarvoor hoeven slechts 2 getallen in het machinetaalgedeelte veranderd te worden. List de regels 10110 en 10120. Op beide regels ziet u de getallen 230,2 staan. Verander op beide regels de 2 in 32. List nu regel 1000. Daar staat CT=52. Het getal 52 bepaalt de tijdsduur van een periode. In de meeste gevallen is dit teveel en kunt u het beter veranderen in 50 of 49. Enig experimenteren met die CT-waarde zal nodig zijn.

Verwijder alle regels t/m 990. Zet het gewijzigde leesprogramma eveneens op een nieuwe cassette en gebruik deze voortaan als u een Basicodeprogramma wilt gaan inlezen.

Voor degenen die het leesprogramma niet hebben is de volledige listing bij dit artikel opgenomen, zodat u het zelf kunt overtypen. Ook zal het programma via de softwarebibliotheek verkrijgbaar zijn.

3. Nu moet u een Basicodeprogramma hebben. De eenvoudigste manier is er een opnemen van de radio. De NOS zendt elke zondag in het programma Hobbyscoop zo'n programma uit. U dient dit wel met enige zorg op te nemen. Vooral de opnamesterkte is van groot belang. Neem het programma HARD op, de meter moet een heel stuk in het rode gebied staan. Zelf neem ik het programma altijd op met een grote spoelenrecorder zodat ik het daarna in verschillende sterkten kan overnemen op een cassette.

Er zijn nog andere manieren om aan Basicodeprogramma's te komen. De NOS geeft een bandje uit met een groot aantal programma's en de stichting Teachip verkoopt (tegen Kostprijs) programma's die op het onderwijs zijn gericht.

4. Nu zijn we gereed om echt aan het werk te gaan.

Leg de cassette met het leesprogramma klaar, de cassette met de subroutines en een lege cassette. Waar die lege cassette voor is zal u weldra duidelijk worden.

5. Laad het leesprogramma. Na het laden typt u RUN en drukt op NEWLINE. Het programma verwelkomt u met de woorden 'Lees basicode NewBrain release 1.0'. Haal het bandje met het leesprogramma uit de recorder en stop het bandje met het Basicodeprogramma erin. **HAAL HET PLUGJE VAN DE AFSTANDSBEDIENING UIT DE RECORDER!** Aan het begin van ieder Basicode-programma staat een 5 seconden durende toon. Speel de band af (met geluid) tot u het begin hoort van die toon. Stop dan de recorder. Inmiddels staat op het scherm 'Naam weg te schrijven programma:'. Typ de naam van het programma in en druk op NEWLINE. Nu komt er te staan 'Start cassette en geef NEWLINE?'. Deze aanwijzing is duidelijk. Eerst de cassetterecorder starten en direkt daarna op NEWLINE drukken. Ik wijs er nogmaals op dat het plugje van de afstandsbediening uit de recorder gehaald moet zijn, omdat de recordermotor niet automatisch gestart wordt.

6. Na verloop van tijd komen er twee mededelingen op het scherm. De eerste is een melding over de kwaliteit van het inlezen. Er kan o.a. komen te staan 'Checksum klopt', 'Checksum klopt niet' of 'Time out Error'. De tweede boodschap is 'Start cassette en geef Return'. Ook al staat er een foutmelding, dan wil dat nog niet zeggen dat het programma niet goed ingelezen is. Een enkel verkeerd gelezen lettertje heeft al een foutmelding tot gevolg.

7. Draai van de nieuwe cassette de aanloopstrook door en stop de cassette in de recorder. **STEEK HET PLUGJE VAN DE AFSTANDSBEDIENING ER WEER IN!**

Zet de recorder op opname en druk op NEWLINE. U kunt nu zien of het programma goed ingelezen is want, terwijl het naar de band geschreven wordt, kunt u tegelijkertijd meelesen via het scherm.

Indien u begrijpelijke taal ziet kunt u de recorder door laten draaien tot op het scherm de boodschap komt dat het programma is weggeschreven.

8. Ziet u allerlei onzin, rare lettertekens e.d. dan kunt u beter op de STOP-toets drukken. Het inlezen is niet goed gegaan. Hiervoor zijn verschillende redenen aan te wijzen. De recorder kan te hard of te zacht hebben gestaan, de CT-waarde kan verkeerd zijn of het programma kan verkeerd van de radio zijn opgenomen.

Er zal niets anders op zitten dan de procedure te herhalen, met een andere volume-instelling en/of een andere CT-waarde.

9. Indien alles goed is gegaan, met hoogstens hier en daar een enkel foutje, dan kunt u verder gaan. Laad het programma dat u zojuist heeft weggeschreven opnieuw in de computer. Na het laden voegt u met het MERGE-commando de subroutines toe. Als dit alles gelukt is kunt u het programma runnen.

Het is mogelijk dat er tijdens het runnen af en toe een foutmelding komt, omdat er een lettertje verkeerd gelezen is. Er staat bijv. PLINT i.p.v. PRINT. Dergelijke kleine foutjes kunt u nu verbeteren. Als het hele programma feilloos werkt kunt u het definitief op een cassette zetten en een zucht van verlichting slaken dat alles weer goed gelukt is.

Omdat Basicode in principe voor tape bestemd is ben ik daar bij het maken van deze handleiding van uit gegaan. Degenen die met een disc willen werken kunnen deze handleiding uiteraard ook gebruiken. Overal waar gesproken wordt van programma's op tape zetten dienen zij dit te lezen als programma's op disc zetten. Alleen het echte Basicodeprogramma zoals de NOS dat uitzendt moet vanaf tape gelezen worden.

## B. HET SCHRIJFPROGRAMMA

Mocht u denken dat het het inlezen van een Basicodeprogramma in een NewBrain een ingewikkelde zaak is, het is nog niets vergeleken bij het vertalen van een NewBrain programma in een Basicodeprogramma.

Ook hiervoor zal ik een stap voor stap handleiding geven.

1. Overtuig u er eerst van dat uw programma aan alle Basicodevoorwaarden voldoet. Bijv. dat de regels niet meer dan 60 karakters bevatten, dat er geen PUT-opdracht ingesloten is enz. Het NewBrain vertaalprogramma controleert op geen enkele wijze of er fouten tegen het Basicodeprotocol in het programma staan. Dat betekent dat het programma wel vertaald zal worden naar dat het op andere merken computers niet zal werken.

In het Basicodeboek, te bestellen bij de NOS, postbus 1200, 1200 BE HILVERSUM, is het hele Basicodeprotocol opgenomen.

2. Voor het vertalen is het schrijfprogramma nodig, zoals dat door de fa Tradecom is geschreven en door de NOS uitgezonden. Voor degenen die niet over dit programma beschikken is de complete listing bij dit artikel opgenomen. Ook is het programma via de softwarebibliotheek verkrijgbaar.

Helaas is Tradecom er vanuit gegaan dat iedere NewBrainbezitter ook over een discdrive beschikt. Hierdoor zal het programma voor cassettegebruikers niet werken.

Om dit te veranderen dient u regel 400 aan te passen.

Regel 400 luidt: IF ERRNO (<) 151 THEN ? ERRNO: END. Verander dat in: IF ERRNO (<) 151 AND ERRNO (<) 133 THEN ? ERRNO: END

Verstandig is om dat veranderde programma opnieuw op de band te zetten en dat voortaan te gebruiken.

3. Om het NewBrainprogramma te kunnen vertalen moet het eerst op een speciale manier op de band gezet worden. Dit gaat niet met het gebruikelijke SAVE-commando.

Laad het te vertalen programma op de gewone manier en stop een nieuwe cassette in de recorder. (Denk om de aanloopstrook). Zet de recorder op opname. Stel dat uw programma TEST heet. Typ OPEN#1,1,"TEST" en druk op NEWLINE. De recorder gaat even lopen en stopt weer. Typ nu LIST#1: PRINT#1: CLOSE#1 en druk op NEWLINE. De recorder gaat weer lopen en het programma TEST wordt op een speciale manier op de band gezet.

4. Laad het schrijfprogramma. Typ daarna RUN en druk op NEWLINE. Op het scherm komt te staan: 'Schrijf Basicode NewBrain release 1.0'. Even later verschijnt de vraag: Programma naam: Stop de teruggespeelde cassette met uw zonet opgenomen programma TEST weer in de computer, typ de naam TEST en druk op NEWLINE.

Het zal u opvallen dat het inlezen nu veel langer duurt. Dat komt omdat elke byte wordt omgezet in een ASCII-code die vervolgens gepoked wordt op een geheugenadres.

Als het hele programma is ingelezen verschijnt de mededeling 'Start cassette en geef NEWLINE'.

5. Nu komt het moment waarop uw programma in Basicode wordt weggeschreven. Daar is wel een speciaal kabeltje voor nodig, dat wordt aangesloten op de COMMONS-poort. Het schema voor dit kabeltje treft u bij dit artikel aan. Let erop dat het plugje voor de afstandsbediening uit de recorder verwijderd is. Bij mij werkt het programma beter als ook het plugje uit de TAPE-1 uitgang op de computer verwijderd wordt.

Stop een nieuwe cassette in de recorder (denk om de aanloopstrook) en breng het bewuste kabeltje aan. Zet de recorder op opname en druk op NEWLINE. Het scherm wordt helaas donker zodat u geen controle heeft over het verloop van het wegschrijven. Na verloop van tijd licht het scherm weer op en komt de boodschap 'Programma weggeschreven'.

6. Als alles goed gegaan is heeft u nu een Basicodeprogramma dat u naar de NOS zou kunnen opsturen.

Maar is het ook goed gegaan? Een VERIFY-mogelijkheid ontbreekt. Om te weten of uw programma goed op te band staat is er geen andere mogelijkheid dan het in te lezen met behulp van het leesprogramma, zoals dat in het vorige gedeelte van dit artikel beschreven staat.

Ik ben benieuwd wanneer het eerste NewBrain-Basicode programma wordt uitgezonden.

Pieter W.M. van Dijk

#### BASICODE LEESPROGRAMMA

```

1 GOTO 990
10 REM Begin subroutines basicode-2
20 REM INITIALISATIE VAN DEVICES
21     OPEN #0,0
22     CLOSE #5 : OPEN #5,5
24     CLOSE #6 : OPEN #6,6
26     BD$="2400" : REM BAUDRATE PRINTER
28     CLOSE #8 : OPEN #8,0,BD$
30     RANDOMIZE
32 GOTO 1010
100 REM SCHOONSCHEM + HOME
102     PUT 31,12
104 RETURN
110 REM PLAATS CURSOR OP (HO,VE)
112     IF HO=0 AND VE=0 AND VE<24 AND HO<40 THEN PUT 22,HO+1,VE+1
114 RETURN
120 REM GEEF CURSORPOSITIE IN (HO,VE)
122     PUT 21
124     GET HO,VE:HO=HO-1:VE=VE-1
126 RETURN
200 REM LEES IN# VAN DEVICE 6
202     GET #6,IN$:IF ASC(IN$)=0 THEN IN$=""
204 RETURN
210 REM LEES IN# VAN DEVICE 5
212     GET #5,IN$:IF ASC(IN$)=0 THEN IN$=""
214 RETURN
250 REM DE NEWBRAIN HEEFT GEEN BEEP,
252 REM DUS LATEN WE HET SCHEM FLITSSEN.
254     PUT 23,49,0,0,0,0,0,0,0,0,0,0,23,48
256 RETURN

