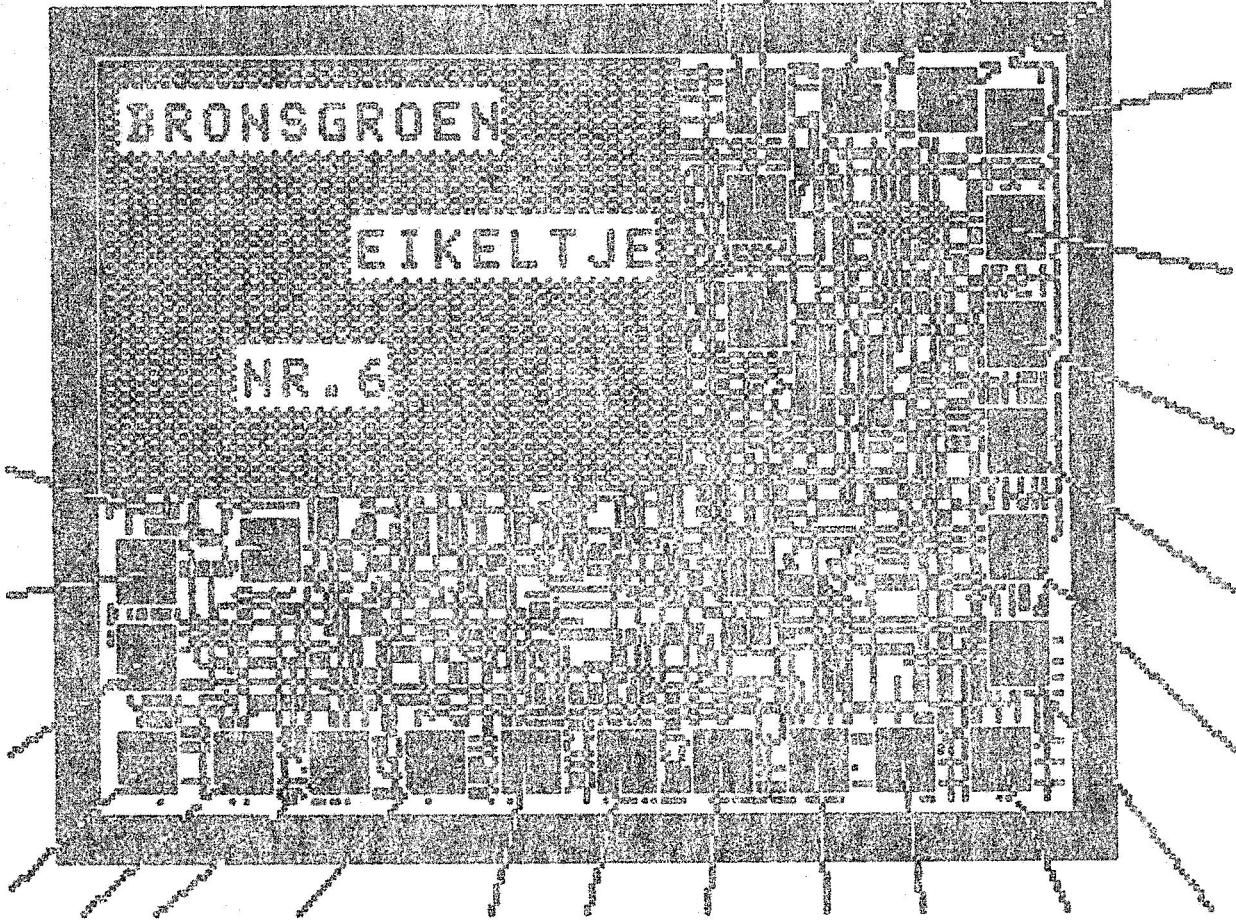


I.C.



ACORN COMPUTERCLUB

LIMBURG

\*\*\*\*\*  
PROLOOG C. CAMPERS  
\*\*\*\*\*

Het heeft ditmaal wel enige moeite gekost, om ons Eikeltje vol te krijgen.

Van de ene kant verklaarbaar vanwege de vakantie, van de andere kant zou men kunnen stellen, dat er weinig gemaakt of ingestuurd wordt.

Toen de redactie de laatste keer bij elkaar kwamen, bleek, dat het verzamelen van copy voor de komende tijd onze speciale aandacht vergt.

Stuurt U, wat U hebt gemaakt, rustig in. Wij, redacteuren, zorgen voor de rest.

Zo, deze noodkreet, als je wilt, moest mij toch even van het hart.

Ik neem aan, dat er tussendoor driftig gecomputerd wordt op alle mogelijke fronten: techniek, administratie, boxen, uitbreidingen, aanpassingen en ga zo maar door.

Het moet dan ook mogelijk zijn om, door onderlinge uitwisseling van een en ander, tot een bevredigende uitgave van ons blad te komen, de ons intussen dierbaar geworden (bronsgroene) eikel.

Ik zag eens in het bos een eekhoorn eikels verzamelen. De gedachte dat de redactie toch iets gemeen had met het dier, schoot mij door het hoofd.

Allaen vergaren wij natuurlijk geen eikels, maar iets waaraan de eikel toch niet vreemd is.

Eikels zijn stevig en lang houdbaar. De redactie eigenlijk ook en als ze maar voer krijgt, kan ze veel uitdelen.

En dat is iets, wat zij het liefst doet.

Kom, Acornvrieden, slaan wij de handen ineen.

Wij doen er toch iets aan ???!!!

Dus op naar dikke bronsgroene eikels, die ons de winter door helpen, net zoals bij de eekhoorn de eikel!

Dus beste Acornvrieden laat eens wat van je horen.