```

```

260 REM PLAATS RANDOMGETAL IN RV
262 RV=RND
264 RETURN
270 REM PLAATS FREE IN FR
272 FR=FREE
274 RETURN
300 REM MAAK SR$ VAN SR ZONDER SPATIES
302 SR$=STR$(SR)
304 SR$=MID$(SR$,2,LEN(SR$)-2)
306 RETURN
310 REM MAAK SR$ VAN SR MET LENGTE CT
312 REM MET CN CYFERS ACHTER DE KOMMA
313 SR=INT(SR*(10^CN)+.5)*10^-CN
314 GOSUB 300
315 IF cn=0 GOTO 320
316 IF INSTR(SR$,".")=0 THEN SR$=SR$+"."
318 SR$=SR$+"0000000000"
319 SR$=LEFT$(SR$,LEN(SR$)-((LEN(SR$)-INSTR(SR$,"."))-CN))
320 SR$=RIGHT$(" "+SR$,CT)
322 RETURN
350 REM STUUR SR$ NAAR PRINTER
352 PRINT #8,SR$;
354 RETURN
360 REM NEWLINE NAAR PRINTER
362 PRINT #8
364 RETURN
400 REM
500 REM Einde subroutines basicode-2
600 REM
700 REM
800 REM begin lees basicode programma
900 REM
990 DELETE 1-990
1000 ct=52: REM delay teller
1010 en=11776: REM start adres z80 code
1020 t=32*1024: REM top van het geheugen
1030 dv=PEEK(19)
1040 OPEN#0,0
1050 PUT31:?:?:?:?"Lees basicode NewBrain release 1.0":?:?
1060 RESERVE TOP-en
1070 RESTORE
1080 FOR i=en TO en+223
1090 READ a: POKEi,a
1100 NEXT i
1110 POKE en+163,ct
1120 c=PEEK(en+3)+PEEK(en+4)*256-1
1130 POKE c+10,2: POKE c+11,3
1140 REMdelete 120-200
1150 LINPUT("Naam weg te schrijven programma: ")A$

```

```

1155 IF dv=12 THEN IF a$="" THEN 1150
1160 ?:"Start cassette en geef NEWLINE?";
1170 GETa
1180 ON ERROR GOTO 1200:?: CALL en
1190 ?:"FOUT ??????":END
1200 a=ERRNO: ON ERROR GOTO 0
1210 ON a GOTO 1350,1220,1230,1240,1250
1220 ?"Checksum klopt": GOTO 1260
1230 ?"Checksum klopt niet": GOTO 1260
1240 ?"Inlezen afgebroken": GOTO 1260
1250 ?"Time out error": GOTO 1260
1260 REM Schrijf Basicode programma naar default uitvoer
1270?: b=c
1280 b=b+1:?:b:PUT28:a=PEEK(b): IF a<>2 AND b<>t GOTO 1280
1290 IF b=t THEN ?:"Geen start tekst gevonden": b=c
1300 IF dv=1 OR dv=2 THEN ?"Start cassette en geef return":GET a
1310 CLOSE#12: OPENOUT#12,dv,a$
1320 b=b+1: IF b=t THEN ?:"Geen einde tekst": GOTO 1340
1330 a=PEEK(b): IF a<>3 THEN PUT#12,a: PUT a: GOTO 1320
1340 PUT#12,4,13:CLOSE#12
1350 ?:"a$;" is weggeschreven"
1360 RESERVE 2*15+TOP
1370 END
10000 DATA 195,19,46,252,46,0,5,229,42,5,46,43,34,5,46,125
10010 DATA 100,225,201,243,33,0,0,57,34,218,46,49,252,46,33,251
10020 DATA 46,34,215,46,205,95,46,205,112,46,205,133,46,121,254,3
10030 DATA 32,242,205,95,46,205,112,46,58,217,46,185,32,4,62,2
10040 DATA 24,2,62,3,50,212,46,24,14,62,5,50,212,46,42,215
10050 DATA 46,35,54,3,34,215,46,0,42,218,46,249,251,55,201,205
10060 DATA 172,46,62,21,187,48,248,205,172,46,62,21,187,48,240,201
10070 DATA 14,128,205,150,46,121,31,79,48,248,42,215,46,35,203,185
10080 DATA 113,34,215,46,201,121,254,130,50,217,46,32,4,175,34,213
10090 DATA 46,169,50,217,46,201,197,6,2,205,172,46,83,205,172,46
10100 DATA 123,130,254,43,48,4,5,194,153,46,193,201,58,36,0,203
10110 DATA 151,50,36,0,203,167,211,7,219,22,230,2,111,30,0,28
10120 DATA 202,204,46,219,22,230,2,189,202,191,46,201,205,7,46,32
10130 DATA 238,195,73,46,0,0,0,0,0,0,0,0,0,0,0,0,0
30000 REM cass
30010 REM ear <-----> cts
30020 REM
30030 REM
30040 REM (optioneel) I I
30050 REM 1k ohm I I
30060 REM I I
30070 REM
30080 REM I
30090 REM
30100 REM
30110 REM grnd <-----> grnd
30120 REM

```

```

30130 REM De pull up tussen cts en rts
30140 REM is alleen nodig voor recorders
30150 REM met te weinig uitgangsspanning
30160 REM *****
30170 REM Tradecom International B.V.
30180 REM Hondsdijk 3
30190 REM Koudekerk a/d Rijn
30200 REM Telefoon 01714 4300
30210 REM *****

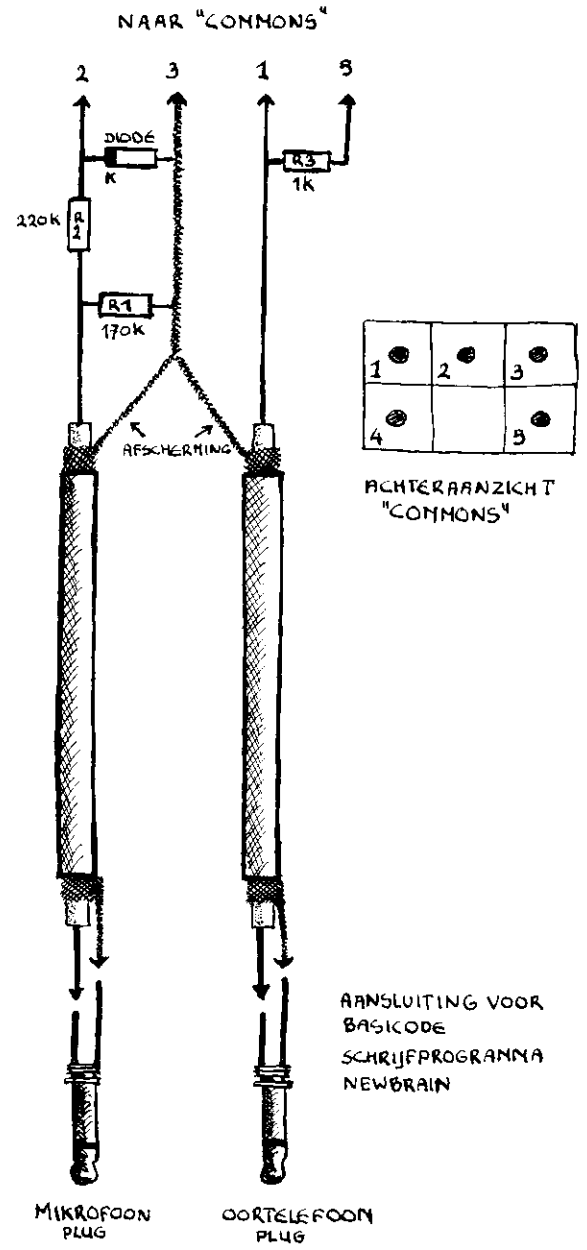
```

BASICODE SCHRIJFPROGRAMMA

```

100 en=11776: t=32*1024
110 dv=PEEK(19)
120 OPEN#0: CLOSE#5: OPEN#5,5
125 PUT31:?:?:?:?:?:?Schrijf basicode NewBrain release 1.0:?:?
130 RESERVE TOP-en
140 RESTORE
150 FOR i=en TO en+223
160 READ a: POKEi,a
170 NEXT i
180 c=PEEK(en+3)+PEEK(en+4)*256
190 LINPUT("Programma naam: ")A$: IF dv=12 THEN IF a$="" THEN 190
200 OPENIN#12,dv,a$
210 ON ERROR GOTO 400
220 b=c: a=130: GOSUB 300
230 GET#12,a: a=a OR 120: IF a=130 THEN 230
240 REMIF a=141 THEN cr=cr-1: IF cr=0 THEN cr=16: a=0
250 GOSUB 300: GOTO 230
300 REM
310 b=b+1: POKE b,a: RETURN
400 IF ERRNO<>151 THEN ?ERRNO: END
410 CLOSE#12
420 a=131: GOSUB 300
430 CALL en
440 ?"Programma weggeschreven"
450 RESERVE 2*15+TOP
460 END
10000 DATA 195,28,46,1,47,225,126,35,229,245,230,127,30,0,231,48
10010 DATA 241,203,127,40,240,30,5,231,49,56,250,201,33,1,47,62
10020 DATA 0,50,224,46,126,246,128,119,58,224,46,174,50,224,46,126
10030 DATA 35,254,131,32,239,58,224,46,119,35,54,0,35,54,0,205
10040 DATA 5,46,83,116,97,114,116,32,99,97,115,115,101,116,116,101
10050 DATA 32,101,110,32,103,101,101,102,32,78,69,87,76,73,78,69
10060 DATA 13,138,243,33,0,0,57,34,222,46,49,1,47,205,169,46
10070 DATA 33,1,47,86,122,167,30,11,212,181,46,220,193,46,55,122
10080 DATA 31,87,29,32,243,35,126,167,204,154,46,32,230,205,169,46
10090 DATA 251,0,42,222,46,249,251,55,63,201,35,126,167,43,200,22
10100 DATA 176,205,172,46,62,13,119,167,201,17,112,23,205,193,46,27
10110 DATA 122,179,32,248,201,62,0,6,122,205,215,46,6,122,195,215
10120 DATA 46,62,0,6,60,205,215,46,6,60,205,215,46,6,60,205
10130 DATA 215,46,6,60,195,215,46,211,7,16,254,238,32,201,0,0

```



ACHTERBAAZICHT "COMMONS"

AANSLUITING VOOR BASICODE SCHRIJFPROGRAMMA NEWBRAIN

## KASBOEK

Kasboek is een programma dat uit twee delen bestaat welke niet onafhankelijk van elkaar kunnen werken. Het eerste programma maakt een file (bestand) aan waar het tweede programma mee kan werken. Voordat we gaan beginnen moet u het deel 'kasboek1' draaien. De vragen die dit programma stelt worden hieronder kort uitgelegd.

Vraag	Verklaring (beknopt)
1.	Dat is dit wat U nu leest.
2.	De naam van het bestand met gegevens (max 25 kar.).
3.	Welk jaar begint U met invoer van gegevens?
4.	De maand waar U begint (vb sept = 9).
5.	Bij 'N' als antwoord gaat U naar vraag 3 terug.
6.	Hoeveel groepen inkomsten heeft u (1 - 15)?
7.	En hoeveel groepen uitgaven (1 - 15)? (totaal van vraag 6 en 7 mag niet meer dan 15 bedragen)
8.	Indien U 'N' invult gaat U naar vraag 6 terug.
9.	De naam die U aan een groep wilt geven (supermarkt, bijv.)
10.	Indien U 'N' intypt gaat U de eventuele fout herstellen.
11.	Type de letter in van de groep die fout is (is dit een E dan is de foutcorrectie klaar.)
12.	Wat is Uw geheugengrootte (32 of 96 kilobyte)?
13.	Deze vraag is om U er aan te herinneren dat U een blanco tape in Uw cassette recorder heeft gedaan.

Druk op de 'record' en 'play' toetsen om dit bestand op te nemen voor verder gebruik door kasboek 2.

U heeft gezien hoe we een nieuw bestand opzetten. Nu gaan we bekijken hoe men het eigenlijke programma gebruikt.

Als U het programma 'runt' krijgt men een menu op het scherm met negen opties en daaronder hoeveel geheugen er nog vrij is. Om een optie te kiezen moet U gewoon het nummer intypen van wat U wilt gaan doen. Hier volgen wat voorbeelden van opties.

## 5. Groepstotaal

Type de keuzeletter in, hierna komt het groepstotaal van deze maand in beeld en tevens de balans. Hier zijn weer opties mogelijk namelijk:

- F volgende groep
- B een groep terug
- T totaal inkomsten en uitgaven
- V wijzigen (als bij 4)
- H histogram. Hier wordt om een maximum getal gevraagd, neem hiervoor het grootste bedrag dat in een maand voorkomt. Indien dit histogram voor alle groepen is typt U 'J' in (alle inkomsten) en indien 'N' dan zijn dit alle uitgaven.
- P print (als bij 4)
- R herstel (als bij 4)

## 6. Vernietig een groep van een maand

Type de maand in. Selecteer daarna de groep (keuze-letter). Hierna wordt de groep op het scherm getoond en de vraag of U deze wilt vernietigen. Indien U 'N' intypt gebeurt er niets. Indien 'J' dan wordt voor deze maand de groep vernietigd. Dit kan enige tijd in beslag nemen omdat automatisch gesorteerd wordt. Hierna komt weer het menu of het scherm.

## 7. Zoek een post.

Dit zoeken kan op twee manieren gebeuren nl. op het bedrag (het sleist sleist) of op de opgegeven omschrijving. Er wordt gevraagd of het bedrag bekend is; indien 'J' dan wordt naar de grootte gevraagd. Indien 'N' is ingevuld dan wordt naar een omschrijving gevraagd die het meeste informatie geeft om hem uniek te houden (verkorting zoektijd). Indien een overeenkomst is gevonden, dan wordt deze op het scherm getoond, waarna drie opties mogelijk zijn namelijk:

- D vernietig, waarna een sortering van de maand wordt uitgevoerd.
- G verplaats naar een andere groep. Er wordt om de keuzeletter gevraagd van de groep waar de post heen moet. Hierna volgt het menu weer.
- S Zoek verder naar nog een post die overeenkomt met de zoek-sleutel.

Iedere andere toets dan deze drie geeft weer het menu, zonder dat er iets veranderd is.

## 8. Copieer data naar cassette.

Zorg dat een cassette klaarstaat en type 'J' in zodat alles wordt weggeschreven naar de cassette. Dit kan enige tijd duren. Als het programma klaar is volgt weer het menu op het scherm. Indien men 'N' op de vraag intypt, gebeurt er niets en volgt weer het menu op het scherm. Men moet natuurlijk dit wegschrijven alleen doen als men iets veranderd heeft aan de data anders blijft het bestand zoals het was.

## 5. Groepstotaal

Type de keuzeletter in, hierna komt het groepstotaal van deze maand in beeld en tevens de balans. Hier zijn weer opties mogelijk namelijk

- F volgende groep
- B een groep terug
- T totaal inkomsten en uitgaven
- V wijzigen (als bij 4)
- H histogram. Hier wordt om een maximum betaald gevraagd, neem hiervoor het grootste bedrag dat in een maand voorkomt. Indien dit histogram voor alle groepen is typt U 'J' in (alle inkomsten) en indien 'N' dan zijn dit alle uitgaven
- P print (als bij 4)
- R herstel (als bij 4)

## 6. Vernietig een groep van een maand

Type de maand in. Selecteer daarna de groep (keuze-letter). Hierna wordt de groep op het scherm getoond en de vraag of U deze wilt vernietigen. Indien U 'N' intypt gebeurt er niets. Indien 'J' dan wordt voor deze maand de groep vernietigd. Dit kan enige tijd in beslag nemen omdat automatisch gesorteerd wordt. Hierna komt weer het menu of het scherm.

## 7. Zoek een post.

Dit zoeken kan op twee manieren gebeuren nl. op het bedrag (het snelst) of op de opgegeven omschrijving. Er wordt gevraagd of het bedrag bekend is, indien 'J' dan wordt naar de grootte gevraagd. Indien 'N' is ingevuld dan wordt naar een omschrijving gevraagd die het benodigde informatie geeft om hem uniek te houden (verkortings zoektijd). Indien een overeenkomst is gevonden, dan wordt deze op het scherm getoond, waarna drie opties mogelijk zijn namelijk:

- D vernietig, waarna een sortering van de maand wordt uitgevoerd.
  - G verplaats naar een andere groep. Er wordt om de keuzeletter gevraagd van de groep waar de post heen moet. Hierna volgt het menu weer.
  - S Zoek verder naar nog een post die overeenkomt met de zoek-sleutel.
- Iedere andere toets dan deze drie geeft weer het menu, zonder dat er iets veranderd is.

## 8. Copieer data naar cassette

Zorg dat een cassette klaarstaat en type 'J' in zodat alles wordt weggeschreven naar de cassette. Dit kan enige tijd duren. Als het programma klaar is volgt weer het menu op het scherm. Indien men 'N' op de vraag intypt, gebeurt er niets en volgt weer het menu op het scherm. Men moet natuurlijk dit wegschrijven alleen doen als men iets veranderd heeft aan de data anders blijft het bestand zoals het was.

## 9. Einde

Als men klaar is en eventuele veranderingen weggeschreven heeft kan men via deze keuze het programma verlaten zonder dat het programma vernietigd wordt.

Dit was de beschrijving van kasboek 1 en 2. Indien men vragen heeft over dit programma kunt U deze uitsluitend schriftelijk stellen onder bijsluiting van een postzegel van het huidige tarief voor een brief van 20 gram (nu 22 april '84 is dit 70 cent). Waarna zo spoedig mogelijk antwoord volgt (ongeveer twee weken). U kunt Uw vragen sturen aan:

J. J. Vooijs  
postbus 17245  
2502 CE s-Gravenhage

J. J. Vooijs.



## KAARTENEAF

De gebruiksaanwijzing van het programma KAARTENEAF is zeer eenvoudig, aangezien het programma zichzelf gedeeltelijk uitlegt. Het enige belangrijke is dat het eigenlijke programma KAARTEN2 is en dat KAARTEN1 een bestand aanmaakt voor KAARTEN2. Dit is net zo toegepast bij HANDEK 1 en 2.

Men moet dus om het programma KAARTEN2 te kunnen gebruiken, eerst het programma KAARTEN1 gebruiken en dus diverse vragen beantwoorden. Zoals hoeveel kaarten men wil gebruiken (of een indicatie hoeveel men er krijgt van het programma, afhankelijk van de geheugenruimte), wat er allemaal op de kaarten komt (aantal regels), enz.

Het programma KAARTEN2 bestaat uit een menu waarvan u men door het gehele programma kan gaan. Er zijn mogelijkheden om te bladeren, te sorteren, te zoeken en te rekenen.

Het mogelijk om het programma aan te passen aan disks, door een aantal programmaregels te wijzigen. Men moet met name een aantal OPEN-statements veranderen, zodat device nummer 12 wordt gebruikt.

Heeft U vragen over dit programma dan kunt U deze uitsluitend schriftelijk, onder bijsluiting van een postzegel van 70 cent, stellen aan:

J.J. Vooijs  
postbus 17245  
2502 DE den Haag

J.J. Vooijs

NEWBRAIN TEXT EDITOR version 3, release 7

January 20th, 1984

Jos Haring, Mario Pinto

CONTENTS

<b>NewBrain Text Editor</b>	<b>1</b>
Korte Beschrijving	1
Het werken met de editor	2
De editor-kommando's	4
PF-toetsen	4
Het programma	5
De printroutine	6

NEWBRAIN TEXT EDITOR

Een veelvuldig voorkomende applicatie voor mikro's is die van een text editor. Een dergelijk programma is in feite bedoeld als een gebruikersvriendelijke vervanging van een schrijfmachine, met vele extra mogelijkheden voor de opmaak van de tekst. In dit artikel zullen we trachten uiteen te zetten wat de mogelijkheden zijn van de door ons geschreven editor.

Korte Beschrijving

Diegenen die beroepshalve weleens met IBM-pakketten als VSPC en SPF in aanraking zijn gekomen, zullen ongetwijfeld veel eigenschappen van die editors kunnen terugvinden in deze NEWBRAIN text editor. Het was namelijk onze bedoeling om een editor te ontwikkelen die als eerste vereiste gebruikersvriendelijkheid bezit en daarbij ook nog over een groot arsenaal van functies beschikt.

De door ons toegepaste configuratie bestaat uit een standaard NewBrain mikro, een cassetterecorder en een STAR DP-510 matrixprinter. De editor werkt alleen met een beeld van 80 tekens per regel; daarom verdient het aanbeveling om een monitor te gebruiken inplaats van een gewoon TV-toestel in verband met de scherpte van de tekens.

De editor is (in tegenstelling tot de meeste andere editors) geschreven in BASIC. Een belangrijke reden

hiervoor is, dat uitbreidingen c.q. wijzigingen nu gemakkelijk kunnen worden aangebracht en bovendien is het programma op deze manier nog enigszins te volgen. Natuurlijk wordt door het gebruik van BASIC een relatief lage snelheid geïntroduceerd, maar al met al is het toch gelukt om deze traagheid binnen aanvaardbare grenzen te houden. Het is b.v. mogelijk om tijdens de invoer van de tekst een typesnelheid aan te gebruiken van 18 tekens per seconde, zodat zelfs ervaren typiste(s)(n) geen problemen hoeven te ondervinden.

### Het werken met de editor

De editor bevat een zgn. tekstbuffer, waarin alle ingebrachte tekst wordt opgeslagen. Bij het opstarten van de editor verschijnt er een scherm waar de gebruiker de grootte van deze buffer (in regels) kan invoeren. De defaultwaarde voor deze buffergrootte is 200 regels en kan maximaal 999 regels bedragen. In een standaard NewBrain bevindt zich ruimte voor ca. 6000 tekens tekst, maar met geheugenuitbreiding kan dit natuurlijk naar eigen believen worden opgevoerd. Nadat nu de buffergrootte is ingevoerd verschijnt het eigenlijke editor-scherm, dat er als volgt uitziet:

Het scherm bestaat van boven naar beneden uit vier gedeeltes:

1. een boodschappenregel,
2. een kommandoregel,

3. 20 tekstregels,
4. een statusregel

De boodschappenregel dient voor het laten zien van de fout- en informatieboodschappen die door de editor worden gegeven. De COMMAND INPUT==> regel dient (uiteraard) voor het invoeren van de verschillende kommando's. Let erop dat kommando's alleen met kleine letters kunnen worden ingevoerd! Het volgende gedeelte, de twintig tekstregels, laten een gedeelte van de tekstbuffer zien waarin direkt kan worden gewerkt en gebladerd. In de daaropvolgende regel zien we achtereenvolgens het 'current block' (het gebied waarbinnen de editorfuncties werkzaam zijn), de ingestelde buffergrootte (BUFSZ), de tekstbreedte (LINSZ) en de naam van de te verwerken tekstfile.

Nu kan de gebruiker beginnen met het uitvoeren van de verschillende editor-kommando's, waarvan er zo'n 35 ter beschikking staan. Deze kommando's zijn onder te verdelen in de volgende groepen:

- **File-kommando's**  
file, load, merge, save, verify
- **Printer-kommando's**  
hardcopy, print
- **Edit-kommando's**  
(, ), =, change, clear, copy, delete, find, input, insert, join, move, repeat, sort, split
- **Scherm-kommando's**  
+, ++, -, --, cols, end, locate, reset, show

- **Status-kommando's**  
block, free, last, linesize, memory, time

De editor-kommando's

Bovenstaande kommando's zullen in het algemeen met een of meerdere operands uitgevoerd moeten. Meestal zullen de kommando's betrekking hebben op het ingestelde 'current block', wat in veel gevallen met een \* moet worden aangegeven. Alle kommando's zijn 'auto repeating', d.w.z. door slechts op de NEW LINE-toets te drukken wordt het kommando nogmaals uitgevoerd. Dit is in het bijzonder gemakkelijk bij het bladeren in de tekst met de + en - kommando's.

Een uitgebreide beschrijving van de editorkommando's met syntaxregels kan gevonden worden in het aanhangsel.