C. Campers  
Hoofdredacteur:  
"Het Bronsgroen Eikeltje"

\*\*\*\*\*  
AANP. JOY STICK RHINOCEROS J VD BLINK  
\*\*\*\*\*

Het spelletje "RHINOCEROS" beviel met met de toetsbediening niet geheel. Omzetting voor de joy-stick bracht de volgende wijzigingen:

- 220 IF?Z=237; IPV. IF ?T=57
- 230 IF?Z=231; IPV. IF ?T=41
- 240 IF?Z=243; IPV. IF ?T=45
- 250 IF?Z=249; IPV. IF ?T=34
- 260 IF?Z=239; IPV. IF ?T=53
- 270 IF?Z=231; IPV. IF ?T=46
- 280 IF?Z=247; IPV. IF ?T=43
- 290 IF?Z=253; IPV. IF ?T=39

Toegevoegd werd : 215 Z=?#B001

Voor een goede werking moest regel 740 ook veranderd worden:

Na LI.A werd toegevoegd: ;N=1;R=0;E.70 (R. vervalt dus)

\*\*\*\*\*

HET ONDERBREKEN VAN BASIC-PROGRAMMA'S W.FEYER DELFT

\*\*\*\*\*

Onderstaand programma is overgenomen uit ATOMIX Acorn  
Computerclub Delft

De meeste computers geven bij het onderbreken van een programma een of andere melding op het beeldscherm, bijv./BREAK AT LINE 230/.

Het hierna volgende programma laat onze ACORN het zelfde doen.

```

10 DIM LL4
20 F.Q=1T02
30 P=#3000;I
40:LL0 CMP#3E;BNELL4
50 LDA#0;CMP1;BNELL1
60 CMP2;BEQLL3
70:LL1 LDA#1;CMP#6;BEQLL3
80 JSR#F7D1;J
90 P=LL1+22
100 *(P-13)="STOP IN LINE"
110 ?(P-1)=#EA;I
120:LL2 LDA#0;STA4;STA#43
130 STA#34;LDA1;STA#16
140 LDA2;STA#25;LDA#FE
150 STA#209;LDA#52;STA#208
160 JSR#C589;LDA#LLO/256
170 STA#209;LDA#LLO%256
180 STA#208
190:LL3 LDA#3E
200:LL4 JMP#FE52;J;N.
210 ?#208=LLO%256
220 ?#209=LLO/256;END

```

OPMERKINGEN:

P in regel 30 mag ook een andere waarde hebben.

Nem de regels 90,100 en 110 precies over.

Bij verandering van de string in regel 100 moet het getal 22 in regel 90 verhoogd worden met het aantal karakters dat de nieuwe string langer is dan de oude, resp. verlaagd worden als de nieuwe string korter is.

Het zelfde geldt voor het getal 13 in regel 100.

Na een BREAK kan het programma gestart worden door de pointers op #208 en #209 naar het begin van het gecompileerde programma te laten wijzen(in dit geval #3000).

Let op dat dit programma niet functioneert als men met andere programma's werkt die ook gebruik maken van write character pointers (zoals grmod van de AXR1).

\*\*\*\*  
TIPS  
\*\*\*\*

1: Tip voor mensen met onleesbare cassettes.

Leatst kreeg ik van het bandjesarchief een cassette die volkomen onleesbaar was voor mijn Atom. Na goed luisteren bleek dat de toonhoogte te hoog was. Het was dus met een te lage snelheid opgenomen. Gelukkig kun je hier heel eenvoudig iets aan doen, je moet wel een JOSSOX hebben. Nu komt het programma. Het kost een regel type-werk...

```
RELOC#FBEE, #FC2B, #3C00; ?#3C14=#4B; !#214=#3C00
```

Het gaat hier om 300 Baud. Als u op #3C00 geen RAM heeft, kunt u een ander adres kiezen. Met de waarde van #3C14 moet u misschien wat experimenteren. Standaard is dit #53. Voor 1200 Baud moet zoiets ook te maken zijn. Wie nu nog problemen met het laden van andermans cassettes heeft mag het zeggen.

2: Wil je na het laden van een (lang) programma wakker gemaakt worden, save het dan als volgt:  
\$SAVE "Naam" (start), (eind), FDI  
en laadt het met \$RUN "Naam".

3: Vaak werkt het erg makkelijk en snel als je in machinetaal een beslissing neemt (b.v. door het testen van een bit) en op grond daarvan DIREKT naar een stuk basic springt. Om te zorgen dat zo'n stuk basic niet bij elke verandering van het programma van plaats verandert, zet je het op het begin van een pagina verderop in het geheugen.

Staat je programmastukje op pagina #30 (?18=#30), roep je dat als volgt in machinetaal aan:  
LDA#3; STA5; LDA@#30; STA6; JMP#C2F2.

Helaas deze constructie is de schrik van elke liefhebber van "Gestructureerde Programme's", maar veel sneller dan de vrij trage IF...THEN constructie.

Uit "DE CURSOR" Brabant Oost Reinier Gerritsen.

```
*****
KIENEN 4 B.TOSSAINT.
*****
```

Kienen 4 biedt U de mogelijkheid om m.b.v. de ACORN ATOM de nummers te laten trekken. Deze nummers zijn alleen waarteneem door de "Afroeper".

In het Archief bevindt zich sinds kort ook "KIENEN 5" van B.TOSSAINT.