PF-toetsen

Zeer gemakkelijk in het gebruik zijn de PF (Programmeerbare Functie) - toetsen (CNTL/A t/m CNTL/T). Deze kunnen worden gebruikt voor het opslaan van vaak voorkomende en/of moeilijke strings, die dan overal in de tekst kunnen worden ingebracht door simpelweg de betreffende PF-toets in te drukken. Het definiëren van deze toetsen is zeer eenvoudig: ze corresponderen met de eerste 20 regels van de tekst (CNTL/A = regel 0, CNTL/T =

regel 19). Bij deze regels is ook altijd de letter van de bijbehorende PF-toets aangegeven.

Het gebruik van deze toetsen is natuurlijk niet verplicht; de eerste 20 regels kunnen ook gewoon voor het opslaan van normale tekstregels worden gebruikt.

Het programma

Het programma is, zoals de listing al laat zien, van behoorlijke omvang en beslaat ca. 19 Kbyte RAM. Door het schermgeheugen en de variabelen blijft er nog ca. 6K beschikbaar voor de tekst (in een niet uitgebreide NewBrain). Daar dit niet erg veel is, zal de tekst vaak op cassette moeten worden opgeslagen. De file-kommando's en met name verify, verzekeren een veilige en (ten opzichte van de invoersnelheid van de tekst) snelle opslag van de tekst. Voor de cassette in- en uitvoerpoort wordt een buffergrootte (blocksize) van 512 bytes gebruikt.

Het programma kent een modulaire opbouw. Tot regel 60 worden alle variabelen geïnitieerd, waaronder de tekstbuffergrootte en de default printer-parameters. Voor een warme start (zonder verlies van data) moet naar regel 60 worden gesprongen (GOTO 60). Tot regel 1000 worden de kommando's ontleed en geanalyseerd. Verder bestaat het programma uit subroutines voor de uitvoering van de verschillende kommando's, die allemaal beginnen op een lijnnummer dat een 1000-voud is.

De printroutine

Een van de belangrijkste en zeker de omvangrijkste routines van dit programma is de printroutine. Het vervelende hierbij is dat er bijna evenveel typen printers bestaan als mikro's met ieder zijn eigen mogelijkheden en besturingskonventies. Om het toch mogelijk te maken om een willekeurige serieele printer aan de NewBrain te koppelen worden hieronder de noodzakelijke gegevens gegeven.

De editor werkt met een serieele printer die op de COMMS-poort aangesloten moet worden. Dit is nodig om geen last te hebben van de carriage return (ASCII 13) kodes die na iedere 80 tekens op de printerpoort verschijnen. Gebruik stream #9 voor het printen naar de printer en stel de baudrate op 9600 in (default). Deze baudrate is desgewenst te wijzigen bij het openen van de COMMS-stream op regel 60.

Bij de hier gebruikte printroutine wordt op uitgebreide wijze gebruik gemaakt van alle mogelijkheden die de STAR DP510 printer te bieden heeft. Deze mogelijkheden zijn bij het printkommando beschreven. Om zelf een printroutine te schrijven is het noodzakelijk het volgende te weten.

Deze routine is geschreven vanaf regel 11000 tot 11999. De tekst is opgeslagen in een 1-dimensionale array 'l\$( )' met OPTION BASE 0. Het te printen block begint bij regel 'bb' en eindigt bij regel 'be'. Bij het verlaten van de printroutine moeten de volgende statements worden gebruikt:

```
PUT31:GOSUB500:GOSUB1000:m$=" BLOCK PRINTED ":RETURN
```

Hieronder geven we een voorbeeld van een printroutine op zijn simpelst, waarbij gewoon de gehele tekstbuffer wordt leeggeprint, met of zonder regelnummers. Er wordt geen gebruik gemaakt van control characters.

```
11000 OPEN#9,9
11010 PUT31:INPUT("Line numbering (y/n):")a$
11020 IF a$<>"y" AND a$<>"n" THEN GOTO11010
11030 FOR i=bb TO be
11040 IF a$="y" THEN PRINT#9,i[3];
11050 PRINT#9,l$(i)
11060 NEXT i
11070 PUT31:GOSUB500:GOSUB1000:m$=" BLOCK PRINTED "
11080 RETURN
```

Deze printroutine zou in het begin als vervanging kunnen dienen van de huidige routine voor diegenen die niet in het bezit zijn van een STAR-printer.

**{block[b] [begin] [end]}**

Defaults: 'begin'=0, 'end'=999

Dit kommando stelt de grenzen vast waarbinnen in principe alle kommando's actief zijn. In het bijzonder wordt bij de volgende kommando's het block specifiek aangegeven:

```
change *
clear *
copy *
delete *
find *
locate *
move *
show *
```

Voorbeeld

```
block 12 50
```

**{change[c] [\*|line] ['string1'string2'|"string1"string2"]}**

Het change-kommando verandert een opgegeven string 'string1' in de substitutiestring 'string2', binnen het current block of in de opgegeven regel.

Voorbeeld

```
change 20 'abc1'abc2'
c * "xx"x"
```

**{clear[c1] [\*|line]}**

Hiermee wordt de opgegeven regel of het current block uitgewist, maar niet verwijderd. Er blijft dus een blanco tekstgedeelte over.

Voorbeeld

```
clear 45
```

**{cols[cs] [line]}**

Door het geven van dit kommando verschijnt er op de opgegeven regel een referentielijn voor het vaststellen of uittellen van posities. Wanneer geen regelnummer is opgegeven, verschijnt de cols-regel direkt onder de COMMAND INPUT-regel. In dat geval blijft de cols-regel bij het scrollen bewaard. De lengte van de cols-regel is gelijk aan de geldende linesize.

**{copy[cp] [\*|line1] line2}**

Met het copy-kommando wordt de regel 'line2' overschreven met een kopie van regel 'line1'. Wanneer '\*' werd gespecificeerd dan wordt een kopie van het current block gemaakt die begint op regel 'line2'.

Voorbeeld

```
copy * 80
cp 23 28
```

**delete [\*|line]}**

Hiermee is het mogelijk om een of meerdere regels uit de tekstbuffer te verwijderen. Hierbij schuiven de achterliggende regels naar boven. Wanneer '\*' is gespecificeerd, dan wordt het current block uit de tekstbuffer verwijderd.

Voorbeeld

```
d *
delete 23
```

**end**

Met dit kommando wordt met een speciaal teken het einde van iedere regel aangegeven (blanks tellen ook als characters!). Met het reset-kommando kan dit weer ongedaan worden gemaakt.

file filename

Dit kommando dient voor het toekennen van een naam aan de tekst-buffer. Deze naam wordt gebruikt wanneer de file op cassette wordt gezet. De naam mag maximaal 29 tekens lang zijn.

{find|f}{\*|line] {'string'|"string"[ASCII-kode]}

Met het find-kommando kan men zoeken naar een string in een bepaalde regel of in het current block. Wanneer de string wordt gevonden, dan wordt met horizontale en verticale pijltjes duidelijk gemaakt waar deze zich bevindt. Wanneer men in plaats van een string een ASCII-kode opgeeft, dan wordt er naar dit betreffende character gezocht. Een regelnummer geeft de laatste regel aan waar de opgegeven string in voorkomt.

Voorbeeld

```
f * 'NEXT'
find "start N"
find 132
```

free

Dit kommando laat zien hoeveel geheugen er nog beschikbaar is om tekst in op te slaan. Om een tekst op te slaan moeten er tenminste ca. 500 bytes beschikbaar zijn.

{hardcopy|h}

Hiermee wordt een letterlijke afdruk gemaakt van het scherm via de COMMS-poort van de NewBrain.

{input|i} [line] [start]

Defaults: 'line'=current line, 'start'=1

Het input-kommando dient voor het invoeren van tekst, te beginnen op positie 'start'. De cursor-toetsen kunnen worden gebruikt om fullscreen te kunnen werken. Het scherm scrollt automatisch wanneer dat nodig is. Om weer naar de kommandoregel terug te keren kan de HOME toets worden gebruikt. De regels die langer zijn dan de current linesize worden automatisch afgekapt.

Het invoeren van tekst begint op de laatst gebruikte cursorpositie of op de door de gebruiker aangegeven regel. De laatste ingevoerde regel wordt na het beëindigen van de invoer door een > gemarkeerd.

Opmerkingen bij het input kommando

Tijdens het invoeren van de tekst hebben vele toetsen een bepaalde functie om het werken te vergemakkelijken. Hieronder volgt een overzicht van deze functies.

PUT#	Toets	Functie
1	SHIFT/INSERT	Invoegen van een nieuwe regel
2	SHIFT/↓	Verwijderen van een regel
3	CNTL/NEWLINE	Verwijder de rest van een regel vanaf de cursorpositie
8		Cursor naar links
9	CNTL/ESCAPE	Tab 8 spaties
10		Cursor naar beneden
11		Cursor naar boven
12	HOME	Terug naar COMMAND INPUT===>
13	NEWLINE	Invoeren van regel
14	SHIFT/↑	Attribute on (speciale tekenset)
15	SHIFT/ESCAPE	Attribute off
17	INSERT	Insert modus
18	GRAPHICS/↑	Kopieer de laatst ingevoerde regel op de 'current line'
24	SHIFT/←	Verwijder teken links van de cursor
25	SHIFT/→	Verwijder teken rechts van de cursor
26		Cursor naar rechts
27	ESCAPE	Zet cursor op volgende woord in de regel
28	CNTL/←	Cursor naar begin van de regel
29	CNTL/→	Cursor naar eind van de regel
30	CNTL/HOME	Regel schoonmaken

**{insert[is] line [no. of lines]}**

Defaults: 'no. of lines'=1

Hiermee kan men regels tekst tussenvoegen in de tekstbuffer. De regels worden tussengevoegd voor regel 'line'. Desnoods kunnen er meerdere regels worden tussengevoegd door achter het regelnummer het aantal tussen te voegen regels op te geven. Houdt er rekening mee, dat bij iedere tussen te voegen regel de laatste regel uit de tekstbuffer verdwijnt! De default-waarde voor 'no. of lines' is 1.

Voorbeeld

```
insert 17
is 12 5
```

**{join[j] line}**

Dit kommando voegt twee opeenvolgende regels samen tot een enkele regel tekst, zolang deze niet langer is dan de current linesize.

Voorbeeld

```
j 120
```

**{last[l]}**

Dit kommando brengt het laatst uitgevoerde kommando terug naar de COMMAND INPUT==> regel, echter zonder dit uit te voeren. Op deze wijze is het eenvoudig om snel veranderingen aan te brengen in een nog te geven kommando.

**{linesize[ls] [linesize]}**

Defaults: 'linesize'=73

Met het linesize-kommando kan men de maximale breedte van de tekst vaststellen. Deze maximale breedte is van invloed bij het input-kommando, het change-kommando, het join-kommando en het ==-kommando.

**load**

Het load-kommando laadt een eerder opgeslagen file vanaf cassette in de tekstbuffer. Ook de volgende parameters worden geladen: current block, buffergrootte, regelbreedte, laatste cursorpositie en de printer-parameters.

**{locate[lc] [\*|line]}**

Met het locate-kommando kan een bepaalde regel tekst of de eerste regel van het current block gelokaliseerd. Wanneer deze bestaat, verschijnt deze als eerste regel in het scherm.

**{memory[mm]}**

Dit kommando laat rechtsboven in het scherm het aantal characters zien dat zich in het current block bevindt.

**merge filename**

Op deze manier kan een andere file van cassette worden geladen en over de tekst worden gelegd, d.w.z. blanco regels worden overschreven, maar bestaande blijven ongewijzigd.



`{move|m} [*|line] dest-line`

Hiermee kan een bepaalde regel of een aantal regels naar een andere plaats worden gebracht. De regel die eerder op 'dest-line' stond, schuift hierdoor naar beneden. Wanneer '\*' werd gespecificeerd wordt het current block in zijn geheel verplaatst.

#### Voorbeeld

```
move 20 1
m * 100
```

`{print|p}`

Dit kommando print het current block op de aangesloten printer. Wanneer het kommando gegeven is, verschijnt een menuscherm met default-waarden die naar believen kunnen worden gewijzigd, voor besturing van control characters, line spacing, bladnummering etc..

#### Opmerkingen

Het is mogelijk om de momentele verticale positie (current vertical position) per keer op te geven. Als dan de files aan elkaar geprint moeten worden dan dient hiervan de default-waarde niet veranderd te worden.

Het is ook mogelijk om losse vellen papier te gebruiken omdat er geen form-feed kommando wordt gegeven wanneer een current positie van nul (0) wordt gespecificeerd.

Tevens kan steeds de 'current page' worden opgegeven of eventueel omgewijzigd worden gelaten. Door 'not' op te geven als paginanummer wordt de nummering uitgeschakeld.

#### Control characters

Control characters zijn tekens voor het sturen van de printer. Wanneer deze worden gebruikt behoren ze altijd in de posities 1 en/of 2 van de tekstregels te staan. Hiervan dient de eerste positie om de line spacing te sturen en de tweede om de tekensets van de printer te sturen. Wanneer slechts een CC wordt gebruikt, dient deze altijd voor de line spacing. Het spreekt eigenlijk vanzelf dat wanneer CC's worden toegepast, de eerste (twee) positie(s) niet voor tekst mogen worden gebruikt.

Hieronder volgt een opsomming van de verschillende CC's:

#### CC's op positie 1

- F of f: Formula spacing (halve line spacing)
- G of g: Graphics spacing. Graphics characters worden aan elkaar vast geprint.
- + : Geen line feed, maar wel een carriage return
- & : Geen line feed, geen carriage return
- 0 : Reservering van drie lijnen op een pagina om formules zeker aan een stuk op een pagina te krijgen.
- 1 - 9 : 1 t/m 9 maal een halve inch spatie plus 3 regels spatie reserveren voor het gebruik van illustraties tussen de tekst.

CC's op positie 2

X : stelt een 10-pitch (characters per inch) modus in.  
 Y : stelt een 12-pitch modus in.  
 Z : stelt een 17-pitch modus in.

De volgende CC's voor pos. 2 werken per regel:

P : superscript  
 G : superscript/italic  
 R : subscript  
 B : subscript/italic  
 ! : italic  
 \$ : underlined  
 % : italic/underlined  
 ( : bold  
 ) : italic/bold  
 , : underlined/bold  
 - : italic/underlined/bold  
 0 : enlarged  
 1 : italic/enlarged  
 4 : underlined/enlarged  
 5 : italic/underlined/enlarged  
 8 : bold/enlarged  
 9 : italic/bold/enlarged  
 < : underlined/bold/enlarged  
 = : italic/underlined/bold/enlarged

Het is niet mogelijk om bold met 12- of 17-pitch characters te combineren. Bij beide control characters geldt dat spaties een normale linespacing c.q. lettertype inhouden.

**{repeat[rp] line [factor]}**

Hiermee kan een regel een willekeurig aantal malen worden herhaald. Default voor 'factor' is 1. De regels worden niet over andere heen geschreven, maar worden tussengevoegd. Hierdoor is het mogelijk dat regels aan het einde van de tekstbuffer verdwijnen.

**{reset[r]}**

Het reset-kommando ontdoet het beeldscherm van alle onnodige informatie zoals messages, end-tekens enz.

**{save[sv] filename}**

Het save-kommando slaat het current block op op cassette. Alle informatie betreffende current line, current block, linesize, buffersize, laatste cursorpositie en de printerparameters wordt eveneens opgeslagen.

**{show[sh] [\*]line}**

Met dit kommando wordt de opgegeven regel of de eerste regel van het current block getoond op de regel direkt onder de COMMAND INPUT==> regel. Wanneer '\*' is gespecificeerd, dan wordt de eerste regel van het current block getoond.

**sort** [start] [length]

**asort** [start] [length]

**shuffle**

Defaults: 'start'=1, 'length'=75

Het sort-kommando sorteert alle records binnen het current block in oplopende (sort) of aflopende (asort) volgorde. De sorteer-key begint op 'start position' en heeft een lengte 'length'.

Het shuffle-kommando bewerkstelligt het tegenovergestelde van de sort- en asort-kommando's: het brengt alle records in een willekeurige volgorde.

Voorbeeld

```
sort 20 5
asort 2 10
```

**{split[s]} line {position}{'string'|"string"}}**

Hiermee is het mogelijk om een bepaalde regel te splitsen in twee regels, vanaf de opgegeven positie, of vanaf het begin van de opgegeven string. De nieuwe regel begint met deze string of met het restant van de oorspronkelijke regel vanaf de opgegeven positie.

Voorbeeld

```
split 20 1
split 20 'the'
s 1 "x"
```

**STOP**

Het drukken op de STOP-toets, evt. gevolgd door een NEWLINE zorgt voor het verlaten van de editor, zonder aantasting van de gegevens. Om terug te keren naar de editor vanuit het OS moet GOTO 60 worden ingetoetst.

**{time|t}**

Dit kommando geeft de tijd die verstreken is sinds het opstarten van de editor.

**{verify|v}**

Dit kommando controleert of een file goed is opgeslagen op cassette, door de opgeslagen file opnieuw te lezen en record voor record vergelijkt met de records die in het current block zijn opgeslagen.

**( {shift amount} [start]**

Defaults: 'start'=1

Hiermee kan de tekst in de gehele buffer over 'shift amount' posities naar links verschoven tot aan de linkergrens 'start'

Voorbeeld

```
{ 1
( 10 2
```

**) {shift amount} [start]**

Defaults: 'start'=1

Hiermee kan de tekst in de gehele buffer over 'shift amount' posities naar rechts verschoven, te beginnen bij 'start'

Voorbeeld

```
) 4
) 20 1
```

+ [no. of lines]

Hiermee kan men vooruit 'bladeren' in de tekstbuffer over een zelf op te geven aantal regels. Wanneer 'no. of lines' niet is opgegeven, wordt er met 20 regels tegelijk gebladerd.

++

Dit kommando 'bladert' naar de laatste 20 regels van de tekst.

- [no. of lines]

Hiermee kan men terug 'bladeren' in de tekstbuffer over een zelf op te geven aantal regels. Wanneer 'no. of lines' niet is opgegeven, wordt er met 20 regels tegelijk gebladerd.

--

Dit kommando 'bladert' naar het begin van de tekst.

= [line[\*]] [percentage]

Defaultis: 'percentage'=80

Dit kommando 'format' de tekst in het current block, d.w.z. de tekst wordt aan de rechterzijde voorzien van een rechte marge op de positie aangegeven door de current linesize wanneer de lengte van de te bewerken regels groter is dan een door de gebruiker op te geven percentage van de current linesize. Per default staat dit percentage op 80. Het kommando verwijdert tevens alle 'trailing blanks' van iedere te verwerken regel.

Ledenlijst NewBrain o.g. per 3 mei 1984 (op postcode).

Zuure, S.	1000 BP AMSTERDAM
Bakker, Klaas T.	1011 AZ AMSTERDAM
Remeus, M.	1011 TJ AMSTERDAM
Bleeker, W.K.	1012 CT AMSTERDAM
Leth, P.J.A. van	1013 TC AMSTERDAM
Foeken, Paul	1015 EA AMSTERDAM
Boerboom, Simon K.	1015 MR AMSTERDAM
Cocu, Paul J.	1024 NL AMSTERDAM
Techninc Europe	1025 RX AMSTERDAM
Oudshoorn, W.	1034 LL AMSTERDAM
Technosure	1034 LL AMSTERDAM
Beukelaar, M.J.	1054 LS AMSTERDAM
Ruth, Tj. van	1056 AS AMSTERDAM
Degens, B.	1056 XB AMSTERDAM
Scholingengemeenschap, Osdorper	1068 MR AMSTERDAM
Beina, Bas	1073 EL AMSTERDAM
Scull, John	1073 HG AMSTERDAM
Albada, Robert van	1073 JX AMSTERDAM
MCC NewBrain GG, Secr.	1073 JX AMSTERDAM
Oss, P. van	1078 GB AMSTERDAM
Werkker, F.	1078 GL AMSTERDAM
Ridder, D.W. de	1078 LV AMSTERDAM
Stevens, H.C.C. (Menco)	1092 JV AMSTERDAM
Rooska Abbink, Paul	1094 JG AMSTERDAM
Rekers, Jan	1094 JH AMSTERDAM
Kroel, Jodi	1095 AB AMSTERDAM
Pellicaan, Kees	1098 CK AMSTERDAM
Hoek, Wim A. van	1098 KL AMSTERDAM
Piets, Theo	1102 CA AMSTERDAM
Lohstrof, C.L.	1104 AK AMSTERDAM
Bruin, J.W.	1106 GE AMSTERDAM 20
Elfferich, A.	1121 CP LANDSMEER
Woude, Hidde v.d.	1121 CP LANDSMEER
Rijnbeek, Elis	1141 JA MONNICKENDAM
Rijnbeek, Bob	1141 RM MONNICKENDAM
Bouma, B.	1151 ER BROEK IN WATERLAND
Corba, C.	1181 BZ AMSTELVEEN
Nieuwenhuis, W.H.M.	1182 HM AMSTELVEEN
Nurrahamed, M.T.	1183 AC AMSTELVEEN
Gerszt, S.	1188 EJ AMSTELVEEN
Maris, Rob J.	7311 BB APELDOORN
Hekman, A.	1211 KX HILVERSUM
Wielemaker, C.	1214 CP HILVERSUM
Wielemans, Cornelis	1214 CP HILVERSUM
Valster, Cornelis Andries	1215 RL HILVERSUM
Hageman, J.	1215 AA HILVERSUM
Rheenen, B. van	1217 TE HILVERSUM
Rutten, G.W.H.	1221 EE HILVERSUM
Houten, W. van	1222 RD HILVERSUM
Morgen, A. von	1231 AP LOOSDRECHT
Harwelen, M.W.J. van	1241 LV KORTENHOEF
Laere, G.A. van	1241 XE KORTENHOEF

Hilhorst, C. 1243 KC 's-GRAVELAND  
 Vos, D. E. 1273 GZ HUIZEN  
 Hiddink, H. 1273 XJ HUIZEN  
 Zomer, G. 1315 CL ALMERE  
 Uyldeert, Ton 1353 CH ALMERE  
 Steunebrink, Klaas 1381 AJ WEESP  
 Boterenbrood, P. 1381 EE WEESP  
 Leeuw, A. de 1382 JX WEESP  
 Ballast, N. H. J. 1399 GK HUIDERBERG  
 Dorenbos, Karnoe H. 1401 BM BUSSUM  
 Ditzhuyzen, G. L. G. van 1402 CG BUSSUM  
 Vreedenburg, M. S. 1402 GA BUSSUM  
 Persbureau Koolhoven 1412 AN NAARDEN  
 Oosterink, M. 1412 BK NAARDEN  
 Ditzel, J. 1422 EJ UITHOORN  
 Osborn, Paul 1422 JS UITHOORN  
 Raij, Wia van 19 1442 NG PURMEREND  
 Eggermont, F. J. 1444 VN PURMEREND  
 Vries, F. R. de 1484 PC GRAFT  
 Stolp, J. J. 1504 AT ZAANDAM  
 Denneaan, G. 1505 WP ZAANDAM  
 Duivenbode, K. van 1509 BS AMSTERDAM  
 Zanen, J. van 1509 ZAANDAM  
 Berkel, H. van 1541 BV KOOG a/d ZAAAN  
 Duyvenbode, C. van 1562 TP KROMMENIE  
 Kok, A. S. P. 1688 CW NIBBIXWOUDE  
 Wouterson, R. G. 1703 KB HEERHUGOWAARD  
 Nieuwenhuizen, R. 1779 EL DEN OEVER  
 Ploeg, J. van der 1782 AT DEN HELDER  
 Kanon, J. H. 1782 GD DEN HELDER  
 Sch. v. Den Helder, Alg. Techn. 1782 ND DEN HELDER  
 Koolstra, R. 1784 AG DEN HELDER  
 Lageaen, A. J. J. 1823 EV ALKMAAR  
 Paassen, N. C. J. van 1829 HJ OUDERP NH  
 Kalis, D. 1834 AR ST. PANCRAS  
 Leeuw, L. de 1852 KV HEILOO  
 Borst, A. L. M. 1861 VS BERGEN N-H  
 Knippenis, H. P. A. M. 1871 EV SCHOORL  
 Reinking, P. H. W. 1962 SG HEEMSKERK  
 Bree, L. H. F. C. van 1931 1963 AV HEEMSKERK  
 Akerboom, P. 2025 ZD HAARLEM  
 Heyden, E. v. d. 2032 ZN HAARLEM  
 Koning, E. 2102 BE HEEMSTED  
 Technisch Bureau DELTA-T 2105 SK HEEMSTED  
 Hoogland, C. 2105 XV HEEMSTED  
 Koning, J. P. 2161 RV LISSE  
 Dusoswa, D. 2172 JV SASSENHEIM  
 Joha, R. A. F. 2181 XE HILLEGOM  
 Zwarts, J. M. S. 2201 BN NOORDWIJK Z.H.  
 De Spinbaan 2240 AA WASSENAR  
 Dijk, Pieter M. M. van 2241 SB WASSENAR  
 Holst, F. B. v. d. 2242 XP WASSENAR  
 Pellikaan, H. 2266 ES LETDSCHEINDAM  
 Bom, W. A. M. v. d. 2273 SH VOORBURG