Kienen 5 toont de getrokken nummers in grootbeeldformaat.

```
10REM KIENEN4
20P.$12" KIEN PLEZIERIE""
30P."DE TREKKING WORDT UITGEVOERD""
40P."M.B.V. EEN ACORN ATOM COMPUTER""
50P." VEEL SUCCES "" ;F.I=0 TO 200;WAIT;N.
60P.$12;P." "WILT U NUMMERS CONTROLEREN""
70P."DRUK DAN "" B VOOR DE BEGIN-STAND"
80P." " G VOOR GETROKKEN NUMMERS"
90P." " C VOOR TREKKING IN VOLGORDE"
100F.I=0 TO 200;WAIT;N.
110P." "return VOOR EEN TREKKING"
120P." " N VOOR EEN NIEUW SPEL"
130DIM A(91), B(91), C(91), K(64)
140F.T=1 TO 91;A?T=T;B?T=0;C?T=0;N.T
150T=1;S=3
160F P." ";IN.$K;P." "
170IF?K=CH"B";G0S.a;G.f
180IF?K=CH"G";G0S.b;G.f
190IF?K=CH"C";G0S.c;G.f
200IF?K=CH"N";G.140
220REM KIESROUTINE
230R=A.R.%90+1
240IFA?R=0;G.230
250REM OVERZETTEN GETROKKEN BETAL
260A?R=0;B?R=R;C?T=R
280IFT<=90;G.d
290P." ""NIEUW SPEL OF EINDE ?"" "DRUK N OF E "" ;IN.$K
300IF ?K=CH"N";G.140
310P." " "E.
320P.$12$14;P." "" DE TE TREKKEN NUMMERS ZIJN""
330P." "" "" ;W=A
340E M=1;L=10
350DD
360 F.I=M TO L
370 P.$?I
380 N.;F."
390 M=M+10;L=L+10
400U. L=100
410R.
420P.$12$14"DE GETROKKEN NUMMERS ZIJN""
430P." "" "" ;W=B
440G0S.e
450R.
460P.$12$14"DE VOLGORDE VAN TREKKING IS""
470P." "" "" ;W=C
480G0S.e
490R.
500P." "TREKKING NO"99 T 99"NUMMER"99 R
502 IF T=90;G.g
505T=T+1
510G.f
```

\*\*\*\*\*  
GTHLOC C. CAMPERS  
\*\*\*\*\*

Het volgende programma is weer echt iets voor de zendamateurs onder ons.

Het programma berekent via de GTH locators de:  
coördinaten van de plaatsen in decimalen  
coördinaten " " " in graden, minuten en seconden  
afstand tussen de plaatsen  
de eigen antenne richting  
de antennerichting van de ander

Zoals U ziet kan het programma nogal wat.

```
10 REM GTH AFSTANDSBEREKENING;#32CA
20 REM B.J.V/D MORP, PEOWOR, 3-4-1980, STM-I
30 REM VOOR ACORN ATOM BEWERKT DOOR PAOCCR EN PAOWCR ROERMOND
31 REM 1-9-1982; VERGIE 1-8-1984
40 P.#12
50 F.W=1T031;P.#A3;N.;P.;DOP.#DF;U.C.=4
62 P."gth"#128"afstandsberkening"
65 DOP.#DF;U.C.=31;P.;F.W=1T031;P.#D0;N.;P."""
75 P." (BEST 73 ES GD DX DE PAOCCR)""""
78 P."DRUK SPATIEBALK !!!"
79 LINK #FFE3
80 P.#12
100 DIM RR(10),SS(10),B(5),Q(5),A(3),G(3)
120 RR(1)=2;SS(1)=0;RR(2)=3;SS(2)=0;RR(3)=3;SS(3)=1
121 RR(4)=3;SS(4)=2;RR(5)=2;SS(5)=2
130 RR(6)=1;SS(6)=2;RR(7)=1;SS(7)=1
131 RR(8)=1;SS(8)=0;RR(9)=0;SS(9)=0;RR(10)=2;SS(10)=1
140 P.#7;IN."MYN GTH-LOCATOR "#B
180 GQ=#B;GOS.300;GOS.590
190 XU=XX;XV=XY
200 P.#7;IN."ZYN GTH-LOCATOR "#G
210 GOS.300;GOS.590
220 XZ=ZY-XV
230 XD=C.XZ#C.XU#C.XX+SINXU#SINXZ
240 XD=-ATN(XD/SGR(-XD#XD+1))+PI/2
245 XD=XD
250 XD=XD#111.2#180/PI
260 N=X(ZD+0.5);@=1
265 P.#7#7#7
270 P."AFSTAND IS "N" KM.""
271 XC=(SIN(XZ)-SIN(XU)#COS(XD))
272 XC=XC/(COSXU#SINXD);XC=ACSXC;XC=XC#180/PI
273 P=X(XC+0.5);@=3
274 FIF XV>XY P=360-P
275 P."MYN ANTENNERICHTING : "P" GRADEN"
277 IF P<=180 P."ZYN ANTENNERICHTING : "P+180" GRADEN"
278 IF P>180 P."ZYN ANTENNERICHTING : "P-180" GRADEN"
280 IN."NOG EEN BEREKENING (JA/NEE) "#A
281 IF #A="JA" T.G.90
282 IF #A="NEE" T.E.
300 REM X-Y INDEX-ROUTINE
310 I=1;GOS.500
```

7

```

320 L=A; I=2; GOS. 500
340 B=A
350 F=(Q72)-48
360 H=(Q73)-48
370 IF F>7 AND H>0 OR F>8 G. 430
380 IF H=0 F=F-1; H=10
390 K=(Q74)-64
400 IF K>10 OR K=9 OR K<1 G. 430
410 S=SS(K); R=RR(K)
420 ZX=(40+B-F/B-S/24)*PI/180
425 FP.ZX*180/PI" GRADEN N.B."
430 ZY=(2*(L-1)+(H-1)/5+R/15)*PI/180
435 FIF ZY<0 FP.ZY*180/PI" GRADEN W.L." ;G. 440
436 FP.ZY*180/PI" GRADEN O.L."
440 R.
450 P."NIET BESTAANDE GTH-LOCATOR OF"
455 P." TIKFOUT !!!!!!!!!!"
456 DOP. 07; U.C.=10
460 IN."MOG EEN BEREKENING (JA/NEE)"*G
461 IF *G="JA" T.G. 80
462 IF *G="NEE" T.E.
500 REM OMZET SUBROUTINE
510 A=Q7(I-1)
515 IF A>90 OR A<65 G. 450
520 IF I=2 G. 550
530 IF A<=84 G. 570
540 G. 560
550 IF A<=85 G. 570
560 A=A-90; G. 580
570 A=A-64
580 R.
585 REM GMS OMZET ROUTINE
590 ZE=ZX*180/PI
595 @=0
610 E=ZE
630 XJ=(ZE-E)*60
640 J=XJ
660 XG=(XJ-J)*60
665 G=XG
670 FIF (XG-G)>=0.5 T. XE=XG+1
675 IF XG>=60 T. XJ=XJ+1
676 IF XG>=60 T. XE=XG-60
677 IF XJ>=60 T. XE=XE+1
678 IF XJ>=60 T. XJ=XJ-60
680 P. XE". "XJ". "XG" N.B."
690 ZT=ZY*180/PI
710 T=ZT
720 @=0
730 XM=(ZT-T)*60
740 M=XM
760 XW=(XM-M)*60
765 W=XW
770 FIF (XW-W)>=0.5 T. XN=XW+1
775 IF XW>=60 T. XM=XM+1
776 IF XW>=60 T. XW=XW-60
777 IF XM>=60 T. XT=XT+1
778 IF XM>=60 T. XM=XM-60
780 FIF XY>=0 T.P. XT". "XM". "XW" O.L."
790 FIF XY<0 T.P. XT". "XM". "XW" W.L."
800 R.