Kobus, H. W. 2274 CH VOORBURG  
 Leger, M. 2282 HL RIJSWIJK ZH  
 Huijzenga, T. 2285 HK RIJSWIJK ZH  
 Ders, H. A. van 2295 LH KWINTSHEUL  
 Kouwenhoven, B. J. M. 2295 LX KWINTSHEUL  
 Rijckevorsel, M. van 2311 GL LEIDEN  
 Repe, C. A. v. d. 2312 GP LEIDEN  
 Thielen, T. 2316 AC LEIDEN  
 Pelt, J. J. M. van 2324 EA LEIDEN  
 Lange, E. L. de 2353 JB LEIDERDORP  
 Citroen, Charles L. 2396 CK KOUDEKERK a/d RIJN  
 Wilmink, G. M. T. 2402 ALPHEN a/d RIJN  
 Leenes, R. E. 2403 JK Alphen a/d Rijn  
 Duiker, J. H. 2403 PD ALPHEN A/D RIJN  
 Nierop, B. van 2411 BS BODEGRAVEN  
 Hendriks, C. A. 2481 BA WOUBRUGGE  
 Schoone, X. 2502 CC 's-GRAVENHAGE  
 Lagendijk, C. L. W. 2563 GZ 's-GRAVENHAGE  
 Roos, G. J. 2574 SD 's-GRAVENHAGE  
 Ploeg, Arend 2582 LN 's-GRAVENHAGE  
 Voors, J. J. 2502 CE 's-Gravenhage  
 Vreugd, R. de 2 2582 XK 's-GRAVENHAGE  
 Pol, R. van der 2586 TB 's-GRAVENHAGE  
 Heumen, T. van 2587 ED 's-GRAVENHAGE  
 Meij, v. d. L. 2613 BM DELFT  
 Reuvers, D. C. 2624 CM DELFT  
 Meyer, M. C. 2624 LC DELFT  
 Davis, P. 2624 L6 DELFT  
 Lievaart, Hans 2691 DX 's-GRAVENZANDE  
 Valkenburg, C. H. L. 2713 CE ZOETERMEER  
 Meyer, F. V. H. 2726 ZOETERMEER  
 Hoog, G. de 2741 XR WADDINXVEEN  
 Hermans, F. F. M. T. 2771 CC BOSKOOP  
 Glass, A. 2806 DA GOUDA  
 Zwaan, A. G. 2811 AN REEUWIJK  
 Das, P. 2905 AT CAPELLE A/D IJSSSEL  
 Erkelens, G. 2951 BW ALBLASSERDAM  
 Neut, F. v. d. 2982 VN RIDDERKERK  
 Pillay, M. 3008 AE ROTTERDAM  
 Caron, 3061 AR ROTTERDAM  
 Verbeek, M. B. 3061 GN ROTTERDAM  
 Bothof, B. 3076 DL ROTTERDAM  
 Boender, Wio 3116 AG SCHIEDAM  
 Vrijland, L. 3116 AJ SCHIEDAM  
 Bont, N. J. J. M. de 3116 BK SCHIEDAM  
 Schuddebeurs, P. 3133 KR VLAARDINGEN  
 Meij, L. K. v. d. 317 RC LEIDEN  
 Jongema, S. 3171 NB POORTUGAAL  
 Vermeul, M. S. 3174 EN POORTUGAAL  
 Wenink, P. 3203 BG SPIJKENISSE  
 Keriklaan, J. 3206 PR SPIJKENISSE  
 Bakelaar, B. 3257 LJ OOLTGENSPLAAT  
 Jongema, S. 3262 BP OUD BEVERLAND  
 Berg, A. v. d. 3295 LE 's-GRAVENDEEL

Veen, L. van  
 Hoogenhuizen, P.  
 Pinto, M.  
 Naburgh, E. P.  
 Spall, R. M. P.  
 Zwanenburg, P.  
 Scholten, J. C. J. M.  
 Truijens, Kees L.  
 Brinkers, G.  
 Hussen, A. van  
 Rijke, D. de  
 Vries, Jan de  
 MCC NewBrain GG, Voorz.  
 Bode, A. W. D.  
 Voorn, A. G.  
 Jansen, C. D.  
 Koffenaar, I.  
 Regteren, J. van  
 Jansen, J.  
 Born, M.  
 Bruggen, J. v. d.  
 Verbraeken, R.  
 Wijnoltz, Fred  
 Ouden, J. den  
 Erve, H. J. v. d.  
 Kamperman, Th. M.  
 Unlandt, A.  
 Hendriks, P. J. M.  
 Sluijs, Jan  
 Hoogenhuijzen, P.  
 Alba Heijdenrijsk, E.  
 Roos, E.  
 Doesburg, E. J. van  
 Hoepaerden, V. van  
 Weststrate, P. W.  
 Moutwik, Aart S. van  
 Veltema, J.  
 Stroop, Remko  
 Luijk, J. van  
 Koppelaar, R. Y.  
 Kampen, I. van  
 Milliano, Etienne de  
 Velde, N. D. v. d.  
 Luijt, M. M.  
 Eekelen, J. van  
 Kanter, A. A. M. de  
 Dekker, L.  
 Elias, G.  
 Dekker, H. G.  
 Aarts, M. J.  
 Tilburg, H. W. van  
 Wijgerde, K.  
 Bont, R. J. N. M. de  
 Weijters, A. J. M. M.

3295 VC 's-GRAVENDEEL  
 3312 XR DORDRECHT  
 3314 HH DORDRECHT  
 3319 VL DORDRECHT  
 3431 SK NIEUWEGEIN  
 3435 BH NIEUWEGEIN  
 3511 VD UTRECHT  
 3512 HF UTRECHT  
 3514 EJ UTRECHT  
 3562 GH UTRECHT  
 3571 SL UTRECHT  
 3581 SP UTRECHT  
 3581 SP UTRECHT  
 3605 HJ MAARSSSEN  
 3641 AK MIJDRECHT  
 3705 BW ZEIST  
 3712 AR HUIS TER HEIDE UT  
 3721 BR BILTHOVEN  
 3732 EG DE BILT  
 3742 EN BAARN  
 3811 AM AMERSFOORT  
 3811 LR AMERSFOORT  
 3828 VC HOOBLAND  
 3844 MK HARDERWIJK  
 3925 RY SCHERPENZEEL GLD  
 3972 VB DRIEBERGEN RIJSENB.  
 3972 XE DRIEBERGEN RIJSENBURG  
 4051 EX OCHTEN  
 4121 BC VIANEN ZH  
 4142 AG LEERDAM  
 4201 BG GORINCHEM  
 4206 AG GORINCHEM  
 4254 STEENWIJK  
 4285 CB WOUDERICHEM  
 4333 EM MIDDELBURG  
 4337 AD MIDDELBURG  
 4384 BP VLISSINGEN  
 4384 LC VLISSINGEN  
 4386 CC VLISSINGEN  
 4388 HW OOST SOUBURG  
 4414 RS WAARDE  
 4453 BA 's-HEERENHOEVE  
 4475 AG WILHELMINADORP  
 463 VM GOES  
 4726 AE HEERLE  
 4816 CM BREDA  
 4818 BC BREDA  
 4841 LP PRINSENBEEK  
 4847 SR TETERINGEN  
 4901 AV OOSTERHOUT NB  
 4902 BA OOSTERHOUT (N-E)  
 4905 AX OOSTERHOUT  
 5012 GT TILBURG  
 5014 BL TILBURG

Hermans, Jos  
 Haans, A. C. M.  
 Brock, J.  
 Wijk, H. van  
 Blom, P.  
 Velden, M. F. J. A. v. d.  
 Alisic, S.  
 Mijnen, T. J. C. M.  
 Wiersma, G. J.  
 Linden, Hans (J. G. J.) v. d.  
 Vossenbergh, M. W. F. M. v. d.  
 Berkelmaans, R.  
 Maasland Coll. tav F. vd Velde  
 Velden, F. W. J. v. d.  
 Tonies, Ton  
 Kruysbergen, J. W. J. van  
 Neys, Ch.  
 Haanstra, H. B.  
 Jansen, P. M.  
 Hurk, T. v. d.  
 Hurk, Theo W. H. v. d.  
 Welten, H. Th.  
 Lemmen, G.  
 Pronk, C. M.  
 Han, W. A.  
 Russel, A. J. M.  
 Kurstjens, Jan  
 Steinmann, H.  
 Sonnemans, A. J. F.  
 Schoeber, M.  
 Muller, E. G.  
 Ariaans, W. H.  
 Janssen, N.  
 Dassen, F.  
 Meertens, A.  
 Prins, A.  
 Drongelen, R. van  
 Hauptmeijer, W. M.  
 Hauptmeyer  
 Balenans, H. C. G. C.  
 Christiaans, M.  
 Prins, J. H.  
 Pleaen, M. M. F.  
 Elema, P.  
 Kok, A. de  
 Naelten, Mark A. M. G. van  
 Spek, W.  
 Wit, J. M.  
 Pynappel, Leon  
 Hoppenbrouwers, G. A. J.  
 Verhoeven, N. G.  
 Reus, B. M. de  
 Hoffman, D.  
 San Salvatoreaavo

5001 GG Tilburg  
 5032 BB TILBURG  
 5037 DG TILBURG  
 5042 BH TILBURG  
 5044 ND TILBURG  
 5051 BT GOIRLE  
 5109 TZ 's-GRAVENMOER  
 5151 GJ DRUNEN  
 5161 GH SPRANGCAPELLE  
 5171 DG KAATSHEUVEL  
 5212 PH 's-HERTOEGENBOSCH  
 5271 NK ST. NICHIELSGESTEL  
 5343 XK OSS  
 5343 XK OSS  
 5368 AJ HAREN NB  
 5434 SZ CUYK  
 5473 GZ HEESWIJK DINTHER  
 5550 AB VALKENSWAARD  
 5641 MK EINDHOVEN  
 5703 XR HELMOND  
 5703 XR HELMOND  
 5708 JD HELMOND  
 5712 BR SOMEREN  
 5737 EG LIESHOUT  
 5752 BE DEURNE  
 5802 AD VENRAY  
 5861 BA WANSUM  
 5914 VR VENLO  
 5922 XJ VENLO  
 5953 LS REUVER  
 6031 BV NEDERWEERT  
 6121 JE BORN  
 6121 KP BORN  
 6171 KK STEIN (L.)  
 6212 CM MAASTRICHT  
 6214 AR MAASTRICHT  
 6227 TD MAASTRICHT  
 6228 BT MAASTRICHT  
 6228 BT MAASTRICHT  
 6267 B6 CADIER EN KEER  
 6301 EX VALKENBURG (L.)  
 6305 AZ SCHIN OP GEUL  
 6336 XN HULSBURG  
 6372 MX SCHAESBURG  
 6433 AJ HOENSBROEK  
 6524 LM NIJMEGEN  
 6531 CK NIJMEGEN  
 6532 XL NIJMEGEN  
 6537 TB NIJMEGEN  
 6542 RM NIJMEGEN  
 6545 NK NIJMEGEN  
 6551 AP WEURT  
 6561 EX GROESBEEK  
 6591 GN DENEKAMP

Gendt, P. M. A. G. van  
 Tolkamp, G. J. A.  
 Kramer, P.  
 Feddes, R. A.  
 Draagt, A.  
 Aalderink, R. H.  
 Roelofs, R. A.  
 Dieckmann, P. A.  
 Kuiken, J. C. J.  
 Koetsier, A.  
 Berge, W. F. ten  
 Hendriks, H.  
 Horstink, F. H.  
 Schuurman, H. J. G.  
 Rienks, R.  
 Steverink, M.  
 Scholengemeenschap, Chr.  
 Kreuzen, A.  
 Scholengemeenschap Marianu  
 Aloysiusmavo  
 Timmerman, W.  
 Weisman, B. G.  
 Alma, P. H.  
 Scholengew. Isendoorn  
 Isendoorn, S. G.  
 Wijnen, F. M.  
 Commandeur, N. J. M.  
 Bijleveld, Th. J.  
 Manrooy, W. van  
 Krops, J. B.  
 Zantinge, R.  
 Doesschate, J. F. ten  
 Knipstra, Sietze  
 Wilke, R. A. P.  
 Ruyter, G. de  
 Lubberhuizen, J. W.  
 Marssinkhof, B. H. J.  
 Bolt, J. H.  
 Wisselink, M.  
 Wit, F. de  
 Computerwinkel Oost Ned.  
 Meide, H. v. d.  
 Riswick, M.  
 Land, K.  
 Spiele, H.  
 Kapper, B. J.  
 Jong, W. O. de  
 Heuw, H. B. ten  
 Smit, T. A.  
 Wisselink, M. A.  
 Duthler, J. W.  
 Scholengemeenschap Zuid  
 Braspenning, N. P. A.  
 Tax, Henny

6601 SC MIJCHEN  
 6602 XA MYCHEN  
 6661 BN ELST (Gld.)  
 6706 CC WAGENINGEN  
 6711 AD EDE (GLD)  
 6718 WP EDE GLD.  
 6813 JL ARNHEM  
 6842 AP ARNHEM  
 6865 BH DOORWERTH  
 6874 BR WOLFFHEZE  
 6931 CV WESTERVOORT  
 6942 WD DIDAM  
 6991 TR RHEDEN  
 7004 CX DOETINCHEM  
 7051 ZS VARSSEVELD  
 7061 BH TERBORG  
 7122 AN AALTEN  
 7122 KH AALTEN  
 7141 EA GROENLO  
 7141 KE GROENLO  
 7151 MK EIBERGEN  
 7152 EP EIBERGEN  
 7161 PH NEEDE  
 7200 AH WIJNEN  
 s 3437200 AN ZUTPHEN  
 7201 HI ZUTPHEN  
 7206 BZ ZUTPHEN  
 721 WT GRONINGEN  
 7221 AC STEENDEREN  
 7301 BG APELDOORN  
 7311 AA APELDOORN  
 7313 AP APELDOORN  
 7412 XM DEVENTER  
 7431 CG DIEPENVEEN  
 7441 HS NIJVERDAL  
 7451 NL HOLTEN  
 7491 GN DELDEN  
 7511 DP ENSCHEDE  
 7511 JK ENSCHEDE  
 7511 LK ENSCHEDE  
 7514 BC ENSCHEDE  
 7514 CR ENSCHEDE  
 7514 DW ENSCHEDE  
 7522 BT ENSCHEDE  
 7522 HL ENSCHEDE  
 7522 KM ENSCHEDE  
 7522 LD ENSCHEDE  
 7524 AD ENSCHEDE  
 7524 CW ENSCHEDE  
 7531 HP ENSCHEDE  
 7532 RK ENSCHEDE  
 7541 VG ENSCHEDE  
 7542 EJ ENSCHEDE  
 7545 NC ENSCHEDE

Essen, W. van  
 Galama, Ysbrant Hays  
 Meenderink, J.  
 Hoek, R. G.  
 Moerthuis, J.  
 Koekkoek, W.  
 Laak, J. J. C. ter  
 Mulder, B.  
 Oude Nijhuis, J. G. M.  
 Johannink, C. F.  
 Meijer, R.  
 Scholengen. Erasmus, Rijks  
 Walraven, C. D.  
 Kok, A. A.  
 Melchior Winhef MAVO  
 Meyerling, Tom  
 Dijkstra, J. F.  
 Vermaet, Rinus  
 Seerts, K.  
 Gelden, F. G. van  
 Scholten, H.  
 Hevius, Bert  
 Ramaker, W. P. J.  
 Bakker, G.  
 Bakker, D.  
 Hel'sloot, H. D. F.  
 Verhoef, J.  
 Sanders, A. F. M.  
 Hoeve, Jan Willem  
 Laprot, L. G. E.  
 Gelden, R. E. van  
 Berg, J. v. d.  
 Heuumen, T. L. van  
 Bakker, T.  
 Koekoek, B.  
 Reylevelt, G.  
 Nentjes, I.  
 Oldert, J. H.  
 Popkema, G.  
 Wortelboer, C.  
 Lemeer, P. H. J.  
 Alting, M.  
 Jong, L. de  
 Pappert, N.  
 School, Gem. LEAD  
 Johanneschool  
 Visser, P.  
 Snedes, I. J.  
 Bakker, Bart F.  
 Snabilie, R.  
 Schutgens, N.  
 Boonstra, G.  
 Brons, J. H.  
 Brom, P.

7546 AD ENSCHEDE  
 7551 HH HENGELO  
 7552 GT HENGELO (G.)  
 7558 BC HENGELO (G)  
 7558 NK HENGELO (G)  
 7558 RJ HENGELO (G.)  
 7576 TJ OLDENZAAL  
 7577 GJ MULDERZAAL  
 7581 EZ LOSSER  
 7591 TR DENEKAMP  
 7602 HA ALMELO  
 7602 HT ALMELO  
 7626 LW HERTME  
 7631 AT OOTMARSUM  
 7631 BR OOTMARSUM  
 7612 HS EMMEN  
 7661 AS OOSTERHESSELEN  
 7944 VM MEPEL  
 7951 EC STAPHORST  
 8011 HC ZWOLLE  
 8011 MS ZWOLLE  
 8011 XM ZWOLLE  
 8014 ZR ZWOLLE  
 8032 BB ZWOLLE  
 8032 BB ZWOLLE  
 8072 NN ZWOLLE  
 8070 AB NUNSPEET  
 8081 JT ELBURG  
 8096 AS OLDEBROEK  
 8096 ZG OLDEBROEK  
 8101 GB RAALTE  
 8172 AM VAASSEN  
 8181 KC HEERDE  
 8251 HR DRONTEN  
 8266 GA KAMPEN  
 8302 BW EMMELOORD  
 8321 AS URK  
 8332 BB STEENWIJK  
 8426 CX APPELSCHA  
 8442 BJ HEERENVEEN  
 8536 TD OOSTERZEE  
 8608 VV SNEEK  
 8661 JE HARLINGEN  
 8861 JE HARLINGEN  
 8861 JE HARLINGEN  
 907 EL RIDDERKERK  
 9247 BE URETERP  
 9257 RS NOORDBERGUM  
 9257 WP NOORDBERGUM  
 9351 DB LEEK  
 9355 TK MIDWOLDE (LEEK)  
 9405 GK ASSEN  
 9462 SB GASSELTE  
 9477 RH ZUIDLAREN

Born, P. 9477 RH ZUIDLAREN  
 Bruins, Wenzel 9602 AJ HOOGEZAND  
 Wierenga, W. 9602 NP HOOGEZAND  
 Altenburg, B.S.F. 9711 KS GRONINGEN  
 Blik, H.J. 9718 RT GRONINGEN  
 Hoogendijk, J. 9722 GK GRONINGEN  
 Verhage, Mw. N.B. 9724 AW GRONINGEN  
 Seinhorst, B.H. 9724 NA GRONINGEN  
 Kiers, Jan 9727 CK GRONINGEN  
 Emerencia, S.F. 9727 NL GRONINGEN  
 Kokhuis, A. 9741 CE GRONINGEN  
 Drijver, Evert 9742 LH GRONINGEN  
 Vreven, René B-2480 MOL (BELGIE)  
 Houwer, Wim de B-2610 WILRIJK (BELGIE)  
 Vinck, R. B-2730 ZWIJNDRECHT (BELGIE)  
 Droogmans, Guillaume J.J. B-3060 BERTEM (BELGIE)  
 Hennis, Frans B-3110 ROTSELAAR, BELGIE  
 Jas, D.H. B-3761 LANAKEN (BELGIE)  
 Vermeulen, Jef B-8200 BRUGGE (BELGIE)  
 Commissaris, P. B-9720 DE PINTE (BELGIE)  
 Sieghenhorst, Astrid LEIDERDORP  
 Frenkel, E.J. V-30 32 HINTERKAPPELEN (ZWIJ)