```

\*\*\*\*\*  
VERKLARENDE WOORDENLIJST MICROCOMPUTERS A t/m B N. SCHREURS  
\*\*\*\*\*

Met ingang van dit nummer zal dhr. Schreurs een rubriek verzorgen, die de meeste (engelstalige) termen die op microprocessors betrekking hebben in het nederlands verklaart.

ACCESS TIME - De tijd die verloopt tussen het aanbieden van een adres aan het geheugen en het toegang krijgen tot de inhoud van een geheugenlocatie.

ACCUMULATOR - Een speciaal register in de centrale verwerkingseenheid waarin de resultaten van de rekenkundige en logische bewerkingen worden geplaatst.

ACIA - Asynchronous Communication Interface Adapter. Invoer/uitvoerschakeling voor asynchrone communicatie, term wordt gebruikt door Motorola(Uart).

A/D-OMZETTER - Verzorgt de omzetting van een analoge spanning of stroom in een digitale voorstelling zoals die door de computer wordt gebruikt.

ADRES - Binair getal dat de plaats van een bepaalde geheugenlocatie of van een in- of uitgangspoort aanwijst. Het adresbereik van de meeste typen microprocessors loopt van 0 t/m 65535.

ALGOL - ALGOarithmic programming language. Hogere programmeertaal. In het bijzonder geschikt voor het oplossen van wiskundige problemen.

ALGORITHMIE - Stap-voor-stap weergave van de oplossingsmethode voor een probleem.

ALFANUMERIEK - Het totaal aan alfabetische en numerieke karakters. (alle letters en cijfers).

ALU - Arithmetic and Logic Unit. Schakeling in de processor waarin de rekenkundige en logische bewerkingen worden uitgevoerd.

ANALOG - Van toepassing op een spanning of stroom die elke willekeurige waarde kan aannemen.

ANALYZER - Apparaat dat verschillende signalen van een component, print of systeem meet en ze in een overzichtelijke vorm weergeeft voor nader onderzoek.

APL - A Programming Language. Hogere programmeertaal ontwikkeld door Iverson. De taal is interactief en wordt gebruikt voor het oplossen van wiskundige problemen.

ASCII - American Standard Code for Information Interchange; 7-bit code voor het weergeven van letters, cijfers en leestekens.



ASR - Automatic Send Receive. Een randapparaat dat zowel beschikt over een ponsbandlezer/ponser als over een toetsenbord en afdrukmechanisme.

ASSEMBLER - Programma dat een in mnemonische vorm (letters en cijfers) geschreven gebruikersprogramma vertaalt in machine-code die door de computer kan worden begrepen.

ASYNCHROON - Gebeurtenissen (signalen) die niet "in de pas" lopen met bewerkingen in de centrale verwerkingseenheid.

BANK - Hoeveelheid opeenvolgende geheugenlocaties of registers.

BAR CODE - Codering van gebruikersartikelen waarbij gebruik wordt gemaakt van verticale lijnen met verschillende dikten. Ontworpen om door een optische lezer te worden omgezet in elektrische signalen.

BASIC - Beginners All Purpose Symbolic Instruction Code. Oorspronkelijk ontwikkeld op het Dartmouth College in de VS voor onderwijsdoeleinden. Hogere programmeertaal die tegenwoordig op de meeste microcomputers beschikbaar is. Er zijn vele versies in omloop, die meestal niet geheel uitwisselbaar zijn.

BATCH - Werkwijze waarbij een programma wordt aangeboden aan de computer en het resultaat terug wordt gezonden aan de gebruiker. Communicatie tussen het programma en de gebruiker is niet mogelijk.

BAUD - Aantal binaire eenheden dat per seconde wordt getransporteerd. Komt meestal overeen met bit/seconde. Bij een Teletype bijv. is de snelheid 110 baud. Elk karakter is opgebouwd uit 11 bit en er worden dus 10 karakters per seconde overgezonden.

BAUDOT - Vrij oude code die nog wordt gebruikt bij telexmachines. Tegenwoordig wordt ook wel ASCII en EBCDIC gebruikt.

BCD - Binary Coded Decimal. 4-bit code voor de weergave van de decimale getallen 0 t/m 9. Zes van de 16 mogelijke coden worden niet gebruikt en de meeste typen microprocessors kennen een speciale instructie "decimal adjust" om bijv. een optelling toch correct uit te voeren, d.w.z. zodanig dat het resultaat een geldige code is.

BENCHMARK - Testprogramma om de snelheid van een computersysteem te kunnen bepalen. Een benchmarkprogramma moet zodanig zijn geschreven dat het representatief is voor de toepassing waarin de computer gebruikt gaat worden.

BI-DIRECTIONEEL - Over een en dezelfde lijn kan in twee richtingen data worden getransporteerd. Aan beide einden bevindt zich zowel een zender als een ontvanger. De meeste bi-directionele lijnen zijn uitgevoerd met 3-state buffers (3 toestanden: hoog, laag en hoog-ohmig).

**BIT** - Afkorting van Binary Digit. Een bit is de eenheid van informatie en kan de toestand 0 of 1 aannemen. Wordt gebruikt in digitale systemen voor het weergeven van informatie, instructies enz.

**BOOTSTRAP** - Programma dat vast in het geheugen van een computer is opgeslagen en dat de computer, na het inschakelen van de voedingsspanning, "opstart", d.w.z. registers met de beginwaarden vult of reset, in- en uitvoerapparaten initialiseert en het besturingsprogramma (monitor) inleest vanaf bijv. disk of cassette.

**BPI** - Bit Per Inch. Wordt gebruikt om de informatiedichtheid op een magneetschijf of een cassetteband aan te geven.

**BRANCH** - Vrijwel identiek aan de "JUMP"-instructie. Wordt gebruikt om het programma te vervolgen bij een andere dan de volgende instructie. Een branch-instructie maakt bij de meeste typen processoren gebruik van relatieve adressering; een "JUMP" maakt meestal gebruik van directe adressering.

**BREAKPOINT** - Plaats waar het programma door een ander programma of door een apparaat wordt onderbroken om register- en geheugeninhouden te kunnen onderzoeken. Deze methode wordt gebruikt tijdens het testen van een programma.

**BUBBLE MEMORY** - Magneetbellengeheugen waarin de informatie wordt weergegeven d.m.v. microscopische kleine magnetische gebieden (domeinen). Een magneetbellengeheugen is niet-vluchtig, d.w.z. het behoudt de informatie ook zonder voedingsspanning en er kan zowel informatie in worden geschreven als uit worden gelezen. Er zijn op dit moment magneetbellengeheugens met een capaciteit van 256 Kbit.

**BUG** - Een fout in een schakeling of programma. Het herstellen van deze fouten wordt "DEBUGGEN" genoemd.

**BUS** - Aantal parallel lopende lijnen waarover een bij elkaar behorende hoeveelheid informatiebits wordt getransporteerd. Een standaard microcomputer kent een adresbus, een databus en een besturingsbus.

**BYTE** - Eenheid van 8 bit. Een byte wordt meestal gebruikt om een karakter te coderen. Een instructie van een microcomputer is een, twee of drie bytes lang. Een byte bestaat uit twee nibbles (groepjes van 4 bits). .a

\*\*\*\*\*  
BEGINNERSHOEKJE E. SANDERS  
\*\*\*\*\*

## 7.2 BYTE VECTOREN

Het is soms geheugenverkwisting om ruimte te reserveren, voor getallen met een bepaald bereik, door middel van word arrays, zodat er nog een andere manier is om arrays voor te stellen, nl. een die voor ieder array element slechts een byte reserveert in plaats van 4 bytes.

Dit type array noemen we "byte vectoren" en ze hebben dezelfde uitwerking als 1 dimensionale arrays.

Byte vectoren verschillen van de word arrays doordat ze gebruik maken van de gewone variabelen A tm Z om het "basis" adres in op te slaan; d.w.z. het geheugenadres waarin het nulde element van de array zal komen. De array-subscript zijn stomweg "offsets" van dit basisadres; d.w.z. het subscript wordt opgeteld bij het basisadres om het geheugenadres van het betreffende element te verkrijgen.

Dus als  $A = \#2900$  dan is  $A?1 = \#2900 + 1 = \#2901$ .

De vectorelementen worden als volgt genoteerd:

$A?0, A?1, A?2, \dots$  enz.

waarin A dus de gewone variabele is om het basisadres van de vector in op te slaan en het getal wat na het vraagteken staat is het subscript.

Let op: het nulde element van een vector,  $A?0$ , is gelijk aan ?A, dus de inhoud van geheugenplaats A. Op diezelfde manier is  $A?1$  gelijk aan  $?(A+1)$  en zo verder.

Byte vectoren worden ook gedimensioneerd door het DIM statement, om bijv. een byte vector met de elementen  $A?0$  tot  $A?11$  te dimensioneren luidt de syntax:

```
DIM A(11)
```

Omdat het DIM statement arrays en vectoren dimensioneert vanaf het einde van een programma (+1) is het bovenstaande DIM statement gelijk aan:

```
T=TOP; A=T; T=T+12
```

waarin T de variabele is die bijhoudt waar de eerste vrije geheugenplaats is.

LET OP: ruimte voor vectoren kan overal in het geheugen worden gereserveerd, dit in tegenstelling tot arrays die alleen kunnen worden gereserveerd vanaf de TOP d.w.v. het DIM statement.

Om bijvoorbeeld ruimte toe te wijzen voor de vector S, welke moet corresponderen met het beeldschermgeheugen, luidt de syntax:

```
S=#8000
```

De elementen van de vector zouden dan kunnen corresponderen met plaatsen op het scherm; bijv.  $S?31$  is de plaats die correspondeert met de rechterbovenhoek van het scherm.

Ieder element van een byte vector kan een positief getal tussen 0 en 255 onthouden, of een enkel karakter.

Strings zijn derhalve in feite gewone byte vectoren die karakters bevatten.

Merk op dat het subscript van een byte vector een willekeurige expressie kan zijn, die tussen haakjes moet staan; bijv.  $A?(6*2*7)$ .

### 7.3 WORD VECTOREN

Een tweede voorstelling van word arrays is mogelijk in onze ACORN ATOM door gebruik te maken van de zgn. "word indirection operator" de "!". We noemen de word vectoren voor de volledigheid, want, om eenvoudige problemen op te lossen, is de word array (AA tm ZZ) gemakkelijker om mee te werken.

Word vectoren zijn gelijk aan de byte vectoren, echter ieder element van de vector bestaat uit een woord (in het geval van word vectoren een woord van 4 bytes) in plaats van 1 byte.

Ieder element bestaat uit de (basisadres) variabele, gescheiden van de subscript (of offset) door middel van een uitroepteken "!".

Denk eraan dat het subscript, voor ieder element, met 4 moet worden verhoogd, omdat ieder element(adres) 4 bytes verder ligt dan het voorgaande.

Bijv. een word vector W zou 6 elementen kunnen hebben:

```
W!0, W!4, W!8, W!12, W!16, W!20
```

Ruimte voor word vectoren kan door het DIM statement worden gedimensioneerd en laat 4 bytes per element toe; om bijv. geheugen voor 6 elementen te reserveren luidt de syntax:

```
DIM W(23)
```

Denk eraan dat het nulde element van de vector, W!0, gelijk is aan !W.

#### 7.3.1 PRIEMGETALLEN

Het volgende programma zoekt alle priemgetallen (priemgetallen zijn ondeelbare getallen) tussen 0 en 99999. Het maakt gebruik van een word vector om de priemgetallen die het programma reeds gevonden heeft in op te slaan en test alleen nieuwe getallen op hun deelbaarheid:

```
1 REM PRIME NUMBERS
10 @=8;S=4;Z=0;J=TOP;G=J;!G=3;P=G+S
20 FORT=3TO99999STEPZ
30 IFT%!G=Z G=J;N.
40 IFT>!G&!G G=G+S;G.c
50 F.T;!P=T;G=J;P=P+S;N.
60 END
```

Beschrijving van het programma:

- 10 Opzetten van de vector
- 20 Test alle oneven nummers
- 30 Indien deelbaar, pak dan de volgende
- 40 Hebben we genoeg delers geprobeerd?
- 50 Dit moet er een zijn - Print hem uit

Variabelen:

- !G - Deler die wordt getest
- J - In dit programma gelijk aa TOP
- !P - Vector met de delers
- S - Bytes per word
- T - Te onderzoeken getal
- Z - Constant 0

Programma lengte : 155 bytes  
Vector : als nodig

### 7.3.2 AANROEP DOOR VERWIJZING

Een groot voordeel van de word vectoren boven de word arrays is dat hun basisadres te gebruiken is als waarde en kunnen dus overgaan naar subroutines.

Het volgende programma moet dit duidelijk maken:

```
10 A=TOP; B=A+40
.
.
90 P=A; GOSUB p; REM Output A
94 P=B; GOSUB p; REM Output B
98 END
100pREM Print 10 Elementen van e array P
105 @=8; PRINT '
110 FOR J=0 TO 39 STEP 4
120 PRINT P!J
130 NEXT J
140 PRINT '
150 RETURN
```

In dit voorbeeld kan de subroutine p worden gebruikt om iedere array uit te printen door zijn basisadres over te laten gaan in de variabele P; dit procedee staat bekend als een "call by reference" (vrij vertaald: aanroep door verwijzing) omdat de subroutine een verwijzing geeft naar de array in plaats van naar de aktuele waarde in de array.

### 7.3.3 Willekeurige machtsverheffingen

Het volgende programma illustreert het gebruik van word vectoren om de waarde te berekenen van ieder getal dat verheven wordt tot een vrij te kiezen macht.

De limiet hangt alleen af van de beschikbare geheugenruimte.

Het programma slaat 4 decimale getallen op per word, zodat het produkt van 2 words geen overflow veroorzaakt. De uitkomst wordt dus een aaneenschakeling van word vector elementen en is dan ook een word vector.

```
1 REM ARBITRARY PRECISION POWERS
5 T=#3BFF
10 H=(T-TOP)/3; DIM P(H),S(H),D(H)
15 H=10000
20 @=0;PRINT" POWER PROGRAM"
30 PRINT" COMPUTES Y^X, WHERE X>0 AND Y>0"
40 INPUT" VALUE OF Y^Y," VALUE OF X^X
50 IFX<10RY<1PRINT" VALUE OUT OF RANGE";RUN
60 M=Y;N=X;GOSUBp
70 PRINT Y^^X="P!!P; IF!P<8 RUN
90 F.L=!P-4TD4STEP-4
95 IFL!P<1000P.0
100 IFL!P<100P.0
110 IFL!P<10P.0
120 P.L!P;N.;RUN
140#
200pJ=M; IFN%2=0J=1
```

210 R=P;GOS. e; J=M;R=S;GOS. e; IFN=1R.  
 250 B=S;DDA=B;GOS. n; B=E  
 255 N=N/2;A=P; IFN%2GOS. n; P=E  
 260 U.N<2;R.  
 280%  
 300m!D=!A+!B+4;F.J=4TD!D+4S.4  
 310 D!J=0;N.;W=D-4  
 320 F.J=4TD!B S.4;C=0;G=B!J  
 325 V=W+J;F.L=4TD!A S.4  
 330 G=A!L\*G+C+V!L;V!L=GXH  
 340 C=G/H;N.;V!L=C;N.  
 370 DD!D=!D-4;U.D!!D<>0;E=D;D=A;R.  
 380%  
 400e!R=0;DD!R=!R+4;R!!R=J%  
 410 J=J/H;U.J<1;R.

**Beschrijving van het programma:**

5 Zet in T de top van de lage geheugenruimte  
 10 Verdeel het geheugen tussen P, S en D  
 20-40 Lees de waarden van Y en X in  
 50 Zorg dat deze niet negatief zijn  
 60 Bereken de macht  
 70 Print het resultaat als het tenminste in een word past  
 90 Print de rest van het resultaat,  
 140 Legt regel (om de listing leesbaarder te maken)  
 200-260 p: Berekent de macht.  
 300-370 m: Vermenigvuldigt de vectoren, aangewezen door A en B en stop het resultaat in de vector aangewezen door D. De wijzers naar de vectoren worden gewijzigd. E bevat nu het resultaat.  
 400-410 e: Verdeel J in de vector aangewezen door R. Sla het aantal words op in !R.

**Variabelen:**

D!0 - Werkruimte vector  
 H - Rekenkundig grondtal  
 P!L - Vector voor het in stukken gehakte resultaat  
 !P - Aantal elementen gebruikt in P  
 S!0 - Werkruimte vector  
 T - Top van de beschikbare geheugenruimte.

Programmalengte : 733 bytes  
 Bijkomende opslagruimte : als beschikbaar

**Voorbeeld:**

>RUN

POWER PROGRAM  
 COMPUTES Y^X, WHERE X>0 AND Y>0  
 VALUE OF Y716  
 VALUE OF X764  
 16^64=1157  
 2089237316  
 95423570985008687907853269984665640564039457584007913129639936

### 7.3.4 VECTOREN VAN VECTOREN

Een tweede manier om 2 dimensionale arrays voor te stellen is mogelijk door gebruik te maken van ATOM's indirection operators "?" en "!"; op deze manier vermiijdt men de noodzaak om de subscripts via een vermenigvuldiging te moeten berekenen. Deze methode gebruikt echter een beetje meer geheugen. De clou is dat je een twee dimensionale array moet zien als vectoren van vectoren. Er wordt eerst de vector gemaakt die het adres van de rijen van de matrix bevat. Bijvoorbeeld we hebben een matrix genaamd X met de kolommen 0 tm M, en rijen 0 tm N. Het volgende statement zet de vector op voor de rij adressen:

```
DIM X(2*N-1)
FOR J=0 TO N*2 STEP 2; DIM Q(M); X!J=Q; NEXT J
```

Een word vector wordt dus gebruikt om het basis adres in op te slaan. Nu de vector van de rij adressen is opgezet is het element met de subscripts A,B:

```
X!(A*2)?B
```

Zo, dat was dan het hoofdstuk Arrays en vectoren. De volgende keer gaan we verder op de manier zoals het boek, dus vanaf hoofdstuk 1.

Zijn er nog vragen of problemen geef dan even een gil.

E. Sanders, Roermond

```
*****
ATOMBUS      E. SANDERS
*****
```

	A		B	
+5v	0	1	0	---
A15	0	2	0	---
A14	0	3	0	PB7
NWDS	0	4	0	PB6
NRDS	0	5	0	PB5
NRST	0	6	0	PB4
A8	0	7	0	PB3
A7	0	8	0	PB2
A6	0	9	0	PB1
A5	0	10	0	PB0
A4	0	11	0	CB2
A3	0	12	0	CB1
A2	0	13	0	PA7
A1	0	14	0	PA6
A0	0	15	0	PA5
D7	0	16	0	PA4
D6	0	17	0	PA3
D5	0	18	0	PA2
D4	0	19	0	PA1
D3	0	20	0	CA2
D2	0	21	0	CA1
D1	0	22	0	---
D0	0	23	0	---
A13	0	24	0	---
A12	0	25	0	RDY
A11	0	26	0	SD
A10	0	27	0	---
A9	0	28	0	IRQ
D2	0	29	0	NMI
R/W	0	30	0	SINC
BLK	0	31	0	---
0v	0	32	0	0v

Dit is dan onze ATOM-bus. Er zijn alle vele bussen geplubiceerd, maar deze vindt ik nog niet, vandaar.

Let er op dat op diverse AB connectors de aanduiding er niet correct opstaat.

Controleren dus!!!!

\*\*\*\*\*  
 GAS-STROOM-WATERVERBRUIK A. RONGEN  
 \*\*\*\*\*

Het volgende programma geeft over een aantal jaren exact het gas-water-stroom verbruik aan. Er wordt tevens een statistiek getekend.  
 Het programma is waarlijk de moeite waard.

Tevens is dit programma een voorbeeld zoals het zou moeten:  
 1) het geeft aan hoe het is aan te passen voor eigen gebruik  
 2) aangegeven wordt welke vreedse kommando's gebruikt worden  
 3) veel REM statements  
 4) en heel belangrijk: iedereen kan het zonder gebruiksaanwijzing gebruiken.  
 Goed zo !!

```

10REM VOOR EIGEN GEBRUIK DE REBELS 30,80,230,240,280,290,300
15REM EN NATUURLIJK DE DATA AANPASSEN
20REM GAS-STROOM-WATER VERBRUIK
30REM VAN 1978 T/M 1984
50REM O IS HET LAAGSTE JAARTAL
60REM B IS HET HOOGSTE JAARTAL
65REM Y BEPAALT DE SCHAALVERDELING
70REM FUNCTIES;KEY, READ, DATA, RESTORE, ON ERROR
80O=1978;B=1984
90M=(B-O+1)*12-1
100DIM G(M),EE(M),WW(M)
110DIM CC(11);DING(10),S(10),H(10),A(5),X(5),D(6);REB.
120FORN=OTOM;READG(N)
130NEXTN
140FORN=OTOM;READEE(N)
150NEXTN
160FORN=OTOM;READWW(N)
170NEXTN
180G$="GAS";$S="STROOM";$H="WATER"
190P.$12;P."DIT PROGRAMMA GEEFT HET VERBRUIK ""
200P."VAN (M3) GAS""
210P." (KWH) STROOM""
220P." (M3) WATER""
230P."OVER DE JAREN 1978 T/M 1984""
240P."VAN HET PAND: PUTSTR.54 BORN""
250P."WELK VERBRUIK MENST U GAS/ELEKTRO/WATER: ""
260P."BEDIEN TOETS E/E/W"
270KEY1
280IF I=CH"G";Y=50;Z=Y/3; $X="M3";$Q=$G;G.320
290IF I=CH"E";Y=50;Z=Y/3; $X="KWH";$Q=$S;G.320
300IF I=CH"W";Y=3;Z=Y/3; $X="M3";$Q=$H;G.320
310G.270
320P.$12;
330DN.G.340
340IN."OVER WELK JAAR", $A
350E=VAL $A
360IF E < O;G.320
370IF E > B;G.320
380F=(E-O)*12
390FOR L=0 TO 11

```



```

400IFI=CH"G";CC(L)=GG(F+L)
410IFI=CH"E";CC(L)=EE(F+L)
420IFI=CH"W";CC(L)=WW(F+L)
43ONEXT L
435REM TEKENEN VAN HET STAAFDIAGRAM
436REM DE HAANDEN VERTIKAAL AFLEZEN
440P.#12;CLEAR O; @=5
450MOVE @0,15;DRAW12,15;DRAW12,46
460FOR P =11 TO 0 STEP -1
470IFP=10;P.#11
480IF P#2=0;P. P#Y
490P.#;NEXTP;P.#11
500P.#X;P.#
5107#EO=7;P."J F M A M J J A S O N D "
520P.#@
5307#EO=7; P."A E R P E U U U E K D E"
540P.#@;P." N E T R I N L G P T V E"
550C=0
560FOR N =0 TO 11; J=15+4#N
570MOVE J,15;DRAW J,(CC(N)/Z +15)
580C=C+CC(N)
590NEXT N
600P."TOTAAL JAARVERBRUIK : ",C," ",#X"
610P."VOOR VERVOLG BEDIEN EEN TOETS "
620LINK #FFE3
625REM JAARVERBRUIK IN EEN TABEL
630P.#12;P.#@,"VERBR. PER MAAND IN ",E"
640P."JAN. : ",CC(0)," ",#X"
650P."FEB. : ",CC(1)," ",#X"
660P."MRT. : ",CC(2)," ",#X"
670P."APR. : ",CC(3)," ",#X"
680P."MEI. : ",CC(4)," ",#X"
690P."JUN. : ",CC(5)," ",#X"
700P."JUL. : ",CC(6)," ",#X"
710P."AUG. : ",CC(7)," ",#X"
720P."SEP. : ",CC(8)," ",#X"
730P."OKT. : ",CC(9)," ",#X"
740P."NOV. : ",CC(10)," ",#X"
750P."DEC. : ",CC(11)," ",#X"
760P."TOTAAL ",C," ",#X"
770P."WILT U EEN TOTAAL OVERZICHT J/W"
780KEY1;IFI=CH"J";G.#10
790IFI=CH"N";P.#12;E.#180
800G.#750
805REM TOTAAL OVERZICHT VAN ALLE VERBRUIK OVER ALLE JAREN
810P.#12;P." JAAR GAS STROOM WATER"
820P." (M3) (KWH) (M3)"
830F.E=0 TO B;F=(E-0)#12
840U=0;V=0;W=0
850F.L=0 TO 11;U=U+GG(F+L);V=V+EE(F+L);W=W+WW(F+L);N.L
860@=7;P.E,U,V,W"
870N.E
880P.#"
890P."VOOR VERVOLG BEDIEN "N TOETS";LINK#FFE3;P.#12;E.#180
900REM DATA GAS 1978/1984 IN M3
910DA.500,410,328,250,120, 20, 15, 25, 40,156,270,355
920DA.527,411,294,179,109, 19, 12, 23, 72,149,312,302
930DA.504,321,348,252,136, 76, 40, 43, 55,195,311,379

```

```

940DA.385,358,239,174, 89, 49,          38, 35, 61,168,217,423
950DA.460,390,341,239,132, 42,          41, 28, 54,159,254,372
960DA.312,304,296,196,156, 49,          34, 36, 72,148,239,346
970DA.350,341,342,215,157, 62,          34,000,000,000,000,000
980REM DATA STROOM 1978/1984 IN KWH
990DA.230,200,230,190,170,141,          85,145,160,198,192,222
1000DA.214,191,204,174,176,111,          78,140,171,221,275,218
1010DA.455,393,391,296,297,232,          147,195,276,364,365,370
1020DA.360,310,341,289,238,194,          196,173,193,218,246,276
1030DA.290,312,310,251,228,219,          210,174,257,271,268,304
1040DA.303,267,268,235,227,196,          184,159,232,238,297,386
1050DA.324,288,289,270,253,204,          152,000,000,000,000,000
1060REM DATA WATER 1978/1984 IN M3
1070DA.14,12,13,8,8,12,3,10,4,7,7,9,12,8,7,8,9,11,6,10,9,12,15
1080DA.13,14,12,13,13,28,14,8,15,14,14,14,14,13,11,16,15,17,17
1090DA.19,13,16,16,17,17,17
1100DA.19,19,18,22,19,23,18,21,19,19,19
1110DA.21,17,21,19,24,36,35,25,19,19,20,20
1120DA.18,19,20,24,22,23,30,00,00,00,00,00
O(C) A RONGEN 1984
1130END

```

\*\*\*\*\*  
DE NIEUWE 128K GEHEUGENKAART H. HEUTS  
\*\*\*\*\*  
BETREFFENDE GEHEUGENUITBREIDINGEN E. D.

Wie met programma's als bv. de INFOMASTER gaat werken komt al gauw tot de ontdekking dat het standaard aanwezige geheugen toch te beperkt is om met grote hoeveelheden gegevens te werken.

Ook voor toepassingen waarbij veelvuldig van het DIN-statement gebruik gemaakt wordt, kan een geheugenuitbreiding zeer wenselijk zijn. Tot voor kort bestond de mogelijkheid om 64K-kaarten en zelfs 128K-kaarten te kopen. De prijzen van deze kaarten echter waren behoorlijk hoog. Voor 64K betaalde men 495 gulden en voor een 128K-kaart, -echter maar met 64K bestukt-, betaalde men 695,-. Voor de tweede 64K moest dan nog eens 350 gulden worden neergeteld, zodat de totale prijs ruim over de 1000 piek kwam.

Vanaf de volgende club-avond kan een compleet geteste geheugenkaart worden gekocht met de volgende specificaties:

- 512 K maximaal
- 128 K aanwezig in voeties
- uitgedecodeerd in 4 banken van 32 K
- rechtstreeks aan te sluiten op de databus
- uitbreidingen in stappen van 64 K
- prijs per 64 K ongeveer 200,-
- doorkoppeling van meerdere kaarten tot maximaal 16 Megabyte
- speciale EPROM met handige routines bij de prijs inbegrepen

De prijs ligt rond de 600 gulden (gebouwd en geteste kaart, incl. 128 K en EPROM).

Informatie bij:  
Hans Heuts, Kleine Steeg 44, Sittard, tel.: 04490-19969