## Ledenlijst: Sleutel op naam

Aald 6718 WP, Aart 4901 AV, Aker 2025 ZD, Alba 4201 BG, Alba 1073 JX, Alis 5109 TZ, Alma 7161 PH,  
 Aloy 7141 KE, Alite 9711 KS, Alti 8608 VV, Aria 6121 JE, Bake 3257 LJ, Bakk 9257 WP, Bakk 9632 BF,  
 Bakk 0032 BB, Bakk 1011 AZ, Bakk 8251 HR, Bale 6267 BG, Ball 1399 GK, Bart X-00 00, Beim 1473 EL,  
 Berg 3295 LE, Berg 8172 AM, Berg 6931 CV, Berk 1541 GV, Berk 5271 NK, Beuk 1054 LS, Bevi 8303 BW,  
 Bijl 721 WT, Blee 1012 CF, Blik 9718 RT, Blou 5044 ND, Bode 3605 HJ, Boen 3116 AG, Boer 1015 HR,  
 Bors 1861 VB, Bote 1301 EE, Both 3076 DL, Bouw 1151 ER, Bras 7542 EJ, Bree 1963 AV, Brin 3514 EJ,  
 Broc 5037 DG, Broe 9477 RH, Bron 9462 SB, Brug 3811 AM, Brui 1106 GE, Brui 9602 AJ, Card 3061 AR,  
 Chri 6301 EX, Citr 2396 CK, Cocu 1024 NL, Coma 7206 BZ, Coma B-9720, Conc 7514 BC, Corb 1181 BZ,  
 Das, 2905 AT, Dass 6171 KK, Davi 2624 LG, De B 2240 AA, Dege 1056 XB, Dekk 4847 SR, Dekk 4818 BC,  
 Denn 1505 WP, Diec 6842 AF, Dijk 2241 SB, Dijk 7861 AS, Ditz 1422 EJ, Ditz 1403 CG, Does 4254 ST,  
 Boes 7313 AP, Dore 1401 BM, Drag 6711 AD, Drij 9742 LH, Dron 6227 TD, Droc B-3060, Duik 2403 PD,  
 Duij 1509 BS, Duso 2172 JV, Duth 7532 RK, Duyv 1562 TP, Eeke 4726 AE, Egge 1444 VN, Elem 6372 MX,  
 Elff 1121 CP, Elia 4041 LP, Emer 9737 NL, Erke 2951 BW, Erve 3925 BX, Esse 3546 AD, Fedd 6706 CC,  
 Foek 1015 EA, Fren V-30 32, Gala 7551 HH, Geer 7951 ED, Geid 8011 CG, Geid 8101 GE, Gend 6601 SC,  
 Gers 1186 EJ, Glas 2006 DA, HCD 1073 JX, HCD 3581 SP, Haan 5032 BR, Haan 5550 AB, Hage 1216 AA,  
 Haas 5752 BE, Hara 1241 LV, Haud 6228 BT, Heka 1211 KX, Hets 8032 NN, Hend 2481 BA, Hend 6942 WC,  
 Hend 4051 EX, Henn B-3110, Hema 5014 JD, Hema 2771 CC, Heum 2587 ED, Heum 8181 KC, Heum 7524 AD,  
 Heyd 2032 ZN, Hidd 1273 XJ, Hillh 1243 KC, Hoag 4285 CB, Hoek 7550 BC, Hoek 1098 KL, Hoey 8096 AS,  
 Hoff 6561 EX, Hols 2242 XP, Hoog 2741 XR, Hoog 9722 GK, Hoog 3312 XR, Hoog 4142 AG, Hoog 2105 XV,  
 Hopp 6542 RM, Hors 6991 TR, Houw 6228 BT, Hout 1222 RD, Houw B-2610, Hovi 8011 YM, Huiz 2205 HK,  
 Hurk 5793 XR, Hurk 5703 XR, Huss 3562 GH, Isen 7200 AN, Jans 3705 BW, Jans 3732 ES, Jans 5641 MK,  
 Jans 6121 KP, Jas, B-3761, Joha 2181 XE, Joha 907 EL, Joha 7591 TR, Jong 8061 JE, Jong 7522 LG,  
 Jong 3171 NB, Jong 3262 BP, Kali 1834 AR, Kamp 4414 PS, Kamp 3972 VE, Kand 1782 GD, Kant 4816 CM,  
 Kapp 7522 KM, Kapp 8061 JE, Kerk 3206 PR, Kier 9727 CK, Kni, 7412 XM, Knib 1871 EV, Kobb 2274 CM,  
 Koek 7550 RJ, Koek 8266 GA, Koff 6874 BK, Koff 3712 AR, kok, 7631 AT, Kok, 1688 CN, kok, 6433 AJ,  
 Kokh 9741 CE, Koni 2102 BE, Koni 2161 RV, Kool 1784 AG, Kopp 4380 HW, Kouw 2295 CX, Kram 6661 BM,  
 Krau 7122 KH, Krol 1095 AB, Krop 7301 BG, Kruij 5434 SZ, Kuik 6865 BH, kuis 5861 BA, Laak 7576 TJ,  
 Laer 1241 XE, Lage 1823 EV, Lage 2563 GZ, Land 7522 BT, Lang 2353 JB, Laor 8096 ZG, Leen 2403 JY,  
 Leeu 1382 JX, Leeu 1852 KV, Lege 2282 HL, Lemm 8536 TD, Lemm 5712 BR, Leth 1013 TD, Liew 2691 DX,  
 Lind 5171 BG, Lohs 1104 AK, Lubb 7451 NL, Luij 4386 CC, Luij 463 VM, Maas 5343 XK, Mari 1211 KH,  
 Meen 7552 GT, Meer 6212 CM, Meij 317 KC, Meij 2613 BM, Meij 7602 HA, Melc 7631 GR, Meye 2726 ZD,  
 Meve 2624 LC, Meve 7012 HS, Mill 4453 BA, Misp 5151 GJ, Morg 1231 AP, Mors 7491 GN, Mouw 4327 AD,  
 Muld 7577 GJ, Mull 6031 BV, Nabu 3319 VL, Nael 6524 LM, Nent 8321 AS, Neut 2982 WN, Neys 5473 GJ,  
 Nier 2411 BS, Nieu 1182 HM, Nieu 1779 EL, Nurm 1183 AC, Oers 2295 LH, Oide 8332 BB, Oost 1412 GK,  
 Osbo 1422 JS, Oss, 1070 GB, Oude 7581 EZ, Oude 3844 MK, Ouds 1031 LL, Paas 1829 HJ, Pell 2266 EG,  
 Pell 1098 CK, Pelt 2324 EA, Pers 1412 AN, Piet 1102 CA, Pili 3008 AE, Pint 3314 HH, Ploer 1702 AT,  
 Ploer 6356 XN, Pol, 2586 TB, Popk 8426 CX, Prin 6214 AR, Prin 6305 AZ, Pron 5737 EG, Pyna 6537 TB,  
 Raij 1442 NG, Rama 8014 ZR, Regt 3721 BE, Rein 1962 SG, Reke 1074 JH, Reme 1011 TJ, Repe 2312 GP,  
 Reus 6551 AP, Reuv 2624 CM, Rhee 1217 TE, Ridd 1070 WV, Rien 7051 ZS, Rijs 2311 GL, Rijs 3571 SL,  
 Rijs 1141 RM, Rijs 1141 JA, Risw 7514 DW, Roel 6013 JL, Roos 4206 AG, Roos 2574 SD, Rosk 1094 JG,  
 Russ 5002 AD, Ruth 1056 AS, Rutt 1221 EE, Ruyt 7441 HS, San 6591 EN, Sand 0001 JT, Sch, 1782 ND,  
 Scho 5953 LS, Scho 7602 HT, Scho 7200 AH, Scho 7141 ER, Scho 7541 V6, Scho 7122 AW, Scho 1060 MR,  
 Scho 8011 MS, Scho 3511 VD, Scho 8061 JE, Scho 2502 CC, Schu 3133 XR, Schu 9355 TK, Schu 7004 CX,  
 Scul 1073 HG, Sein 9724 NA, Sleg LEIDERD, Slui 4121 BC, Smed 9257 RS, Smit 7524 CW, Snab 9351 BK,  
 Sonn 5922 XJ, Spa, 3431 SK, Spek 6531 CK, Spie 7522 HL, Stei 5914 VR, Steu 1381 AJ, Stev 1092 JY,  
 Stev 7061 BH, Stoi 1504 AT, Stro 4384 LC, Tax, 7545 NC, Tech 1025 RX, Tech 2105 SX, Tech 1074 XL,  
 Thie 2316 AC, Till 4902 BA, Timm 7151 MK, Tolk 6602 XA, Toni 5368 AJ, Trui 2512 HF, Unla 3972 XB,  
 Uylid 1353 CH, Valk 2713 CE, Vals 1215 RL, Veen 3295 VC, Veld 4475 AG, Veld 5343 XK, Veld 5051 BT,  
 Velt 4384 BP, Verb 3061 GN, Verb 3811 LR, Verh 9724 AW, Verh 8070 AB, Verh 6545 NK, Verm 7944 VM,  
 Vera 3174 EN, Vera B-8200, Vinc B-2730, Vism 9247 BG, Voor 3641 AK, Vooj 2582 VA, Vos, 1277 GJ,  
 Voss 5212 PH, Vree 1462 GA, Vreu 2582 XK, Vrev B-2480, Vrie 1484 PC, Vrie 3581 SP, Vrij 3116 AJ,



Waar 7626 LW, Wann 7221 AG, Weid 7514 CR, Weij 5014 BL, Weim 7152 GP, Weit 5708 JD, Weni 3203 BG,  
 Werk 1078 GL, West 4333 EM, Wiel 1214 CP, Wier 1214 CP, Wier 9602 NP, Wier 5161 GH, Wijs 4905 AX,  
 Wijk 5042 BH, Wijn 7201 HI, Wijn 3828 VC, Wijk 7431 CG, Wilm 2402 AL, Wiss 7511 JK, Wiss 7531 HP,  
 Wit, 7511 LK, Wit, 6532 XL, Woer 7558 NK, Wort 8442 BJ, Woud 1121 CP, Wout 1703 KB, Zane 1509 ZA,  
 Zant 7311 AA, Zome 1315 CL, Zuur 1000 BP, Zwaai 2811 AN, Zwan 3435 BH, Zwar 2201 BN,

dealerlijst

C.T.M.	J.Wattstr. 66	1006 AT Amsterdam	020-941654
Wolfkamp comp.boekh.	Weteringschans 221	1007 KG Amsterdam	020-278931
Solution Data Syst.	Scheldestraat 53	1078 GG Amsterdam	020-727757
Maban Ned.	Ruytersweg 60a	1211 KX Hilversum	035-233769
Comp.house Bussum	Herenstraat 19	1404 AB Bussum	02159-11285
Taylor El.Techn.BV	IJsselstraat 15	1784 VN Den Helder	02230-22346
Computational BV	Engelandlaan 1222	2003 EL Haarlem	023-359544
Comp.house Haarlem	Raamsingel 32	2012 DT Haarlem	023-314032
Informatic Service	Dotterbloemkreek 65a	2350 AC Leiderdorp	071-411230
Electr.Centr.Delft	Voldersgracht 26	2511 EV Delft	015-134429
Fa. Rueb	Fred.Hendriklaan 141	2582 BZ Den Haag	070-559919
St.Bilt Computer	Hessenweg 228	3731 JP De Bilt	030-760519
Brain Wave	Telgterweg 226	3853 NS Ermelo	03417-56346
A.M.S.	Arkelse Onderweg 95	4206 AG Gorinchem	01830-22967
Imha BV	Kreukelmarkt 7	4461 HW Goes	01100-13941
Compart Data	Bernardusplein 17	5042 HP Tilburg	013-633589
Fa. Hamco	Langestraat 3	5752 BE Deurne	04930-16107
PrinService	Mauritiusingel 61	6305 AZ Schin op G.	04459-2319
Vogelezang BV	Akerstraat 19	6411 GV Heerlen	045-716055
Kroes Kasregisters	Hertogstraat 58	6511 SC Nijmegen	080-231873
Com Teach	Beethovenlaan 38	6865 EB Doorwerth	085-332772
Audilog BV	Leliestraat 158	7004 CX Doetinchem	-
Radio Nijhuis	De Heurne 30-32	7511 GW Enschede	053-315169
Comp.w.Oost Nederl.	Deurningerstraat 3a	7514 BC Enschede	053-337296
Radio Nijhuis	Telgen 11	7551 CL Hengelo	074-917567
Logical	Schneiderstraat 15	7555 LZ Hengelo	074-917342
Radio Nijhuis	Marktstraat 12	7607 HD Almelo	05490-19191
Comp.h.Zuidwolde	Mettelerweg 24	7921 VK Zuidwolde	05287-1458
Micro Source	Ossemarkt 25	8001 BE Zwolle	038-223698
Radio Nijhuis	Oude Vismarkt 29	8011 TB Zwolle	038-213804
Inst.bedr.Ferwerd	Leeuwerikstraat 127	8916 CC Leeuwarden	058-121060
Comp.h.Leeuwarden	Kleine Kerkstraat 19	8911 DL Leeuwarden	058-134746
Telec BV	Steentilstraat 36/40	9511 GP Groningen	050-129374/141616
De Computerwinkel	Oostersingel 1	9713 EW Groningen	050-131427

AANMELDING NIEUW LID / BESTELKAART

Naam en voorletters :.....  
Straat :.....  
Postc. en Woonpl. :.....  
Telefoon :.....

- is reeds lid van de H.C.C., lid no:..... en geeft zich hierbij op als lid van de NewBrain G.G. (gratis)
- geeft zich op als lid van de H.C.C. en NewBrain G.G.
- geeft zich op voor regelmatige toezending van NEWBRAIN-ON-LINE (prijsstelling is voorlopig Fl. 5,50 per 50 pag.)

DATUM:

HANDTEKENING:

AANMELDING NIEUW LID / BESTELKAART

Naam en voorletters :.....  
Straat :.....  
Postc. en Woonpl. :.....  
Telefoon :.....

- is reeds lid van de H.C.C., lid no:..... en geeft zich hierbij op als lid van de NewBrain G.G. (gratis)
- geeft zich op als lid van de H.C.C. en NewBrain G.G.
- geeft zich op voor regelmatige toezending van NEWBRAIN-ON-LINE (prijsstelling is voorlopig Fl. 5,50 per 50 pag.)

DATUM:

HANDTEKENING: